

Managing Requirements for Service-oriented Architectures (SOA)

*Anthony Kesterton, Technical Consultant
Rational software, IBM
akesterton@uk.ibm.com*

IBM Rational Software Development Conference UK 2007



▶ What keeps me **Rational**?



Agenda

- Introduction
- What is Service-oriented Architecture?
- BIG SOA requirements
- Use cases in SOA-based solutions
- little soa requirements
- Other important concepts
- Summary



Agenda

- Introduction
- What is Service-oriented Architecture?
- BIG SOA requirements
- Use cases in SOA-based solutions
- little soa requirements
- Other important concepts
- Summary



Introduction

- Why are SOA requirements important?
- Service-oriented architecture (SOA) is very topical!
- Where do Requirements and Requirements Management fit in?
 - ▶ Where do use cases fit in (and user stories, storyboards, etc)?
- But SOA is an architectural style....so should this really affect requirements?
- IBM Rational Unified Process
 - ▶ Service-oriented modeling and architecture (SOMA) capability pattern



Agenda

- Introduction
- What is Service-oriented Architecture?
- BIG SOA requirements
- Use cases in SOA-based solutions
- little soa requirements
- Other important concepts
- Summary

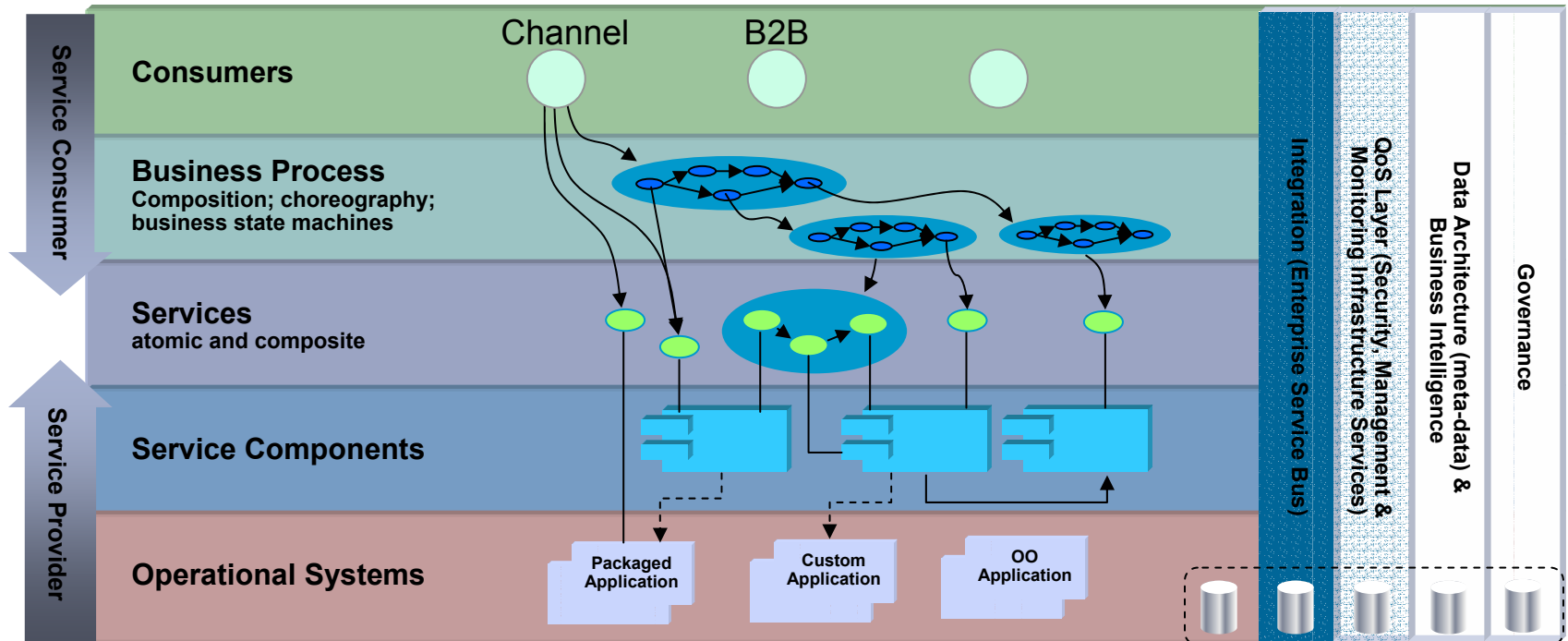


What is Service-Oriented Architecture (SOA)?

- *“An approach for designing and implementing distributed systems that allows a tight correlation between the business model and the IT implementation”*
- Service-oriented architecture is also
 - ▶ Architectural style
 - ▶ Set of technologies
 - ▶ The answer to everything (or maybe not)
- Distinguish between SOA-based solutions, creating services and consuming services
 - ▶ We will look at all these aspects



A Service-Oriented Architecture-based solution



Snake Oil-oriented Architecture?

