



The Fundamentals of Systems Engineering

5-day Course

Waters Confidential

The Fundamentals of Systems Engineering

Systems Engineering Short Course

CONTENTS

Introduction: Aims and Objectives

Systems Theory

1. This is Systems Engineering

A Systems Approach to Requirements

2. Requirements Engineering

3. Determining System Requirements

4. Gathering Requirements

5. Analysing Requirements

6. Need Means Analysis

7. Systemic Textual Analysis

8. Functional Determination

9. Functional Modelling

10. Sensitivity and Failure Analysis

11. Quality Function Deployment phase 1

12. Specifying Requirements

A Systems Approach to Systems Design

13. Systems Concept Design

14. System Architecting

15. Technological Concept Generation,

16. Concept Integration

17. Concept Selection

18. Quality Function Deployment phase 2

A Systems Approach to System Optimisation

19. A Systems Approach to Robust Design

A Systems Approach to Verification and Validation

20. System Verification and Validation

Copyright Notice: The work is unpublished but copyright exists and is held by the Burge Hughes Walsh Partnership. This work must not be copied, distributed or otherwise used without the express permission of BHW

Purpose of the Course

To help Waters deliver excellence

by:

- introducing the concepts and benefits of applying Systems Thinking to the creation of a new, or modification to an existing, system - aka Systems Engineering
- emphasising the importance of integrating people, process and tools in the design of systems
- showing how systems engineering is applicable to both product and service (process) based systems
- demonstrating the use of Systems Engineering tools to provide participants with a common toolset
- enabling participants to identify stakeholders, elicit their requirements and translate these into specific, precise and measurable technical requirements
- enabling participants to generate and down-select alternative design concepts
- demonstrating the systems approach to robust optimisation (how SE feeds Robust Design)
- developing some competencies required for successful team projects
- considering the impact on future business of adopting a systems approach to engineering

Agenda

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

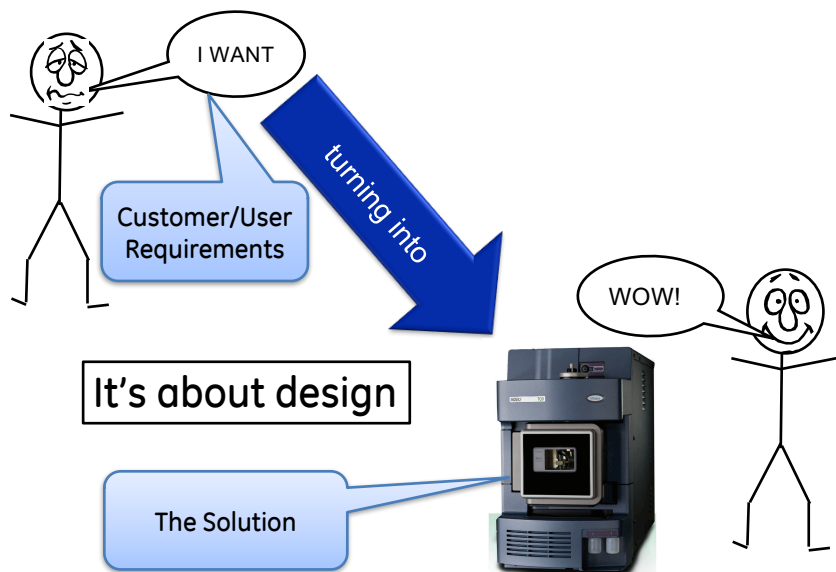
Monday	Tuesday	Wednesday	Thursday	Friday
0830 Introductions, Aims and Agenda				System Concept selection and evaluation using a Pugh Matrix
1000 This is Systems Engineering	Gathering Requirements	Deducing Unspoken Basic Requirements using Functional or Viewpoint Analysis	Verifying and Validating Requirements using Quality Function Deployment	Verifying the System Design Concept using QFD 2
1100	Analysing Requirements	Understanding the concept of operation using Functional Modelling	Specifying Requirements	A Systems Approach to Optimisation and Robustness
1200	Lunch	Lunch	Lunch	Lunch
1230	A Systems Approach to Requirements Engineering	Understanding the concept of operation using Functional Modelling	A Systems Approach to Systems Design	A Systems Approach to Verification and Validation
1500	Determine Systems Requirements	Analysing Expressed Requirements using Systemic Textual Analysis	System Architecting: Finding the best Architecture	Summary, Review and close
1700		Exploring undesirable behaviour using Sensitivity and Failure Analysis	Generating System Concepts	

This is Systems Engineering

This Session

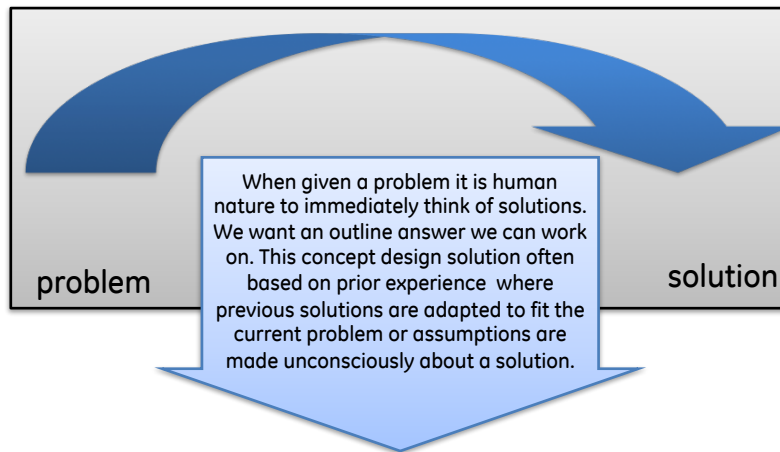
- Defines Systems Engineering and its benefits
- Explains the relationship between Systems Thinking, Approach and Engineering
- Introduces Systems Thinking as the foundation of Systems Engineering
- Provides an overview of Systems Thinking
- Introduces the basic Systems Engineering process, tools and role of people

In Simple Terms – Systems Engineering is about:



Traditional Design

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

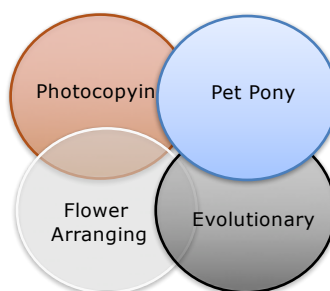


©2016 Waters Corporation

3

Approaches – The Choices

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

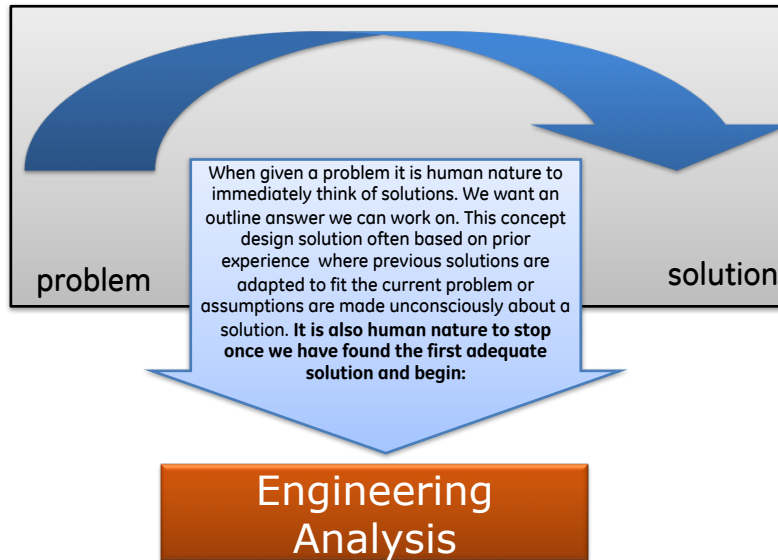


©2016 Waters Corporation

4

Analysis to the Rescue

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



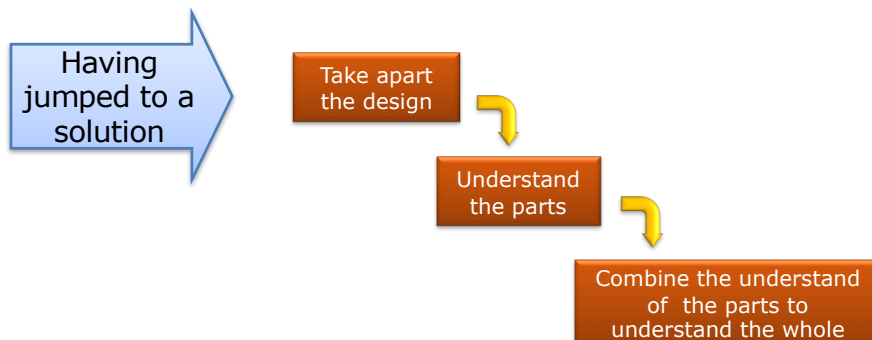
©2016 Waters Corporation

5

Analysis and Engineering Design

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Analysis is the core of traditional engineering design where we use engineering science to understand



©2016 Waters Corporation

6



Conclusion

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- We no longer have the time or the money to rely on traditional “solutioneering” approaches (1 in 10 chance of getting it right¹)
- We need:
 - to identify and fix issues early
 - a through-life view – design for the life cycle
 - a concurrent (Product, Manufacturing, Operation, Support, Disposal) view
 - a customer focused view to deliver performance, quality and cost
- We need a Systems Approach to Systems Design

¹Elm J.P and Goldenson D.R “The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey” November 2012 SPECIAL REPORT, CMU/SEI-2012-SR-009, Carnegie Mellon University

©2016 Waters Corporation 8

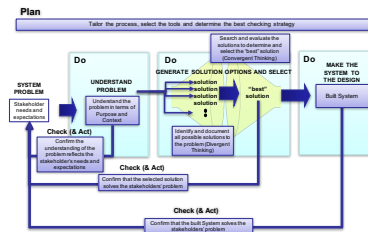
A Systems Approach to Systems Design



Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ Rigorous process based on Systems Thinking



■ Benefits

- 97% success rate (Carnegie Mellon University, 2012)

■ Issues

- Perceived to be slow – but actually faster because of low rate of avoidable re-design

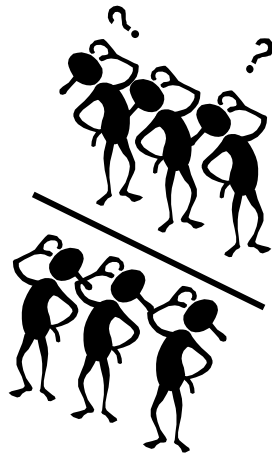
©2016 Waters Corporation

9

Consequence of Solutioneering?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Form two lines standing opposite each other
- Each team member should use one finger to support the paper tube from the underside
- Keeping every finger in constant contact with **underside** of the tube, as a team, lower it to the ground
- Time 10 minutes

©2016 Waters Corporation

10

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

The “helium” tube exercise illustrates what is called

Emergent Behaviour

.. and is why
Systems Engineering
is absolutely essential

Note:

1. Changing a person would have little effect on the behaviour of the team
2. Rhetoric will have little effect
3. Organization of the team has the biggest effect

©2016 Waters Corporation 11

Emergence and Emergent Behaviour

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



The properties of the pen as a whole are not found in any of the individual components. Only when the components are assembled in the right sequence and the right place do we get the desirable emergent behaviour of being able to make marks on paper

All systems display emergent behaviour – this can be DESIRABLE and/or UNDESIRABLE!

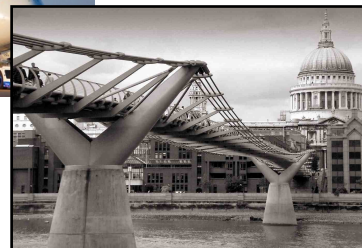


©2016 Waters Corporation 12

A Few Problems

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



In all these examples the issues were:

- Discovered late on: - they emerged
- Expensive: - because they were discovered late
- Avoidable:

©2016 Waters Corporation

13

Desirable and Undesirable Emergence

Waters

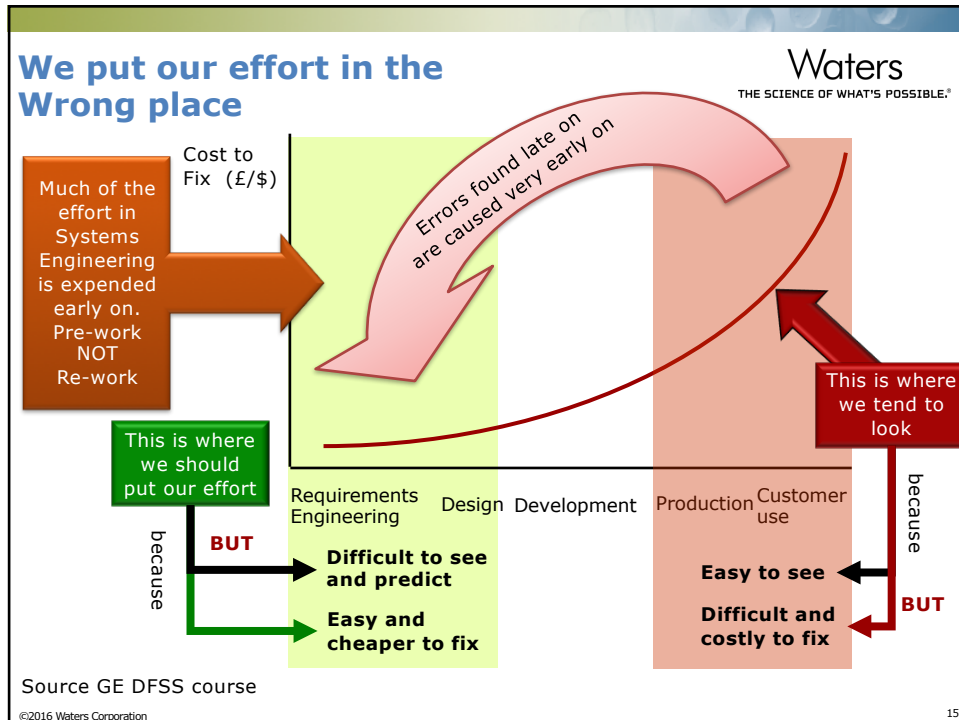
THE SCIENCE OF WHAT'S POSSIBLE.®

- We **DESIRE** emergent behaviour in our products and services but we can also get **UNDESIRABLE** emergent behaviour
- Undesirable emergent behaviour has three causes
 - **Variation**: unexpected cliff edge performance where small variations cause a significant change in behaviour
 - **Sub-optimisation**: unexpected low performance – the mistaken belief that the “best” system must have the “best components”
 - **Dynamic**: unexpected time related behaviour - we assume things will carry on in the same way – they don’t
- Emergent behaviour is a consequence of bringing together the components of a system. Therefore to understand and predict emergent behaviour we need to consider the WHOLE and how the components INTERACT with each other. This is obvious

BUT

©2016 Waters Corporation

14



Why is it Difficult to See Issues Early?

- When undesirable emergent behaviour appears, ***hindsight shows the actual cause to be "obvious"***

So why did we not spot it earlier?

- There are several interrelated reasons
 - Complexity**
 - Functional Divides
 - Mind set**
 - Project Management/Life Cycle Management

We are going to explore a couple of the reasons

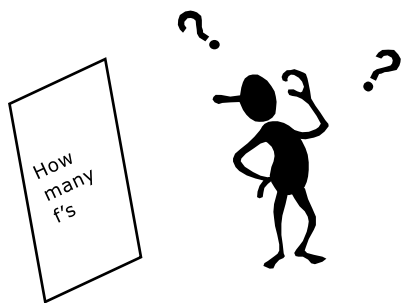
©2016 Waters Corporation

16

Exercise: Complexity and Humans

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- You are about to be shown a series of 4 slides
- Each slide presents a sentence
- Your task is to count the number of occurrences of the letter "f" in each sentence
- Each sentence will become longer – but you will be given pro-rata more time in which to count
- Time 5 minutes

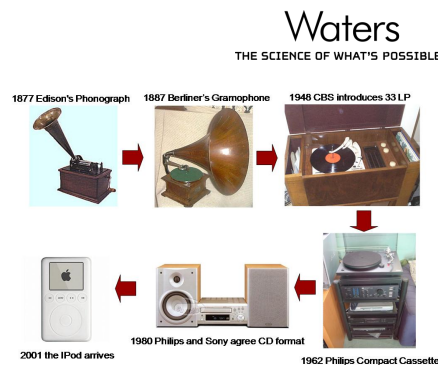
Slide	A	B	C	D
Count				

©2016 Waters Corporation

17

Complexity

- Complexity lies at the heart of the problem because humans cannot handle complexity
- In order to satisfy customer needs and expectations, organizations are designing increasingly more complex products and services
- This increased complexity has resulted in problems and even failures that demonstrate symptoms of:
 - Late delivery
 - Cost overruns



Pick any product, including service based products, and as it evolved its complexity will have increased

The increased complexity increases the opportunity for an error and makes errors more difficult to find. These errors, if undetected, will result in undesirable emergent behaviour

©2016 Waters Corporation

18

Our Brain

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Miller showed that this has a processing limitation of 7 ± 2

But we can "chunk" – grouping information together into chunks of 7 ± 2

Pre-frontal Cortex



Limbic System

Brain Stem

©2016 Waters Corporation

19

Emergence is the Future

Waters

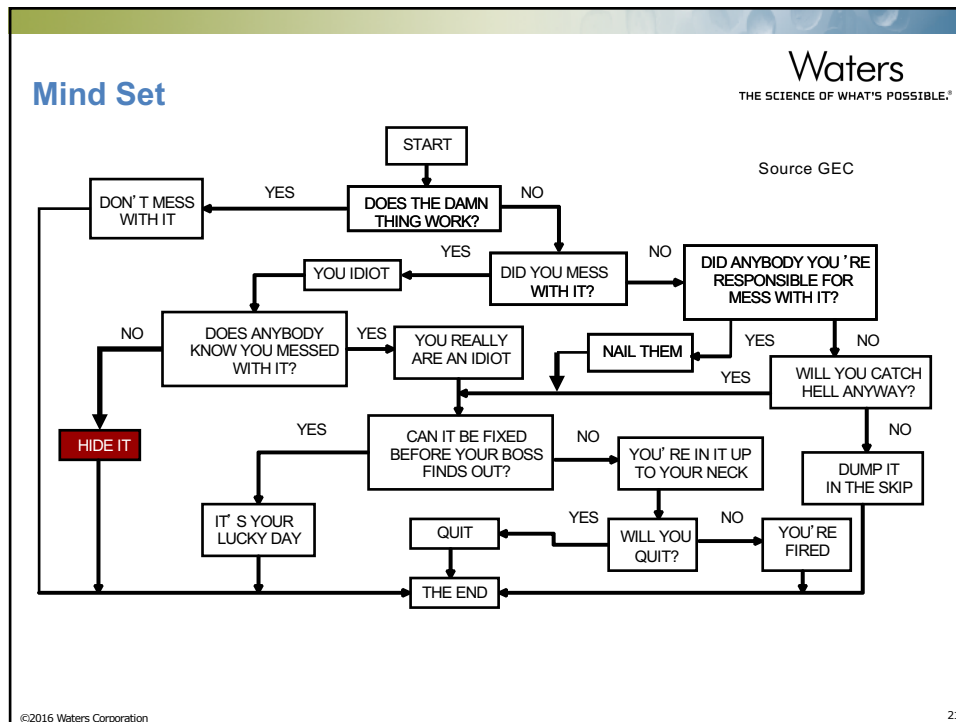
THE SCIENCE OF WHAT'S POSSIBLE.®

- Products and services WILL get more complex and therefore issues of undesirable emergent behaviour are going to INCREASE
- Organizations no longer provide just a product or service but a cradle to grave solution comprising several interrelated and interdependent systems each of which is becoming increasingly more complex compounding the issues of emergent behaviour

Unless we do something

©2016 Waters Corporation

20



Mind Set

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Mind Set plays an important role at two levels:
 - Individual level - e.g. “FLY” locked-in thinking
 - Organizational – e.g. “hide it”
- As products, processes and services get more complex and functional divides persist, mind set is an increasingly important factor in not uncovering undesirable emergent behaviour

Quote from NASA
“The biggest advantage of not doing Systems Engineering is that failure takes us by surprise”

Unless we do something

©2016 Waters Corporation

22

Complexity, Mind Set and Emergence

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

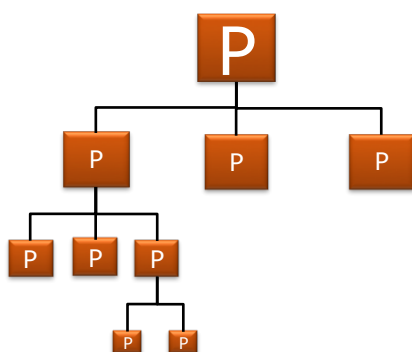
- Human kind has only developed two approaches to handling complexity:
 - Reductionism
 - Systems Thinking

©2016 Waters Corporation

23

Managing Complexity: Reductionism

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



- Rene Descartes
 - “I think Therefore I am”
- Discourse on Method and the Meditations
 - *“to divide each of the difficulties under examination into as many parts as possible, and as might be necessary for its adequate solution.”*

©2016 Waters Corporation

24

Handling Emergence

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

• The only known way to handle issues of emergence, caused by complexity (and the other reasons) is to view the situation (the product, process, or service) as a system and apply Systems Thinking in an organized and disciplined way ... and do it early

The beauty of Systems thinking is that it provides a very powerful framework for understanding complex situations by:

- simplifying and clarifying situations to understand the key aspects
- allowing different aspects and perceptions to be portrayed and considered simultaneously to understand and predict emergent behaviour
- looking for dependencies and influences between elements to understand and predict emergent behaviour
- bringing experience to bear upon a situation to understand and predict emergent behaviour

What is Systems Thinking

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Applying the concept of a system to a situation in order to gain insight and understanding

Who does Systems Thinking

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Anybody

- Systems Thinking has and can be applied to every discipline
- Some of mankind's greatest achievements have come about from systems thinking:
 - Isaac Newton
 - John Maynard Keynes
 - Albert Einstein
 - Charles Darwin
 - Barnes Wallis
 - W Edwards Deming

Most humans are not natural Systems Thinkers, we have to work hard to do Systems Thinking

©2016 Waters Corporation

27

Systems Thinking, Systems Approach and Systems Engineering

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Systems Thinking is:

Applying the concept of a system to a situation in order to gain insight and understanding

Systems Approach is:

Applying Systems Thinking in a systematic and repeatable manner

Systems Engineering is:

Applying a Systems Approach to the creation of a system

©2016 Waters Corporation

28

The Beauty and Power of Systems Thinking

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

In Systems Thinking we treat the situation as a system

Product, Process, Service



We can do this even if the situation is NOT a system! This makes it truly universal

We assume that the situation has all the properties and characteristics of a system and explore these to gain more understanding and make predictions about behaviour both desirable and undesirable

Let's have a look at some of these properties and characteristics of a system

29

The Basis of Systems Thinking: The System

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

An assembly of components, connected together in an organized way to achieve something that is a consequence of the whole

Peter Checkland - Lancaster University

This definition of a system allows for the development of a number of characteristics and properties.

Note: this is not the only definition of a system there are many others but, generally, they all agree that a system comprises three key ingredients:

1. Components
2. Interconnections
3. Purpose

30

Systems Thinking: Some Key Concepts

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Anything can be treated as a system
 - A system is an assembly of components, connected together in an organized way to form a whole, this showing properties of the whole, rather than properties of the components
- Systems are made up of Subsystems
 - Whenever we look “inside” the chosen system of interest we see further smaller systems or subsystems
 - Whenever we look outside the chosen system of interest we see that it is part of a larger system
- All systems seek to achieve a PURPOSE
 - Every system has a purpose that defines fundamentally what it has to do
 - Humans tend not to think about purpose explicitly – but implicitly through the thing that achieves the purpose
- All systems have CONTEXT
 - Every system has context that defines the environment and constraints that impact upon that system
 - System context often restricts the solution options that satisfy the purpose

©2016 Waters Corporation

31

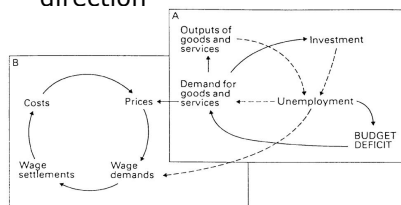
One More Key Systems Concept

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Everything is connected to everything else

- **Reinforcing (positive) Feedback:** successive changes add to previous changes in the same direction
- **Balancing (negative) Feedback:** resist change in one direction by producing change in the opposite direction



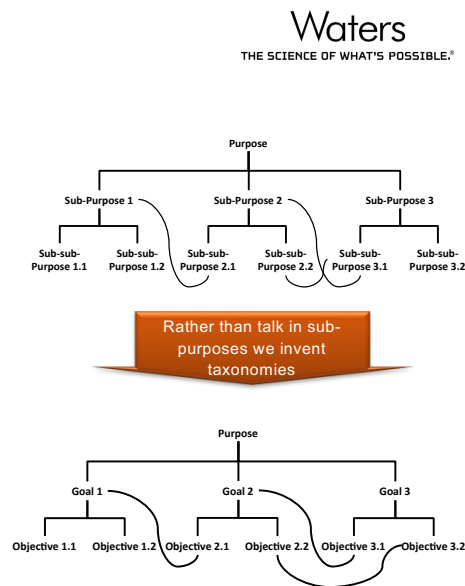
The whole is greater (or less than) the sum of parts not because of the parts, but because of the connections between the parts. Systems Thinkers care more about what happens between the components of the system than the components themselves

©2016 Waters Corporation

32

System Purpose

- All systems seek to achieve a purpose
- The purpose is a property of the whole and not found in any of the components
- Some systems have more than one purpose
- The purpose demands the achievement of lower level purposes as a consequence of the immutable laws of the universe (physics, chemistry etc.)
- These lower level purposes in turn demand even lower level purposes which BUT it is rarely a simple hierarchy
- People who use, or are part of, a system are often totally unaware of its purpose. Moreover, when asked, different users have different, and even sometimes conflicting, views as to the exact nature of the purpose



©2016 Waters Corporation

33

Purposeful Thinking

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Defining the purpose of a system is not easy!

The purpose of a system is often a sentence or two – it's easier to refer to the object!

WE ARE OBJECT ORIENTED we have been brought up from a early age to talk about things and NOT what they DO

**IN SYSTEMS THINKING (and ENGINEERING)
WE THINK ABOUT THE**

PURPOSE

©2016 Waters Corporation

34

A key question when Systems Thinking is

What is the purpose?

System Context

- All systems have context which depends on:
 - The purpose
 - The application (the bigger task or job which the system is part of)
 - The environment of the system
 - The observer
 - an understanding of a system depends upon the perception of the observer. A tax payer will have a different perception of the "Tax System" to the tax collector or government official who is going to spend the tax revenue.

Does Context Matter

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



The PURPOSE is the same in each situation "turn thoughts into marks on paper"
BUT can the same pen solution be used in all three situations?

Context matters and we need to be able to understand it by defining the extent of the system of interest and its environment

©2016 Waters Corporation

37

Context is more than just the environment

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Recording event in a journal



Flip Chart in a Training class



Record of birth in a Registry Office

The purpose is the same, but the bigger need or application is different

©2016 Waters Corporation

38

Properties and Characteristics of Systems System Boundary and Environment

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Those parts of the outside world which do NOT significantly interact with the system are part of its universe. These can be physically very close

The environment of a system is that part of the outside world which significantly interacts with it and impacts upon it. Usually by being the source of system inputs and destination of system outputs. A system's environment is also a major contributor to the context of that system.

Universe

Environment

System Boundary

A system has a boundary that defines the extent of the system.



©2016 Waters Corporation

39

What's the System Here?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Defining what is and what is not part of the system of interest is not easy

- A system boundary is an artificial (a mental creation) but useful concept in Systems Thinking.
- It provides a way of "bounding" or "scoping" the situation of interest. It helps decide what is inside and what is outside the system
- System boundaries can be temporal (time related), logical (purpose), or geographical (physical)
- The choice of what is and what is not the chosen system of interest depends upon the observer and their **reason** for examining that system

©2016 Waters Corporation

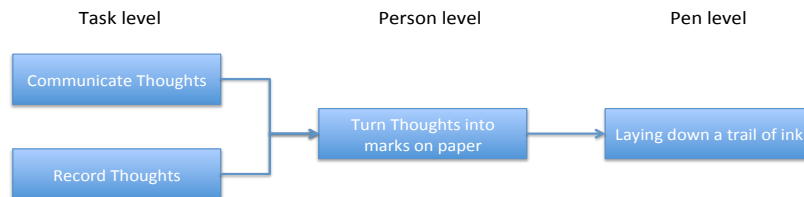
40

Using Boundaries

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The "right" boundary for thinking about a situation rarely coincides with existing political or physical definitions
- We invent boundaries in order to simplify and clarify a complex situation – to help decide "what's in" and "what's out"
- Where we draw the boundary impacts both PURPOSE and CONTEXT



- We should always reconsider the boundary for each situation whether new or old
- We should also be prepared to change the boundary as our understanding grows. Sometimes we don't fully know why we are investigating a situation! We "have a feeling" – and its only by exploring the situation as a system we begin to appreciate what needs to be done which in turn will define the boundary

©2016 Waters Corporation

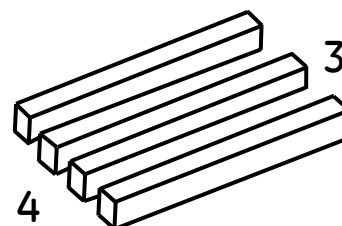
41

The Observer

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The context of a system also depends on who's looking at it
- We see different things
- When exploring a system's context we need to take multiple perspectives to increase our understanding



©2016 Waters Corporation

42

A key question when Systems Thinking is

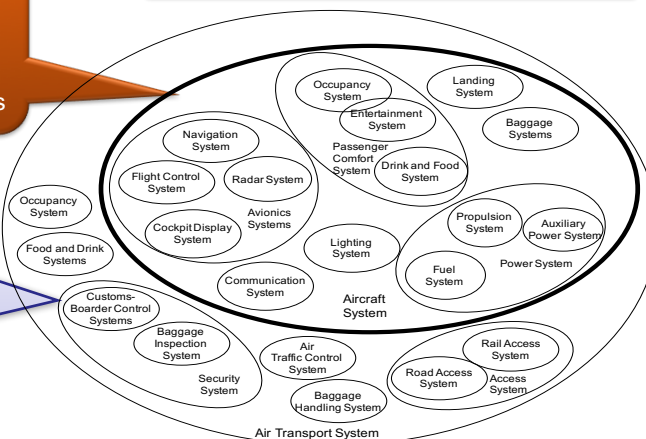
What is the context?

**Systems are made up of subsystems
and Systems are subsystems of bigger Systems**

We can use this concept to manage complexity through the use of subsystems

Whenever we look "inside" the chosen system of interest we see further smaller systems or subsystems

Whenever we look outside the chosen system of interest we see that it is part of a larger system

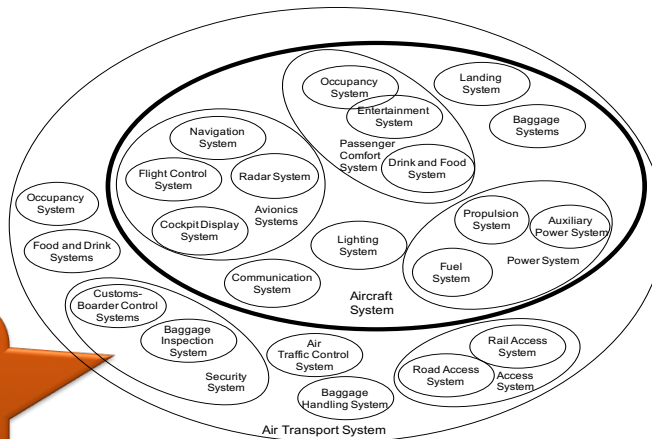


The context of a subsystem comes from the system it sits within

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

At any level it is the other systems outside the boundary of our system of interest that provide its context. These other systems will provide constraints (requirements) on our system of interest



45

Subsystems and Information Hiding

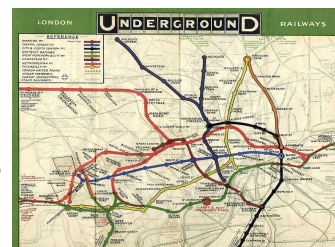
Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The fact that any system is made up of subsystems is extremely useful in Systems Thinking.
- By “hiding” the detail in sub-systems it provides a way of managing complexity
- This “abstraction” of reality is very powerful – it allows complex situations to be explained in simple ways

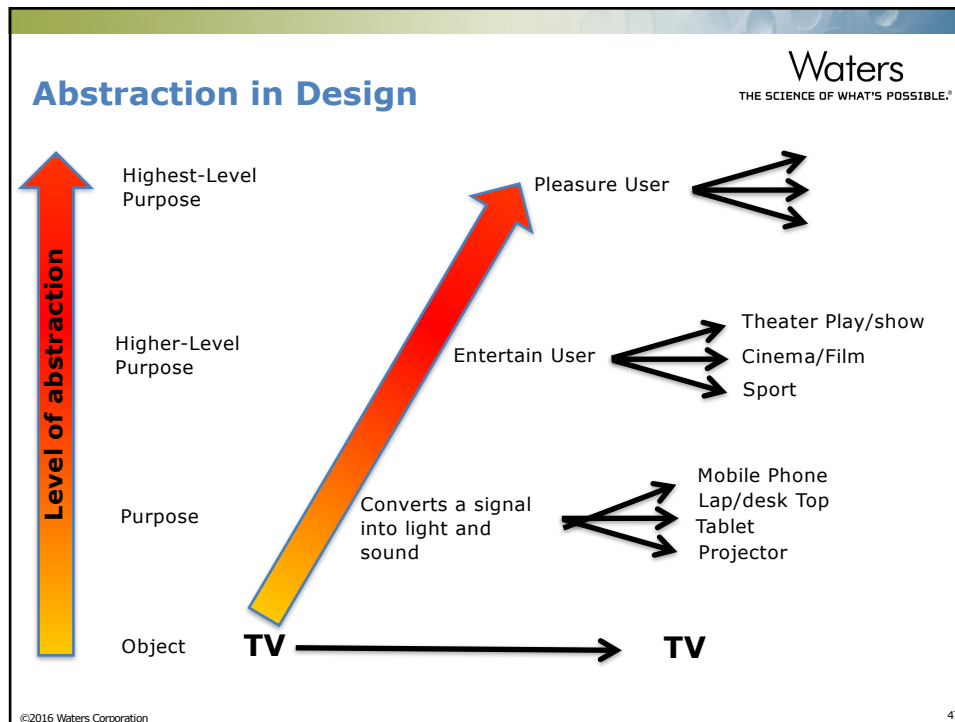


The map of the London underground is a classic example of abstraction to make complexity understandable. The “real” map is just too complex to be of any use in navigating the system.



©2016 Waters Corporation

46



Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

A key question when Systems Thinking is

What's inside? What's Outside?

©2016 Waters Corporation 48

System Behaviour and Connections

- Many Systems (particularly Business and Engineering Systems) are often concerned with a transformation – converting inputs to outputs. How the output varies over time is called the system behaviour

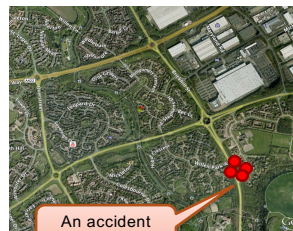


- The behaviour of a system manifests itself in two related ways:
 - Events** are things that happen at a certain point in time. Events are the output of a system moment by moment. For example a commercial aircraft retracting its undercarriage can be treated as an event, so can a road traffic accident or the purchase of a new business unit.
 - Patterns** are an accumulation of events that display a recurring trend over time. For example a commercial aircraft pushes off from the stand, taxis, accelerates down the runway, rotates, becomes airborne, retracts its undercarriage, etc. in a series of events that defines a pattern of behaviour repeated by most commercial aircraft.

All systems will display patterns of behaviour if we observe them for long enough.

Events, Patterns and Behaviour

- Humans find events more memorable and often view reality just in terms of events. If you were involved in a road traffic accident – you might call the local police and exchange driver details. Indeed, it's our ability to react to events that has ensured our survival!
- Patterns or "trends" over time offer a deeper understanding of reality. It's our ability to spot patterns that has lifted us from "just surviving" – but whereas reacting to events is instinctive, finding patterns is not! Reducing road traffic accidents at the pattern level could involve finding accident "hot spots" in terms of location and time and staffing the emergency services accordingly
- BUT something is behind, or causing, the patterns – the System Structure**

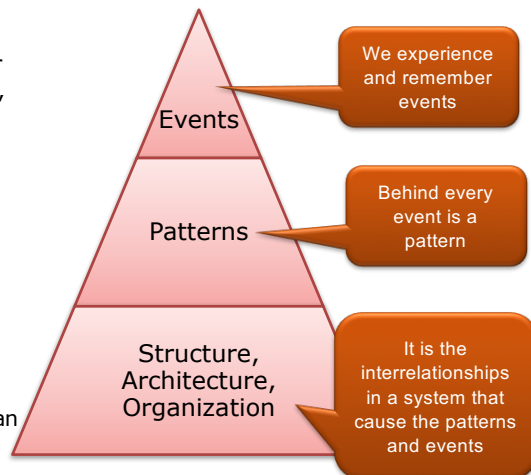


Interrelationships, Structure and Behaviour

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- To fully understand a system we need to examine its behaviour over time. This behaviour can be simple, complex, desirable or undesirable but it is due to how the system components are interrelated – the system structure, organization or architecture
- Because "Structure" is defined by the interrelationships of a system's components, and not the components themselves, the structure is invisible
- It is by changing the structure or organization of a system that we can **significantly affect** its behaviour



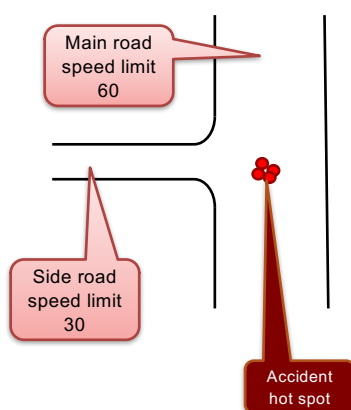
©2016 Waters Corporation

51

Change the Structure – the Systems Intervention

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



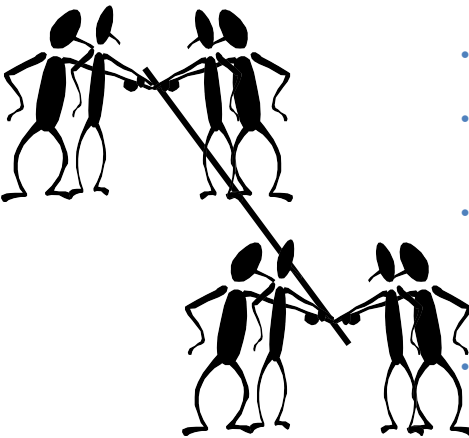
- What could we do to reduce or remove this accident hot spot?
- Ideas?

Reacting to events and patterns can make a difference but the real differences come with changes in the structure of the system

©2016 Waters Corporation

52

Exercise




THE SCIENCE OF WHAT'S POSSIBLE.®

- Same people who did the "Helium" tube exercise
- Organize yourselves into two equal groups
- Each group should form a "star" with their fingers touching
- Keeping the finger in constant contact with **underside** of the tube lower it to the ground
- Time 2 minutes

©2016 Waters Corporation
53

Behaviour, Interrelationship and Feedback

- The behaviour we observe in every system – designed or evolved is a consequence of the interrelationships between the components
- Remarkably the dynamic nature of these interrelationships comprise just two forms (for the whole universe!!):
 - Reinforcing feedback
 - Balancing feedback
- The behaviour of every system is due to these two forms of feedback and synchronisation of the feedback

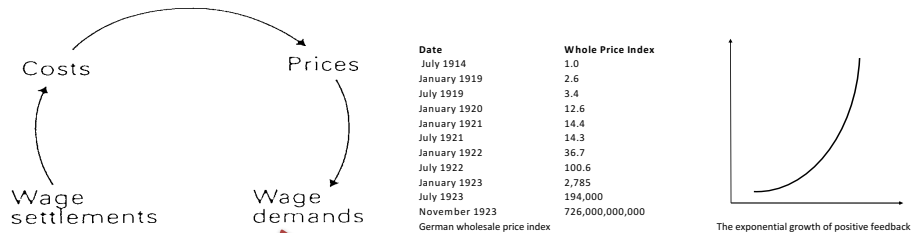

THE SCIENCE OF WHAT'S POSSIBLE.®

©2016 Waters Corporation
54

Properties and Characteristics of Systems Dynamic Behaviour and Feedback

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Reinforcing (positive) Feedback: successive changes add to previous changes in the same direction



This model of "inflation" is a pure positive feedback loop which if left unchecked will lead to exponential growth – what economists call hyper-inflation. In this case undesirable – but such feedback can be beneficial

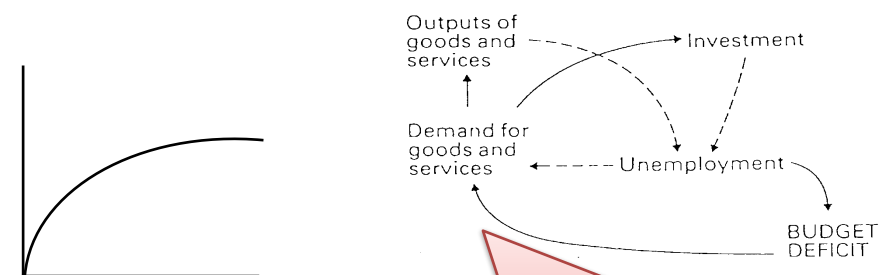
©2016 Waters Corporation

55

Negative and Positive Feedback

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Balancing (negative) Feedback: resists change in one direction by producing change in the opposite direction



Balancing feedbacks stabilise behaviour - they calm things down

In this economic example the full lines represent positive or reinforcing feedback, while the dotted lines represent balancing or negative feedback

©2016 Waters Corporation

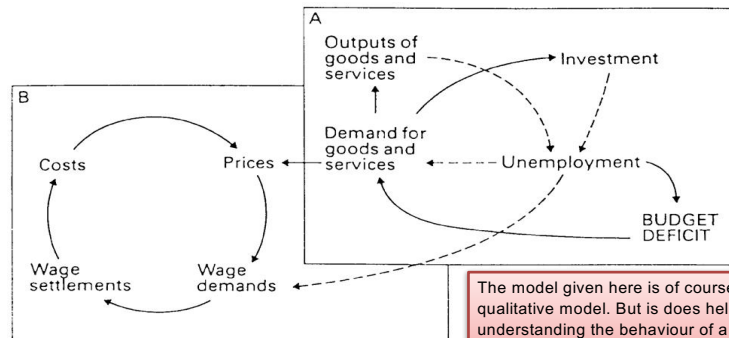
56

Behaviour is Affected by all the Elements in a System

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

This is the model developed by Keynes. He was able to "see" further connections and feedback loops in the bigger system. We often find that a lower level system is subject to feedback at a higher level.



The model given here is of course a qualitative model. But it does help in understanding the behaviour of a complex situation like a national economy.

©2016 Waters Corporation

57

Properties and Characteristics of Systems System Stability

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Stable

A system that is initially in equilibrium is disturbed, it is said to be stable if it returns to that equilibrium condition.



Imagine a ball, that was originally resting in the bottom of a bowl, moved up one side and released – it will eventually return to the bottom of the bowl

Marginally Stable

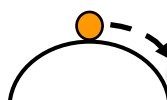
A system that is initially in equilibrium is disturbed, it is said to be marginally stable if it finds a new equilibrium condition.



If the ball is on a flat surface then if it is moved it will stay in its new position

Unstable

A system that is initially in equilibrium is disturbed, it is said to be unstable if it continues to move away from equilibrium condition.



If a ball placed on an upside down bowl is disturbed it will move away from its original position



Unstable systems can offer significant performance improvements but they can also destroy themselves

Date	Whole Price Index
July 1914	1.0
January 1919	2.6
July 1919	3.4
January 1920	12.6
January 1921	14.4
July 1921	14.3
January 1922	36.7
July 1922	100.6
January 1923	2,785
July 1923	194,000
November 1923	726,000,000,000

German wholesale price index

©2016 Waters Corporation

58

System Stability

- System Stability is determined by the interconnectedness and synchronization of the system elements
- System Stability can be changed by increasing or decreasing (removing or adding) “connections”
 - A stable system can be made less stable by reducing/removing balancing feedback or by increasing/adding positive feedback
 - An unstable system can be made more stable by reducing/removing positive feedback or increasing/adding balancing feedback
- System Stability can be changed by increasing or shortening the delays in a system – this is often counter intuitive.



Motorway Queues experience shock waves: fast-slow and stop-start behaviour because of an increase in the volume of vehicles causing a positive feedback. If we reduce the number of vehicles (reduce the positive feedback) the problem goes away!

The stop-start behaviour can also be influenced by introducing negative feedback into the system – variable speed limits. It should also be possible to change the behaviour by changing some of the delays in the system.

Systems Thinking in Engineering

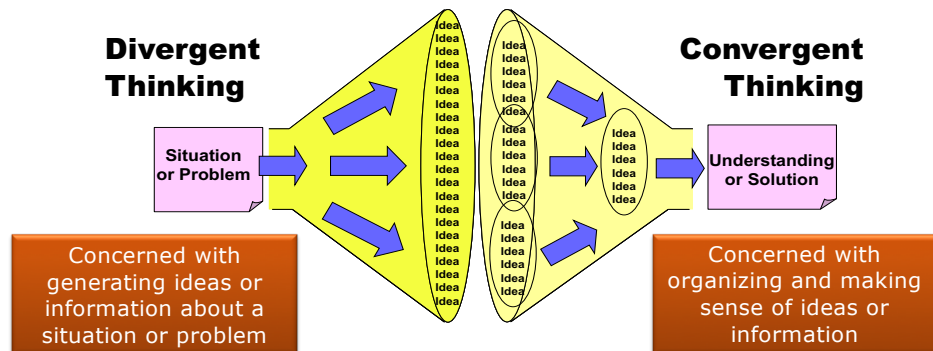
- Utilises Systems Thinking to
 - Understand context and purpose
 - To elicit and analyse requirements
 - To generate and explore solutions through abstraction
 - Manage complexity
 - Through determination of sub-systems
 - Through multiple perspectives
 - Manage emergent behaviour
 - By paying attention to the “connections” between sub-systems
 - By consideration of a lifecycle view
- Putting it in to practice requires a systematic and repeatable approach – we need a process.
 - The Systems Engineering Process is based on two fundamental basic processes:
 - Divergent and Convergent Thinking
 - Plan-Do-Check-Act

Divergent-Convergent Thinking a Systems Meta Process

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- When attempting to understand a situation or solve a problem the task is to find amongst all the possibilities the critical aspects. This task logically comprises two activities:
 - **Generating information**
 - **Organizing information**
- We often try to do this in one step by "jumping" to the "answer" – the problem is we stop thinking once we find a first adequate "answer"
- It is better to divide the task up into its component activities



©2016 Waters Corporation

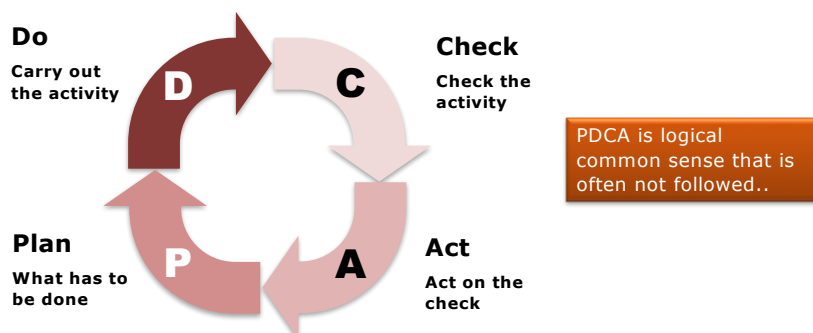
61

Plan Do Check Act a Systems Meta Process

Waters

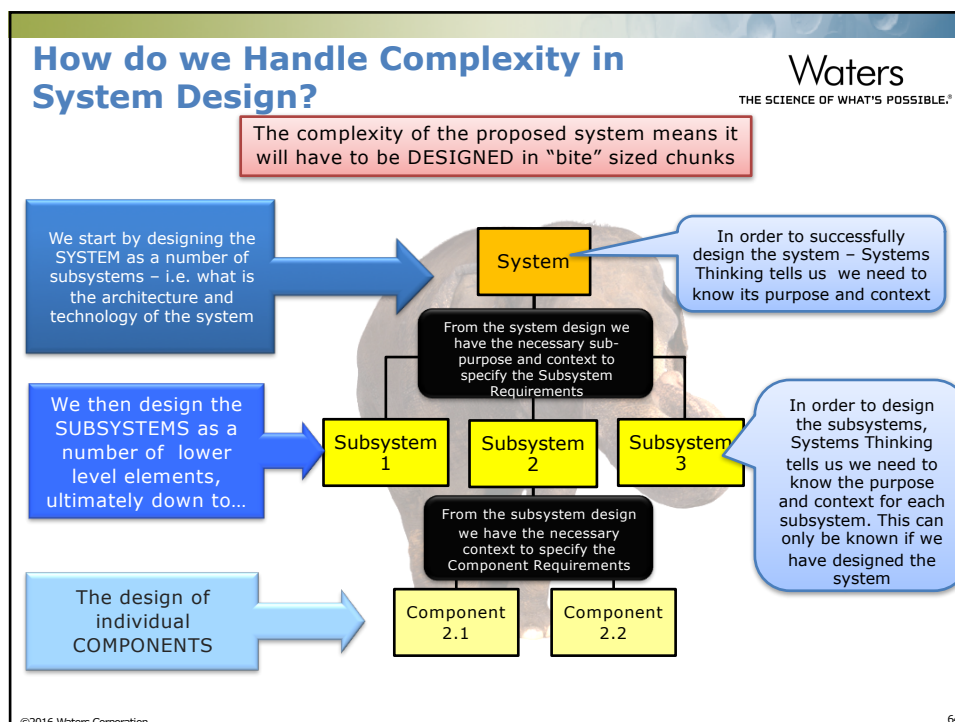
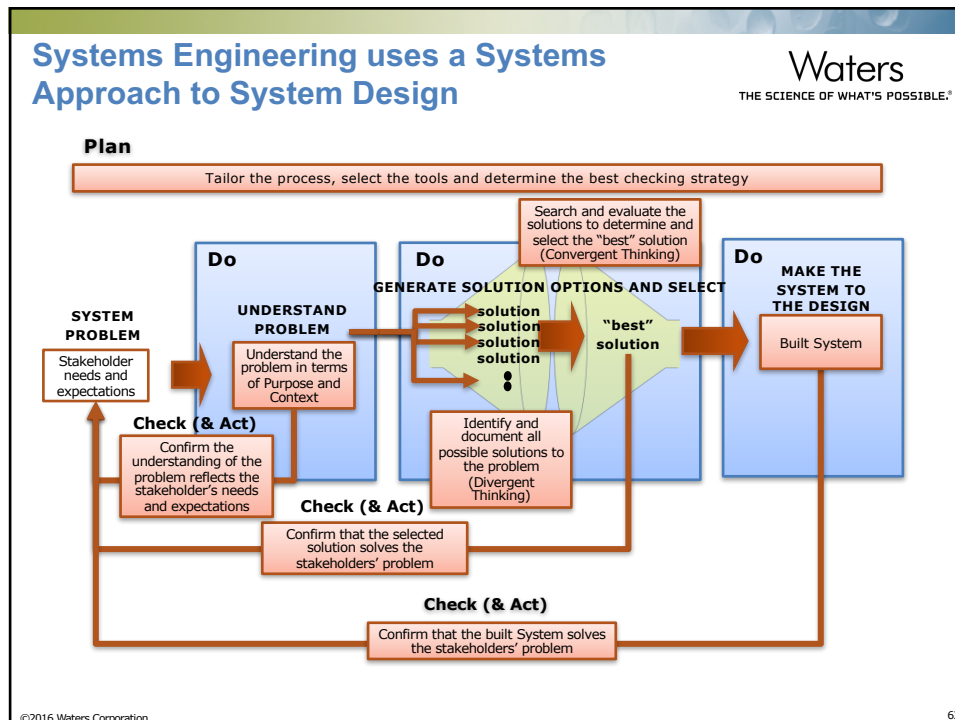
THE SCIENCE OF WHAT'S POSSIBLE.®

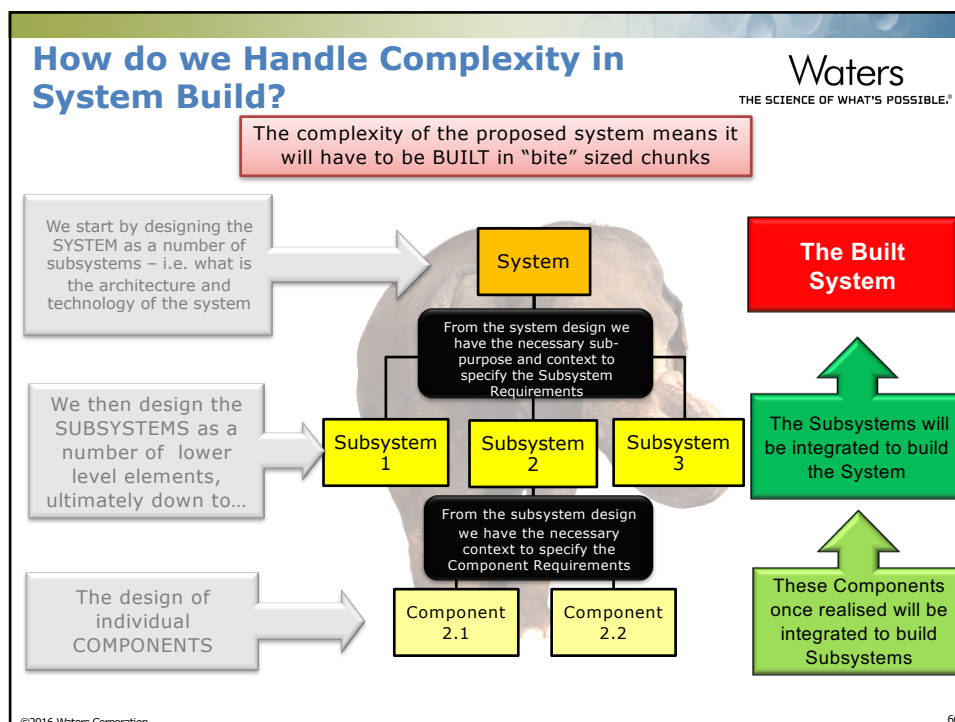
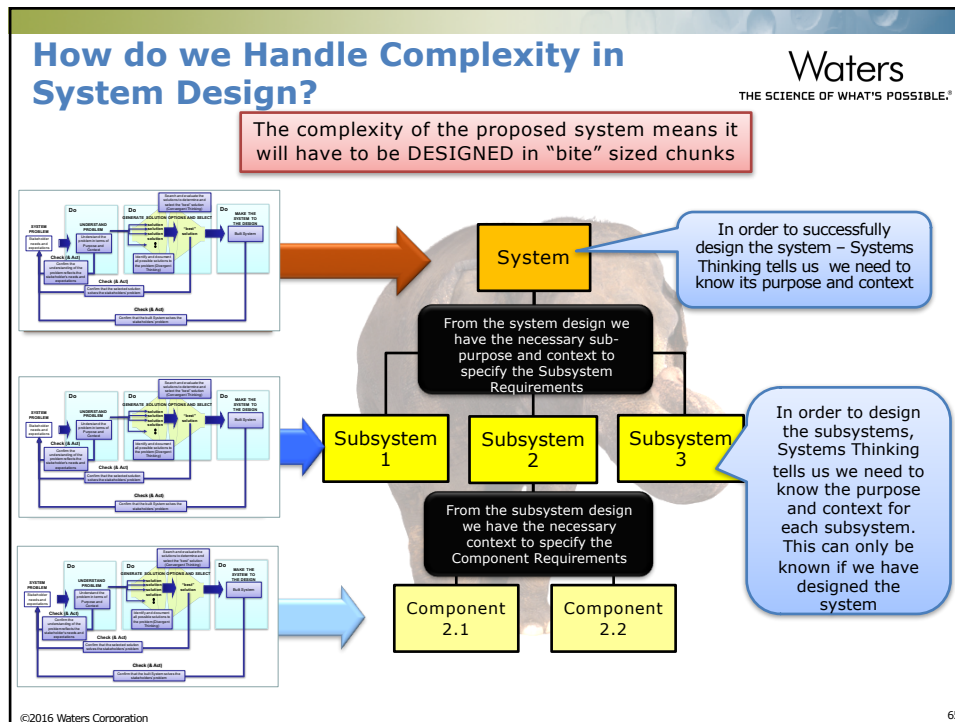
Often attributed to the American quality guru Edwards Deming, he said he got the idea from Walter Shewhart the inventor of Statistical Process Control. But even Shewhart's work can be traced to the Englishman Francis Bacon who first expounded the scientific method of Hypothesis, Experiment and Evaluation.

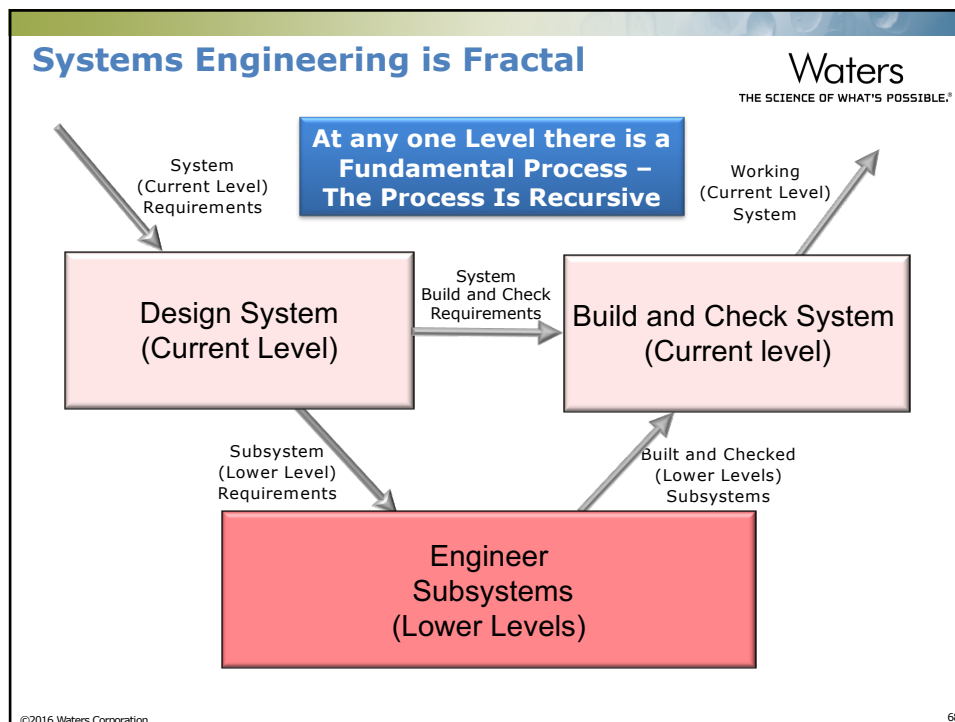
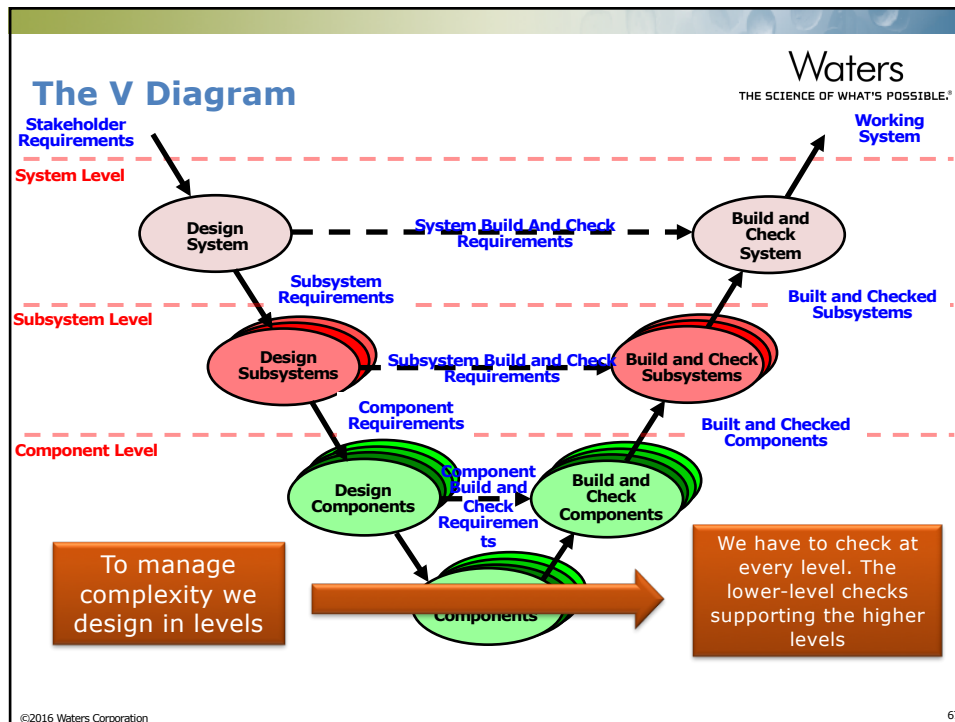


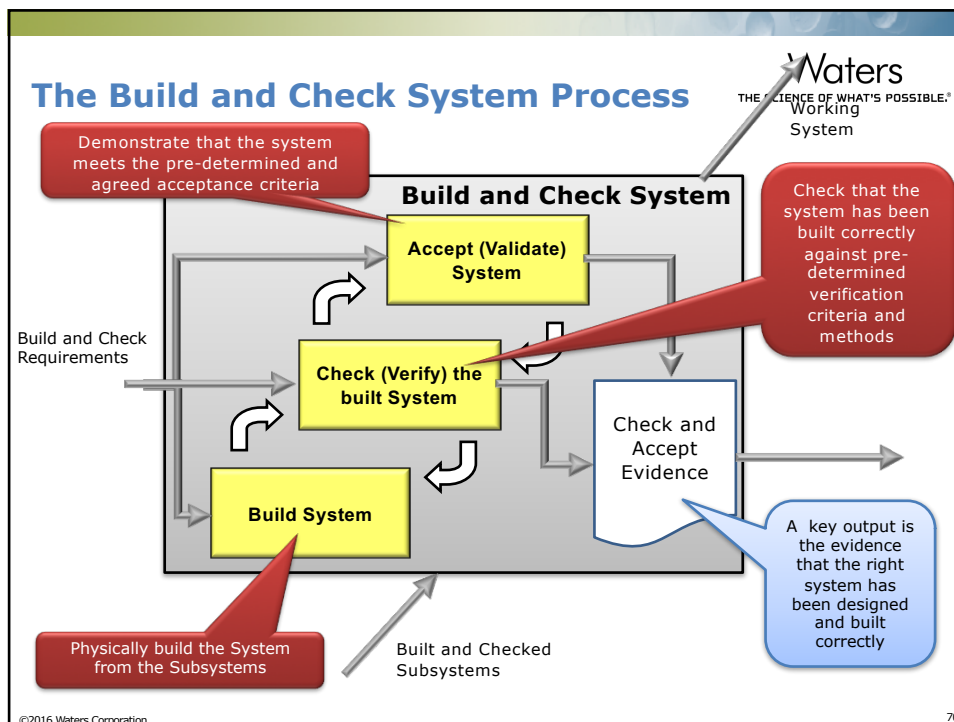
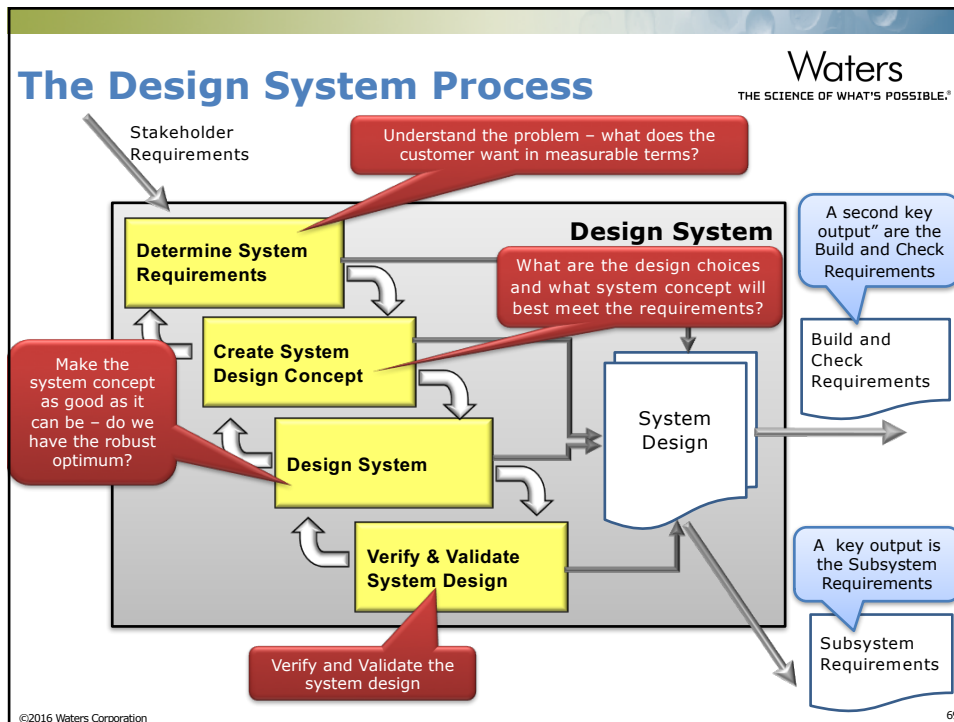
©2016 Waters Corporation

62

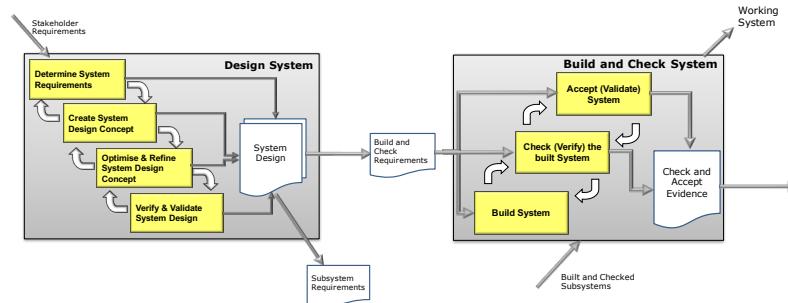








The Output of Systems Engineering

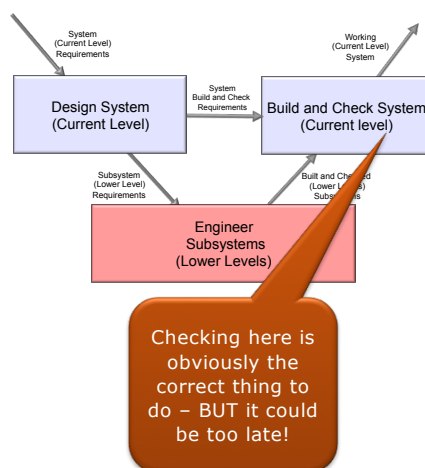


- It is important to note that the output of Systems Engineering is

INFORMATION

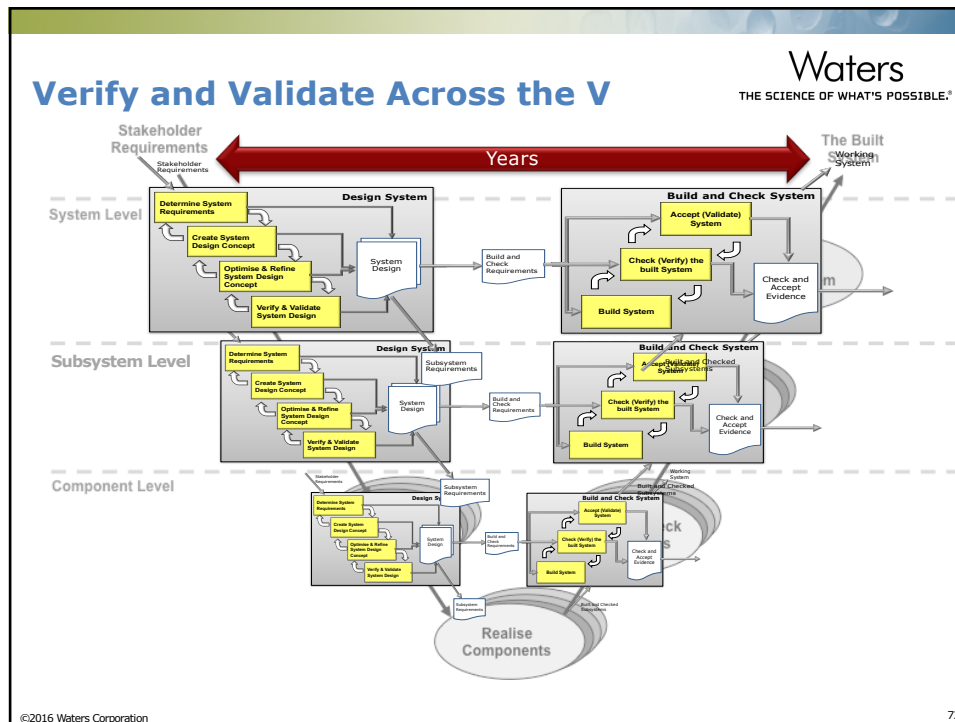
- Indeed you can argue that the purpose of Systems Engineering is to generate information about the system being created such that it can be successfully built

Checking



Checking as we Build

- As the components are put together to make subsystems, and subsystems put together to make the system, common sense dictates that we check what has been built at each stage, that is:
 - Components and subsystems meet their requirements
 - The system behaves as expected by the customer
- Both are necessary to have complete confidence in the final system



By Checking we Mean Verification and Validation (V&V)

- **Verification:** Confirmation by examination and provision of objective evidence that the specified requirements to which an end system is built, coded or assembled have been fulfilled

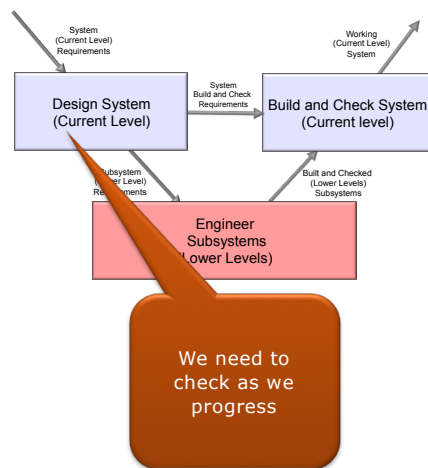
Did we do the thing right?

- **Validation:** Confirmation by examination and provision of objective evidence that the specified intended use of an end system or aggregation of end systems is accomplished in an intended usage environment

Did we do the right thing?

These are the generally accepted systems engineering definitions of V&V. The implications of these are quite profound since Validation becomes "something that can only be done when the thing is built! However, it is important to realise validation can apply throughout the system design activity.

Checking



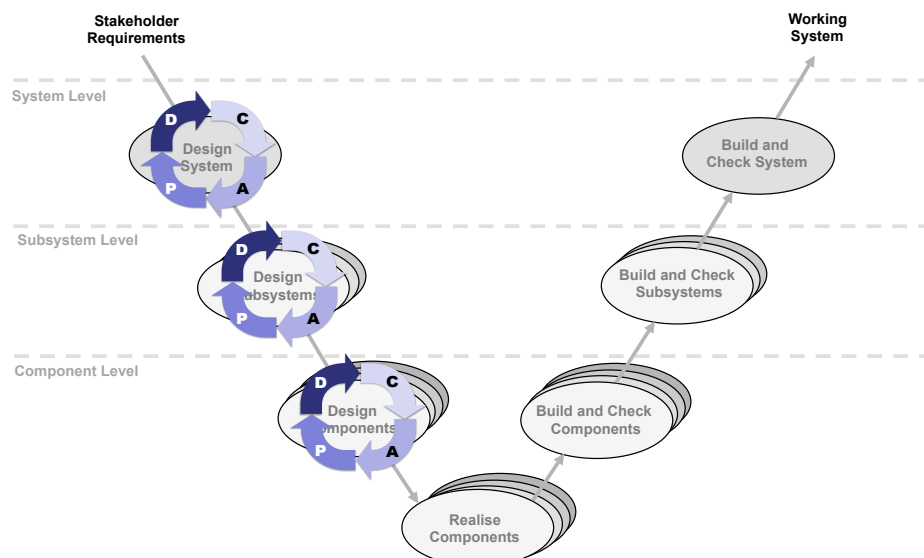
Checking as we Design

- As system elements are designed common sense dictates that we check as we progress that the
 - requirements are correct - are we designing the right system? (Validation)
 - design meets the requirements - has the system been designed right? (Verification)
- Both are necessary to have complete confidence in the final system

©2016 Waters Corporation

75

Verify and Validate Inside the V



©2016 Waters Corporation

76

Checking by Review

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ Why

- It is business critical that any project remains on track and at each stage fulfils the specified requirements.
- Reviews at specified stages of the lifecycle ensure that all the business and functional requirements fully comply with their agreed specifications.
- Reviews are essential to reduce risk - experts are used to independently assess the status of the project.

■ Generic Review Principles

- Reviews held in accordance with current engineering plan even if project is late
- Review uses existing project documents where possible
- Reviews are objective
- Reviews are evidence-based
- Reviews are conducted by acknowledged "expert" assessors
- Review Chair is independent of the project
- Results reported to the project director
- Reviews referred to by numbers, e.g. 2
- Reviews are key learning points - a basis for corporate learning

But WHEN do we Review?

©2016 Waters Corporation

77

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

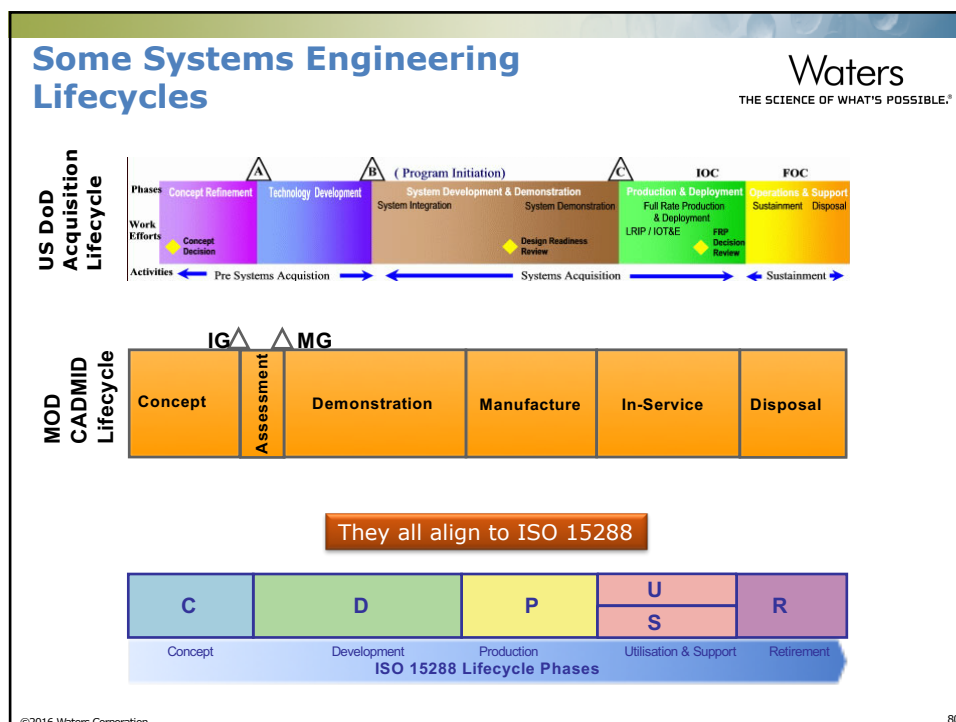
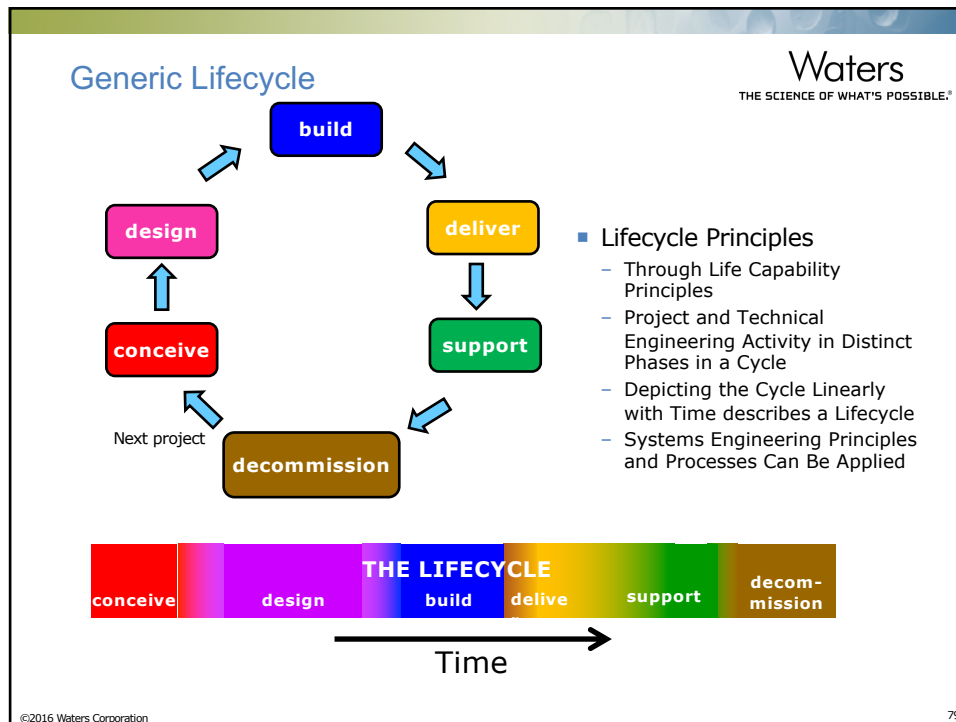
Systems Engineering operates in two dimensions

Complexity

Lifecycle

©2016 Waters Corporation

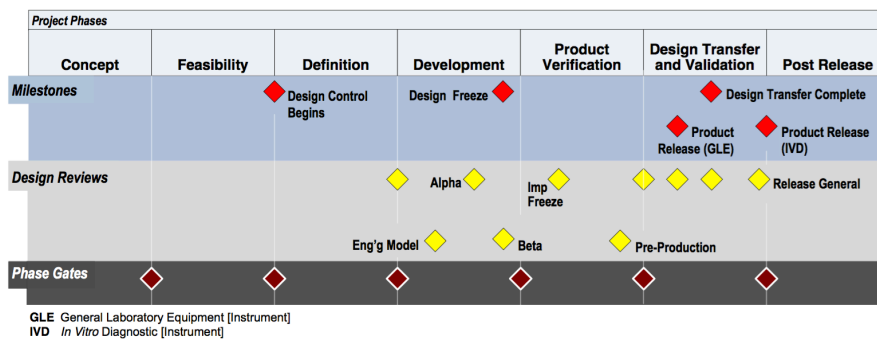
78



In Waters

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



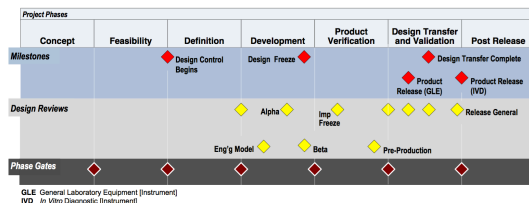
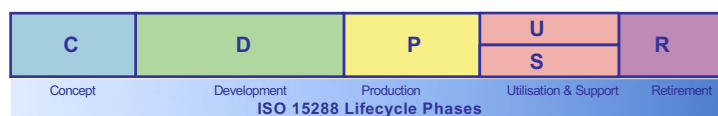
©2016 Waters Corporation

81

They all fit 15288

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

82

Lifecycle Activity – Lifecycles within Lifecycles

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

System-Level



Application Level



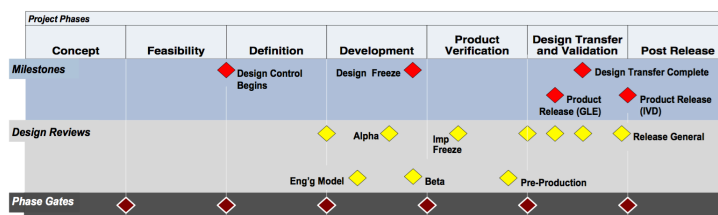
If delivering an integrated capability, for example, MS , Chemistry and Application Systems, these different elements of the system of systems may need to be at a higher level of maturity earlier than the main system

©2016 Waters Corporation

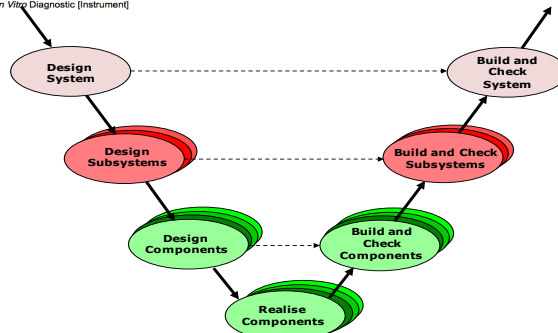
83

Lifecycle and the V-Diagram

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



GLE General Laboratory Equipment [Instrument]
IVD In Vitro Diagnostic [Instrument]

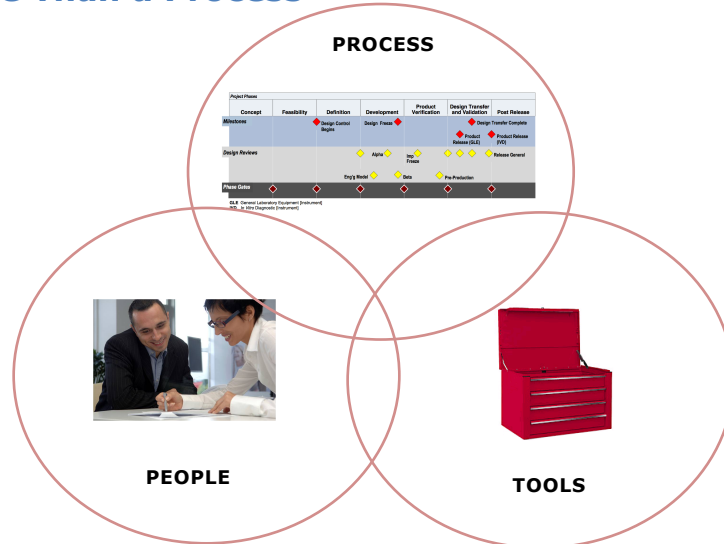


©2016 Waters Corporation

84

More Than a Process

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

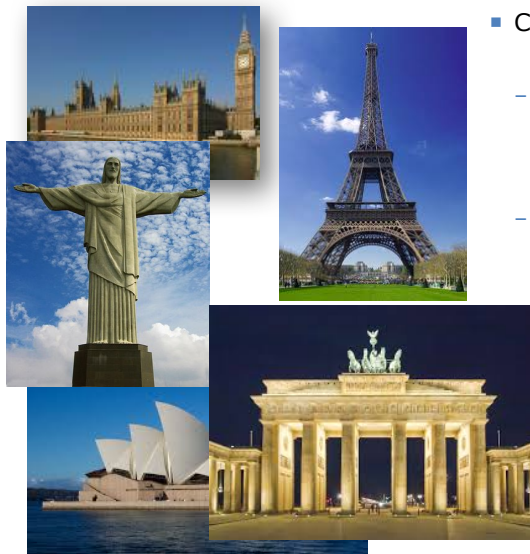


©2016 Waters Corporation

85

Exercise: People

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



■ Capital Cities

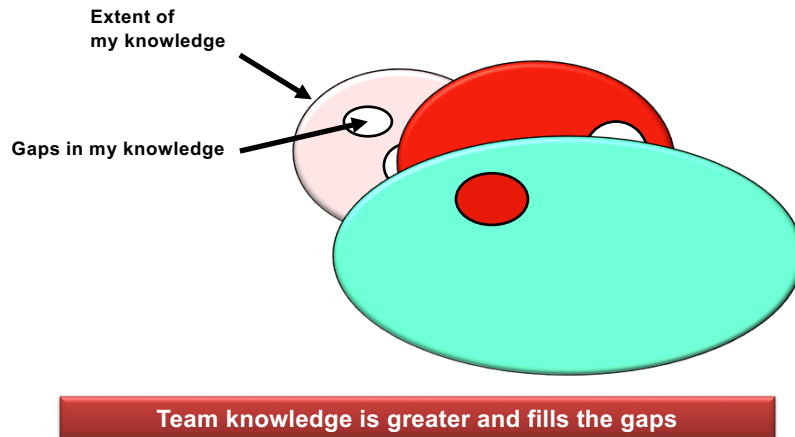
- Part 1: individually, list as many Capital Cities of the world as you can in two minutes
- Part 2: in small teams, use your individual lists to create a team list – time five minutes

©2016 Waters Corporation

86

Sharing Knowledge

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

87

Creating Knowledge

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Working as a team will cause us to share knowledge but
 - System Design often requires us to be creative
- We can be individually creative, BUT we are better when we work as a team
 - Voicing my ideas can help other team members to be more creative and vice versa



©2016 Waters Corporation

88

People and Tools in Systems Engineering

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

PEOPLE

- Systems Engineering is a team sport!
- People are essential - they generate the information required
- Teams are better than individuals because of
 - the sharing of knowledge and expertise
 - the information and knowledge generated by working together
- System design and development can involve many disciplines. Hence for complex multidisciplinary systems we have to have highly skilled multidisciplinary teams
- For a multidisciplinary team to deliver they must be able to communicate with each other to share information and have a common understanding to avoid having different perceptions

TOOLS

- Processes tell you what to do and when to do it (may also say who does it) – they should NOT specify how to do it – this is the job of tools
- BUT we need a set of tools that all teams members can use, with confidence, that allow us to handle the complexity of the system but at the same time are simple enough for the whole team to use – we need generalist systems tools
- Systems Engineering Tools are primarily communication devices that allow different disciplines to explore systems issues, often by building abstract models. They are scalable (any size of system), universal (any type of system) and above all simple (anybody can use them)**

©2016 Waters Corporation

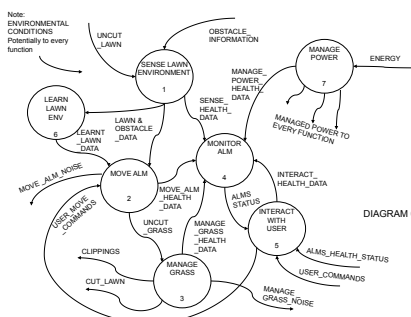
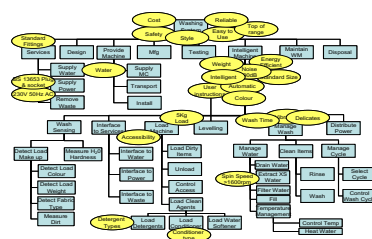
89

Systems Tools Build Models to Help the Team Reach a Common Understanding

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Systems Engineering makes great use of MODELS to look for structures, relationships and dependencies at an abstract level to help predict behaviour at the real level



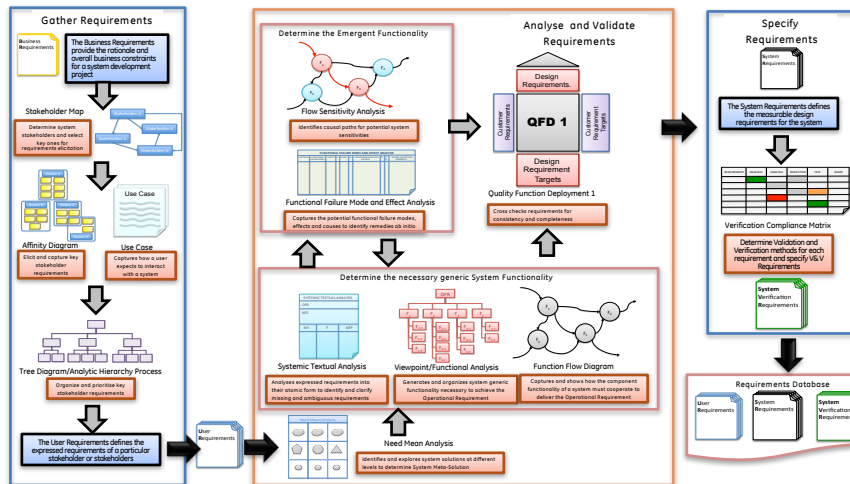
©2016 Waters Corporation

90

Determine System Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



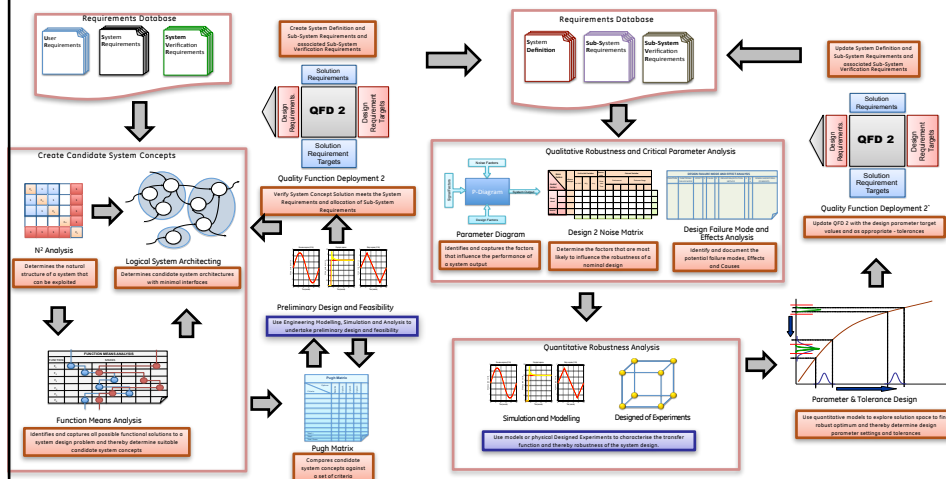
©2016 Waters Corporation

91

Create System Design Concept & Design System

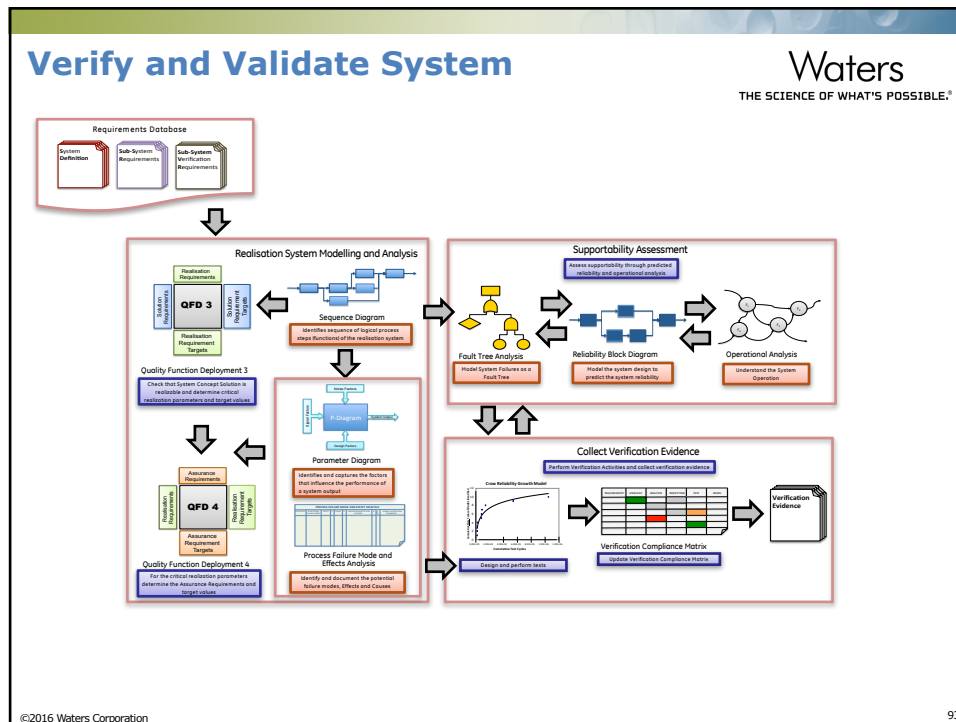
Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

92

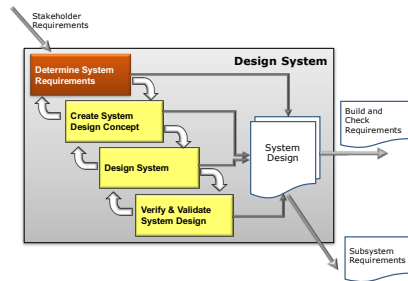


This Section:

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Explained why Systems Engineering is necessary
 - Undesirable emergent behaviour as a consequence of complexity
- Explained the relationship between Systems Thinking, Systems Approaches and Systems Engineering
- Introduced Systems Thinking as the foundation of Systems Engineering
- Provided an overview of Systems Thinking
 - The only way to manage complexity
- Showed how Systems Engineering utilises Systems Thinking to
 - Understand context and purpose
 - Manage complexity
 - Manage emergent behaviour
- Introduced the three ingredients of Systems Engineering
 - Process (Lifecycle and Complexity)
 - People (multidisciplinary teams)
 - Tools (easy to use, scalable and universal)

©2016 Waters Corporation



Systems Engineering Principles Requirements Engineering

This Session:

- Describes and explains the purpose and activities of Requirements Engineering

What is a Requirement?

In simple terms a requirement is a need or want expressed by a "customer"

Why we Need Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Requirements:
 - Express a need or want
 - Are the basis for design
 - Are the basis for confirming something has been done correctly

Define what we are commercially obligated to deliver against

If we do not have a “good” set of requirements we do NOT understand our commercial obligation and are therefore at RISK

©2016 Waters Corporation

3

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



- In small groups
- Develop a set of customer/user requirements for a Toaster!
 - Flip chart the requirements
 - Time 15 minutes

©2016 Waters Corporation

4

Satisfied?



©2016 Waters Corporation

5

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- If these requirements were delivered would you be happy?
 - When developing the requirements did you think of your toaster at home?
 - If you were the retailer would your requirements be different?
 - Are all the requirements for the Toaster?
 - Are your requirements complete?
 - Are your requirements realistic?
 - Are your requirements unambiguous?
 - Will your requirements change?

Requirements: Market Aspects

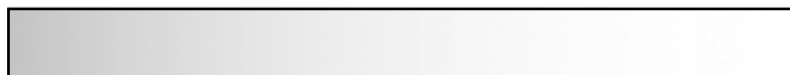
Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The need for a system will arise, in general, from one of two principal sources:
 - Broad market: where the requirement is generated by the producer
 - Narrow market: where the requirement is generated by the customer

Broad

Narrow



In practice most organizations sit somewhere between the two extremes

©2016 Waters Corporation

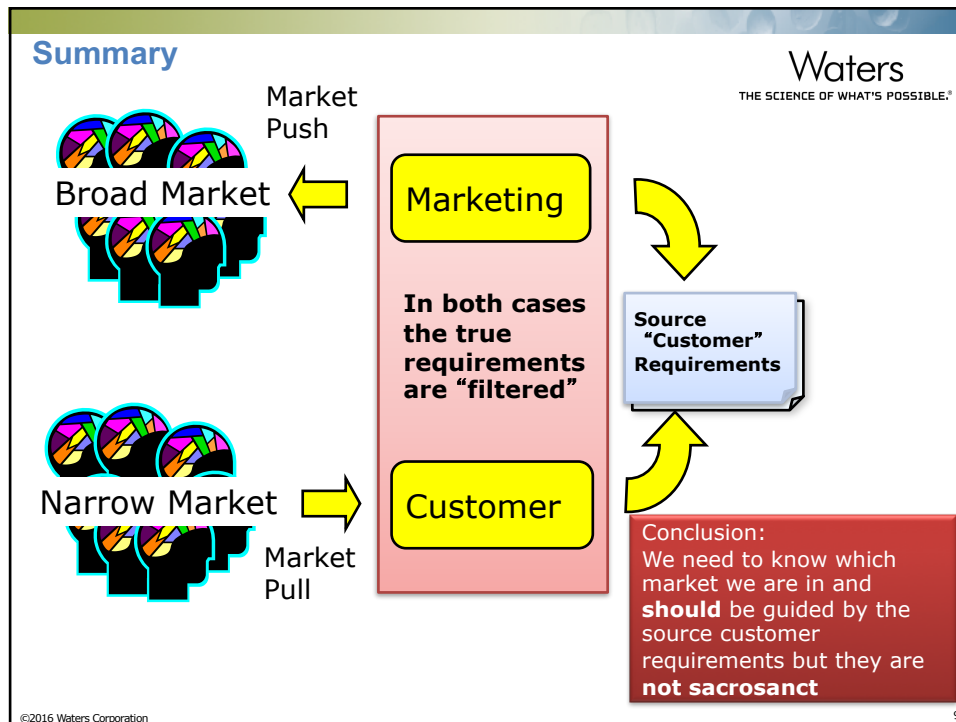
6

Broad Market Characteristics

- A Broad Market is where the intention is to sell the same system to as many customers as possible e.g. a TV or motor car
- It is impossible to “ask” every customer and so requirements are elicited from a representative sample of customers
- The organization has to go to the customer to elicit their requirements
- Usually Broad Market organizations establish a marketing function which should:
 - *determine the need*: is there a market for a particular system, are there any gaps in the market, etc.
 - *define market sector*: what group of people will buy this system, how big is this group, what price are they willing to pay, etc.
 - *determine customer requirements*: What does the customer want
 - *communicate customer requirements clearly and accurately*: to the design organization

Narrow Market Characteristics

- A Narrow Market contains one or very few customers who have a specific need - e.g. Defence, Infrastructure, Health Care, Power/Energy Provision are Narrow Markets
- The systems are typically very complex and expensive
- The customer comes to the organization with a set of requirements they have generated
- Narrow market requirements are a formal statement of customer needs and expectations in terms of:
 - purpose of the system
 - performance characteristics
 - including environmental and usage conditions, reliability, safety, maintenance requirements, cost
 - applicable standards and statutory regulations
 - quality levels
 - supportability
 - packaging etc.



Requirements:

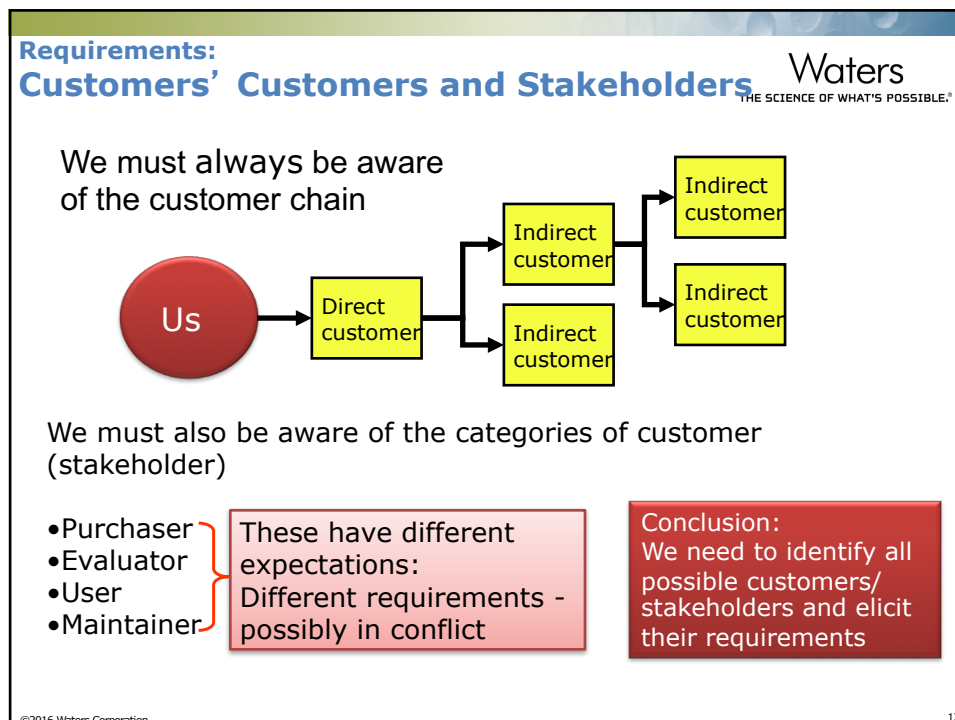
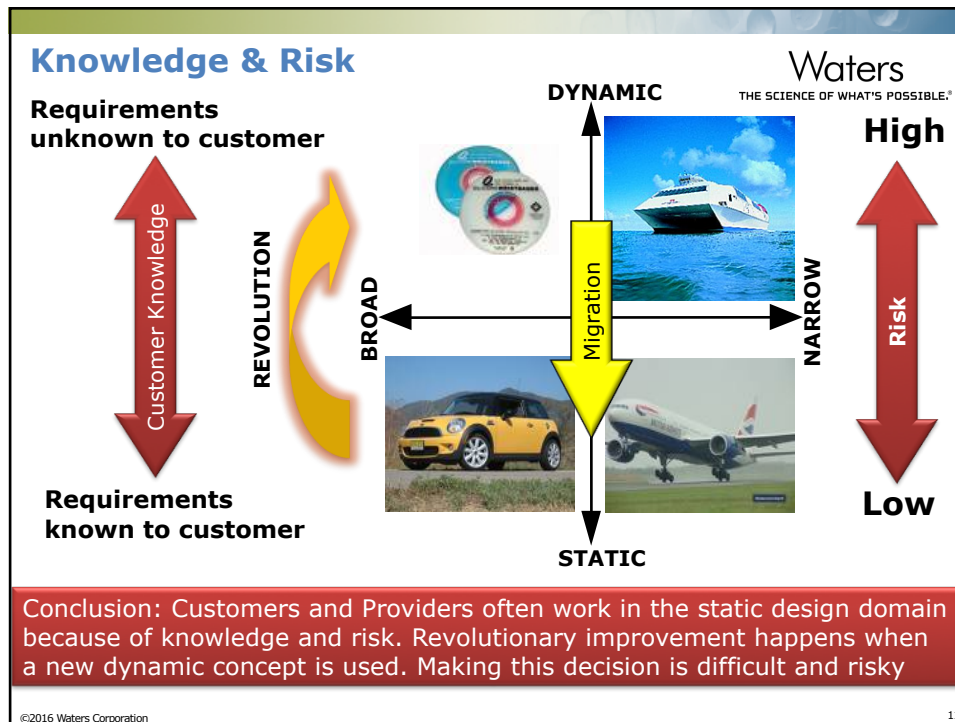
Static and Dynamic System Concepts

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Another dimension to customer requirements comes from the idea of static and dynamic design concepts
- Most systems are conceptually static as an overall design concept
 - For example: motor cars, washing machines, civil airliner conceptually have not changed for several decades
- Few systems are conceptually dynamic
- Static concept designs are often dynamic at the subsystem level
 - For example: Compare today's motor car with those of thirty years ago – they now have Engine Management Systems, ABS, Climate Control Systems, Traction Control Systems etc.

©2016 Waters Corporation

10



Requirements:

The PRIME System's Friends

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Every PRIME system is associated with several other systems. When we ask "stakeholders" what are their requirements for the PRIME system they will also tell us requirements for the other systems!

The project or programme is also a system

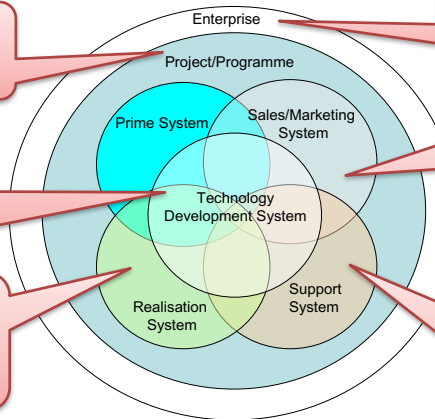
The enterprise itself is also a system

The System responsible for developing new technologies for ALL the other systems

The System responsible for marketing and selling the prime system

The Systems (design, development, supply chain) that realise the Prime System

The System responsible for the support of the prime system in operation and ultimately disposal



Conclusion: We cannot stop customers and stakeholders giving requirements for the associated systems. This is not a bad thing (in fact it is good that they do this) but it does mean we have to sort the raw requirements and align them to their respective systems

13

Requirements:

Customers can't get no Satisfaction

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

CUSTOMER EXCITEMENT	High	Expressed Those requirements that are sufficiently important to be verbalised by the customer. In general, the more the better!	Delight Those requirements that cannot be articulated because the customer does not know they want them, but if provided will surprise and delight. These are NOT over design!
	Low	Basic Those requirements that are so obvious to the type of system that the customer will not mention them. However, if not provided the customer will be dissatisfied!	Assumed Those requirements that while not basic yet are starting to be taken for granted. If not provided the customer will be disappointed!
		Low	High
		CUSTOMER SATISFACTION	

Conclusion:

- Customers/Stakeholders require more than satisfaction - we need to think about excitement
- Customers/Stakeholders only tell us their expressed requirements; typically only 20% - 30% of the full set of requirements (spoken vs. unspoken requirements). We are going to have to determine the missing ones!

©2016 Waters Corporation

14

Requirements: Ambiguity

I would like a bag of chips please.



Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Conclusion:

- We need to identify where there is ambiguity – and resolve it

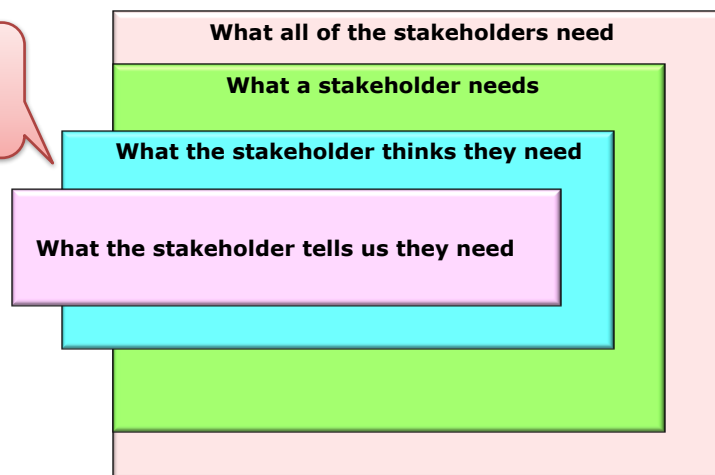
©2016 Waters Corporation

15

Requirements: Just a Little bit More than Last Time

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Requirements that while expressed may not actually be needed



Conclusion:

- We should never assume, because a customer/stakeholder has expressed a requirement, that it is correct or actually needed - they often express goals or "wish lists"

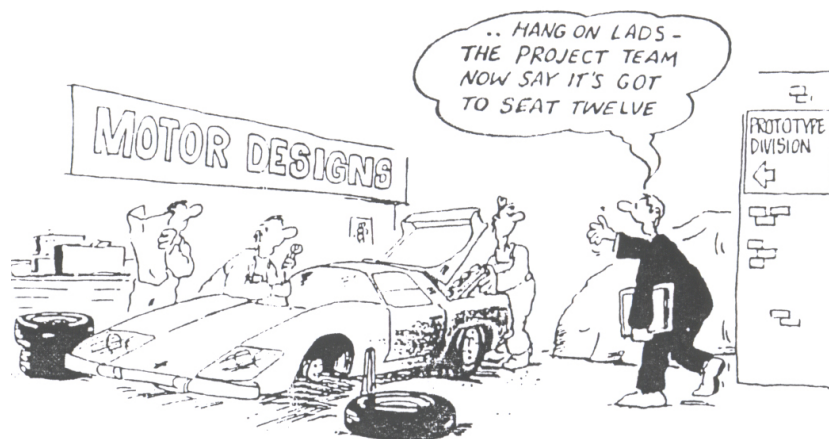
©2016 Waters Corporation

16

Requirements: Customers Change their Minds!

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Conclusion:
We have to manage and cope with requirements that will change

©2016 Waters Corporation

17

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Summary: A Bit of a Mess

The Facts

- Customers are biased by what they know
- Customers are not the only stakeholder and they all have different requirements
- Customer requirements are never complete
- Customers will ask for things they don't actually need
- Customers give ambiguous and vague requirements
- Customers will change their mind

The Questions

- How do we remove bias from requirements?
- Who are the stakeholders and what are their requirements?
- How do we find out what requirements are missing?
- How do we find out what requirements are not needed?
- How do we interpret vague and ambiguous requirements?
- How do we cope with changing requirements?

©2016 Waters Corporation

18

What do we need From Requirements?

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Having just identified some of the issues with requirements – the reverse suggests that the key characteristics of a “good” requirement are:
 - Clear
 - Concise
 - Consistent
 - Complete
 - Correct
 - Confirmable
- There are other aspects to good requirements that relate to using them to design the system – Systems Thinking says we need to know
 - The fundamental purpose of the system
 - The major constraints that affect the system
 - What the system has to do in order to meet the fundamental purpose
 - How well the system has to perform

©2016 Waters Corporation

19

Getting to Grips with Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Achieving
 - **Completeness**: requires several viewpoints to be taken
 - **Consistency, Conciseness and Clarity**: requires the identification and removal of ambiguity this in turn requires communication to share information and arrive at a common understanding - to avoid having different perceptions
 - **Correctness and Confirmation**: requires the verification and validation of requirements

This demands discipline and organization

A consistent approach to engineering the requirements – We need a process

A consistent way of organizing and interpreting requirements – a framework

©2016 Waters Corporation

20

Remember

Requirements

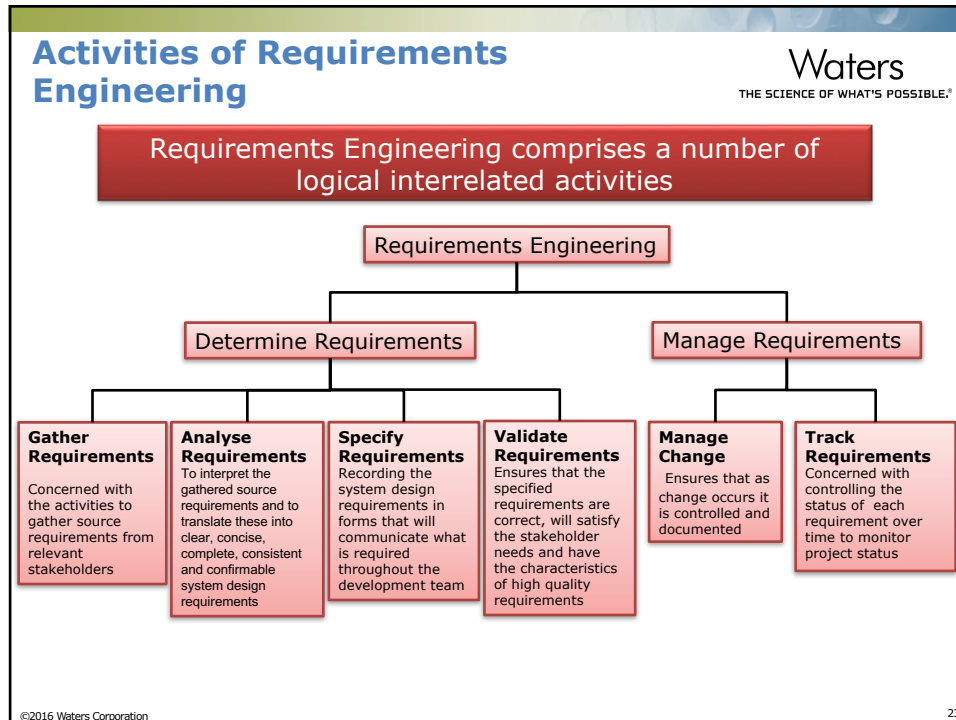
Define what we are commercially obligated to deliver against

If we do not have a "good" set of requirements we do NOT understand our commercial obligation and therefore at RISK

The Answer: Requirements Engineering

What is Requirements Engineering

- Requirements Engineering is a relatively new discipline that has emerged from Systems Engineering and Software Engineering in recognition of the need to address the issues raised
- Requirements Engineering is concerned with:
 - Gathering and analysing (understanding) requirements
 - Organizing, specifying and validating requirements
 - Specifying evidence-based means of confirming requirements have been met
 - Managing and tracking requirements throughout the life of a particular system
- It is an often overlooked activity BUT it is the key to both design success, and RISK reduction!



Requirements Engineering Summary

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

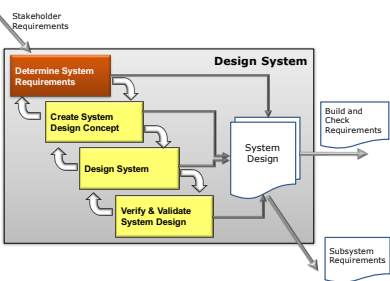
- This section has
 - Explained the issues surrounding customer requirements
 - Presented the purpose of Requirements Engineering and its key activities

©2016 Waters Corporation 24





THE SCIENCE OF WHAT'S POSSIBLE.®



A Systems Approach to

Determine System Requirements

This Session

- Defines a Systems Approach to Determining Systems Requirements
- Introduces the Holistic Requirements Model – a systems approach to understanding requirements
- Demonstrates the importance of system functionality as an invariant and logically defensible method of handling complexity

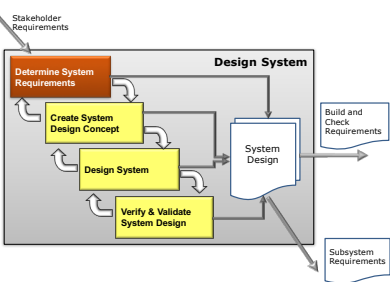
©2016 Waters Corporation

1

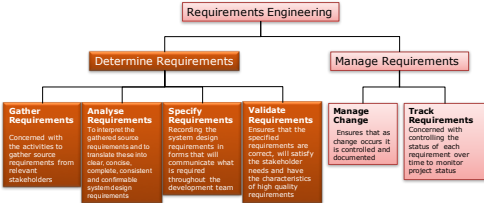
Where are we? And what are we trying to do?



THE SCIENCE OF WHAT'S POSSIBLE.®



- Purpose :
 - To elicit, capture, identify and interpret customer, and other stakeholder, requirements accurately, and to translate these into clear, concise, complete, consistent and confirmable technical system design requirements
- Outcomes :
 - A set of source requirements that define the customer and stakeholder needs and expectations (especially those aspects that are Critical or Key Requirements)
 - A complete and consistent set of measurable system design requirements that have been validated and verified as satisfying the customer and stakeholder requirements



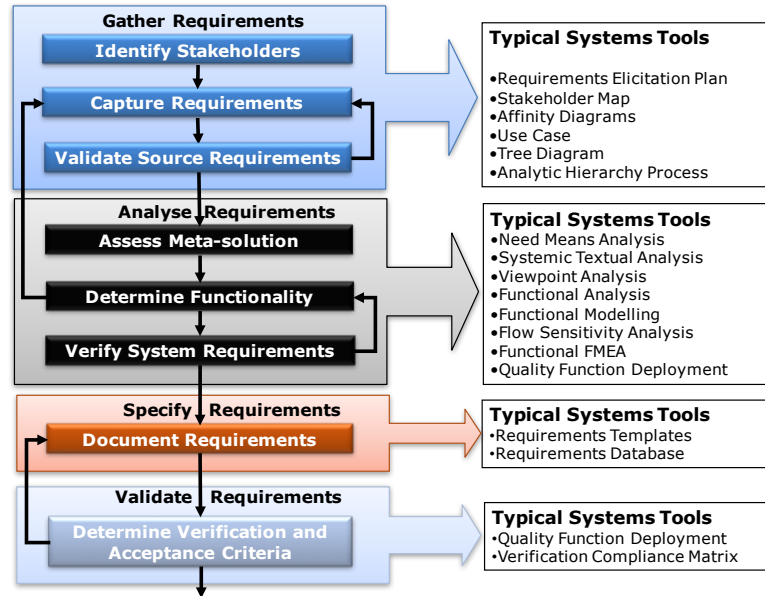
©2016 Waters Corporation

2

Determine Requirements: Tasks and Tools

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

3

Understanding Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Requirements originate from various sources
- Source requirements are:
 - Incomplete
 - Inconsistent
 - Incorrect
 - Ambiguous
 - Rambling
 - Unconfirmable

©2016 Waters Corporation

4

Getting to Grips with Requirements

- Achieving
 - **Completeness:** requires several viewpoints to be taken
 - **Consistency, Conciseness and Clarity:** requires the identification and removal of ambiguity this in turn requires communication to share information and arrive at a common understanding - to avoid having different perceptions
 - **Correctness and Confirmation:** requires the verification and validation of requirements

This demands discipline and organization

A consistent approach to engineering the requirements – a process (high level shown earlier)

A consistent way of organizing and interpreting requirements – a requirement model or architecture

Organizing Requirements

There are many ways of organizing requirements, but only one systems view. This systems view of requirements provides a complete and consistent framework for organizing and understanding the deficiencies of a set of requirements.

It is called the **Holistic Requirements Model** and says that a complete and consistent set of system requirements will comprise:

- **Operational Requirements**
- **Functional Requirements**
- **Non-functional Requirements**
 - **Non-functional Performance Requirements**
 - **Non-functional System Requirements**
 - **Non-functional Implementation Requirements**

Operational Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- An **Operational Requirement** defines the major purpose of a system (i.e. what it fundamentally does - its capability) together with the key overarching constraints (that define the context)

System	Operational Requirement
Dishwasher	To clean domestic eating and cooking utensils and crockery efficiently and without damage
Civil Aircraft	To transfer passengers and their baggage from one point to another efficiently and safely
Toaster	
Overhaul Aircraft	To achieve high aircraft availability in a cost-effective and safe fashion

All Systems have purpose and context - the Operational Requirement captures these

All systems have a purpose and a context. The job of the Operational Requirement is to define these in simple terms - to define the "mission" of the system

p.s. customers often don't tell us their Operational Requirement

©2016 Waters Corporation

7

1961: Kennedy and the Ultimate Operational Requirement

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- President John F Kennedy in a speech to a joint session of Congress broadcast on TV and radio around the United States
- "I believe that this nation should commit itself to achieving the goal,
 - before this decade is out, of landing a man on the Moon and returning him safely to Earth"



One Small Step



Writing an Operational Requirement is not easy. It can take several hours to arrive at an agreed set of 20 or fewer words!

Do you know the Operational Requirement of the system you are currently working on? Is it written down in 20 or fewer words that is the common understanding of the whole team?

©2016 Waters Corporation

8

Functional Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- **Functional Requirements** specify the **functions** of the system - what the system has to DO in order to satisfy its Operational Requirement
- For example some of the functions of a civil aircraft are:

- Navigate aircraft
- Manoeuvre aircraft
- Store passengers
- Control passenger storage environment
- Communicate with other aircraft and ATC
- Etc.

We try to write Functional Requirements in way that does not specify, imply or assume a solution (like seat passengers). However, every system has to integrate with legacy systems and therefore this is not always possible

- Note that a Functional Requirement contains a function expressed as a phrase comprising a verb (action) followed by a noun (object)
 - A functional requirement should define something the system has to DO
 - A phrase can have a verb but not be a function! For example "easy to use" has a verb but this is not a function - we don't say the system "does easy to use" it has "to be easy to use" - a property or attribute
 - Best practice is to define functions as verb noun combinations that describe an action on an object
 - Beware of pseudo-functions, verb noun combinations that do not define an action on an object. For example "provide style" is a pseudo-function because style is a property and not a thing - it is an abstract noun rather than a common noun. Functions that end with an abstract noun are pseudo-functions

©2016 Waters Corporation

9

Non-functional Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Non-functional Requirements are constraints on the system

- Non-functional requirements do NOT specify what the system has to do (but they might imply it!)
- Non-functional requirements specify:
 - How well it does something
 - How it does something
- They fall into three categories:
 - Non-functional Performance
 - Non-functional System
 - Non-functional Implementation

©2016 Waters Corporation

10

Non-functional Performance Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- **Non-functional Performance Requirements** are linked to functions and define how well a particular function has to perform – they are the constraints on a particular function. They comprise an attribute or property with an associated target value

System	Function	Non-functional Performance Requirement
Aircraft	Navigate aircraft	- Update Rate < 10s - Accuracy < ± 1km in 4000km - Precision < 1km standard deviation - Reliability > 1000 MTBF - Weight < 56kg - Etc.

There is no verb! Hence these are not functional requirements (they do, however, IMPLY a Functional Requirement)

©2016 Waters Corporation

11

Non-functional System Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- **Non-functional System Requirements** define constraints that affect the whole system (at least more than one function) and include:
 - Physical Attributes
 - System Size
 - System Weight
 - etc.
 - The ilities
 - System Reliability
 - System Maintainability
 - System Interoperability
 - etc.
 - System Performance
 - System Cost
 - System Speed
 - System Manoeuvrability
 - etc.

Performance Requirements come in two forms: those that affect **the system** and those that affect **a specific function**

WE DELIBERATELY SEPARATE THEM

Note: the Non-functional System Requirements will get “flowed down” in to lower level requirements

©2016 Waters Corporation

12

Non-functional Implementation Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- **Non-functional Implementation Requirements** define how a system is to be built in terms of a specific technology. They are solution-based requirements from the customer or legislation

System	Function	Non-functional Implementation Requirement
Aircraft	Communicate ..	Phillips A/C 1267 VHF Radio
Dishwasher	Remove Waste	Electric Pump
Toaster	Receive Power	UK Domestic 13A Plug to BS1363
Overhaul Aircraft	Maintain Component History	Microsoft Access database on a PC

Non-functional Implementation Requirements come in two types:

Justifiable: where the requirement is reasonable - typically because it relates to a design decision made at the higher system level

Lazy: where the stakeholder has resorted to a solution-based requirement because they do not know how to express it otherwise

We need to identify the Non-functional Implementation Requirements and determine whether they are lazy or justifiable

©2016 Waters Corporation

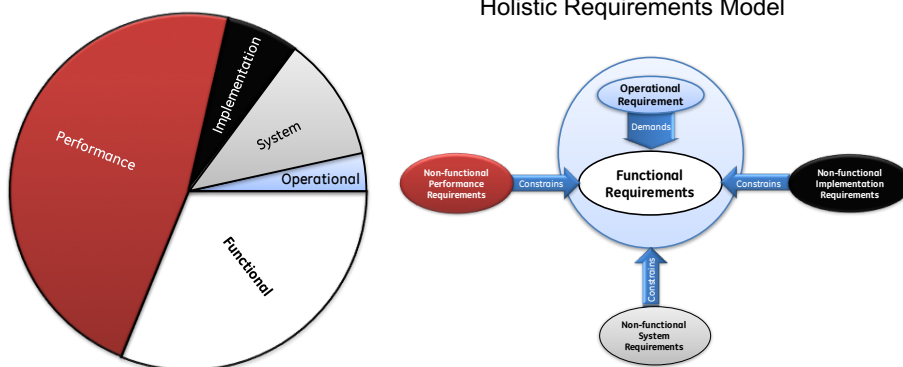
13

Summary: The Holistic Cutting of the Cake!

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Holistic Requirements Model



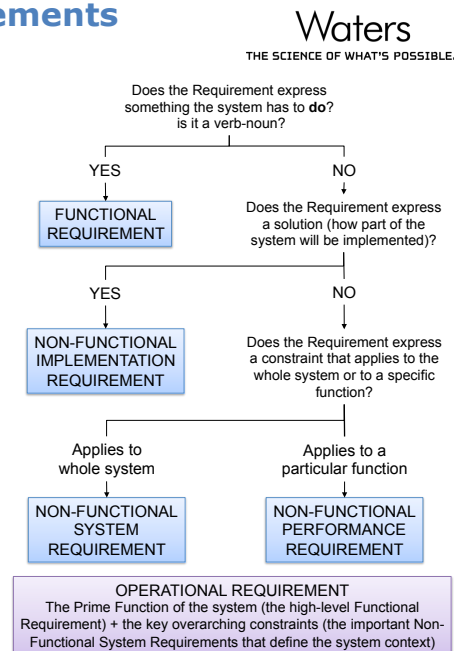
The requirement categories of the Holistic Requirements Model are not arbitrary – together they form a **complete and consistent** set of requirements that has a structure which we can exploit to identify missing requirements and provide a powerful universal framework for system design

©2016 Waters Corporation

14

Using the Holistic Requirements Model

- The flowchart opposite can help in consistently and correctly categorizing a set of source requirements
- Its use does require practise BUT it can provide a fast method to identify the deficiencies in a set of source requirements
- Even experienced users will “get it wrong” on the first pass – but know subsequent iteration will address any miscategorization



©2016 Waters Corporation

15

Exercise – Requirements Categories

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- In teams go through your toaster requirements and indicate which are:
 - Operational requirements OP
 - Functional requirements F
 - Non-functional requirements
 - System NFS
 - Performance NFP
 - Implementation NFI
- Time 20 minutes

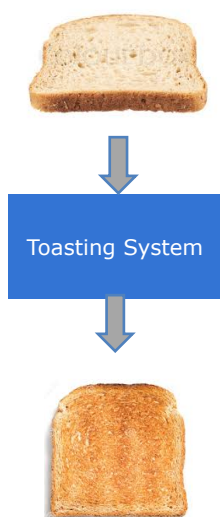
©2016 Waters Corporation

16

Nature of Customer Requirements

- Humans tend not to specify functionality!
- It is assumed or implied through constraints
- What is often missing from a set of requirements is the functionality

- It's not surprising, actually! Customers are often not interested in how something is done but how well it's done
- We must, however, determine has to be done - the functionality - because it is this that the system has to deliver the customer's expressed requirements



- What is the generic functionality necessary to
 - “to toast bread product”
- Time 10 minutes

- **To toast bread product**

Generic System Functionality



Generic System Function	Implementation
Load bread	Stick bread slice on end of toasting fork
Generate heat	Open coal fire
Apply heat	Use toasting fork to bring bread close to open fire
Monitor toasting	Look at the bread from time to time
Control toasting	Move bread to get even toasting
Remove heat	Use toasting fork to bring toast away from coal fire
Unload toast	Take toast off the toasting fork

©2016 Waters Corporation

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



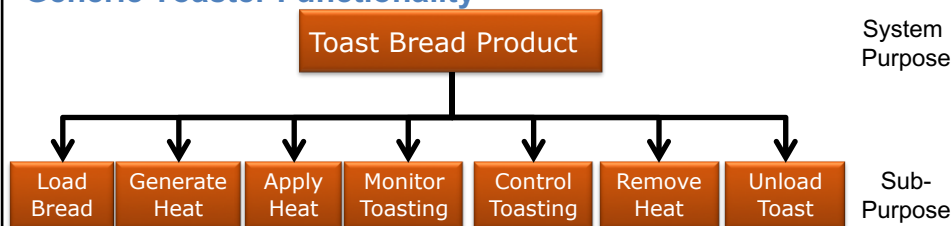
Generic System Function	Implementation
Load bread	Put bread into slots
Generate heat	Electric elements
Apply heat	Lower bread next to elements using lever mechanism
Monitor toasting	Look at the bread from time to time
Control toasting	Timer
Remove heat	Lift bread from elements using lever mechanism
Unload toast	Take toast from slots

19

Generic Toaster Functionality

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Did your requirements for the toaster specify any of these?
How do your requirements relate to the system functions?

Every engineered system has an underlying set of functionality that enables it to achieve its purpose. Similar systems will have similar functionality; indeed, systems that have the same purpose will have a set of functionality that is identical. This functionality is the Generic System Functionality (GSF) and is time invariant (provided the purpose does not change the GSF does not change)

Whilst the GSF is a consequence of the system purpose and is fixed, it can be implemented in many ways which change as technologies evolve and emerge. The generic system functionality is not the complete set of system functionality – a system can have functionality that is not generic:

©2016 Waters Corporation

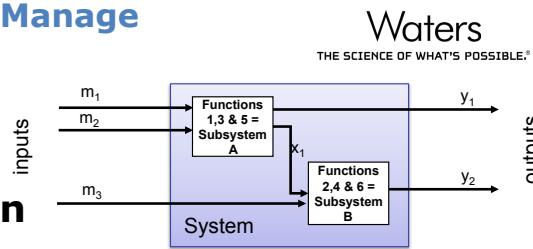
20

Using Functionality to Manage Complexity

- In Systems Engineering

Purpose $\equiv$ Function

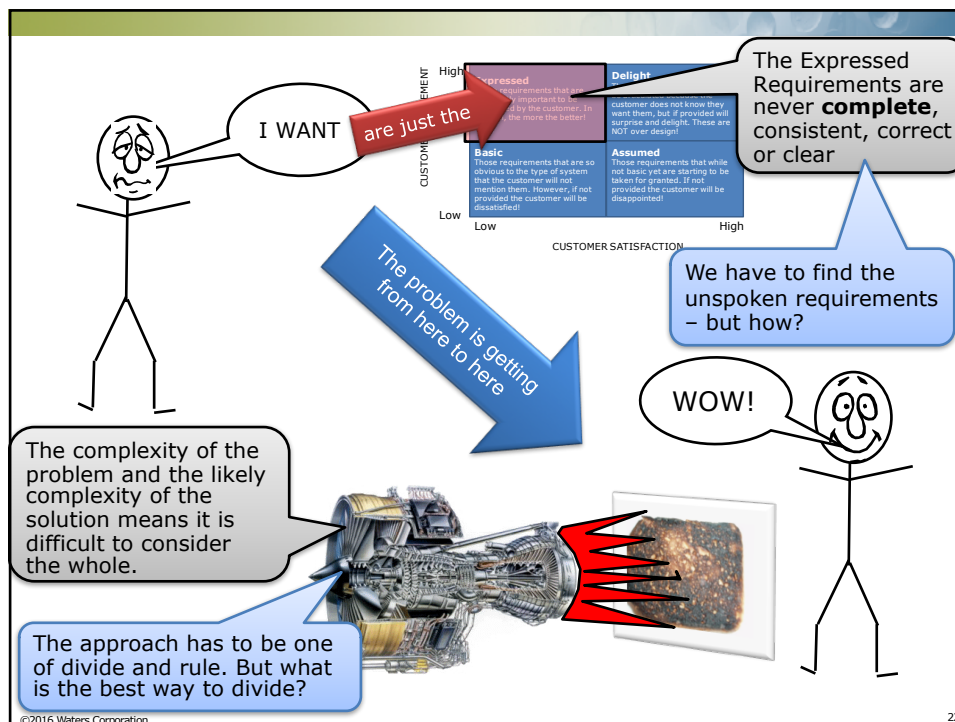
- Systems Engineering handles complexity by dividing the problem up into simpler yet integrated and interdependent sub-elements. It helps to be able to identify the sub-elements at an early stage, but how this is done is critical - the best way is to determine an invariant system architecture



For a given problem (Operational Requirement) there will be a fixed set of generic functions that will not change over time. These generic functions provide a rational and invariant basis for the division of that problem

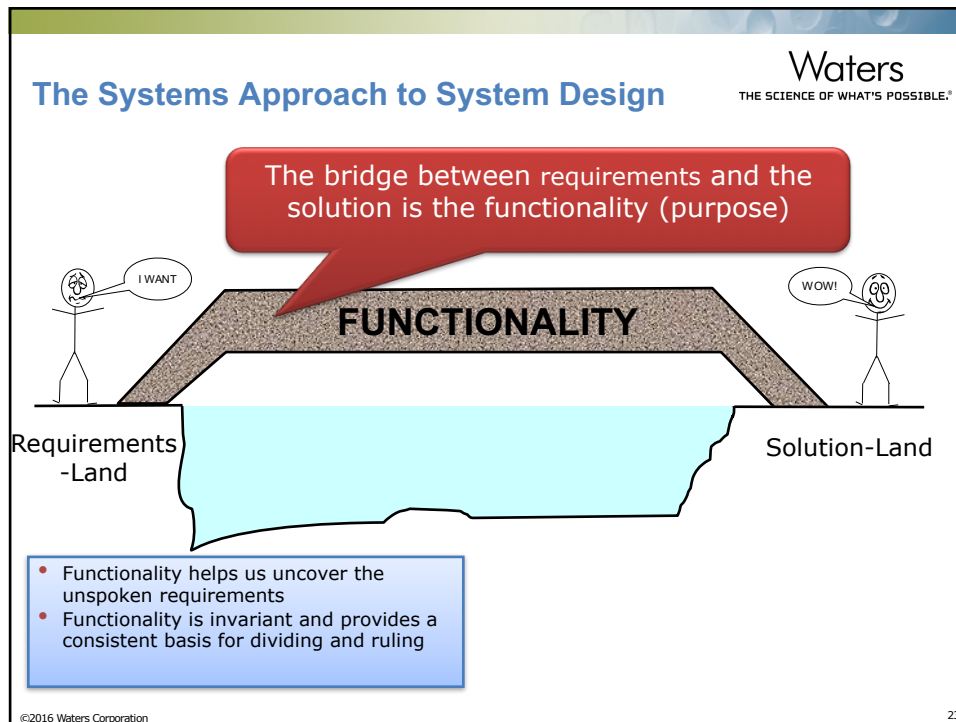
©2016 Waters Corporation

21



©2016 Waters Corporation

22



Functionality Delivers what the Customer Wants

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Generic System Functions

	Load Bread	Generate Heat	Apply Heat	Monitor Toasting	Control Toasting	Remove Heat	Unload Toast
Even Toasting	○	⊙	⊙	⊙	⊙	△	
Quick	○	○	△	△	○	△	△
Affordable	○	△	△	△	○	△	○
Easy to Use	⊙			○	⊙	△	⊙
Attractive	△		△		△		△
Reliable	○	⊙	○			△	○
etc							

What the customer wants

Key = ⊙ strong ○ medium △ weak

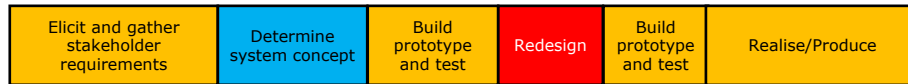
©2016 Waters Corporation 24

The Consequences

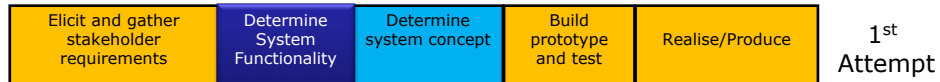
Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

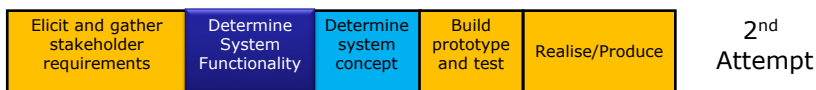
The Traditional approach appears “faster” we have an answer quickly BUT!



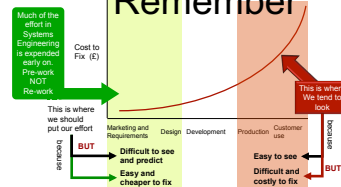
The Systems Engineering approach is initially slower! But faster in the end



And it gets quicker every time!



Remember



Thinking Functionally is the Pre-work that will reduce the Rework

©2016 Waters Corporation

25

Generic Functionality Helps Us Innovate: Automate Manual Functions

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Generic System Function	Implementation	Generic System Function	Implementation
Load bread	Stick bread slice on end of toasting fork	Load bread	Put bread into slots
Generate heat	Open coal fire	Generate heat	Electric elements
Apply heat	Use toasting fork to bring bread close to open fire	Apply heat	Lower bread next to elements using lever mechanism
Monitor toasting	Look at the bread from time to time	Monitor toasting	Look at the bread from time to time
Control toasting	Move bread to get even toasting	Control toasting	Timer
Remove heat	Use toasting fork to bring toast away from coal fire	Remove heat	Lift bread from elements using lever mechanism
Unload toast	Take toast off the toasting fork	Unload toast	Take toast from slots

The next generation of toasters will aim to automate monitoring!

©2016 Waters Corporation

26

Generic Functionality Helps Us Innovate: Expand the System Boundary

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



Generic System Function	Implementation
Load bread	Put bread into slots
Generate heat	Electric elements
Apply heat	Lower bread next to elements using lever mechanism
Monitor toasting	Look at the bread from time to time
Control toasting	Timer
Remove heat	Lift bread from elements using lever mechanism
Unload toast	Take toast from slots

Include external
boundary
functionality

Include external
boundary
functionality



©2016 Waters Corporation

27

Generic Functionality helps us innovate: Explore functional solutions

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTION – MEANS ANALYSIS CHART

System: Toaster		Sub-System		Date 1/1/2010		Author I M Burmt		Issue 1.0	
FUNCTION		MEANS							
Load bread	Stick bread slice on end of toasting fork	Put bread into slots	Put bread on to hinged door	Put bread on to conveyor	Load bread on to horizontal tray or drawer	Robotic Arm	Hold in free space		
Generate heat	Open coal fire	Electric element	Gas element	Laser	Nuclear element	Friction	Solar	Chemical	Microwave
Apply heat	Use toasting fork to bring bread close to heat source	Lower bread next to heat source using lever mechanism	Direct heat source at bread	Move heat source to bread	Slide bread next to heat source	Robotic arm brings bread to heat source	Conveyor carries bread to heat source		
Monitor toasting	Look at the bread from time to time	Visible light based reflectance sensor	IR based reflectance sensor	CCD and simple software based vision system	CCD and Neural net of Fuzzy logic	None			
Control toasting	Move bread to get even toasting	Timer	Multi-elements individually controllable	Fuzzy logic or rule based controller	Neural Net control with learning	Speed of Conveyor	Varying heat output of element		
Remove heat	Use toasting fork to bring toast away from heat source	Lift bread from heat source using lever mechanism	Un-slide toast from heat source	Timer cuts off heat source	Conveyor	Robot arm	Ejection mechanism	Mask heat source	
Unload toast	Take toast off of the toasting fork	Take toast from slots	Take toast from tray/ drawer	Toast falls off conveyor	Robotic arm removes toast from heat	Pull down hinged door			

©2016 Waters Corporation

28

Discussion – Generate Heat - Solutions

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- What are all the solutions to the GENERATE HEAT function

©2016 Waters Corporation 29

Generic Functionality helps us Create Holistic Solutions

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTION – MEANS ANALYSIS CHART

System: Toaster	Sub-System	Date 1/1/2010	Author I M Burnt	Issue 1.0
FUNCTION	MEANS			
Load bread	Slide bread slice on end of toaster fork	Put bread into slots	Put bread on to hinged door	Put bread on to conveyor
Generate heat	Open coil fire	Electric element	Gas element	Laser
Apply heat	Use toasting fork to bring bread close to heat source	Lower bread next to heat source using lever mechanism	Direct heat source at bread	Move heat source to bread
Monitor toasting	Look at the bread from time to time	Visible light based reflectance sensor	IR based reflectance sensor	CCD and simple software based vision system
Control toasting	Move bread to get even toasting	Timer	Multi-elements individually controllable	Fuzzy logic or rule based controller
Remove heat	Use toasting fork to bring toast away from heat source	Lift bread from heat source using lever mechanism	Un-slide toast from heat source	Timer cuts off heat source
Unload toast	Take toast off of the toasting fork	Take toast from slots	Take toast from tray/ drawer	Toast falls off conveyor

©2016 Waters Corporation 30

There is more to a system than its Generic Functionality

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The generic system functionality is not the complete set of system functionality – a system can have functionality that is not generic:
 - **Solution Dependent Functionality:** A particular generic function solution may require further solution specific functionality. For example Generate Heat can be solved by Electric Element but this solution demands that you “Earth Case” to protect the user
 - **Legislation Dependent Functionality:** legislation requires us to incorporate additional functionality. For example electric toasters have to isolate electricity
 - **Architectural Functionality:** At the system level the collected functional solutions require further functionality typically to support, align, contain and protect. These functions manifest as structures, casings, frameworks, chassis etc.
 - **Emergent Functionality:** Functionality aimed at managing emergent behaviour/properties. Usually relates to informing the user of failure situations

©2016 Waters Corporation

31

Finally: Functions can be Modelled Mathematically

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- In certain cases it is possible to model the system mathematically

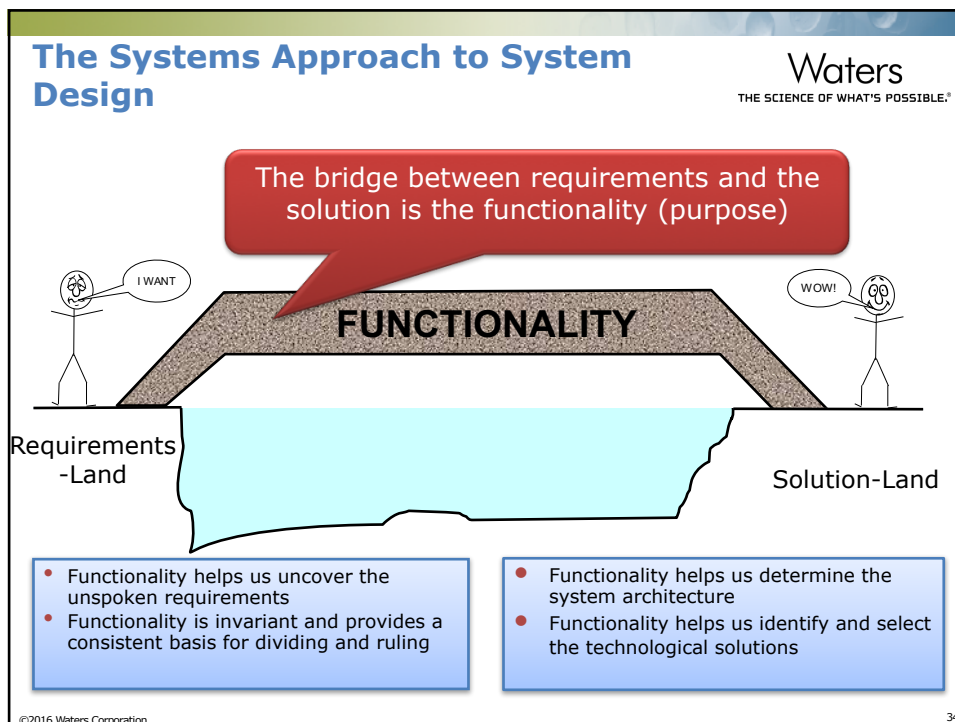
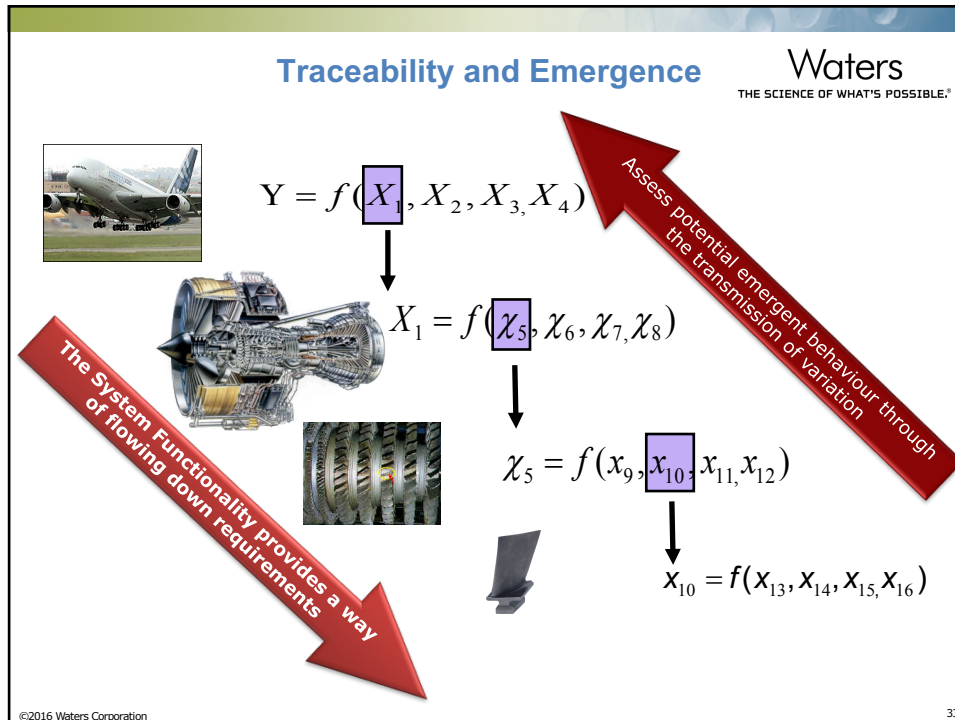


$$\text{Outputs}(Y) = \text{function}(\text{Inputs}(U), \text{State}(X))$$

- If we can model the system before it is built we can predict its behaviour – there are NO SURPRISES!
- What limits us is our mathematical ability to handle uncertainty

©2016 Waters Corporation

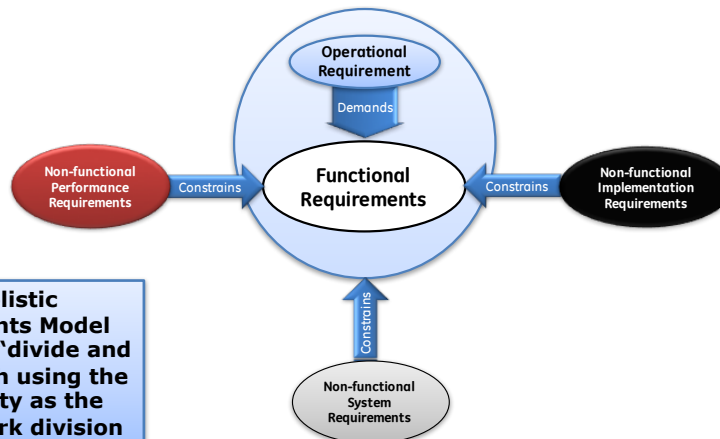
32



Holistic Requirements Model

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

The categories of requirements provide a complete and consistent framework that is driven by the Operational Requirements and has the Functional Requirements (what is to be done) at its heart



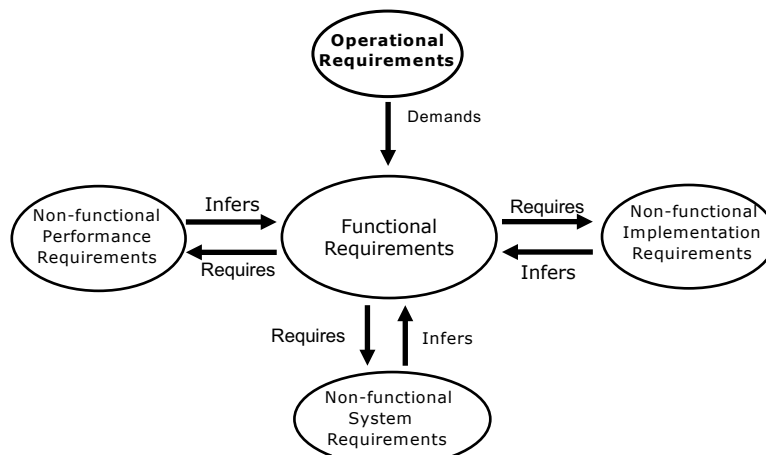
The Holistic Requirements Model allows us to "divide and rule" through using the functionality as the basis for work division

©2016 Waters Corporation

35

Using The Holistic Requirements Model

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



We can use the Holistic Requirements Model to deduce and infer the requirements that have not been expressed by the customer

©2016 Waters Corporation

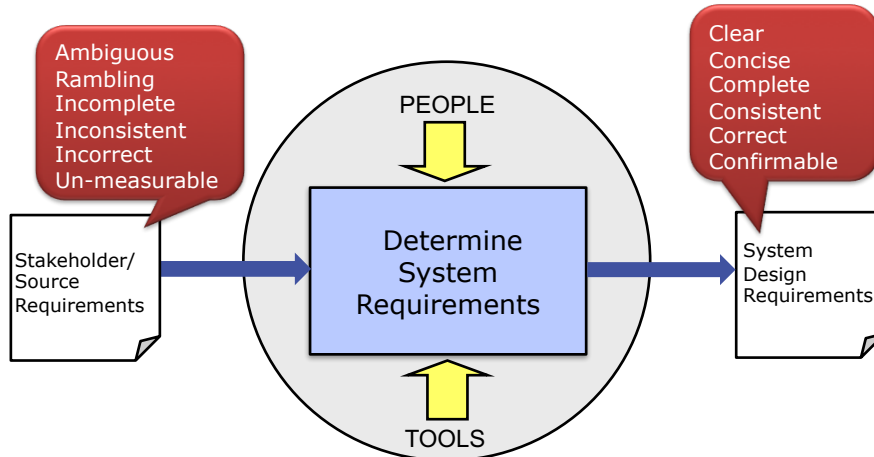
36

Holistic Requirements Model: Not the Silver Bullet

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

The Holistic Requirements Model provides a sound framework for understanding requirements and indicating where they are deficient but to achieve this requires



©2016 Waters Corporation

37

Making it Work: People and Tools

Waters

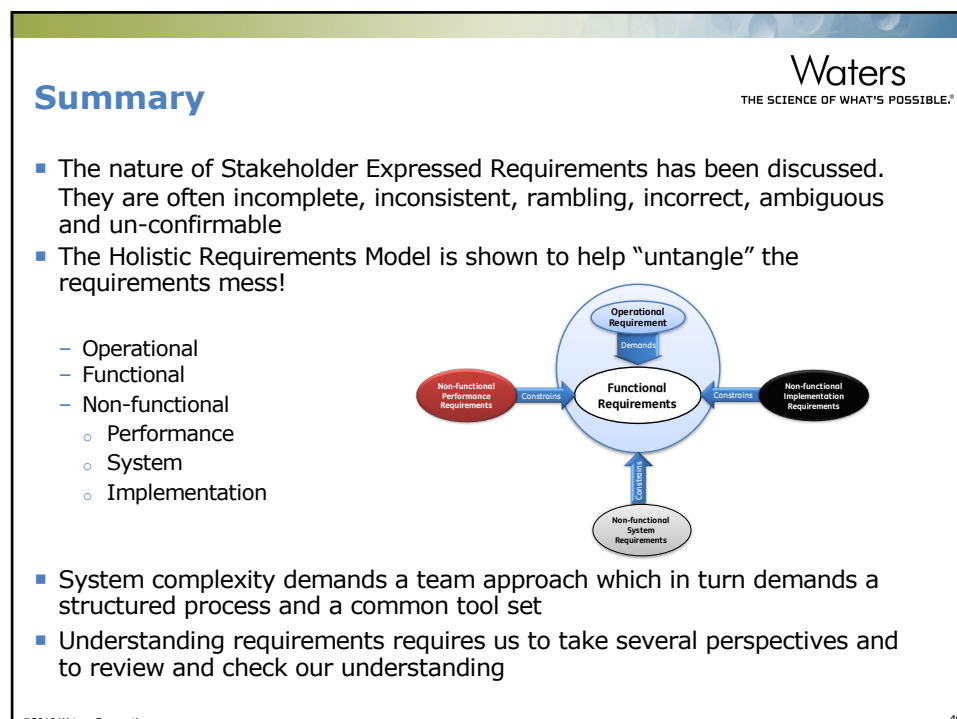
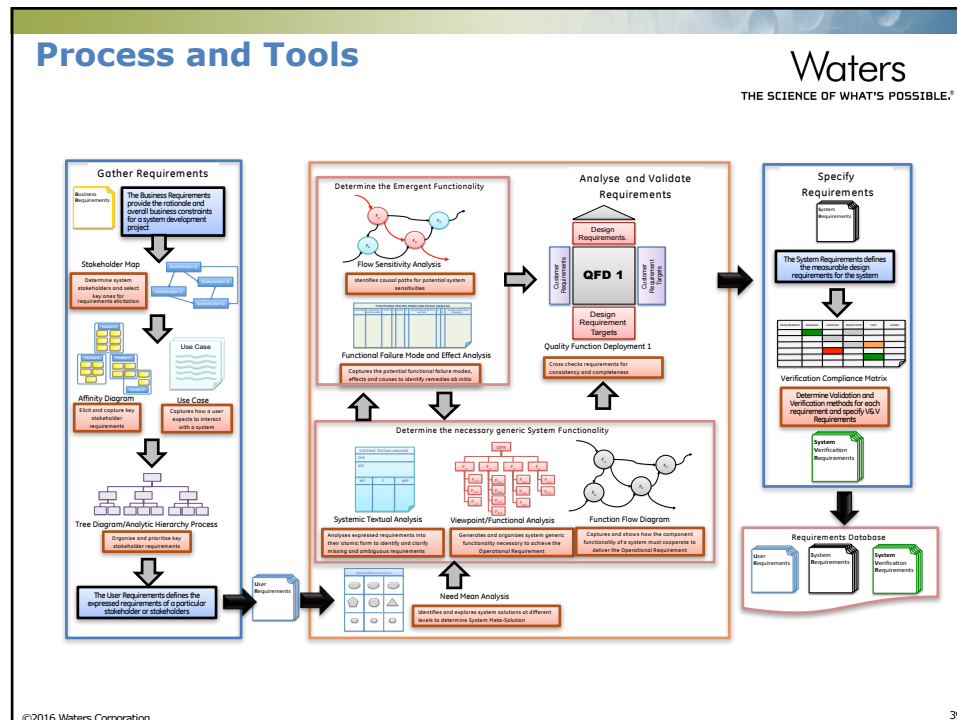
THE SCIENCE OF WHAT'S POSSIBLE.®

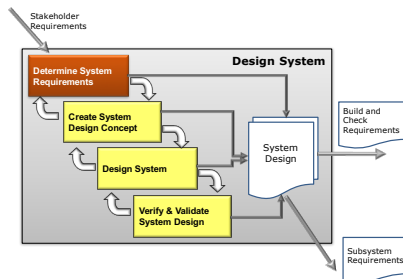
- Achieving completeness requires several viewpoints to be taken (demanding a multidisciplinary team approach)
- Achieving consistency, conciseness and clarity (removing ambiguity) requires tools that allow the team to communicate, to share information and have a common understanding to avoid having different perceptions but simple enough for the whole team to use
- Correctness means we need tools that can verify and validate requirements

There are many similar tools that can accomplish the same goal. We must NOT, however, introduce overly complex tools thinking that they will do a better job. The fundamental maxim of good Systems Engineering is KISS (Keep It Simple - Stupid)

©2016 Waters Corporation

38



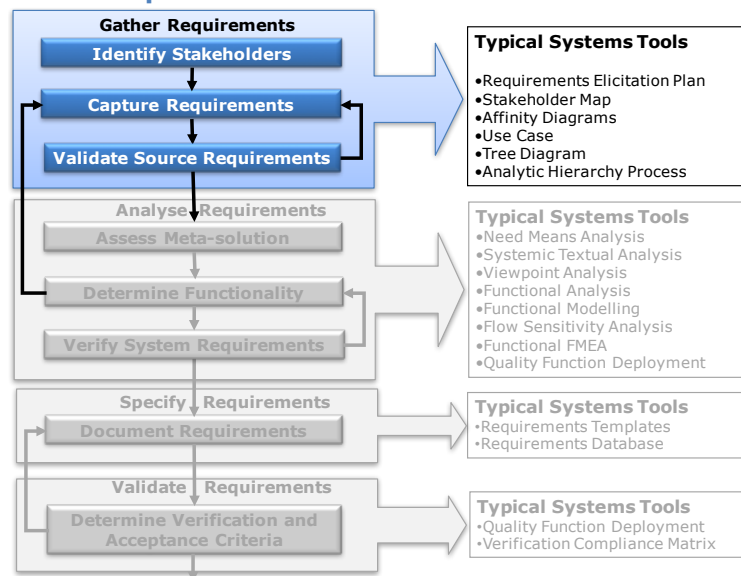


Determine System Requirements Gather Requirements

This Session:

- Explains the need for a Systems Approach to Gathering Requirements
- Presents a process for Gathering Requirements
- Describes and provides an opportunity to practise tools for Gathering Requirements
 - Requirements Elicitation Plan
 - Stakeholder Influence Map
 - Affinity Diagram
 - Use Case
 - Requirements Tree
 - Analytic Hierarchy Process

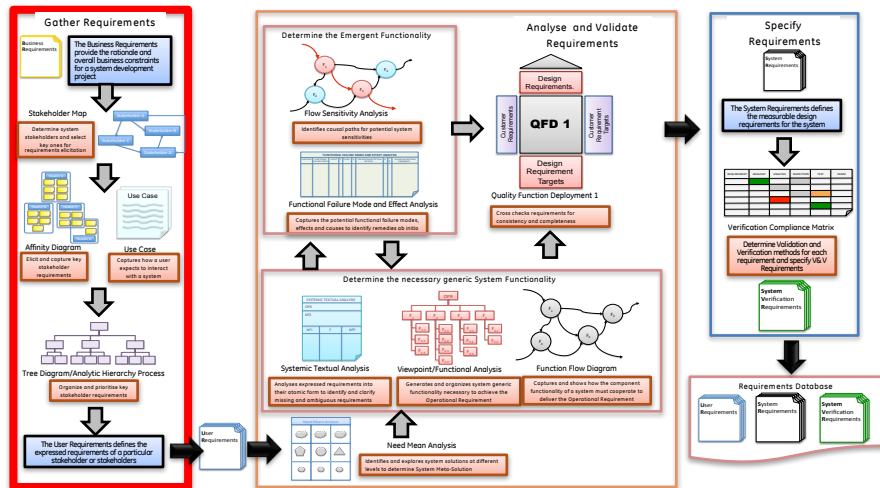
Determine Requirements: Tasks and Tools



Process and Tools

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

3

Customer or Stakeholder?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Confusion Reigns:

- We often use “customer” to mean the person(s) who buys or uses a system (whether it is a product or service based system). Hence the phrase “Customer Requirements” refers to those expressed needs and wants of the system user or purchaser
- In some situations, where they are different people, we also differentiate the User from the Purchaser
- Within a complex supply chain it is also typical to call the person who expresses requirements for a subsystem or component the “customer”
- If we take a life-cycle view of a system, subsystem etc., we find that there are many more interested parties. These people also have requirements for the system of interest and include:
 - Sales, Design, Manufacture, Test, Support and Maintainers etc.
- There can be other people who have an interest in a system that are external to it such as legislators, pressures groups etc. These may also have requirements for a particular system, subsystem etc.
- Collectively all these people are called stakeholders
- It is also usual to include the system as a stakeholder
- It is important to be clear who is who! – every requirement has a source and that needs to be captured

©2016 Waters Corporation

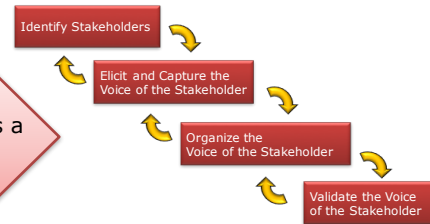
4

What's the problem?

- All systems have a range of stakeholders
- All the stakeholders will have their own expectations and needs
- Understanding stakeholders' expectations and needs is therefore about:

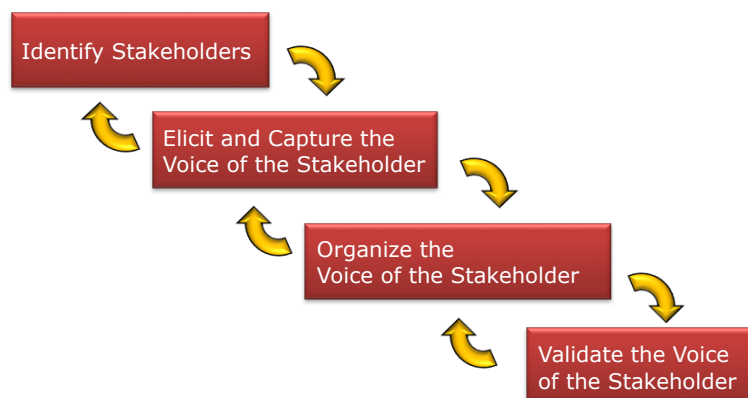
- **Identifying** all the stakeholders
- **Eliciting** and **capturing** their expectations
- **Organizing** their expectations
- **Validating** their expectations

There is a Process



Note that sometimes our direct customers will provide a detailed set of requirements. We should never assume that this is complete – we still need to undertake the above

Gather Requirements: An Iterative Process



Planning to Gather Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Requirements Elicitation Plan					
Stakeholder	Requirement Elicitation Method	Source Documents	Responsible	Validation Method	Responsible

Requirements elicitation is not a random or haphazard process it needs to be carefully thought through and planned in detail. The above template captures the outcomes of that planning activity – it defines the requirements gathering strategy

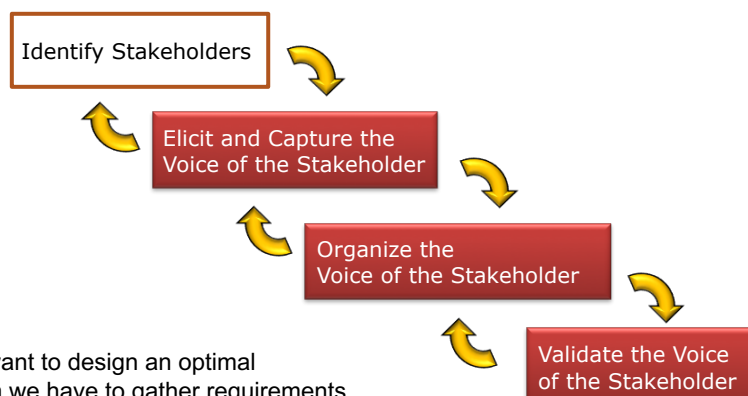
©2016 Waters Corporation

7

Identify Stakeholders

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



If we want to design an optimal system we have to gather requirements from all potential stakeholders. In order to do this we need to identify the potential stakeholders
A useful tool is a **Stakeholder Influence Map**

©2016 Waters Corporation

8

THE SCIENCE OF WHAT'S POSSIBLE.®

-
- The diagram illustrates the stakeholder map for 'The Company'. It shows the following relationships:
- Internal Stakeholders:**
 - Direct Operators** and **Indirect Operators** are connected by a double-headed arrow labeled 'Rents'.
 - Direct Operators** provide 'Orders' to **Airframers** and receive 'Rents' from them.
 - Indirect Operators** provide 'Fuel Price' to **Fuel Suppliers**.
 - Fuel Suppliers** provide 'Fuel Price' to **Direct Operators**.
 - Direct Operators** provide 'Rents' to **The Company**.
 - Indirect Operators** provide 'Rents' to **The Company**.
 - Airframers** provide 'Access To Market' to **The Company**.
 - Fuel Suppliers** provide 'Fuel Price' to **The Company**.
 - Researchers** provide 'Future Technology' to **The Company**.
 - External Stakeholders:**
 - Competitors** and **Pressure Groups** are connected by a double-headed arrow labeled 'Influence/Lobby'.
 - Pressure Groups** provide 'Lobby' to **Legislators**.
 - Legislators** provide 'Legislation' to **The Company**.
 - The Company** provides 'Future profits: Business INT/TRL' to **The Business**.
 - The Company** provides 'Future Orders' to **Supply Chain**.
 - The Company** provides 'Future Technology' to **Future Technology**.
 - The Company** provides 'Research/Innovation' to **Researchers**.
 - Competitors** provide 'Influence/Returns' to **The Company**.
 - Pressure Groups** provide 'Influence/Returns' to **The Company**.
 - Legislators** provide 'Influence/Returns' to **The Company**.
 - The Business** provides 'Influence/Returns' to **The Company**.
 - Future profits: Business INT/TRL** provides 'Influence/Returns' to **The Company**.
 - Future Orders** provides 'Influence/Returns' to **The Company**.
 - Future Technology** provides 'Influence/Returns' to **The Company**.
 - Research/Innovation** provides 'Influence/Returns' to **The Company**.
 - Advise** provides 'Influence/Returns' to **The Company**.

©2016 Waters Corporation

9

THE SCIENCE OF WHAT'S POSSIBLE.®



10

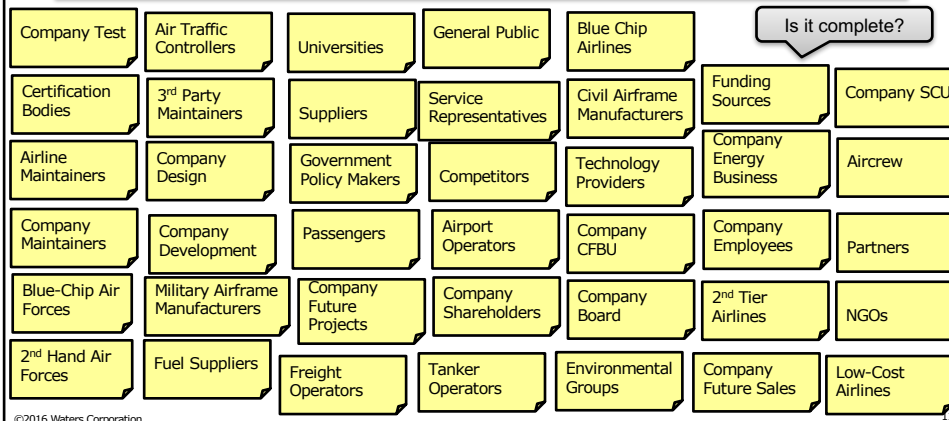
STEP 1: Identify all potential System potential stakeholders

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Usually performed as a simple Brainstorm to capture potential stakeholders on sticky notes (one stakeholder per sticky note)
- **A stakeholder is anybody who comes into contact with, or has an interest in, the system throughout its entire life**

Future Civil A/C Power Plant: brainstorm of potential Stakeholders



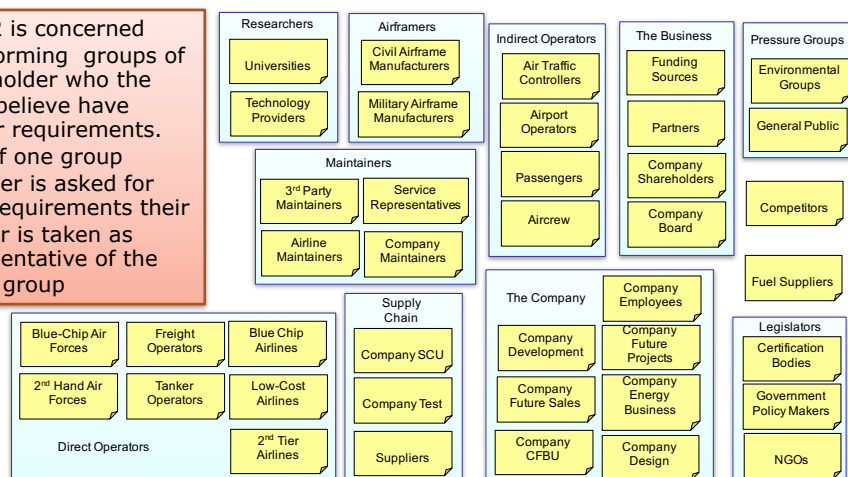
©2016 Waters Corporation

Step 2: Identify Groups of Stakeholders (based on similar/common requirements)

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Step 2 is concerned with forming groups of stakeholder who the team believe have similar requirements. Such if one group member is asked for their requirements their answer is taken as representative of the whole group



©2016 Waters Corporation

12

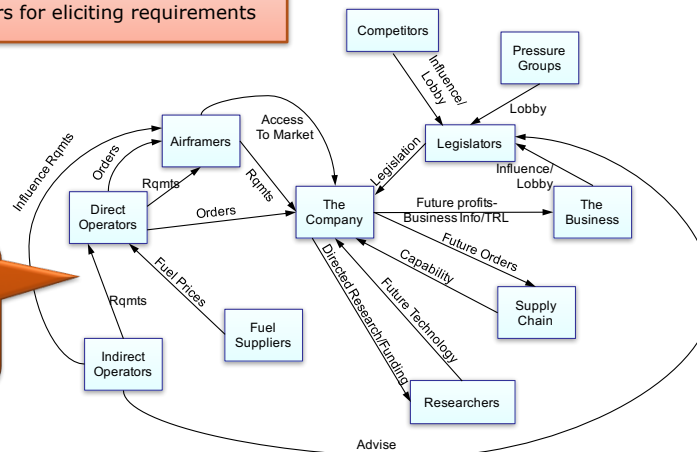
Step 3: Construct Stakeholder Influence Map

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

The Stakeholder Influence Map is not essential to identify the stakeholders – but is often worthwhile constructing to:

- understand the system's market environment
- prioritize Stakeholders for eliciting requirements

If one stakeholder group is felt to influence another capture with a "named" arrow showing the direction of the influence and a describing name

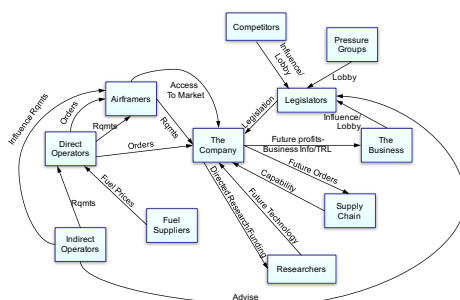


©2016 Waters Corporation

13

Exercise – Stakeholder Influence Map

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



- In small teams
 - Construct a Stakeholder Influence Map for the system described to you by the Trainer
 - Be prepared to share your outcome with the rest of the class
- Time 35 minutes
 - 15 minutes brainstorm
 - 10 minutes grouping
 - 10 minutes constructing influence map

©2016 Waters Corporation

14

So What? Step 4: Review Stakeholder Influence Map

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- For any system there is potentially a large number of stakeholders. In the ideal world we should wish to elicit and capture requirements from all these stakeholders. Pragmatically, we are likely to have neither the time nor the money to do so and therefore need to be selective – but which ones?
- We use the Stakeholder Influence Map to decide:
 - Which groups of stakeholders MUST we get requirements from?
 - Which groups of stakeholders can be ignored (this time and why?)
 - Which group of stakeholders SHOULD we get requirements from



©2016 Waters Corporation

15

Stakeholder Influence Map Summary

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- | | |
|---|---|
| <ul style="list-style-type: none"> ■ Pros <ul style="list-style-type: none"> – Simple - anybody can contribute – If performed in a team can promote <ul style="list-style-type: none"> ○ Discussion ○ Thinking – Identifies stakeholders – Helps to understand relationships and influences between stakeholders – Helps determine which stakeholders to elicit requirements from | <ul style="list-style-type: none"> ■ Cons <ul style="list-style-type: none"> – Deciding who does this – who should be involved – Assumes we know what the system is that we are intending to design and develop: <ul style="list-style-type: none"> ○ The scope of the system could change after collecting some requirements thereby introducing new stakeholders from which we will have to collect requirements – expect iteration |
|---|---|

©2016 Waters Corporation

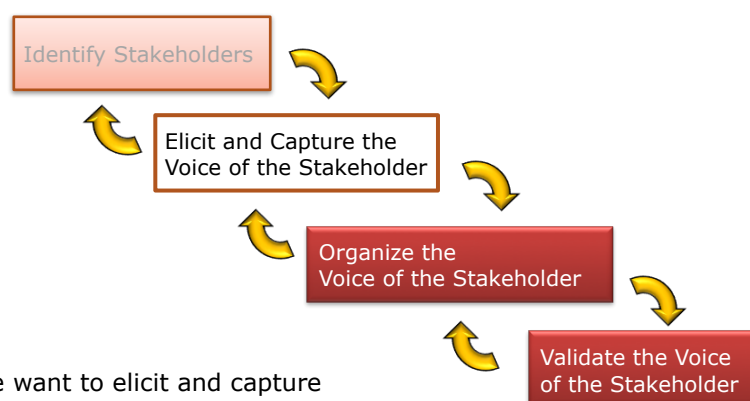
16

Update Requirements Elicitation Plan

Requirements Elicitation Plan					
Stakeholder	Requirement Elicitation Method	Source Documents	Responsible	Validation Method	Responsible

Add the Stakeholders, these can be groups of stakeholders from the Stakeholder Influence Map, individual stakeholders or any suitable combination

Elicit and Capture the Voice of the Stakeholder



If we want to elicit and capture stakeholder needs and expectations – it is not just a question of asking “what do you want?” We have to understand their issues and concerns - we have to **listen**

Remember Spoken vs. Unspoken

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

If we ask a stakeholder "what do you want?" we will get only those requirements that are important to them at that point in time: **the expressed requirements**

CUSTOMER EXCITEMENT
High
Low

Expressed

Those requirements that are sufficiently important to be verbalised by the customer. In general, the more the better!

Delight

Those requirements that cannot be articulated because the customer does not know they want them, but if provided will surprise and delight. These are NOT over design!

Basic

Those requirements that are so obvious to the type of system that the customer will not mention them. However, if not provided the customer will be dissatisfied!

Assumed

Those requirements that while not basic yet are starting to be taken for granted. If not provided the customer will be disappointed!

Low

High

CUSTOMER SATISFACTION

There are other requirements that are left unspoken

Conclusion: Stakeholders only tell us their expressed requirements; typically only 20% - 30% of the full set of requirements. They do not tell us the basic, assumed or delighter requirements (spoken vs. unspoken requirements). We need to determine the missing ones

©2016 Waters Corporation

19

Different Approaches for Different Types

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- We need to remember that:
 - Different stakeholders have different needs and expectations
 - Different stakeholders have different priorities
- Stakeholders often express their needs, expectations/issues etc. in vague general terms and tell us what they think is important today

Spoken Expressed

- User/Customer supplied requirements
- Face-to-face interviews/Focus Groups
- Pretend to be the stakeholder - Put yourself in their shoes

Unspoken Basic and Assumed

- Customer complaints
- Manufacturing issues/failures
- Determine functional requirements

Unspoken Excitement (Delighters!)

- Gemba visits (watch the user using the system)
- Customer modifications
- Systems Thinking (purpose based thinking)

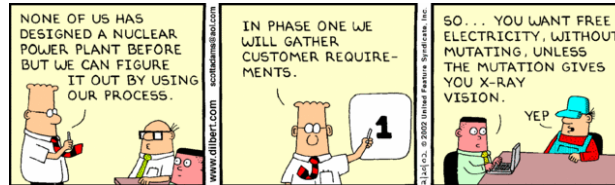


©2016 Waters Corporation

20

Approach 1: Asking the Stakeholder

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



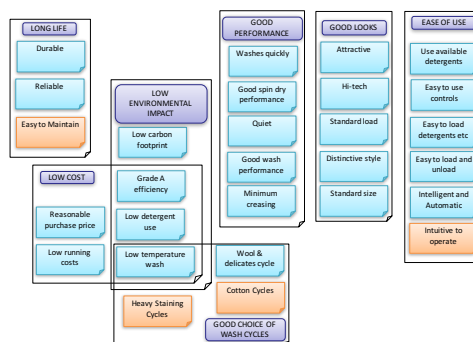
- The answers you elicit will depend on:
 - The questions you ask
 - What they think you want to know
- For all stakeholders, you should ask questions and listen to the answer
- Typical questions are:
 - What are your requirements for the system (product, process or service)?
 - What do you want?
 - What don't you want?
 - Are there any features/aspects in the previous design you wish to keep?
 - Are there any features/aspects in the previous design that must be avoided?
 - What are your programme requirements? (time, cost, resource etc.)
- A good tool for eliciting and capturing requirements is the **Affinity Diagram**

©2016 Waters Corporation

21

Affinity Diagrams

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

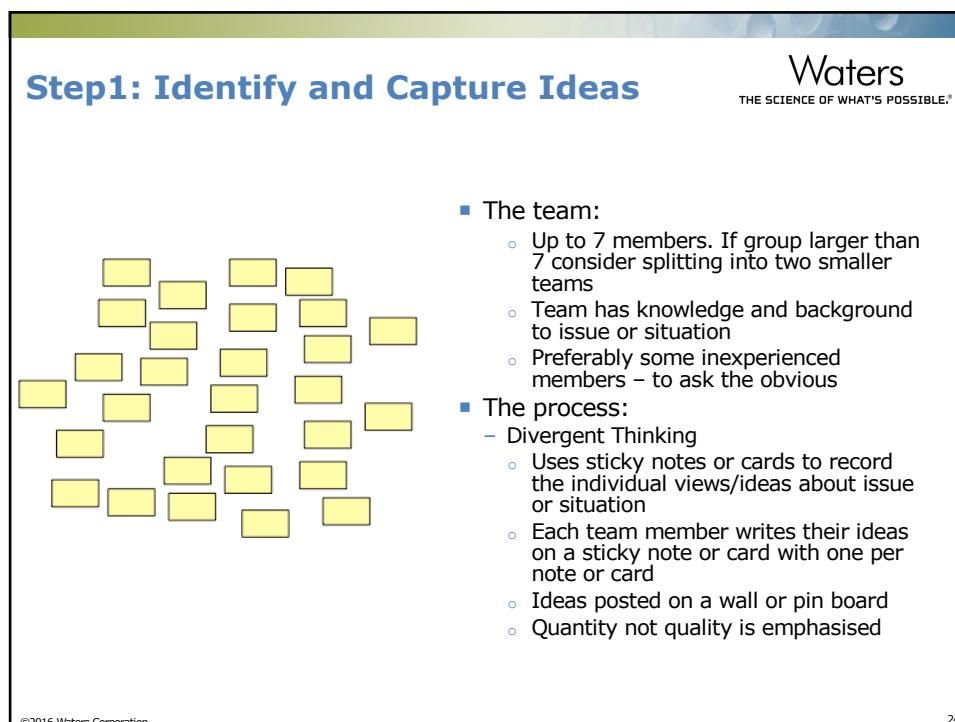
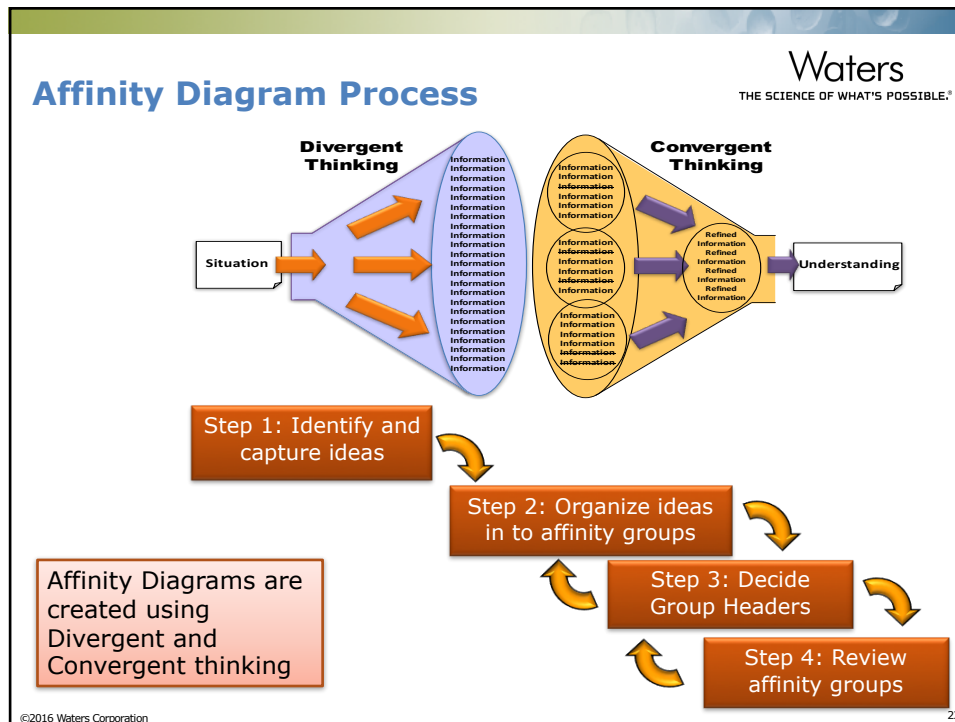


In the example Affinity Diagram above, the individual requirements are captured on sticky notes or cards. These have been put in to form groups of requirements that have an affinity. Each group is given a header to create a higher-level requirement

- What is it?
 - Affinity Diagramming is a simple tool that integrates both divergent and convergent thinking. It allows a team or individual to
 - Generate ideas about a situation or problem
 - Organize a large number of ideas, opinions, issues into meaningful groups
- Why do it?
 - An alternative to brainstorming when
 - The situation under investigation is too large or complex
 - There is a need to get agreement
 - There is a need to communicate ideas to others
 - There are issues with group personalities

©2016 Waters Corporation

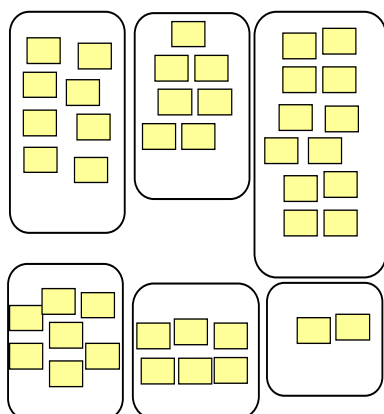
22



Step 2: Organize Ideas into Affinity Groups

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Should be done in SILENCE!
- The team organize the sticky notes or cards into related groups (themes) – that have an affinity
- Everyone moves sticky notes or cards to place them with others with which they consider they have affinities
- Allow new groups to emerge if a sticky note “moves around” a lot
- Consider sub-dividing large groups (>10 items)

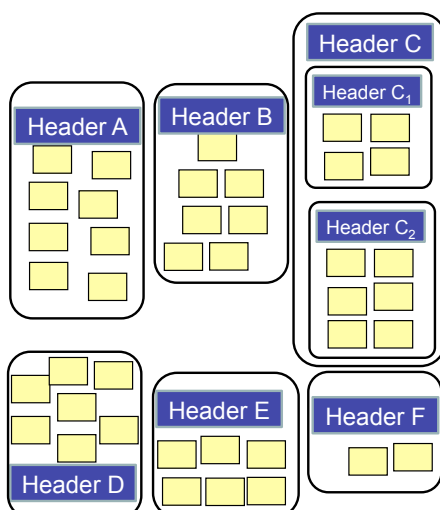
©2016 Waters Corporation

25

Step 3: Decide Group Headers

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Each group should be given a header that describes the contents of the group
- This may be an existing sticky note or card
- If you have difficulty finding a suitable header – question the grouping
- Make sure the “headers” reflect the context of the Affinity Diagram

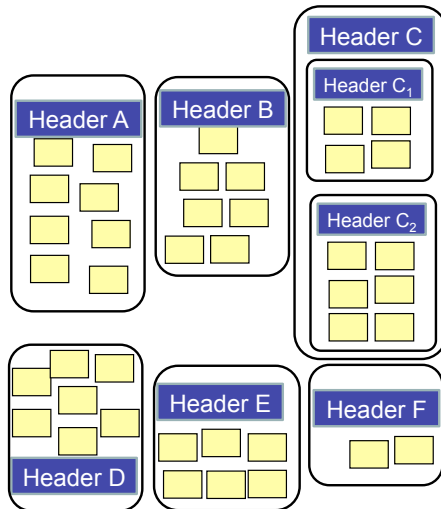
©2016 Waters Corporation

26

Step 4: Review Affinity Groups

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Review groups to:
 - Remove duplications
 - Sticky notes or cards that are the same
 - Sticky notes or cards that are similar
 - Consider each group for new items
 - Consider alternative groupings

©2016 Waters Corporation

27

Affinity Diagrams in Gathering Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The Affinity Diagram is tool for generating and organizing ideas and information
- It is a very powerful alternative to Brainstorming
- It can be used for:
 - Root cause analysis
 - Solution generation
 - Goal and objective definition
 - Task and activity definition
 - Requirements capture
- It is particularly useful when gathering requirements to elicit and capture requirements from stakeholders because:
 - Encourages the stakeholder to write their own requirements so we capture their words
 - Overcome any individual dominating the requirements capture session
 - Allows everybody to contribute
 - Organizes the requirements in a simple structure

©2016 Waters Corporation

28

Example: Step1: Identify and Capture Requirements

A Focus Group (a small team of Users) was used to brainstorm and organize requirements for a washing machine

Step 1:
Team members individually capture their thoughts on sticky notes and put on a wall



©2016 Waters Corporation

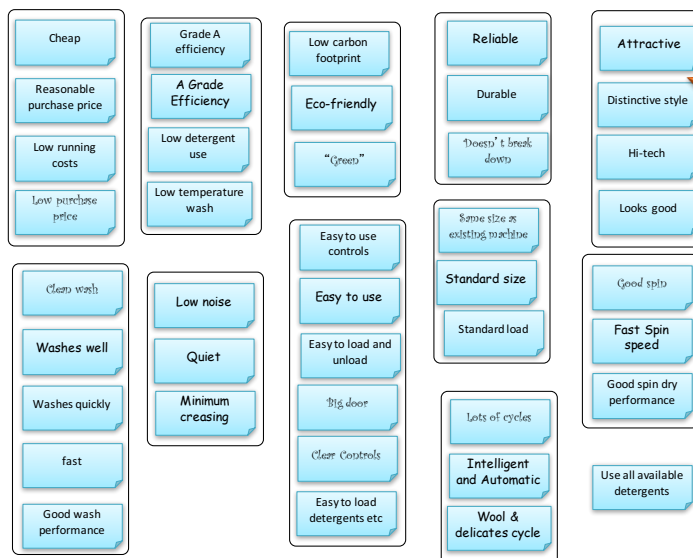
29

Step 2: Organize Requirements in to Affinity Groups

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Step 2:
The team, in silence, organize the requirements into groups that have affinities



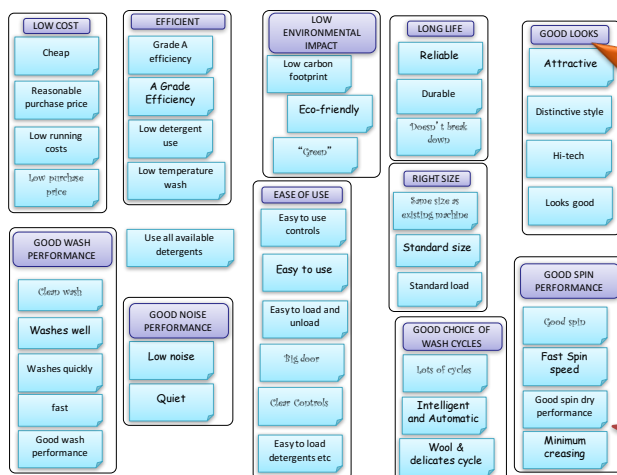
©2016 Waters Corporation

30

Step 3: Decide Group Headers

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Step 3:
The team, decide headers for each group, the header may be an existing sticky note. The chosen headers can lead to a re-organization of the sticky notes

In this requirements example the headers themselves should be requirements!

©2016 Waters Corporation

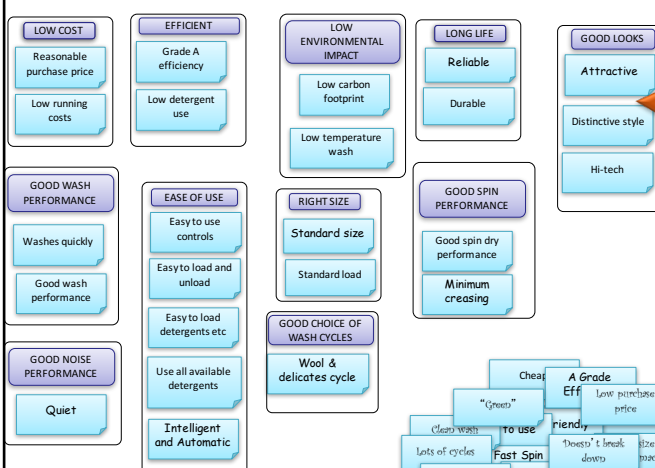
31

Step 4: Review Affinity Groups

a) Remove/Reconcile Duplications

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Step 4a)
The sticky notes are reviewed and duplications discussed and where appropriate removed and revised

Removed
Duplications

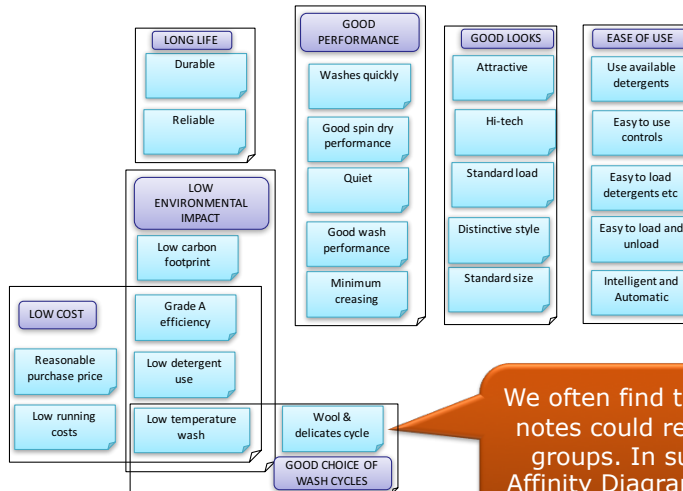
©2016 Waters Corporation

32

Complex Groupings

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



We often find that some sticky notes could reside in several groups. In such situations Affinity Diagrams can be used to capture complex groupings

©2016 Waters Corporation

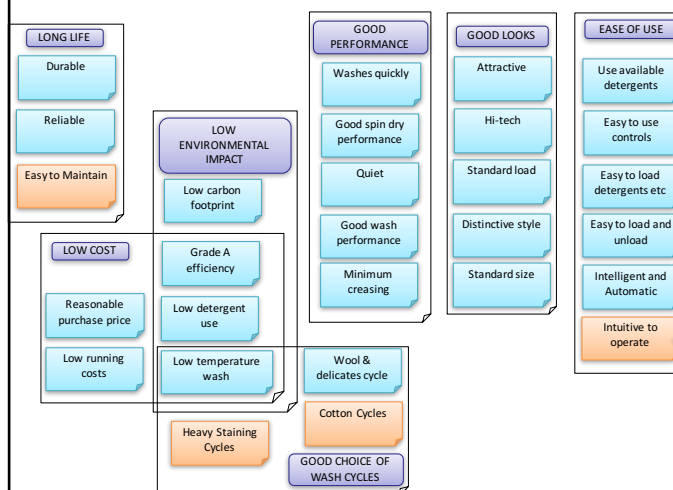
33

Step 4: Review Affinity Groups

b) Review Groups for Further Elements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Step 4b)
The groups are reviewed for further items

Have we got all the requirements that relate to "ease of use?"

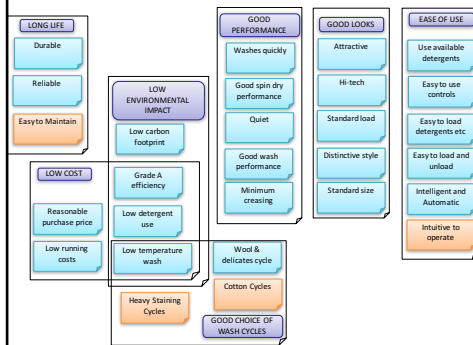
©2016 Waters Corporation

34

Exercise: Affinity Diagram

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- In small Teams
 - Create an Affinity Diagram for the User Requirements of the system described by the Trainer
- Time 25 minutes

©2016 Waters Corporation

35

Affinity Diagram Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Pros
 - Easy to use
 - Generates lots of ideas (two heads are better than one)
 - Promotes discussion
 - Consensus view – a shared and common understanding
 - Organized output
 - Useful in hierarchical groups – everybody contributes “evenly”
- Cons
 - Difficulties in finding agreed affinities
 - Deciding on the headers
 - Don’t spend the time checking out duplications
 - Don’t spend the time investigating complex groupings
 - Don’t spend the time seeking additional elements
 - Output format not intuitive to novice reader

©2016 Waters Corporation

36

Probing for Requirements

- Stakeholders are often biased by their prior experience and couch the requirements in terms of what they know – i.e. a solution-based requirement (NFI): e.g. I want a touch screen display or Grade A energy efficiency
- We need to identify such situations and ask “probing” questions - “why do you want a touch screen display?” to get at the true requirement – which might be “easy to use”

Approach 2: The Use Case

- What is it?
 - An alternative to “what do you want?” is to ask the stakeholder (particularly the user) to describe in a story how they expect to USE the system (hence the name Use Case)
- Why use it?
 - It is a natural intuitive communication
 - The stories situate requirements in context
 - Define logical chunks of functionality
 - Can help define both wanted and unwanted behaviour
- How to do it
 - Ask the user to describe in a time-line how they expect to interact with the system and how they expect the system to respond. For moded or multipurpose systems it may be necessary to have a number of Use Cases for different “scenarios”



USE Case for a Pedestrian Crossing

- Pedestrian approaches crossing that informs pedestrian not to cross
- Pedestrian makes cross request
- System waits, shows red light to cars, waits, informs pedestrian it is safe to cross
- Pedestrian crosses
- System continues to inform pedestrian it is safe to cross
- System informs pedestrian it is not safe to cross, waits, shows green light to cars

In the example above, the Use Case describes, generically, how a user (pedestrian) has to interact with the crossing to safely cross

Solution to Generic

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

One of the potential drawbacks with Use Cases is that it elicits a description of an existing system design. In such situations it may be necessary to "abstract" the Use Case to get the generic functions

Use Case for Pedestrian Crossing

- Pedestrian approaches crossing that shows red light to pedestrian
- Pedestrian presses cross button, button lights to indicate press successful
- System waits, shows red light to cars, waits, shows green light to pedestrians
- Pedestrian crosses
- System gives visual and audible warning
- System shows red light to pedestrians, waits, shows green light to cars



Generic Version

Use Case for a generic Pedestrian Crossing

- Pedestrian approaches crossing that informs pedestrian not to cross
- Pedestrian makes cross request
- System waits, shows red light to cars, waits, informs pedestrian it is safe to cross
- Pedestrian crosses
- System continues to inform pedestrian it is safe to cross
- System informs pedestrian it is not safe to cross, waits, shows green light to cars

Why still Red and Green lights for the cars?

©2016 Waters Corporation

39

Exercise: Use Cases

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Generic Version

Use Case for a generic Pedestrian Crossing

- Pedestrian approaches crossing that informs pedestrian not to cross
- Pedestrian makes cross request
- System waits, shows red light to cars, waits, informs pedestrian it is safe to cross
- Pedestrian crosses
- System continues to inform pedestrian it is safe to cross
- System informs pedestrian it is not safe to cross, waits, shows green light to cars

■ In Small Teams

- Write a simple Use Case as to who the User would expect to interact with the system under development

■ Time 20 Minutes

©2016 Waters Corporation

40

Use Case Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Pros
 - Can extract system functionality from the stakeholder
 - Intuitive to stakeholder
 - Can be useful for unprecedented systems to capture the perceived concept of operation
- Cons
 - Stakeholder often relates to and describes an existing design solution – we may need to “abstract” the Use Case to determine the generic system functions
 - Getting the “right” level for the functionality – not too high or too low.
 - Not very useful for embedded, automatic or some unprecedented systems. The for example asking a pilot to write a Use Case for an aircraft power plant will not provide much information
 - Does not elicit Non-Functional constraint type requirements

©2016 Waters Corporation

41

Approach 3: Put Yourself in Their Shoes

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- In many situations, gathering requirements is a lengthy activity - but it is possible to generate requirements (and therefore proceed with the system design) by “putting yourself in their shoes”
- This will certainly generate some realistic requirements but there is the risk that these are what we “want” and not what stakeholders actually want!
- Again an Affinity Diagram can be used to generate, capture and organize the requirements

Beware

SAAB engineers were convinced that customers needed a glove box that could fit an A4 file. This could only be achieved through a design that compromised access. Customers hated it at production evaluation and it had to be re-designed at x1000 times the original development costs.

The lesson is: -
Validate your
view of
stakeholder
requirements

©2016 Waters Corporation

42

Approach 4: Gemba Visits

- People can find it hard to verbally articulate their thoughts – other methods are necessary
- People very often accept a sub-standard level of performance and don't believe there are problems when using a product because 'I thought this was it'.
- People are flexible and learn to adapt to using a system of inferior quality.
- Observing end users (Gemba Visit) is one way to unearth some unarticulated needs.
- Observing end users with other stakeholders also aids our understanding of their needs and requirements.



©2016 Waters Corporation

43

Other Sources of Requirements

Gathering requirements does not have to be proactive, some requirements will arrive come what may. If we do a "bad job" stakeholders will complain and these are a good source of basic and assumed requirements

Proactive systems

- You need to put effort into gathering the information
 - Interviews and focus groups
 - Affinity and Tree Diagrams
 - Use Cases and Scenarios
 - Pretend to be the stakeholder (Put yourself in their shoes)
 - Affinity and Tree Diagrams
 - Use Cases and Scenarios
 - Direct observation (Gemba visits)
 - Market Research and Benchmarking
- This information tells us about past, current and future needs

Reactive systems

- Information comes to you whether you take action or not
 - Complaints
 - Problem or service messages
 - Technical support messages
 - Sales team reporting
 - System return information
 - Warranty claims
- This information tells us about current and past problems
- Very useful in determining unspoken requirements and in identifying delighters

©2016 Waters Corporation

44

Requirements Elicitation

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- We have to think carefully about the “best” elicitation method or methods for a particular stakeholder – it’s easy to try and “do all of them”
- It is important to consider the drivers:
 - The business case
 - The stage of design
 - The point in the life cycle
 - The project/programme requirements – particularly timescales

©2016 Waters Corporation

45

Update Requirements Elicitation Plan

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Requirements Elicitation Plan					
Stakeholder	Requirement Elicitation Method	Source Documents	Responsible	Validation Method	Responsible

Update Requirements Elicitation Plan with the elicitation methods chosen for each stakeholder or stakeholder group

Any documents that contain potential requirement sources – e.g. Customer Specifications, Customer complaints, Manufacturing defect reports should be noted here

©2016 Waters Corporation

46

THE SCIENCE OF WHAT'S POSSIBLE.®

- When stakeholders speak their requirements we will have to capture them
- This can be done as:
 - A list of requirements
 - We should always try to capture the exact words spoken by the stakeholder
 - When interviewing stakeholders work in pairs: one asking the questions one writing down the reply
 - Affinity Diagrams
 - Captures the actual words of the stakeholder but they may need some help
 - A Use Case
 - A text based description (story) of how a stakeholder expects a system to respond to a request from a stakeholder – particularly useful for software/process intensive systems
 - A Requirements Model
 - Builds up a pictorial view of the stakeholders' requirements – see functional modelling later

47

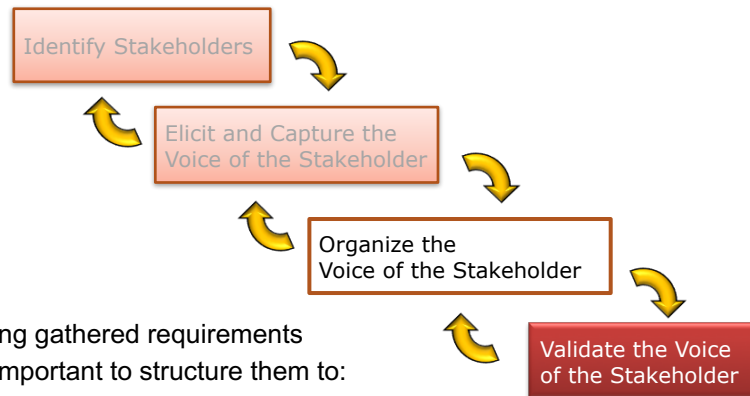
THE SCIENCE OF WHAT'S POSSIBLE.®

- If we are eliciting requirements from a Stakeholder we can also ask how important the requirements are. Typically on a 1 – 5 scale (5 more important)

48

An Iterative Process

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



Having gathered requirements
it is important to structure them to:

- help understand relationships between requirements
- simplify the requirements
- help manage the requirements

A useful tool is a **Tree Diagram**

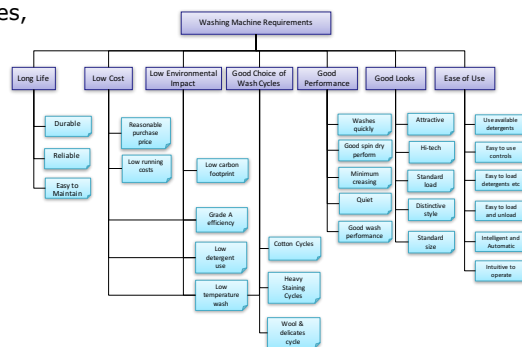
©2016 Waters Corporation

49

Tree Diagram (alias Systematic Diagram)

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- What is it?
 - A simple tool that helps with the logical hierarchical breakdown of ideas, situations, events, activities, or things
- Why do it?
 - To manage complexity through sub-division
- Presents information in an intuitive format
- Can be used to organize:
 - Goal or objective
 - Task or activity
 - Thing or product
 - Need or requirement
 - Problem or Issue



In the example Tree Diagram above, the requirements for a washing machine are shown in a simple hierarchy.

©2016 Waters Corporation

50

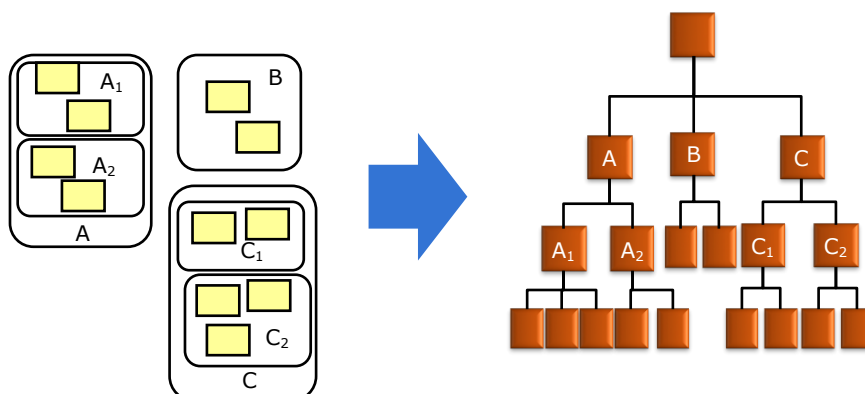
How to do it: Tree Diagrams from Affinity Diagrams

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Affinity Diagrams

- In some situations it is easier and more convenient to start with a brainstorming session to generate ideas that can be logically grouped (and sub-grouped)
- This grouping can be used to form the basic Tree Diagram



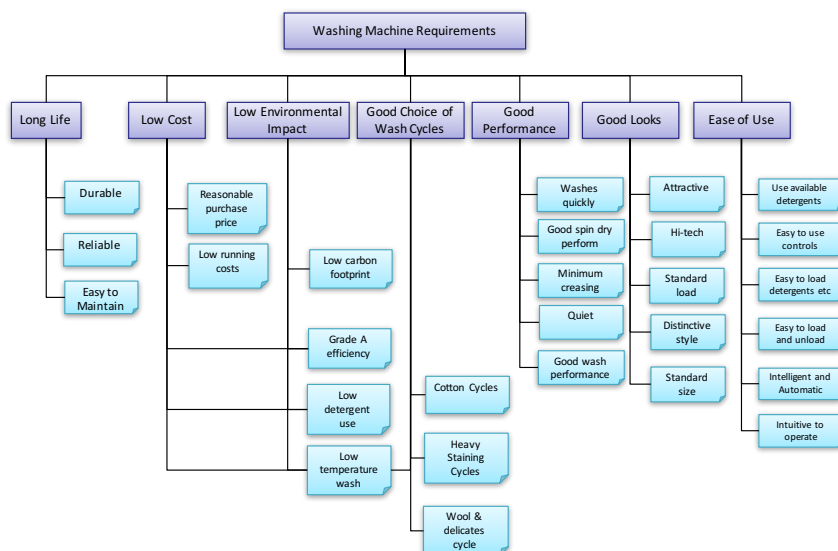
©2016 Waters Corporation

51

Requirements Tree Diagram

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

52

Tree Diagram Summary

- Pros
 - Easy to build and read
 - Promotes discussion
 - Organized output
 - Output intuitive to novice reader
- Cons
 - Encourages reductionist thinking rather than system thinking (there will be relationships beyond the simple hierarchy)

Prioritizing Requirements

- The Tree Diagram provides one approach to organizing requirements
- It can be useful to also consider the relative importance of the requirements
 - This can provide additional information that will be useful later
- Tools for prioritizing requirements include
 - Stakeholder (Customer) Importance Numbers
 - Analytic Hierarchy Process

Stakeholder Importance Numbers

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Each Stakeholder Requirement is assigned an importance rating typically expressed using a scale of 1 to 5 with higher numbers indicating a greater importance to the “customer” (stakeholder)
- It is important that these reflect the values of the stakeholder and NOT internal company beliefs; however, many organizations start from an internal assessment
- There are several ways to determine the Customer Importance numbers.
 - Absolute scales use definitions like:
 - 1 Lowest importance to the customer
 - 2 Moderate importance to the customer
 - 3 Medium importance to the customer
 - 4 Very important to the customer
 - 5 Of highest importance to the customer
 - Absolute scales are perhaps the easiest to apply but do lead to only the upper numbers being used (3 to 5). This is because if the stakeholder has articulated a requirement then it is hardly going to be rated as a 1
 - An alternative is a “Sales Driven” scale

Sales Driven Scale

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- An alternative to the absolute scale is the sales driven scale based on the concept of Order Winners and Order Qualifiers
 - **Order Winners:** those aspects of a system that WIN it orders in the market place. If we are better than our competition we will win on these aspects.
 - **Order Qualifiers:** those aspects of a product that KEEP it in the market place. We only need to be as good as the competition in these aspects. Bettering the competition does not win any further orders. These are sometimes called ‘givens’. However, these can be order-losing sensitive. That is we must be as good as the competition, if we are worse we will not win ANY orders.

The advantage of this scale is that it avoids emotional scoring!

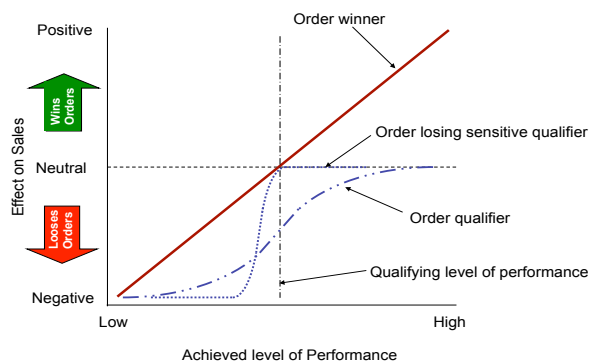
Order Winners and Qualifiers

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Order qualifying criteria are 'givens' that the customer expects to be there at a certain level. Exceeding this level is unlikely to bring in more sales and failure to deliver to this level leads to customer dissatisfaction. On the 1 to 5 scale, such customer requirements are rated as a 1 or 2. Some qualifying criteria can be sensitive in that failure to reach the perceived minimum standard will result in a loss of orders. These

criteria are typically aspects like safety where a minimum safety level is expected (due to legislation for example) and failing to meet that level will halt customer purchases. These are called *order losing sensitive qualifiers*. Such criteria are rated between 3 or 4. Order winners are those aspects in which improvements will result in an increase in sales. These are rated 4 or 5.



©2016 Waters Corporation

57

Analytic Hierarchy Process

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Top Level Criteria							
Is the row item more (+1), equally (1), or less (-1) important than the column item? Fill the lower triangular matrix.							
1 long life							50%
2 low cost	1/5						8%
3 low env impact	1/5	1/5					4%
4 choice of cycles	1/5	1/5	3				6%
5 good wash	1/5	1	1	1			6%
6 looks good	1/5	5	7	5	5		42%
7 ease of use	1/5	3	5	3	3	1/5	20%
Consistency Ratio (should be < 0.1)							
0.09							

In the example Analytic Hierarchy Process matrix above, the requirements for a washing machine are prioritised for importance

What is it?

- The Analytic Hierarchy Process (AHP) provides an effective structure for group decision-making. It can be used to:

- Select between a number of options
- Prioritise requirements/decision criteria

- It helps the group assign numerical values to subjective judgments about a number of criteria and consequently combining the judgments into a single scale for decision-making.

Why do it?

- Prioritising or deciding often concerns a number of criteria whose selection is at the bidding of the decision-makers. These various criteria are likely to be measured on different scales, such as weight and height, or are intangible, as no scales currently exist. Ideally, we require a way of combining these criteria in a meaningful way.
- The Analytic Hierarchy Process provides a structured and disciplined approach to taking these multiple criteria into account when reaching a decision providing a result on a single scale

©2016 Waters Corporation

58

AHP Output for Requirement Prioritisation

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The main output of interest is the priority of each of the requirements on a true ratio scale

Top Level Criteria							
Is the row item more (>1), equally (1), or less (<1) important than the column item? Fill the lower triangular matrix.							
	1 long life	2 low cost	3 low env impact	4 choice of cycles	5 good wash	6 looks good	7 ease of use
1 long life							12%
2 low cost	1/2						8%
3 low env impact	1/4	1/2					4%
4 choice of cycles	1/2	1/2	3				6%
5 good wash	1/2	1	1	1			6%
6 looks good	7	5	7	5	5		42%
7 ease of use	5	3	5	3	3	1/2	23%
Consistency Ratio (should be < 0.1)							
				0.08			

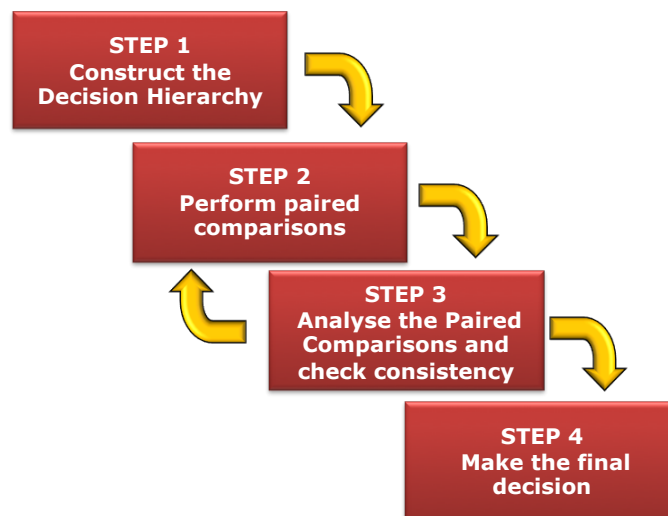
An Importance of 40% for **Safer to use** versus 19% for **Easier to use** means that **Safer to use** is twice as important as **Easier to use**. Rankings do not give you this discrimination

Another key output is a consistency ratio. Have we been consistent in our assessment?

©2016 Waters Corporation

AHP Process

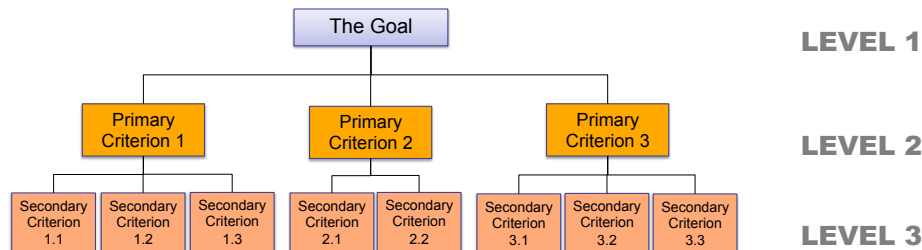
Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

Step 1: Construct Decision Hierarchy

- The Analytic Hierarchy Process requires us to decompose the decision problem into a hierarchy of more easily comprehended sub-problems, each of which can be analyzed independently
- The elements of the hierarchy can relate to any aspect of the decision problem: tangible or intangible, carefully measured or roughly estimated, well, or poorly-understood, indeed anything that applies to the decision at hand



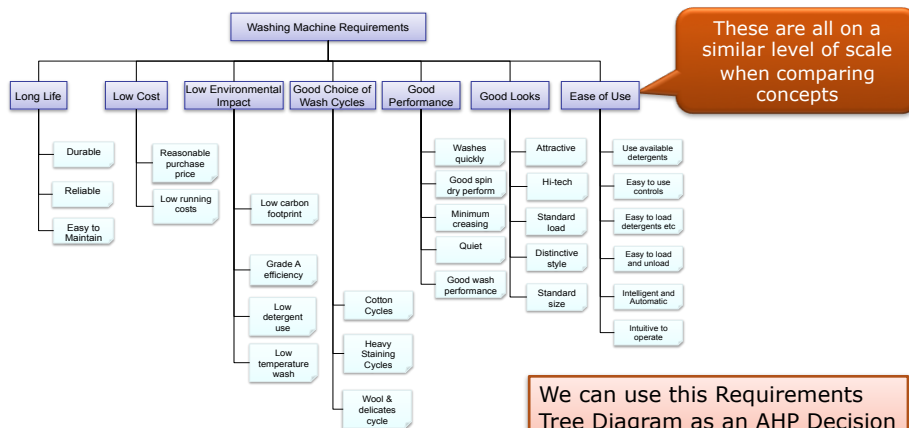
The key is to apply a series of AHPs at each level.

- Level 2 – to prioritize the primary criteria
- Level 3 – a series of AHPs that prioritize each cluster of secondary criteria beneath their respective primary criterion

61

The AHP Decision Hierarchy

- AHP hierarchies are best explained by example. In this case the aim is to prioritise a set of gathered User Requirements by Affinity Diagram and Tree Diagram such that we have the tree



Setting Up the Paired Comparison Matrix

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- To perform the AHP a group of criteria at the same level are subject to a pairwise comparisons systematically using an NxN matrix – where N represents the number of criterion being prioritized.
- For example returning to the washing machine User Requirements the following matrix is constructed from the primary requirements

	Long life	Low Cost	Low Env. Impact	Choice of Cycles	Good Wash	Looks Good	Ease of Use
Long Life							
Low Cost							
Low Env. Impact							
Choice of Cycles							
Good wash							
Looks good							
Ease of Use							

- The leading diagonal is filled with 1s, indicating that first row requirement “Long Life” is equally as important as the first column requirement “Long Life”

©2016 Waters Corporation

63

Setting up - Continued

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The lower left half of the Paired Comparison Matrix is filled with formulae to compute the inverses of their conjugate cells

	Long life	Low Cost	Low Env. Impact	Choice of Cycles	Good Wash	Looks Good	Ease of Use
Long Life	1						
Low Cost		1					
Low Env. Impact			1				
Choice of Cycles				1			
Good wash					1		
Looks good						1	
Ease of Use							1

This cell will contain a number

This cell is its inverse

©2016 Waters Corporation

64

Step 2: Perform Paired Comparisons

- Paired comparisons are then performed on the criteria – order is important - it's always the rows versus the columns
- When performing the paired comparisons the phrasing of the question is VERY important
- If we were comparing the criteria of Long Life and Low Cost we could ask:
 - “How much more important to the User is improved Long Life than improved Low Cost”
- Or we could be more specific if there are improvement targets x and y:
 - “How much more benefit to the User would it be to improve Long Life by x% than to improve Low Cost by y%”

Getting the question right is very important. It is always worthwhile pre-preparing the Comparison Matrix and questions before the comparison session

AHP Paired Comparison Scale

We use a standard scale for making these judgments which accounts for cases when a criterion is less important than another and when it is more important

Intensity of Importance	Definition	Interpretation
1	Equal importance	Two factors contribute equally to the objective
3	Moderately more important	Experience and judgment slightly favour one factor over another
5	Strongly more important	Experience and judgment strongly favour one factor over another
7	Very strongly more important	A factor is favoured very strongly over another; its dominance is demonstrated in practice.
9	Extremely more important	The evidence favouring one factor over another is of the highest possible order of affirmation.
2,4,6,8	For compromise between the above values	Sometimes one needs to interpolate a compromise judgment numerically because there is no good word to describe it.
Reciprocals of above	If factor j has one of the above numbers assigned to it when compared with activity k, then k has the reciprocal value when compared with j.	The inverse meaning to the above; If comparing A with B then a reciprocal value would mean that A was dominated by B; e.g. 1/5 means that A is strongly dominated by B.

Washing Machine Example

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

The question here was "How much more important to the customer is improved Long Life than improved Low Cost"?

The judgment was "it is somewhat more important"

	Long Life	Low Cost	Low Env Impact	Choice of Cycles	Good Wash	Looks Good	Ease of Use
Long Life	1	3	3	3	3	0.143	0.2
Low Cost	0.333	1	3	3	1	0.2	0.333
Low Env Impact	0.333	0.333	1	0.333	1	0.143	0.2
Choice of Cycles	0.333	0.333	3	1	1	0.2	0.333
Good wash	0.333	1	1	1	1	0.2	0.333
Looks good	7	5	7	5	5	1	3
Ease of Use	5	3	5	3	3	0.333	1

With the table populated, we can do the analysis to calculate the importance weightings.

©2016 Waters Corporation

67

Step 3: Analyse the Paired Comparisons and check consistency

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The analysis behind the computation of the relative priorities of each of the criteria is fairly involved, but software packages ease the burden.
- The analysis comprises two basic steps:
 - Calculation of the importance weighting
 - Calculation of the consistency ratio
- The Consistency Ratio is an important metric for assessing the quality of the analysis – i.e. how good our decision is!

©2016 Waters Corporation

68

The Results

The following results were obtained using the software package called Qualica

Criteria	Importance
Long Life	12%
Low Cost	7%
Low Env Impact	4%
Choice of Cycles	6%
Good wash	6%
Looks good	42%
Ease of Use	23%
Consistency Ratio = .08	

This result is saying "Looks Good" is approximately three as important as "Long Life"

This result is saying "Looks Good" is approximately twice as important as "Ease of use"

The Consistency Ratio = 0.08 is below 10% and therefore shows the judgments on the Pairwise table are consistent

©2016 Waters Corporation

69

Guidance on Applying AHP

- The example given to illustrate the process had a small number of criteria (Requirements)
- In practice we could be dealing with a large number of criteria especially if we are using AHP to prioritize requirements we would construct a requirement hierarchy
- AHP works best when the number of requirements at any one level is 7 or less. If you have too many then look to aggregate requirements to reduce the number
 - It is critical that the requirement hierarchy is carefully constructed and that all the requirements at a level in the hierarchy are truly at the same level – don't mix high and low in a single AHP table
 - The Affinity/Tree Diagrams are useful tools to accomplish this

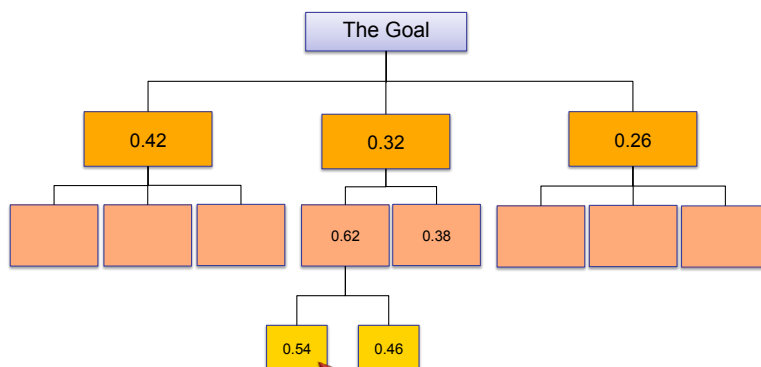
©2016 Waters Corporation

70

Step 4: Make the Final Decision

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- We can multiply the “importances” down through the hierarchy to get lowest level weightings



Overall Weighing of this tertiary requirements is
 $0.32 \times 0.62 \times 0.54 = 0.107 = 10.7\%$

©2016 Waters Corporation

71

Some words of advice

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The Analytic Hierarchy Process gives discipline and structure to our decision making and prioritisation – BUT because we have done some mathematics and have numbers we can be lulled into believing in the precision of the outcome.
 - Use the weightings to help guide subsequent activities
 - Recognise there is still a large degree of subjectivity in the assessments
 - Perform a sensitivity analysis to test the robustness of the final prioritisation

©2016 Waters Corporation

72

Exercise – AHP Pairwise Table

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

	Long life	Low Cost	Low Env Impact	Choice of Cycles	Good Wash	Looks Good	Ease of Use
Long Life	1	3	3	3	3	0.143	0.2
Low Cost	0.333	1	3	3	1	0.2	0.333
Low Env Impact	0.333	0.333	1	0.333	1	0.143	0.2
Choice of Cycles	0.333	0.333	3	1	1	0.2	0.333
Good wash	0.333	1	1	1	1	0.2	0.333
Looks good	7	5	7	5	5	1	3
Ease of Use	5	3	5	3	3	0.333	1

- In teams
 - Use the primary requirements from the User Requirements Affinity (Tree) Diagram and complete the pairwise table
 - At a break the Trainers will perform the analysis

- Time: 20 minutes

Analytic Hierarchy Process Summary

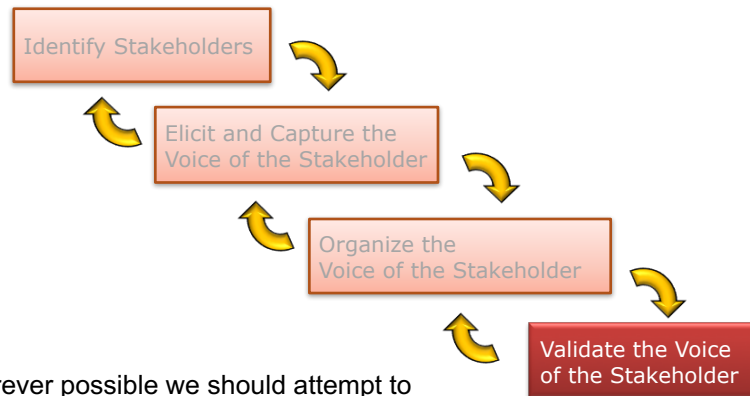
Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Pros
 - If performed in a team can promote
 - Discussion
 - Thinking
 - Structured and disciplined process
 - Prioritises requirements on a true ratio scale
 - Checks consistency
- Cons
 - Deciding who does this – who should be involved
 - Mathematical basis adds unfounded credibility
 - Tedious in many requirements (limit of 7 – 10 per level)

An Iterative Process

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



Wherever possible we should attempt to validate any requirements – (validate = is a particular requirement correct)

©2016 Waters Corporation

75

Validate Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- This can take many forms including:
 - Presenting and walking through the requirements to different groups of representative stakeholders for comment and agreement
 - Providing the stakeholders with copies of our written and structured requirements for comment and agreement
 - Cross checking with other requirements – in particular the system design requirements – this is usually part of analyse requirements

Note: The ultimate validation of requirements happens when the system is delivered – but by then it is too late!

©2016 Waters Corporation

76

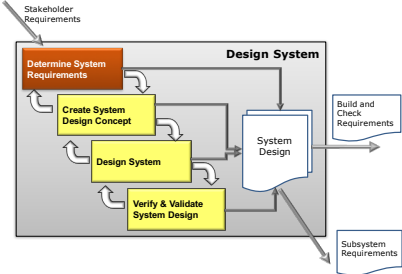
Gather Requirements Summary

- This section has
 - Explained the need for a systematic approach to gathering requirements
 - Presented a process for gathering requirements
 - Introduced some tools and techniques for gathering requirements
 - Requirements Elicitation Plan
 - Stakeholder Influence Map
 - Affinity Diagram
 - Use Case
 - (Requirements) Tree Diagram
 - Analytic Hierarchy Process





THE SCIENCE OF WHAT'S POSSIBLE.®



Determine System Requirements

Analyse Requirements

This Session:

- Explains the need to analyse gathered stakeholder requirements
- Presents a process for analysing requirements
- Provides an introduction to some of the tools for analysing requirements

©2016 Waters Corporation

1





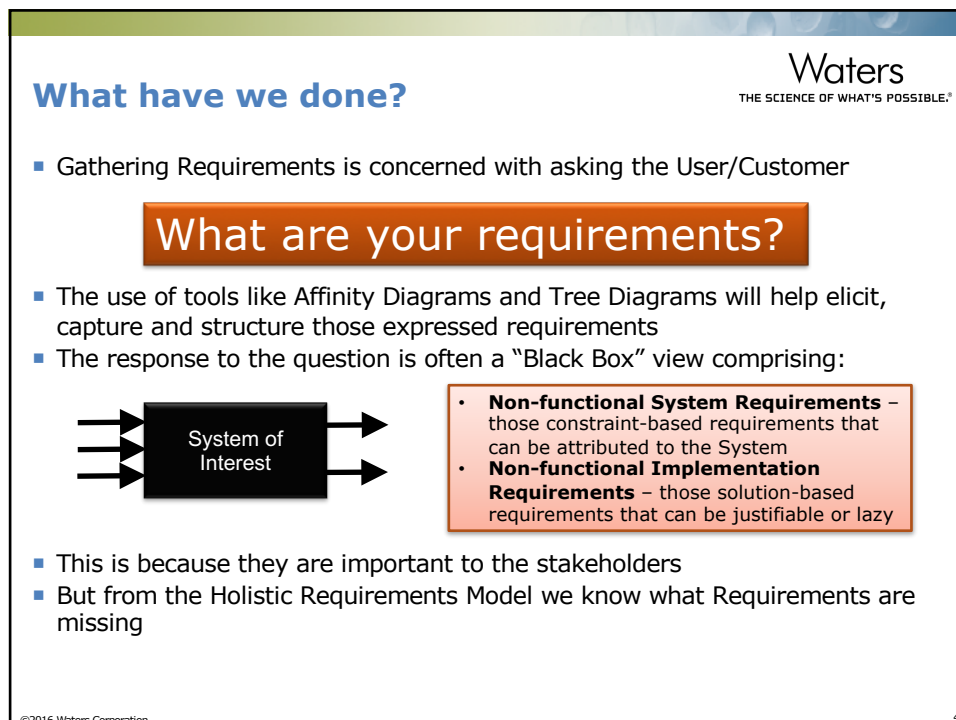
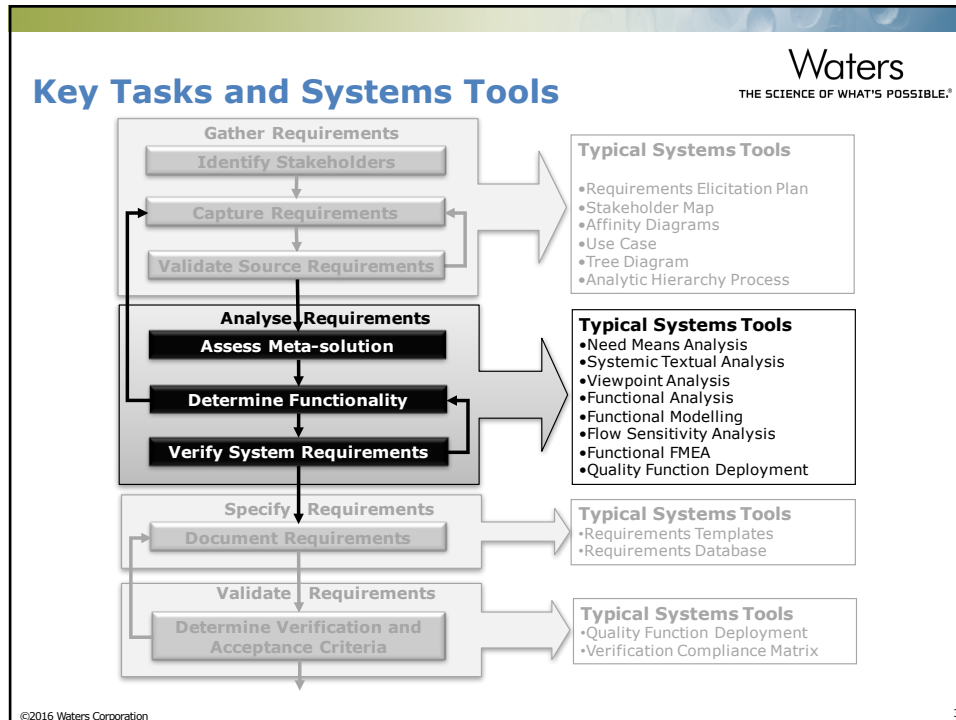
THE SCIENCE OF WHAT'S POSSIBLE.®

Why Analyse Requirements

- Despite every effort in gathering requirements, stakeholder requirements will be:
 - Ambiguous
 - Rambling
 - Incomplete
 - Inconsistent
 - Incorrect
- The purpose of Analysing Requirements is to translate the Expressed (spoken) stakeholder requirements into a set of System Design Requirements that are:
 - Clear (Unambiguous)
 - Concise
 - Complete
 - Consistent
 - Correct
 - and ultimately Confirmable (Measurable)

©2016 Waters Corporation

2

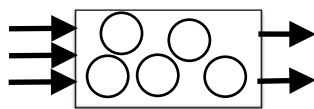


What Are We Going to do?

- To “get at” the missing requirements we will have to analyse what has been expressed. The question we ask is

What are the necessary requirements?

- This is a “White Box” view of the system concerned with determining what is needed to deliver the “Black Box” performance



- **Operational Requirement**
- **Functional Requirements**
- **Non-functional System Requirements**
- **Non-functional Implementation Requirements**
- **Non-functional Performance Requirements**

- The White Box view will focus on uncovering the functionality (Functional Requirements) necessary to deliver the Operational Requirement of the System of Interest

The Rationale

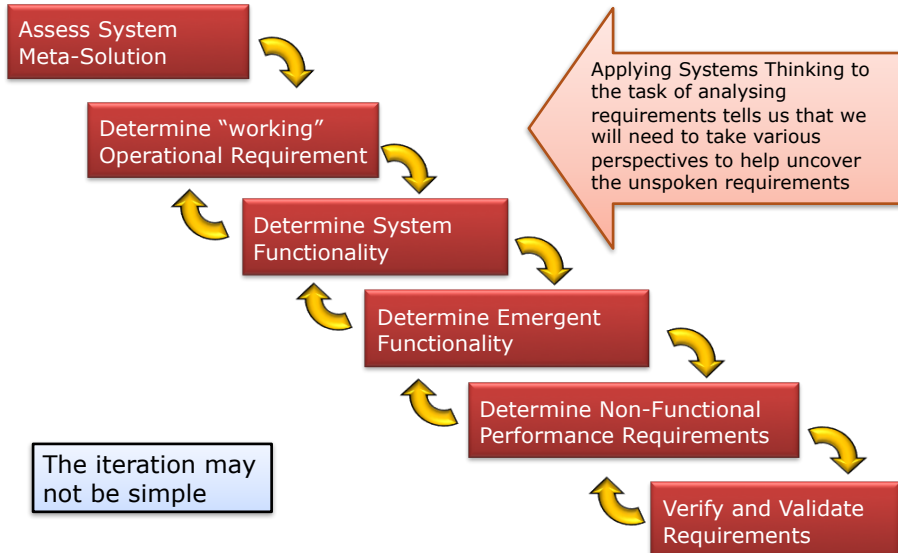
- Given a problem statement in terms of an Operational Requirement we are going to use the laws of physics and logic to deduce the necessary Functionality needed to deliver it
- From that collection of Functions we can ask two important questions:
 - How well does a particular Function have to perform?
 - What can go wrong?

- To “put a Man of the moon” the laws of physics and logic tells me that I WILL have to:
 - Escape Earth’s Gravity
 - Accommodate Crew
 - Navigate to Moon
 - Land on Moon
 - Explore Moon
 - Escape Moon’s Gravity
 - Navigate to Earth
 - Land on Earth
 - and many more

Analyse Requirements: An Iterative Process

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

7

And

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

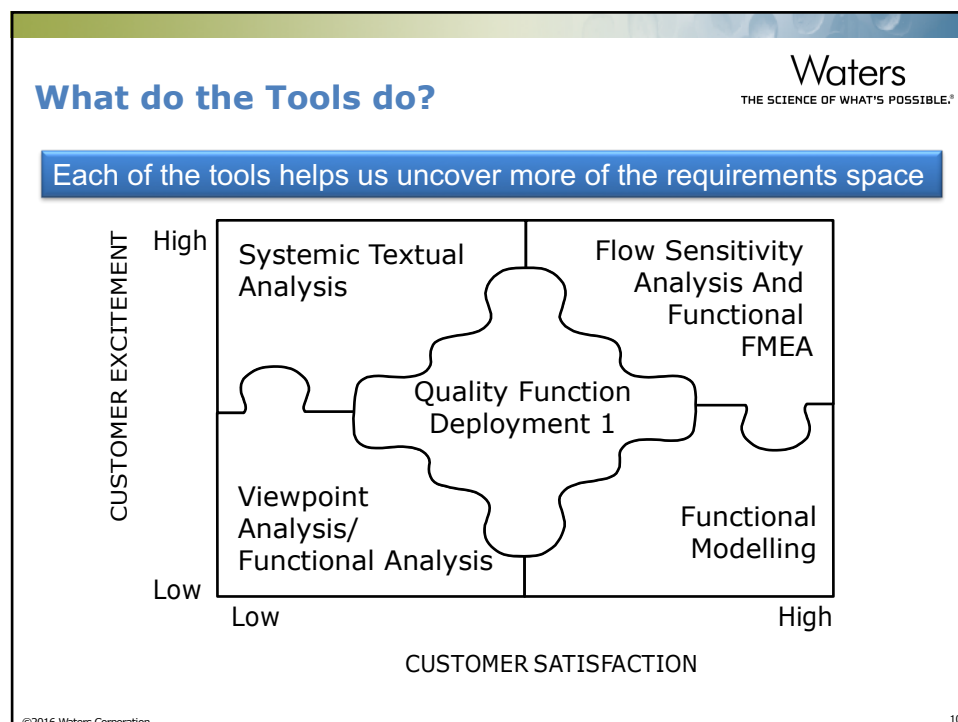
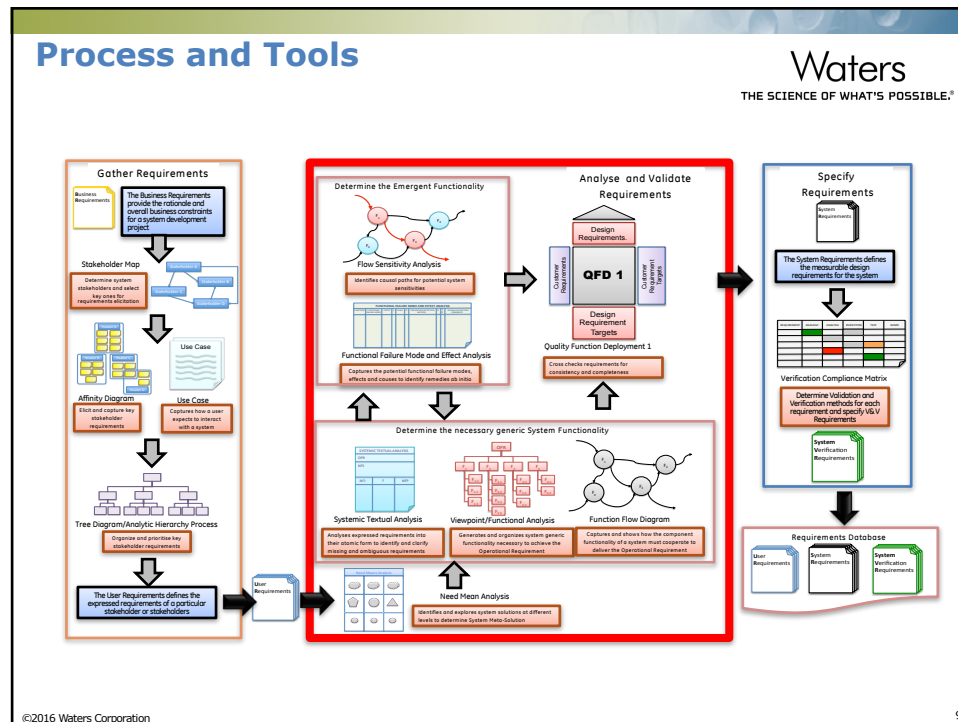
- We will also need to reconcile inconsistency
- To address ambiguity we first need to spot it and then resolve it

There is no single tool that will accomplish these tasks. Instead we use a System of Tools each contributing to the overall objective

The Holistic Requirements Model provides the reference structure

©2016 Waters Corporation

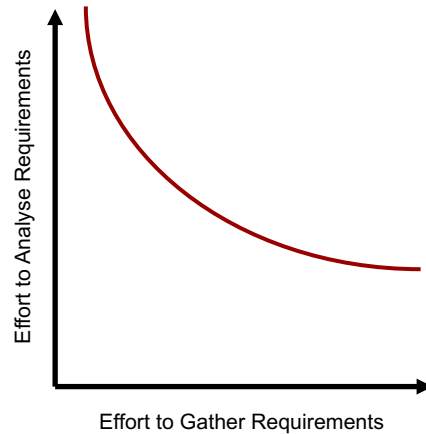
8



Gathering vs. Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- We have both to gather and analyse requirements
- The more effort we put into gathering requirements the easier (faster and cheaper) the analysis becomes
- There is a trade-off however
- Tools like Quality Function Deployment can help in measuring the maturity of the requirements set
- It is worth recording the effort expended in each activity to provide a reference for future projects

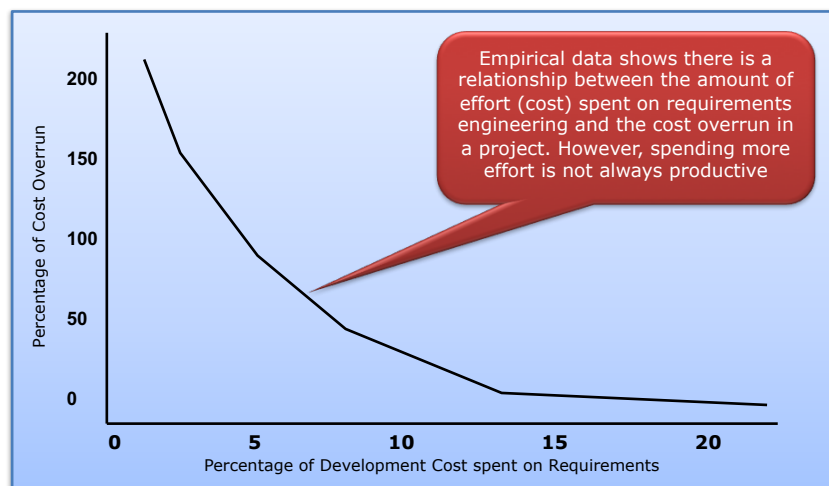
©2016 Waters Corporation

11

How Much Effort?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Adapted from: Hooks, Managing Requirements, 1994 - Figure 1
and Young, *Effective Requirements Practices*, Addison-Wesley, 2001 - Figure 4-1

©2016 Waters Corporation

12

This section

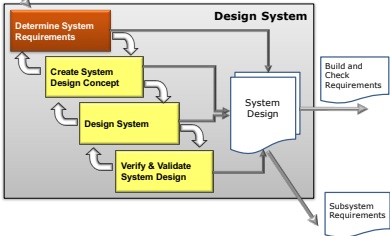
Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Explained the need to analyse gathered stakeholder requirements
- Presented a process for analysing requirements
- Provided an introduction to some tools for analysing requirements
 - Systemic Textual Analysis
 - Viewpoint Analysis
 - Functional Analysis
 - Functional Modelling
 - Sensitivity and Failure Analysis
 - Quality Function Deployment





THE SCIENCE OF WHAT'S POSSIBLE.®



Analyse Requirements Assess the System Meta-solution using Need Means Analysis

This Session:

- Highlights the need to explore potential system Meta-solutions early to scope system functionality
- Describes Need Means Analysis as a simple practical tool for quickly assessing the potential system Meta-solutions

©2016 Waters Corporation

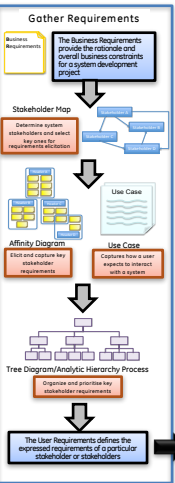
1

Process and Tools

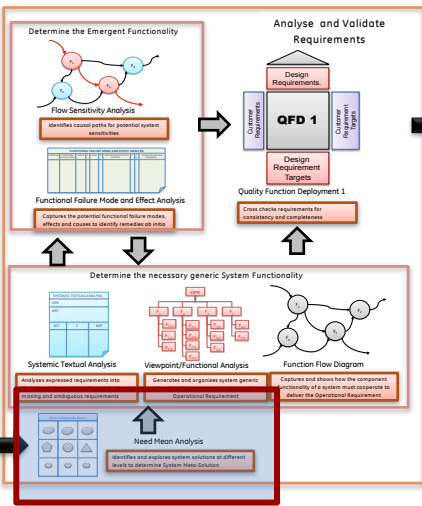


THE SCIENCE OF WHAT'S POSSIBLE.®

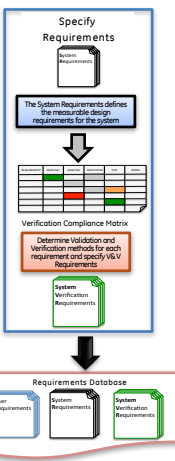
Gather Requirements



Analyse and Validate Requirements



Specify Requirements



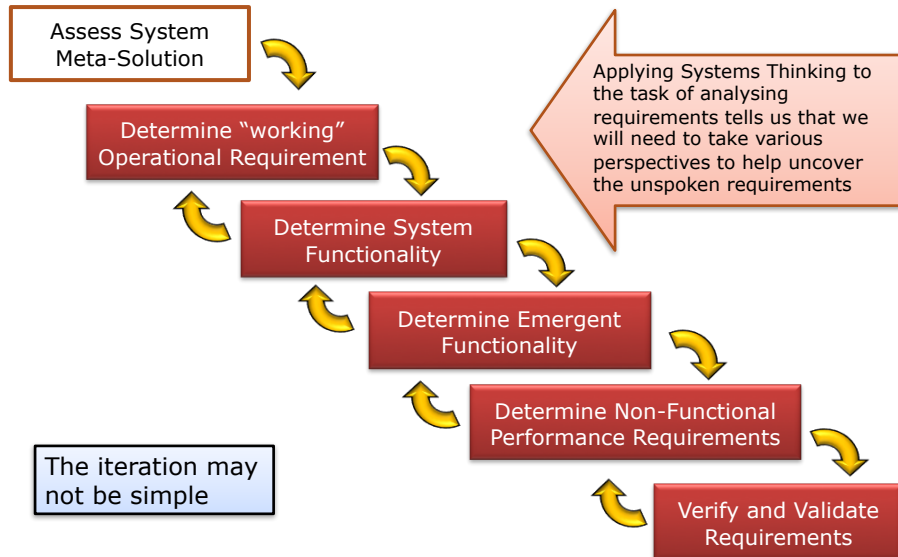
©2016 Waters Corporation

2

Where are we?: Analyse Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



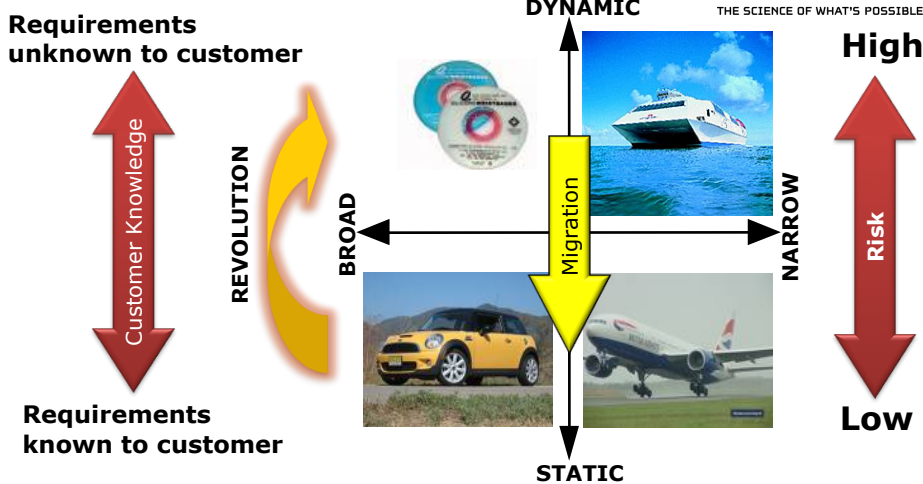
©2016 Waters Corporation

3

Remember: Knowledge & Risk

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Conclusion: Customers and Providers often work in the static design domain because of knowledge and risk. Revolutionary improvement happens when a new dynamic concept is used. Making this decision is difficult and risky

©2016 Waters Corporation

4

System Meta-solution

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Operational
Requirement

To transport passengers and their
baggage from one point to
another efficiently and safely

Company	Meta-solution
Toyota	
Airbus	
Bombardier	
Yamaha	

A Meta-solution is an organization's current
preferred solution to a system "problem"

It is often possible to identify and decide upon the Meta-solution very
early in System Development. This will allow us to bound or scope
the system functionality relative to that chosen Meta-solution - BUT

©2016 Waters Corporation

5

Failure to Explore New Meta- solutions - A Sorry Tale

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Up to the late 1970s the carburettor was the dominant method of mixing air and petrol. Because of its cost and reliability, fuel injection was rare.
- The carburettor manufacturers considered their business safe and focused on making better carburettors! They should have been investigating the engine manufacturer's need for "providing the engine with an air-fuel mix"
- Other people started working on cheap and effective fuel injection systems and before the carburettor manufacturers reacted their business had gone – there are no carburettor manufacturers left in the domestic automotive market!

©2016 Waters Corporation

6

Successful Dynamic Design Concept

Stena HSS – the largest fast ferry in the world carrying up to 1500 passengers, the Stena HSS travels at speeds of up to 46 miles per hour, cutting journey times in half – it is a revolution in ferry design



©2016 Waters Corporation

7

Unsuccessful Dynamic Design Concept

The Sinclair C5 – A potential revolution in personal transport – A dynamic concept that failed dramatically.



©2016 Waters Corporation

8

System Meta-solution

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Every “system of interest” is a subsystem of a bigger system. Examining the system of interest from the viewpoint of the super-system to which it belongs forces us to consider what it does and the NEED it satisfies.
- Remember, customers do not buy systems, they buy capability, they have a need and if somebody was to come along with a “better” alternative meta-solution that delivers their need they will buy it!
- When considering the design of a new system, subsystem, or component, we should always review the current Meta-solution to see if it is viable within the project timescales. There will always be alternatives and some of these may be more acceptable to the customer/stakeholders
 - Indeed, a sensible organization should always be researching alternative meta-solutions in preparation for a change - if necessary

©2016 Waters Corporation

9

Need Means Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Cleaning Service	Disposable Clothes/ Items	Holographic Clothing
Twin Tub	Washing Machine OR: Wash Items Need:	Combined Dishwasher
Ultrasonic Cleaning	Fuzzy Logic Control	Auto Powder Dispensing

The Need Means Analysis output above is for a domestic washing machine that explores potential solutions at different levels to agree the system Meta-solution

- What is it?
 - A lateral thinking approach aimed at getting people away from conventional solutions to explore and select the Meta-solution for the system of interest
 - The approach is based around identifying the “need” that a system satisfies and then using this to investigate alternative solutions
- How to do it?

Step 1: Identify the Operational Requirement of the current Meta-solution to the system of interest

Step 2: Derive the NEED that this system satisfies

Step 3: Use “the box of nine” to identify new solutions to the need at different levels

Step 4: Review “the box of nine” to decide on the Meta-solution for the system of interest

©2016 Waters Corporation

10

Step 1: Identify the Operational Requirement of the current Meta-solution to the system of interest

- The starting point for a Need Means Analysis is to identify the current Meta-solution and determine its Operational Requirement
- If Systems Engineering has been used in its development, the Operational Requirement should exist – otherwise one will have to be created
- Remember

An **Operational Requirement** defines the major purpose of a system (i.e. what it fundamentally does - its capability) together with the key overarching constraints (that define the context)

- Example: Domestic Automatic Washing Machine



To clean effectively domestic fabric items using minimum water and power and currently available detergents

Step 2: Derive the Need that this Systems Satisfies

- An Operational Requirement defines what a system fundamentally does
- There is always a higher level reason why that purpose is being carried out – this is the NEED
- It is likely not to be the same as the Operational Requirement – but will be related
- For example the operational Requirement for a pen is to “lay-down a trail of ink”
- But why do we NEED to lay-down a trail of ink” – because we wish to communicate our thoughts

- Example: Domestic Automatic Washing Machine

Why do we want

to clean effectively domestic fabric items?



Hygiene

Step 3: Use “the box of nine” to identify new solutions to the need at different levels

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

How can the Need be achieved without the User/Customer having to purchase/own a Washing Machine

➔

Cleaning Service	Disposable Clothes/ Items	Holographic Clothing	New system	
Twin Tub	Washing Machine OR: Clean domestic fabric Items Need: Hygiene	Combined Dishwasher		New system architecture
Ultrasonic Cleaning	Fuzzy Logic Control	Auto Powder Dispensing		

The User/Customer still purchases/owns a Washing Machine, but it's a different Architecture

➔

The same Meta-solution but changes made at the subsystem level

➔

The “box of nine” encourages us to think of at least 3 options per level providing sufficient to explore the choices. Can include further options at any level

©2016 Waters Corporation 13

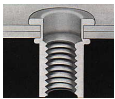
Need Means Applies at any level!

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Higher Level

Same Level

Lower Level

Machine from solid	Adhesive	Weld	Can include further options New system New system architecture New subsystem
 Rivet Nut	Rivet OR: Join 2 or more components together Need: Structural integrity	 Pop Rivet	
 Dome Head	 Countersunk	 Flat Head	

©2016 Waters Corporation 14

Need Means Applies To Any System!

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Can include further options

Higher Level

User Repair

Postal Replacement

Redundant System

New system

Same Level

Reliability Centred Maintenance System

Man in a Van to repair system at user's home OR: Fix System Need: Working System

Prognostic based system

New system architecture

Lower Level

Computer – internet based technical publications

Electronic/ software based diagnostics

Internet based booking service

New subsystem

©2016 Waters Corporation

15

Exercise – Need Means Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

NEEDS MEANS ANALYSIS		
System: Domestic Central Heating System	Author: A. Warm	Date: 1 July 2016
Ideas at a higher level: new meta solution	Ideas at a higher level: new meta solution	Ideas at a higher level: new meta solution
Centrally generated piped hot water system	Underground house with ground-source or air-source heat pump	Additional clothing Improved house insulation – no windows but flat-screen views
Ideas at the same level: new system architecture	Existing meta solution: Domestic Central Heating System Op R: To provide space heating and hot water Need: to be comfortable	Ideas at the same level: new system architecture
Intelligent Domestic Heating System		Coal/Wood Fires distributed through house
Ideas at a lower level: new sub-system	Ideas at a lower level: new sub-system	Ideas at a lower level: new sub-system
Electric Storage Heaters House battery	Fuzzy Logic Thermostat Individual Room Thermostats	Non—water working fluid

In small Teams

- Perform a Need Means Analysis for the System Described by the Trainer
 - Be prepared to share your outcome with the rest of the class
- Time 15 minutes

©2016 Waters Corporation

16

Step 4: Review “the box of nine” to decide on the Meta- solution for the system of interest

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The purpose of a Need Means Analysis is to reflect upon the alternative levels of solutions available - with the intention of agreeing the system Meta-solution
- It is always possible to operate at higher and higher levels of abstraction in order to retain solution choices – BUT often the problem context allows the meta-solution to be decided at this point thereby focusing and scoping the design activity
- By agreeing the Meta-solution we are bounding our system functionality
- Example
- In 1998-1999 a well known washing machine designer/manufacture undertook a technology demonstration programme to develop an “intelligent washing system”.
- One possible Meta-solution was a “dry-cleaning machine”. However, this particular company had no design or manufacturing experience with “dry-cleaning” which, combined with the health and safety issues, led them to retaining the same basic Meta-solution – a water-based electrically powered system

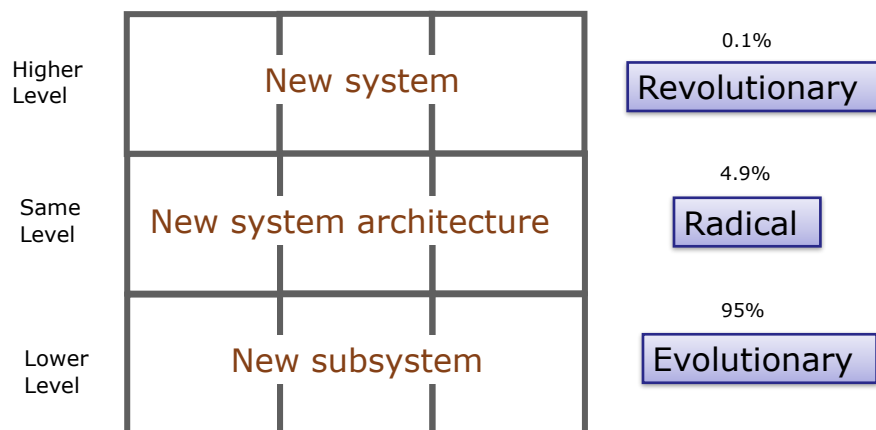
©2016 Waters Corporation

17

Meta-solutions, Level of Innovation and Risk

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

18

Radical Innovation

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- What are the issues with adopting any of your “same” level solutions? i.e. a change of architecture.
 - e.g. what would a move from a hi-bypass turbofan to an open-rotor architecture mean for the airframer?
- Radical change (change of architecture) at the subsystem level often requires a change in architecture at the system level
- Evidence suggests this won't happen unless there is a performance improvement >40%!

©2016 Waters Corporation

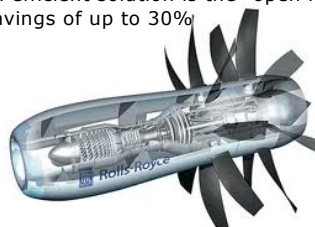
19

Example

Commercial aircraft typically use hi-bypass turbo fans to generate thrust



A far more fuel efficient solution is the “open rotor” offering fuel savings of up to 30%



BUT

Such a power plant will require the airframer to change their architecture



©2016 Waters Corporation

20

Need Means Analysis Summary

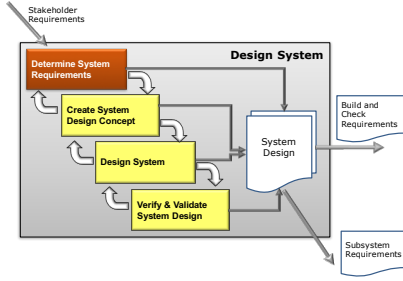
- Pros
 - Simple - anybody can contribute
 - If performed in a team it promotes
 - Discussion
 - Thinking
 - Identifies alternative solutions at different levels
 - Can lead to an agreed meta-solution and therefore scope the problem space
- Cons
 - Deciding who does this? Is it an engineering or marketing tool
 - Can give a false feeling of security
 - Can give a solution that allows an organization to jump straight into detailed design without doing the full system design



BURGE
HUGHES
WALSH



Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



Analyse Requirements

Systemic Textual Analysis

This Session:

- Highlights the issues of “traditional” Textual Analysis
- Introduces and describes Systemic Textual Analysis as a powerful alternative to Traditional Textual Analysis
- Provides an opportunity to practise using Systemic Textual Analysis

©2016 Waters Corporation

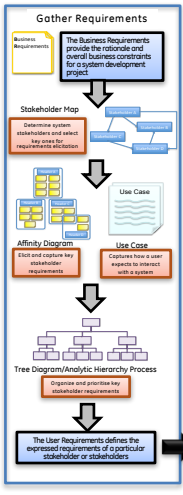
1

Process and Tools

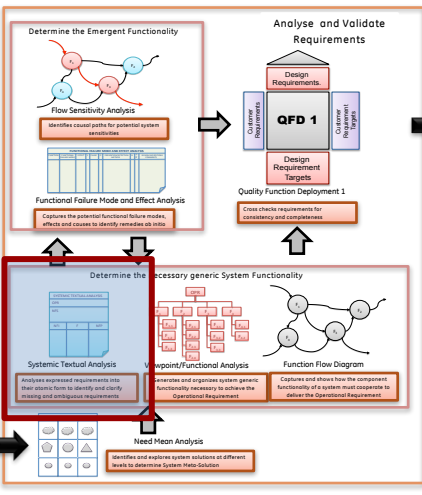


Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

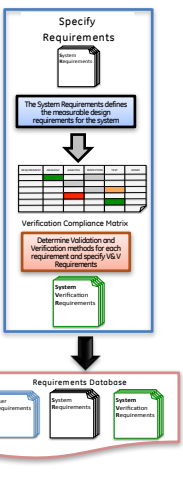
Gather Requirements



Analyse and Validate Requirements



Specify Requirements



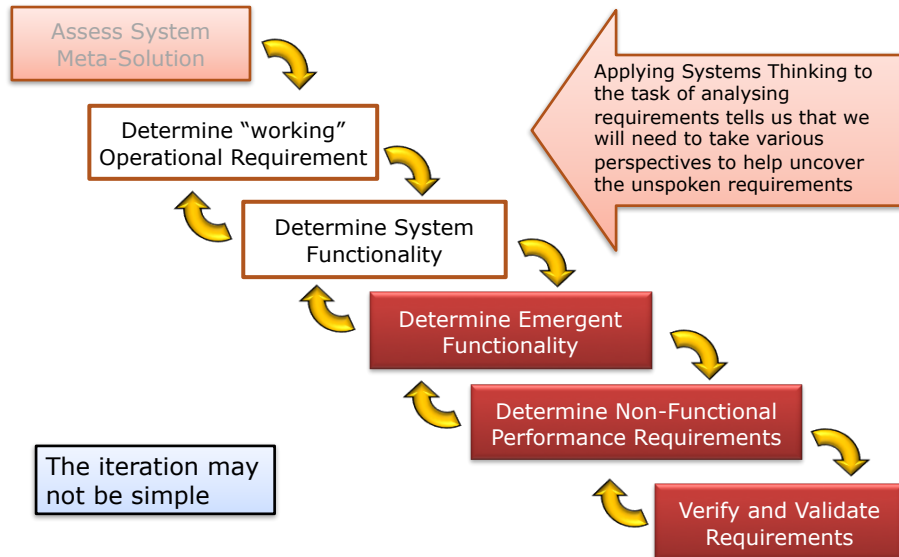
©2016 Waters Corporation

2

Where are we?: Analyse Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

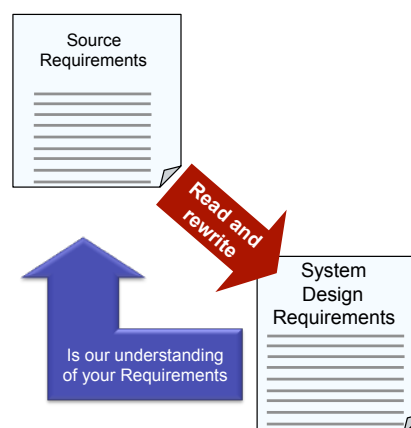
3

Traditional Requirements Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Source (customer/stakeholder) Requirements are typically written documents and are passed to engineers to analyse and interpret in terms of a set of System Design Requirements
- The traditional method of analysis and interpretation is through conducting Textual Analysis
- Textual Analysis is a line by line analysis of the written requirements to decompose, clarify and expand as appropriate. Ultimately, leading to a set of written System Design Requirements
- While this approach has undoubtedly been successful, there is growing evidence that it is failing!



©2016 Waters Corporation

4

Failings of Traditional Textual Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- One of the prime reasons Traditional Textual Analysis is failing is the increasing complexity implicit and explicit in the requirements
- Indeed, it is no longer possible for a single person to understand the “whole” of the requirements – we all too quickly reach a “brain full”
- It's the old maxim “How do you eat an Elephant? - one bite at a time!”
 - The Source Requirements have to be broken down into manageable “chunks” in some way so that individuals can address their “chunk”
 - It is, of course, assumed that the individuals will interpret their “chunk” in a way consistent with all the other individuals – i.e. if we gave the same “chunk” to two individuals their interpretation would be the same?
- In recent times
 - The requirements document to define the replacement for the Spitfire and Hurricane contained some 46 requirements in 3 ½ pages and led to the Gloster Meteor the RAF's first operational jet aircraft
 - The requirements document for the Astute class submarine contained 16,000 requirements in some 2000 pages

©2016 Waters Corporation

5

Exercise – Perceptions

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Read the short story
- Individually determine whether the statements are:
 - True
 - False
 - Don't know
- Feedback how many you found to be True, False and Unknown

©2016 Waters Corporation

6

Perceptions Introduce Bias

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- We all have different perceptions of the same thing! When faced with ambiguity our experience, our “baggage”, causes us to make assumptions that may differ significantly from others – we make a different interpretation.
- When we analyse a set of Source Requirements, how do we resolve the fact that we will have to deal with the ambiguity that will introduce bias into the analysis?
- Conclusion:
 - We need a way of parsing¹ the individual source requirements, such that different individuals will interpret the requirements in a consistent way

¹ Parsing is a common term used in describing language comprehension. In the current context, parsing refers to the way that human beings analyse a sentence or phrase “in terms of grammatical constituents, identifying the parts of speech, syntactic relations, etc. that lead to a complete understanding”.

Systemic Textual Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Systemic Textual Analysis		
Project: Intelligent Washing Machine		Date: 1/6/2015
Author: 1 M Clean		Issue: Draft
Requirements	Comments	
Context: 1.1 Studies show that there are an increasing number of single persons with relatively high disposable incomes. Moreover, the current generation of “pc-enabled” high-income professionals will be replaced by a larger generation of “internet-enabled” higher income professionals. Their lifestyle and priorities are such that they wish to minimise domestic chores by making maximum use of automation and machine intelligence. An opportunity therefore exists for an intelligent washing system that is capable of automating much of the current manual functionality associated with washing domestic items. 1.2 The market studies indicate that there is a potential total market of 1,000,000 machines per annum for this sector (our share is estimated at 20% - 30%) and current spend analysis indicates a selling price in the region of £600 - £750.	It is often useful to provide any contextual information such as market information It is good practice to provide traceability back to the original source requirements	
Operational Requirement: To automatically clean clothes with minimal user interaction, efficiently and without damage Non-functional System Requirements: standard size (595x580x850) easy to use average useful life of the machine is to be 7 years first year failure rate is to be less than 10% energy efficiency should be grade A systems must conform to local safety standards. Attractive and distinctive Intelligent	There is no clear operational requirement in the document. In such cases the team should “have a go” at generating a suitable operational requirement Many of the requirements in the original document are compound of several requirements. These have been separated out and each corresponding atomic requirements	
Non-functional Implementation Requirement	Functional Requirement	Non-functional Performance Requirement
	take a standard 5kg load	
	determine the load make up and fabric characteristics	mixed loads
	determine and the best cleaning cycle	operate at appropriate temperatures using

The example above shows a partially complete Systemic Analysis for an Intelligent Washing Machine

What is it?

- A line-by-line analysis (parsing) of the source requirements using the Holistic Requirements Model as a framework for organizing the analysis

Why Do it?

- To interpret, expand and clarify the individual expressed requirements
- To identify missing requirements
- To deduce requirements

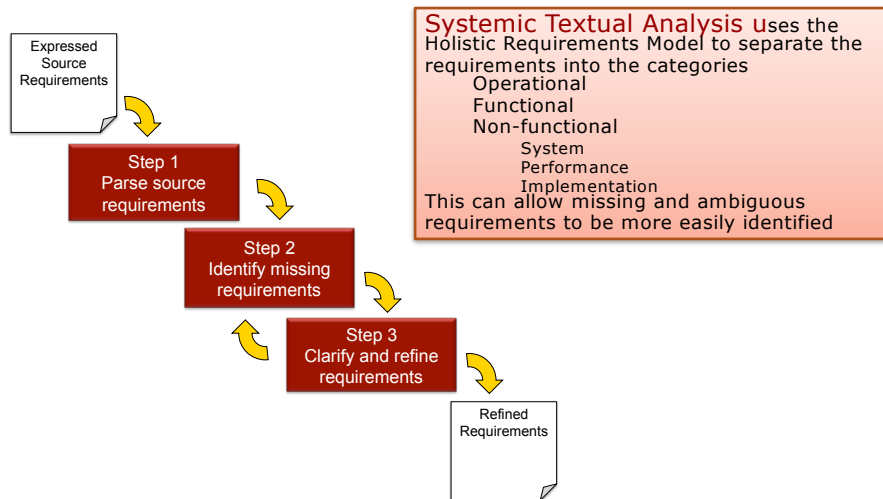
When to do it?

- Whenever a set of expressed Source Requirements are available

How to do it?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



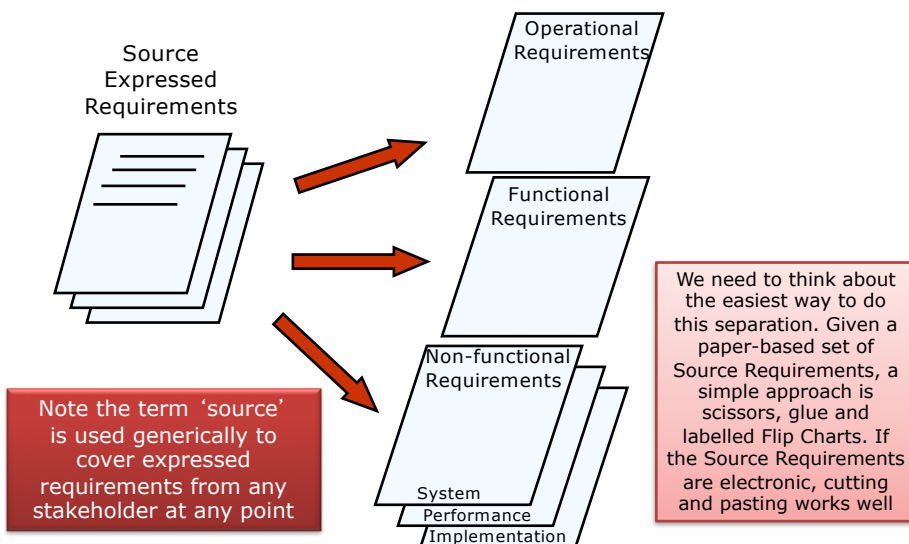
©2016 Waters Corporation

9

Step : Separate and Sort Source Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

10

The Need to Analyse Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Requirements cause problems for two key reasons:
 - In general, people find it difficult to write requirements – usually because they have received no guidance
 - Equally, people find it difficult to analyse and understand requirements – usually because they have received no guidance
- We do get better with practice, but a few simple tips can help in both areas

©2016 Waters Corporation

11

Parsing (Untangling) Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

TIP #3
Look for the “be” segments - they often specify Non-functional System requirements

TIP #1
Look for any VERBS - they may specify or imply Functional Requirements

The mower must be safe and needs to navigate around a lawn avoiding permanent and moving obstacles. An electronic memory facility to record the shape and position of fixed obstacles might be useful.

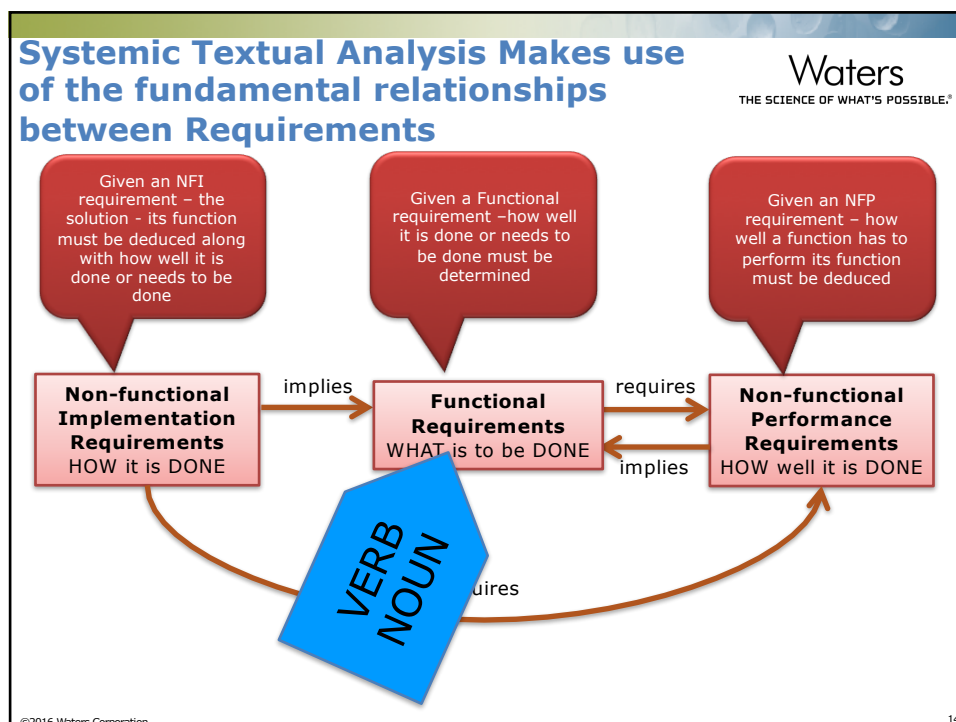
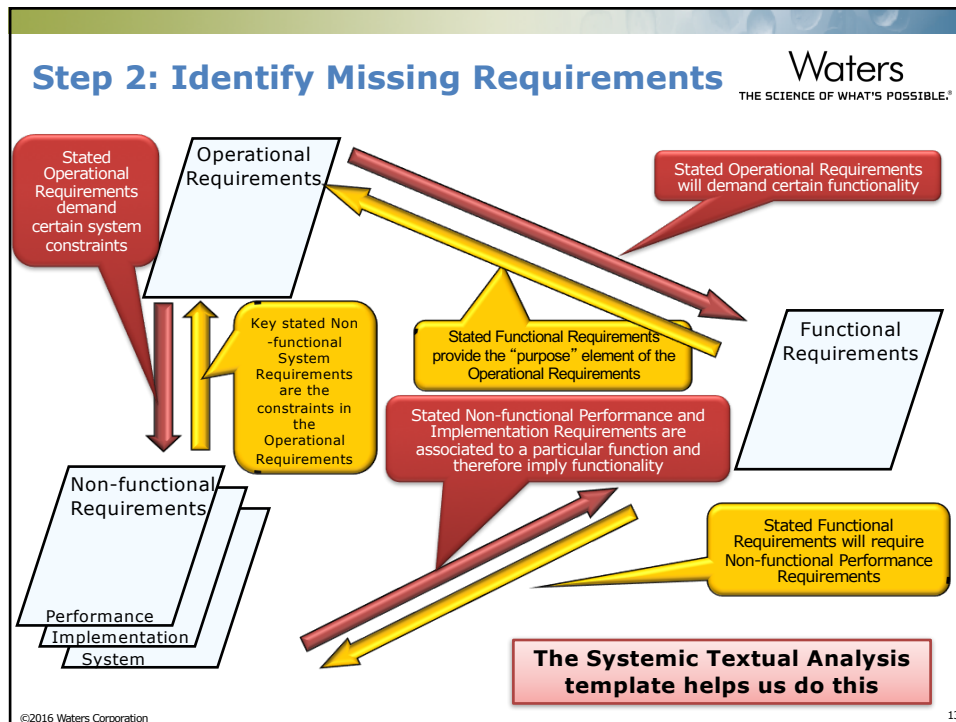
This is a typical “customer” requirement. At first glance okay – but it actually comprises several requirements of different categories

TIP #2
Look for conditions after the verbs - they may specify or imply Non-functional Requirements

TIP #4
Look for “things” or objects - they may specify or imply Non-functional Implementation Requirements

©2016 Waters Corporation

12



Systemic Textual Analysis Template

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Systemic Textual Analysis			
Project:		Date:	
Author:		Issue:	
Requirements		Comment:	
Context:			
Operational Requirement:			
Non-functional System Requirements:			
Non-functional Implementation Requirement (How it is done)	Functional Requirement (What has to be done)	Non-functional Performance Requirement (How well it is done)	

Source Requirements Documents often contain contextual information - this is captured here

Operational Requirement goes here – if there is one!

Non-functional System Requirements go here. Note that these constrain what is below

Non-functional Implementation, Functional and Non-functional Performance Requirements go here. The order helps to identify missing requirements

©2016 Waters Corporation 15

Operational Requirement

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- We often find that the set of source requirements does not contain a clear Operational Requirement – in such cases we need to develop one – it's likely not to be the finished definition, that will emerge as the analysis proceeds, and is called the "working" Operational Requirement
- A good approach here is to get all teams members to draft there own – remember the 20 word target (but do not get hung-up on it – 25 words is okay, as might be 13!)
- For example:

To toast bread products and other toastable items safely effectively and efficiently with the minimum of human intervention

It may be necessary to define what is meant by:

- Intelligently
- Effectively
- Efficiently

This is the "Working OR" – a starting point! It actually is not a particularly "good" requirement because in its present form is not measurable! – BUT trying to be "perfect" at this point can be nugatory work

©2016 Waters Corporation 16

Template Layout helps identify Missing Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

The Systemic Textual Analysis Template makes use of the relationships between the Functional Requirements (what is to be done), Non-functional Performance Requirements (How well it is done) and the Non-functional Implementation Requirements (How it is done)

Non-functional Implementation Requirement	Functional Requirement	Non-functional Performance Requirement
		Capacity 2 to 4 slices
		Different sizes of bread
	Apply Heat to sides of bread	
	Monitor the browning	
	Control browning	
Standard Single phase 230V 50 Hz ac		
Crumb Tray		
		Cool to the touch

"Gap" Missing Requirements

"Gap" Missing Requirement

©2016 Waters Corporation

17

Exercise – Systemic Textual Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Systemic Textual Analysis		
Project:	Date:	
Author:	Issue:	
Requirements	Comments	
Context:		
Operational Requirement:		
"to provide hot water and controllable space heating in a multi-room dwelling with optimum energy usage"		
Non functional System Requirements:		
Intelligent		
Easy to install		
Easy to operate		
Require minimum maintenance		
System Reliability less than 1 failure in ten years		
System life of 15 years		
Installation cost ≤ comparable gas-fired system		
Running cost ≤ comparable gas-fired system		
Non-Functional Implementation Requirement	Functional Requirement	Non Functional Performance Requirement
Electrically based		
Off-peak electricity	Optimise electricity usage	
	Calculate Solar Heating	
	Calculate Thermal losses	Due to: wind outside air temp
	Predict Temperature Profile	Each Room
	Deliver Heat Content	Exact amount
	Set Room Temperature	Correct time individual rooms ±0.5°C

- In small teams:
 - Consider the Source Requirements Document given to you by the Tutor
 - Perform Steps 1 and 2 of a Systemic Textual Analysis
 - Step 1: Parse the individual written requirements to identify the atomic requirements and categorise as
 - the operational requirements
 - the functional requirements
 - the non functional requirements
 - System
 - Performance
 - Implementation
 - Tip: use sticky notes and annotate with the original requirements number to maintain traceability
 - Step 2: identify any missing requirements
 - Record the outcomes on the Systemic Textual Analysis form provided and be prepared to share with the rest of the group
- Time 45 minutes

©2016 Waters Corporation

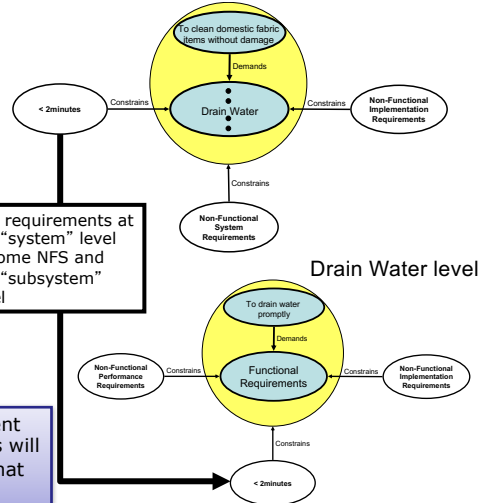
18

Issues: What's My System?

- When performing a Systemic Textual Analysis it is essential we are clear on what is the system because the allocation of requirements to the NFS or NFP categories is dependent on this
- Remember one person's system is another person's subsystem!
- That is, NFS requirements at the system level are NFS requirements at the sub-system level and so on

If you are uncertain as to whether a requirement is NFS or NFP – start by assuming it is NFP this will force you to ask the question: if it is an NFP what is the associated “function”

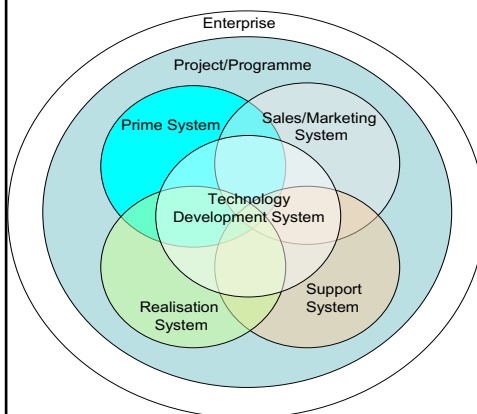
Washing M/C level



©2016 Waters Corporation

19

Issues: The Wrong Requirements



- Source Requirements documents contain requirements for the systems associated with the Prime System
- One of the most common situations is to include requirements for the project/programme
- This is to be expected - but it does mean we have to sort the raw requirements and align them to their respective systems
- This may require us to start a number of Systemic Textual Analyses - one for each system!

©2016 Waters Corporation

20

Missing Requirements → Deduce Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The purpose of the Systemic Textual Analysis is not just to parse requirements and find where requirements are missing but to deduce those requirements
- All the “gaps” in the STA template should be discussed to deduce suitable requirement(s)
- At this stage this may lead to re-categorising the source requirements. For example, a requirement initially considered to be Non-functional Performance, for which no single function can be deduced or where several emerge should be re-categorised as Non-functional System
- It is important to record, in some way, which requirements are deduced – hence the comments column on the template
- There are three possible situations where the expressed requirement is a:
 - Non-functional Implementation
 - Non-functional Performance
 - Functional

©2016 Waters Corporation

21

Deducing Requirements from Non-functional Implementation Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Non-functional Implementation Requirements specify how part of the system is to be built – they are solution-based requirements

Non-functional Implementation requirements should be tested as to whether they are justifiable or lazy

Non-functional Implementation Requirement	Functional Requirement	Non-functional Performance Requirement
Standard Single phase 230V 50 Hz ac		
Crumb Tray		

Where there are gaps for lazy NFI requirements we need to determine the function that the “solution” performs

Where there are gaps for justifiable NFI requirements we should also determine the function and later (when undertaking system design) explore alternative solutions to see if any of these could “delight” the customer

Having determined the function we need to identify the associated Non-functional Requirements – see later

©2016 Waters Corporation

22

Deducing Requirements from Non-functional Performance Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Non-functional Performance Requirements specify how well a particular function has to perform

Non-functional Implementation Requirement	Functional Requirement	Non-functional Performance Requirement
		Capacity 2 to 4 slices
		Different sizes of bread

Complete Non-functional Performance Requirements comprise a metric, an attribute and associated numerical target values. Expressed requirements are often not complete but will form the basis for determining numerical targets

Note: We are not designing the system and therefore the Non-functional Implementation column should not be completed. In the ideal world this column should be empty

Remember stakeholders like users or customers rarely specify functionality but imply it through the NFP requirements. We need to determine the function that is implied by the partial NFP requirement. Note in doing so we often have choice over the verb and need to choose verbs that in themselves don't infer a solution.

©2016 Waters Corporation

23

Deducing Non-functional Performance Requirements from Functional Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Functional Requirements specify what the system has to do in order to meet the Operational Requirement of the system

Non-functional Implementation Requirement	Functional Requirement	Non-functional Performance Requirement
	Generate Heat	
	Control Browning	

Note: We are not designing the system and therefore the Non-functional Implementation column should not be completed. In the ideal world this column should be empty

Functional Requirements are typically captured at this point as a simple verb noun description

The function defines the "job" to be done we now have to determine "how well" that job has to be done in order to specify the Non-functional Performance Requirements

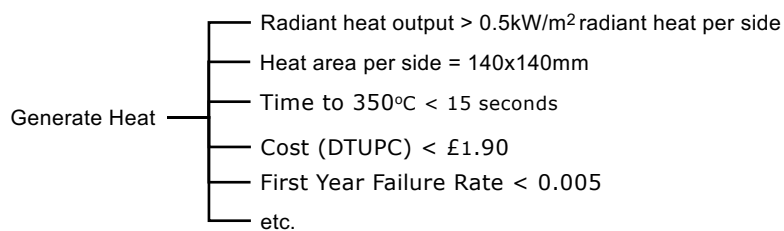
©2016 Waters Corporation

24

The Nature of Non-functional Performance Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Non-functional Performance Requirements comprise: a metric, measure or property and a numerical value as a target, lower or upper limit, that defines how well a particular function has to perform
- For any given function there WILL be several NFP requirements that are specific to that function



©2016 Waters Corporation

25

Deducing Non-functional Performance Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- If we are designing a system with a known meta-solution we should have most of the NFP requirements. It will be a matter of collecting and collating them against the Functions
- If not, a common approach here is to consider two solutions, say A and B, to each function and ask what could be measured to decide between them? The key here is to identify measurable characteristics
- We should also consider the flow-down of the Non-functional System requirements

©2016 Waters Corporation

26

Systemic Textual Analysis: Step 3 Clarify and Refine the Requirements

- The individual requirement fragments identified from step 2 should now be examined one at a time asking the following questions:
 - Is it correct? (is it asking for something possible?)
 - Is it complete? (how do I turn it in to a sentence?)
 - Is it clear and concise? (is it unambiguous in simple direct language?)
 - Is it consistent? (with the other requirements)
 - Is it confirmable? (is there a way in which we can demonstrate that we have met this requirement?)
- In answering these questions we may need to seek clarification from the source stakeholder. We should avoid making assumptions, but if there is no choice document the assumption

- The requirement should be, if necessary, rewritten to have positive outcomes to the above questions
- The set of requirements must be examined to ensure they form a complete consistent set with no contradictions
- We should always maintain traceability between our clarified and derived set of requirements and the original customer requirements

Much more later when Specifying Requirements

Systemic Textual Analysis

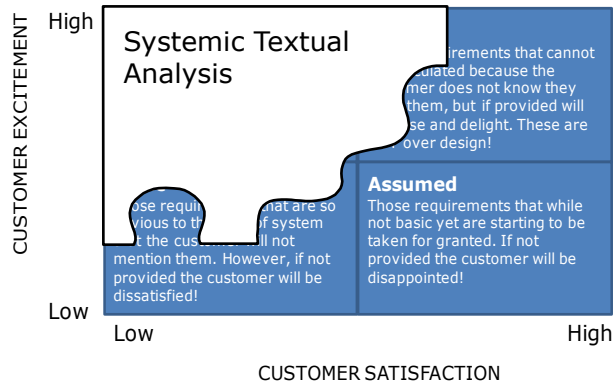
- | | |
|--|---|
| <ul style="list-style-type: none"> ■ Pros <ul style="list-style-type: none"> – Simple - anybody can contribute – Start with a list and end with a bigger list – If performed in a team can promote <ul style="list-style-type: none"> ○ Discussion ○ Thinking – Helps structure the requirements – Helps identify missing and ambiguous requirements – Helps in deducing requirements | <ul style="list-style-type: none"> ■ Cons <ul style="list-style-type: none"> – Requirements categories require some practice to be comfortable – Not very useful if the requirements set is very small – Difficult to convey structure and relationships between requirements – Difficult to verify completeness and consistency – Will NOT uncover all of the unspoken requirements |
|--|---|

Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Systemic Textual Analysis often helps us “get into” the problem by sorting the requirements into logically related categories that enable missing requirements to be identified. It will not guarantee to uncover all of the requirements because it is very much dependent upon the quality of the source requirements

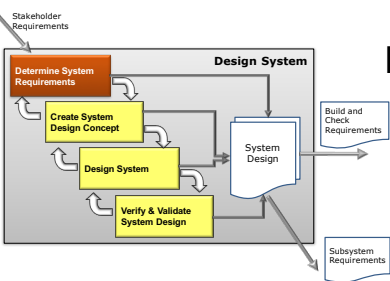


To get at the other unspoken requirements we will have to tackle the problem from another direction





THE SCIENCE OF WHAT'S POSSIBLE.®



The diagram shows a 'Design System' box containing 'Determine System Requirements', 'Create System Design Concept', 'Design System', and 'Verify & Validate System Design'. Arrows show a clockwise cycle. External boxes include 'Stakeholder Requirements' (input), 'Build and Check Requirements' (output), and 'Subsystem Requirements' (output).

Analyse Requirements Modelling Requirements using Functional Modelling

This Session:

- To describe and practise Functional Modelling as a tool that can be used to help in Analysing Requirements by building an operational model of the system of interest

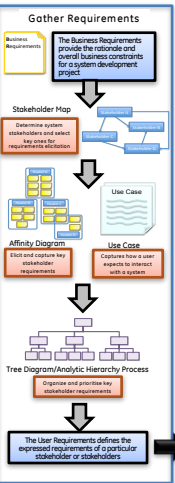
©2016 Waters Corporation

1

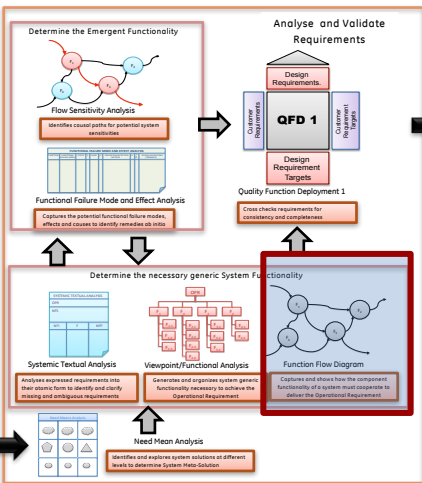
Process and Tools



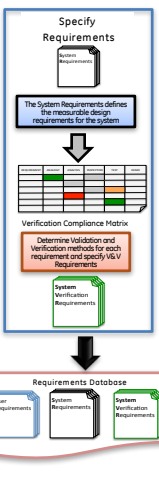
THE SCIENCE OF WHAT'S POSSIBLE.®



Flowchart for 'Gather Requirements' showing steps: Stakeholder Map, Affinity Diagram, Tree Diagram/Analytic Hierarchy Process, and User Requirements.



Flowchart for 'Analyse and Validate Requirements' showing steps: Determine the Emergent Functionality, Determine the necessary generic System Functionality, and Need Mean Analysis.



Flowchart for 'Specify Requirements' showing steps: Specify Requirements, Verification Compliance Matrix, and Requirements Database.

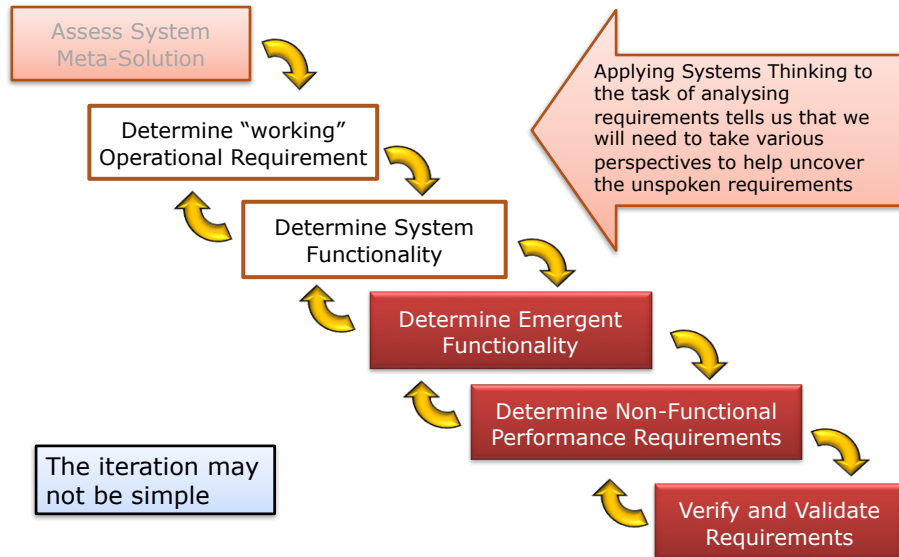
©2016 Waters Corporation

2

Where are we?: Analyse Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



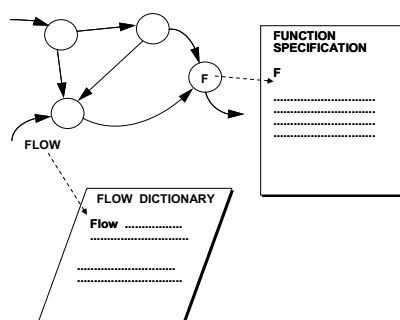
©2016 Waters Corporation

3

Functional Modelling

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- What is it?
 - Functional Modelling is a simple, but incredibly powerful tool to create a readable and maintainable model of an existing or planned system that highlights the functionality and the interconnectivity between the system functions
- Why use it?
 - Functional Modelling uses the logic of the system "problem" to determine the functionality that is necessary to deliver the Operational Requirements of a system
- Where to use it?
 - Whenever attempting to determine and analyse a set of requirements.
 - Particularly useful when there is a need to ensure that the required system functionality is complete and consistent
- How does it work?
 - Functional Modelling uses three sub-tools to build a model of the system. These are:
 - FUNCTION FLOW DIAGRAM
 - FLOW DICTIONARY
 - FUNCTION SPECIFICATION

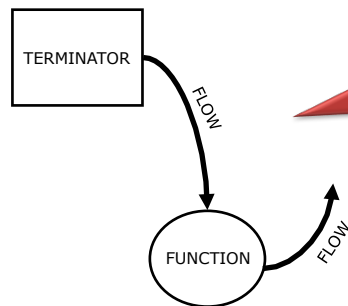
©2016 Waters Corporation

4

FUNCTION FLOW DIAGRAM (FFD) (Similar to Data Flow Diagrams)

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Is a network representation of the system in terms of its component functions and the interdependencies or “flows” between the functions



One of the beauties of the Function Flow Diagram is its limited diagramming convention. This enables the team to concentrate on understanding the system rather than the modelling notation

©2016 Waters Corporation

5

Function Flow Diagram Conventions

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

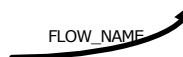
- FUNCTION:** transforms input flows into output flows
- TERMINATOR:** a Function or source or destination of flows external to the system
- FLOW:** an input or output quantity (information, control, material or energy)



The **FUNCTION NAME** - best defined as a simple verb noun phrase



TERMINATOR NAME - either a verb noun phrase (**FUNCTION**) or a noun phrase (**thing**)



The **FLOW NAME** - a noun (**thing**)

Note: Software Engineers may have come across a similar modelling method called Data Flow Diagrams. These have similar diagramming conventions and use a fourth symbol to represent data stores. In general system modelling this fourth symbol is less useful and has been omitted. The aim with FFDs is simplicity.

©2016 Waters Corporation

6

Flow Dictionary and Function Specification

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Flow Dictionary (FD)
 - Complements the Function Flow Diagrams by documenting each of the flows on any of the diagrams
 - It is a complete set of formal definitions which are accomplished by declaring the component elements of each input or output flow and the relationships that apply among them
- Function Specification (FS)
 - Specifies the component functions by defining the transformation that converts inputs to outputs
 - Function specifications are written in a structured language

©2016 Waters Corporation

7

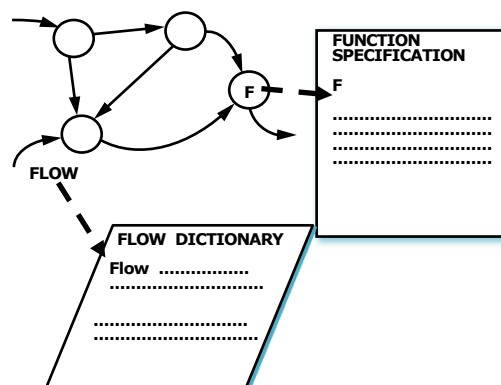
The Model

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

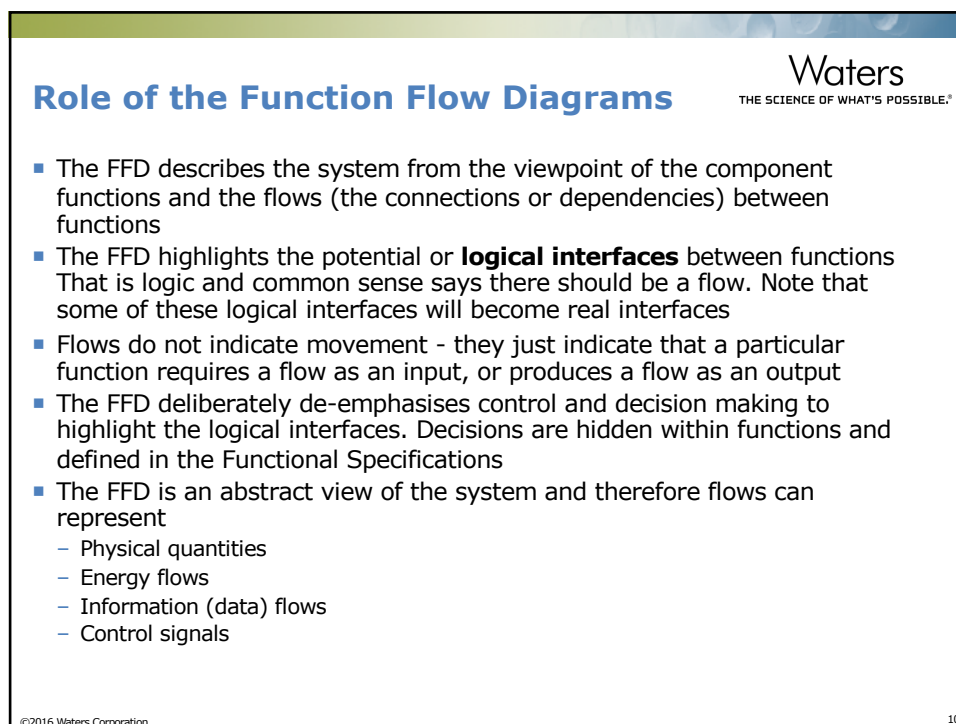
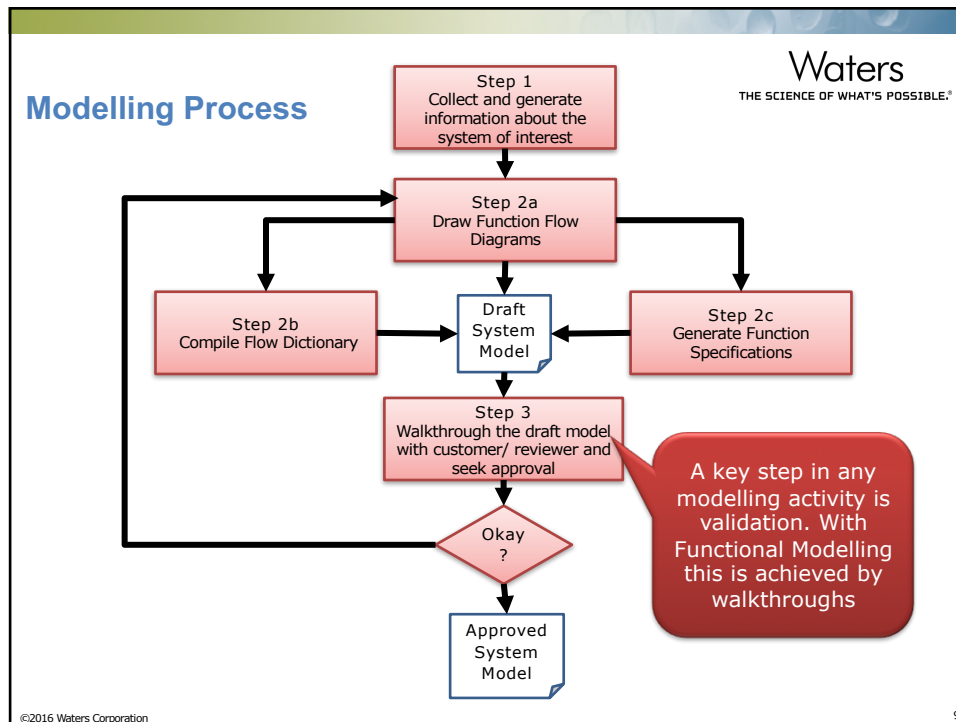
- The three sub-tools allow the construction of a model which is consistent, complete and unambiguous
- The key element of the model is the Function Flow Diagram
- This model provides the first estimate of

$$Y = f(x, u)$$



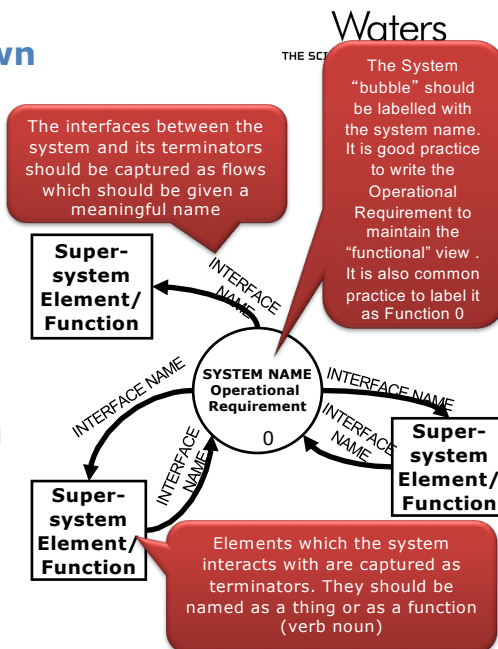
©2016 Waters Corporation

8



Structured and Top Down

- For anything larger than a very small system, a single FFD will be cluttered with detail and not easily comprehended
- To overcome this a 'Top-Down' treatment is used to produce a levelled set of FFDs
- The starting point is to represent the system of interest as a single function; showing the boundary of the system and its interactions and interfaces with the environment - the "super-system"
- This top level FFD is called the **CONTEXT DIAGRAM** - it shows the system in its operational context

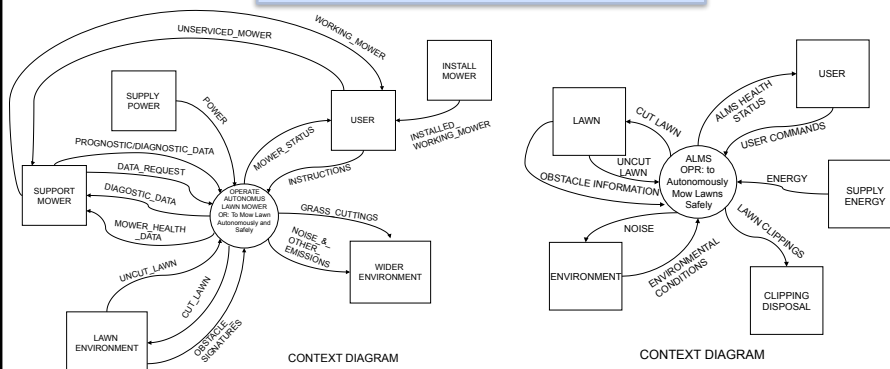


©2016 Waters Corporation

11

Context is Important

Both Diagrams are "right" but show a different perspective of the system.



It is often convenient to start by considering the operational perspective first.

©2016 Waters Corporation

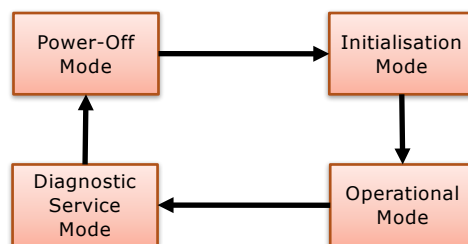
12

Modes of Operation

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Many Systems have several modes of operation, for example: Deployment, Start up, Operation, Shut down, etc.
- When attempting to model such systems, a single “all-encompassing” model is often not possible
- In such cases it is best to identify the modes of operation and construct a model for each mode



It is often easier to consider each mode separately and construct several models

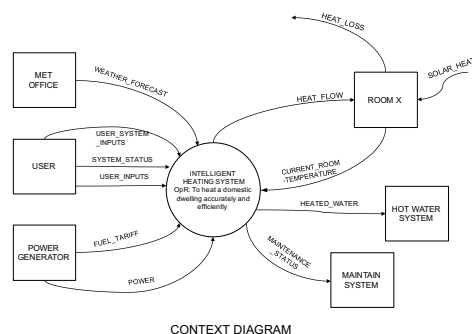
©2016 Waters Corporation

13

Exercise – Context Diagram

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



In teams

- Construct a CONTEXT DIAGRAM for the case study example
- Assume the system is fulfilling its prime purpose
- Flip chart the outcomes and be prepared to share with the rest of the group

Time: 20 Minutes

©2016 Waters Corporation

14

Defensible Logic to Create the Lower Level FFDs

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

From the system purpose, the approach is to logically deduce the necessary system functionality i.e. what the system will have to do to meet its purpose.

We use the input flows from the CONTEXT DIAGRAM to determine the functionality logically necessary to generate the output flows.

Central to this logic is that every function **MUST** have at least one input flow and one output flow

Rules:

- The CONTEXT DIAGRAM is decomposed on the next level of FFD into a number of functions that can realistically be represented on a single diagram. This is called DIAGRAM 0
- A rule of thumb is that there should be no more than 7 functions on any one FFD
- The functions of DIAGRAM 0 are taken in turn and decomposed further to form lower-level FFDs
- This activity of successive decomposition is continued down to the 'primitive' level where the functions cannot usefully be further decomposed.

©2016 Waters Corporation

15

Lower Level FFDs

Waters

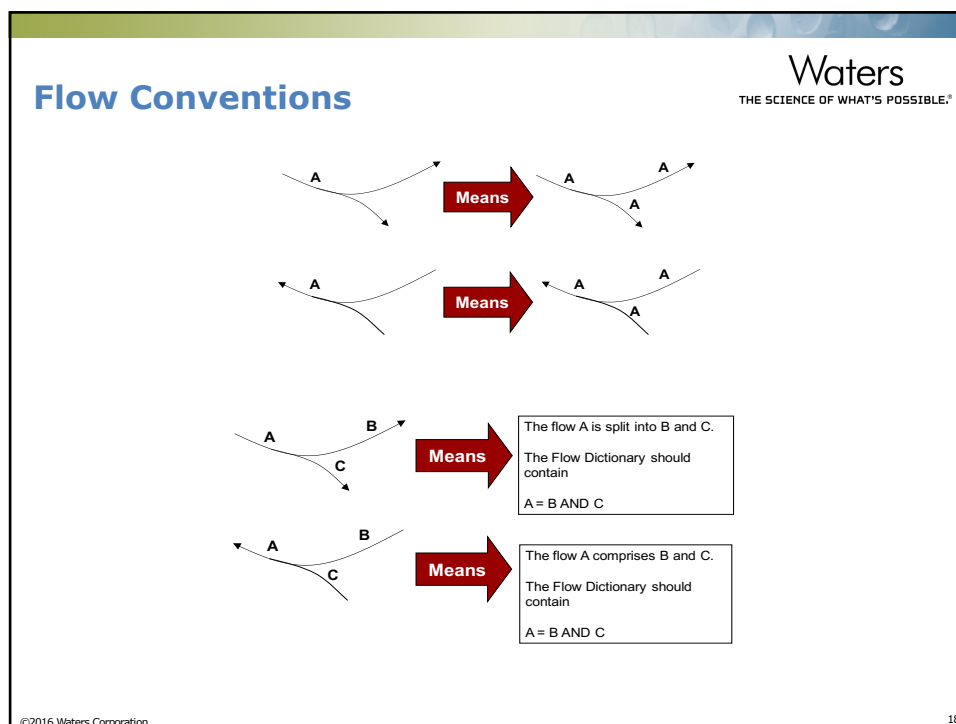
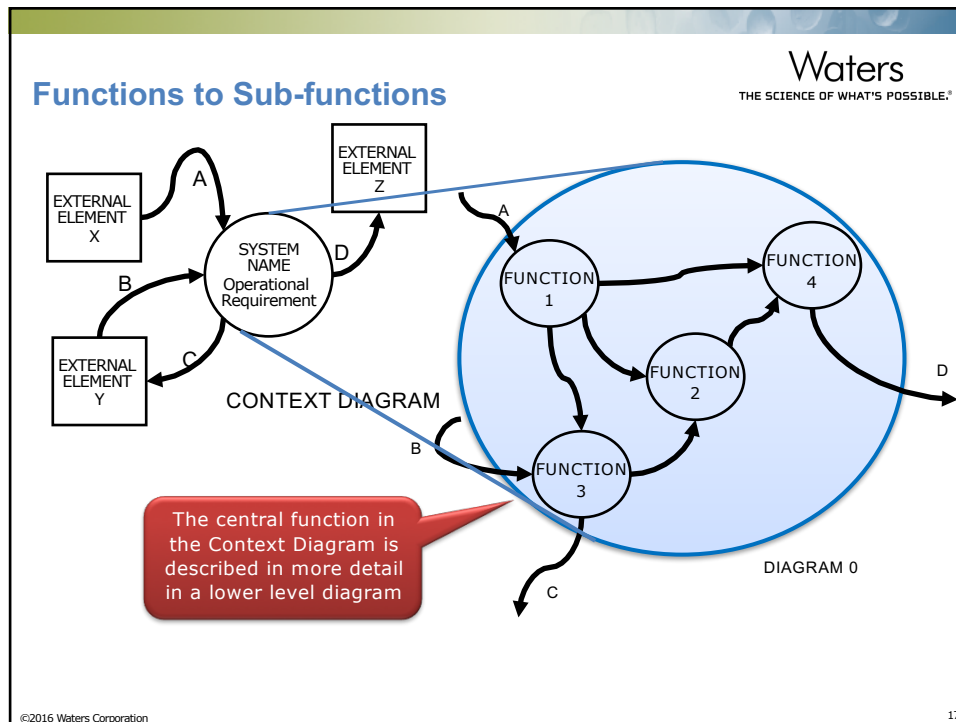
THE SCIENCE OF WHAT'S POSSIBLE.®

- The CONTEXT DIAGRAM is decomposed on the next level of FFD into a number of functions that can realistically be represented on a single diagram. This is called DIAGRAM 0
- A rule of thumb is that there should be no more than 7 functions on any one FFD
- The functions of DIAGRAM 0 are taken in turn and decomposed further to form lower-level FFDs
- This activity of successive decomposition is continued down to the 'primitive' level where the functions cannot usefully be further decomposed:
 - decomposition should continue down to, and stop at the level where it is necessary to make design decisions

– The purpose of functional decomposition is to logically deduce the necessary system functionality – we use the tool to help us logically deduce functionality i.e. what the system will have to do. At this point we are actually engineering requirements. This is not design: we are not specifying how something is to be done but deciding what is to be done. Please note that “bad” requirements engineering cannot be compensated for by “good” design!

©2016 Waters Corporation

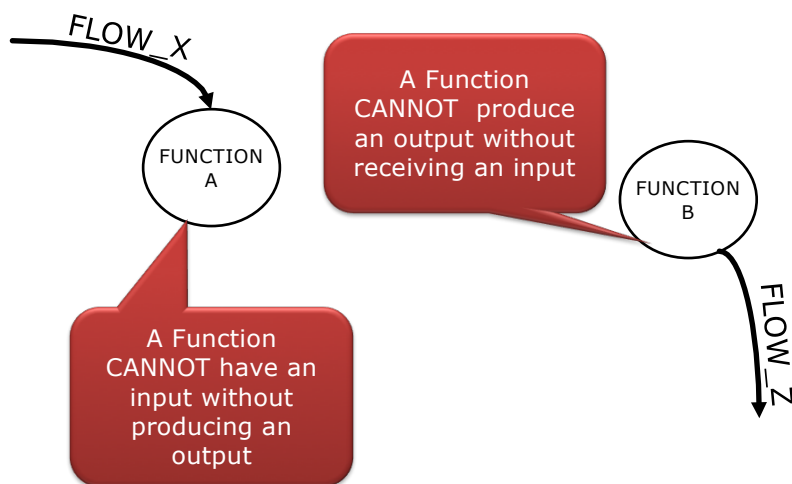
16



Use Logic

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



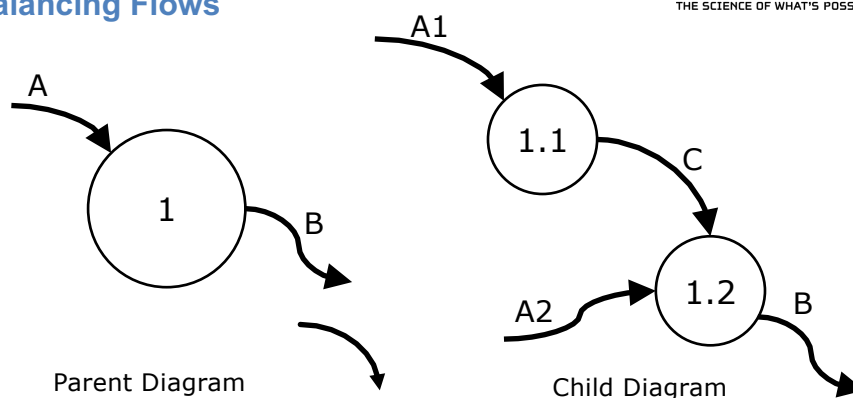
©2016 Waters Corporation

19

Balancing Flows

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

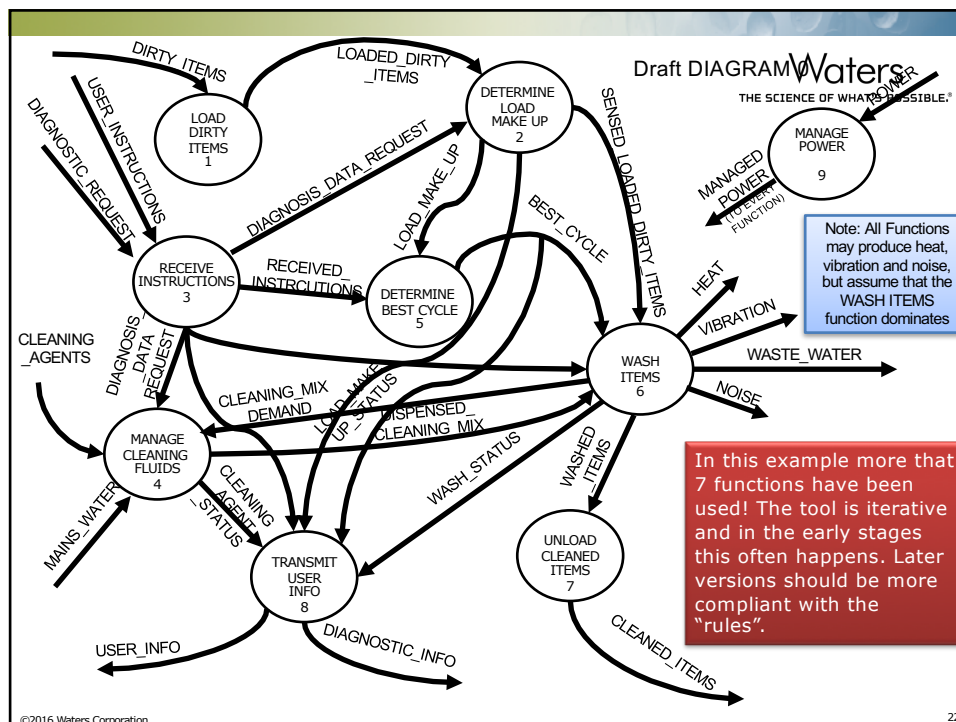
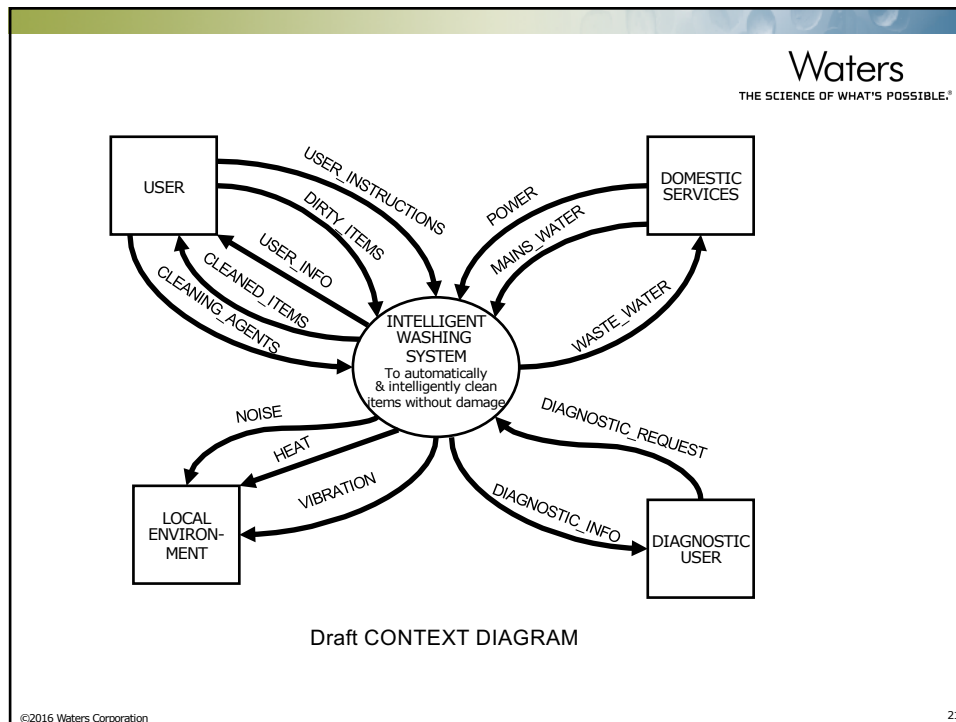


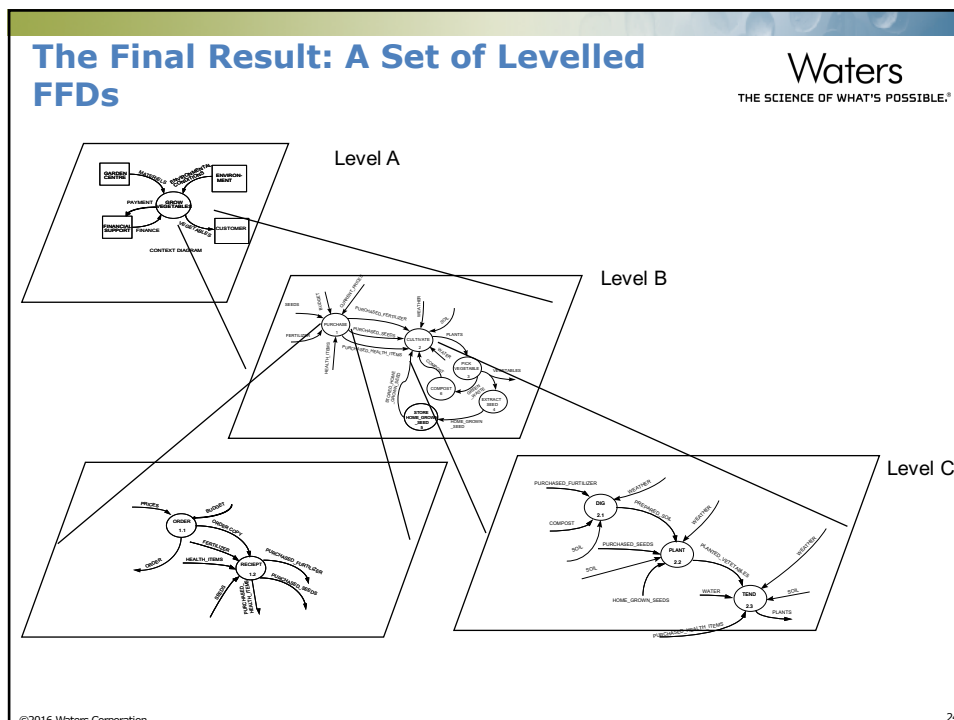
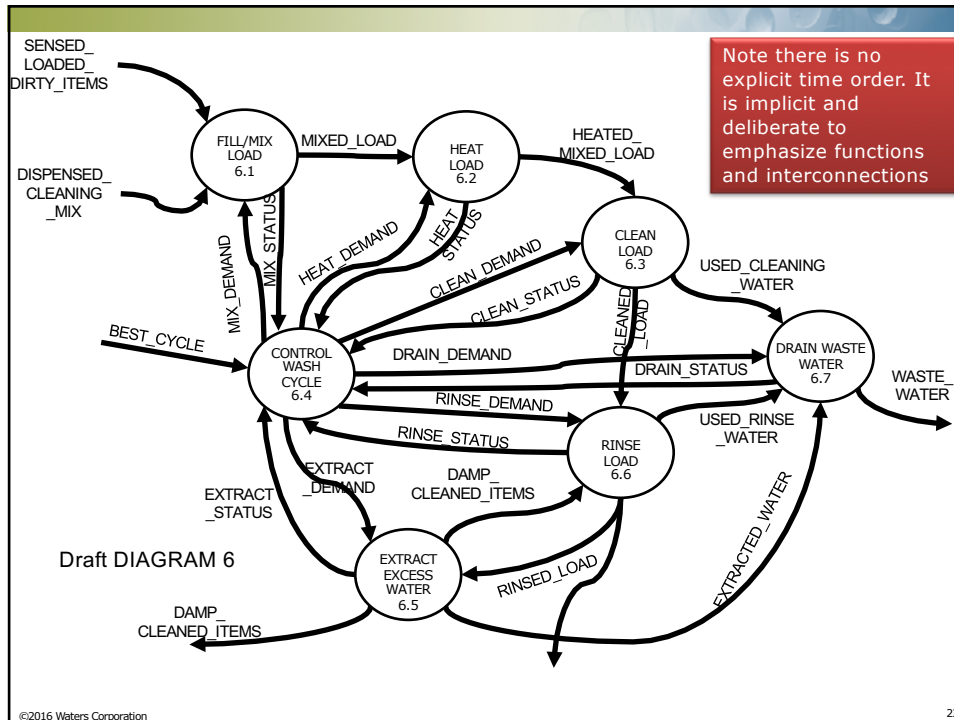
Inputs and outputs should balance between parent and child diagrams. Just as we have functional decomposition to show increasing detail, diagrams can also have flow decomposition with the relationship defined in the Flow Dictionary

$$A = A1 \text{ AND } A2$$

©2016 Waters Corporation

20

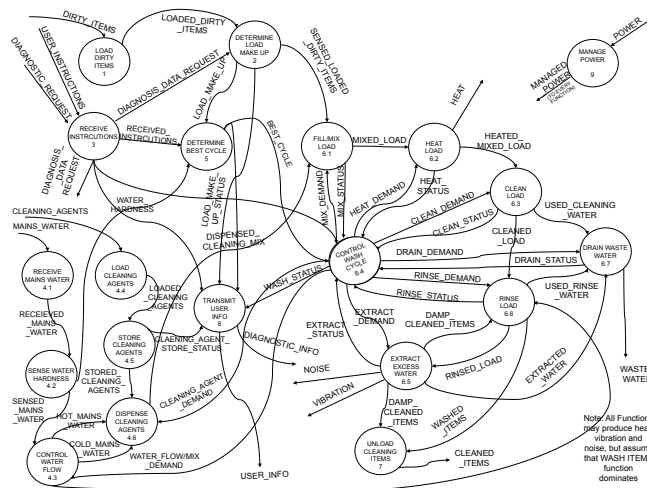




At the Primitive Level

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



If we were to combine all the Function Flow Diagrams together at the lowest level of detail (the primitive level) we would have a single picture of the system requirements in terms of the system functionality and the interdependencies

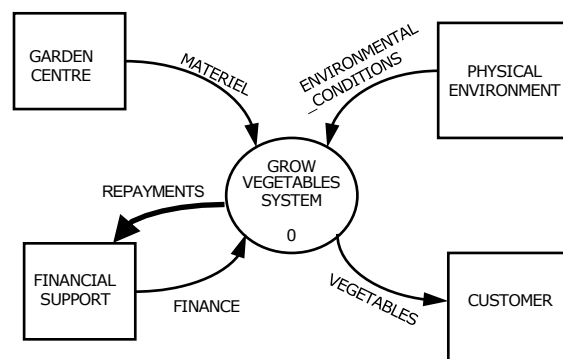
©2016 Waters Corporation

25

Example: Service Type System Context Diagram

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



CONTEXT DIAGRAM

©2016 Waters Corporation

26

Example Diagram 0

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

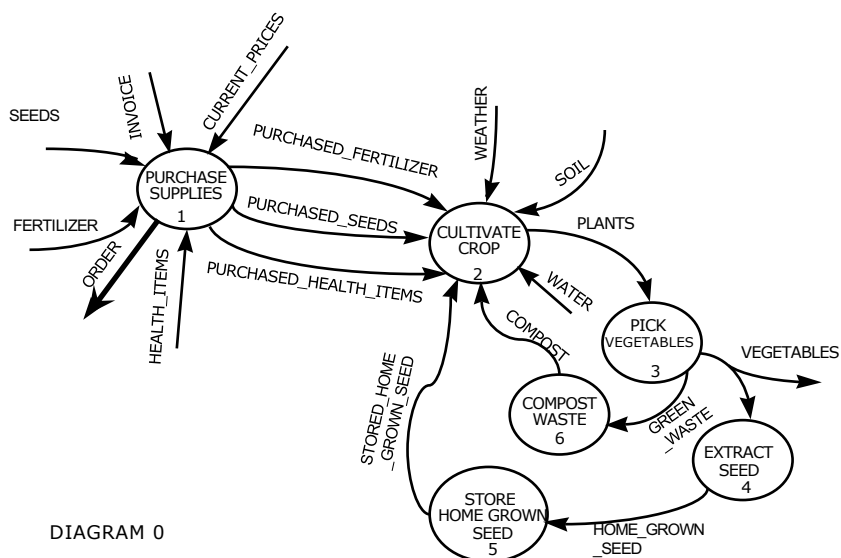


DIAGRAM 0

©2016 Waters Corporation

27

Example Diagram 2

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

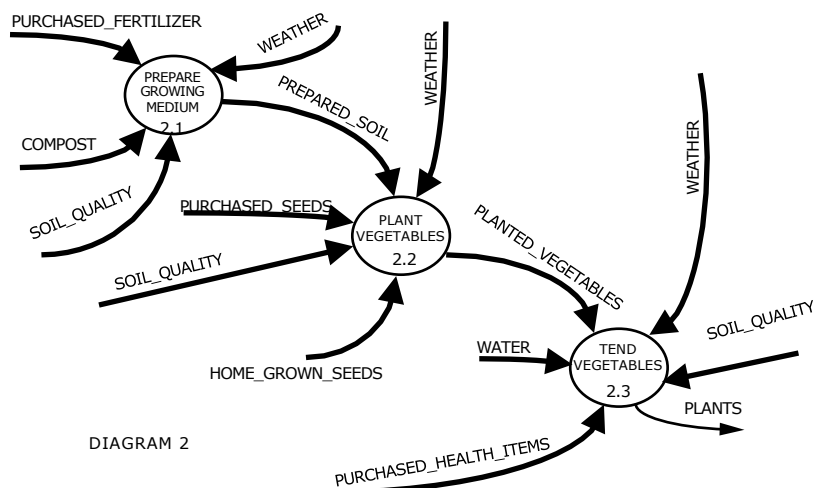


DIAGRAM 2

©2016 Waters Corporation

28

Constructing Function Flow Diagrams

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Constructing a set of Function Flow Diagrams is iterative
 - Start with the Context Diagram but expect it not to be complete until some of the lower diagrams have been drafted
 - Be prepared to modify diagrams
 - Work forwards and backwards
 - Forwards: start with a system input (look for the one that initiates the operational requirement) and ask what happens to it as a "sequence" of functions that leads to an output
 - Backwards: start with a system output (look for the one that relates to the operational requirement) and determine what functions would deliver this output, then ask what inputs are necessary, then what functions generate those inputs etc.
 - Diagrams should "tell a story"
 - Consider the operational view of the prime system first
- The Viewpoint Structure Chart is often a good starting point for constructing DIAGRAM 0 – BUT BEWARE:
 - You need to be clear on the system you are modelling and the context (e.g. Operational model vs. lifecycle model)
 - All the functions identified on a Viewpoint Structure Chart should appear on the Function Flow Diagrams BUT the groupings/levels selected may be different (at some point the structure chart should be updated to reflect the Function Flow Diagrams)

©2016 Waters Corporation

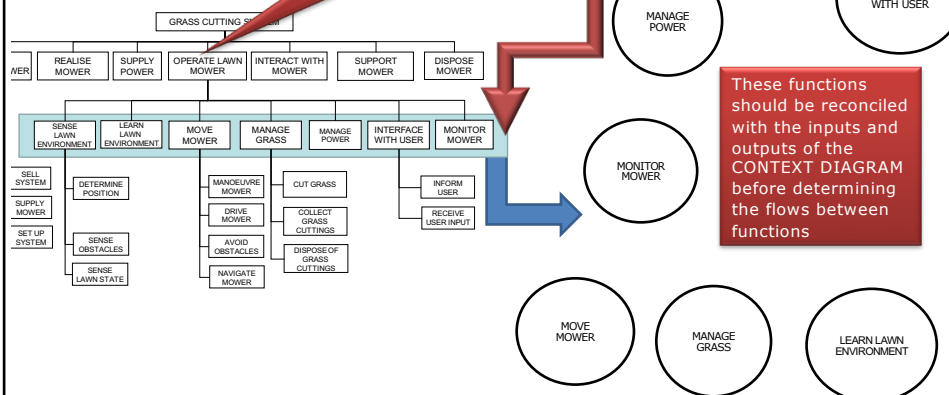
29

Starting DIAGRAM 0

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Use the Viewpoint Structure Chart to help draw DIAGRAM 0. Identify the System of Interest



©2016 Waters Corporation

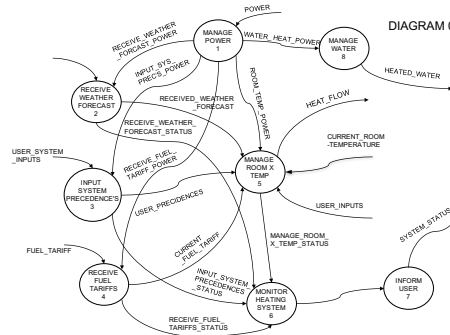
30

Exercise: Diagram 0 and Beyond

In teams

- Using the earlier CONTEXT DIAGRAM construct lower level Function Flow Diagrams – require at least Diagram 0 and one lower level FFD (don't bother with the Flow Dictionary or Function Specifications)
- Flip chart the outcomes and be prepared to share with the rest of the group

Time: 45 Minutes



Role of the Flow Dictionary

- The Flow Dictionary (FD) specifies precisely what is meant by each and every flow on the FFDs
- Just as the FFDs show the decomposition of the functions, the Flow Dictionary performs a similar task for the Flows
- Remember that the Flows are often abstracted and show “information” such as voltages, fluid flows, material flows etc. and the dictionary is important in making sure this is clear.
- Remember also that every Flow represents a logical interface (that could be a real interface in the real world) hence the Flow Dictionary is specifying the system interfaces

Flow Dictionary Conventions

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The Flow Dictionary makes use of 5 operators that allow for the definition of any possible entry:
 - IS EQUIVALENT TO or =
 - AND or +
 - EITHER-OR or [Item1, item2]
 - REPETITIONS OF { item }
 - OPTIONAL (item)

Example

- TELEPHONE_DIRECTORY = { [BUSINESS_NAME, PERSONAL_SURNAME + (INITIALS)] + ADDRESS + TELEPHONE_NUMBER }

The underscore is used to indicate that this is a single item and not two or more items

- The Flow Dictionary specifies the logical interfaces of the system – it is an early Interface Requirements Document (IRD)

Examples

- CLEANING_AGENTS = DETERGENT + (STAIN_REMOVER) + (FABRIC_CONDITIONER) + (BLEACH) + (DYESTUFF)

At the root definition a common approach is to change case

- DETERGENT = all currently available powders, tablets and liquids in IS 9756
- POWER = 28V dc supply to EN 719745

It is in the Flow Dictionary that we can capture the Non-functional Requirements

©2016 Waters Corporation

33

Role of the Function Specification

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The Function Specification provides more information about a particular function.
- It should describe in more detail what the function does and if possible describes the rules governing the transformation of input flows to output flows
- Usually written in a structured specification language based on high level procedural software languages

FUNCTION 8.1 MONITOR IWM – this function will collect appropriate health data from potentially all other functions in order to perform a self-test and report on the current status of the system.

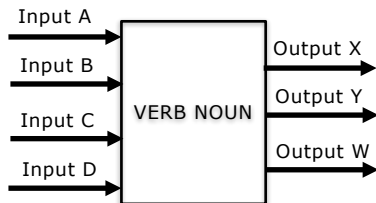
```
IF POWER on
  DO THE FOLLOWING
    Collect Health Data
    Execute SELF_TEST
    IF SELF_TEST Failed
      Generate USER_MESSAGE
      Deliver USER_MESSAGE
      If critical initiate system shut-down
    ELSE IF SELF_TEST ok
      Generate USER_MESSAGE
      Deliver USER_MESSAGE
      Initiate WASH_CYCLE
  END
UNTIL POWER OFF
```

©2016 Waters Corporation

34

Creating a Functional Dictionary

- The initial Functional definition should comprise:



Or a
CONTEXT
DIAGRAM
for
each Function

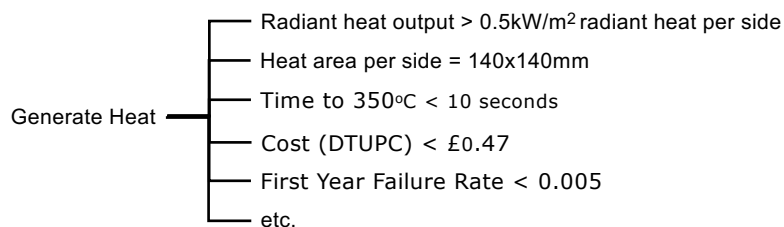
```

FUNCTION 8.1 MONITOR IWM – this function
will collect appropriate health data from
potentially all other functions in order to
perform a self-test and report on the
current status of the system.
IF POWER on
DO THE FOLLOWING
  Collect Health Data
  Execute SELF_TEST
  IF SELF_TEST Failed
    Generate USER_MESSAGE
    Deliver USER_MESSAGE
    If critical initiate system shut-
    down
  ELSE IF SELF_TEST ok
    Generate USER_MESSAGE
    Deliver USER_MESSAGE
    Initiate WASH_CYCLE
  END
UNTIL POWER OFF
    
```

Function Performance Requirements

Functions to Requirements

- Functions specify what a system has to do to meet a system's Operation Requirement. In turn we need to specify how well the individual functions have to perform. The Holistic Requirement Model calls these the Non-functional Performance Requirements
- Non-functional Performance Requirements comprise: a set of metrics, measures or properties and associated numerical value as a target, lower or upper limit, that defines how well a particular function has to perform
- For any given function there WILL be many NFP requirements that are specific to that function



Old World meets New World

- Function Flow Diagrams, Flow Dictionary and Function Specification can provide a link between Systems Engineering and modern Object Oriented approaches and Agile programming
 - The Context Diagram and Lower Level Diagrams are a nothing more than a sophisticated USE Case Diagram
 - The Function Specification (Function Dictionary) are both internal and external USE Cases
 - The Flow Dictionary provides a draft Data Structure
- Care must be exercised as to when the decision to move into software is appropriate

Deducing Non-functional Performance Requirements

- If we are designing a system with a known meta-solution we should have most the of the NFP requirements. It will be a matter of collecting and collating them against the Functions
- If not, a common approach here is to consider two solutions, say A and B, to each function and ask what could be measured to decided between them? The key here is to identify measurable characteristics
- We should also consider the flow-down of the Non-functional System requirements

Summary Functional Modelling

■ Pros

- Simple to use (only three modelling symbols)
- Universal – can model any type of system
- Builds a defensible model of how functions logically have to co-operate to deliver the system purpose
- Identifies logical system interfaces
- Whole team involvement giving more shared knowledge and a common understanding
- Oversights and errors detected early
- Visual standard communication
- Systematic method
- Documentation as outcome
- Customer involvement

■ Cons

- Very useful for operational type functionality in the first iteration
- Restricted symbology can be confusing
- Difficult to accurately model real time systems¹ – does not show control flows

¹There are many other Functional/Behavioural modelling tools. Many are able to handle real-time systems, command and control issues, etc. The modelling tool presented here, however, is SIMPLE which allows the WHOLE TEAM to be involved leading to greater team understanding of the system problem and consequently a better system design

Other Modelling Tools

■ There are many other functional/behavioural modelling tools:

- IDEF0 (Integrated Definition for Functional Modelling)
- SADT (Structured Analysis and Design Technique)
- Data Flow Diagrams
- Control Flow Diagrams
- State Transition Diagrams
- Entity Relationship Diagrams
- CORE (Controlled Operational Requirements Expression)
- UML (Unified Modelling Language)
- SysML (System Modelling Language)
- and hundreds more

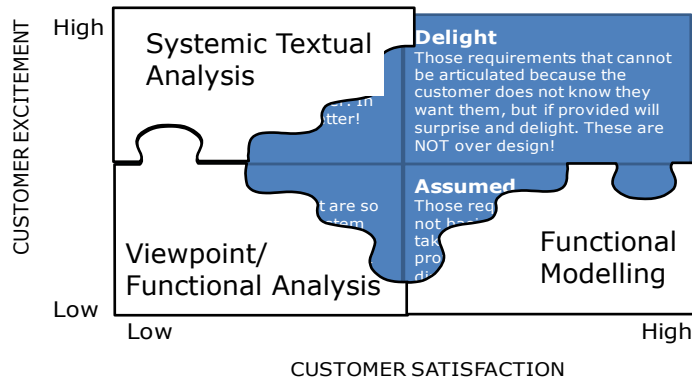
■ It is horses for courses – they all have value but some are not simple

Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Functional Modelling helps the team construct a model of the necessary system functionality and how it logically “co-operates” to deliver the Operational Requirements. This allows for the identification of further system functionality and moreover the logical system interfaces providing an early understanding of $Y = f(x,u)$

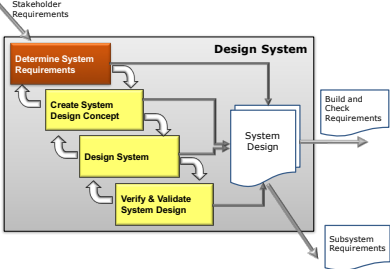




BURGE
HUGHES
WALSH



Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



The diagram shows a 'Design System' box containing 'Determine System Requirements', 'Create System Design Concept', 'Design System', and 'Verify & Validate System Design'. Arrows show a cyclical process. External boxes include 'Stakeholder Requirements', 'Build and Check Requirements', and 'Subsystem Requirements'.

Analyse Requirements Sensitivity and Failure Analysis

Assessing Requirements for Emergent Sensitivities

This Session:

- Introduces the concept of system sensitivity where small variations in system parameters can have a large impact on system performance
- Introduces the concept of robust design
- To describe and practise Sensitivity and Failure Analysis as tools that can be used to assess system sensitivity and thereby design out issues *ab initio*

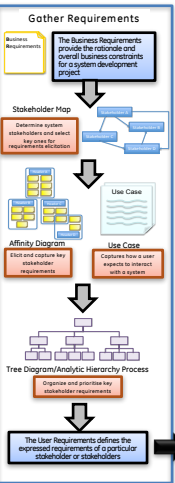
©2016 Waters Corporation

1

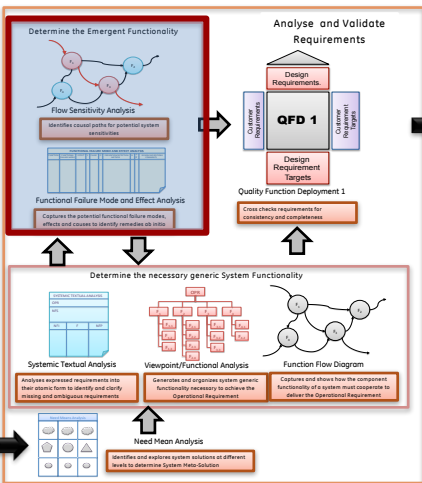
Process and Tools



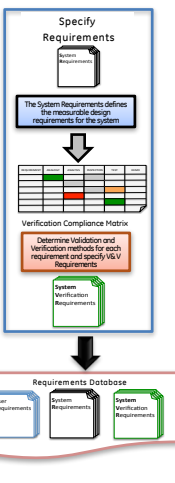
Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



The diagram shows a vertical flow from 'Business Requirements' through 'Stakeholder Map', 'Affinity Diagram', 'Tree Diagram/Analytic Hierarchy Process', and 'Use Case' to 'The User Requirements defines the expressed requirements of a particular stakeholder or stakeholders'.



This central section includes 'Determine the Emergent Functionality' (with Flow Sensitivity Analysis and Functional Failure Mode and Effect Analysis), 'Determine the necessary generic System Functionality' (with Systemic Textual Analysis, Viewpoint/Functional Analysis, and Function Flow Diagram), and 'Need Mean Analysis'. It also shows 'QFD 1' and 'Quality Function Deployment 1'.



The diagram shows a flow from 'Specify Requirements' through 'Verification Compliance Matrix' to a 'Requirements Database' containing 'User Requirements', 'System Requirements', and 'System Verification Requirements'.

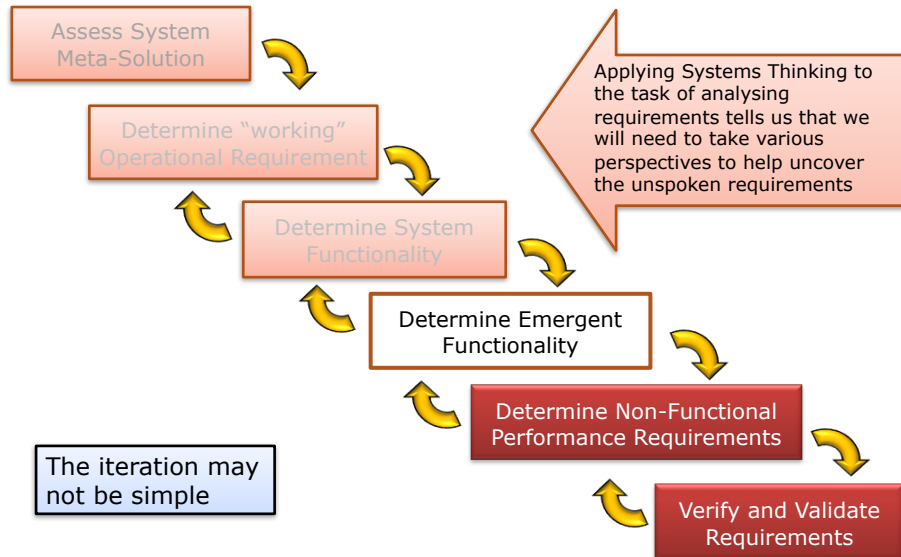
©2016 Waters Corporation

2

Where are we?: Analyse Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

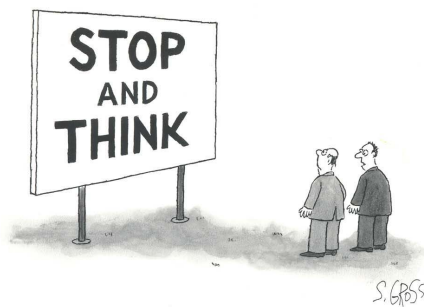
3

Introduction

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The increase in complexity of systems has resulted in many displaying undesirable emergent behaviour. Hindsight often shows these to be "obvious" and therefore avoidable – "if only we had time to think!"



"It sort of makes you stop and think, doesn't it."

- One aspect of undesirable emergence is system sensitivity and failure which can be assessed before a design exist to provide:
 - design suggestions to prevent or detect the potential issue(s)
 - an early understanding of possible trade-offs
 - an identification of critical functions
 - the identification of "sensitive flows"
 - the identification of potential system failure modes, their effects and causes
- Such analyses will provide an early assessment of risk and promotes ideas and guidance for generating and selecting conceptual solutions

©2016 Waters Corporation

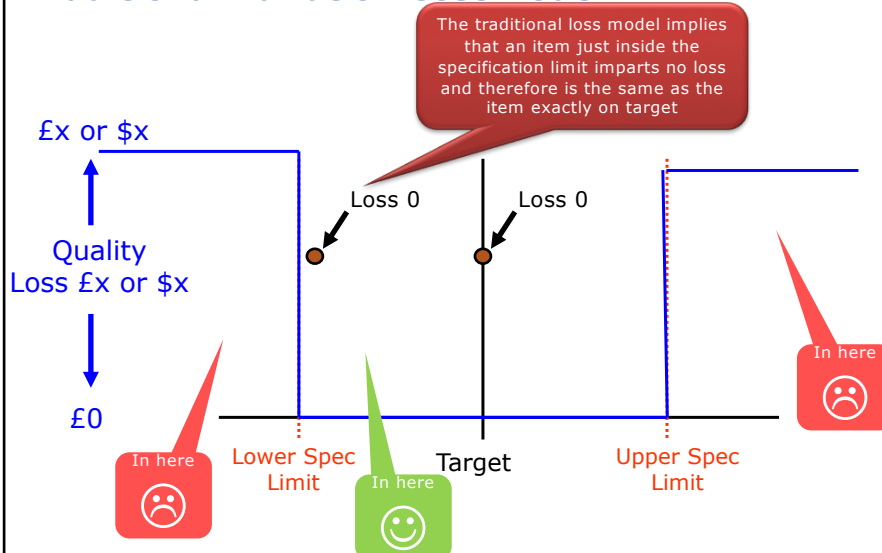
4

Philosophy

- An understanding of undesirable emergent behaviour due to system sensitivity can be provided through the work of Dr Genichi Taguchi
- Taguchi realised that apparently minor variations in a system parameter can have a significant effect on overall system performance
- He also realised that aspects of the traditional approach to sensitivity through the use of tolerances and specification limits can be very misleading
- Taguchi proposed the idea of robust design



Traditional Variation Cost Model



THE SCIENCE OF WHAT'S POSSIBLE.®



7

THE SCIENCE OF WHAT'S POSSIBLE.®

- Taguchi used the Taylor series to mathematically prove to the first order:

$$Loss = k(y - m)^2 = \Delta^2$$

- Taguchi applied his concepts to products, we now realise that it applies to any system

©2016 Waters Corporation

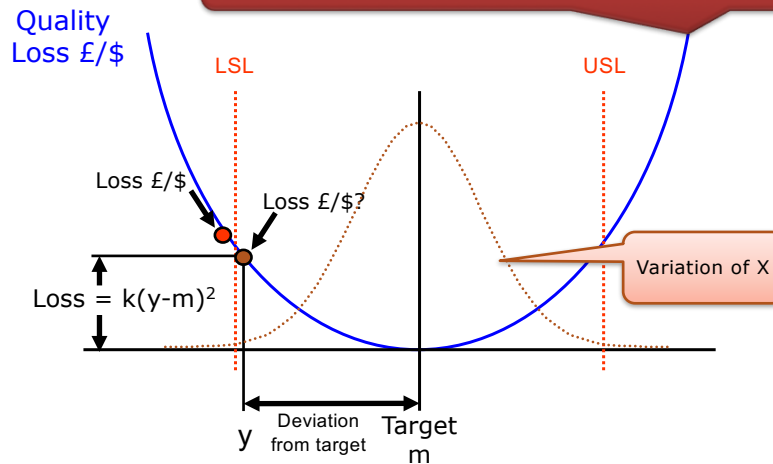
8

Taguchi Variation Cost Model

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Taguchi's loss function states that any system that does not meet its target specification will either impart a loss to the company or a loss to the customer (he generalised this to talk about a loss to society)



©2016 Waters Corporation

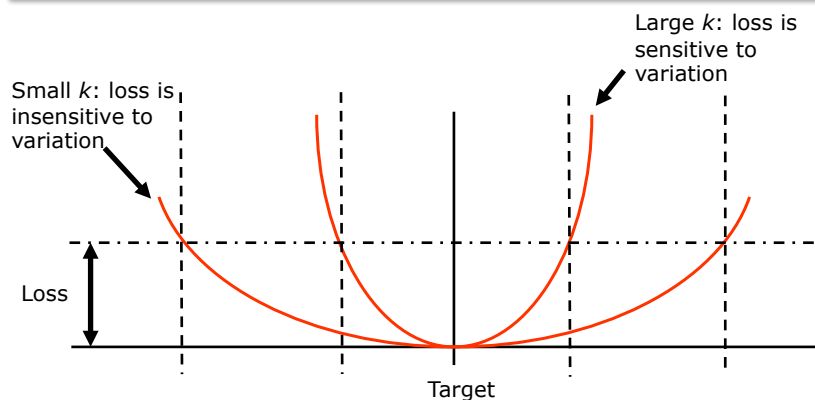
9

Effect of Variation

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Every component in a system will have its own characteristic loss function, but the 'degree of loss' is dependent upon the constant k .

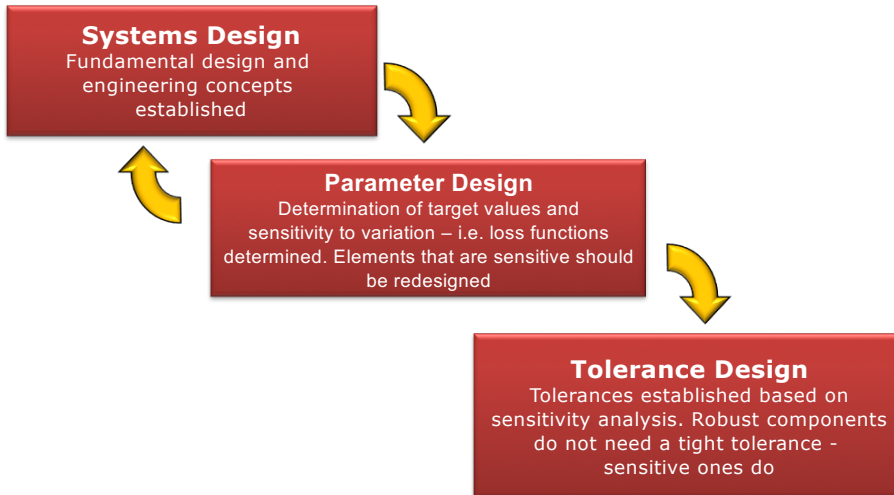


Note: Manufacturing Variation is only one of three sources of variation that lead to performance degradation. The other two are environmental and use. System Designers often take account of these as a matter of course – it's the Manufacturing variation they often choose to ignore!

©2016 Waters Corporation

10

The Robust Design Approach



©2016 Waters Corporation

11

Taguchi and Requirements Interpretation

- Taguchi's approach is applied only once the basic concept design has been agreed which could result in an expensive redesign (but not as expensive as a redesign after production has started)
- It is possible to apply his philosophy during requirements determination since:
 - all parameters in a system are subject to target values
 - functionality will be affected through the variation of input quantities about their respective target values
 - functionality will be affected by the variation of the function's operation
- In other words it is possible to use previous experience and common sense to identify potential issues before design commences and hence design out problems ab initio

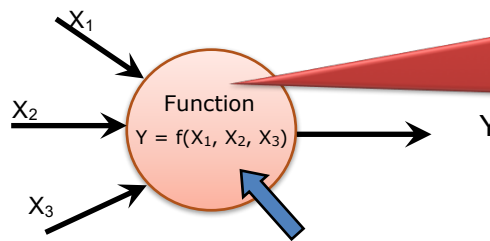
©2016 Waters Corporation

12

Functional Sensitivity Analysis

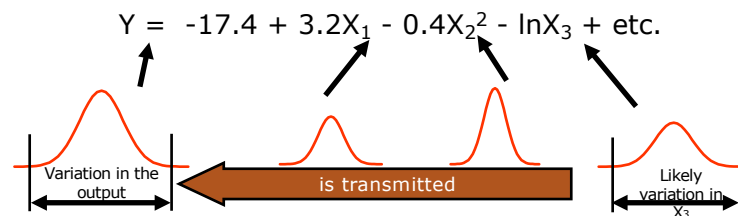
Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Taguchi's approach requires us to know the transfer function between the inputs and outputs of a particular function. This transfer function can typically only be determined after it has been designed

What is f ? - the relationship between the Y and the X s - if we know what f is and the variation in the X s we could determine the impact on the output



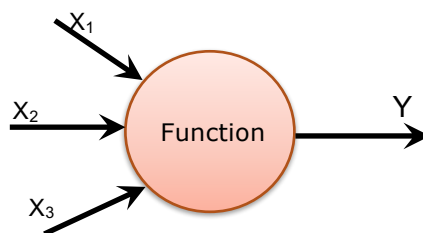
©2016 Waters Corporation

13

Functional Requirement Sensitivity Analysis

Waters

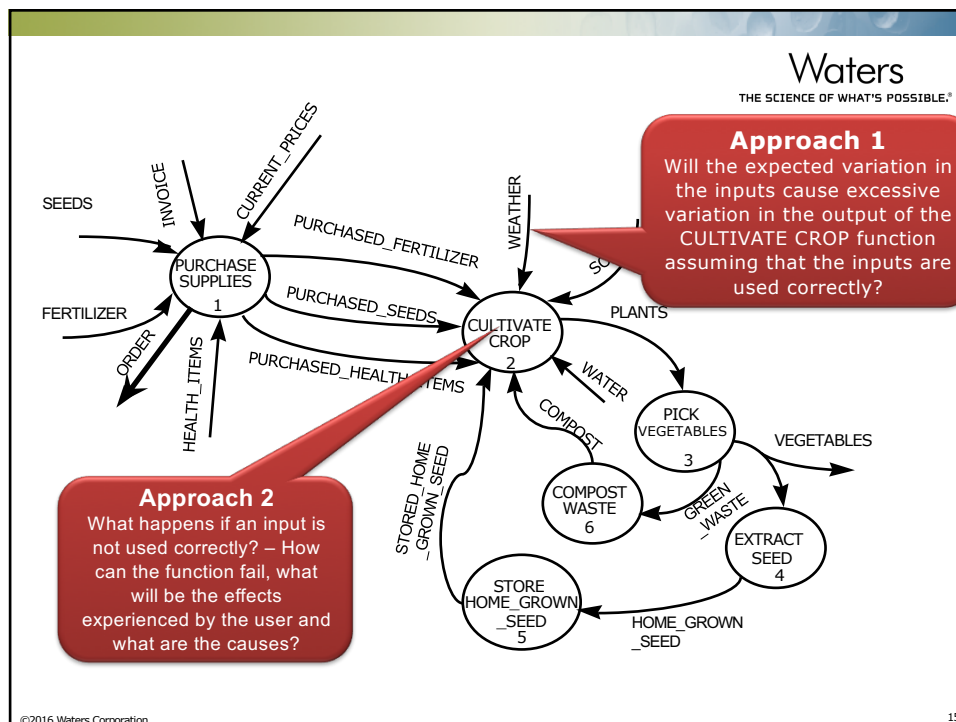
THE SCIENCE OF WHAT'S POSSIBLE.®



When engineering requirements we do not know what " f " is (we know its name but not the mathematics) – but we can use the knowledge of the team to assess how sensitive a function's outputs are to the likely input variation and how that function might fail

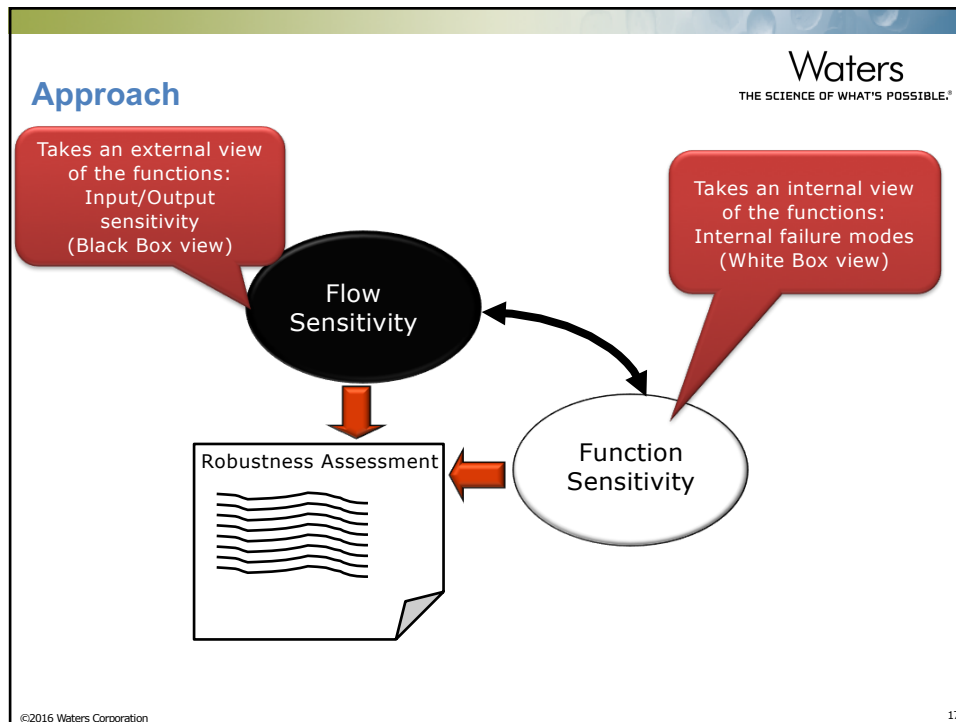
©2016 Waters Corporation

14



Sensitivity Analysis the Approach

- The basic approach is one of applied common sense! It is about questioning to seek information in a structured manner
- The questions that need to be asked at this stage include:
 - how sensitive is the function to the expected variation in an input?
 - what are the likely variations in the inputs?
 - how will variations in the inputs affect the variation of the outputs?
 - how could the functions fail and what are the potential failure modes and effects?
- The answers to these questions generate information about the system and this knowledge is then helpful in:
 - concept generation and selection
 - assessing risk
 - determining trade-off studies

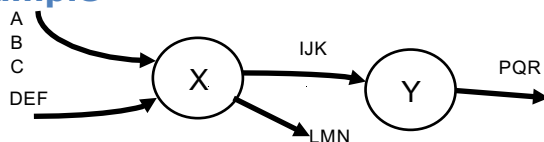


Flow Sensitivity Analysis

- What Is it?
 - Flow Sensitivity Analysis is a simple tabular method for assessing the relative effect of flow variation on the performance of a system
- Why do it?
 - To determine where there may be potential undesirable emergence and therefore design it out ab initio
- How to do it:
 - The approach is to form an input/output matrix for the lowest level FFDs:
 - identify and list, from the functional model, the input flows and their receiving functions
 - Identify and list, from the functional model, the output flows
 - Consider each input/output combination in the matrix and assess the effect of variation in the input flow on that function and output flow as judged on a simple three level scale:
 - insensitive  the input variation will not cause significant variation in the output
 - med. sensitive 
 - very sensitive  the input variation causes significant variation in the output

©2016 Waters Corporation 18

Example



List inputs
& receiving Functions

Input	Receiving Function	Output			
		IJK	LMN	PQR	
ABC	X	⊙	○		
DEF	X	△			
IJK	Y			△	

This cell is empty since there is no direct link between ABC and PQR

Record here the sensitivity of the input - output pair

©2016 Waters Corporation

19

Sensitivity Input-Output

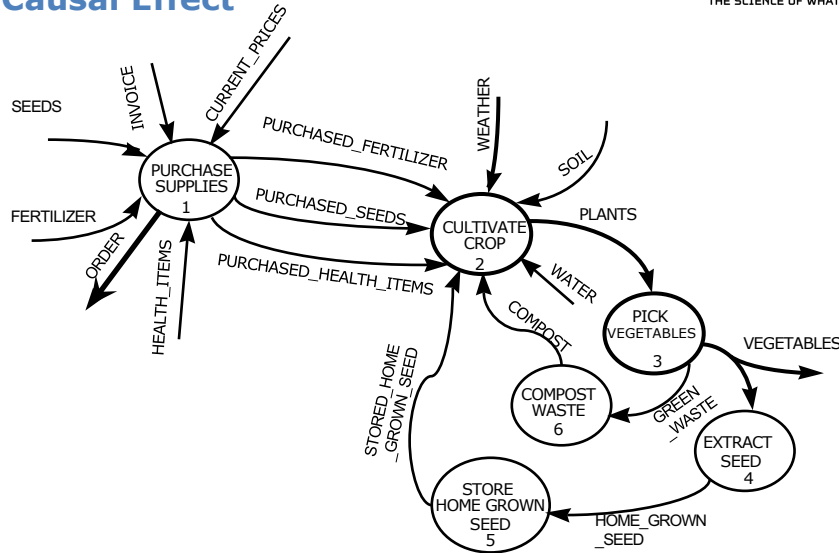
Input	Receiving Function	Output						
		PURCHASED_FERTILIZER	PURCHASED_SEEDS	PURCHASED_HEALTH_ITEMS	PLANTS	VEGETABLES	COMPOST	HOME_GROWN_SEEDS
SEEDS			△					
FERTILIZER		△						
HEALTH_ITEMS				△				
BUDGET		○	○	○				
CURRENT_PRICES		○	○	○				
PURCHASED_SEEDS					○			
PURCHASED_FERTILIZER					△			
PURCHASED_HEALTH_ITEMS					△			
WEATHER					⊙			
SOIL					△			
PLANTS						⊙		
VEGETABLES								⊙
COMPOST							○	
HOME_GROWN_SEED								○

©2016 Waters Corporation

20

Causal Effect

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

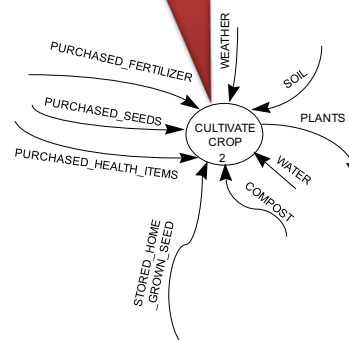
21

Function Sensitivity

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Flow sensitivity examines the effect of variation in flows on functionality. However, while this does indicate whole system effects we also need to consider the internal failure of the functions.
- Although the precise nature of the functional failure is unlikely to be known until its implementation is decided, it is still possible to make an assessment by considering functional failure modes, effects and causes
- Rather than 'invent' a new tool Failure Mode and Effects Analysis (FMEA) is ideal for conducting the necessary functional failure assessment

Assuming that the inputs are correct how can the function fail?



©2016 Waters Corporation

22

Functional Failure Modes and Effects Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTIONAL FMEA									
System: Washing Machine		1 - probability of occurrence		2 - Very rare to Frequent		Date: 11/01		No. 11/01	
Subsystem: Drum		3 - Severity of occurrence		4 - No Effect - No Worst Source		Author: J. Pender		Issue: 11/01	
		5 - Detectability of failure		6 - Cannot be Detected with Current Design		Checked: J. Pender			
FUNCTION	FUNCTIONAL FAILURE MODE	EFFECTS	5 CAUSES	6	DEFECTION Current Equipment Method	7	RPN	Design Suggestions/Comments	
LOAD DIRTY CLOTHES	No Load	No wash	3 User Error	2	None - no equipment for weight load detection	6	18	Include load sensor or other sensor for load detection	
	Over Load	Very poor wash	5 User Error	6	None - no equipment for weight load detection	30	150	Include load sensor or other sensor for load detection - will need to guard against weight sensor failure in the event sensor fails	
	Under Load	Poor Wash	4 User Error	4	None - no equipment for weight load detection	16	80	Include load sensor or other sensor for load detection	
	Intermittent Load (Hidden extreme mix of load)	Colour run	6 Items shielded by others	9	None - no Sufficiently placed sensors to detect this situation	54	480	Use of light sensing will not detect this situation and therefore system should attempt to alert the user	
		Fabric Shrink	7 Items shielded by others	9	None - no Sufficiently placed sensors to detect this situation	63	567	Use of light sensing will not detect this situation and therefore system should attempt to alert the user	
	Unintended Load (pet or object in items)	Injury/death of pet	8 User Error	2	None	16	160	Could include a test step button that disables any operation	
		Object damages items	7 User Error	3	None	21	126	For heavy objects the use of load cell may indicate the presence	
		Object damages machine	8 User Error	2	None	16	160		

The FMEA example above shows the failure modes for the LOAD DIRTY ITEMS function of a Washing Machine

- What is it?
 - A systematic approach to identifying, documenting, and prioritising potential functional failure modes, effects and causes
- Why do it?
 - Helps identify additional functionality or features ab initio that will make the system more robust in the hands of the customer
- How to do it?
 - Functional FMEA works best when drawing upon the expertise and knowledge of a multi-disciplinary team
 - The team should consider the system functions and how they might fail, identifying potential:
 - Failure mode
 - Effects of failure
 - Causes of failure
 - Ratings are assigned to the probability of occurrence, severity of an occurrence and probability of detection, and these are then used to determine an overall risk value

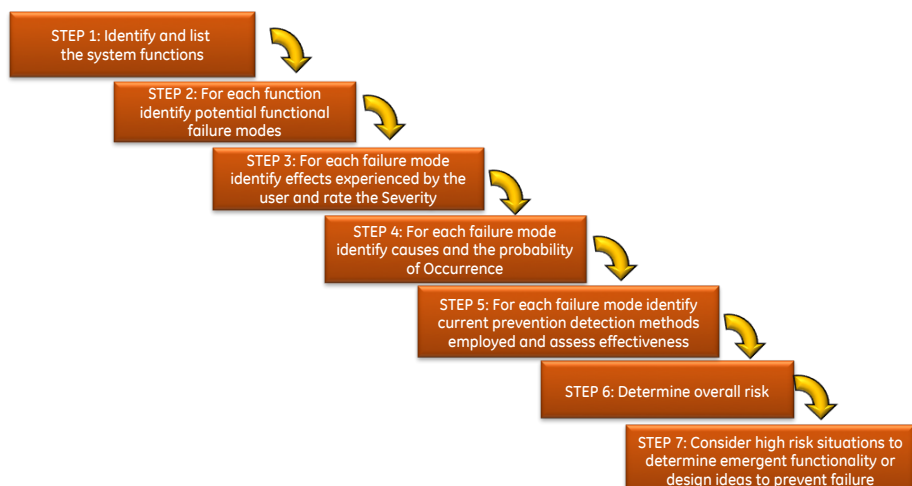
©2016 Waters Corporation

23

Functional FMEA - How to do it

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

24

The Functional FMEA Template

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTIONAL FMEA									
System:		O – probability of Occurrence		1: Very rare → 10: Frequent		Date:		No Issue	
Subsystem:		S – Severity of occurrence		1: No Effect → 10: Most Severe		Author:			
Element:		D – probability of Detection		1: Certain to Detect → 10: Cannot Detect		Checked:			
FUNCTION	FUNCTIONAL FAILURE MODE	EFFECTS	S	CAUSES	O	DETECTION/PREVENTION		RPN	Design Suggestions/Comments
						Current Employed Method	D		

©2016 Waters Corporation

25

Failings of FMEA

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Define what you understand by Failure?

FMEAs often fail to deliver because people apply their own understanding of what is meant by failure. It is an everyday word that is often associated with schooling and education, poor services as well as equipment breakdowns.

In the serious world of engineering we have to be precise – people's lives depend on it!

©2016 Waters Corporation

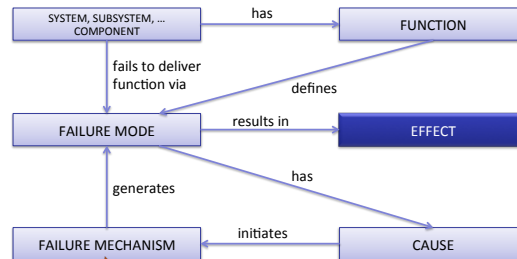
26

A Systems View of Failure

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- **FAILURE:** The termination of the ability of an item to perform a required function. (BS4778-3.1:1991)
- **FUNCTION:** what the item does, its purpose
- **FAILURE MODE:** the manner in which the item fails to meet its intended function. (SAE J1739 JAN2009)
- **EFFECT:** The consequence(s) a failure mode has on the operation, function, or status of the highest indenture level. This is expressed as what the CUSTOMER/USER might experience as a result of the failure mode. [MIL Std 1629A]
- **CAUSE:** the underlying event that leads to a failure mode, it initiates the failure mechanism.
- **FAILURE MECHANISM:** The physical, chemical or other process that results in failure. (BS4778-3.1:1991)



Most FMEA templates do not include a column to capture **Failure Mechanism** and in consequence when users identify a failure mechanism they capture it as the Failure Mode or Cause.

This leads to an inconsistent use of the tool, confusion and frustration – p.s. even the standards get it wrong!

©2016 Waters Corporation

27

The Functional FMEA Form

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTIONAL FMEA									
System:		O – probability of Occurrence		1: Very rare → 10: Frequent		Date:		No Issue	
Subsystem:		S – Severity of occurrence		1: No Effect → 10: Most Severe		Author:			
Element:		D – probability of Detection		1: Certain to Detect → 10: Cannot Detect		Checked:			
FUNCTION	FUNCTIONAL FAILURE MODE	EFFECTS	S	CAUSES	O	DETECTION/PREVENTION		RPN	Design Suggestions/Comments
						Current Employed Method	D		
SYSTEM ELEMENT	has	FUNCTION							
	fails to deliver function via	defines							
		results in							
	generates	has							
		initiates							

There is NO column for the MECHANISM

But teams identify the mechanism (it's important) but don't know where to put it! So ANYWHERE!

©2016 Waters Corporation

28

The Functional FMEA Form

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTIONAL FMEA									
System:		O – probability of Occurrence		1: Very rare → 10: Frequent		Date:		No Issue	
Subsystem:		S – Severity of occurrence		1: No Effect → 10: Most Severe		Author:			
Element:		D – probability of Detection		1: Certain to Detect → 10: Cannot Detect		Checked:			
FUNCTION	FUNCTIONAL FAILURE MODE	EFFECTS	S	CAUSES	O	DETECTION/PREVENTION		RPN	Design Suggestions/Comments
						Current Employed Method	D		

Use this column to record the system functions

Use this column to record the functional failure mode

Use this column to record the symptoms of the failure experienced by the customer

Use this columns to rate the Severity of occurrence

Use this column to record the potential and actual causes of the failure

Use this columns to rate the probability of Occurrence

Use this column to record the currently employed detection method

Use this columns to rate the probability of Detection

Use this column to record Risk Priority Number RPN

It is the comments column that is the important output since it flags design ideas/issues

©2016 Waters Corporation

Functions

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The first column of a FFMEA contains the system functions
- These must be described as a verb-noun phrase – i.e. they define something the system has to do – an action on an object
- In the ideal world every function of a system should be subject to a FFMEA – but in the real world this is unlikely because of time or resource constraints
 - In such instances it is important to identify the critical system functions by one means or another

©2016 Waters Corporation

Functional Failure Mode

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- A Functional FMEA examines the functions and the failure modes must be expressed in this context
 - In fact all Failure Modes must be defined relative to a function since it is central to the definition of failure
 - Failure modes are used in preference to failures, since a given failure may have several failure modes.
- Failure modes should be written in the context of the verb description, as “anti-functions”
- There are only 5 types of failure modes – the “usual suspects”:
 - Over, Under, No, Intermittent, Unintended
 - Example:
 - Wash Clothes
 - Over wash
 - Under wash (wash cycle completed and wash cycle not complete)
 - No wash
 - Intermittent wash (sometimes washes sometimes doesn't)
 - Unintended wash (oops it's the cat!)
 - Note: for a specific function not every type of failure mode will apply and there may be several interpretations of one type

The key to Functional Failure Modes is to put the function's verb after the usual suspects and then interpret for the particular situation

©2016 Waters Corporation

31

Effects

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The EFFECT of a potential failure mode is what the USER of the system might actually experience as a result of that failure mode
- User dissatisfaction is not an effect – all the effects will result in dissatisfaction – the purpose of the effects column is to record why the user is dissatisfied
- It is important to recognise here that for any potential functional failure mode there can be several possible effects experienced by the user of the system. Each of these should be recorded. For example the “wash items” function has the potential functional failure mode of “over-wash” this could have several effects including:
 - shrinkage
 - colour run
 - colour fade
 - physical fibre damage
- Note that different failure modes could have the same effect
- Other stakeholders may also experience effects as a consequence of the failure mode – these are not recorded on the FFMEA – if necessary a FFMEA can be performed for another stakeholder

©2016 Waters Corporation

32

Severity S

- Severity relates to the effect experienced by the user/customer (i.e. to the description given in the “effects” column). It is a measure of how dissatisfied they would be if they experienced the effect
- Severity is assessed on a simple 1 to 10 scale



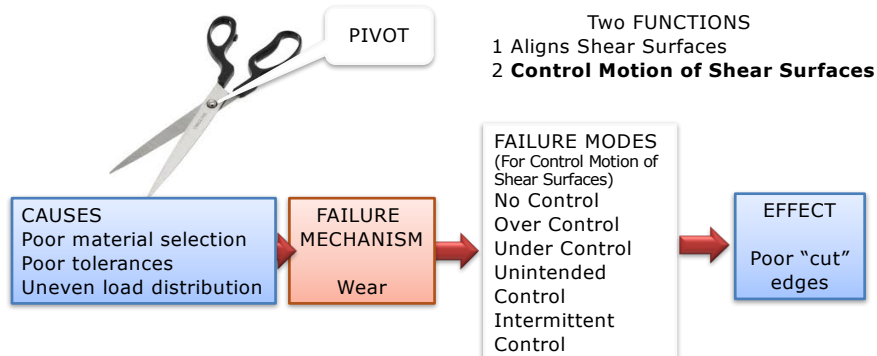
- Severity needs to be “calibrated” against the context and user

Causes

- Causes should be limited to design concerns i.e. we assume that the system is manufactured/installed correctly
- Analysis must stay within the defined scope (applicable system and interfaces to adjacent systems)
- There is usually more than one cause of failure for each failure mode
- Causes must be identified for a failure mode, not an individual effect. Causes may be repeated if a failure modes has several effects
- Causes can be either potential or actual. FFMEA is about stopping people making the same mistake as well as potentially new mistakes
 - It is therefore important to have the right people in the room
 - FFMEA can be used to capture “corporate knowledge”
- As part of the cause description we should also capture the Failure Mechanism

Failure Mechanism

- Most FMEA forms, including the one presented here, don't have a column for the failure mechanism
- This is indeed unfortunate because in completing FMEAs is a common error to confuse failure modes, causes and failure mechanisms!



We should (we MUST) identify the complete Cause -> Mechanism -> Failure mode -> Effect chain

Handling Failure Mechanisms

- Failure Mechanism is the physical, chemical or other process that results in failure. (BS4778-3.1:1991)
- Most FMEA Templates do not have a column for the mechanism – but as part of our analysis we identify it!
- We capture the MECHANISM as part of the cause description
- Typical Failure Mechanisms
 - Products
 - Corrosion
 - Fretting
 - Creep
 - Fatigue
 - Processes
 - Operator Fatigue
 - Complacency
 - Fear
 - Misinformation

Occurrence O

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Occurrence relates to the likelihood that a particular cause will occur leading to the failure mode and ultimately the effect
- Occurrence is assessed on a simple 1 to 10 scale

Very rare 1  10 Frequent

OCCURRENCE RATINGS			
Rating	Description	Criteria	Probability of Failure
1	Remote Occurrence	Failure highly unlikely to occur. History of similar design satisfactory with no failures	1 in 1,500,000
2	Very Slight Occurrence	Failure very unlikely to occur. History of very few failures	0.0000067
3	Slight Occurrence	Very few and infrequent failures	0.000067
4	Low Occurrence	Few and infrequent failures may be expected	0.0005
5	Medium Occurrence	Some failures likely, but not in major proportions	0.0025
6	Regular Occurrence	Regular failures may be expected	0.0125
7	Moderately High Occurrence	Frequent failures may be expected	0.05
8	High Occurrence	Failure of major proportions may be expected	0.125
9	Very High Occurrence	Frequent failures. History of similar design issues	0.33
10	Extremely High Occurrence	Constant failure	0.5

©2016 Waters Corporation

37

Detection/Prevention Currently Employed Method

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- On a Functional FMEA “Detection/Prevention” relates to what we did **last time**.
 - There is an implicit assumption that we have designed a similar system before and may, therefore, have experienced the failure mode.
 - In which case, it is possible some detection or prevention method has been incorporated in the previous design.
- The purpose of the Detection/Prevention column is to identify and record this and assess how effective it was
- If there was none – write none. This often happens if it is an unprecedented function or system

©2016 Waters Corporation

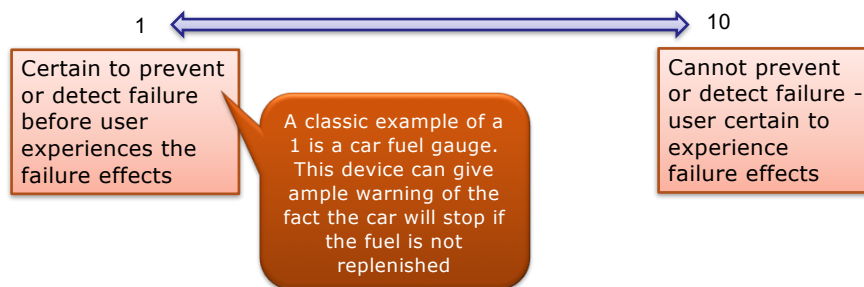
38

Prevention/Detection D

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The “D” column is an assessment of “how good” the currently employed Prevention/Detection method is at preventing or providing a warning of the impending failure
- The 1 to 10 Detection scale is based upon whether or not the user will experience the effects of the failure mode



©2016 Waters Corporation

39

Detection Ratings

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

DETECTION RATINGS		
Rating	Description	Criteria
1	Remote probability of failure reaching the user	A known detection method exists for similar systems with a strong history of excellent performance and is included in the system requirements
2	Very Slight probability the failure mode will reach the user	A known detection method exists for similar systems and is included in the system requirements
3	Slight probability the failure mode will reach the user	A known and robust detection method exists for other systems and is included in the system requirements
4	Low probability the failure mode will reach the user	A known detection method exists for other systems and is included in the system requirements
5	Medium probability the failure mode will reach the user	A known and robust detection method exists for other systems but is not included in the system requirements
6	Moderately probability the failure mode will reach the user	A known method exists for other systems but is not included in the system requirements
7	Moderately High probability the failure mode will reach the user	An experimental detection method exists for other systems and is included in the system requirements
8	High probability the failure mode will reach the user	An experimental detection method exists for other systems but is not included in the system requirements
9	Very High probability the failure mode will reach the user	No known detection method but one is included in the system requirements
10	It is certain that the failure mode will reach the user	No known detection method and no requirement to do so specified

©2016 Waters Corporation

40

Risk Priority Number RPN

- The values assigned to S, O and D are now multiplied together to give a Risk Priority Number:

$$S \times O \times D = RPN$$

- The purpose of the RPN is to direct action where it is required
- Care must be exercised, since the S, O and D ratings are not absolute but relative! Different teams constructing the same FMEA will typically generate the same failure modes, effects and causes - but will assign different values to SOD - BUT the prioritisation is likely be the same

Design Suggestions/ Comments

- The final column of a Functional FMEA is that allocated for fixes!
- Although the system has yet to be designed per se, knowledge of potential failures can lead to design ideas which if built in ab initio **will be minimum cost**
- It is important to remember why we are conducting the FFMEA
 - to identify new functionality/ideas for either eliminating the potential failure mode or detecting it. Accordingly there are levels of intervention:
 - Design the failure mode out completely
 - Provide a detection method that warns the user of impending failure
 - e.g. a fuel gauge on a motor car warns when the car is about to fail (run out of fuel)
 - Inform the user the failure has occurred

Example

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTIONAL FMEA									
System: Washing Machine		O – probability of Occurrence		1: Very rare → 10: Frequent		Date: 1/1/01		No 123	
Subsystem:		S – Severity of occurrence		1: No Effect → 10: Most Severe		Author: S. Powder		Issue 1	
Element:		D – probability of Detection		1: Certain to Detect → 10: Cannot Detect		Checked: A. Spinner			
FUNCTION	FUNCTIONAL FAILURE MODE	EFFECTS	S	CAUSES	O	DETECTION		RPN	Design Suggestions/Comments
						Current Employed Method	D		
LOAD DIRTY CLOTHES	No Load	No wash	3	User Error	2	None – but requirement for weigh load function identified	9	54	Include load cell or other sensor for load detection
	Over Load	Very poor wash	5	User Error	6	None – but requirement for weigh load function identified	9	270	Include load cell or other sensor for load detection – will need to guard against wet towels or similar giving false-positive type signal
	Under Load	Poor Wash	4	User Error	4	None – but requirement for weigh load function identified	9	144	Include load cell or other sensor for load detection
	Intermittent Load (Hidden extreme mix of load)	Colour run	6	Items shielded by others	9	None – but functionality identified for detecting mixed loads	9	486	Line of sight sensing will not detect this situation and therefore system solution should attempt to avert this
		Fabric Shrink	7	Items shielded by others	9	None – but functionality identified for detecting mixed loads	9	567	
	Unintended Load (pet or object in items)	Injury/death of pet	8	User Error	2	None	10	160	Could include a fast stop button that overrides any interlocks
		Object damages items	7	User error	3	None	10	210	For heavy objects the use of load cell may indicate the presence
		Object damages machine	8	User Error	2	None	10	160	

It is the comments column that is the important output since it is used to capture design ideas that can be built into the systems *ab initio*

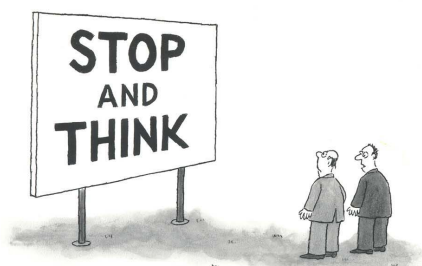
©2016 Waters Corporation

43

Exercise – FFMEA

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



"It sort of makes you stop and think, doesn't it."

- In teams
 - Conduct a functional FMEA for a particular function
 - List all the Failure Modes
 - Select one Failure Mode and list all the Effects
 - Select one Effect and list all the Causes
 - Select one Cause and list the Current Detection Methods (guess)
 - Assign S, O, and Ds as you proceed
 - Determine some suitable design ideas
 - Flip chart the outcomes and be prepared to share with the rest of the group
 - Time: 30 Minutes

©2016 Waters Corporation

44

THE SCIENCE OF WHAT'S POSSIBLE®

- Operational Functionality: the functionality necessary for a system to deliver its operational requirement
Emergent Functionality: the functionality of a system that addresses emergent behaviour

Operational Functionality: the functionality necessary for a system to deliver its operational requirement
Emergent Functionality: the functionality of a system that addresses emergent behaviour

©2016 Waters Corporation

45

THE SCIENCE OF WHAT'S POSSIBLE.®

- [illegible]

[illegible]

©2016 Waters Corporation

46

Functional FMEA

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Pros
 - Simple to use
 - Leads to shared and common understanding
 - Identifies potential problems early so that they can be designed out
 - Identifies potential HAZARDS before design commences and therefore provides information to influence concept generation and selection
 - Since for a given Capability, the underlying functionality will not change over time, the FFMEAs will have aspects that will also not change:
 - Functions
 - Function Failure Modes
 - Effects
 - We can therefore reuse which will reduce the analysis effort in the longer term
- Cons
 - Often left until it's too late – once the design has been done
 - Can become tedious so people don't think
 - Since no design has been undertaken, causes may not be root causes. Hence there is a need to undertake Design FMEA
 - Difficulty in identifying causes

©2016 Waters Corporation

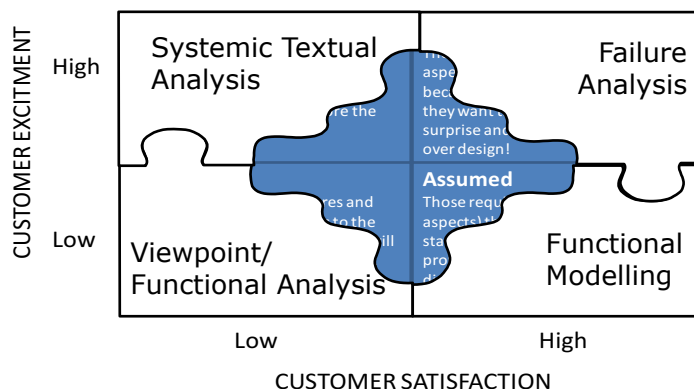
47

Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Functional FMEA and Flow Sensitivity are simple tools that use the knowledge and experience of the team to identify potentially sensitive functions and flows that could result in undesirable emergent behaviour – they identify the critical functions and inputs



©2016 Waters Corporation

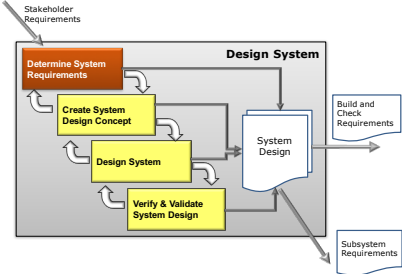
48



Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Analyse Requirements

Verifying Customer – Technical Requirements using Quality Function Deployment



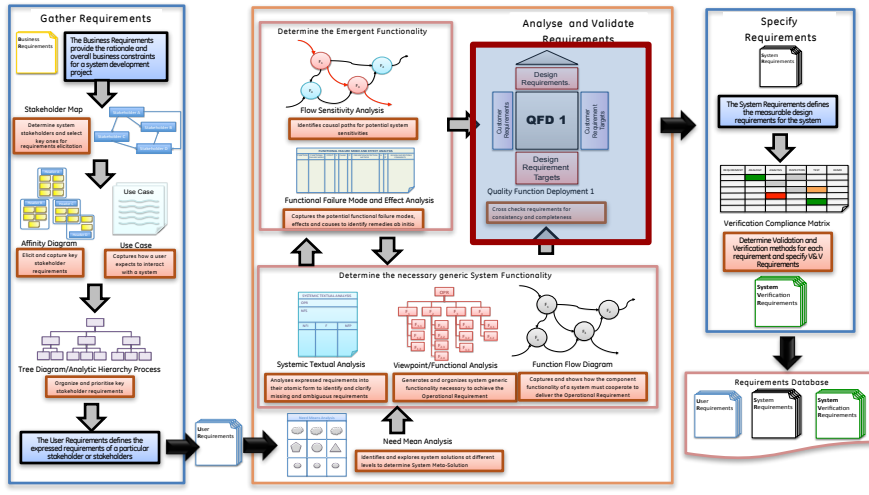
This Session:
To describe and practise Quality Function Deployment as a tool that can be used to help in the Determine System Requirements process to:

- Check for completeness and consistency
- Summarise the requirements set
- Help define requirements
- Provide a framework for ancillary analyses
- Provide a reference for future systems

©2016 Waters Corporation 1

Process and Tools

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

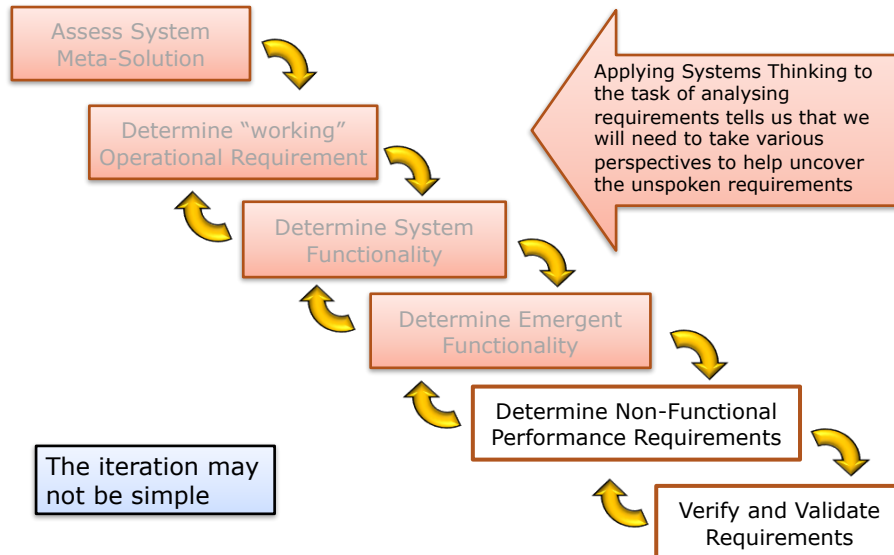


©2016 Waters Corporation 2

Where are we?: Analyse Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

3

A QFD Success: The Neptune Payphone

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Outdoor Payphones



Neptune 1000 Interactive Kiosk



MARCONI'S WEB PHONE AND INTERACTIVE KIOSK WIN TOP INDUSTRY AWARDS

KioskCom 2003 Interactive Kiosk Excellence Awards recognise excellence of Marconi's Neptune 800 and 1000

London and Las Vegas — 10 April 2003 —

At a ceremony held in Las Vegas on 8 April 2003, Marconi's (MONI) Neptune web phones and interactive kiosks won two categories in the annual KioskCom Interactive Kiosk Excellence Awards. The Awards, widely seen as the most prestigious in the kiosk industry, recognise achievements in the design and application of interactive kiosks. Marconi won the "Best Public Communications Kiosk Application" award for the nationwide deployment of its Neptune 800 web phone and Neptune 1000 interactive kiosk in the UK, the world's largest roll-out of its kind. In addition, the Neptune 800 won the "Most Innovative/Creative Kiosk Enclosure Design" award, with the Neptune 1000 finishing a close second.

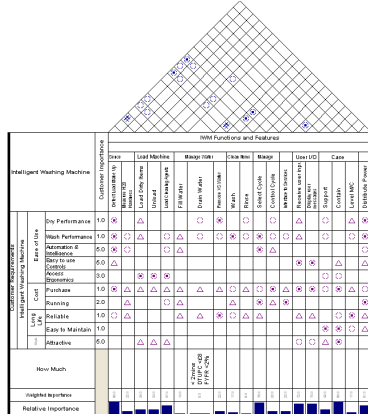
©2016 Waters Corporation

4

Quality Function Deployment

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



The example above is simplified Quality Function Deployment phase 1 matrix for the Intelligent Washing Machine. It shows the relationship between the User Requirements and the System Design Requirements to demonstrate completeness and consistency

- What is it?
 - QFD is more than a tool – potentially it's a complete methodology that can help an organization capture, translate, summarise and manage requirements throughout the system introduction process
 - It is a structured and systematic approach to flowing customer requirements into measurable organization requirements at each stage from market research through design to realisation.
 - The QFD chart is basically a type of matrix diagram that allows a team to assess the relationships between two sets of requirements thereby helping assure:
 - completeness
 - consistency
- When to use it
 - Whenever attempting to determine and analyse a set of requirements in order to deliver a customer centric solution.

©2016 Waters Corporation

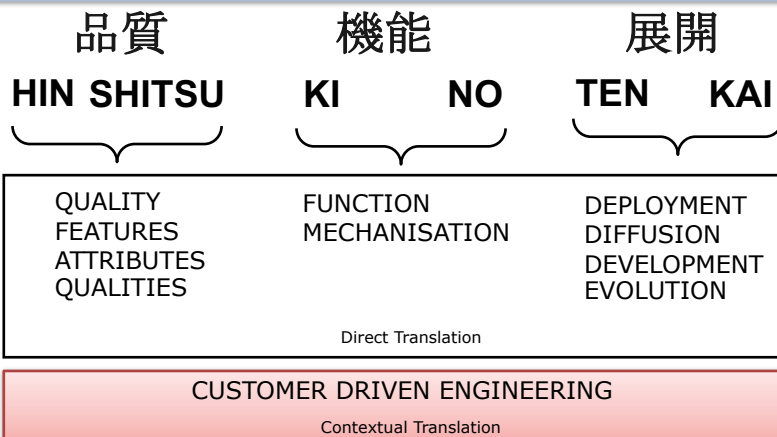
5

Lost in Translation

Waters

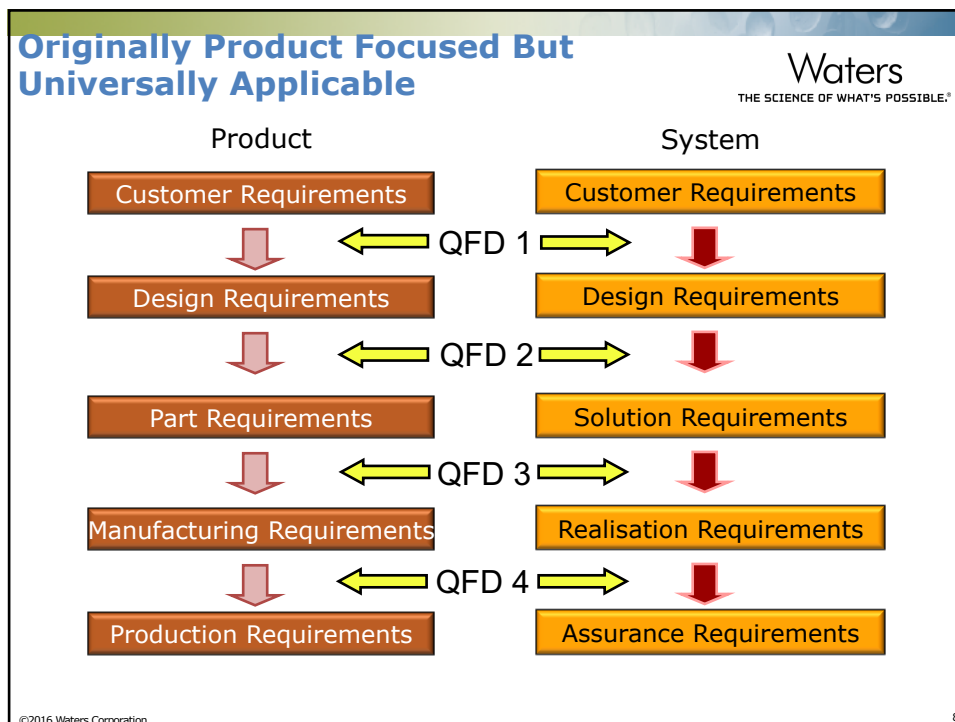
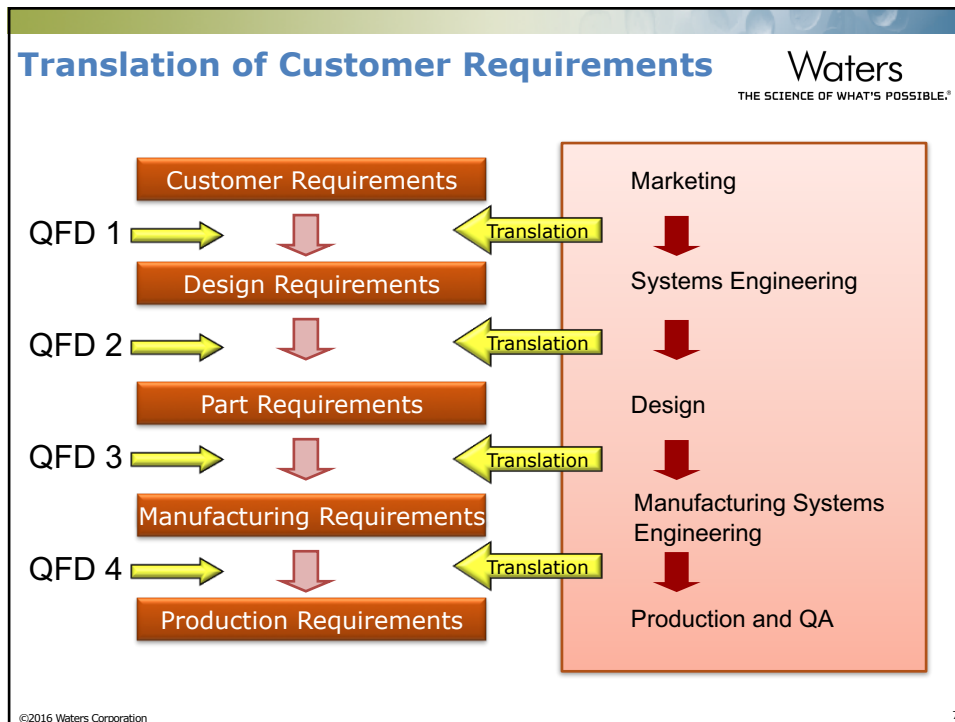
THE SCIENCE OF WHAT'S POSSIBLE.®

Created in Japan in the late 1960s' and first used in a ship yard (low volume production), Quality Function Deployment (QFD) is a structured approach for translating customer requirements into company requirements at each stage from market research through to manufacture.



©2016 Waters Corporation

6



Translation via a Matrix Diagram

A **Matrix Diagram** is a Systems Thinking tool to identify, quantify and capture the many-to-many relationships that exist between two list of items

Secondary List of Items (must have something in common)

Primary List of Items (must have something in common)

		List 2				
		Item A	Item B	Item C	Item D	Item E
List 1	Item 1	⊙	⊙		△	
	Item 2					
	Item 3			○	△	
	Item 4		○	△		
	Item 5	△		△		○
	Item 6		○		⊙	

Many-to-many Relationship Matrix

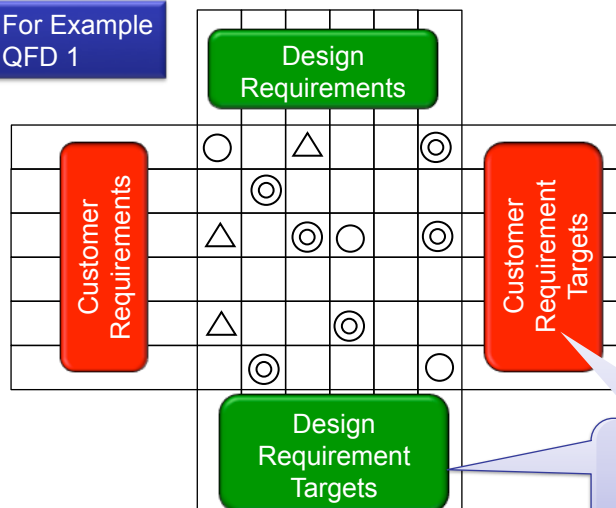
⊙ STRONG
○ MEDIUM
△ WEAK

©2016 Waters Corporation

9

In QFD the Lists are Requirements

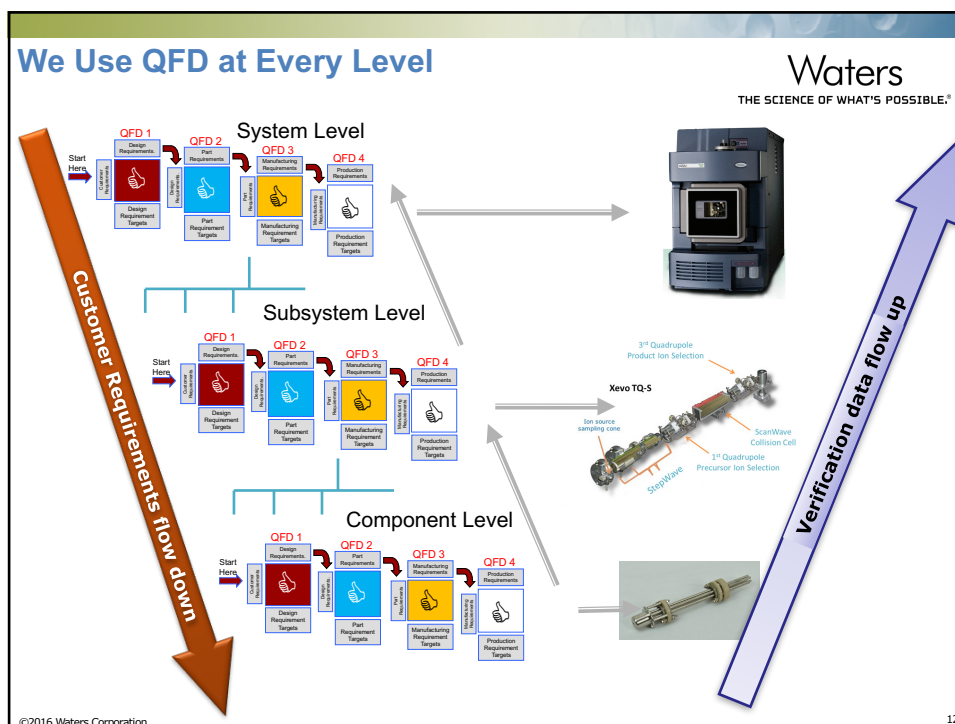
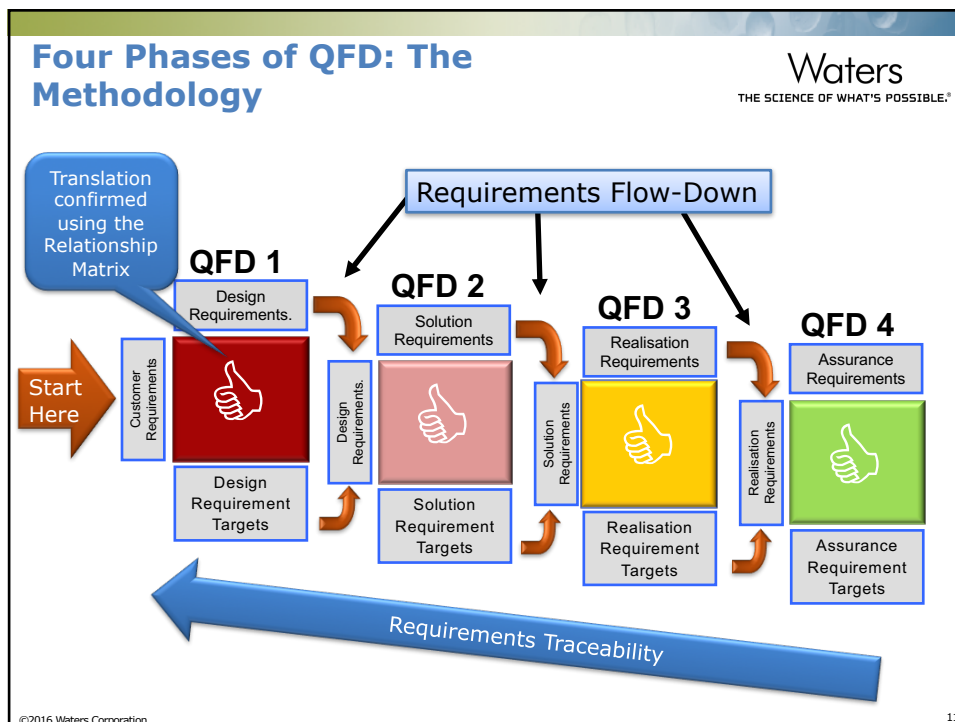
For Example
QFD 1

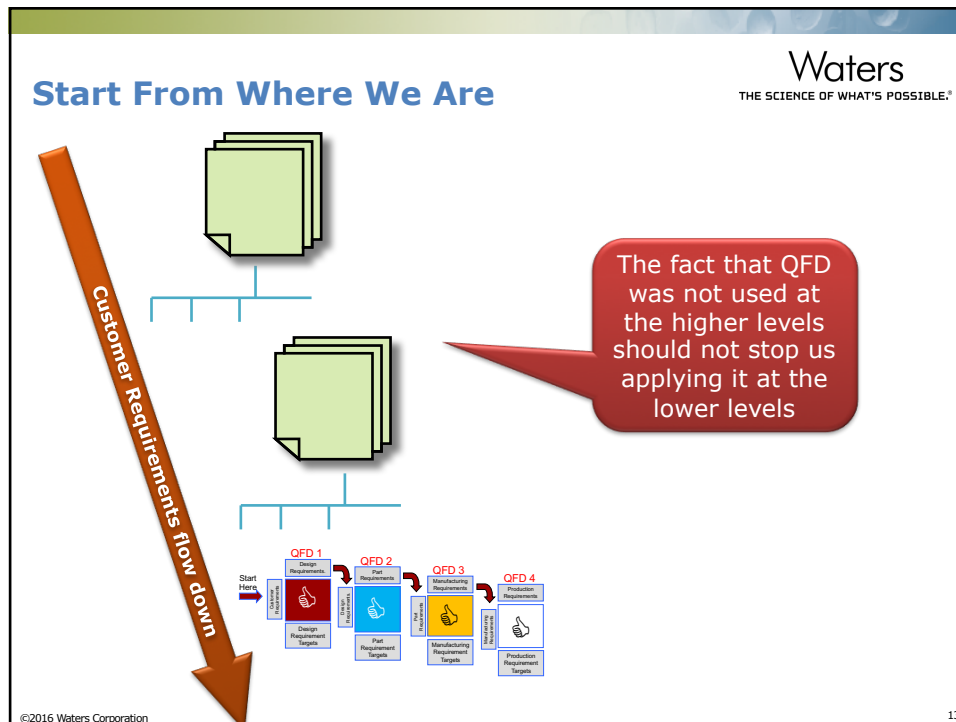


QFD often contains further lists of associated requirements

©2016 Waters Corporation

10



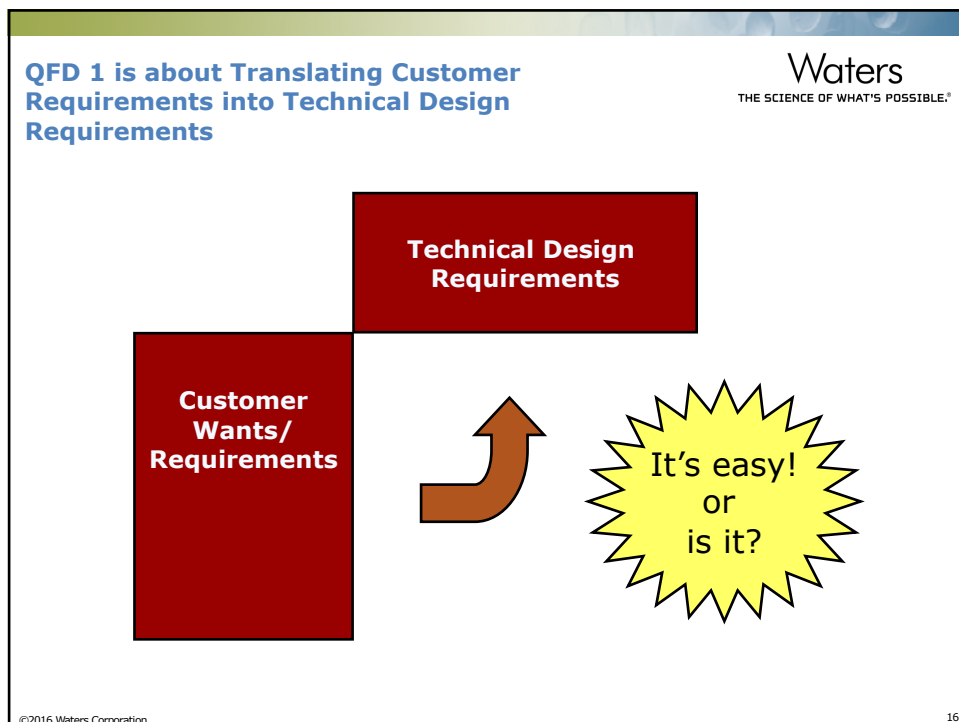
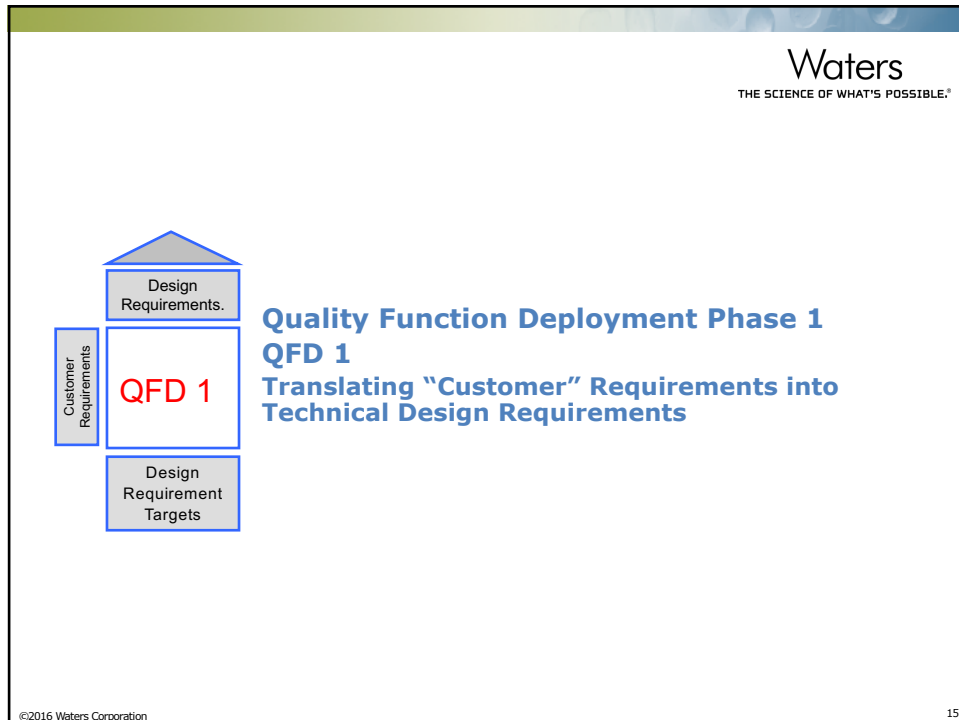


A Perfect Tool!

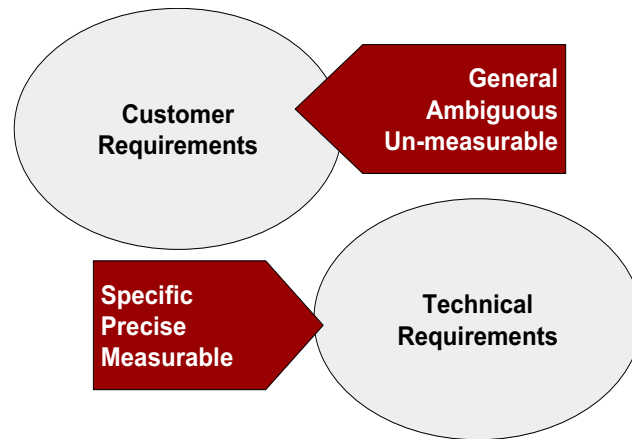
Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- At face value, QFD sounds an ideal tool for the Research/ Medical industry
 - Traceability
 - Low volume etc
- However, applying QFD in the aerospace/power industry can be difficult and frustrating because of the nature and origin of the customer requirements

©2016 Waters Corporation 14



The QFD Assumption

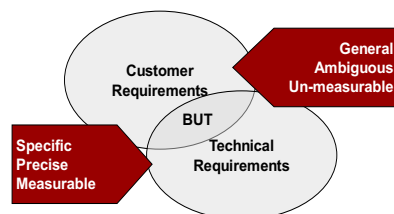


Text book QFD makes the implicit assumption that customer requirements are general, ambiguous and un-measurable.

Reality: Markets and Requirements

Broad

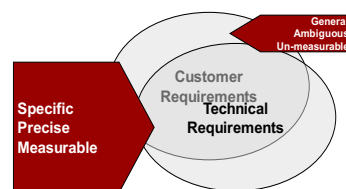
- Broad Market Requirements
- Typically the requirements are ambiguous and general in nature because the customer cannot express in technical terms their precise needs and expectations



It is often easier to apply QFD in a Broad Market, because the spoken or expressed customer requirements rarely contain any technical requirements (but there may be some)

Narrow

- Narrow Market Requirements
- The customer often has a very good knowledge of the system and is able to describe their requirements in considerable technical detail

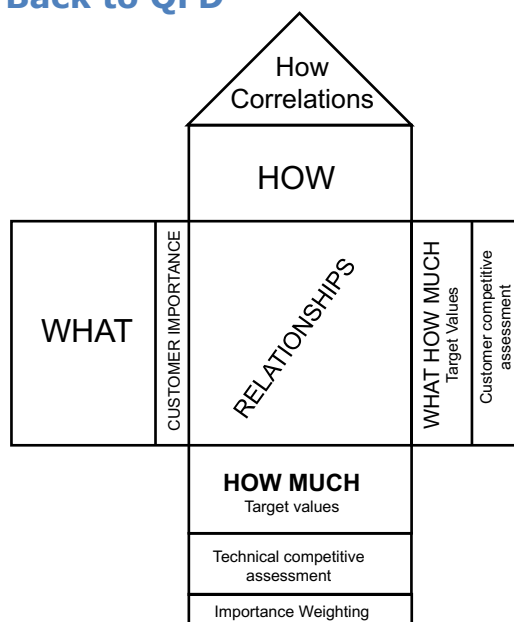


It is often more difficult to apply QFD in a Narrow Market, because the spoken customer requirements contain a large number of technical requirements

QFD in a Narrow Market

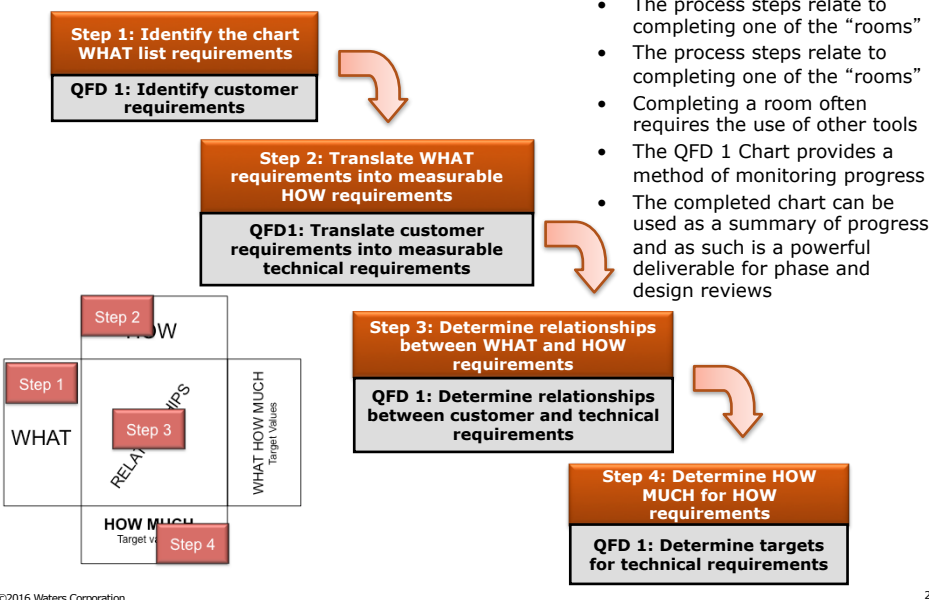
- QFD is often (if not always) explained or taught from a Broad Market perspective
- Certain Industries – Defence, Infrastructure, Research, Medical etc are Narrow Markets
- If a broad market perspective is used with a set of Narrow Market requirements the outcome is confusion, frustration and ultimately failure!
- This is due to the customer providing a considerable amount of technical detail in their requirements – which is logical since they are spending a great deal of money and therefore they want to be careful
- Applying QFD in this Narrow Market situation requires a slightly different approach

Back to QFD



- QFD charts look complex. They are complicated but not complex.
- The chart comprises a roof and a series of rooms that form a matrix diagram between the WHAT Requirements and the HOW Requirements
- The QFD method follows a set of steps to build up the rooms and roof
- The core process focuses on the WHAT, HOW, RELATIONSHIPS and HOW MUCH rooms

QFD Core Process



21

Step 1 in a Broad Market

- With Broad Markets it is necessary to “talk” to many customers and stakeholders and listen to their expressed customer requirements
- WHAT the customer wants is called “the voice of the customer”
- In a Broad Market situation the requirements that make up the WHATs are typically general, vague and un-measurable
- Capturing and recording these is highly desirable because it is “the voice of the customer” – but it is not directly actionable

WHAT

Easy to use
Pay anyway
Working
Clean

For example “easy to use” is an obvious user requirement for a payphone, but could have a variety of meanings for different people.

If questioned as to “how easy to use?” the response is usually “very” i.e. equally vague

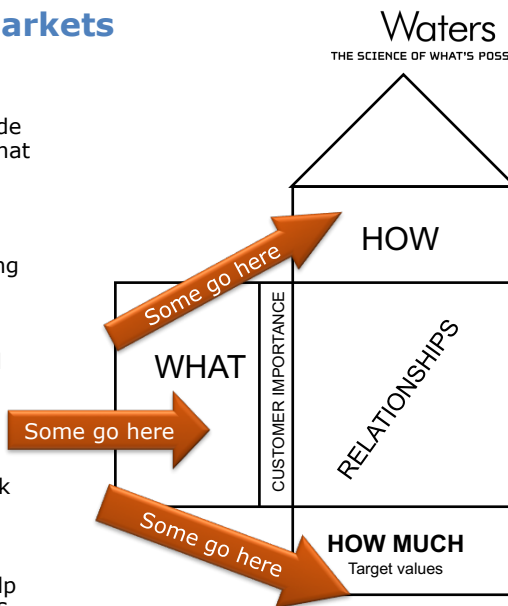
Note, however, while the customer cannot describe specifically what they mean they can and will assess how “easy to use” it is once the system is delivered

©2016 Waters Corporation

22

Step 1: in a Narrow Markets

- Narrow Market customers provide a detailed set of requirements that contain both vague, general requirements and specific technical requirements. Indeed, the customer will have put considerable effort into not giving any vague requirements
- Although these source requirements represent "what" the customer wants they should NOT all go in the WHATs!
- In such cases customers are providing some of the HOW and HOW MUCH requirements
- In consequence we have to work out where best to put each requirement in the QFD chart
- This can be time consuming but worthwhile because QFD will help us identify missing requirements



©2016 Waters Corporation

23

Discussion – Voice of the customer

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Consider the case study system
 - What are the primary requirements of customers/user?
 - What type of requirements are these?
- Time: 10 minutes

©2016 Waters Corporation

24

Customer WHATs and Narrow Market Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The requirements (WHATs) in a narrow market are the:
 - Operational Requirements (that element relating to the overall system purpose)
 - Non-functional System Requirements

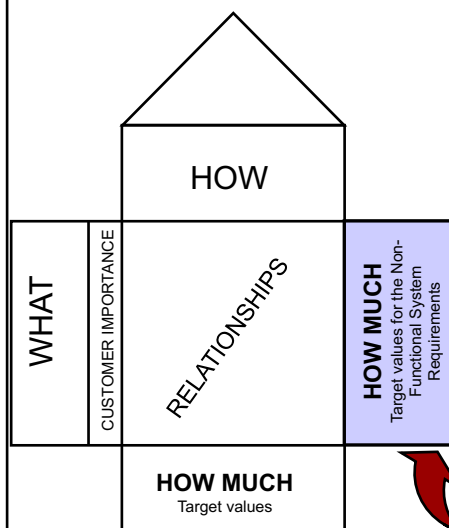
This is also true of a Broad Market, but in these markets customers naturally talk in these terms. We should be wary, however, of any “technical requirements” from the customer particularly if they are not at a system level but appear to refer to a specific aspect. Such requirements may sit better elsewhere in the QFD chart!

©2016 Waters Corporation

25

Non-functional System Requirement Target Values

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



- Narrow Market Non-functional System Requirements are often measurable (unlike Broad Markets). This is because it is important to the customer.
- Narrow Market Non-functional System Requirements will therefore have target values. This often leads to confusion since requirements with targets are often just categorised as “performance requirements”
 - it is important to distinguish between those “performance requirements that relate to a particular FUNCTION (NFP) and those that relate to the SYSTEM (NFS)
- To capture the target values for Non-functional System Requirements we modify the basic QFD chart to include an extra room

©2016 Waters Corporation

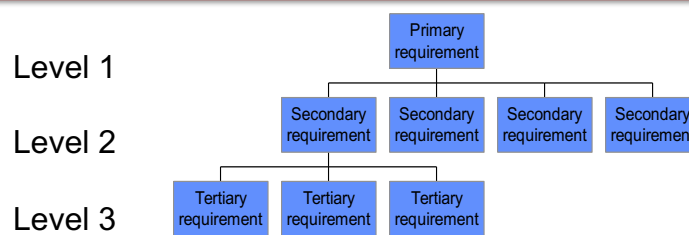
26

Structure the Voice

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

With both Broad and Narrow markets there may well be a significant number of WHATs requirements. In such cases it is important to structure the customer requirements into the general primary requirements and the more detailed secondary and tertiary requirements



In practice this can be accomplished by means of Affinity Diagramming and Tree Diagramming – see the section on Gathering Requirements

©2016 Waters Corporation

27

QFD and Multiple Customers/Stakeholders

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

One key advantage of QFD is its ability to consider simultaneously multiple customers/stakeholders and thereby develop a set of System Technical Design Requirements that provide universal satisfaction - failing that identify where trades may be necessary

		Design Requirements	Technical Design Requirements				
		Importance to Customer					
User Requirements	User	Easy to use	4				
		Multi-language	3				
		Any Payment	5				
		Clear Sound	3				
		Clean	5				
Operator Requirements	Operator	Durable	5				
		Reliable	4				
		Cleans easily	2				
		Maintainable	5				

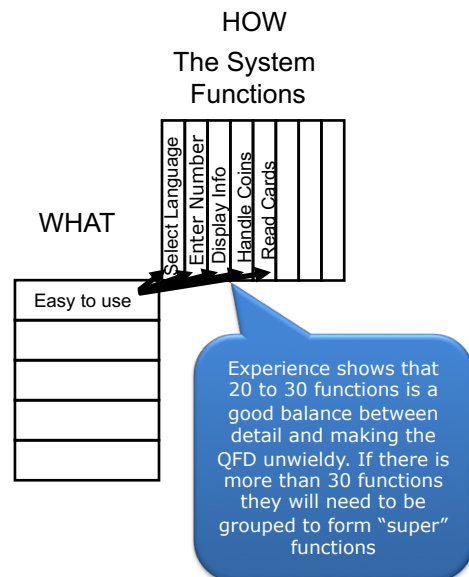
One of the advantages of QFD is that it can help determine a set of System Design Requirements that satisfy all stakeholders

©2016 Waters Corporation

28

Step 2: Translate into Technical Requirements

- The idea behind QFD is that the vague customer requirements should be translated into specific, precise and measurable technical design requirements
- This is HOW not in terms of solutions but HOW in terms of technical design requirements
- The HOWs must be measurable which allow the system to be evaluated once built
- This is can achieved through consideration of the necessary functionality



©2016 Waters Corporation

29

HOWs and Narrow market Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

The HOWs are the Functional Requirements

Warning:

Customer specified Non-functional Implementation (NFI) requirement present a particular challenge with QFD – where do we put them? The temptation is to put them in the HOWs.

Firstly, any NFI requirement should be tested to see if they are justifiable or lazy requirements, i.e. does the customer really want "door mirrors" or "to see behind"? Remember, customers think in objects not functions.

If lazy, and not already done, determine the function inferred by the solution-based requirement and write the function in the HOWs. The link to a solution can be captured in the Requirements Database or as a note on the QFD matrix

If justifiable, determine the function inferred by the solution-based requirement and write the function in the HOWs AND write the solution in the corresponding HOWMUCHs together with any specifications (see later)

©2016 Waters Corporation

30

Step3: Form WHAT-HOW Relationship Matrix

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

WHAT	HOW Functions					
	Function 1	Function 2				
Customer RQMT 1	⊙		○			△
Customer RQMT 2	△	⊙	○			
	○					○
	○	○	⊙			△
	△	⊙			⊙	

- ⊙ STRONG relationship
- MEDIUM relationship
- △ WEAK relationship

This method allows complex relationships to be depicted and easily interpreted

It also permits cross checking or requirements compliance - blank or sparse rows or columns indicate inadequate translation of WHATs into HOWs

For example, a blank or sparse row indicates that there is a customer requirement for which no function or feature has been identified

The matrix can also be used to “test” potential “delighters”

©2016 Waters Corporation

31

Payphone Example

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

		Design Requirements	Importance to Customer	Select Lang	Enter No.	Display Info	Handle Coins	Read Cards	TX/RX Aural Messages
		Customer Requirements							
User	Easy to use	4	○	⊗	○	△		○	
	Multi-language	3	⊗	△	⊗	○		△	
	Any Payment	5				⊗	⊗		
	Clear Sound	3						⊗	
	Clean	5		△		△		⊗	
Operator	Durable	5			○				
	Reliable	4		⊗	△	⊗	⊗		
	Cleans easily	2		△					△
	Maintainable	5		△		⊗	⊗		

Vague ambiguous customer requirements

Measurable technical interpretation through functionality

Relationships between customer requirements and technical interpretation

Sparse Rows! something is missing!

Vague ambiguous customer requirements

Measurable technical interpretation through functionality

Relationships between customer requirements and technical interpretation

Sparse Rows! something is missing!

©2016 Waters Corporation

32

Payphone Example

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

		Design Requirements	Importance to Customer	Customer Requirements	Select Lang	Enter No.	Display Info	Handle Coins	Read Cards	TX/RX Aural Messages	Contain & Protect
User	Easy to use	4	○	⊙	○	△			○	△	
	Multi-language	3	⊙	△	⊙	○			△		
	Any Payment	5				⊙	⊙			△	
	Clear Sound	3							⊙		
	Clean	5		△		△			⊙	⊙	
Operator	Durable	5			○					⊙	
	Reliable	4		⊙	△	⊙	⊙			○	
	Cleans easily	2		△					△	⊙	
	Maintainable	5		△		⊙	⊙			△	

What's missing is the box! It has functionality: it protects, supports, contains etc. Can either include as a set of functions or as a new feature to address the customer requirements

©2016 Waters Corporation

33

Relationship Guidance!

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Complete the relationship matrix row-wise – we are interested in delivering the “customer’s/stakeholder’s” requirements
- Relationships are NOT digital – it is easy to argue that the absence of a function will stop the system working and therefore it must be a strong relationship – We are concerned with how changing the “performance” of a particular function impacts on the customer/stakeholder requirement. One simple approach is to consider a “volume” knob that turns-up a particular function!
- Do not over argue relationships – consider first order effects
- In any row the function that has the biggest impact on the customer requirement should not be automatically categorised as “strong” – think carefully about its relationship!
- Typically the relationship matrix is 50% - 60% empty

©2016 Waters Corporation

34

Overly full Rows or Columns

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ ROWS

- It is common to find overly full rows in the QFD relationship matrix because the WHATs are “system” level requirements (NFS for the item we are considering)
- What can be important here is the distribution of Strong, Medium and Weak symbols.
 - A good guideline is the Pareto Principle or 80/20 rule

■ Columns

- Overly full columns on the other hand are an indication of:
 - Over-aggregation of lower level functionality
 - Overly full columns should be decomposed into two or three lower level functions and the relationships re-evaluated
 - Equally empty or sparse columns should be tested for over-decomposition
 - Critical Functions that have a significant impact on many of the customer facing requirements

©2016 Waters Corporation

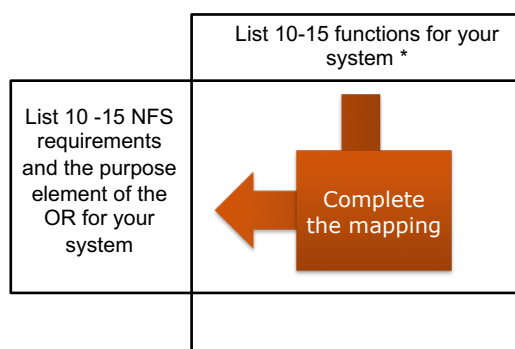
35

Exercise – QFD1 steps 1 to 3

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

* The full spectrum of functionality for your system should be captured



- In teams for the case study exercise
- Complete steps 1, 2 and 3 of a QFD1 chart
- Flip chart the outcomes and be prepared to share with the rest of the group
- Time 30 Minutes

©2016 Waters Corporation

36

So What

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- QFD 1 will tell us where there are missing requirements or requirements not needed!
- Typically the missing requirements relate to missing functionality, particularly that to do with the structure and architecture of the system functionality that is necessary to implement the operational functionality
- We should also examine the relationship matrix for "structures"
 - Full columns could indicate the presence of critical functions (or over aggregated lower level functionality)
 - Full rows indicate requirements that will need careful management across the system

©2016 Waters Corporation

37

What's Important?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Assign Customer Importance to WHATs on 1 to 5 scale

Densely populated columns can indicate critical functions – but we can be intelligent by taking into account the customer requirements and their importance

WHAT		HOW						
		Function 1	Function 2	Function 3				
Customer Rqmt 15	5	⊙	○					
Customer Rqmt 23	3			○			△	
Customer Rqmt 31	3	○	⊙				○	
	2	△	△					
	1	○	△			⊙		
	5		○			⊙		
Importance Rating		53	42	9	45	9	6	

△ = 1
○ = 3
⊙ = 9

The second column is fuller – but has a weight score of 42

$$5 \times 9 + 1 \times 3 + 2 \times 1 + 1 \times 3$$

HOW MUCH

©2016 Waters Corporation

38

Customer Importance Numbers

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- QFD charts often make use of Customer Importance numbers to assist in prioritizing requirements
- The WHAT IMPORTANCE RATING is determined from an assessment of the WHATS
- Each WHAT is assigned an importance rating typically expressed using a relative scale of 1 to 5 with higher numbers indicating a greater importance to the "customer" (stakeholder)
- It is important that these reflect the values of the stakeholder and NOT internal company beliefs; however, many organizations start from an internal assessment
- There are several ways to determine the Customer Importance numbers.
 - Absolute scales
 - Sales Driven Scales (order winners/qualifiers)
 - Analytic Hierarchy Process

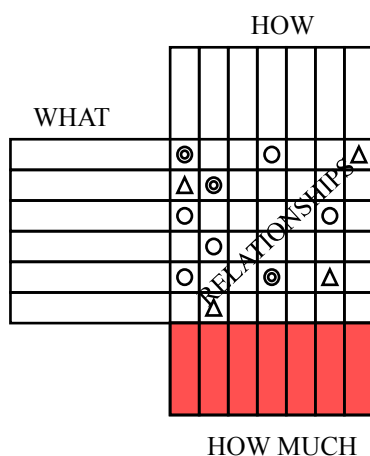
©2016 Waters Corporation

39

Step 4: Determine Targets and Ceilings HOW MUCH is Enough?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- The fourth element of the QFD chart is the HOW MUCHs
- These are measures or metrics for the HOWs and the associated target values that define how good the HOWs have to be in order to satisfy the customer
- The HOW MUCHs are the Non-functional Performance Requirements and "flowed-down" Non-functional System Requirements
- Empty cells indicate where we need to identify appropriate performance targets
- For any particular HOW function there will be many associated HOW MUCH metrics and target values

©2016 Waters Corporation

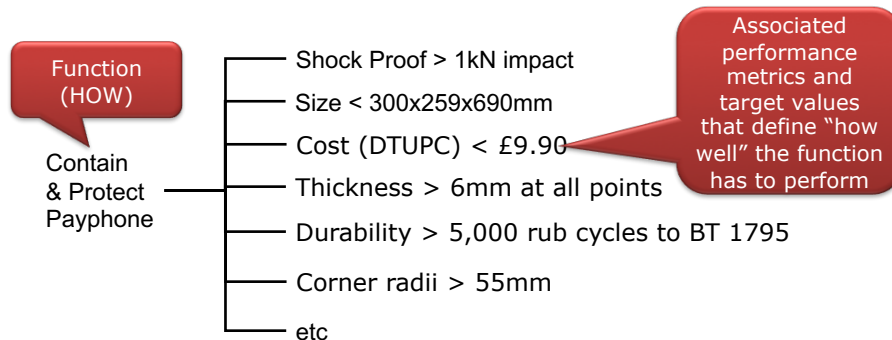
40

Example

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The “HOW MUCHs” complete the technical interpretation and comprise: A metric, measure or property and a value as a target, lower or upper limit.
- It is where most of the non-functional requirements align to the QFD structure.
- These can be very detailed and refer to other documents



©2016 Waters Corporation

41

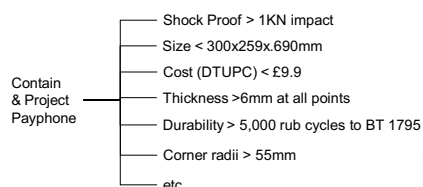
Deriving HOW MUCHs is Top-Down and Bottom-Up

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ Bottom-Up

- Common sense, experience, judgement says that the way you measure Function 4 is by ...



A common approach here is to consider two solutions to a particular function. Say A and B, and ask what metrics could be used to decide between them?

• Top-Down

WHAT	HOW						
	Function 1	Function 2	Function 3	Function 4	Function 5	Function 6	Function 7
RQMT A	⊗			○			△
RQMT B	△	⊗	○	△	△		
RQMT C	○					○	
RQMT D		○					
RQMT E	○			⊗		△	
RQMT F		△					

The HOW MUCH for Function 4 must contain metrics/measures that relate to Customer Requirements A, B and E – or an error has been made!

HOW MUCH

©2016 Waters Corporation

42

High Level NFS Flow Down

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Non-functional
System
Requirements
(System level
performance
requirements)

WHAT	HOW						WHAT HOW MUCH
	⊙	⊙	⊙	⊙	⊙	⊙	
Cost	Δ	⊙	⊙	Δ	Δ	⊙	£X
	⊙					⊙	
Reliability	⊙	⊙		⊙	Δ		MTBF Y
		Δ					
	Sub-cost £A	Sub-cost £B	Sub-cost £C	Sub-cost £D	...		

Can be "flowed
down" through the
relationship
symbols

The QFD matrix can also be helpful in "flowing-down" system level performance requirements to individual functional requirements – it can help set and manage budgets for critical system level requirements such as cost, weight, reliability etc.

$$X = A+B+C+D+ \dots$$

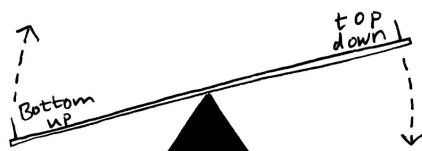
©2016 Waters Corporation

43

Exercise – QFD 1

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- In teams
- Choose a Function and identify suitable HOW MUCH metrics or measures (cannot determine targets values)
- Use both approaches described:
 - Bottom up: given two possible solutions A and B, what metrics would you use to decide between them?
 - Top down: Using the relationship symbols, is a related customer requirement addressed by the "bottom metrics"? Is the relationship symbol selected appropriate?
- Time 15 Minutes

©2016 Waters Corporation

44

Terminology

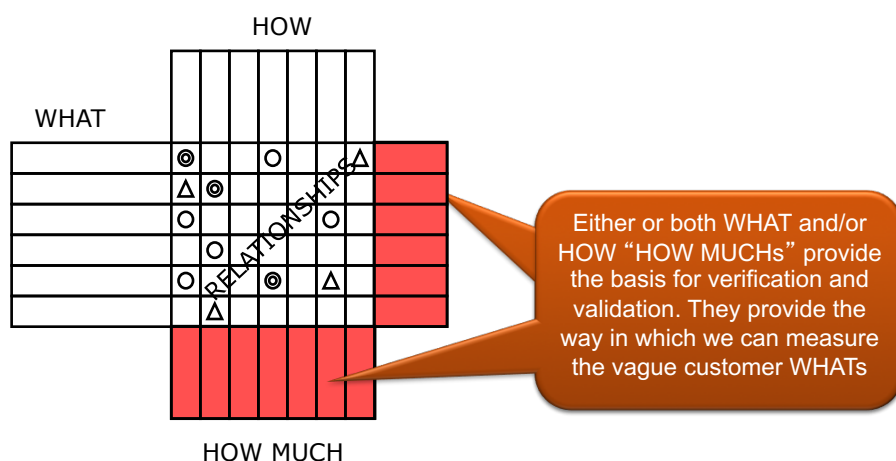
- QFD, and the Holistic Requirements Model, helps us define what a Technical Requirement is:

$$\text{Function} + \begin{cases} \text{Measure/property + value} \\ \text{Measure/property + value} \\ \text{Measure/property + value} \\ \text{etc.} \end{cases}$$

- Technical Requirement is QFD terminology. The equivalent system engineering terminology is a System Design Requirement

System Design Requirement $\equiv$ Technical Requirement

V&V through HOW MUCHs



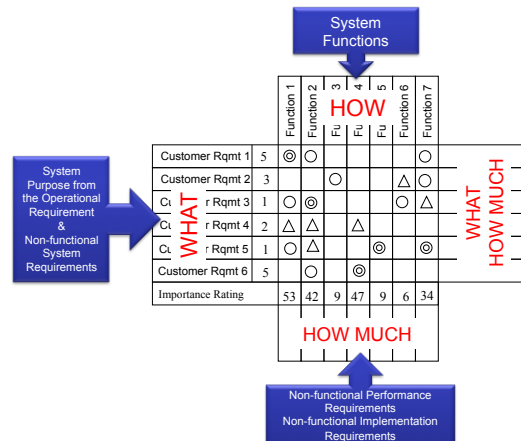
Holistic Requirements Model and QFD 1

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Note that the structure of the Holistic Requirements Model is reflected in the structure of the QFD. We are exploiting this structure to consolidate and verify the requirements set. Blank or sparse cells, columns or rows indicate where the requirements set is incomplete

Note that we DO NOT have to have a complete QFD 1 chart to proceed. There may well be elements of the HOW MUCHs to be decided (TBD) that can only be determined through design activities

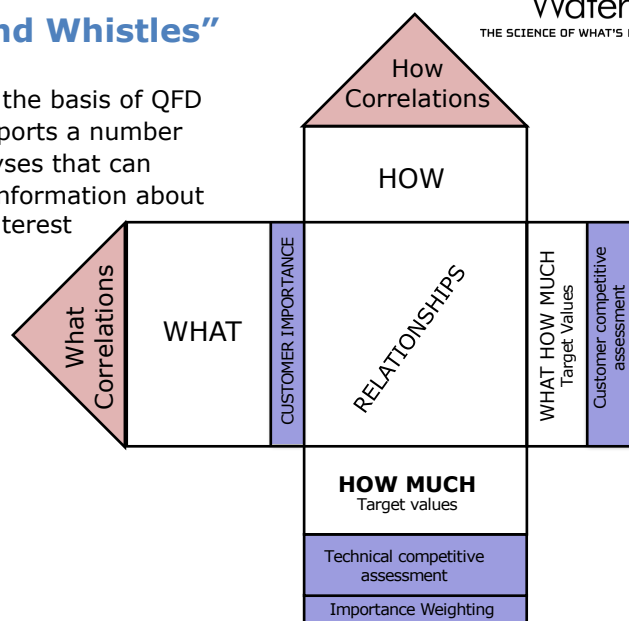


©2016 Waters Corporation

47

The "Bells and Whistles"

- Steps 1 to 4 are the basis of QFD
 - But it also supports a number of ancillary analyses that can provide further information about the System of Interest



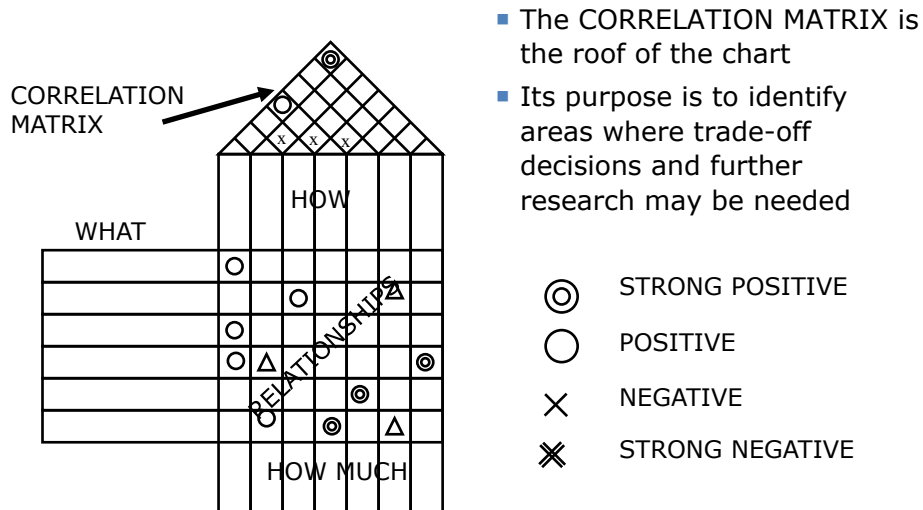
©2016 Waters Corporation

48

(Step 5): Determine Requirement Correlation and Trade-offs

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



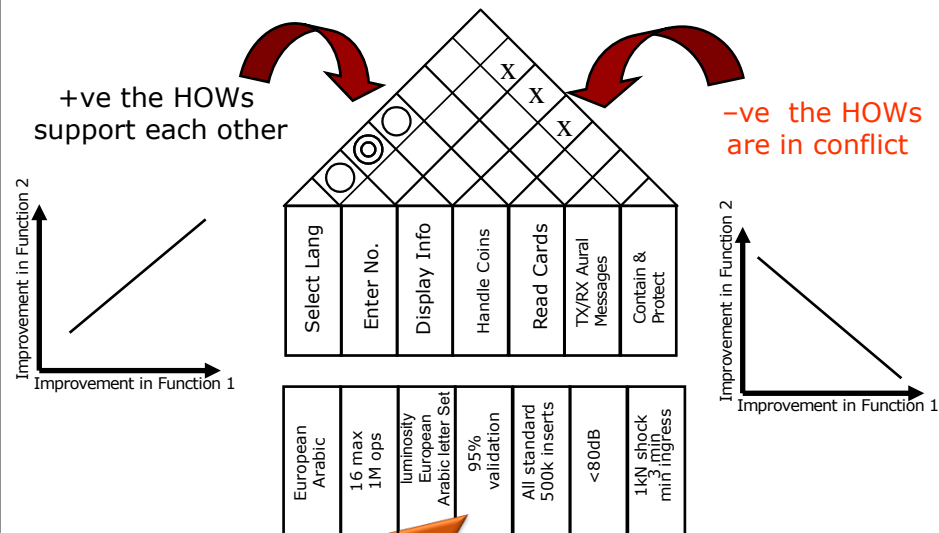
©2016 Waters Corporation

49

Correlation : Trade-offs

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



The HOW MUCHs often define the correlations

©2016 Waters Corporation

50

What do Correlations tell you?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Positive correlations indicate possible win-win situations
- Negative correlations indicate where trade-off decisions are likely BUT:
 - The initial response to a negative correlation should be to make the trade-off go away. This may require innovation or research. A negative correlation indicates where functions are in conflict.
 - Some negative correlations will require high level decisions, either by the company's senior management or by the customer. In both cases early resolution is essential
- Many negative correlations imply trade offs which are resolved by adjusting the HOW MUCHs

Tips

- If there are no negative correlations there has probably been an error
- The correlation matrix should be about 10% full
- It is possible to argue some situations both ways (i.e. positive or negative correlations). In such cases capture the negative on the chart but record the discussion. This often occurs with aggregated functions – consider decomposing the aggregated functions into their atomic form
- To fully understand the correlation between two functions it is often useful to consider the functional metrics (non-functional performance requirements = how that function is measured)

©2016 Waters Corporation

51

Conservatory Correlations

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

The idea of a correlation matrix can be extended to look for relationships between the stakeholder (customer) requirements

A strong negative correlation indicates a conflict between the requirements of Stakeholder 1 and Stakeholder 2!

A strong positive correlation between two requirements of Stakeholder 2

WHAT	HOW					
	Function 1	Function 2	Function 3			
SH ₁ Requirement 1	○					
SH ₁ Requirement 2			○		△	
SH ₁ Requirement 3	○				○	
SH ₁ Requirement 4	△	△				
SH ₂ Requirement 1	⊙			⊙		
SH ₂ Requirement 2		○	⊙			
SH ₂ Requirement 3						
	HOW MUCH					

©2016 Waters Corporation

52

Rome and a Day



- Rome wasn't built in a day!
- Likewise you CANNOT complete a QFD in a day
- Indeed, the best way to construct a QFD is step-by-step over a number of weeks since each step may require some effort

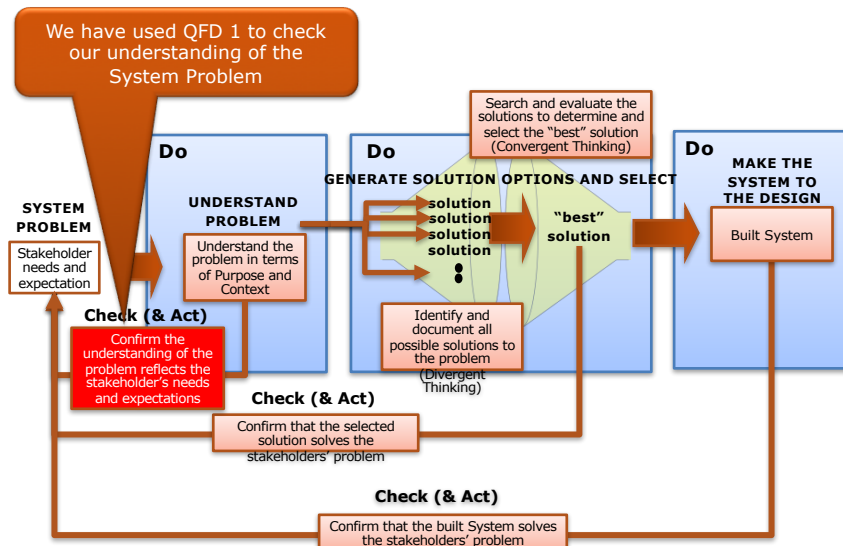
Key Success Factors

- Decide the need! Why is the QFD being constructed? – what questions will it help answer?
- Plan the completion of a QFD over several sessions
- Determine an appropriate set of WHAT requirements that are structured and validated and clearly defined (appropriate to the system and at the right level)
- A full set of functionality at a sensible level of generality that are clearly defined – if aggregated define the components
- Relationships strengths that are not “over-egged”
- Develop realistic metrics and measures for the HOW MUCHs

©2016 Waters Corporation

53

What have we done?



©2016 Waters Corporation

54

Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ Pros

- Checks completeness and consistency of requirements set (expressed and deduced) leading to:
 - Improved company performance
 - satisfied customers
 - Lower costs in design and manufacture
- Helps in determining measurable requirements
- Traceability of requirements
- Improved communication throughout company (standard tool)
- Maintains traceable customer focus

■ Cons

- Can become unwieldy if the detail is not controlled
- Can over-argue relationships
- Need to be clear which requirements go where (mixing technical requirements with customer requirements)
- Management understanding, support and commitment
- Seen as an imposition rather than a useful tool

©2016 Waters Corporation

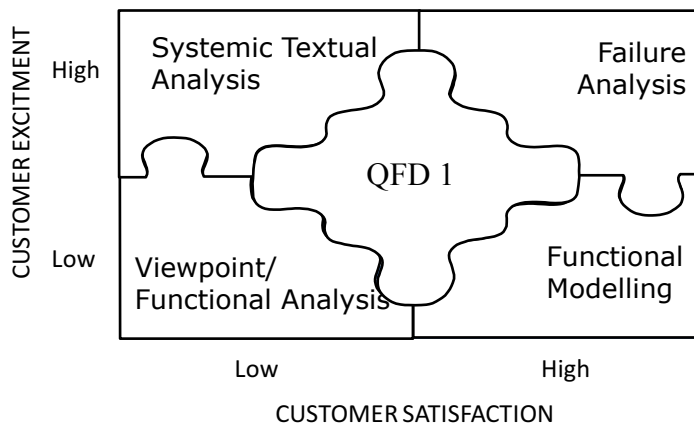
55

Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Quality Function Deployment 1 brings together the work completed so far providing a check on completeness and consistency as well as providing a summary of the full requirements set



©2016 Waters Corporation

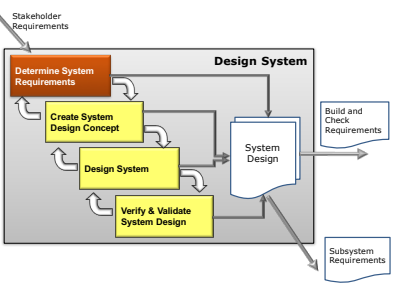
56



BURGE
HUGHES
WALSH



Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



The diagram shows a 'Design System' box containing four steps: 'Determine System Requirements', 'Create System Design Concept', 'Design System', and 'Verify & Validate System Design'. Arrows indicate a clockwise cycle between these steps. External inputs include 'Stakeholder Requirements' pointing to the start, and outputs include 'Build and Check Requirements' and 'Subsystem Requirements' pointing away from the cycle.

Specifying and Managing Requirements

This Session:

- Provides guidance on how to specify good requirements and structure requirements documents
- Introduces the elements of Requirements Management

©2016 Waters Corporation

1

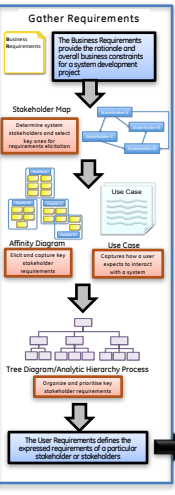


BURGE
HUGHES
WALSH

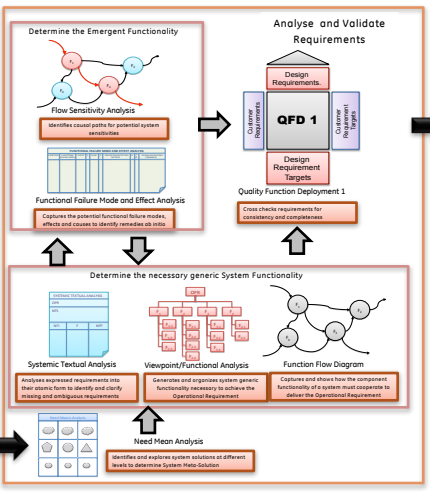


Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

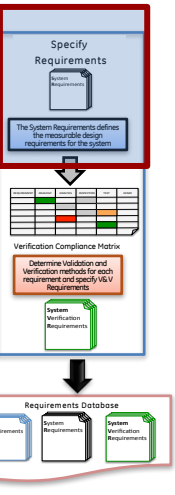
Process and Tools



This section details the initial gathering of requirements, including steps like 'Stakeholder Map', 'Affinity Diagram', 'Use Case', and 'Tree Diagram/Analytic Hierarchy Process'.



This central section covers the analysis and validation of requirements, featuring 'Flow Sensitivity Analysis', 'Functional Failure Mode and Effect Analysis', 'QFD 1', 'Quality Function Deployment 1', 'Systemic Textual Analysis', 'Viewpoint/Functional Analysis', 'Function Flow Diagram', and 'Need Mean Analysis'.



This section focuses on specifying requirements, including 'Specify Requirements', 'Verification Compliance Matrix', and 'Requirements Database'.

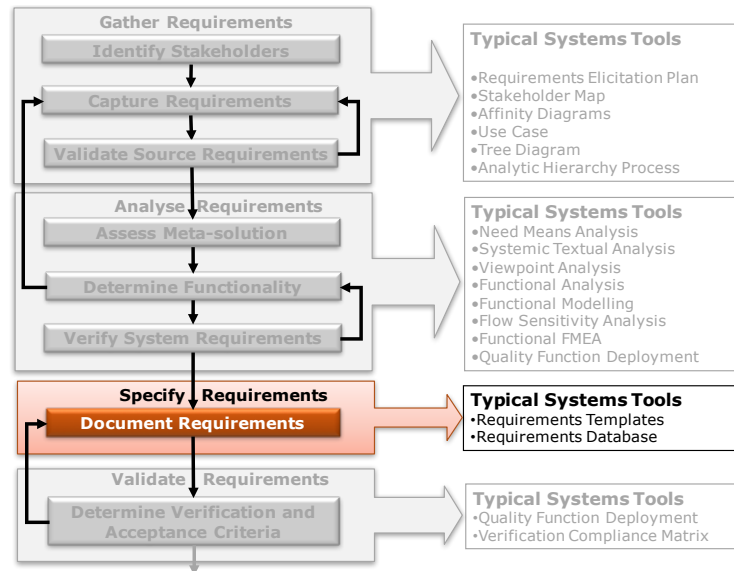
©2016 Waters Corporation

2

Determine Requirements: Tasks and Tools

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

3

Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Requirements cause problems for two key reasons:
 - In general, people find it difficult to specify requirements – usually because they have received no guidance
 - Equally, people find it difficult to analyse and understand requirements – usually because they have received no guidance

©2016 Waters Corporation

4

Specifying Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Specifying Requirements is a crucial activity in Requirements Engineering
- It is more than just generating a list of written requirements - it is about:
 - Documenting requirements
 - Writing requirements
 - Diagramming requirements
 - Structuring Requirements
 - Within a Requirement Set (Document)
 - Across and between Requirement Sets (Document)

A well documented set of requirements may still be difficult to understand because they are poorly organized.

Requirements are a "system" where individual requirements are components in that system. As we know from Systems Thinking the organization of the components (architecture) is often more critical to system performance than the individual components

In the olden days we had requirements documents. The advent of requirements software now means we have the requirements in a single database – but those requirements can be divided in to a number of interrelated sets

©2016 Waters Corporation

5

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

"A bad system will beat a good person every time."

W. Edwards Deming

Structuring Requirements

©2016 Waters Corporation

6

Introduction

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- How we organize our requirements has a significant impact on their readability and therefore their understandability
- Most requirements documents are poorly organized, overly long, boring and difficult to read
- This has happened because
 - They are organized around the system's physical architecture which itself has evolved in a piecemeal (disjointed, haphazard) fashion as technology has advanced. The requirements document is therefore piecemeal in organization
 - An over-reliance on text based requirements
 - Failure to recognise that a human designed system does NOT exist in isolation but always with associated systems that impose requirements on the designed Prime System and vice versa

©2016 Waters Corporation

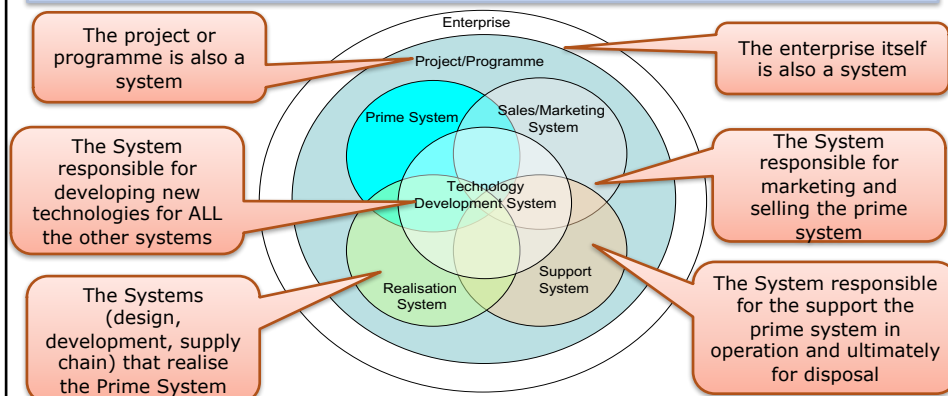
7

The PRIME System's Friends

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Every PRIME system is associated with several other systems. When we ask "stakeholders" what are their requirements for the PRIME system **they will also tell us requirements for the other systems!**



Conclusion: We cannot stop customers and stakeholders giving requirements for the associated systems. This is not a bad thing (in fact it is good that they do this) but it does mean we have to sort the raw requirements and align them to their respective systems

©2016 Waters Corporation

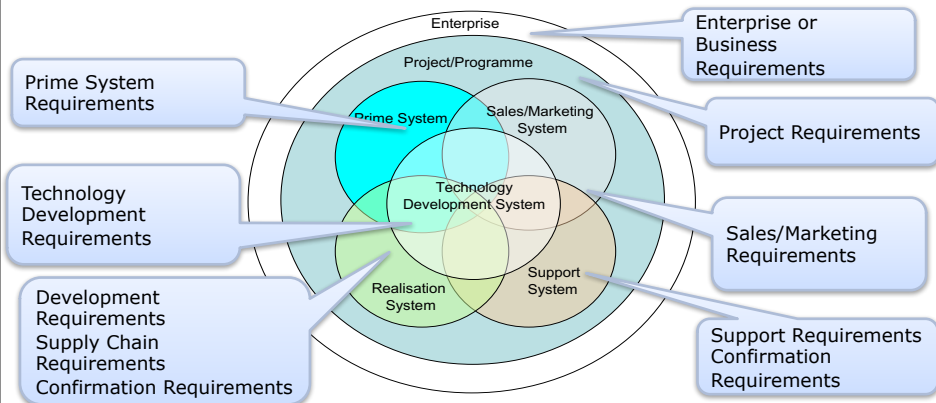
8

The Enterprise View

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Each of the associated systems will have requirements
- We try to re-use many of these systems, but there will be changes with every new prime system
- Part of planning the Systems Engineering on a project/programme is to decide what Requirements Sets will be needed



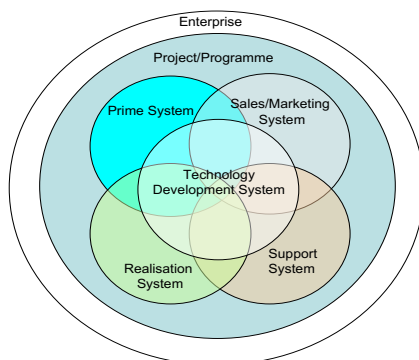
©2016 Waters Corporation

9

Requirements for the Systems

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Given a particular project or programme, each of the associated systems, including the prime, will have its own requirements
- Those requirements can be broadly divided into four main sets (documents)
 - Business
 - Stakeholder/User
 - System
 - Confirmation

©2016 Waters Corporation

10

The Basic Requirements Sets

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

For each system the basic requirements can be put into 4 sets (documents):

Business Requirements Set

Business Requirements specify why a project is being undertaken and what is expected from a business viewpoint. The business requirements may also contain marketing, outsourcing and project requirements, although on large projects or programmes these may be separate

Stakeholder Requirements Set

Stakeholder (User) Requirements specify what the user/customer will be able to do with the system together with their KEY performance needs and constraints. Such requirements may come directly from users or we may need to gather them

System Requirements Set

System Requirements also specify "what" information but in terms of the systems functional requirements and the system constraints. Systems requirements define in clear, concise, confirmable terms the system design problem that if solved will deliver the stakeholder requirements

Confirmation Requirements Set

Confirmation Requirements specify how the other requirements sets are to be proven to have been met. They define the method or methods to be used (test, demonstration, review, analysis, or analogy) and the evidence that is sought

©2016 Waters Corporation

11

More on the Basic Requirements Sets

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

For each system the basic requirements can be put into 4 sets (documents)

Business Requirements Set

Why - needed by the business and the project. Who - should be expressed by the business leaders based on information and evidence from marketing, sales, engineering, manufacturing and support.

Stakeholder Requirements Set

Why - needed by the project and the design team. Who - expressed by the key stakeholders. Either passively or actively gathered. Typically the responsibility of marketing to create - but supplemented by work from engineering and marketing to gather from lifecycle stakeholders

System Requirements Set

Why - needed by the system designers. Who - Expressed by engineering (systems engineering) following the analysis of the BRS and SRS. Should detail how the system will deliver the SRS and BRS through its functions and functional performance requirements

Confirmation Requirements Set

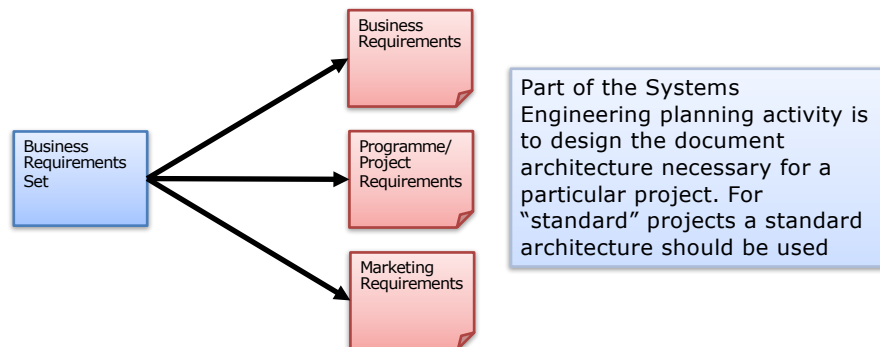
Why - needed by the system integrators, project, business and "customer" Who - expressed by various people because each of the above sets need confirmation evidence; BRS: how in an evidenced-based way will we demonstrate to the business that the "project" has met its business requirements. It should be written at the same time as the BRS - by the same people! SRS: typically written by the SE community

©2016 Waters Corporation

12

Requirements Sub-sets

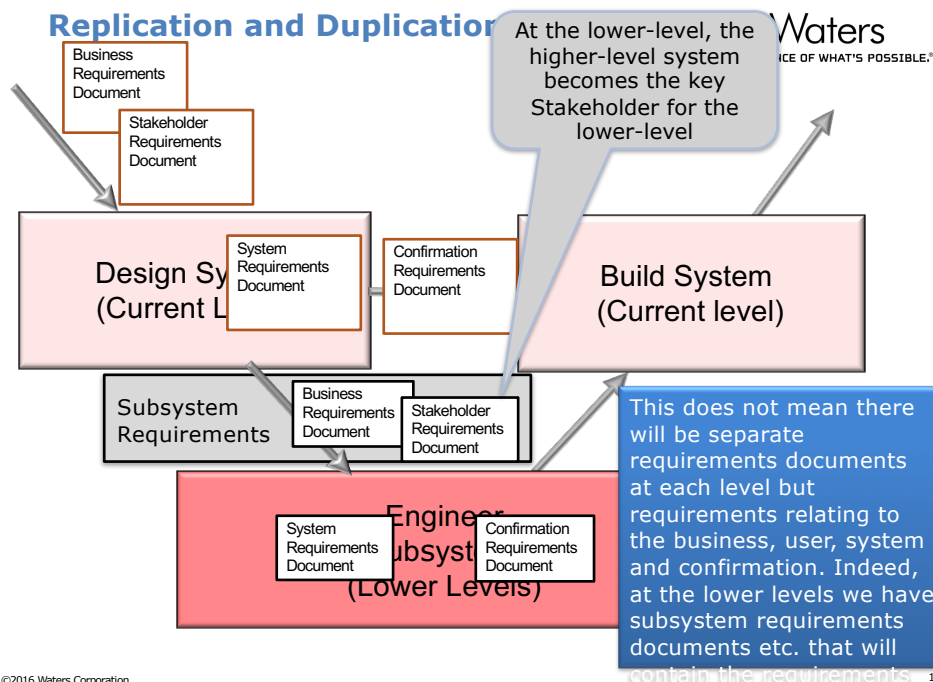
- The four basic sets of requirements can be captured in multiple levels. For example:



©2016 Waters Corporation

13

Replication and Duplication



©2016 Waters Corporation

14

Internal Requirement Document Organization

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Organization must also happen inside requirements documents
- Even relatively small projects can have hundreds of requirements. While each requirement should be individually understandable, so should the complete set. In other words the document also needs to show how the requirements fit together – and this is the role of the document structure
- Humans can understand any degree of complexity if it is presented in small 'chunks' together with a 'map' of how the chunks are structured together
 - The goal is to show the reader one simple structure (map) that can allow for any amount of detail which is presented at lower levels
 - The higher levels leave out the fine detail and concentrate on the essential information at that level. The information that is hidden at higher levels can then be revealed in the lower levels
 - Experience shows that you will probably need at least three levels of detail

©2016 Waters Corporation

15

Typical Contents of a Requirements Document

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Summary/Abstract – a brief description of the purpose and contents of the requirements document
- Document Information – Project, Document Title, Author, Revision history, Review and Approval information
- Table of contents
- Introduction
 - Purpose
 - History
 - Scope
 - Assumptions
 - Approach (how the requirements in the document were obtained)
 - Compliance Levels
 - Abbreviations
- Requirements
- Appendices

©2016 Waters Corporation

16

The Requirements Section

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- It is the organization of the requirements section that is most critical
- It is important to always remember

someone will have to understand what has been written

- Requirements are linked and therefore we need to find ways in which the links can be clear
- Therefore always consider:
 - Diagrams
 - Tables

©2016 Waters Corporation

17

What is a Stakeholder Requirement?

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Stakeholder Requirements are what the key Stakeholders want of the system
- They are typically:
 - Non-functional System Requirements – constraints at the system level
 - Non-functional Implementation Requirements – solution based requirements (also constraints)

©2016 Waters Corporation

18

Example Stakeholder Requirements Structure

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Non-Verifiable Requirements must identified early and resolved

No.	Key User Requirement	Source	Priority	Subordinate Requirements	Verification
1.1	The PCS shall improve how P&S develops its engineers	T1C7 T3C4 T3C10	E	- The PCS shall generate development plans for: - Individuals - BU - The PCS shall support individual PDRs - The PCS shall provide employees with clear job and role descriptions - The PCS shall promote "experiences" as development interventions	Demonstration Not verifiable Not verifiable Not verifiable
1.2	The PCS shall provide strategic capability analysis	T1C11 T2C3 T3C11	E	- The PCS shall highlight and identify critical strategic resources and competencies - The PCS shall allow "talent searches" across the enterprise - The PCS shall provide statistical data on skill distribution across BU	Demonstration Demonstration Demonstration

©2016 Waters Corporation

19

What are System Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The Holistic Requirements Model helps us define what System Requirements are:

Function A +
 { Non-functional Performance Requirement
 Non-functional Performance Requirement
 Non-functional Performance Requirement
 etc.

Function B +
 { Non-functional Performance Requirement
 Non-functional Performance Requirement
 Non-functional Performance Requirement
 etc.

+ any interface requirements decided by the higher-level system

©2016 Waters Corporation

20

Deducing Non-functional Performance Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- If we are designing a system with a known meta-solution we should have most the of the NFP requirements. Their parameters/metrics at least – we may not yet have the numerical target values - TBD
- It will be a matter of collecting and collating them against the Functions
- If not, we will have to derive them:
 - a common approach here is to consider two solutions, say A and B, to each function and ask what could be measured to decided between them? The key here is to identify measurable characteristics
- We should also consider the flow-down of the Non-functional System requirements e.g. reliability, cost, weight, size etc.

©2016 Waters Corporation

21

Structuring the System Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

The high-level structure of a requirements document is critical. Generally there are two approaches

Physically Based

- Requirements organized around the physical architecture (structure) of the previous design
- Advantages:
 - Easy to relate to the physical "product"
 - Based on an existing known (and therefore working) architecture
 - The enterprise is also often organized in this way and so work allocation is easy to apportion.
- Disadvantages:
 - Focuses on "things" rather than capability
 - Not natural to users/customers
 - Leads to sub-optimisation
 - Difficult to read *in toto*

Use (functionally) Based

- Requirements organized around the natural patterns of use (scenarios)
- Advantages:
 - Focuses on delivering capability to the user
 - Easy for users/customers to understand
 - Ensures whole system (holistic) view
 - Presents requirements as a story
- Disadvantages:
 - Doesn't relate easily to the physical "product"
 - Doesn't match the enterprise architecture

©2016 Waters Corporation

22

Organizing Requirements in Practice

- Prepare Aircraft for Flight
 - 1.1 Fuel Aircraft
 - 1.2 Clean Aircraft Interior
 - 1.3 Load Baggage
 - 1.4 Load Meals and Supplies
 - 1.5 Load Passengers
 - 1.6 Perform Pre-flight Checks
 - 1.7 Push-off from stand
- Fly Aircraft from A to B
 - 2.1 Communicate with Air Traffic Control
 - 2.2 Taxi to Runway
 - 2.3 Takeoff Aircraft
 - o 2.3.1 Accelerate Down Runway
 - o 2.3.2 Abort Takeoff
 - o 2.3.3 Rotate Aircraft
 - 2.4 Fly Aircraft
 - o 2.4.1 Climb Aircraft
 - o 2.4.2 Navigate Aircraft

Requirements Heading form a natural sequence of activities

Related activities are grouped logically into scenarios or modes of operation

The hierarchical nature of organization lends itself to simple diagrams like a Function Tree Diagram or tabular layout

Significant exceptions are listed where they would logically occur

System Requirements Tabular Layout

No.	Functional Requirement	Non Functional Performance Requirement	Source	Verification
1.0	The system shall determine the current location to:	1. Accuracy $\pm$ 300mm	Derived from SRD requirement 1.6	Test
		2. Repeatability $\pm$ 30mm	Derived from SRD requirement 1.6	Test
		3. Weight < 1.7kg	Derived from SRD requirement 2.6	Test
		4. Size Envelope < 300x50x60 (WxHxD)mm	BS1759	Test
		5. Reliability > 1000hrs MTBF	Derived from SRD requirement 29.0	Analysis and Test
2.0	The system shall	1. XXXX		

Writing Requirements

I have made this [letter] longer, because I have not had the time to make it shorter.

Blaise Pascal, "Lettres provinciales", letter 16, 1657
French mathematician, physicist (1623 - 1662)

- Written Requirements are still the prime way in which one stakeholder conveys their needs and expectations
 - Yet we struggle to achieve this even though we are using our natural language - English
 - To write a good requirement does require us to understand the English language – but some simple guidelines can help

What are “Good” Requirements?

- A good requirement should be:
 - Clear: unambiguous and using the active voice
 - Concise: simple structure with few words
 - Consistent: with itself and other requirements
 - Complete: it expresses complete meaning
 - Correct: asks for something that is possible (do-able and legal)
 - Confirmable: can be proven to have been met

Guidelines for Writing Good Requirements

- The 11 DOs
 - Do write requirements one at a time as a simple sentence
 - Do avoid compound or complex sentences
 - Do use simple direct sentences – don't ramble
 - Do use the active voice
 - Do use simple English words
 - Do avoid "let-out" clauses or wishful thinking
 - Do refrain from designing the system
 - Do identify the stakeholder (remember we treat the system of interest as a stakeholder) who wants each requirement
 - Do focus on stating the result to be provided to the stakeholder
 - Do ensure the requirement is confirmable
 - Do define the level of compliance sought

The Art of Writing Requirements

Clear, Concise, Consistent, Complete, Correct & Confirmable

Writing "good" requirements is not easy



- 3 Possible styles
- Classic
 - Alternative
 - Tabular

It is always important, when writing requirements, to remind yourself:

- This requirement is a contractual obligation
- Somebody has to understand this requirement

Anatomy of the Classic Style Requirement

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Born out of the Defence industry, the classic style for writing requirements uses the following format

Describes the conditions that need to occur for the requirement to become active. They are:

- Ubiquitous
- Event Driven
- State Driven
- Optional

Defines the level of compliance:

- SHALL
- MUST
- WILL
- SHOULD
- MAY

Specifies any required performance level or other conditions for the response

[preconditions][the stakeholder][compliance level][response][qualifier]

The person or thing that wants the system response

Describes what the system must actually do that is "visible" to the stakeholder

©2016 Waters Corporation

29

Preconditions

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- There are four basic types of preconditions to a requirement
 - Contonuous
 - Requirement is always active
 - No preconditions or trigger
 - Event-driven (keyword When)
 - Required response to a triggering event
 - State-driven (keyword While)
 - Required response in a specified state
 - Option (keyword Where)
 - Applicable only if the "feature" is included

Note: it is possible to use combinations of When, While and Where for complex conditional clauses – but beware complex conditional cases are sources of misunderstanding – consider alternative approaches e.g. Flow Charts, State Transition Diagrams etc.

©2016 Waters Corporation

30

Ubiquitous and Event Driven

Continuous

- Used to define behaviour that must be active at all times (“continuous”)
- Examples
 - The control system shall prevent engine over-speed
 - The user shall be able to check their account balance

Event Driven

- Initiated only when a triggering event is detected at the system boundary
- Examples
 - When power is applied to the system, the system shall initiate the Built In Test Equipment
 - When the account balance is greater or equal to the withdrawal request, the user shall be able to withdraw cash

State-driven and Optional

State-driven

- Requirement is active while the system is in a defined state
- Examples
 - While the autonomous lawn mower is in mow mode the manage power function shall monitor energy levels
 - While a person is a registered guest, they shall be able to charge food and drink purchases to their room bill

Optional

- Applicable only to systems that include a particular “feature”
- Examples
 - Where electronic components are used in the control system, they shall comply with IR2301
 - Where the guest is a gold card holder, the guest shall be provided with internet access free of charge

The Auxiliary Verb and the Level of Compliance Sought

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

The System [what goes here] Toast Bread Products

- **Must**
 - Indicates a mandatory requirement that originates in the laws of the land
- **Shall**
 - Indicates a mandatory requirement
- **Should**
 - Indicates a desirable requirement
- **May**
 - Indicates either an optional requirement, or a statement relating to how mandated requirements can be achieved
- **Will**
 - Indicates either a statement of intent, or a statement relating to something outside the scope of the system being developed, but that is relevant to the system
- **Note:**
 - These are not the only possible auxiliary verbs, others include: could, ought, might, would, can.
- The auxiliary verb specifies the level of compliance of a requirement.
- It is recognised good practice that a “good set” of requirements contains only “shall”. However, when receiving requirements expect anything and expect it to be incorrectly used.

©2016 Waters Corporation

33

Guidance: Writing and Analysing

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- **Writing Requirements**
 - Never assume that other people will use the same definitions as you! When writing requirements always define what you mean by these terms
 - In the ideal world, use only SHALL
- **Analysing Requirements**
 - Expressed requirements are likely to contain Must, Shall, Should, Will and May
 - “Should” and “May” indicate non-mandatory optional requirements that need sorting out and agreeing with the customer - quickly
 - Assume that all “should”s are “shall”s
 - Assume that all “may”s are not “shall”s
 - Ideally, following requirements analysis only “shall”s remain (some writers like to use “must” for legal requirements). However, this might not be possible until after the design activities have taken place and requirements have been allocated to subsystems

©2016 Waters Corporation

34

Unwanted Behaviour Requirements

- There are situations where “unwanted behaviour” occurs and we should wish the system to respond in a certain way
- “Unwanted behaviour” includes
 - Failures
 - Deviations from desired behaviour
 - Any unexpected behaviour of interacting systems
- Unwanted Behaviour Requirements therefore define the required response of the system to unwanted external events. They are a variation of the Event-driven requirement:

If [preconditions] the system [auxiliary verb] [response][qualifier]

- This format forces the separation of:
 - Circumstances in which the requirement can be invoked (preconditions)
 - The expected system behaviour (response)
- Examples
- If the Automated Teller System is out of cash, then the Automated Teller System shall inform the user it is out of cash

Common Pitfalls in Writing Requirements

- As has been said before, writing good requirements is not easy and mistakes will be made that include:
 - Ambiguity
 - Multiple requirements
 - Complex logic
 - Let-out clauses
 - Rambling
 - Designing the system
 - Mixing different kinds of requirements
 - Speculation
 - Un-definable, un-confirmable requirements
 - Possibilities
 - Wishful thinking
 - Negative Requirements

See Appendix A
For more detail

The Alternative Style of Requirement

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Classic Style

The system shall deliver a thrust of at least 70,000lbs at sea level, 120kts and International Standard Atmosphere + 15°C

Alternative Style

A thrust of at least 70,000lbs shall be delivered at sea level, 120kts and International Standard Atmosphere + 15°C

The differences between the two styles may seem only stylistic, but readability of requirements is critical and so is emphasis. The "alternative" style places the emphasis clearly on what is required – a thrust of at least 70,000lbs, whereas in the classic style the emphasis is on the "system". Most customers (stakeholders) actually couldn't care less about the system but they do care about the response!

So What?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The alternative style of writing requirements results in requirements that:
 - Are more concise
 - Are easier to read
 - Place the emphasis on the response
- However, the alternative style is in passive voice which is often considered poor practice
- All the requirements written in the classic style can be rephrased in the alternative style
- Mixing the two styles is not recommended because of the emphasis they confer
- Therefore as a general rule, use the active voice. But there are occasions when writing requirements where the use of the passive voice can be useful.
- If we really don't care how something is done, as long as it is done, then the passive voice can be useful because it promotes innovation – for example
 - The gas turbine shall deliver a thrust of at least 70,000lbs at sea level, 120kts and International Standard Atmosphere + 15°C
 - The system shall deliver a thrust of at least 70,000lbs at sea level, 120kts and International Standard Atmosphere + 15°C
 - A thrust of at least 70,000lbs shall be delivered at sea level, 120kts and International Standard Atmosphere + 15°C

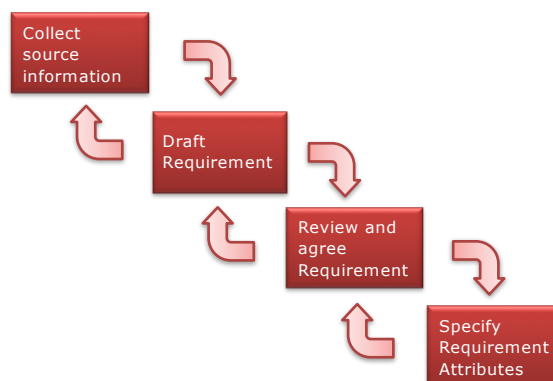
Tabular Style

- Tables often provide a powerful way of writing requirements because they convey the information in easily digestible chunks
- Tables are also a format that is amenable to computerisation and therefore computer-based requirements management. This means that additional columns can be added to provide further information about a particular requirement (or group of requirements) such as the requirement attributes like number, status, author, owner etc.
- Critical though to using a tabular style is the:
 - Table headings
 - Table layout
 - Numbering system

Functional Requirement	Non Functional Performance Requirement	Comments
The system shall determine the current location to:	Accuracy ± 1ft	
	Repeatability ± 2 inches	
	Weight < 1.7lbs	
	Dimensions 7x6x3 inches (w,d,h)	
	etc	

A Process for Writing Requirements

- Writing “good” requirements is not easy – but can be made easier by following a simple process



Requirement Attributes

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

A system can have many hundreds if not thousands of requirements. These will originate from many stakeholders and may have differing priorities. Requirements should have a reason for their existence and a defined and agreed method of confirmation.

- These aspects are typically captured as the attributes of a particular requirement
- Typically Requirement Attributes include:
 - Number: all requirements must be uniquely numbered for reference and traceability
 - Source/Author: person, document or other origin of a particular requirement. The source provides useful information on whom to ask regarding questions
 - Priority: a statement of the relative importance of a particular requirement
 - Owner: who within the organization owns (is responsible for making sure the requirement is met) a particular requirement
 - Rationale: why the requirement is needed
 - Comments: reviewer's or writer's comments about a particular requirement
 - Status: the standing of a particular requirement: draft, agreed, modified, confirmed or retired
 - Change History: documents the changes to a particular requirement
 - Risk: confidence level of meeting the requirement (may include date information)
 - Due by: date that the requirement must be provided
 - Method of Verification: type of confirmation method(s) to be used in verifying that the requirement has been met

A Final Quality Check

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- A useful check on the quality of an individual requirement is to ask "how would this be confirmed?"
- It is only when we consider actually proving a requirement has been met do its deficiencies appear
- Remember requirements are a commercial obligation – if we cannot confirm that they have been met we have not fulfilled our commercial obligation

Exercise : Writing Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



- In pairs consider the case study exercise:
 - Individually write one System Design Requirement for the Case Study system
 - In pairs share and critique each others requirements
- Retain these requirements for later use.
- Time 15 minutes

©2016 Waters Corporation

43

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

"a picture paints a thousand words"
unknown¹

Diagramming Requirements

1: It is believed that the modern use of the phrase stems from an article by Fred R. Barnard in the advertising trade journal *Printers' Ink*, promoting the use of images in advertisements that appeared on the sides of streetcars.[1] The December 8, 1921 issue carries an ad entitled, "One Look is Worth A Thousand Words."

©2016 Waters Corporation

44

Diagramming

- Humans communicate information using three basic methods

Method	Characteristics
Natural Languages	Natural/intuitive but ambiguous. Linear understanding. Can show small scale structure but does not convey large to medium scale structure
Mathematical Languages	Difficult to use but unambiguous. Linear understanding. Difficult to show structural relationships
Pictures/Diagrams	Natural and easy to use. With appropriate diagramming conventions/notation can be unambiguous. Concurrent understanding. Can show large and medium scale structure. Tedious to show structural detail

Standardised Diagramming Tools

- There are numerous diagrams/pictures that can be used instead of, and to supplement, written requirements.
- These diagramming approaches are only useful if they convey the information the sender intended. This is achieved through the use of Standard Notations.

Tool Category	Tools (not exhaustive)
Define/Describe System Context	Context Diagram , Use Case Diagram, Input-Output Diagram
Define/Describe System Interfaces	Context Diagram, Function Flow Diagrams, N² Diagram , Use Case Diagram
Define/Describe System Purpose/Functionality/Behaviour	Viewpoint Analysis, Function Flow Diagrams , Activity Diagrams, Sequence Diagrams
Define/Describe Data/Information Relationships	Entity Relationship Diagrams, Class Diagrams, Flow Dictionary
Define/Describe System States	State Transition Diagram, State Chart Diagrams
Define/Describe Complex Logic	Flowchart, Structured Specification Languages , Decision Tables/Trees

When to Use Diagrams/Pictures

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Whenever you can

But Remember

This requirement is a contractual obligation
Somebody has to understand this requirement

©2016 Waters Corporation

47

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

*"...that it is very important to have a system. If you have a system
which tells you where you are, then you will never be lost."*

Alexander McCall Smith "Morality for Beautiful Girls"

Managing Requirements

©2016 Waters Corporation

48

The Life of a Requirement

- Requirements have a life! and it is important to understand something about that life since it can provide vital information about the state of the system development
- The life states are:
 - Born (drafted or created)
 - Matured (adolescent)
 - Approved
 - Changed or modified (mid-life crisis)
 - Achieved or Confirmed
 - Retired
- A system can also have many thousands of requirements!

It's important to manage this

And Finally

- Remember the purpose of writing requirements
 - These requirements are a contractual obligation
 - Somebody has to understand these requirements
- Be creative in writing requirement documents
 - Use diagrams
 - Use tables
 - Make sure the structure is simple at the highest level
- Review requirements documents following system commissioning and acceptance
 - Was the system easy to commission?
 - Was the system readily accepted?

Appendix A

©2016 Waters Corporation

51

Avoid ambiguity

- For many reasons ambiguity is perhaps the commonest of all pitfalls. Avoiding ambiguity is difficult since it requires us to write clearly and explicitly. This can, however, lead to the text becoming tedious and boring.
- Words to avoid “or” “/” “and/or”
- Dangerous expressions: any TLAs

- Example

The same subsystem shall also be able to generate a visible or audible caution/warning for the attention of the co-pilot or navigator or MCO

What
“same”

“also” !!

It is visible or
audible? Or will
both be needed?

It is a caution or
warning?

“attention” how is
this verified

Which one or
all?

MCO a
TLA

In this example the writer has attempted to allow some design freedom (requirement should be solution free!) but has ended up introducing ambiguity

©2016 Waters Corporation

52

Avoid Multiple Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Avoid using conjunctions (and, or, also, but, for, yet) when writing requirements – unless part of a clear precondition or qualifier

- Example

The power low indication shall be displayed when the voltage drops below 3.6 volts, and the current workspace or input data shall be saved

©2016 Waters Corporation

53

Avoid Complex Logic

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Complex logic in requirements offers plenty of opportunities for ambiguity and misunderstanding
- Consider the familiar on-line banking situation:

Users shall be able to transfer monies between accounts for which they have access. The user shall be able to specify a withdrawing account, a destination account and a transfer amount. If the transfer amount requested is greater than the currently available balance in a particular account the user shall be able to reduce the transfer amount or cancel the transfer.

- This situation describes a relatively simple situation, but is already heading towards the realm of a very poorly written requirement
- For complex logic situations other approaches have to be used:
 - Structured Specification Languages (see Functional Modelling)
 - Diagramming – even flowcharts!

©2016 Waters Corporation

54

Avoid Let-out Clauses

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Requirements that contain let-out clauses can at first glance look definite and clear. Such requirements, unless spotted early, generate problems during confirmation (validation or verification as appropriate)
- Words to avoid include: except, unless, although
- Words to use with caution: if, when, where
- Example

The fire alarm shall always be sounded when smoke is detected, unless the alarm is being tested or the engineer has suppressed the alarm

Avoid Rambling

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- You don't need a lot of words to convey clear understanding

The Lord's Prayer	56
The 10 Commandments	197
The U.S. Declaration of Independence	300
The EC Directive on the export of duck eggs	29,611

Avoid Designing the System

- “Good” requirements are solution free – within the scope of the meta-solution. This is often overlooked when writing subsystem requirements because in order to specify the subsystem the higher level system has to be designed.
- Danger signs include: materials, component names, brand names, software packages or objects.
- Example:

The TV receiver shall receive the digital signal from the antenna using a shielded copper core with a nylon covering and a waterproof hardened rubber outer covering

Avoid Mixing Different Kinds of Requirements

- A common error in narrow markets when the distinction between Customer/User Requirements and Systems Design Requirements is blurred.
- Danger signs are user perspective requirements (the user shall be able to) mixed in with system, design or installation aspects
- Example

The user shall be able to view the currently selected channel number which shall be displayed in 24pt Arial font in the top right hand corner of the LCD display

Avoid Speculation

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Remember requirements define our contractual obligation – there is no room for wish lists or general expressions about things that someone probably wants
- Words to avoid: usually, generally, often, normally, typically
- Danger signs: vagueness about which stakeholders
- Example:

Users normally like an early indication of low power level

©2016 Waters Corporation

59

Avoid Vague Un-definable Terms

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Adjectives and adverbs should be avoided because they are too vague. The consequence is that they are impossible to confirm without some form of translation into an equivalent quantitative requirement. (This is often true of customer/user requirements – but we should be able to avoid this)
- Words to avoid: maximise, minimise, very, highly, flexible, approximately
- Phrases to avoid: user friendly, easy to use
- Examples:

The system shall be easy to use
The system shall be very reliable
The system status indication shall be displayed as soon as possible after the system self-test is completed

©2016 Waters Corporation

60

Avoid Possibilities

- Requirements that express possibilities should be avoided and are often ignored by the system developer
- Words to avoid: may, might, should, ought, could, perhaps, probably and possibly

- Examples

The aircraft structure should be able to withstand 110% overload

The autonomous lawn mower shall avoid permanent and moving obstacles, a learning facility may be useful

Avoid Wishful Thinking

- Requirements must be realistic, yet it is possible, and easy, to write un-realistic requirements
- Danger phrases: 100% reliable, safe, future-proof, never fail, please all users, error free

- Examples:

The system shall be 100% safe in normal operation

The navigation subsystem shall handle all unexpected errors without crashing

The system shall be state of the art

Avoid Negative Requirements

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- It is good practice to find out what stakeholders “don’t want”, but the response is often a negative requirement. For example

The system shall not burn the toast

- In such cases the intent of the requirement can be retained through a neutral requirement and associated performance. For example:

The system shall control the toasting of the bread product to the level set by the user.

©2016 Waters Corporation

63

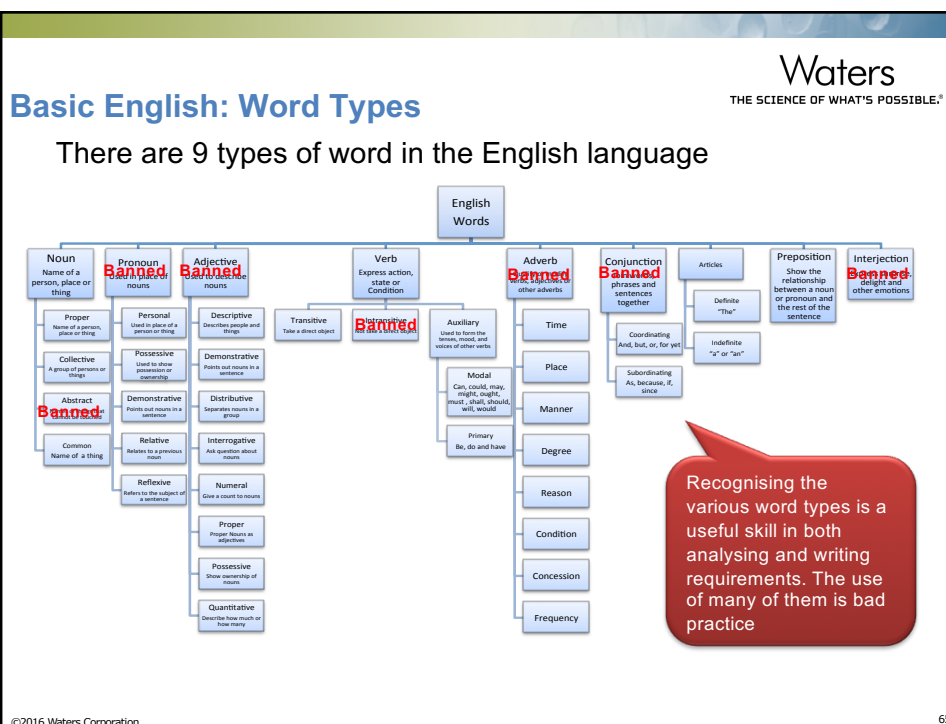
Are My Requirements “Good”?

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

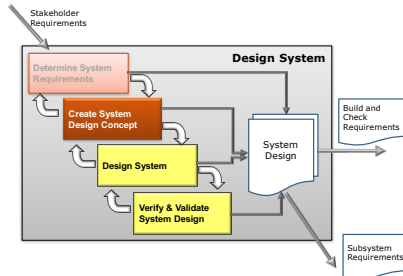
- Writing “good” requirements is a skill that can be honed with practise
- Following the guidelines presented here can help
- A useful “test” of “goodness” is to consider how a particular requirement is to be confirmed – how would I prove that the requirement has been met?
 - this aspect is covered formally in a later section, but is a useful discipline to maintain at all times.

©2016 Waters Corporation

64



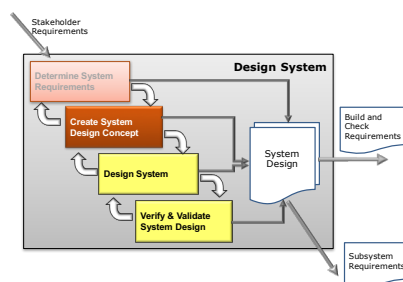
A Systems Approach to Systems Design



This Session

- Explains a Systems Approach to System Design
- Emphasises the importance of using functionality to:
 - Manage complexity
 - Provide an invariant traceability record
 - Help assure a holistic solution

Where are we?



"Design is a funny word. Some people think design means how it looks. But of course, if you dig deeper, it's really about how it works"

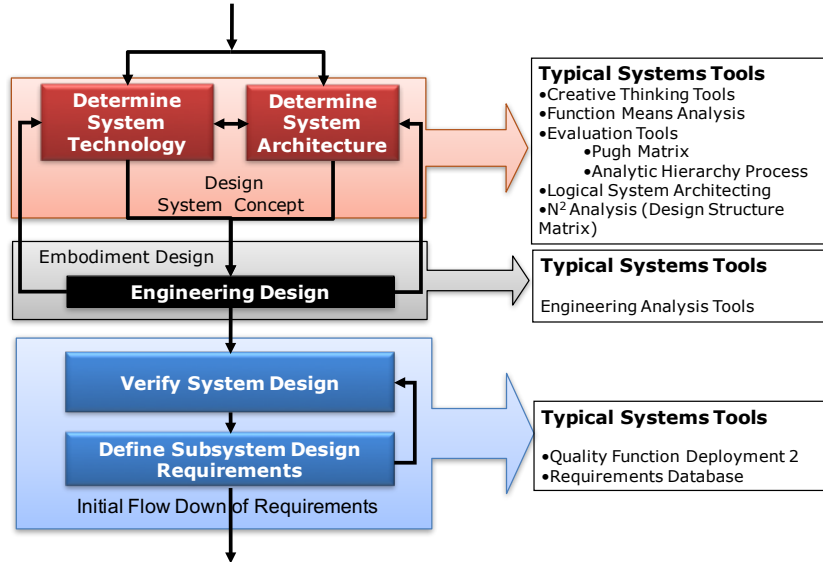
Steve Jobs

- Purpose :
 - To determine the overall system design concept that best matches the requirements
- Outcomes :
 - A conceptual system design solution in terms of
 - the technologies that will deliver the technical system design requirements (and thereby the customer/stakeholder requirements)
 - a system architecture that allows the concept design to be realised
 - Engineering Design and Analysis to determine baseline design
 - Initial flow-down of requirements to subsystem level

Key Tasks and Systems Tools

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



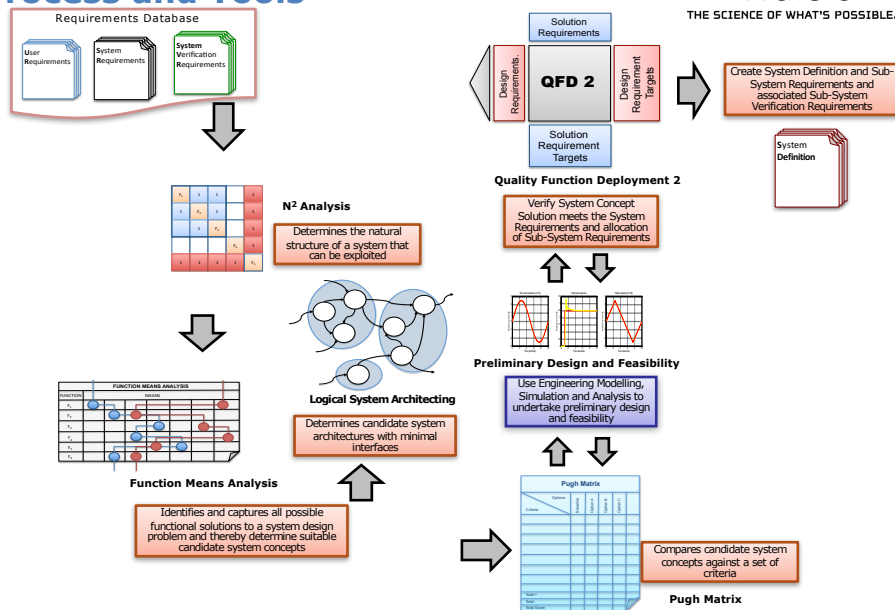
©2016 Waters Corporation

3

Process and Tools

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

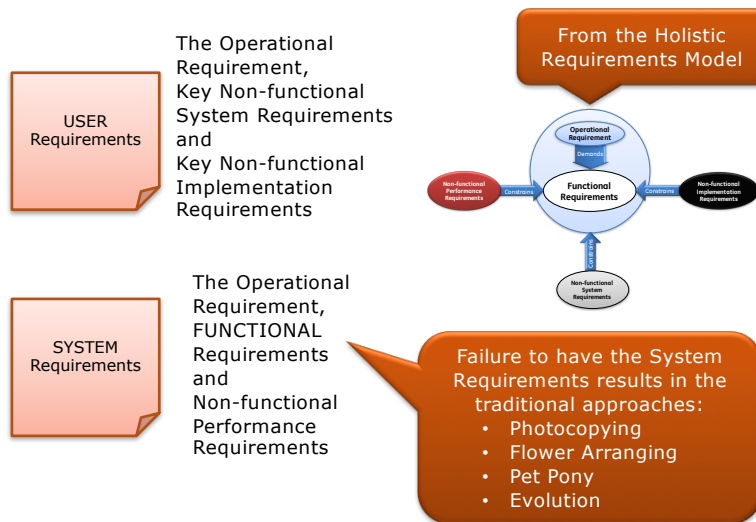


©2016 Waters Corporation

4

In order to Design a System we Must Have

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

5

A bad beginning makes a bad ending Euripides, Aegaeus Greek tragic dramatist (480 BC - 406 BC)

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

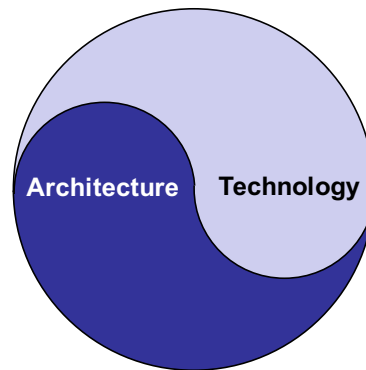
- The use of the various tools in the Determine System Requirements sub-process has provided an understanding of the customer/stakeholder requirements in terms of:
 - The overall purpose of the system - the Operational Requirement(s)
 - The basic (generic, legislative, emergent) functionality needed to achieve the overall purpose - the Functional Requirements
 - The constraints placed upon the system and its functions – the Non-functional Requirements
 - The sensitivity of the functions to variation and failure
- With this information it is now possible to undertake the first steps in developing a concept design solution

©2016 Waters Corporation

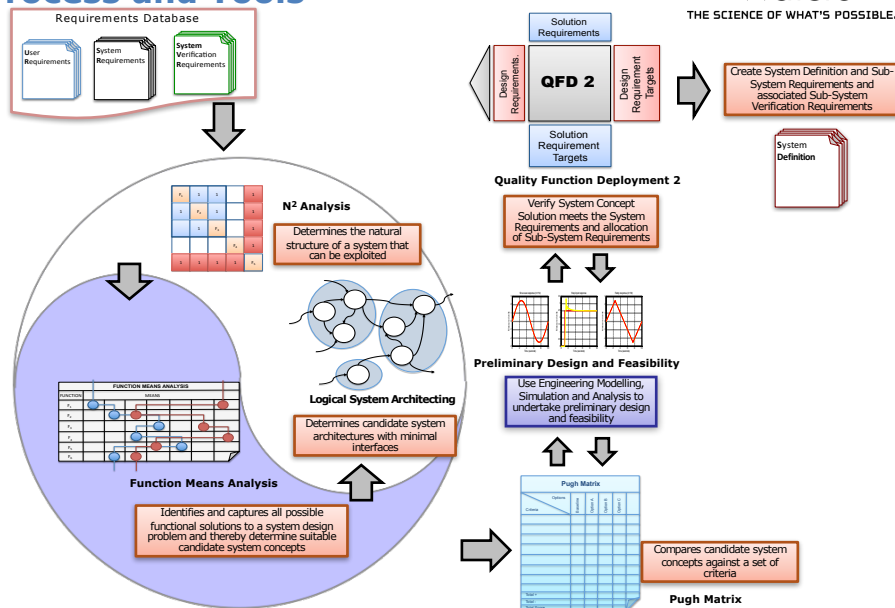
6

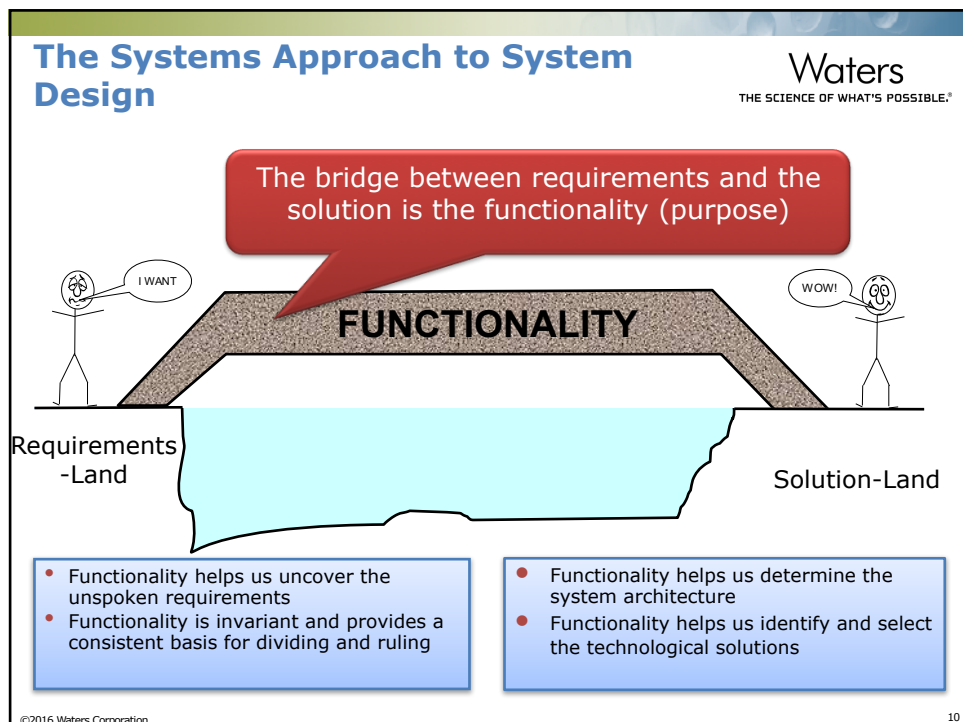
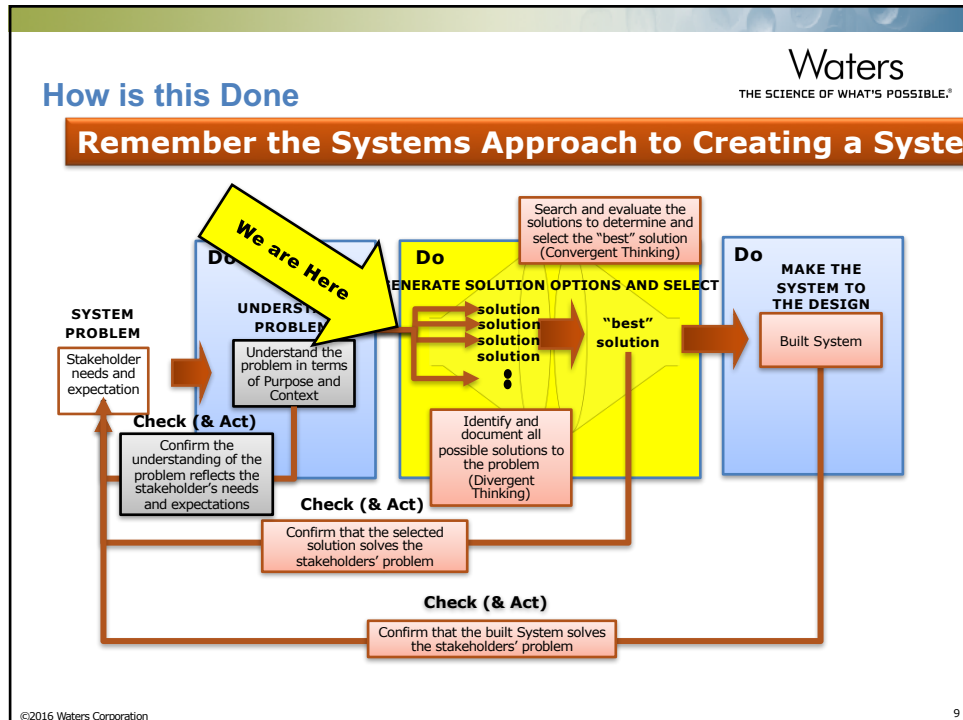
The Yin Yang of System Design

- System Design comprises two interrelated sets of decisions:
 - What TECHNOLOGY is to be used
 - What is the ARCHITECTURE of the system
- Both have to be decided in order to specify the complete system design
- Which comes first?
 - Decisions made in terms of the technology can restrict and constrain the choice of architecture
 - Decisions made in terms of architecture can restrict and constrain the choice of technology
- Good System Design keeps options open as long as possible



Process and Tools

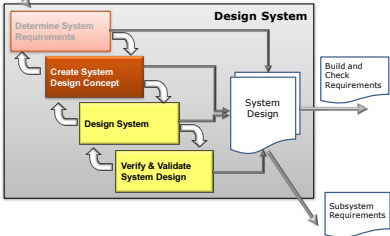








THE SCIENCE OF WHAT'S POSSIBLE.®



Create System Design Concept

System Architecting

This Session:

- Explains the importance of architecture on system performance
- Explains the concepts and principles of System Architecting
- Describes Logical System Architecting and N² Analysis as simple practical tools for generating candidate system architectures
- Emphasizes the importance of interfaces in architectural design and approaches to their design and management

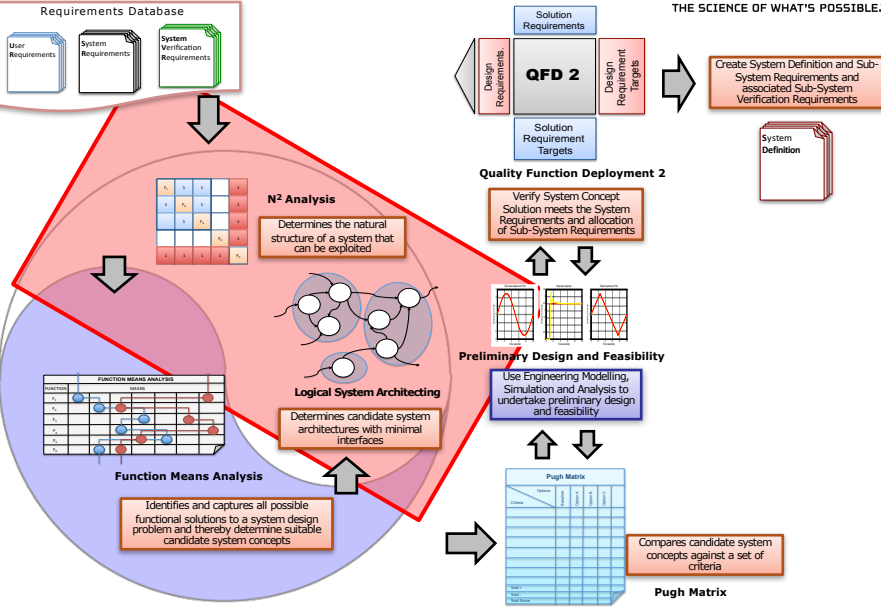
©2016 Waters Corporation

1

Process and Tools



THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

2

Definitions

- **Architecture**
 - mass noun
 - the art or practice of designing and constructing buildings.
 - The style in which a building is designed and constructed, especially with regard to a specific period, place or culture
 - the complex or carefully designed structure of something
- (New Oxford English Dictionary 2007)
- **System Architecture** – the organization and structure of a system
- **System Architect** – a person who designs system architectures
- **System Architecting** – the synthesis of a system architecture

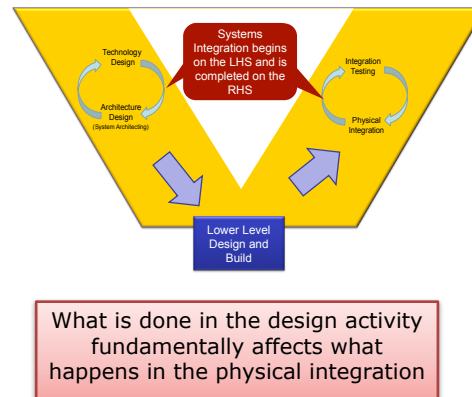
System Architecture

- **System Architecture is about:**
 - Organization
 - Groups of system components into SUBSYSTEMS
 - Interconnection (what are the INTERFACES)
 - Command, control and information
 - communication between subsystems
 - communication between system components
 - Energy flows
 - Force flows
 - Heat flows
 - Material flows
- The way in which a system behaves and performs is often fundamentally affected by the architecture
- Synthesising an architecture is therefore important
- Architecture is key to systems integration

Note that some may be emergent but we still have to consider them in our design

What is Systems Integration

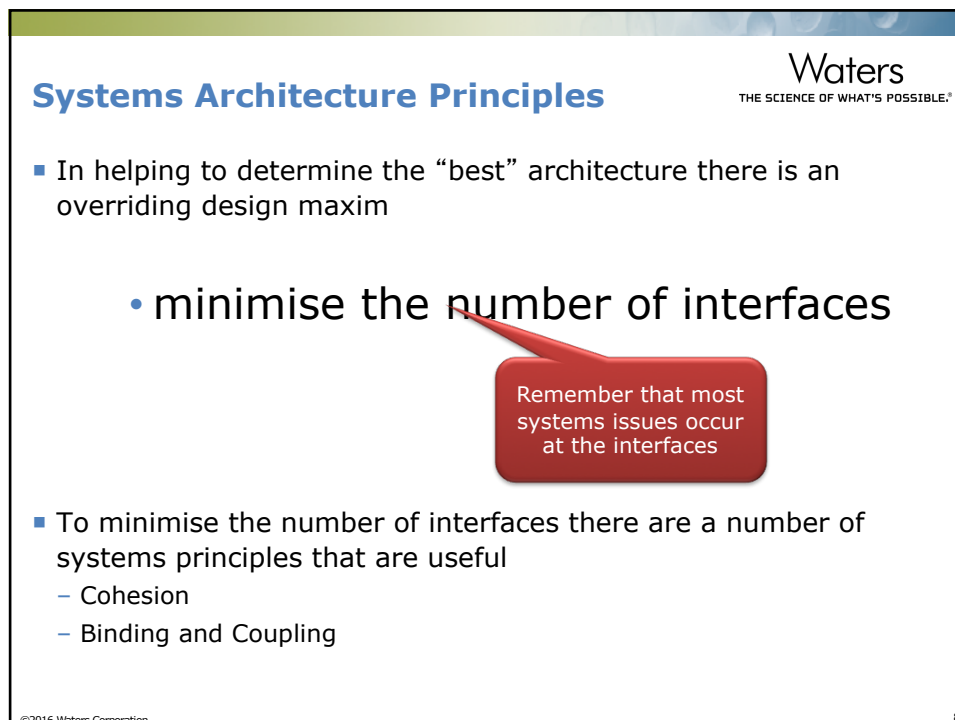
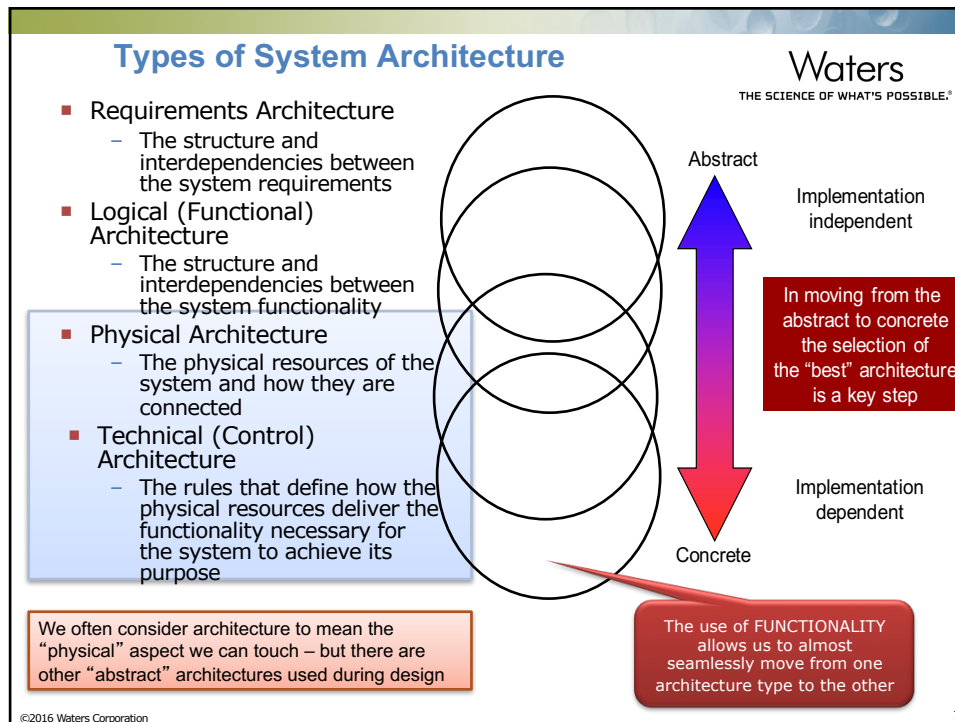
- Systems Integration has two elements:
 - Physical System Integration is the bringing together of the component subsystems into one system and ensuring that the subsystems function together as a system that is able to deliver the desired operational requirements
 - Design Integration is:
 - Deciding on the “best” means to system functionality
 - Deciding on the “best” organization of subsystems
 - Deciding on the “best” interfaces



Implications of Architecture Selection

- Physical Integration – the architecture will define the ground rules for physical system integration and therefore test
- System performance is constrained by the architecture, and equally a specified performance level may force an architecture
- Development – speed and ease of development is a function of the architecture (fundamentally can elements be developed in parallel)
- Realisation (Manufacturing/Test/Implementation) – the choice of architecture is a strong influence on the ease of realisation (manufacturability, testability etc.)
- The inherent reliability of a system is dictated by the architecture and equally the architecture can drive reliability
- Maintainability, longevity, upgradeability etc. are all a function of architecture

Architecture is Important



Cohesion



- Wherever possible functionality should be combined in the design solution
 - Aim to get system components to do more than one thing
 - In simple terms a lower part count increases reliability and reduces cost (in general)
- Functions that are physically connected or correlated (QFD 1) should be investigated for potential synergies
- A classic example of this is the Mini where the gearbox was made integral with the sump of the engine. Such ideas, however, frequently incur some element of risk. In the case of the Mini, there was the need to find a suitable oil that would serve both engine and gearbox

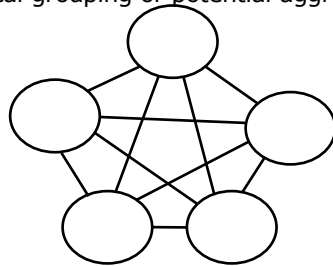
The definition of cohesion here comes from classic mechanical engineering good practice (for example the texts of M J French "Engineering Design"). The term has a different meaning in software. In software engineering it refers to the "single mindedness" of a module, i.e. each module should do one thing, one thing only, and do it well, without needing any other modules to help it do it.

©2016 Waters Corporation

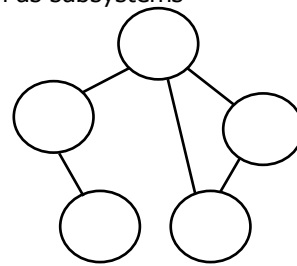
9

Binding

- Functions bind together when they share common interfaces. Groups of functions that are tightly bound should be considered candidates for physical grouping or potential aggregation as subsystems



Functions tightly bound
A potential subsystem?



Functions loosely bound

High binding must be balanced with low "Coupling"

©2016 Waters Corporation

10

Coupling

Coupling is the degree of connectivity between subsystems

Two subsystems with LOW coupling. A "good" architecture

Two subsystems with HIGH coupling. A "poor" architecture

We should always attempt to reduce the degree of coupling – i.e. minimise the number of interfaces

©2016 Waters Corporation 11

Logical System Architecting

What is it?

- Looking for the natural clusters of functionality that have:
 - High binding
 - Low coupling

Why do it?

- Although there is no formal proof (yet), natural architectures appear to be more robust since they minimise the number of interfaces

When to do it?

- Can be done as soon as a functional (Logical) model is available

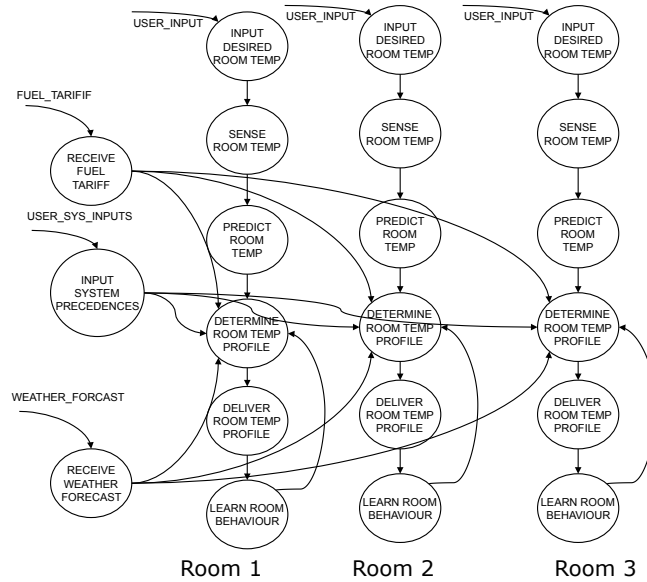
How to do it?

- Grouping and un-grouping functions to find clusters that have the lowest coupling and highest binding by computing the average logical interfaces per candidate subsystem

©2016 Waters Corporation 12

Example Intelligent Heating System

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



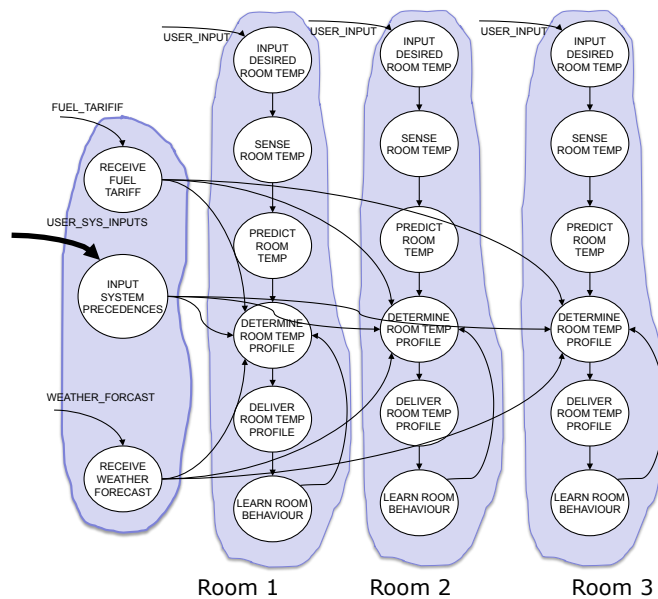
Function Flow
Diagram for an
Intelligent
Heating System

©2016 Waters Corporation

13

Zone Based Architecture

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



Zone Based
Architecture

24/4 = 6
interfaces
per potential
subsystem

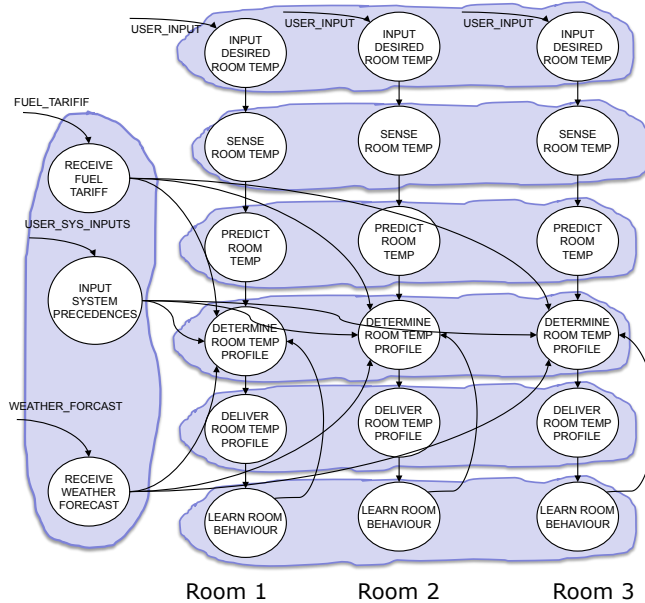
©2016 Waters Corporation

14

Common Purpose Architecture

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Common Purpose Architecture

60/7 = 8.57 interfaces per potential subsystem

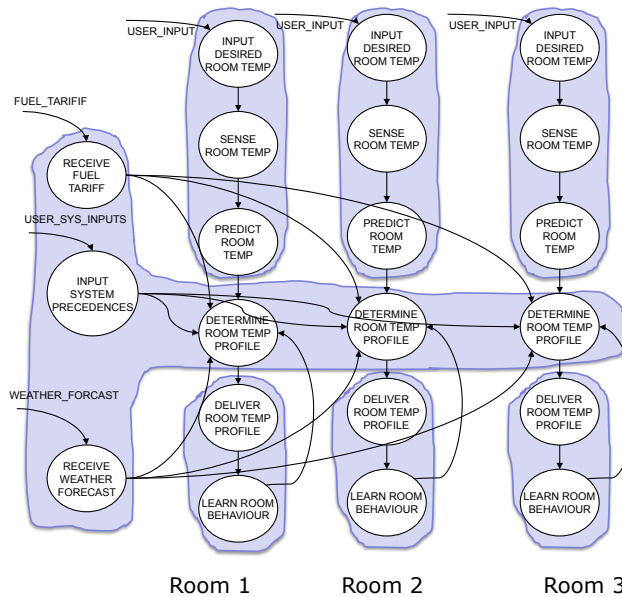
©2016 Waters Corporation

15

Minimum Coupling Design

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Minimum coupling architecture

24/7 = 3.43 interfaces per potential subsystem

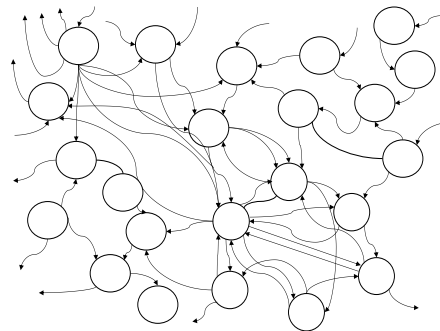
We still have to reconcile this architecture with the chosen technology

©2016 Waters Corporation

16

Grouping in Practice

- This example illustrates the principle of architecting, but for other systems?



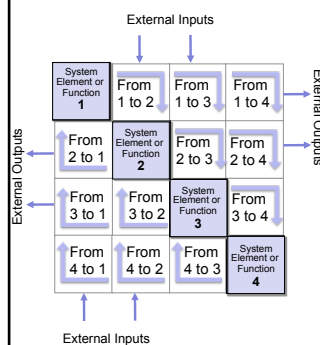
What would be the best logical architecture for this system?

- A very useful tool is N^2 Analysis (Also called Design Structure Matrix DSM)

©2016 Waters Corporation

17

N^2 Analysis



- What is it
 - A tool to identify the interconnectivity of a system
 - It is particularly useful in system design to assess the degree of binding and coupling in a system
- Why use it?
 - N^2 Analysis can be used to help find the natural architecture for a system
 - N^2 Analysis can help identify where redundancy might be necessary
 - N^2 Analysis can be used to document and manage interfaces
- When to use it?
 - The approach could be used even before technological concept generation to help identify natural clusters of functionality
- How to use it
 - An N^2 chart comprises a matrix in which the functions or system elements are recorded on the leading diagonal. The remaining cells of the matrix are used to signify the presence of interfaces between functions

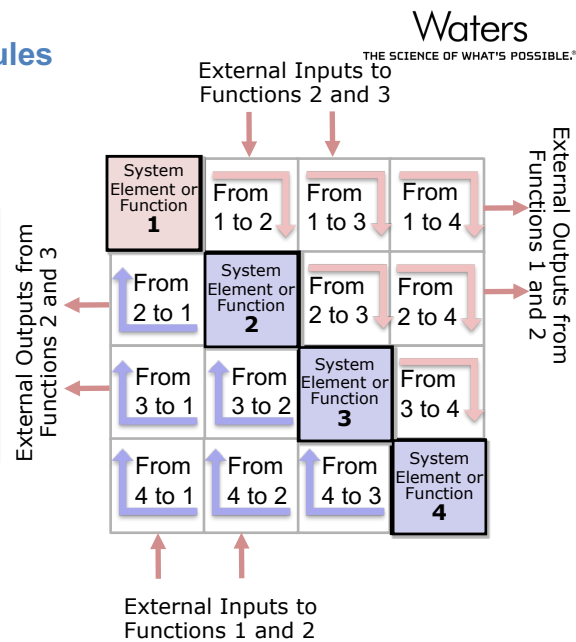
©2016 Waters Corporation

18

Generic N² Diagram Rules

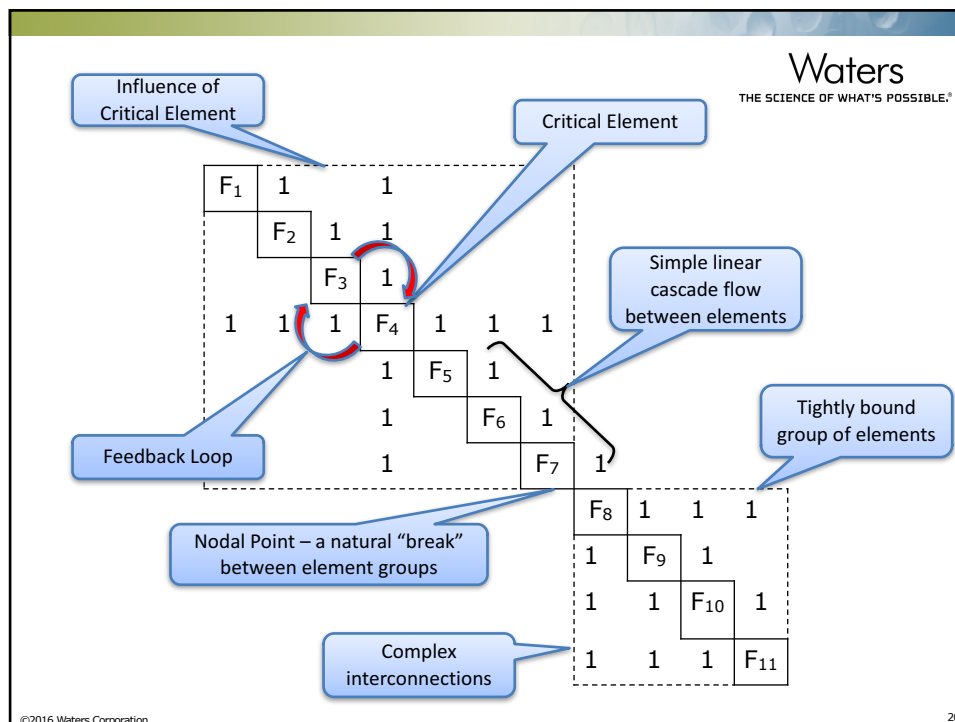
System Elements or Functions:

- are on the diagonal
- have inputs and outputs
- outputs are contained in rows; inputs are contained in columns



©2016 Waters Corporation

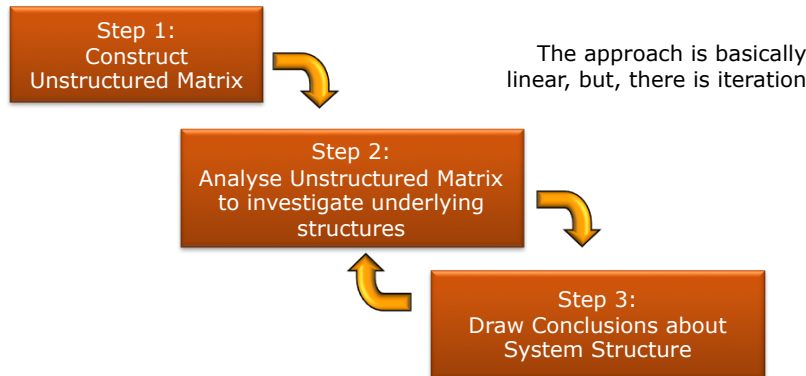
19



©2016 Waters Corporation

20

The N2 Analysis Process

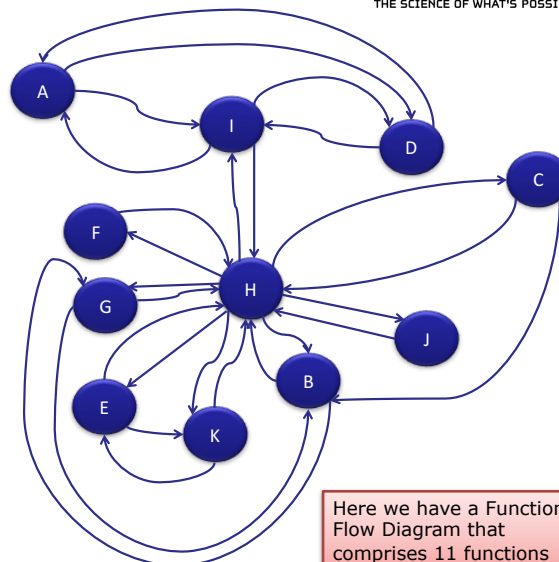


©2016 Waters Corporation

21

Step 1: Construct the Unstructured Matrix

- Step 1, or the input to step 1 is critical to performing a N2 Analysis. You could argue that there is a "Step 0" which is to generate a model of the system of interest that show the components and the interconnections between those components. In terms of the Systems Engineering tools presented in this course the Function Flow Diagram is the obvious choice although there are other Systems Thinking tools that can be used



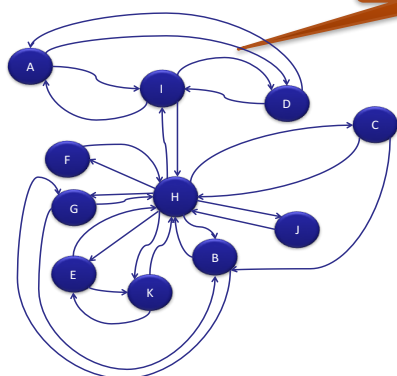
©2016 Waters Corporation

22

Step 1: Example

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



From the FFD we can construct the unstructured N^2 matrix. In this case a "1" is used to signify a flow between two functions

A outputs to D

A			1					1		
	B					1	1			
	1	C					1			
1			D					1		
				E			1			1
					F		1			
	1					G	1			
	1	1		1	1	1	H	1	1	1
1			1				1	I		
							1		J	
				1			1			K

©2016 Waters Corporation

23

Step 2

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- To find the natural structure that exists in most systems we re-order the rows of the N^2 Matrix. To show you the outcome of such a re-ordering exercise, the matrix opposite shows the "hidden" structure that comprises a number of natural clusters of functions (if you are designing a system these could be potential subsystems) and a critical element centred on Function H.
- Moving from the un-clustered matrix to the clustered one can be accomplished by two means. Software packages do exist to perform the task, or it can be achieved by hand. At first glance the "by hand" method may appear somewhat daunting, but with a few simple "tricks" it can be made tractable.

A	1	1								
1	D	1								
1	1	I					1			
			C	1			1			
				B	1		1			
				1	G		1			
		1	1	1	1	H	1	1	1	1
						1	E	1		
						1	1	K		
						1			J	
						1				F

©2016 Waters Corporation

24

The Starting Point

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

The starting point is the unclustered N^2 Matrix. This is often arbitrary as a consequence of the model used.

Should included at least two levels of detail i.e. Diagram 0 and the associated child Diagrams

A			1					1		
	B					1	1			
	1	C					1			
1			D					1		
				E			1			1
					F		1			
	1					G	1			
	1	1		1	1	1	H	1	1	1
1			1				1	I		
							1		J	
				1			1			K

©2016 Waters Corporation

25

Finding Critical Elements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Use row and column sums to look for critical elements

A			1					1			2
	B					1	1				2
	1	C					1				2
1			D					1			2
				E			1			1	2
					F		1				1
	1					G	1				2
	1	1		1	1	1	H	1	1	1	8
1			1				1	I			3
							1		J		1
				1			1			K	2
2	3	1	2	2	1	2	8	3	1	2	

©2016 Waters Corporation

26

Finding Clusters

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

To find the clusters - form the Upper-triangular matrix by placing the lower-triangular entries in their upper reciprocal cells. If the symbol is changed (here from 1 to 0) it becomes easy to see potential clusters.

A			10					10		
	B	O				10	10			
		C					10			
			D					10		
				E			10			10
					F		10			
						G	10			
							H	10	10	10
								I		
									J	
										K

©2016 Waters Corporation

27

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Functions A, D and I are tightly bound.

Move row containing D to just below row containing A.

Move row containing I to just under row containing D.

A			10					10		
	B	O				10	10			
		C					10			
			D					10		
				E			10			10
					F		10			
						G	10			
							H	10	10	10
								I		
									J	
										K

©2016 Waters Corporation

28



THE SCIENCE OF WHAT'S POSSIBLE.®

A similar situation
for B, G and H

A			10					10		
	B	O				10	10			
		C					10			
			D					10		
				E			10			10
					F		10			
						G	10			
						H	10	10	10	
								I		
									J	
										K

©2016 Waters Corporation
29



THE SCIENCE OF WHAT'S POSSIBLE.®

A similar situation
for E, H and K

A			10					10		
	B	O				10	10			
		C					10			
			D					10		
				E			10			10
					F		10			
						G	10			
							H	10	10	10
								I		
									J	
										K

©2016 Waters Corporation
30

Step 3: Draw Conclusions

- Step 3 is perhaps the most difficult one because it requires us to make decisions.
- The Structured N2 Matrix opposite certainly demonstrates that there is structure inside the system problem described by the Function Flow Diagram
- If this was a system design problem, the N2 Matrix can be used to explore potential system architectures.

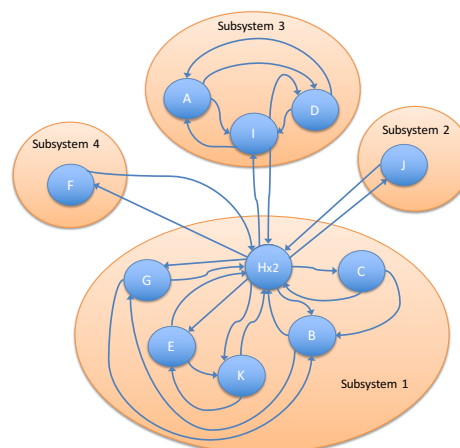
A	1	1									
1	D	1									
1	1	I				1					
			C	1		1					
				B	1	1					
				1	G	1					
		1	1	1	1	H	1	1	1	1	1
						1	E	1			
						1	1	K			
						1			J		
						1				F	

©2016 Waters Corporation

31

A Potential System Architecture

- For example the "grouped" Function Flow Diagram shows such a potential architecture, where the various functions have been allocated to subsystems.
- Most system problems (whether designed, adapted or evolved) happen at the interfaces. It therefore follows that if the system has fewer interfaces there should be fewer issues.
- This of course has to be taken with a large pinch of common sense! Reducing the number of interfaces and the expense of interface complexity may not be a good thing.

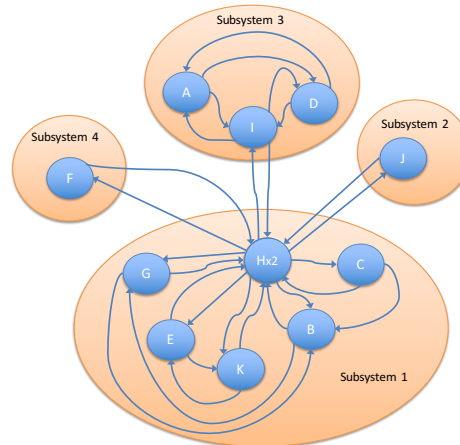


©2016 Waters Corporation

32

Handling Critical Functions

- Function H is critical, and its failure, or degraded performance, will potentially affect the performance of the whole system.
- Context is all important here, in this example we don't really know what the system problem is, however, system elements like H point towards where redundancy may be required.
- In hardware-based systems this may mean duplicating the critical components.
- In Human Activity Systems, the critical element is often an individual. It is important to note from a systems viewpoint, its not the actual individual that is important, but the function they perform that is critical. Again, duplication may be the answer, or at least having a "deputy" or assistant.



Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

©2016 Waters Corporation

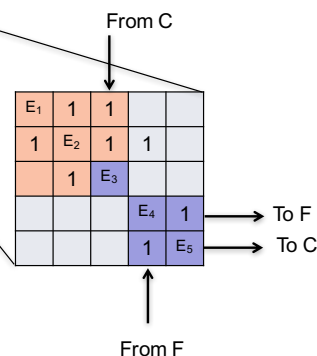
33

Applying Top-Down N2

A	1	1			
1	B	1			
1	1	C	1	1	1
		1	D		
		1		E	1
		1		1	F

Structure that we see at a high level may have further structure at a lower level

Managing complexity is always an issue with N² Analysis. If based on Function Flow Diagrams then the advice is Diagram 0 and the Child Diagrams: - 7x7 = 49x49 Matrix!



©2016 Waters Corporation

34

Example

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Channel Tunnel Security Subsystem: Unclustered chart that shows the potential interfaces with the strength determined on a 0-9 scale

Damage Control	A				6			6
Interpol		B		3		9	3	3
Active Sensors			C			8	8	
Environmental Sensors				D		8		
Customs		3			E	3		7
Baggage Inspection				6	F		1	3
Safety Control						G		
Rail Operations						9	H	
Intelligence							I	
Logistics	2						J	
Immigration		6		1		7	K	6
Local Police		6		2		9	L	
Emergency Services								M
Security			1			7	7	
Operations	9	1	2		9	7	5	

©2016 Waters Corporation

35

Example

Baggage Inspection	F	6						
Customs	3	E	1	3	3	7		
Immigration		1	K	6	6	7		
Interpol		3	3	B	3	9		
Local Police		2	2	6	L	9		1
Intelligence					I		9	8
Emergency Services						M		4
Active Sensors					8		C	8
Security			3	7		1	N	8
Operations			4	5	4	1		9
Rail Operations							O	7
Safety Control								9
Environmental Sensors							9	H
Damage Control							2	G
Logistics							9	8

Natural Subsystem – exploit collocation

Critical Node centred on operations – note the two “systems” are coupled through the local police

Partial Critical Node and natural subsystem

©2016 Waters Corporation

36

Exercise – N² Analysis

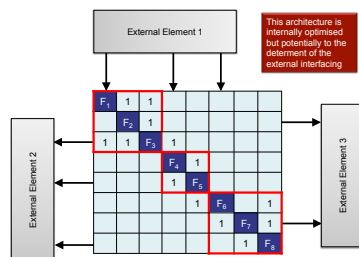
Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

In teams

- Use your Function Flow Diagram to construct an N² matrix for the Case Study
- Investigate the matrix for any natural subsystems or critical nodes
- Identify potential candidate logical architectures

Time: 15 minutes



©2016 Waters Corporation

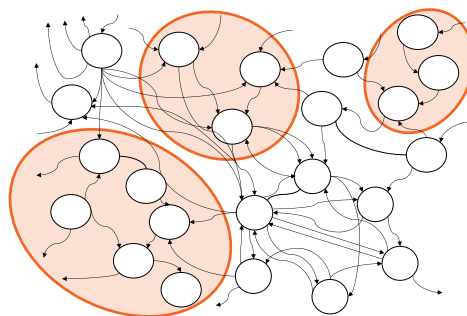
37

Using N2 Analysis

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- N2 Analysis can help determine the best logical architecture and, while this may not be physically realisable, it is often worthwhile modelling the best logical architecture for future reference and analysis



Note that there may be more than one logical architecture – model all the possible candidates at least at highest level

Next Steps

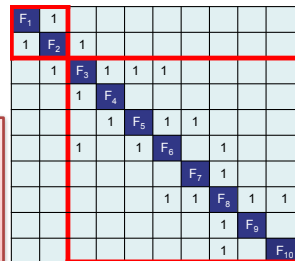
- Reconcile the logical architectures with the implementation choices
 - Technological choices for functionality
 - Use of standard interfaces
 - Design and Define system interfaces

©2016 Waters Corporation

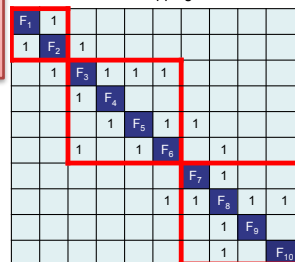
38

Architecture Choices

There is always choice and decisions about the system architecture should be delayed until the technological choices have been identified



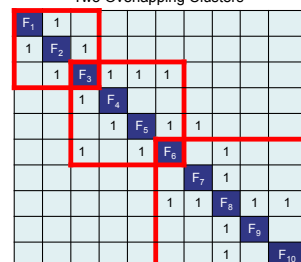
Two Non-Overlapping Clusters



Three Non-Overlapping Clusters



Two Overlapping Clusters



Three Overlapping Clusters

©2016 Waters Corporation

39

External Versus Internal Interfaces

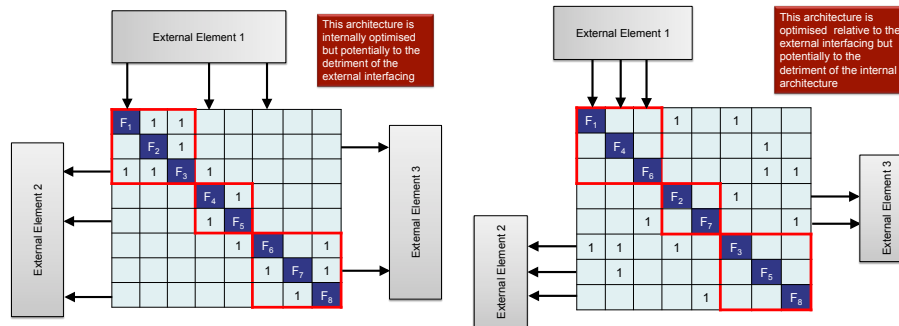
Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- System external interfaces are defined by the higher-level system design and are often fixed and described as requirements on the system design – BUT:
 - Internal interfaces are different since they are created as part of system architecting. This is a powerful opportunity for the development team to optimise either the
 - Local system design: in this case the system architecture is selected to optimise the performance of the system potentially to the detriment of the bigger system to which it belongs
 - The higher level system design: in this case the performance of the local system is sacrificed to yield improved performance at the bigger system level

©2016 Waters Corporation

40

External Versus Internal Interfaces



Lower-level architectures are constrained by the higher-level architecture. Even though there may well be a better way, because at the higher-level, the lower-level architecture is sub-optimal.

Architecture and System Meta-solution

- There is a connection here between the levels of Systems Architecture and the concept of the system Meta-solution.
- A system Meta-solution was defined as an organization's current preferred solution to a systems problem (Operational Requirements)
- We can now be more precise and talk about the Meta-solution in terms of the current preferred architecture to a systems problem
- In a complex system there will be levels; system, subsystem, sub-subsystem, etc. The architectures cascade, and is the reason why changing an architecture at one level often has an impact at the higher level
- Self-organizing systems (Human Activity Systems) will "try"¹ different architectures over time eventually settling on the one that works! (i.e. make the bigger system optimal)

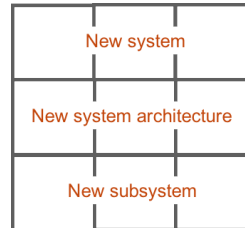
¹ "try" is used to imply an ad-hoc approach. What we should really do is experiment scientifically to find the optimum

Architecture and System Meta-solution

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Accepting a particular Meta-solution (architecture) means that changes should take place within the constraints of the higher system architecture, otherwise sub-optimisation can occur.



- Radical innovation often requires the higher-level architecture to change to accommodate the lower-level change

"Any intelligent fool can make things bigger and more complex. It takes a touch of genius – and a lot of courage – to move in the opposite direction"

E F Schumacher

©2016 Waters Corporation

43

Interfaces

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Definition

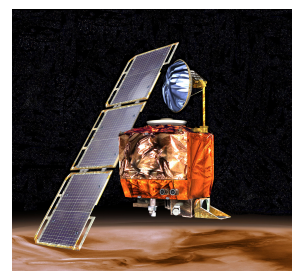
Noun

- A point where two systems, subjects, organizations, etc. meet and interact
- A device or program enabling a user to communicate with a computer

Verb

- Interact with (another system, person etc.)
- Connect with (another computer or piece of equipment) by an interface

In Systems Engineering an interface is like any other system element, they have purpose (function) and context (constraints) but are often overlooked. In consequence most system problems occur at the system interfaces



The Mars Climate Orbiter on September 23, 1999, was lost as the spacecraft went into orbital insertion, due to ground-based computer software which produced output in pound-seconds (lbf s) instead of newton-seconds (N s) specified in the contract between NASA and Lockheed. The spacecraft encountered Mars on a trajectory that brought it too close to the planet, causing it to pass through the upper atmosphere and disintegrate.

©2016 Waters Corporation

44

Need for Managing Internal Interfaces

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- System architecting and system concept design creates a set of subsystems AND the interfaces between them
- 'Ownership' of each subsystem is usually obvious as it is assigned to an individual or group as it is defined
- However, the subsystem interfaces need special attention since each subsystem owner may assume:
 - the other party is responsible for the interface, or
 - each party makes its own assumptions about the interface requirements.
- In both cases, the project runs the risk of future interface incompatibility, since the subsystems mature their concepts (and their interfaces) independently
- The solution is to explicitly identify the owner of every interface

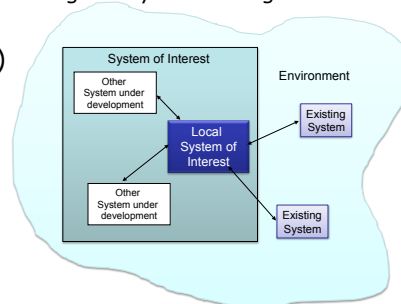
©2016 Waters Corporation

45

Key Interface Documents

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Interface Definition Document (IDD) - defines interfaces to an existing system. It states what interfaces exist that are design constraints. Can be anything, such as mass, type of connector, EMI...
 - Owned by manager of the system with which you want to interface
 - Probably not going to change
- Interface Requirement Document (IRD) - defines interfaces for two developing systems or subsystems. Includes both physical and functional interfaces
 - Jointly managed (NEEDS ONE OWNER) and signed by the managers of the two systems in development.
- Interface Control Document (ICD)
 - defines the design solution for the physical interface



©2016 Waters Corporation

46

Interface Design

- Interfaces should be designed such that
 - Subsystems are as independent as possible. As a result these subsystems will be easier to develop and test independently and when they are integrated there will be fewer problems with compatibility.
 - Group like functions within a subsystem. This reduces the exchange of common data and focuses development expertise.
 - Use standard interfaces, e.g., mechanical, electrical and data wherever possible

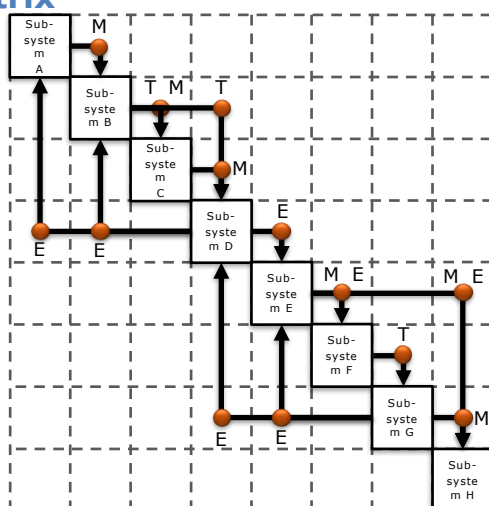
A square peg in a round hole



©2016 Waters Corporation

47

Managing Interfaces Using a N² Matrix



Key

- M - Mechanical
- T - Thermal
- E - Electrical
- - Interface

N² Matrices can also be used to record interfaces. Once an interface has been designed it can be captured on a N² Matrix. They provide a convenient summary of the interfaces. They can therefore be used to manage the interfaces during the life of a system

©2016 Waters Corporation

48

Summary: System Architecture, N2 and Interfaces

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

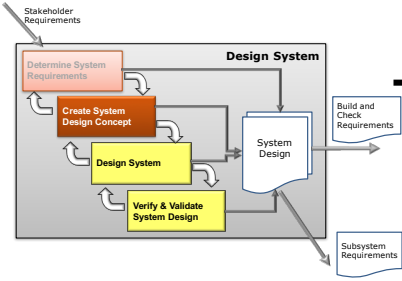
- The choice of architecture has a fundamental impact on system performance
- Architecture is NOT just the layout (how the resources are organised) it is also about control (how the resources are used)
- System architectures are designed and the golden rule is to minimise interfaces
- System interfaces are defined, distributed and managed like other system elements.
- Internal interfaces, since they are created as the system is decomposed and synthesised, can be optimised by the development team.
- In developing interfaces group like functions to minimise the number and complexity of the interfaces, keep them simple and always consider the use of standard interfaces.
- N2 Matrices are a powerful interface identification and management tool. They are used to:
 - Capture the existence and nature of an interface
 - Highlight input and output assumptions and requirements
 - Demonstrate where there are feedback loops between subsystems
 - Identify candidate functional allocations to subsystems – system architecting





THE SCIENCE OF WHAT'S POSSIBLE.®

Create System Design Concept Technology Concept Generation



This Session:

- Explains the concepts and principles of Technological Concept Generation
- Describes and provides opportunities to practise a number of powerful but simple Creative Thinking Tools to generate potential functional solutions
- Describes and provides an opportunity to create candidate whole system concepts via Function Means Analysis

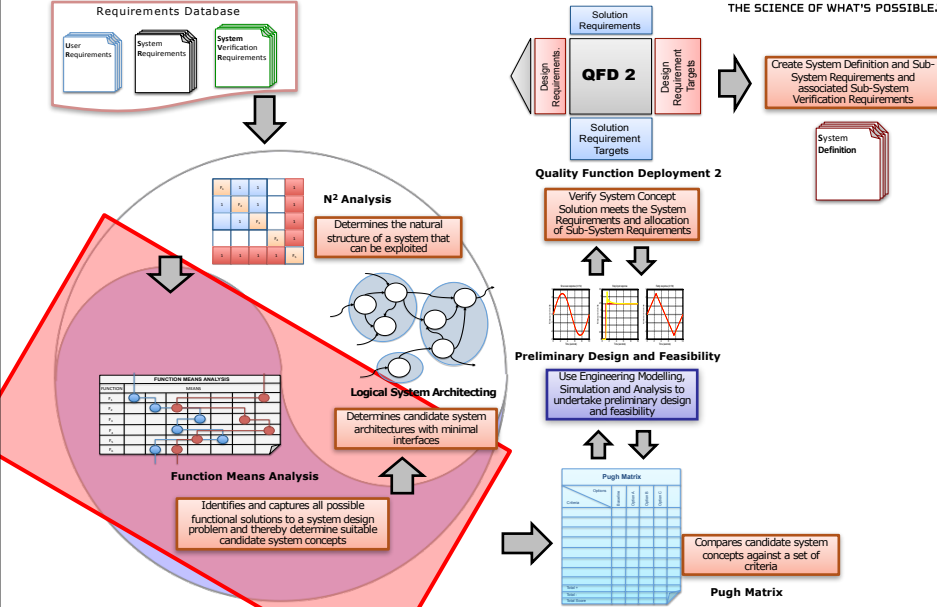
©2016 Waters Corporation

1

Process and Tools

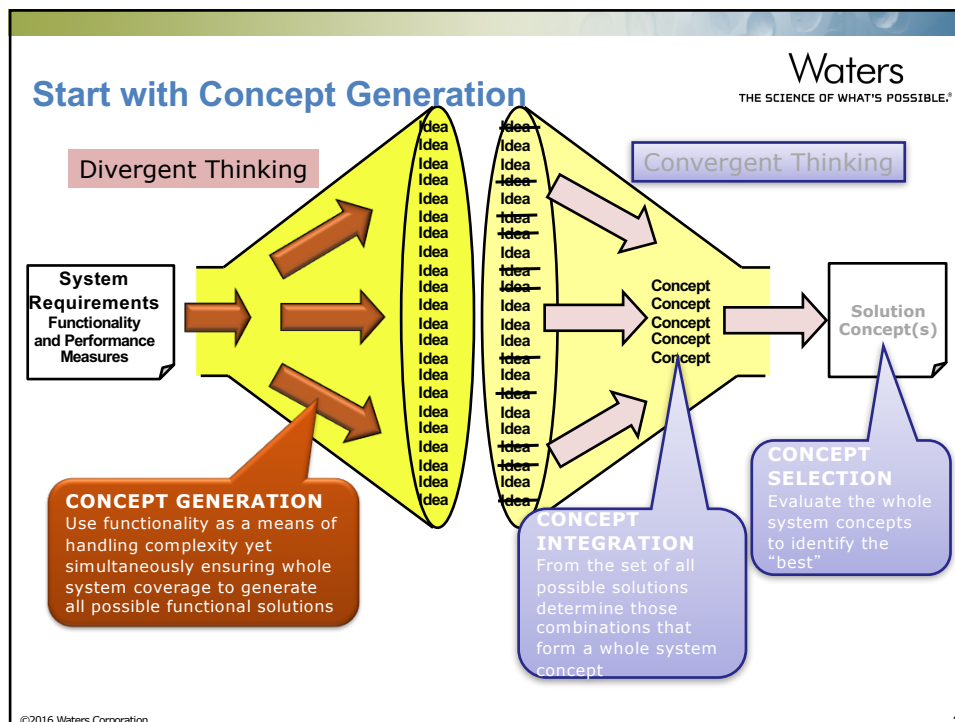
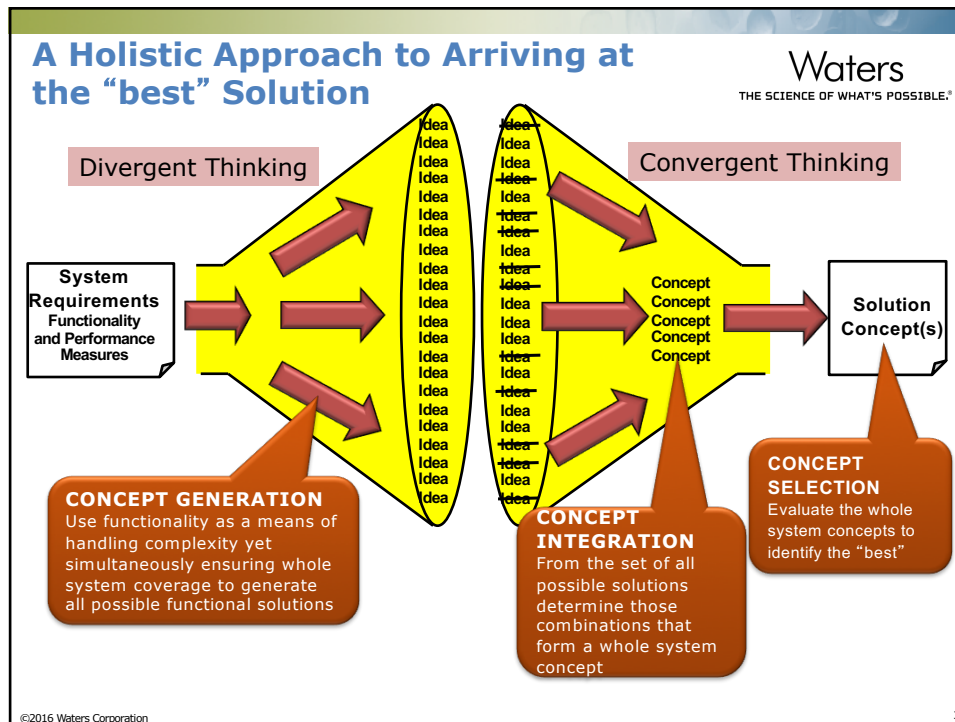


THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

2



Tools for Technology Concept Generation

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Concept generation is a divergent thinking activity – it is about being creative, radical and innovative, but in a disciplined and structured manner
- Achieving “structured innovation” requires two phases:
 - Unconstrained ideas generation where teams members have an open mind and are willing to explore options
 - Logical ideas structuring where the team needs to assemble the ideas in a logical framework
- There are many tools to assist in the unconstrained ideas generation – these include:
 - Brainstorming and derivatives
 - Similarities and Differences
- There are very few tools to help structure ideas – the main tool here is:
 - Morphological Analysis or Function Means Analysis

©2016 Waters Corporation

5



©2016 Waters Corporation

6

Creativity?

- What is creativity?
- Are you creative?
 - Many people believe that only a handful of geniuses are capable of making creative contributions – it just isn't true.
- Creative thinking is a norm in human beings
 - – we're just good at turning it off
- What are your barriers to being creative?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

7

Barriers to Creativity

- We are surrounded by barriers to creative thinking
 - Human limitations
 - Personal “baggage” and experience
 - Organizational and sector culture
 - Internal structures and politics

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

8

Human Limitations: We are Blinded to New Ideas

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Because of -
 - Familiarity (The United States of America)
 - Assumptions (The Short Story)
 - Perceptions (The Old Lady/Young Lady)
 - Psychological Inertia (Space ship bulbs)

©2016 Waters Corporation

9

Analytical Thinking

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- How many different ways can the first five letters in the alphabet be arranged?
- Do the sequence of four words, "triangle, glove, clock, bicycle," corresponds to this sequence of numbers "3, 5, 12, 2."
- What is the next number in the following sequence of numbers: 2, 6, 14, 30...

We use Logic to find an answer

©2016 Waters Corporation

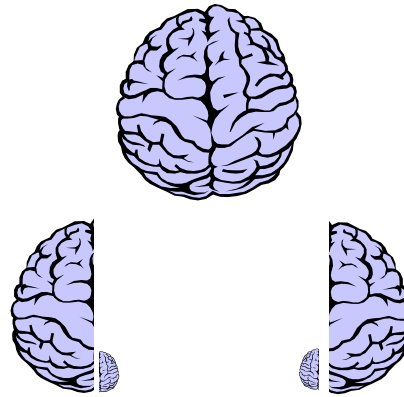
10

Your Brain

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The human brain comprises two hemispheres, for right handed people
 - The LEFT hand side has been shown to be concerned with logical thinking
 - The RIGHT hand side of the brain has been shown to be concerned with artistic creative thinking
- Excessive training can cause one side to dominate
 - Engineers and Scientists – the logical Left
 - Humanities and Arts – the creative Right



Size representation of brain usage for a typical engineer

Size representation of brain usage for a typical artist

©2016 Waters Corporation

11

Breaking Down the Barriers

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- We all have the potential to be creative – BUT
 - We must break down the barriers
- Breaking the barriers demands several approaches
 - Using lateral thinking (solving problems by illogical means) as well as logical thinking
 - Using tools to promote creative thinking and remove organizational barriers
 - Being disciplined and systematic

©2016 Waters Corporation

12

Creative Thinking in Practice

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Characteristics
 - Idea fluency – LOTS of ideas and associations
 - Variety and Flexibility
 - Originality
 - Elaboration
- Creative thinking tools
 - Brainstorming
 - Negative Brainstorming
 - Affinity Diagrams
 - Similarities & Differences

Brainstorming

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- What Is it?
 - Brainstorming is a team based creative thinking method
- When to use it?
 - Brainstorming can be used to tackle all types of problem
- How to do it?
 - The team is given a problem for which they initially generate their own solutions or ideas
 - The group members read out their solutions or ideas, in a round robin fashion, which are collectively recorded. During this initial phase it is important that no criticism is levelled at any of the solutions or ideas – no matter how ridiculous they may appear. These “ridiculous ideas” may blossom into sound solutions or trigger other members of the team to generate a sound solution
 - Once the first stage is complete the list of solutions or ideas can then be evaluated to identify the “best”

Brainstorming Guidelines

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The group should be as diverse as possible
- The group should be non-hierarchical (although one person should take the lead)
- No criticism should be allowed during the early non-evaluative session
- Freewheeling is encouraged
- Concentrate on getting the maximum number of ideas
- Record every idea, even repetitions
- Incubate all the ideas, do not reject anything out of hand
- Ideas/solutions should be kept short and concise
- An independent trained facilitator is often useful

©2016 Waters Corporation

15

Brainstorming Derivatives

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- Brainstorming will work if the rules are followed. It often fails to deliver because:
 - of dominant or submissive personalities within the team
 - “low energy” in the team
- There are alternative tools that can be used in such situations
 - Negative Brainstorming
 - Affinity Diagrams



Already used

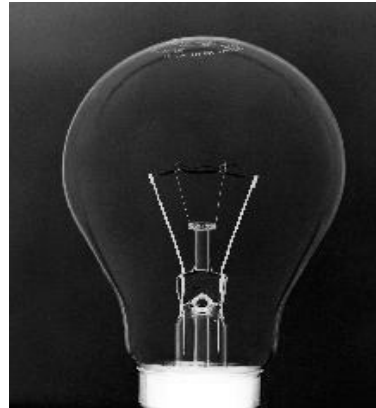
©2016 Waters Corporation

16

Negative Brainstorming

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- What is it?
 - A simple alternative to brainstorming in which the problem is “reversed” to consider negative solutions which can be reviewed and reversed to generate new ideas
- How to do it?
 - Identify the problem for which a solution is sought and generate a “negative” statement
 - How can we ensure meetings are a disaster?
 - How can we ensure the system produces defects?
 - Brainstorm the solutions as per brainstorming
 - Review the ideas and reverse those that have potential



©2016 Waters Corporation

17

Similarities and Differences

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- What is it?
 - A divergent thinking approach aimed at getting people away from the ‘baggage’ that inhibits creativity.
 - The approach is based around comparing the ‘problem’ with an entirely unrelated item or situation and by comparing similarities and differences develop new solutions
- How to do it?
 - Identify an item for comparison. For product based systems try a nature based object (bird, tree etc.), for process based systems try a natural process (river, digestion etc). Do not think too hard about this!
 - List all the similarities
 - List all the differences
 - Identify similarities and differences that may lead to new ideas



©2016 Waters Corporation

18

Example: A Nail (Atherton Nail Co)

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The Atherton Nail Co realised that they needed new products
- A team was assembled to generate some new ideas using Similarities & Differences
- They compared a Nail with Grass (first suggestion from team)
 - Similarities
 - Pointed
 - Slender
 - Many types
 - Found in large numbers
 - Different lengths
 - Head

©2016 Waters Corporation

19

Differences

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Nail

- Various colours
- Metal
- Hard
- Stiff
- Strong
- Straight
- Sharp-ended
- Inorganic
- Non-biodegradable
- Factory produced
- Corrodes
- Manufactured
- No Roots
- No flowers
- Comes in bags

Grass

- Green
- Non-metal
- Soft
- Flexible
- Weak
- Curved
- Sharp-edged
- Organic
- Biodegradable
- Seasonal production
- Rots
- Grown
- Roots
- Flower heads
- Comes in lawns

©2016 Waters Corporation

20

Outcome

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- From these lists conclusions were drawn about nails that are:
 - Non-Metallic
 - Soft, flexible, weak
 - Curved
 - Sharp-edged ✓
 - Grown self-replenishing
 - Biodegradable
 - Rooted ✓
 - Lawn of nails ✓
- It is the differences that the new ideas come from. Listing the similarities is important. It helps the team think laterally
- If the method does not generate any new ideas try a different unrelated item

©2016 Waters Corporation

21

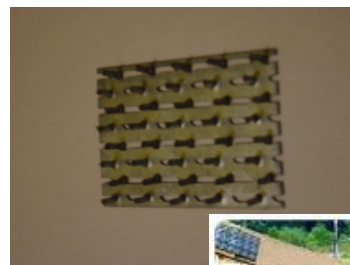
Outcomes

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- “Rooted Nail”



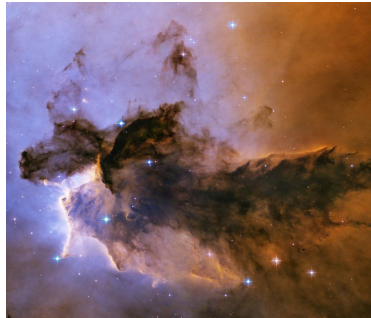
- “Lawn of Nails”



©2016 Waters Corporation

22

Exercise – Similarities and Differences



- In teams
 - Perform a Similarities and Differences on the case study project

- Time: 20 minutes

Summary

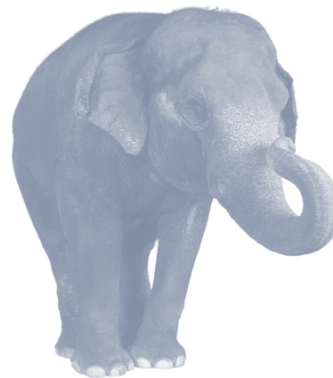
- The tools presented are all concerned with overcoming
 - Baggage
 - Assumptions
 - Perceptions
 - Psychological Inertia
 - Personalities and organizational culture
- In looking for creative solutions be prepared to try several of the tools
- Do not worry if a tool does not yield results for a particular problem – try another



Logical Ideas Structuring

Ideas Generation for a Holistic Solution

- Even for the simplest of systems it is not possible to move from requirements to complete system solution – we are unable to “juggle” all the options and permutations in our head
- The problem has to be broken up into parts and solutions found for each part (using divergent thinking) and the whole assembled to give a complete system solution (using convergent thinking)
- It is obviously important that no part-solution is overlooked. Hence the division of the of the whole system solution must guarantee completeness – it must be holistic
- This can only be achieved through consideration of the system functionality that has been deduced during Determining Requirements
- A simple but powerful tool to accomplish this is a derivative of Morphological Analysis known as Function Means Analysis



Function Means Analysis (Stage 1)

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- What is it?
 - Function Means Analysis (FMA) is a simple structured tool for generating, capturing, selecting and documenting system design concepts.
- How to do it
 - Function Means Analysis has a number of steps that fall into two stages:
 - Stage 1
 - Concept Generation: a divergent thinking activity where the functions of the system are considered in turn and the means of achieving those functions identified
 - Stage 2
 - Concept Integration and Selection: a convergent thinking activity that identifies whole system concepts and down-selects these to a small number of high potential solutions

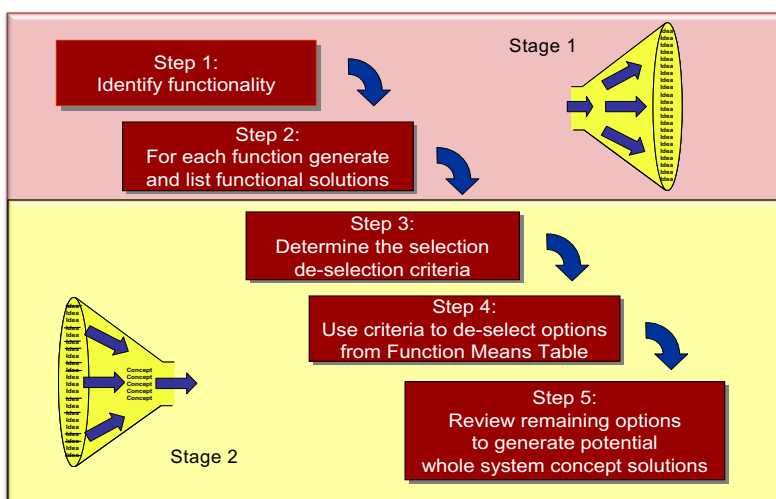
FUNCTION MEANS ANALYSIS TABLE									
FUNCTION		MEANS							
System	Sub-System	Function	Means	Means	Means	Means	Means	Means	Means
Wash Watering	Wash Watering	Deliver Load Culture	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
		Deliver Fabric Type	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
		Deliver Load Weight	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
		Measure DSI	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
Wash Watering	Wash Watering	Measure H2O	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
		Interface to Water	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
		Interface to Power	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
		Interface to Heat	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering	Wash Watering
Load Machine	Load Machine	Load Dirty Items	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Control Access	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Unload Cleaned Items	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Load Detergent/Conditioners	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
Load Machine	Load Machine	Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
Load Machine	Load Machine	Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine
		Load Water	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine	Load Machine

The example above shows a partial Function Means Analysis for the Intelligent Washing Machine

Function Means Analysis Process

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Step 1: Identify Functionality

- To undertake a Function Means Analysis requires knowledge of the functionality of the system. The functionality should be:
 - Complete i.e. encompass the whole of the system
 - At the same level of generality
- These rules are not rigid. Indeed, Function Means Analysis can help to check whether the system functionality is complete. Very few systems are unprecedented and therefore a check can be made to ensure that the current system solution sits within the means identified – if not then perhaps we are missing some functionality
- Once identified the functions should be entered in the Function column of the Function Means Table

Step 2: For each Function Generate Solutions

- Taking each function in turn, the team should identify and record in the table ALL the POSSIBLE means of achieving that function
- Quantity rather than quality is the aim at this point and a good rule of thumb is to aim for 9 ideas per function. This assumes 3 conventional, 3 extreme and 3 with high potential. Ridiculous, silly, impractical ideas should not be shot-down, ridiculed, or ignored – write them down
- If it is difficult to generate solutions to functions:
 - The function may be a pseudo-function like “generate style” these either need rewriting as “proper” functions or removing
 - The function description may be at too high a level and needs to be decomposed into a number of lower level functions for which design options can be explored

Example: Intelligent Washing Machine

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

FUNCTION – MEANS ANALYSIS CHART									
System: Intelligent Washing System		Subsystem		Date 12/4/2000	Author I M Clean	Issue	Reviewed		
FUNCTION	MEANS								
Load Dirty Clothes (and unload)	Front door	Top door	Removable bin/integral drum	Hole and chute in bedroom	Still Wearing Clothes – Walk Through Cabinet	Feed items one at a time (like photocopier)	Filling cabinet door load		
Detect fabric type	Bar coded items & reader in door	Bar coded items & hand held reader	Bar coded items & reader in bin/drum	RFID tags in items & reader in machine	Internal optical sensor	Coded bins/integral drum	Tell system as loading	None use lowest common cycle	
Detect dirt	Smell	Internal optical sensor	User input	Assume medium dirt	User defined quantity	Learn items and likely dirt	None		
Detect load weight	Load cell in machine	Load cell in bin	Don't detect load assume max or average	User informs M/C of load	Measure drum turning torque	None			
Dispense Detergent/ conditioner	Drawer and user inputs quantity	Bigger drawer with draw-off of detergent (lasts several washes)	Internal storage and auto dispense	No detergent or conditioner – "ultrasonic"					
Wash Items	Horizontal Drum agitation	Vertical Drum agitation	Angled Drum agitation	Ultrasonic bath	Jet spray	Free swimming agitators			
Rinse Items	Cold Water soak	Hot Water soak	Jet spray						
Extract Excess Water	Spin	Mangle	Suck/vacuum pump	Squeeze	Wring	Blow Hot air	Attract water through ionisation		
Determine Wash cycle	Look up table	Transfer function	Neural network	Fuzzy logic	Expert system	Clockwork programme	Fixed programme	None – use standard wash	
Drain Waste Water	Pump	Filter and pump	Filter & recycle	Siphon	Gravity	Vacuum suck			

For each function identify and record potential design solutions

For each function identify and record potential design solutions

©2016 Waters Corporation

31

Top Down

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Function Means Analysis can be applied in top down fashion – once a design decision has been made at the higher level, it can be applied again at the next level down.

FUNCTION – MEANS ANALYSIS CHART										
System: XYZ Building		Subsystem		Date 12/4/2001	Author A C Clark		Issue		Reviewed	
FUNCTION	MEANS									
Control Access (to Building)	None – open hole	Side Hinged door	Top hinged door (hopper hung)	Side Sliding door	Upward sliding door	Roller door	Force-field door	Rotating door	Inis door	

For example one of the functions of a building is “control access” for which there are many choices – once the choice has been made it can be subject to its own Function Means Analysis

At higher levels choices can be captured as words. At lower levels pictures/diagrams are more useful

FUNCTION MEANS ANALYSIS									
System: Building	Subsystem: Control Access	Date 12/4/2001	Author A C Clark	Issue	Reviewed				
FUNCTION	MEANS								
Seat Door									
Hinge door									
Control Opening									

©2016 Waters Corporation

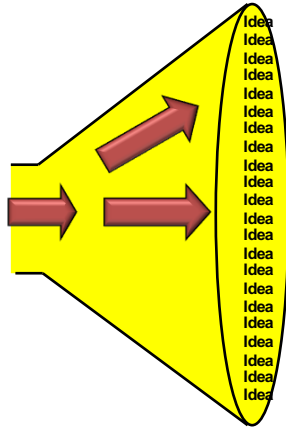
32

Exercise – FMA (Part 1)

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

System
Requirements
Functionality
and Performance
Measures



- In teams
 - Perform a Function Means Analysis for the case study project to generate solutions
 - Consider using several of the creative thinking tools
- Time: 30 minutes

©2016 Waters Corporation

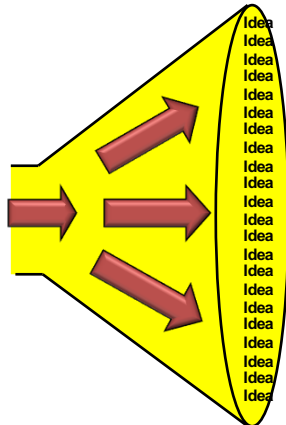
33

Exercise – FMA (Part 1)

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

System
Requirements
Functionality
and Performance
Measures



- In teams
 - Perform a Function Means Analysis for the case study project to generate solutions
 - Consider using several of the creative thinking tools
- Time: 30 minutes

©2016 Waters Corporation

34

Summary FMA (Part 1)

Waters

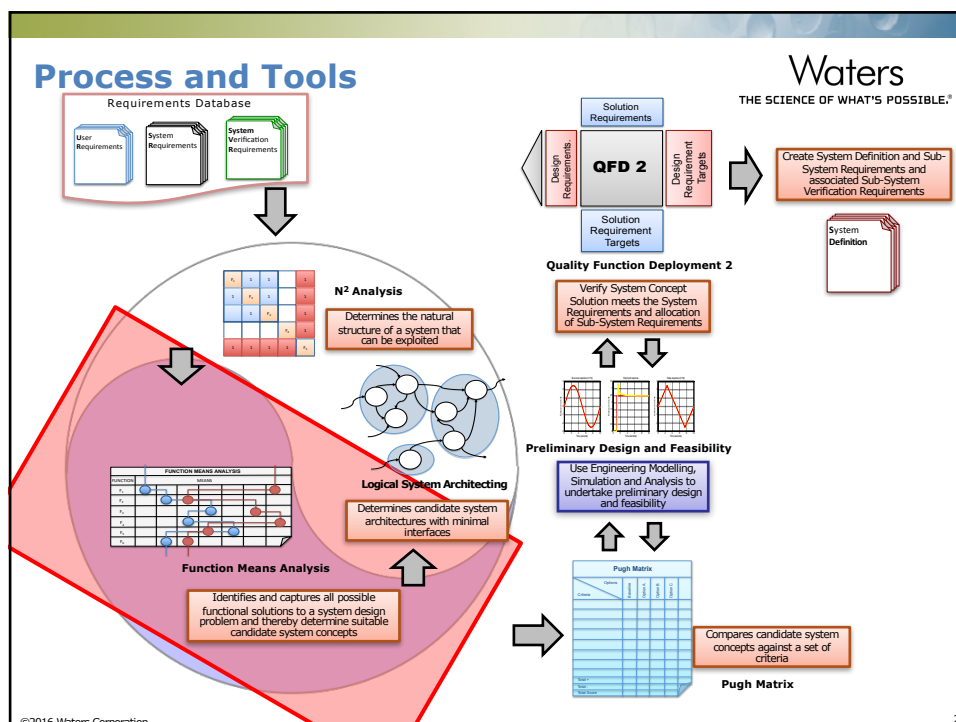
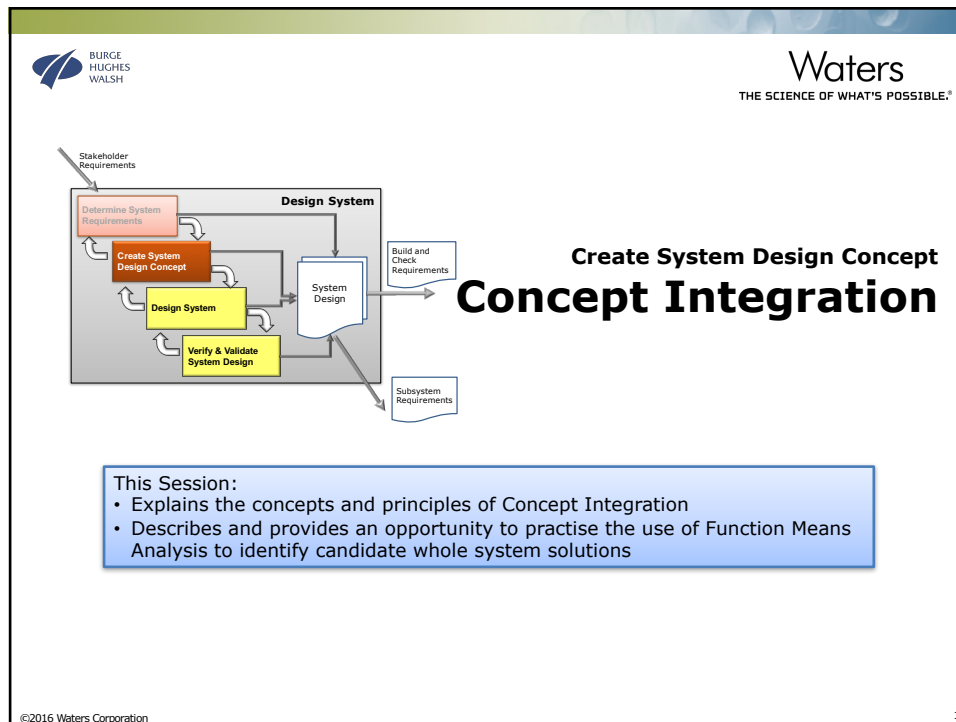
THE SCIENCE OF WHAT'S POSSIBLE.®

■ Pros

- Simple to use
- Allows the design problem to be logically broken down into manageable elements for which solution options can be easily found
- When used with an open mind is extremely powerful for identifying and recording “all” possible design options
- The Function Means Table allows all the design options to be displayed simultaneously allowing for easier review
- Applicable at any level of design (system, subsystem to component)

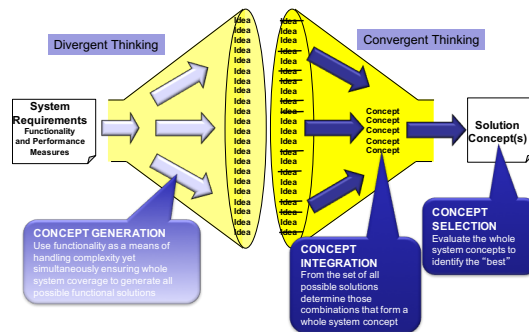
■ Cons

- Getting the functions at the correct level of abstraction in order to explore solution choices
- The outcome can be a significant number of options which must be combined together
- Sometimes viewed as too simplistic and therefore not a serious tool



Introduction to Concept Integration

- The first stage of Function Means Analysis provides a large number of potential functional solutions. The task now is to:
 - Generate a number of potential complete system solutions frequently referred to as design concepts or concept system solutions – this is called Concept Integration



©2016 Waters Corporation

3

Concept Integration

- The purpose of Concept Integration is to arrive at a number of potential whole system solution concepts
 - Remember, we often use the word integration to mean just the final “physical” integration of a system – but the decisions we are about make are fundamentally integration decisions, and poor choices at this point in time will have detrimental effects on the physical integration of the system
- Assuming that the concept generation has been done by Function Means Analysis, there are a number of strategies for Concept Integration
 - Identify the “best” means for each function
 - Cannot guarantee an optimal system solution
 - Consider every permutation from the Function Means Analysis
 - Will take too long
 - **Reduce the number of options and develop a limited number of potential whole concept solutions**
- This last strategy can be accomplished using the Function Means Chart

©2016 Waters Corporation

4

THE SCIENCE OF WHAT'S POSSIBLE.®

- 5

THE SCIENCE OF WHAT'S POSSIBLE.®



Step 3: Determine Selection/De-selection Criteria

- What is required at this step are some key criteria that can be used to quickly eliminate or promote the solutions documented in the Function Means Table
- The aim is to remove the obvious “sore thumbs”.
- A good starting point for the criteria is the primary WHATs from the QFD 1. These are customer facing and therefore deselecting functional solutions that do not meet these will ensure that only “compliant” ideas are carried forward
- The general recommendation here is between 3 and 6 criteria. Too many criteria can make the process time consuming. The drawback with few criteria is that those that are selected have a significant impact on the selected system solution – the wrong criteria lead to the wrong solution

Example: Intelligent Washing Machine

- Step 3: The de-selection criteria and order of importance
 - Easy to use (order winner)
 - Reliability (order qualifier – losing sensitive)
 - Attractive & distinctive (order winner)
 - Wash Performance (order qualifier – losing sensitive)
- Note - having too many de-selection criteria can be counter productive. The aim initially is to remove the obvious non-runners quickly
- Note – The criteria are not just the order winners – also need to consider those losing sensitive criteria – those that unless we meet the expected level will lose orders

Step 4: Use the Criteria to De-select from the Table

- The de-selection criteria, determined in the previous step, are now systematically applied to remove candidate functional solutions
- The aim here is to remove approximately 50% quickly, i.e. the obvious “sore thumbs”
- As functional solutions are removed a note should be made of the criteria and reasoning that led to their removal
- Over arguing at this point is counter productive – functional solutions that generate “excessive” debate should be retained

Example: Intelligent Washing Machine

Step 4: Apply criteria to de-select solutions

FUNCTION – MEANS ANALYSIS CHART									
System: Intelligent Washing System	Subsystem			Date 12/4/2000	Author I M Clean	Issue	Reviewed		
FUNCTION	MEANS								
Load Dirty Clothes (and unload)	Front door	Top door	Removable bin/integral drum	Hole and chute in bedroom	Still Wearing Clothes – Walk Through Cabinet	Feed items one at a time (like photocopier)	Filing cabinet door load		
Detect fabric type	Bar coded items & reader in door	Bar coded items & hand held reader	Bar coded items & reader in drum	RFID tags in items & reader in machine	Internal optical sensor	Coded bins/integral drum	Tell system as loading	None use lowest common cycle	
Detect dirt	Smell	Internal optical sensor	User input	Assume medium dirt	User defined quantity	Learn items and likely dirt	None		
Detect load weight	Load cell in machine	Load cell in bin	Don't detect load assume max or average	User informs M/C of load	Measure drum turning torque	None			Easy to use
Dispense Detergent/conditioner	Drawer and user inputs quantity	Bigger drawer with draw-off of detergent (lasts several washes)	Internal storage and auto dispense	No detergent or conditioner – “ultrasonic”					Reliability
Wash Items	Horizontal Drum agitation	Vertical Drum agitation	Angled Drum agitation	Ultrasonic bath	Jet spray	Free swimming agitators			Attractive
Rinse Items	Cold Water soak	Hot Water soak	Jet spray						Wash Perf.
Extract Excess Water	Spin	Mangle	Suck/vacuum pump	Squeeze	Wring	Blow Hot air	Aspirate water through ionisation		
Determine Wash cycle	Loadable table	Transfer function	Neural network	Fuzzy logic	Expert system	Clockwork programme	Fixed programme	None – use standard wash	
Drain Waste Water	Pump	Filter and pump	Filter & recycle	Siphon	Gravity	Vacuum suck			

The de-selection should be looking for the obvious “sore thumbs”

Step 5: Review Remaining Options to Generate Potential Whole System Concept Solutions

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- With the “sore thumbs” removed the task now is to determine potential whole system concepts
- A simple approach here is to plot routes through the table that represent complete candidate system concept solutions
- For any given function as many means as necessary can be selected for a particular whole system concept – note though if several solutions are necessary it could be a consequence of the function being too abstract! In such situations it is worth considering decomposing the high-level function into lower-level functions
- Selecting the “routes” can be based on technologies that
 - Are compatible or avoid incompatibilities
 - Support the “best” logical architectures
- The aim should be to generate as many whole system concepts as realistically possible – common sense should also prevail

©2016 Waters Corporation

11

Example: Intelligent Washing Machine

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Step 5: Review Chart for candidate whole system concept solutions

FUNCTION – MEANS ANALYSIS CHART							
System: Intelligent Washing System	Subsystem	Date 12/4/2000	Author I M Clean	Issue	Reviewed		
FUNCTION	MEANS						
Load Dirty Clothes (and unload)	Front door	Removable door			Filling cabinet door load		
Detect fabric type		RFID tags in laundry basket in machine	Internal sensor	Code detect integral drum		None use is common	
Detect dirt	Sensor	Internal sensor		Learn items and likely dirt	None		
Detect load weight	Load cell in machine	RFID tags in laundry basket in machine	RFID tags in laundry basket in machine	Measure drum turning torque	None		
Dispense Detergent/conditioner	Biggie drawer with draw-off of detergent (into several hoses)	Internal storage and auto dispense	No detergent or conditioner – “ultrasonic”				
Wash Items	Horizontal drum	Vertical drum	Amber drum				
Rinse Items	Cold Water	Hot Water soak	Jet spray				
Extract Excess Water	Spin				Blow Hot air		
Determine Wash cycle	Look up		fuzzy logic	Expert system	Fixed programme		
Drain Waste Water	Pump	Internal pump	Filter & recycle				
Input user instruction	Low level	Touch screen			Keypad		
Inform user	LCD display	Passive display					

The number of candidate whole system concept solutions should be allowed to emerge. It could be 20 - it could be 2.

Any unused functional solutions should be investigated. Why were they not selected? Were the de-selection criteria correct?

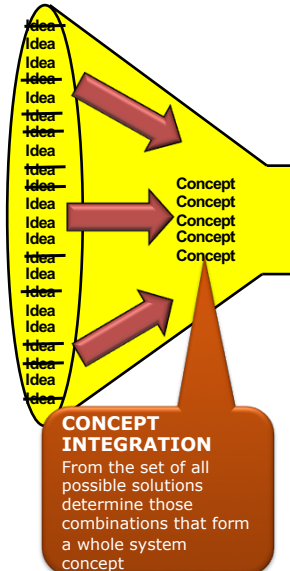
©2016 Waters Corporation

12

Exercise – Concept Selection

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



In teams

- For the case study project
- Identify three potential whole concept system solutions

Time: 45 Minutes

©2016 Waters Corporation

13

Summary: FMA Part 2

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ FMA part 2

– Pros

- Easy to undertake
- Visual and structured tool for the systematic identification of candidate whole system concept solutions
- Clear descriptions of whole concepts
- Captures in a simple way the justified design rationale
- Encourages team discussion and consensus decision making

■ FMA part 2

– Cons

- Ensuring the “best” de-selection criteria
- High level concepts can incorporate many “means”

©2016 Waters Corporation

14

Next Step

Waters

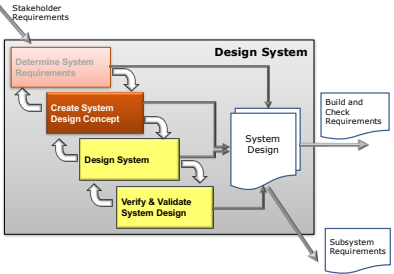
THE SCIENCE OF WHAT'S POSSIBLE.®

- The next step, having identified a number of candidate whole system concept solutions, is to “work up” the concepts to allow them to be evaluated
- This may involve:
 - Sketching out the physical layout/architecture
 - Producing concept drawings – show what a concept looks like
 - Defining the concept of operation – the Use Case or Use Cases for the concept
 - Performing “back of the envelope” calculations on performance for each concept





THE SCIENCE OF WHAT'S POSSIBLE.®



The diagram shows a process flow for a Design System. It starts with 'Stakeholder Requirements' leading to 'Determine System Requirements'. This leads to 'Create System Design Concept', which then leads to 'Design System'. 'Design System' leads to 'Verify & Validate System Design', which loops back to 'Design System'. 'Design System' also leads to 'System Design', which leads to 'Build and Check Requirements'. 'System Design' also leads to 'Subsystem Requirements'.

Create System Design Concept Concept Evaluation and Selection

This Session:

- Explains the concepts and principles of Concept Selection
- Describes and provides an opportunity to practise the use of simple tools to evaluate and select whole system solution concepts
 - Pugh Matrix
 - Analytic Hierarchy Process

©2016 Waters Corporation

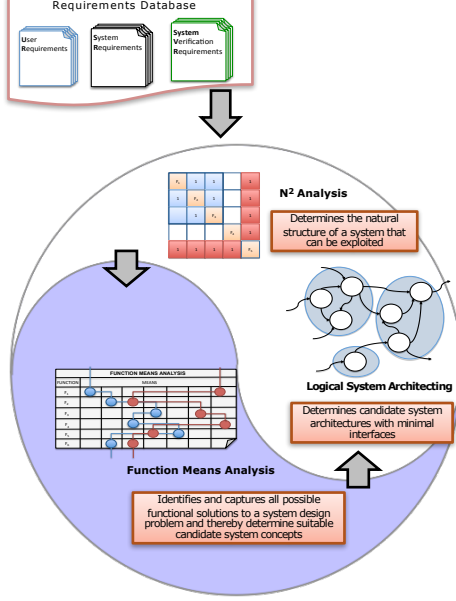
1





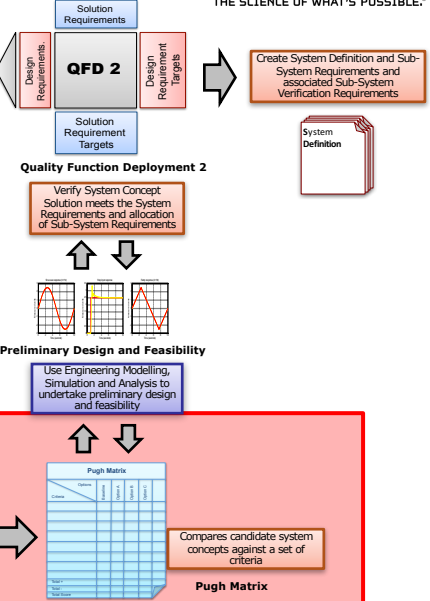
THE SCIENCE OF WHAT'S POSSIBLE.®

Process and Tools



The diagram illustrates the process and tools for system design. It starts with a 'Requirements Database' containing 'User Requirements', 'System Requirements', and 'System Verification Requirements'. This leads to 'N² Analysis', which determines the natural structure of a system. 'N² Analysis' leads to 'Function Means Analysis', which identifies and captures all possible functional solutions. 'Function Means Analysis' leads to 'Logical System Architecting', which determines candidate system architectures. 'Logical System Architecting' leads to 'Preliminary Design and Feasibility', which uses engineering modelling, simulation, and analysis. 'Preliminary Design and Feasibility' leads to 'Quality Function Deployment 2', which verifies the system concept. 'Quality Function Deployment 2' leads to 'Create System Definition and Sub-System Requirements and associated Sub-System Verification Requirements', which results in a 'System Definition'.

Quality Function Deployment 2



The diagram shows the QFD 2 process. It starts with 'Design Requirements' and 'Solution Requirements' leading to 'QFD 2'. 'QFD 2' leads to 'Design Requirement Targets' and 'Solution Requirement Targets'. 'Design Requirement Targets' leads to 'Create System Definition and Sub-System Requirements and associated Sub-System Verification Requirements', which results in a 'System Definition'. 'Solution Requirement Targets' leads to 'Verify System Concept', which checks if the solution meets the system requirements and allocation of sub-system requirements. 'Verify System Concept' leads to 'Preliminary Design and Feasibility', which uses engineering modelling, simulation, and analysis to undertake preliminary design and feasibility. 'Preliminary Design and Feasibility' leads to 'Pugh Matrix', which compares candidate system concepts against a set of criteria.

©2016 Waters Corporation

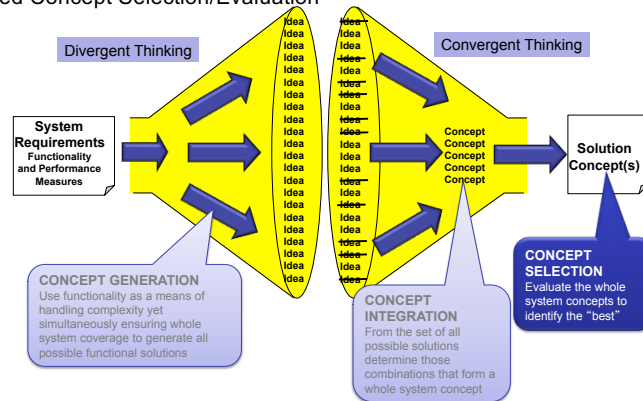
2

Introduction to Concept Integration and Selection

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The second stage of Function Means Analysis provided a number of candidate whole system solutions. The task now is to:
 - Evaluate the concept system solutions to arrive at the final concept solution(s) – this is called Concept Selection/Evaluation



©2016 Waters Corporation

3

Concept Selection/Evaluation

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The purpose of Concept Integration is to identify many alternate candidate whole concept system solutions
- The purpose of Concept Selection/Evaluation is to arrive at the best possible and feasible whole system solution from the candidate list, or at least eliminate the weaker options
- Useful tools include:
 - Pugh Matrix
 - Analytic Hierarchy Process

©2016 Waters Corporation

4

The Pugh Matrix

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

	Design Concept A	Design Concept B	Design Concept C	Design Concept D	Design Concept BC	Design Concept BD
Criteria 1	S	+	S	+	+	+
Criteria 2	S	-	S	+	S	+
Criteria 3	S	S	S	+	S	+
Criteria 4	S	-	+	+	+	+
Criteria 5	S	-	+	+	+	+
Criteria 6	S	-	S	-	S	-
Criteria 7	S	+	S	-	+	+
Criteria 8	S	+	S	-	+	+
Criteria 9	S	-	S	-	S	-
Criteria 10	S	S	-	S	S	S
TOTAL +	0	3	2	5	5	7
TOTAL -	0	5	1	4	0	2
TOTAL SCORE	0	-2	1	1	5	5

What is it?

- The Pugh Matrix is a form of Matrix Diagram that is useful when comparing a number of alternatives and for finding which best meets a set of requirements/criteria

Why do it?

- To quickly identify the best concept or eliminate the weaker ones

When to use it?

- Whenever attempting to decide between a number of alternate system concept designs
- Particularly useful when there is relatively large number of key customer requirements/criteria

Who Does it?

- Best performed by a team that includes all interested disciplines and functions

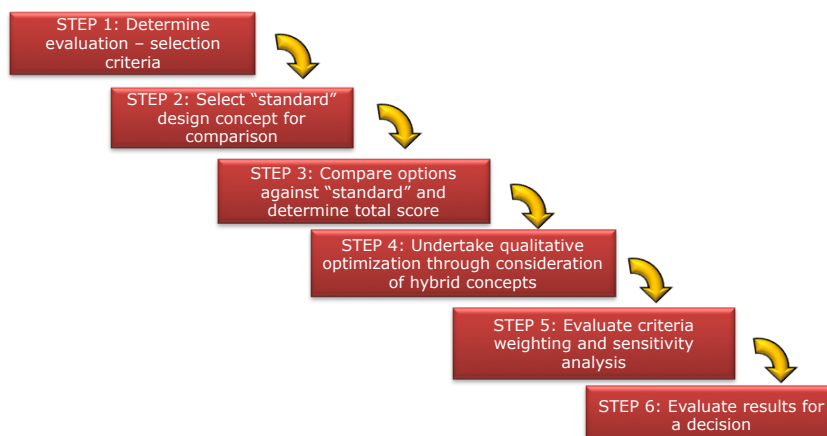
©2016 Waters Corporation

5

Pugh Matrix Process- How to do it

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

6

How to Do it: Steps 1 to 3

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Step 1:

- Determine suitable evaluation criteria. The customer facing requirements particularly the key Non-Functional System Requirements are sensible. It is important that these are supplemented by other stakeholder requirements - criteria that reflect other stakeholder views - particularly those of the business
- It is very important to have a clear definition of each criterion

Step 2:

- One design concept is selected to be the "standard". All criteria are scored 'S' for "Standard". If appropriate, a good choice is to use the previous design for the standard because it exists and therefore its performance should be well known

Step 3:

- Compare each design concept against the "standard" "requirement by requirement" and decide a "pair-wise score" with
 - S = same
 - + = better
 - = worse
- Can use ++ for much better and -- for much worse. Record decisions and rationale
- Sum the +'s and -'s to determine the Total Score

	Concept A	Concept B	Concept C	Concept D					
Criterion 1	S	+	S	+					
Criterion 2	S	-	S	+					
Criterion 3	S	S	S	+					
Criterion 4	S	-	+	+					
Criterion 5	S	-	+	+					
Criterion 6	S	-	S	-					
Criterion 7	S	+	S	-					
Criterion 8	S	+	S	-					
Criterion 9	S	-	S	-					
Criterion 10	S	S	-	S					
TOTAL +	0	3	2	5					
TOTAL -	0	5	1	4					
TOTAL SCORE	0	-2	1	1					

Always record rationale behind decisions

©2016 Waters Corporation

7

How to Do it: Step 4

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Step 4:

- Having scored each candidate design option, consider hybrids by combining where possible the best from each alternative. This is a form of qualitative optimisation.

	Concept A	Concept B	Concept C	Concept D	Concept BC					
Criterion 1	S	+	S	+	+					
Criterion 2	S	-	S	+	S					
Criterion 3	S	S	S	+	S					
Criterion 4	S	-	+	+	+					
Criterion 5	S	-	+	+	+					
Criterion 6	S	-	S	-	S					
Criterion 7	S	+	S	-	S					
Criterion 8	S	+	S	-	+					
Criterion 9	S	-	S	-	S					
Criterion 10	S	S	-	S	S					
TOTAL +	0	3	2	5	5					
TOTAL -	0	5	1	4	0					
TOTAL SCORE	0	-2	1	1	5					

Concept BC - a hybrid between Concepts B and C

©2016 Waters Corporation

8

How to do it: Steps 5 & 6

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Step 5

- The ability of the Pugh Matrix to discriminate between concepts can be improved by weighting the criteria
- The weighting is usually a simple 1 to 5 scale with 1 of the lowest importance and 5 of the highest
- It is also possible to perform a simple sensitivity analysis to assess the impact of small variations in the weightings – i.e. conclusions should be insensitive to small variations in the weightings

If you do have weightings – don't use them for the first pass to rank system solutions. Use them on the second pass – if the system solution ranking order is unchanged you have evidence of a robust decision.

	Weight	Concept A	Concept B	Concept C	Concept D	Concept BC
Criteria 1	5	S	+	S	+	+
Criteria 2	4	S	-	S	+	S
Criteria 3	2	S	S	S	+	S
Criteria 4	3	S	-	+	+	+
Criteria 5	1	S	-	+	+	+
Criteria 6	1	S	-	S	-	S
Criteria 7	4	S	+	S	-	+
Criteria 8	5	S	+	S	-	+
Criteria 9	2	S	-	S	-	S
Criteria 10	1	S	S	-	S	S
TOTAL +		0	14	4	15	18
TOTAL -		0	11	1	11	0
TOTAL SCORE		0	3	3	4	18

Step 6:

- Make the decision and record reasons behind decisions. Quite often there is no clear "winner" but there is often a clear "loser" in such cases perform a sanity check (does the decision make sense) and remove the losing option.

©2016 Waters Corporation

9

Double Check

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The strength and weakness of the Pugh Matrix is the pairwise comparison against a standard. We need to be consistent in our judgments
- If we judge option A to be + (better) than the standard and option B to be - (worse). To be consistent Option A must be what compared to B?
- A good consistency check is having found a winner that is not the standard, it is good practice to make it the standard, redo the pairwise comparison to test consistency, the ranking should be the same!

©2016 Waters Corporation

10

Exercise – Concept Selection



- In teams
 - Construct a Pugh Matrix to make a final selection. Use the customer facing requirements (from QFD 1) and include some internal requirements like:
 - Ease of Realisation/implementation
 - Technology Readiness Level
 - Risk
- Time: 30 Minutes

Summary: Pugh Matrix

- | | |
|--|--|
| <ul style="list-style-type: none"> ■ Pros <ul style="list-style-type: none"> – Provides a rapid assessment of alternative whole concepts – Can use stakeholder requirements to give greater confidence of compliance – Can help develop hybrid system concepts (qualitative optimisation) | <ul style="list-style-type: none"> ■ Cons <ul style="list-style-type: none"> – Temptation to use too many requirements slowing down the process – Objective subjectivity! can select criteria, weightings and scores to obtain any result! – Often no clear winner (but eliminates the weaker options) – Can be treated as a “handle turning” process without thinking |
|--|--|

Analytical Hierarchy Process

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Is the row item more important (>1), equally important (=1), or less important (<1) than the column item? Fill the lower triangular matrix.	1 More attractive appearance	2 Safer to use	3 Easier to use	4 Improved capacity for various types o...	5 Improved capacity for multiple slices ...	Importance
1 More attractive appearance						12%
2 Safer to use	4					40%
3 Easier to use	1	1/2				19%
4 Improved capacity for various types o...	1/2	1/4	1/4			6%
5 Improved capacity for multiple slices ...	3	1/2	1	3		23%
Consistency Ratio (should be < 0.1)	0.03					

What is it?

- The Analytical Hierarchy Process (AHP) is a tool that provides structure and discipline to group decision-making in order to:
 - Select between a number of options
 - Prioritise requirements/criteria
- It helps the group assign numerical values to subjective judgments and synthesise those judgments into true ratio scale numbers to determine which items have the highest priority, and by how much

Why do it?

- Prioritising or deciding often concerns a number of interrelated factors that are sufficiently complex to result in inconsistent and irrational decisions. AHP provides a framework to minimise the inconsistencies

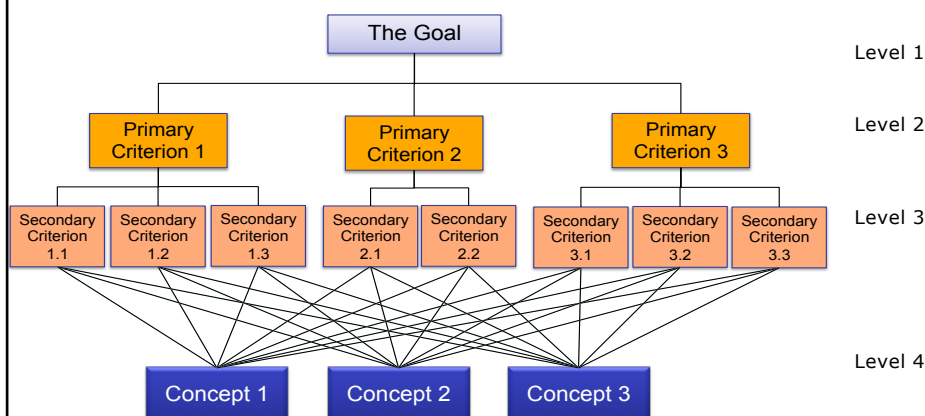
©2016 Waters Corporation

13

The Hierarchy for Concept Selection

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

14

Top Down: Start by Determining which Weighting for the Primary Criterion

- The first step is to establish the weightings of the primary criteria by asking the questions:

- How much more important is improved Primary Criterion 1 than improved Primary Criterion 2 in selecting a concept?
- How much more important is improved Primary Criterion 1 than improved Primary Criterion 3 in selecting a concept?
- How much more important is improved Primary Criterion 2 than improved Primary Criterion 3 in selecting a concept?

	Primary Criterion 1	Primary Criterion 2	Primary Criterion 3	Importance
Primary Criterion 1	1.0			$W_{PC1} = 0.2$
Primary Criterion 2		1.0		$W_{PC2} = 0.5$
Primary Criterion 3			1.0	$W_{PC3} = 0.3$
Consistency < 0.1				

By completing the matrix and performing the analysis we can find out the weightings of the primary criteria

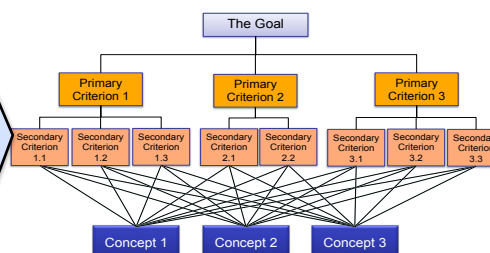
Next level down

For the first Primary Criterion we form the 3x3 matrix

	Secondary Criterion 1.1	Secondary Criterion 1.2	Secondary Criterion 1.3	Importance
Secondary Criterion 1.1	1.0			$W_{SC1.1} = 0.18$
Secondary Criterion 1.2		1.0		$W_{SC1.2} = 0.4$
Secondary Criterion 1.3			1.0	$W_{SC1.3} = 0.42$
Consistency < 0.1				

And ask the questions:

- How much better is it to improve Secondary Criterion 1.1 than improve Secondary Criterion 1.2 in achieving Primary Criterion 1?
- How much better is it to improve Secondary Criterion 1.1 than improve Secondary Criterion 1.3 in achieving Primary Criterion 1?
- How much better is it to improve Secondary Criterion 1.2 than improve Secondary Criterion 1.3 in achieving Primary Criterion 1?



This is repeated for all secondary decision criterion relating to a primary criterion

And Finally

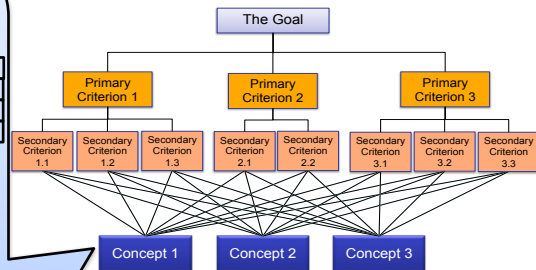
For each decision criterion at this level we form the 3x3 or (2x) matrix

	Concept 1	Concept 2	Concept 3	Importance
Concept 1	1.0			$C_{1-1,1} = 0.1$
Concept 2		1.0		$C_{2-1,1} = 0.5$
Concept 3			1.0	$C_{3-1,1} = 0.4$

Consistency < 0.1

And ask the questions:

- How much better is Concept 1 than Concept 2 in relation to the criterion under consideration?
- How much better is Concept 1 than Concept 3 in relation to the criterion under consideration?
- How much better is Concept 2 than Concept 3 in relation to the criterion under consideration?



This is repeated for all decision criterion at the lowest level

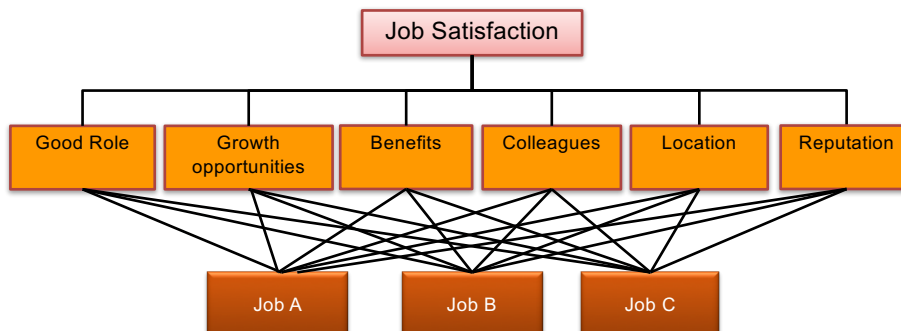
Note: we have to construct 12 AHP decision matrices!

©2016 Waters Corporation

17

An Example

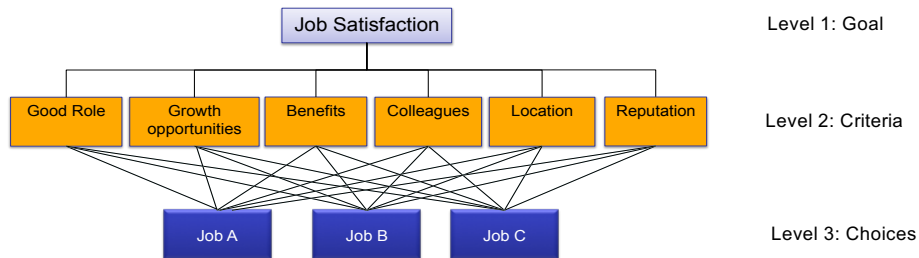
- In looking for a new job a graduate receives offers from 3 companies
- The overall goal is *Job Satisfaction* which the graduate decides comprises six criteria
- The decision hierarchy is:



©2016 Waters Corporation

18

The Decision Hierarchy



Key

Role = R
Growth = G
Benefits = B
Colleagues = C
Location = L
Reputation = Rp

How many AHP matrices are needed and how many paired comparisons?

©2016 Waters Corporation

19

Level 2 of the Hierarchy

Overall Job Satisfaction	R	G	B	C	L	Rp	Importance
R	1	1	1	4	1	0.5	0.16
G	1	1	2	4	1	0.5	0.18
B	1	0.5	1	5	3	0.5	0.20
C	0.25	0.25	0.2	1	0.333333	0.333333	0.05
L	1	1	0.333333	3	1	1	0.16
Rp	2	2	2	3	1	1	0.25

Interpretation

Benefits are 4 times more important than Colleagues (only 4??)

- The matrix is not **consistent**. Why?
 - Acceptable level of inconsistency is covered later, but it is worth noting now inconsistency is a real human trait and is arguably necessary in order for us to grow intellectually.

©2016 Waters Corporation

20

Level 3 of the Hierarchy

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Role	jobA	jobB	jobC	Importance
jobA	1	0.25	0.5	0.14
jobB	4	1	3	0.62
jobC	2	0.33333333	1	0.24

Similar 3x3 matrices are constructed for:

Growth = G

Benefits = B

Colleagues = C

Location = L

Reputation = Rp

The pairwise question:

"How much better is job A than job B in relation to the role on offer?"

The answer was "It's not, it's moderately worse"

©2016 Waters Corporation

21

The Results

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

	R	G	B	C	L	Rp
Level 2 Importance	0.16	0.18	0.20	0.05	0.16	0.25
Job A Importance	0.14	0.10	0.32	0.28	0.47	0.77
Job B Importance	0.62	0.33	0.22	0.65	0.47	0.17
Job C Importance	0.24	0.57	0.46	0.07	0.07	0.05
Consistency Ratio	2%	2%	54%	6%	0.1%	20%

Comparisons of jobs versus benefits (salary etc.), and versus reputation were very inconsistent. In practice these judgments should be re-visited – were the right questions asked?

©2016 Waters Corporation

22

Selecting the Job!

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

	R	G	B	C	L	Rp
Level 2 Importance	0.16	0.18	0.20	0.05	0.16	0.25
Job A Importance	0.14	0.10	0.32	0.28	0.47	0.77
Job B Importance	0.62	0.33	0.22	0.65	0.47	0.17
Job C Importance	0.24	0.57	0.46	0.07	0.07	0.05
Consistency Ratio	2%	2%	54%	6%	0.1%	20%

Overall priority for Job A is

$$(0.16 \times 0.14) + (0.18 \times 0.10) + (0.20 \times 0.32) + (0.05 \times 0.28) + \dots = \mathbf{0.39}$$

Similarly, Job B = **0.35** and Job C = **0.26**

Decision

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Job A = 0.39 Job B = 0.35 Job C = 0.26

- Job A is the one that comes closest to meeting the goal of job satisfaction, based on the criteria used
- Jobs A and B are pretty close though!
- There are a number of things we can do to help make a decision:
 - Identify further criteria to compare Job A and Job B
 - Undertake a sensitivity analysis

Analytic Hierarchy Process Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Pros
 - If performed in a team can promote
 - Discussion
 - Thinking
 - Structured and disciplined process
 - Can be used to aid decision-making using a True Ratio scale
 - Checks consistency of judgements adding confidence to the decision making
- Cons
 - Deciding who does this – who should be involved
 - Ensuring judgements are made correctly
 - Mathematical basis adds unfounded credibility
 - Tedious if many requirements (limit of 7 per level)

AHP in Decision Making

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

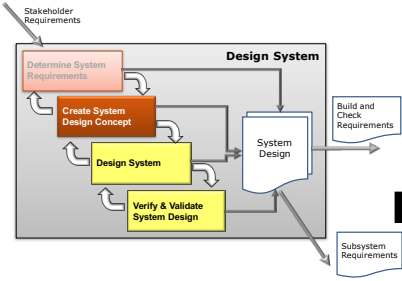
- There is no doubt the Analytic Hierarchy Process works and is highly effective
- It is, however, potentially unwieldy and time consuming! - But in combination with simpler tools like the Pugh Matrix it can provide a very powerful approach to decision making where:
 - STEP 1: Use the Pugh Matrix to eliminate the weaker concept options
 - STEP 2: Use AHP to reach a final decision

It is of course possible that even with AHP that the outcome is no decision. All this is telling you is: - "you do not have enough information to make the decision!"





THE SCIENCE OF WHAT'S POSSIBLE.®



Create System Design Concept

Concept – Requirements Verification using Quality Function Deployment Phase 2

This Session:
 To describe and practise Quality Function Deployment 2 as a tool that can be used to help in the System Concept Design process to:

- Check that design solutions meet specified requirements
- Flow-down requirements to subsystem level
- Provide a reference for future systems

©2016 Waters Corporation

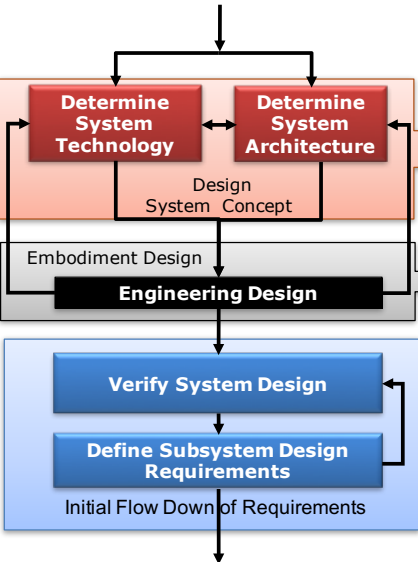
1





THE SCIENCE OF WHAT'S POSSIBLE.®

Key Tasks and Systems Tools



Typical Systems Tools

- Creative Thinking Tools
- Function Means Analysis
- Evaluation Tools
 - Pugh Matrix
 - Analytic Hierarchy Process
- Logical System Architecting
- N² Analysis (Design Structure Matrix)

Typical Systems Tools

Engineering Analysis Tools

Typical Systems Tools

- Quality Function Deployment 2
- Requirements Database

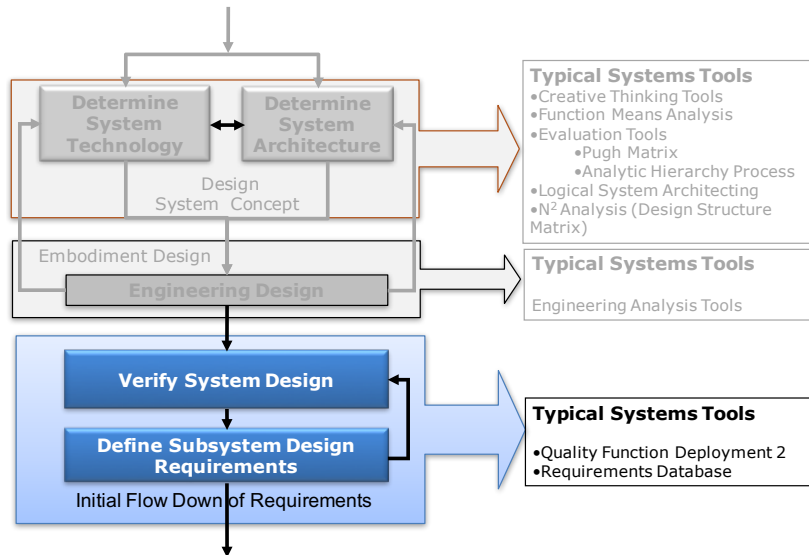
©2016 Waters Corporation

2

Key Tasks and Systems Tools

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



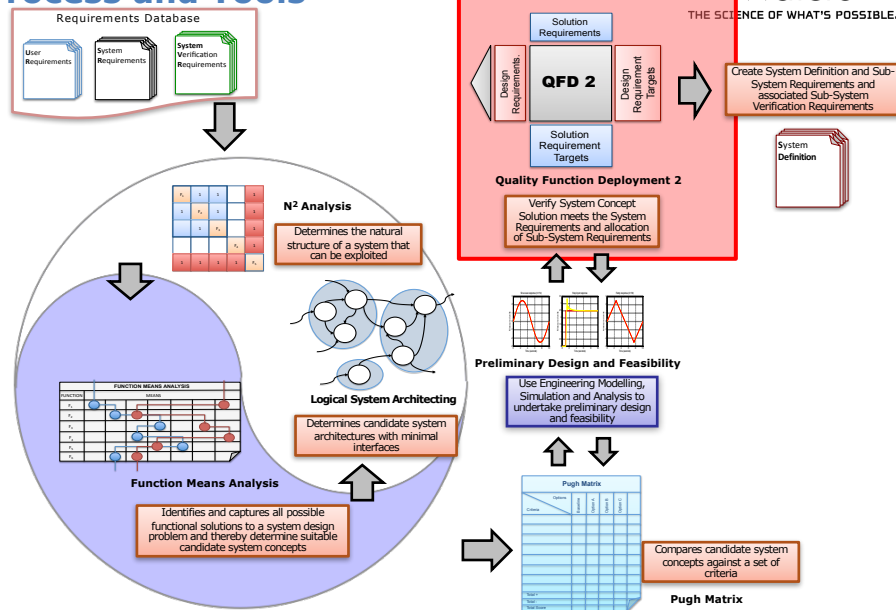
©2016 Waters Corporation

3

Process and Tools

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

4

Quality Function Deployment Phase 2

Waters

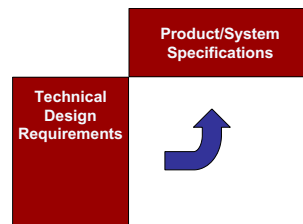
THE SCIENCE OF WHAT'S POSSIBLE.®

Design Requirements	Solution Requirements	Microprocessor	Abc 16 Key pad	LCD Display	XYZ Coin Mech	Card reader	Handle with Cable	S steel casing
Select Lang	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Enter Number	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Display Info	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Handle Coins	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Read card	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
TX/RX Aural Messages	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Containment	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Solution Requirement Target values								

The example QFD 2 is for the Payphone and shows the relationship between the design requirements and the system concept to verify the concept.

What is it?

- QFD 2 is concerned with translating Design Requirements into Solution Requirements. In the context of system development, it can be used to
 - Verify if the selected whole concept solution is compliant with the technical requirements
 - Correctly flow-down lower-level subsystem requirements



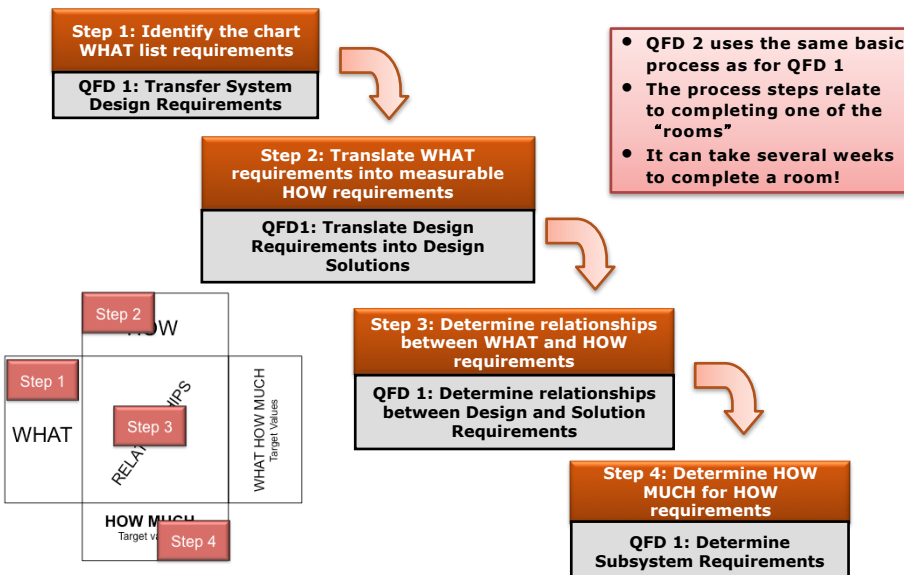
©2016 Waters Corporation

5

QFD Core Process

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

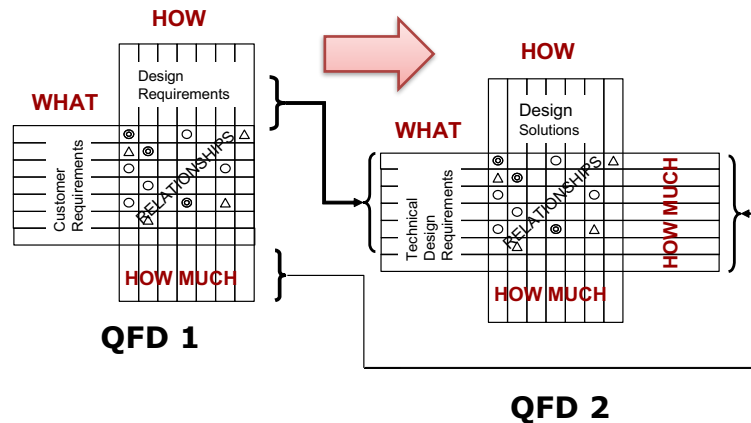


©2016 Waters Corporation

6

Step 1: Transfer Technical Requirements

- The HOWs and HOW MUCHs of QFD 1 form the WHATs for QFD 2. This ensures complete requirements flow-down

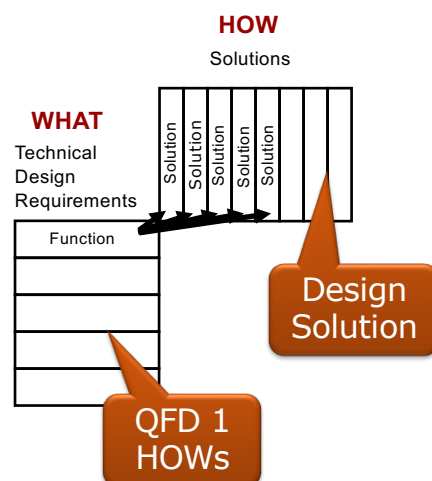


©2016 Waters Corporation

7

Step 2: Translate into Solution Requirements

- The WHATs of QFD 2 are the HOWs of QFD 1 which now require further definition in terms of the system solution
- This is achieved by reference to the concept system solution, adding further decisions as necessary
- If we have performed a Function Means Analysis, then the translation can be taken directly from the FMA table
- The selected concept system solution is written in the HOWs room.
- If there are several candidate system solutions then either
 - Several QFD 2 can be generated – one for each whole system concept
 - A single QFD 2 created with multiple whole system concepts



©2016 Waters Corporation

8

Step 3: Determine HOW-WHAT Relationship Matrix

WHAT Technical Design Requirements	HOW Design Solutions					
					Solution	
	◎			○		△
	△	◎				
	○		◎		○	
Function	Blank or Sparse	Sparse			Blank or Sparse	Low
	○					△
		△		○	◎	
					○	◎

- ◎ STRONG relationship
- MEDIUM relationship
- △ WEAK relationship

This method allows complex relationships between the system functionality and system solutions to be depicted and easily interpreted

It also permits cross-checking. Blank rows or columns indicate places **where** the translation of WHATs into HOWs is inadequate

If the QFD 2 is derived from the Function Means Analysis, then the leading diagonal should/will contain strong relationships. What is of interest is the off diagonal terms! They indicate potential emergent properties, interactions or interfacing issues.

©2016 Waters Corporation

9

Example

Note strong relationships on leading diagonal due to the earlier use of a Function Means Analysis

Functional Solutions from the body of the Function Means Analysis

Design Requirements	Solution Requirements						
	Microprocessor	Abc 16 Key pad	LCD Display	XYZ Coin Mech	Card reader	Handset with Cable	S. steel casing
Select Lang	◎	○	○	△	△		
Enter Number	○	◎	○				
Display Info	△		◎		△		△
Handle Coins	○		△	◎			△
Read card	△		△		◎		○
TX/RX Aural Messages						◎	
Containment		△	△	△	△	○	◎
Solution Requirement Target values			Luminosity > 65W/m ² Size 70x 50mm				

Functions from the LHS of the Function Means Analysis

Off-diagonal relationships indicate potential emergent behaviour or interaction issues

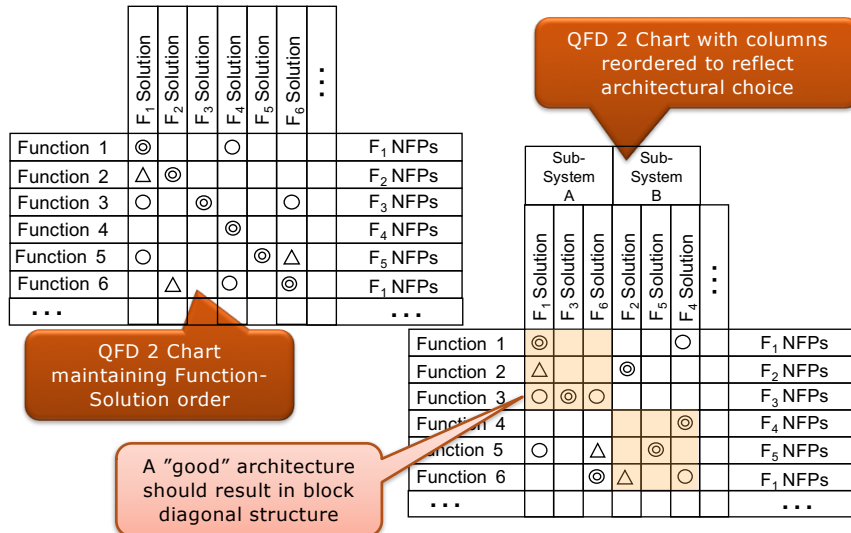
©2016 Waters Corporation

10

Capturing Architecture on QFD 2

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

11

Exercise – QFD 2 Relationship Matrix

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

	F ₁ Solution	F ₂ Solution	F ₃ Solution	F ₄ Solution	F ₅ Solution	F ₆ Solution	...	
Function 1	⊗							F ₁ NFPs
Function 2	△	⊗						F ₂ NFPs
Function 3	○		⊗					F ₃ NFPs
Function 4				⊗				F ₄ NFPs
Function 5	○				⊗	△		F ₅ NFPs
Function 6		△				⊗		F ₆ NFPs
...								...

In teams

- For the case study exercise transfer functions and a system solution from the Function Means Analysis
- Complete the relationship matrix and identify any interesting features
- Flip chart the outcomes and be prepared to share with the rest of the group

Time 20 Minutes

©2016 Waters Corporation

12

Look for "Structure"

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

We should examine the resultant relationship matrix for "structure" as this will show how the system functionality is provided by the solutions and vice versa. It can help in the management of key design aspects

Highly interacting set of solutions and functions – subsystem?

	SOLUTION 1	SOLUTION 2	SOLUTION 3	SOLUTION 4	SOLUTION 5	SOLUTION 6	SOLUTION 7	SOLUTION 8	SOLUTION 9
FUNCTION 1	⊙	⊙	△					△	⊙
FUNCTION 2	△	⊙	⊙			⊙		△	
FUNCTION 3	△		⊙					⊙	
FUNCTION 4				⊙				⊙	
FUNCTION 5					⊙			△	
FUNCTION 6	⊙	⊙	△	△	⊙	⊙		⊙	
FUNCTION 7							⊙		
FUNCTION 8								⊙	
FUNCTION 9		⊙						△	⊙

Functionality that is affected by several solutions. Need careful flow-down of Non-functional Performance Requirements and management of key solution contributors

Critical solution that delivers/contributes to many functions. Need to get the flow-down right and manage solutions development

©2016 Waters Corporation

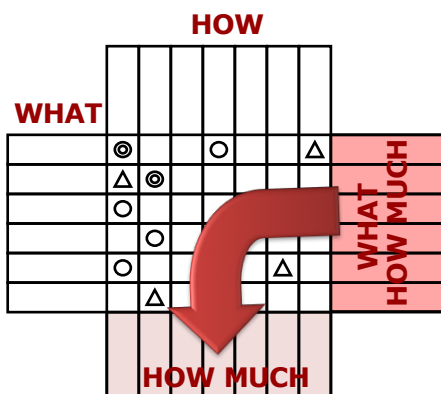
13

Step 4: Flow-down Targets and Ceilings

HOW MUCH is Enough?

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- The fourth element of the QFD 2 chart is flowing down the WHAT-HOW MUCHs to give a set of target values for the chosen design solution
- These are design characteristics and targets for the design solution HOWs and specify how good the HOWs have to be in order to satisfy the customer
- If we are dealing with a system then the HOW MUCHs are the basis for the subsystem requirements
- If we are dealing with components or parts then the HOW MUCHs are the part specifications
- At higher system levels the HOW MUCHs may be a simple flow-down, but at the lower levels it will be necessary to undertake detailed design work (including Robust Design) to obtain the desired information to complete the HOW MUCHs

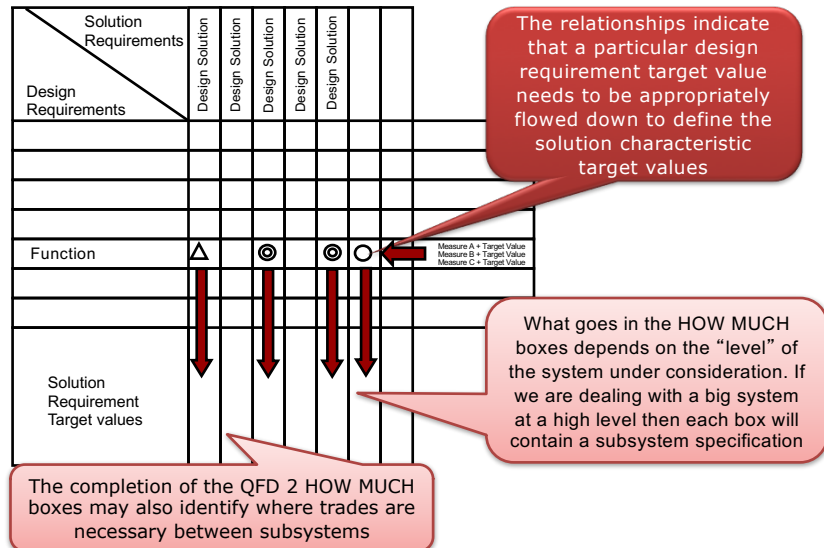
©2016 Waters Corporation

14

Flowing Requirements Down

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



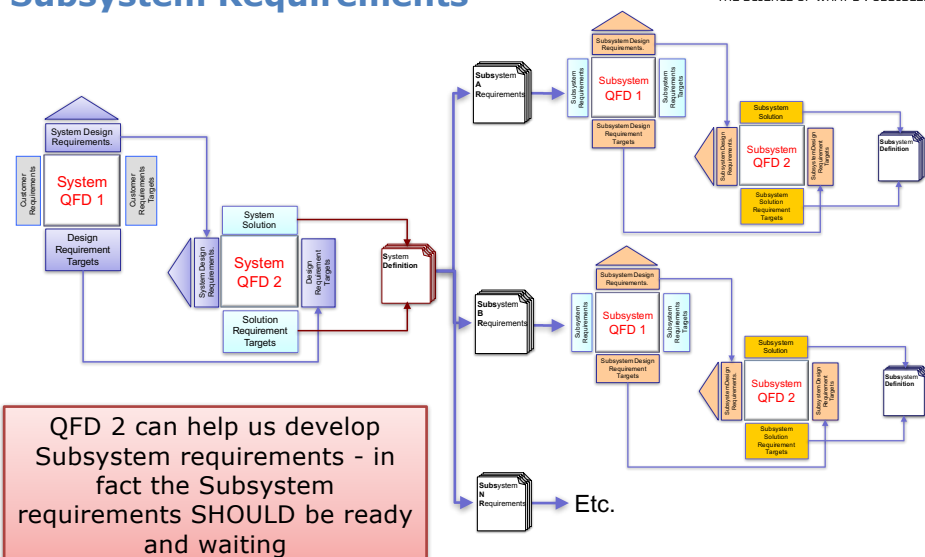
©2016 Waters Corporation

15

Subsystem Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

16

QFD 2 Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Pros
 - Verifies that system concept solution(s) are compliant with the System Design Requirements
 - Identifies potential system interactions that could result in undesirable emergent behaviour
 - Helps in the correct and complete flow-down of requirements from higher levels to lower levels
 - Helps manage requirements and trade-offs
- Cons
 - If several concepts system exist then it will require the development of several QFD 2s
 - Can take time to complete, particularly the flow-down of HOW MUCH requirements

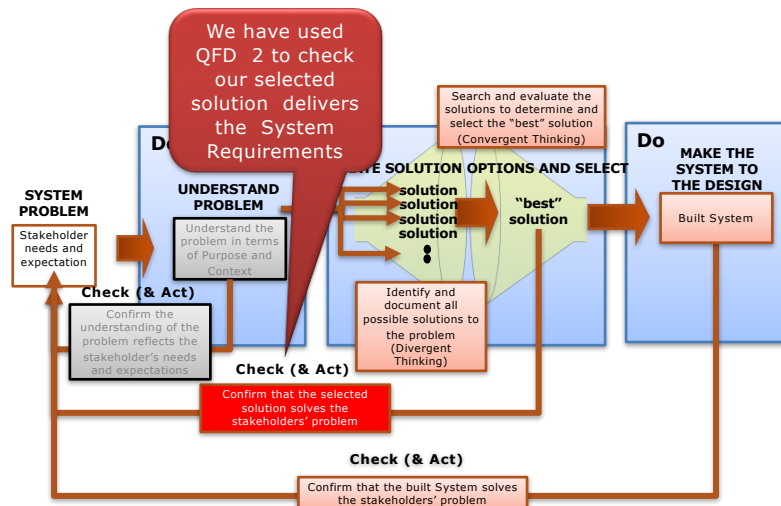
©2016 Waters Corporation

17

What have we done?

Waters

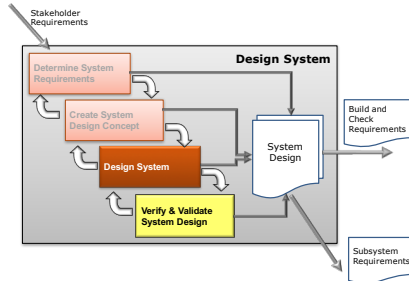
THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

18

Design System A Systems Approach to Robust Design

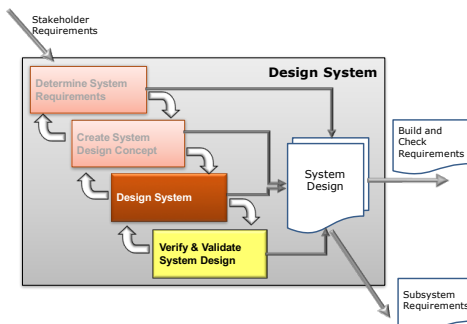


This Session:

- Explains the Systems Approach to Optimising and Refining [the] Solution Concept
- Emphasise the importance of finding the “Robust Optimum”

Where are we: Optimise Solution Concept

Design Space?
Robust Optimum?



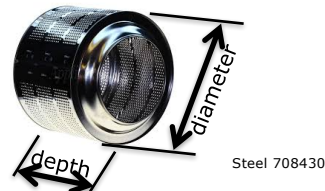
- Purpose :
 - To explore and characterise the design space of the selected system concept design(s) in order to identify the robust optimum
- Outcomes :
 - The robust optimum concept design
 - Requirements flowed down to subsystem level

System Design Optimization

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The performance of a system will be influenced by a number of system design parameters – those variables that the designer selects target values for to achieve the desired level of system performance
- The challenge for the system designer is to find the best target or nominal values for the system design parameters in order to meet or, better still, exceed the performance requirements

- For example if a drum is selected as the means to the “wash items” function of a washing machine



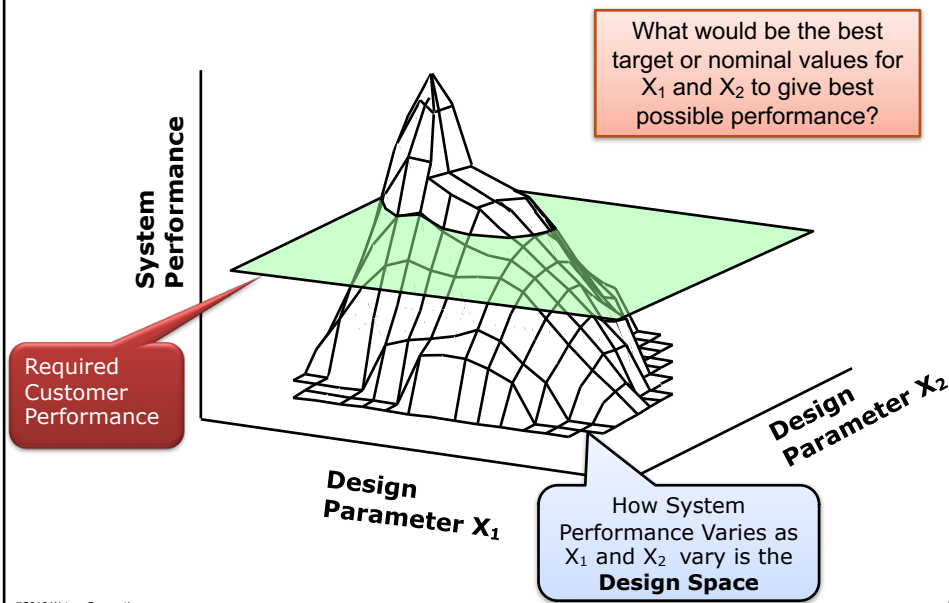
- The quality of the wash will depend on:
 - Speed of rotation
 - Drum diameter
 - Drum depth
 - Drum material
 - etc.

©2016 Waters Corporation

3

Where is the best place to be?

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



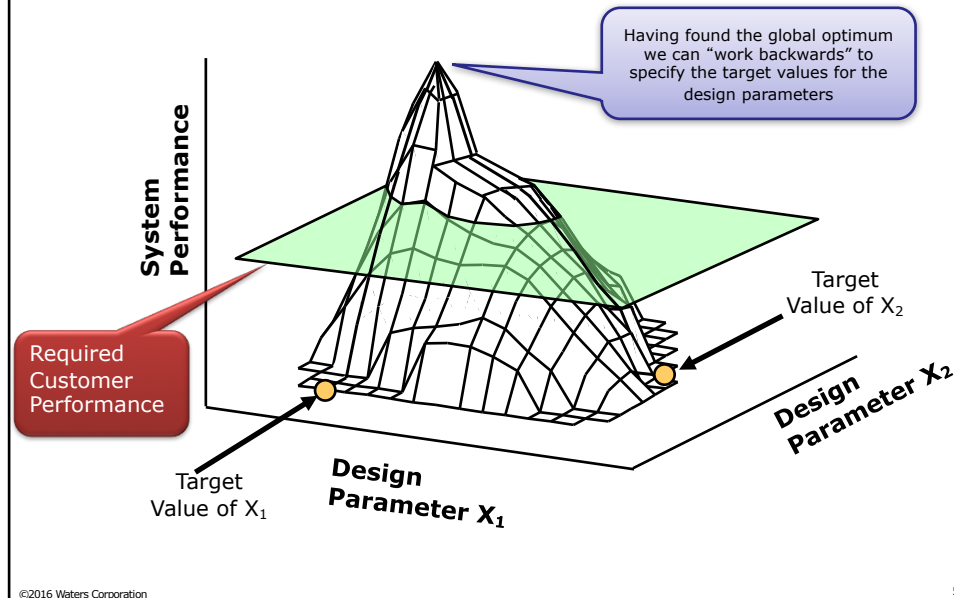
©2016 Waters Corporation

4

The Best Place – the Global Optimum?

Waters

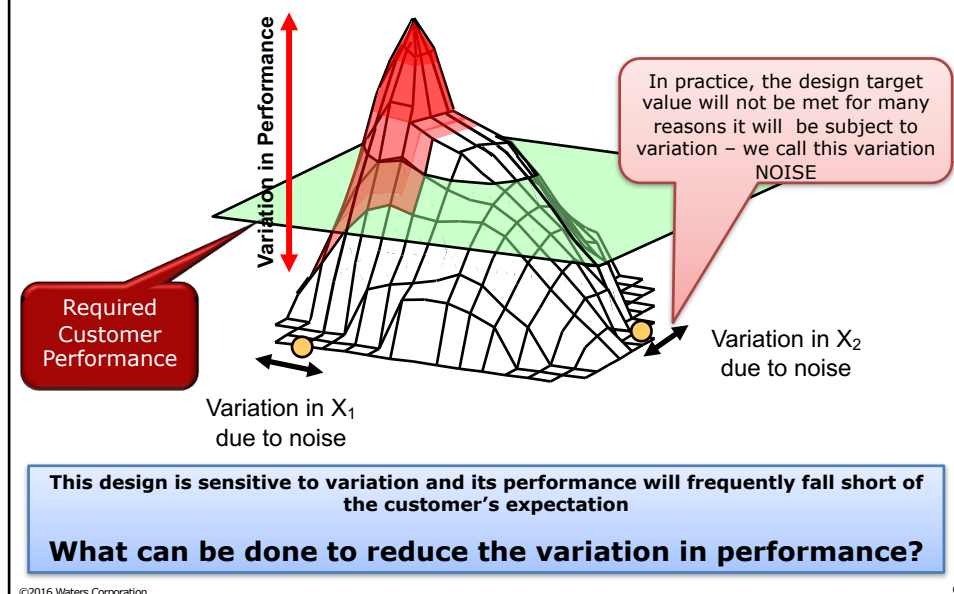
THE SCIENCE OF WHAT'S POSSIBLE.®



All Systems are Subject to Noise

Waters

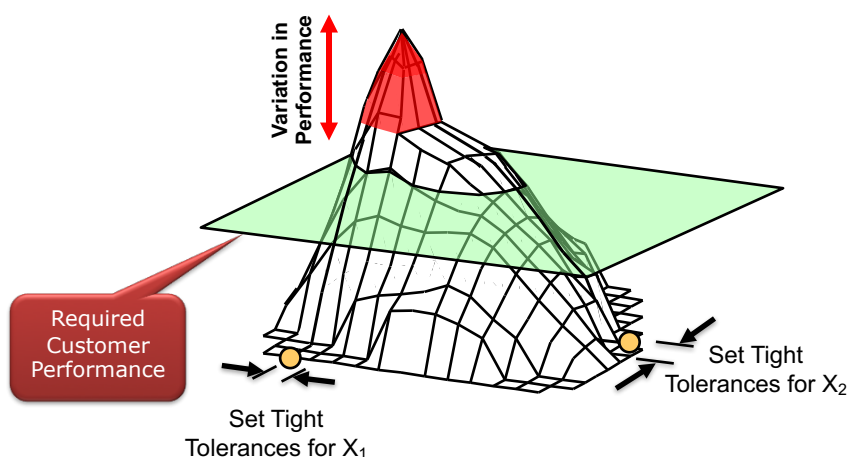
THE SCIENCE OF WHAT'S POSSIBLE.®



Many Sources of Noise Variation

- There are three groups:
 - External: The environment and the load under which a system must function are the two main external sources of variation
 - Temperature, humidity, vibration
 - Operator usage
 - Unit-to-Unit Variation: The variation in a manufacturing process leads to variation in the system from unit-to-unit
 - Differences in diameter from machining
 - Raw material variation
 - Calibration
 - Gauge Repeatability & Reproducibility
 - Deterioration: As time passes the system characteristics can change leading to a deterioration in performance

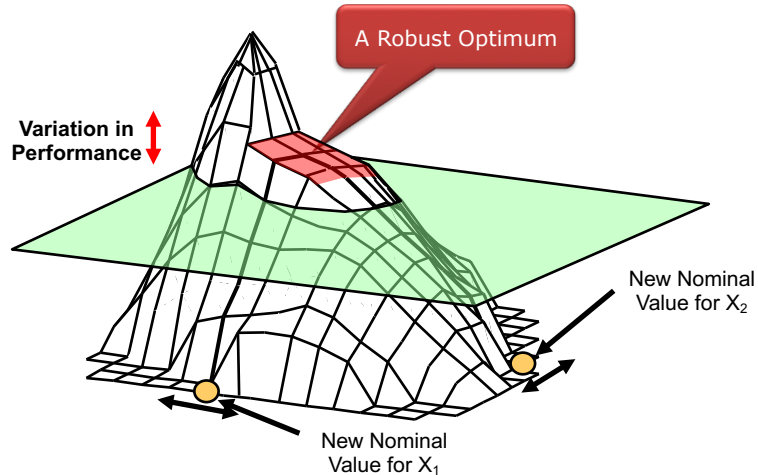
The Classic Strategy: Tighten Tolerances



The classic reaction to a sensitive design is to set tighter tolerances. This does work but often comes at a great cost.

Achieving Design Robustness

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®



This design is less sensitive to variation and will consistently meet the customers expectation

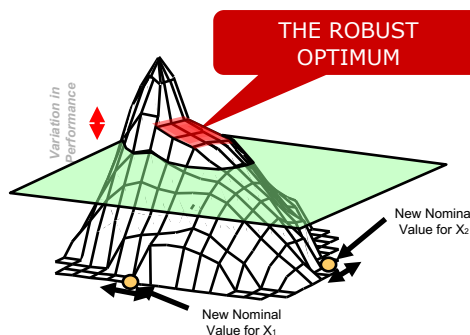
©2016 Waters Corporation

9

Robust System Optimisation

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

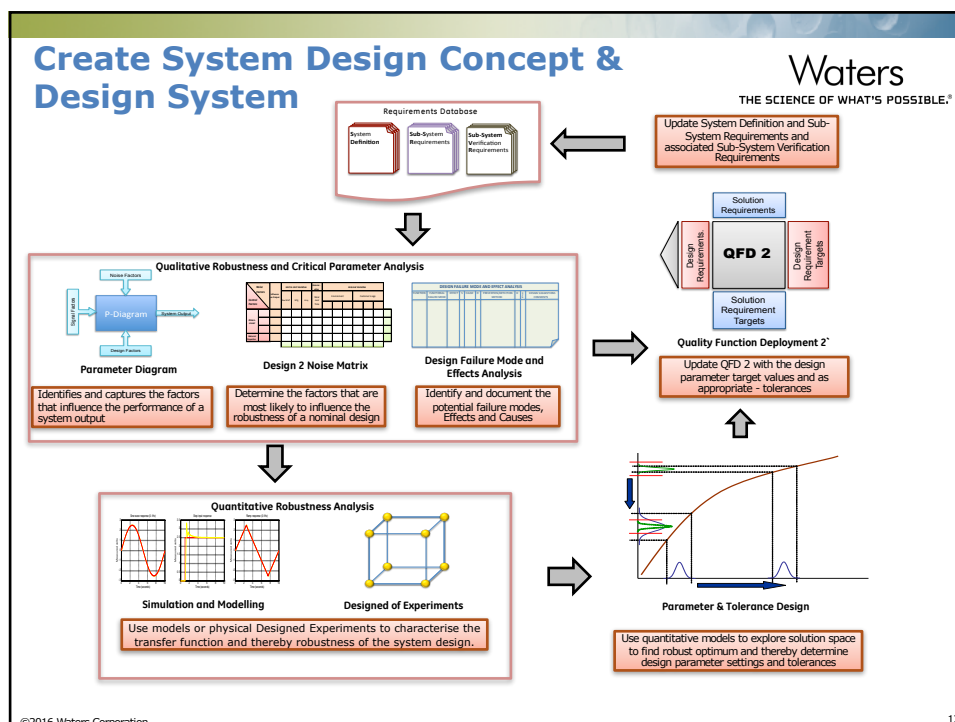
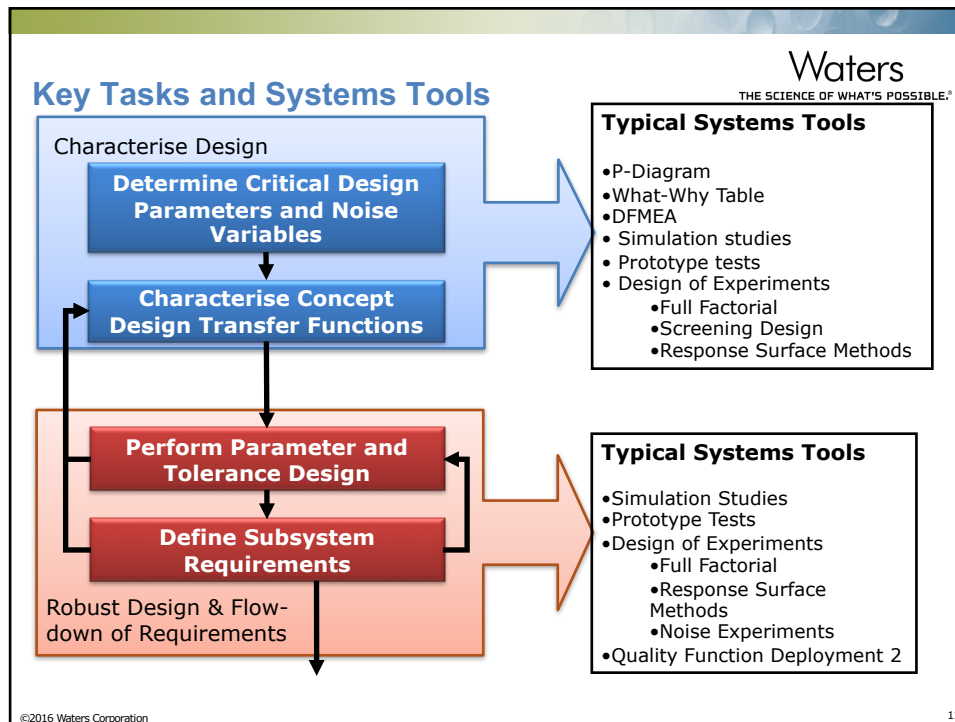
- Finding the target or nominal values for the system design parameters that deliver best performance may sound like the "optimum design" but can result in sensitive designs that suffer excessive variation in performance, causing customer dissatisfaction.
- The optimum design in systems engineering is the robust optimum.
 - The best target values for the system design parameters that meet the customer requirements CONSISTENTLY, robust against the effects of variation.



But how can it be achieved in practice?

©2016 Waters Corporation

10

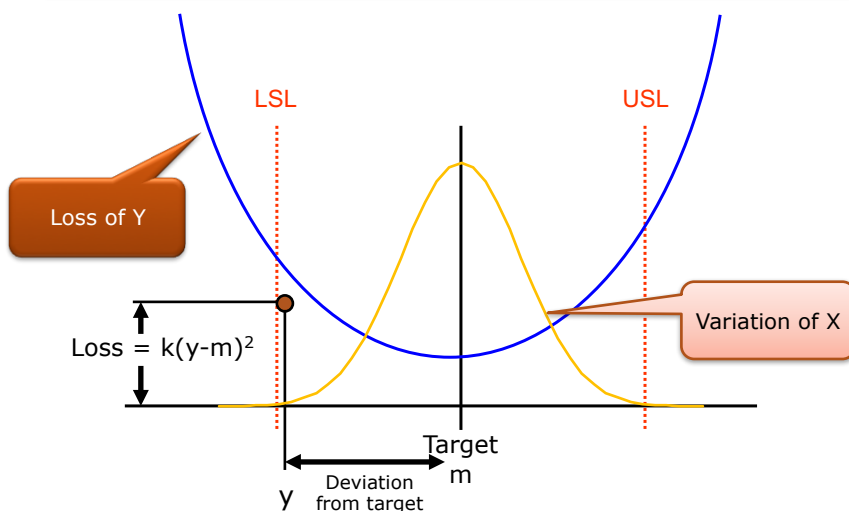


We are Talking about Variation and Loss Functions

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Taguchi's loss function applies to other system outputs and not just cost!



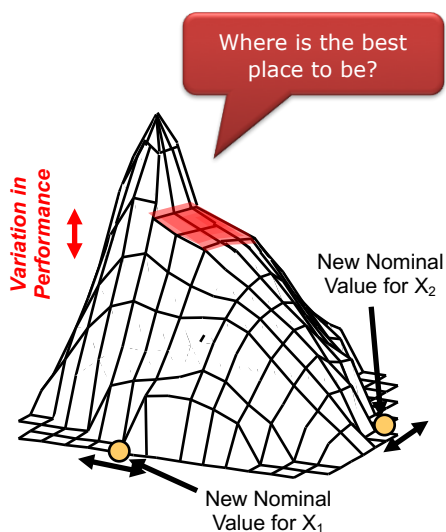
©2016 Waters Corporation

13

Achieving Design Robustness is Therefore

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Design robustness is concerned with:
 - Exploring the design space to find regions where the desired system performance (Y) is achieved even in the face of variation in the noise factors
 - i.e. finding the plateaus and avoiding the peaks and troughs.
- Fundamental approach is to:
 - Identify the transfer function that quantifies the relationship between the performance requirement and the signal, control and noise factors
 - Find the “best” settings for the control factors (design parameters) that optimise the performance requirement while simultaneously reducing the sensitivity to noise

©2016 Waters Corporation

14

Iterate Design: A Systems Approach

System Design

Establish the fundamental design and engineering concepts



Parameter Design

Determination of target values AND sensitivity to variation – i.e. the loss functions are determined. If a sensitive design, iterate the design by changing the target values



Tolerance Design

Tolerances established based on sensitivity analysis. Robust components do not need a tight tolerance - sensitive ones do

Conventional Design

The requirements for the system output express a range of suitable values

System Output Y

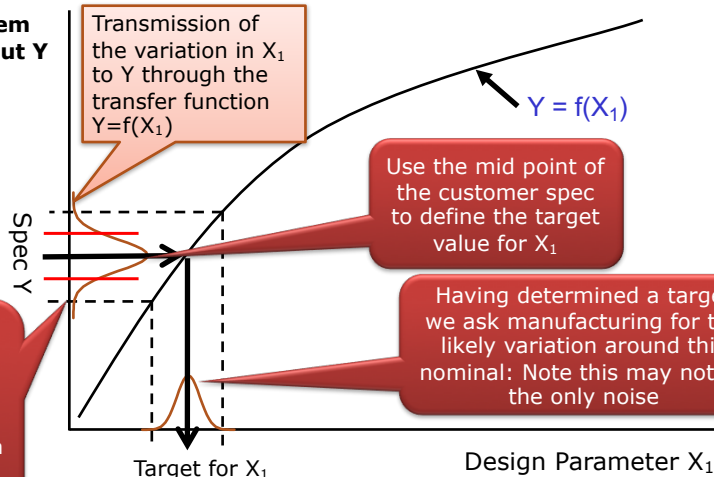
Transmission of the variation in X_1 to Y through the transfer function $Y=f(X_1)$

$$Y = f(X_1)$$

Use the mid point of the customer spec to define the target value for X_1

Having determined a target we ask manufacturing for the likely variation around this nominal: Note this may not be the only noise

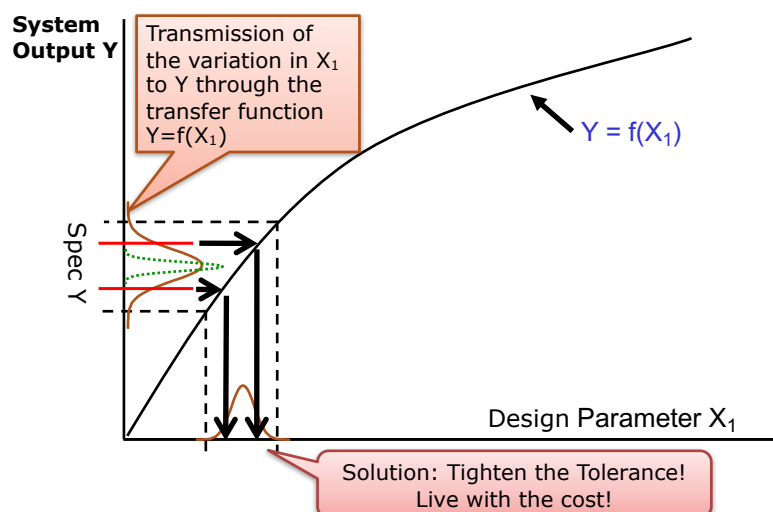
Oh dear!. The manufacturing noise variation is "transmitted" through the transfer function to give this variation at the output



Traditional Solution

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



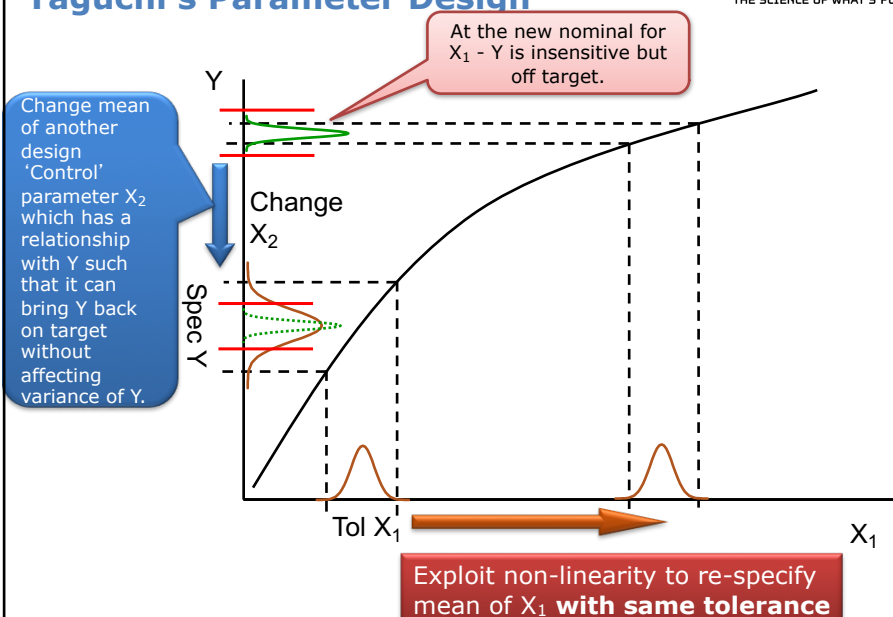
©2016 Waters Corporation

17

Taguchi's Parameter Design

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

18

Example: The Intelligent Washing Machine

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Start by finding out what the customer wants



Customer satisfaction is affected by:

- Attractive and distinctive
- Cost of purchase
- Running costs
- Washing performance
- Dry performance
- Reliability/Durability
- Ease of use
- Access Ergonomics
- Easy to maintain
- Safety

©2016 Waters Corporation

19

Need to Interpret Customer Requirements Technically

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Use QFD 1 to translate into Technical Design Requirements

QFD 1

Washing performance is affected by several system functions including the Drain Water Function. Customer research determines that the target for the Drain Water function is less than 2 minutes. The Drain Water Function also has other high level requirements of reliability and cost

		Customer Importance	IWM Functions and Features											
			Wash Sensing	Load Machine	Manage Water	Clean Item	Manage		User I/D					
Customer Requirements	Intelligent Dishwasher	Wash Performance	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Dry Performance	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Automation & Intelligence	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Easy to use Controls	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Access Ergonomics	3.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Purchase	1.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Running	2.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Reliable	1.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Easy to Maintain	1.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
		Attractive	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
Weighted Importance			5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	
Relative Importance			5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	

How Much might be:
Drain <2min
FYFR <1%
DTUPC <£8.0

©2016 Waters Corporation

20

System Design & Requirements Flow Down

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

The concept design stage (FMA, Pugh, QFD 2) identifies a pump as the solution for the drain water function.

The NFP requirement of:

drain time < 2min

Can be translated in to a subsystem requirements as a target flow rate and allowable variation of 10 +/-1 (litres/min). Below 9 litres/min cannot meet the < 2 min limit. Above 11 litres/min will cause flooding!

Design Requirements	Element Requirements	Switch	Weight sensors	pump	BITE	RFI Tags	Hardness Sensor	Etc.
Activate	○	○	○		△	△	△	Manually activated
Determine Load Distribution	▲	○	○					7Kg Max .1 res
Drain Water	▲	△	○					< 2 mins FYFR <1% DTUPC <£8.0
Self Monitor	▼	△		○	○	△		95% validation
Detect Load make up	▲	○			○			5.6 kg max
Detect H ₂ O hardness	○	△		○	△	○		EU Limits
Inform user	○	○		○				Cycle, status in cycle
Target values		10A 240V Double pole	15 x xyz type	10 l/min +/- 1 FYFR <1% DTUPC <£8.0	15 type xyz type	3 x XYZ type		Etc.

©2016 Waters Corporation

21

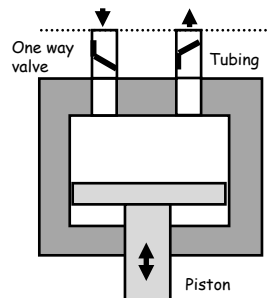
Determining Design Parameter Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- A reciprocating pump is selected as the design solution to the "Drain Water" functional requirement.
- The pump subsystem requirements flowed down from the system requirements are:
 - Flow Rate = 10 litres/min +/-1 litre
 - Reliability FYFR < 1%
 - DTUPC <£8.00

From first principles the transfer function for the flow rate F is



$$F = (10^{-6} \pi R^2 L - B) N$$

Where

R = Piston radius (M)

L = Stroke length (M)

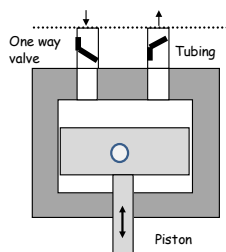
B = Back flow (l)

N = Motor speed (rpm)

©2016 Waters Corporation

22

Target (Nominal) Design



Using: **$F = (10^{-6}\pi R^2 L - B)N$**

A Nominal Design exists for

Piston Radius $R = 40\text{mm}$
Stroke Length $L = 40\text{mm}$
Valve Backflow $B = 0.002L$
Motor Speed $N = 50\text{ rpm}$

$$F = (10^{-6}\pi 40^2 \times 40 - 0.002)50 = 9.95\text{L/min}$$

It is decided that the valves and motor are to be bought out items with the following specifications (obtained from the OEM catalogues). The piston (and cylinder) and connecting rod will be made in house and using standard tolerances (same as last time)

	Make		Buy	
	Piston Radius R (mm)	Con-rod centre length L (mm)	Valve leakage B (L)	Motor Speed N (rpm)
Target	40	40	0.002	50
Lower Limit	39.7	39.4	0.00185	44
Upper Limit	40.3	40.6	0.00215	56

©2016 Waters Corporation

23

Assess Sensitivity of Nominal Design due to Manufacturing Variation

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

	R (mm)	L(mm)	B(L/min)	N(rpm)	F (L/min)
Target	40	40	0.002	50	9.95
High Flow	40.3	40.6	0.00185	56	11.5
Low Flow	39.7	39.4	0.00215	44	8.5

At the extremes of the expected manufacturing variation, the flow rate could vary from 8.5 up to 11.5 l/min – outside the design requirements. The current target (nominal) design is not robust against this source of variation.

What are we going to do?

©2016 Waters Corporation

24

Traditional Strategy: Tighten Tolerances

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- The design requirement of 10 ± 1 L/min can be achieved by tightening the tolerances
- It is possible to meet the design requirements by only tightening tolerances of the bought-out elements!

Tolerance Design: tighten the tolerances

	R (mm)	L(mm)	B(L/min)	N(rpm)	F (L/min)
High Flow	40.3	40.6	0.00194	53	10.9
Low Flow	39.7	39.4	0.00206	47	9.1

The design requirements can be achieved by tightening the tolerances on the bought-out elements. Looks good BUT costs will increase – usually faster than the losses

©2016 Waters Corporation

25

Robust Design Strategy: Parameter Design

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- A robust design strategy asks for a design iteration to find another point in the solution space that exploits the non-linearity in the system to find a “plateau” where the design is insensitive

Parameter Design

Iterate the design to find new target values such as R: 20, L: 60, N: 136 manufacturing variation remains the same

	R (mm)	L(mm)	B(L/min)	N(rpm)	F (L/min)
Target	20	60	0.002	136	9.98
High Flow	20.3	60.6	0.00185	142	10.9
Low Flow	19.7	59.4	0.00215	130	9.1

©2016 Waters Corporation

26

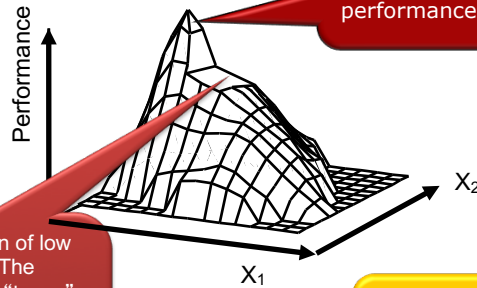
Searching the "Design Space" for the Robust Optimum

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

Robust design requires us to search the design space, not necessarily for the global-optimum, but for the robust-optimum.

Here is the global-optimum! But small variations in X_1 and X_2 will result in a dramatic fall off in performance. This is a sensitive design



Here is a region of low sensitivity. The performance is "lower" but will be maintained for likely variations in X_1 and X_2 . This is the robust-optimum and would be a robust design

BUT how do we "quickly" find the Robust Optimum?

©2016 Waters Corporation

27

Exploring the Design Space

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

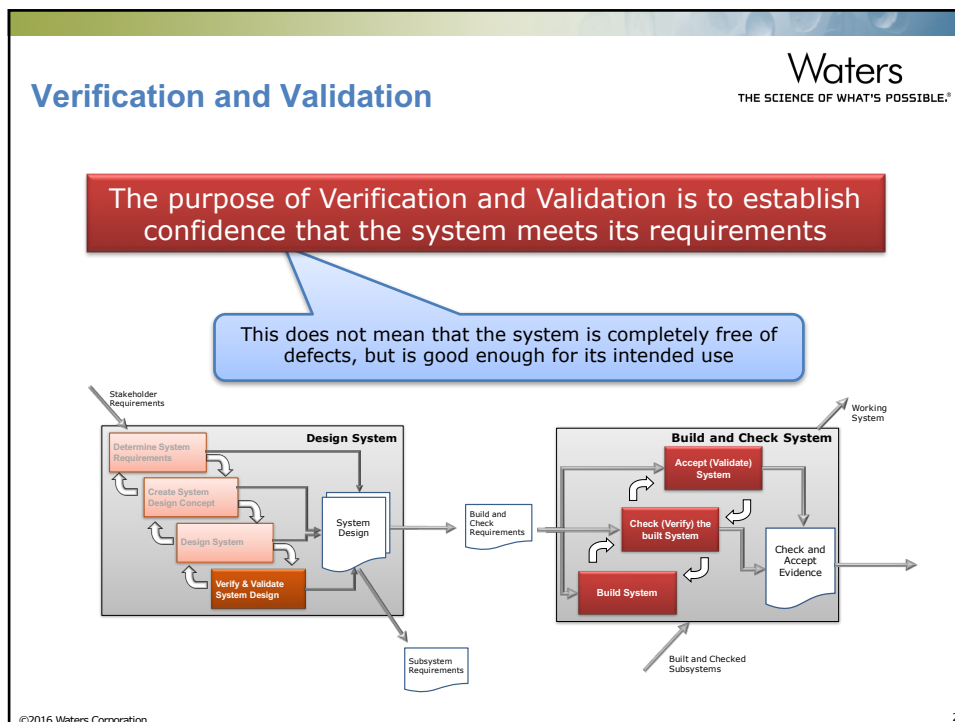
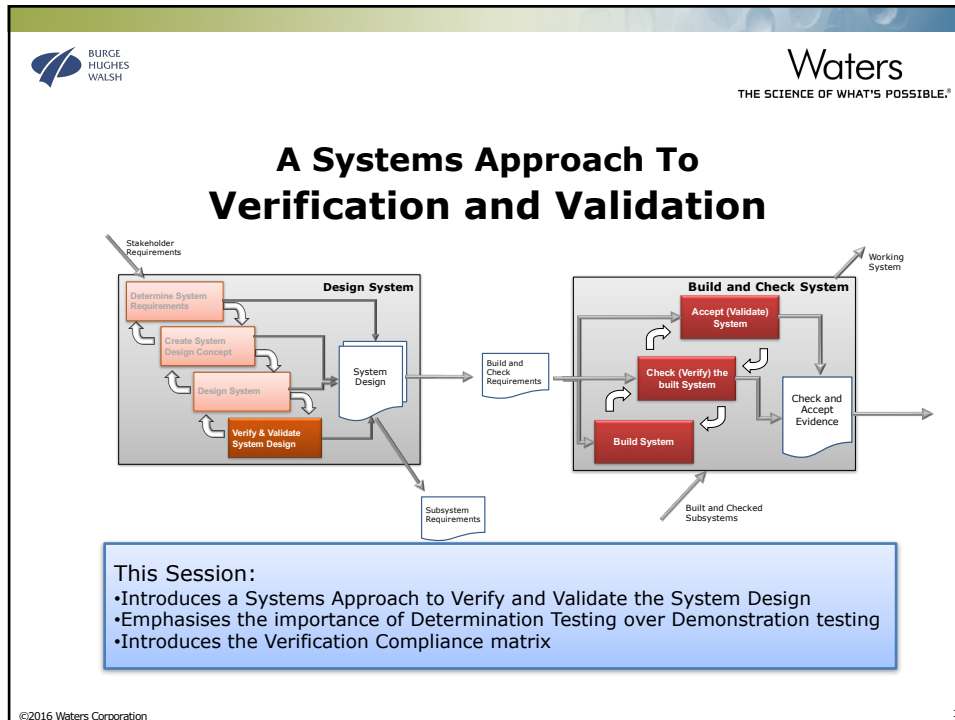
- The task is to characterise the design space to "find" the peaks, troughs and plateaus
- Approaches include:
 - Trial and Error: Searching the design space using judgement, hunches and other magical techniques
- Designed Experimentation
 - Systematically but scientifically search the design space using Design of Experiments

©2016 Waters Corporation

28

Summary

- Customers do not like sensitive systems – they appear to be unpredictable
- Parameter design provides an approach for finding the robust optimum without resorting to tightening of design tolerances
- It does require us however to find an efficient way of searching the design space
- At Waters the intention is to have a 5-day Robust Design training programme starting 2017



Confusion Confusion Confusion

- There is huge confusion over the terms verification and validation. Partly because in everyday language they are often used interchangeably
- The Oxford English dictionary gives:
 - Verify [verb]: make sure or demonstrate the (something) is true, accurate, or justified: *his conclusions have been verified by later experiments* (origin Latin *verificare*)
 - Validate [verb]: check or prove the validity or accuracy of: *all analytical methods should be validated in respect of accuracy* (origin Latin *validus*)
- The distinction is subtle, indeed Oxford English Thesaurus actually says that verify can be used in place of validate and vice versa

Systems Engineering Definitions

- Given the rather vague definitions from the Oxford English Dictionary, more pragmatic definitions of verification and validation are needed. An understanding can be obtained from various references
- NASA Systems Engineering Handbook:
 - Verification: It is essential to confirm that the realized product is in conformance with its specifications and design description documentation (i.e., verification)
 - Validation: However, from a customer point of view, the interest is in whether the end product provided will do what the customer intended within the environment of use (i.e., validation)
- These descriptions lead to the simple but pragmatic definitions:

Verification: Did we build the thing right?

Validation: Did we build the right thing?

Failure to Validate

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Concorde a great aeronautical achievement but the production run was for 14 and these were sold to British Airways and Air France for £1 each. There is no doubt that Concorde met its requirements but were they the right requirements? - a validation error. Concorde was a success only as a political statement, not as a business system.

The Tacoma Narrows Bridge was a scale-up of an old design! The design engineers reused the requirements for an existing bridge, so these requirements were up to the standards of the day. The bridge was built well but it was the wrong bridge for that environment, a validation error.



The Space Shuttle Challenger exploded because of low temperatures causing the O-rings to leak. The air temperature was below the design expectations, and no shuttle had ever been launched in such cold weather. The political decision was to launch in an environment for which the system was not designed, a validation mistake



©2016 Waters Corporation

5

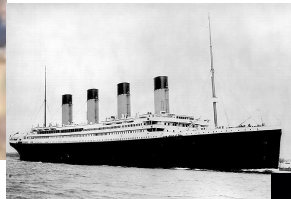
Failure to Verify

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



Ariane 5 blew up on its first launch, destroying a billion dollars worth of satellites. The mistakes on the Ariane 5 launch vehicle were (1) reuse of software on a scaled-up design, (2) inadequate testing of this reused software, (3) allowing the accelerometer to run for 40 seconds after launch, (4) not flagging as an error the overflow of the 32-bit horizontal-velocity storage register, and (5) allowing a CPU to shutdown if the other CPU was already shutdown. The requirements for the Ariane 5 were similar to those of the Ariane 4: So it was easy to get the requirements right. They needed a rocket with a larger payload, and that is what they got: So, that part of validation was fine. Their failure to adequately test the scaled-up software was a verification mistake.



RMS Titanic had poor quality control in the manufacture of the wrought iron rivets. In the cold water of April 14, 1912, when the Titanic hit the iceberg, many rivets failed and whole sheets of the hull became detached. Therefore, verification was bad, because they did not build the ship right.

The Hubble Space Telescope: Looking at the marvelous images we have been getting we can say that this was the right system. But during its development the guidance, navigation, and control (GNC) system, which was on the cutting edge of technology, was running out of money. So they transferred money from Systems Engineering to GNC. It was never tested. They said that the individual components all worked, so surely the total system will work. After they launched it, they found that the telescope was myopic - a verification error.



©2016 Waters Corporation

6

A Systems View

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- A systems view unifies these situations as examples of UNDESIRABLE

Emergent Behaviour

- .. and is why Systems Engineering, including Verification and Validation, is absolutely essential

©2016 Waters Corporation

7

How do we apply Systems Thinking to Verification and Validation?

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

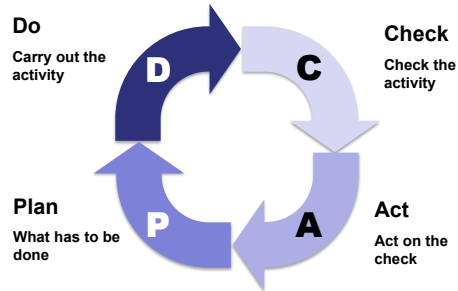
Systems Thinking can be applied to Verification and Validation at three fundamental, but interdependent, levels:

©2016 Waters Corporation

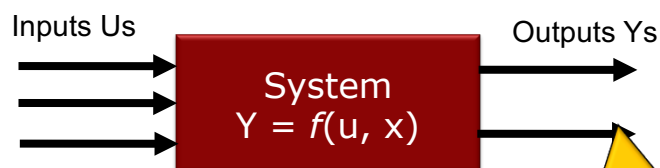
8

1. Verification and Validation are Types of Check

- The Plan Do Check Act (PDCA) cycle attributed to Edwards Deming is a Systems Thinking meta-process
- Verification and Validation are effectively an application of the "check" (and "act") in the PDCA cycle
- However, care needs to be exercised as to when this check is applied!



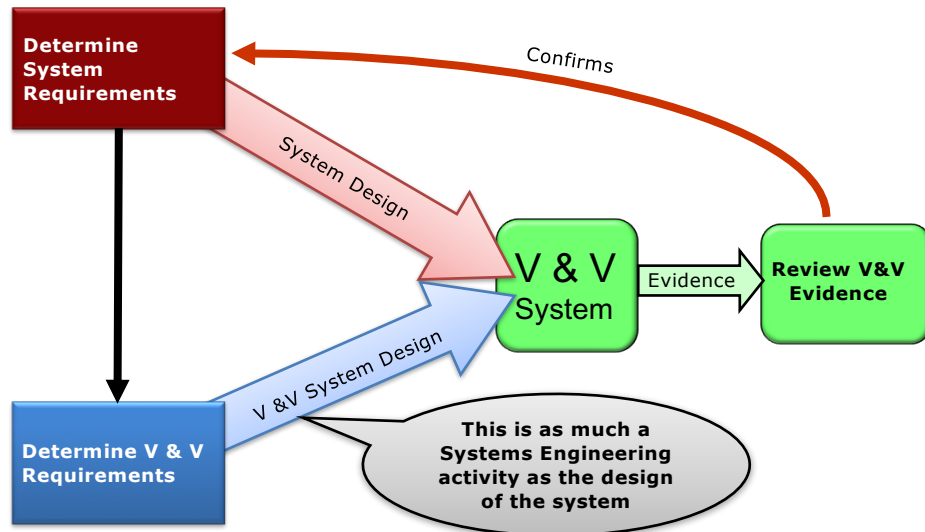
2. But you don't fatten lambs by weighing them!



If we understand $Y = f(u, x)$ we do not have to measure Y to assure its correctness! If we control the inputs, u , and the state, x , we know what Y is.

Measuring the output can be **too late!**
If the transfer function $Y = f(u, x)$ is understood then Y can be determined from a knowledge of x and u

3. Verification & Validation is a System



©2016 Waters Corporation

11

Methods for Verification and Validation

There are five basic methods for verification and validation

- Test** – **Test** is a method used to confirm conformance of performance or functional characteristics with specified requirements
- Demonstration** – **Demonstration** an activity to provide a demonstration of observable system actions - e.g. demonstrate that something is displayed, or demonstrate maintainability
- Review** – **Inspection/Review** is a visual or other examination of something to demonstrate it is present - e.g. code inspection or mechanical inspection
- Analysis** – **Analysis** is the use of a model to determine or demonstrate conformance – e.g. analysing test results or simulation results, usually where it is not possible to determine directly whether a pass or fail has taken place
- Analogy** – **Analogy** is the use of evidence of identical or similar systems in an identical or similar environment is used to confirm performance

©2016 Waters Corporation

12

Verification and Validation = Test?

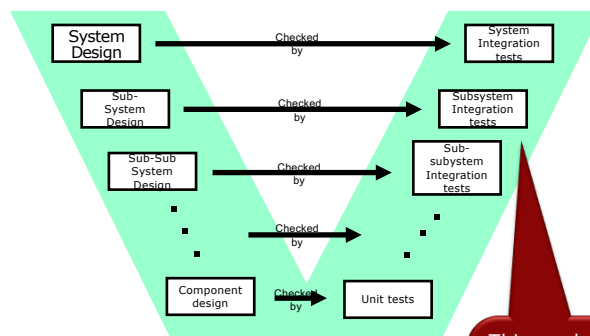
- We have become locked into the belief that testing is the ultimate method of verification and validation - because it is what the customer wants and expects



".. bump, bump, bump, on the back of , his head. It is, as far as he knows the only way of coming downstairs, but sometimes he feels that there really is another way, if only he could stop bumping for a moment and think of it.

- Testing in this context is often performed as a demonstration test on a pass-fail basis which is only conclusive if there is a problem! If there is no problem it is not conclusive
- This reliance on demonstration testing has led to a distortion of the V diagram

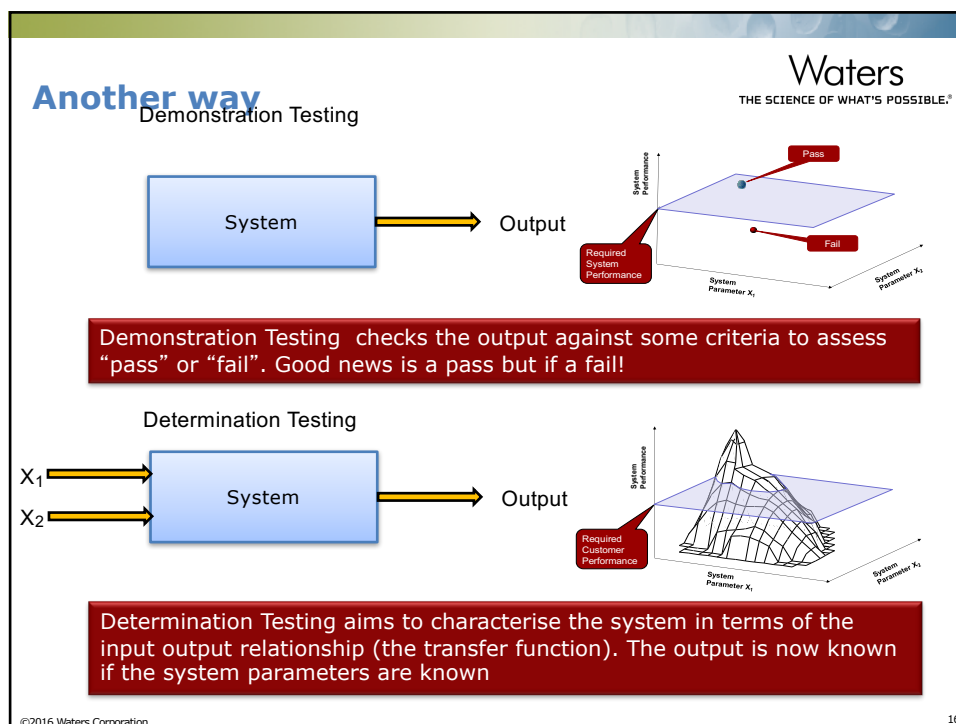
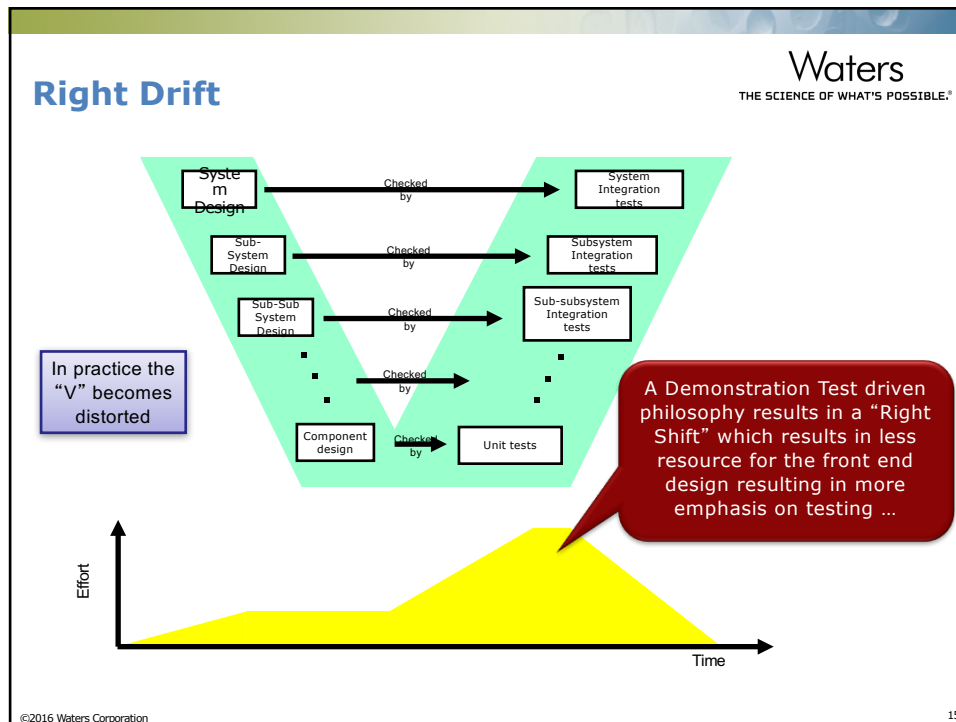
Theoretical V



Questions:

- If a "test" fails what happens?
- Where are the resources found to "fix" the issue?
- What happens to that work whose resource has been allocated to help "fix" the issue?

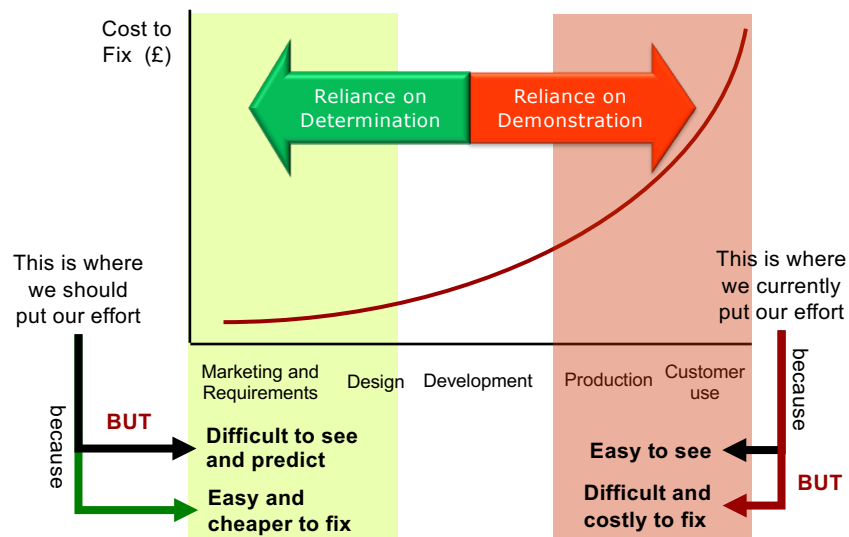
This model has been adapted from a very well known organization – Note the use of the word "tests" on the right hand side!

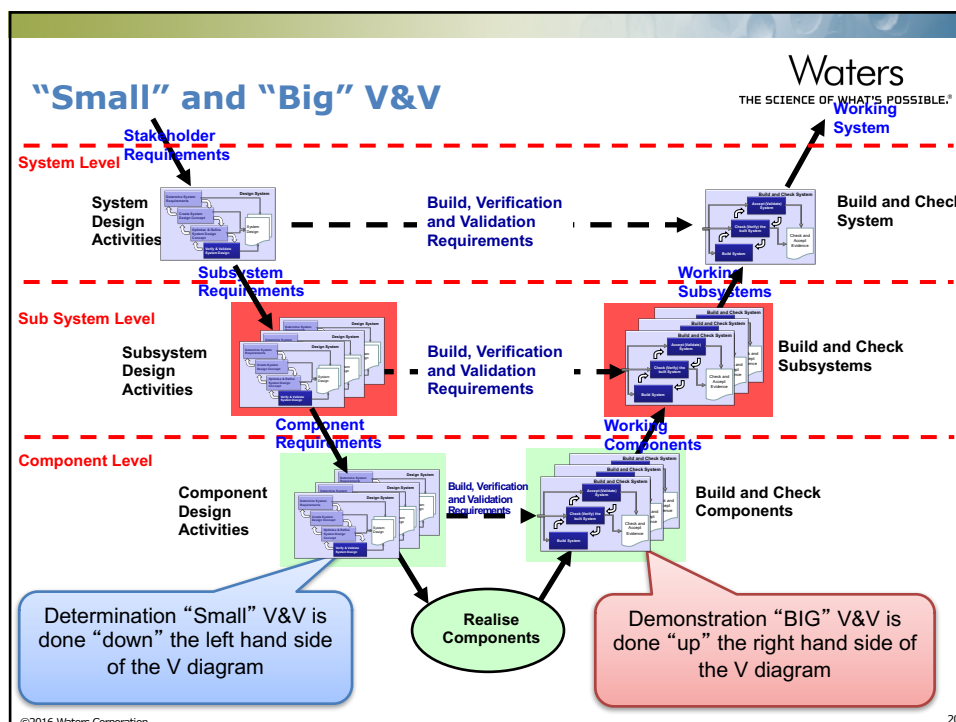
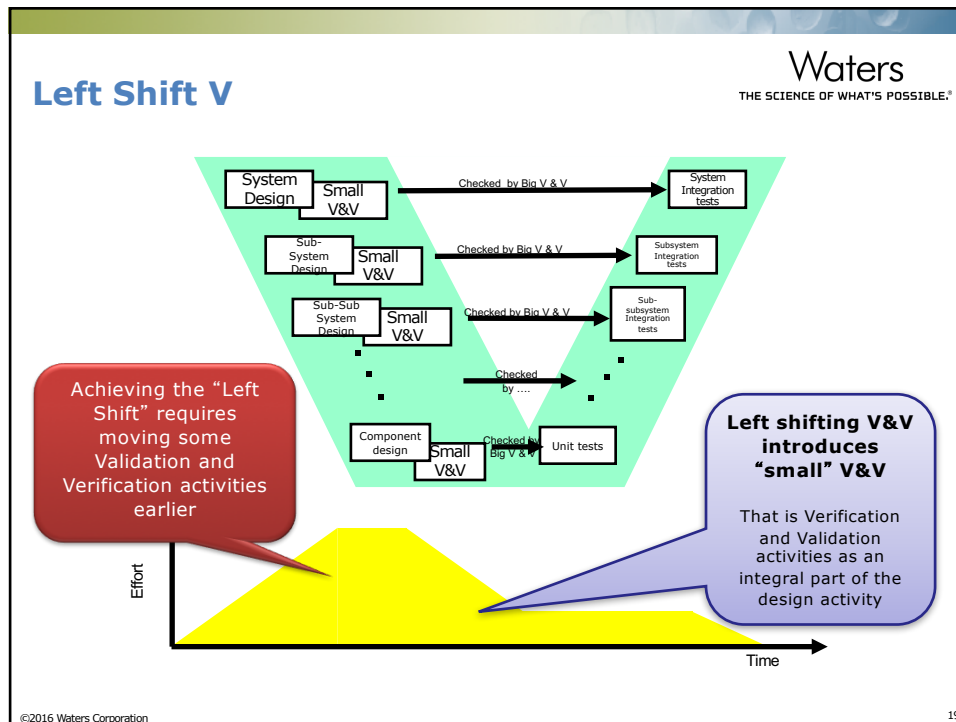


Implications of Determination

- The potential to use models to explore and determine system behaviour (by analysis) has implications for Verification and Validation
- There will be a shift in emphasis to:
 - use models (analysis) on the left hand side of the V to verify designs and design activities
 - use of tests/demonstration on the right hand side of the V to validate the models used to verify the system
- This means we need to think about our Verification and Validation strategy and subsequent plans early

The Systems Approach





Validation & Verification

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Firstly, validate the requirements
 - Do we have the right requirements?
 - Are we designing the right system?
- Secondly, verify the design
 - Does the design meet the requirements?
 - Have we designed the system right?

Design V & V

- Thirdly, verify the system
 - Does the system meet the design intent?
 - Have we built the system right?
- Finally, validate the system
 - Does the customer accept the system?
 - Have we built the right system?

Build V & V

©2016 Waters Corporation

21

Validation & Verification

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Firstly, validate the requirements
 - Do we have the right requirements?
 - Are we designing the right system?
- Secondly, verify the design
 - Does the design meet the requirements?
 - Have we designed the system right?

Design V & V

- Thirdly, verify the system
 - Does the system meet the design intent?
 - Have we built the system right?
- Finally, validate the system
 - Does the customer accept the system?
 - Have we built the right system?

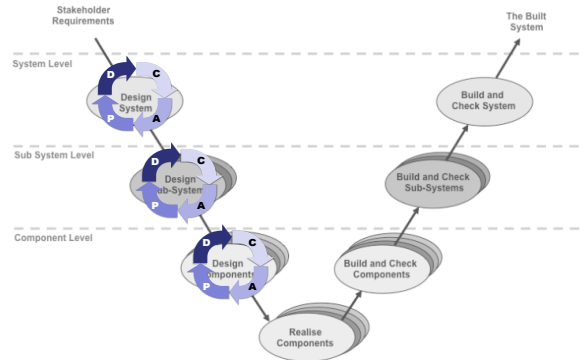
Build V & V

©2016 Waters Corporation

22

Verify and Validate Inside the V

- As system elements are designed common sense dictates that we check as we progress: that the
 - requirements are correct (are we designing the right system)
 - design meets the requirement (has the system been designed right?)
- This checking has two forms:
 - Verification – have we designed the system right
 - Validation – have we designed the right system
- Both are necessary to have complete confidence in the final system



By applying the PDCA within design levels provides greater confidence

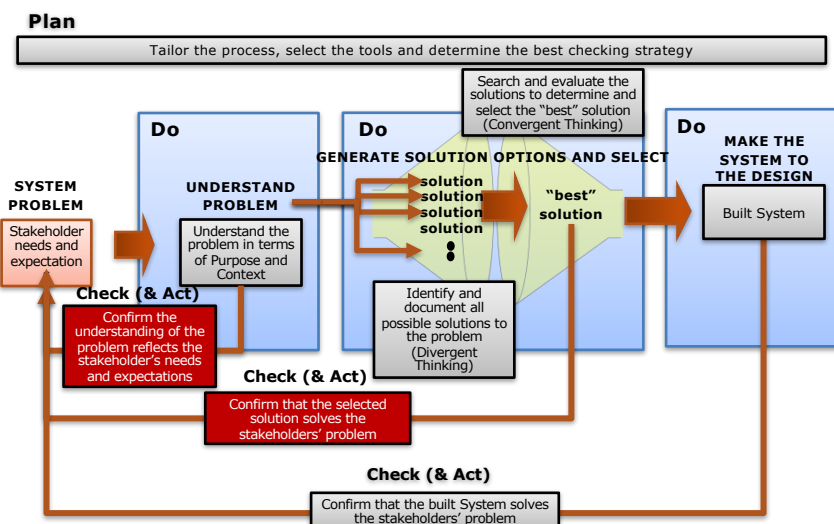
©2016 Waters Corporation

23

Systems Engineering uses a Systems Approach to System Design

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

24

V&V already done

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ Determine Requirements

- Used Flow sensitivity Analysis and FFMEA to identify potential issues early and design them out
- Used QFD to V&V stakeholder – system requirements and check for completeness and consistency

■ Create System Concept

- Used Function Means Analysis to ensure complete system solution
- Pugh to evaluate and select system concept
- Used QFD to V&V system requirements – System solution

■ Optimise System Concept

- Used DoE to explore the design space to find robust optimum (and avoid the peaks)

The important message here is that the work we have done using these tools is the evidence that is presented at DESIGN REVIEWS

Validation & Verification

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

■ Firstly, validate the requirements

- Do we have the right requirements?
 - Are we designing the right system?

■ Secondly, verify the design

- Does the design meet the requirements?
 - Have we designed the system right?

Design V & V

■ Thirdly, verify the system

- Does the system meet the design intent?
 - Have we built the system right?

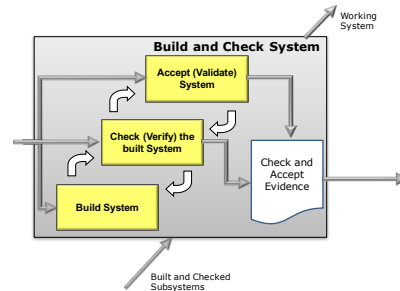
■ Finally, validate the system

- Does the customer accept the system?
 - Have we built the right system?

Build V & V

Big V & V

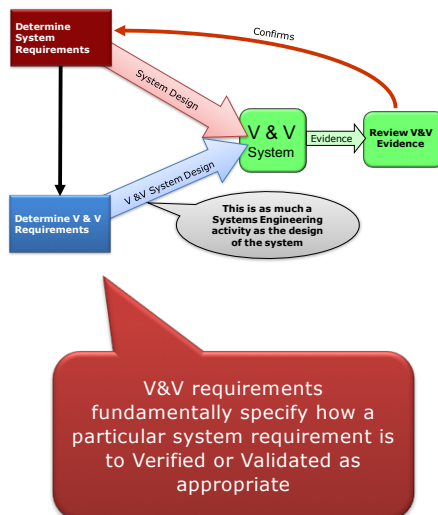
- Purpose :
 - To verify and validate that
 - Has the right system been built (Validate)
 - Has the system been built right (Verify)
 - in the context of meeting the requirements in terms of performance and the rest of its life cycle (manufacture, support and disposal)
- Outcomes :
 - A verified and validated system
 - Confidence that the system is
 - Operational
 - Reliable
 - Manufacturable
 - Supportable
 - Disposable
 - etc



©2016 Waters Corporation

27

It's about Designing Another System



- The Verification and Validation system can be designed in exactly the same way as the prime system
- Just as the prime system is reliant on a “good” set of design requirements, the same applies to the V & V requirements
 - These need to specify for each prime system requirement corresponding V&V requirements
 - V & V requirements should be written early to ensure that we V & V the system we should have built as opposed to system that was built
 - A useful tool is the Verification-(Validation) Compliance Matrix (VCM)

©2016 Waters Corporation

28

Verification and Validation Compliance Matrix

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

	Design	Calc	Analysis	Model	Test rig	Env test	Int test	Trials	Operations	Demo
Detection range										
Target CBA										
Cluster discrimination										
Classification										
Identification										
Tracking										
Lock-on										
Mass										
Power										
Volume										
Radarome swept area										
Transmissivity										
Aerodynamic										
Bird strike										
Driving rain										
Attachment strength										
Lightning strike										
RF energy SHC										
Maintenance access										
Integration										

What is it?

- A simple tool that can assist in the planning and management of V & V activities by:
 - o Recording the method by which each prime system requirement will be shown to have been met in the final system
 - o Capturing maturing compliance evidence
- Assists in ensuring that all requirements
 - o have one or more defined confirmation methods
- May be a by-product of the requirements management software being used

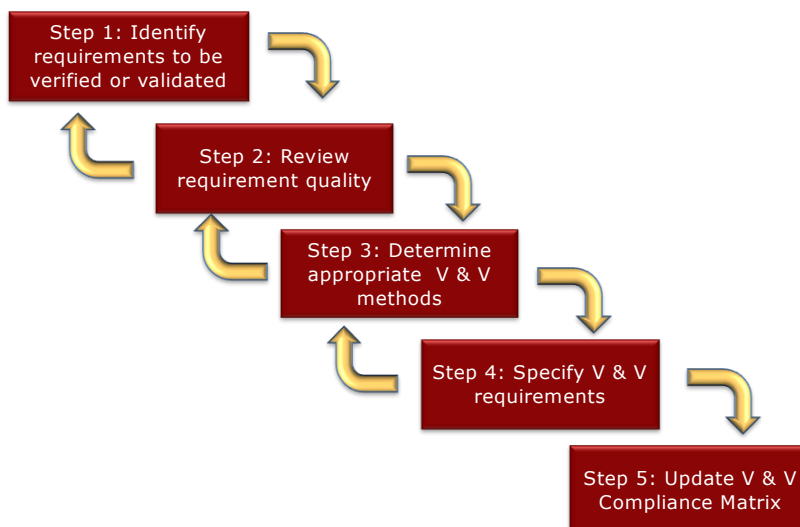
©2016 Waters Corporation

29

V&V Planning: How to do it

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



©2016 Waters Corporation

30

Step1: Identify Requirements to be Verified or Validated

- It is important remember the purpose of verification and validation:

The purpose of Verification and Validation is to establish confidence that the system meets its requirements

- This can be achieved by confirming that requirements have been met
- System Validation is concerned with confirming those requirements that are important to the "customer" have been met
- System Verification is concerned with confirming that other system requirements have been met. Verification therefore provides the opportunity to establish that Validation will be successful
- All requirements should therefore be identified and allocated either as part of the Validation set or Verification set
- It is important to agree the validation requirements with the respective stakeholders. These are sometime called Acceptance Requirements if the stakeholder is the economic customer, Qualification Requirements if the stakeholder is a legislator.

Step 2: Review Requirement Quality

- The key to using a VCM is to recognise that:

All requirements must be *measurable* so that they can be confirmed

Requirements must therefore be specified *quantitatively* -
i.e. present/not present, or have a measurable value -
and there must be a means to measure this

Example of an Un-measurable Requirement

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

"The washing machine must be easy to use"

This is not measurable so
it cannot be confirmed
quantitatively

Need to quantify it somehow

©2016 Waters Corporation

33

Review Process and Example

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

Non Quantitative Requirement

"The washing machine must be easy to use"

"Equivalent" Quantitative Requirement

At least 95% of novice users shall be able to wash the standard load correctly without recourse to the user guide.

Associated Definitions

A novice user is defined as someone who has used a conventional automatic washing machine before.

The Standard Load is defined in IS999 - 2013

Identify qualitative requirements

Determine equivalent quantitative requirement

Agree with Customer/
Stakeholder

©2016 Waters Corporation

34

Step 3: Determine Appropriate V&V Methods

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- Once we are confident that we have a measurable prime system requirement it is necessary to consider the most appropriate method or methods to be used to confirm the particular requirement
- It is important to note that V&V is about showing how a particular prime system requirement is to be shown to have been met in an evidence-based fashion
- It is equally important to note that V&V is not an activity that is "left" to the end! It can be a continuous activity that occurs throughout the development cycle

©2016 Waters Corporation

35

Step 4: Specify V&V Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

"I keep six honest serving-men: (they taught me all I knew) Their names are What and Where and When and How and Why and Who."

The Elephants Child 1902

- V&V requirements are concerned with documenting:
 - The justified (WHY) method or methods (HOW): demonstration, test, analysis, inspection or analogy, most suited to the particular prime system requirement under consideration. Note that several methods may be employed. Indeed, the systems approach emphasises analogy, analysis and review.
 - The requirements applicable to the chosen methods (WHAT) and the acceptance criteria. They should also specify what evidence is to be gained and how it should be reported
 - WHEN the V & V activity is to be carried
 - WHERE are the V & V activities to be performed
 - WHO is responsible for performing the V & V activity

©2016 Waters Corporation

36

V&V Requirements Example

Requirement (based on the meta solution of a vacuum toilet)

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- T001.1 The flush toilet function shall only be activated when
- a flush has been requested
 - no-one is sat on the toilet at the time of the flush request

V&V Requirements

Requirement T001.1 shall be confirmed by:

- Analogy during design with existing designs
- Design Review of the concept and detailed design
- Demonstration through a series of tests being performed on the development system

During design the selected solution shall be examined and evidence collected for suitable carry-over from previous successful designs. Evidence shall also be gathered to show that known failure modes have been adequately addressed and avoided.

The demonstration tests will involve activating the flush request when:

1. no-one is sat on the seat
2. a series of different sized individuals from a small child to a range of adults are sat on the seat

Note: Sizes of individuals to be consistent with 95 percentile standards

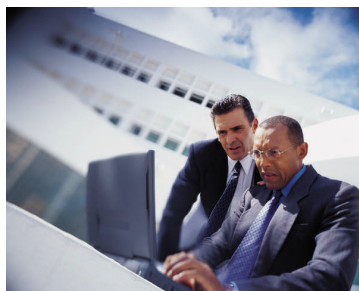
Expected Results: the demonstration will show the system to NOT to perform a vacuum flush when any person is seated and a successful flush when no person is seated

Systems Test will be responsible for developing, performing and recording the test

Exercise :Verifying Requirements

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®



- In pairs consider the case study exercise:
 - Use one of the requirements you generated in the "Specify Requirements" exercise and determine suitable verification requirements
- Time 20 minutes

Generating V & V Requirements is itself a V & V activity

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

- The very act of writing V & V requirements for a particular system requirement often highlights deficiencies in that requirement!
- In essence the activity serves as a "check" on the quality of the system requirement
- It is also important to guard against over "checking". V & V is a cost and we are seeking to determine the minimum acceptable action necessary to confirm that a particular system requirement has been met
- It can also be useful to specify what actions are necessary if a particular requirements is not met or partially met
- Finally, if a system requirement is subject to change, then the corresponding V & V requirements will need to change

©2016 Waters Corporation

39

Managing V&V Maturity

Waters
THE SCIENCE OF WHAT'S POSSIBLE.®

The VCM can also be used to capture the ongoing verification (and validation) maturity

Requirement	Demonstration	Review	Analogy	Analysis	Test
R001.01					
R001.02					
R001.03					
R002.01					
R002.02					
R00n.0n					

Each requirement has one or more V&V methods and requirements declared, together dates to present a complete V&V plan. This VCM can be used to form the basis of a plan for demonstrating compliance and agreeing early payment plans with the customer.

It can also be used as a simple method for monitoring progress can be indicated using colours, e.g.:

Planned 
 Achieved to plan 
 Risk to plan 
 Late to plan 

This capability is often an integral part of Requirements Management software

©2016 Waters Corporation

40

Summary

Waters

THE SCIENCE OF WHAT'S POSSIBLE.®

- Both Verification and Validation are necessary
- There are 5 basic methods of performing Verification and Validation
- Verification and Validation is itself a system that needs designing from a set of requirements derived from the system requirements
- Verification and Validation should be evidence based
- Verification and Validation happens on both the left and right hand sides of the V – key message start V & V early
-