- ▶ “Now, let me make something excruciatingly clear, lest you misunderstand me and thus send incendiary emails and/or impel my IBM management to take me behind the woodshed for a good thrashing: I am a strong proponent of Service-Oriented Architectures (SOA)...
- ▶ ...IMHO, **SOA's value proposition begins with the A in its acronym: architecture.**” [Booch, 2006]

Reminder: What are requirements?

- A Requirement
 - ▶ "a condition or capability to which the system [being built] must conform."
[Oberg et al, 2000]
- Software Requirements
 - ▶ "A specification of an externally observable behaviour of the system; for example, inputs to the system, outputs from the system, functions of the system, attributes of the system, or attributes of the system environment." [RUP, 2007]
- Requirements Management
 - ▶ A systematic approach to eliciting, organizing and documenting the software requirements of the system, and establishing and maintaining agreement between the customer and the project team on changes to those requirements.
[RUP 2007]
- Now replace the word "system" with the word "solution" – it helps to think about SOA.



Point for debate

From a requirements-management perspective, does it matter if we are going to solve the problem using an SOA-based solution or some other architecture?



Our perspective

- NO!
- BUT
 - ▶ We still need requirements and requirements management when building solutions based on SOA.
- Poor requirements management will result in solutions that potentially can cause *more* problems in an enterprise than typical projects
 - ▶ SOA-based solutions generally exposed to the outside world
 - ▶ Services potentially introduce dependencies that cross Lines-of-Business
 - ▶ Non-functional requirements are particularly important



Other contrasts

- BIG SOA vs little soa
 - ▶ BIG SOA are solutions are ones that solve business problems, crossing internal and external organisational boundaries
 - ▶ Little soa are more technical oriented, low-level services
 - ▶ Impact on requirements...
- Solution vs Services
 - ▶ Solutions will be composite services
 - ▶ How do we handle requirements?
- Service consumption vs service creation
 - ▶ How does that affect requirements?



Agenda

- Introduction
- What is Service-oriented Architecture?
- **BIG SOA requirements**
- Use cases in SOA-based solutions
- little soa requirements
- Other important concepts
- Summary



BIG SOA requirements

- A project or even programme of work
- We need to capture
 - ▶ Business goals
 - ▶ Business metrics and key performance indicators
 - ▶ Business processes
 - ▶ Business use cases
 - ▶ Other functional requirements
 - ▶ Non-functional requirements
 - Usability
 - Reliability
 - Performance
 - Supportability
 - Design constraints



Aside: Business modeling and requirements management

- To be relevant to the business – SOA-based solutions need to be aligned to business goals
- Traditional business process and requirements techniques are inter-related and relevant to SOA-based solutions
- Relevant business modeling work products
 - ▶ Business Vision
 - ▶ Business Goals
 - ▶ Key performance indicators (KPI) and Metrics
 - ▶ Business Use Case model
 - ▶ Business Glossary
 - ▶ Domain Model
 - ▶ Business Process Model



Business Goals

- “A business goal is a *requirement* that must be satisfied by the business. Business goals describe the desired value of a particular measure at some future point in time and can therefore be used to plan and manage the activities of the business.” [RUPv7, 2007]
- Goal as a requirement on your SOA-based solution
- Keeps solution focussed on the problem to be solved
- Example:
 - ▶ Reduce application processing costs by 10% by the end of December 2007
- Goals may also have sub-goals
 - ▶ Example: Reduce the number of applications that require external credit checks by 30%



Business Metrics

- A business metric is an value that is measured or derived from the business
- Metrics and KPI's are very important for SOA-based solutions (BIG SOA)
- Metrics tell you what measures are needed to prove you met the business goals
- Metrics and measures could be important for services requirements (little soa) as they provide potential non-functional requirements, and imply functional requirements
- Example metric
 - ▶ Total account application cost in \$\$\$
 - ▶ Implies we need to be able to work out costs



Key Performance Indicators

- A Key performance indicator “...measures performance against business objectives” [SOA Redbook]
- This shows you how to demonstrate you have succeeded
- Example KPI:
 - ▶ Total account application processing cost reduced by 10% from the current application processing cost (June 2007)



Point for debate

Is the business process a requirement or a solution?

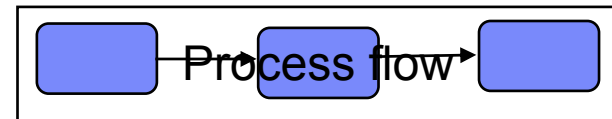


Our viewpoint

- The business process is a source of requirements and constraints
 - ▶ Provides tasks that may be automated
 - ▶ Defines roles that perform the tasks
 - ▶ Provides input to service identification
- We are very certain that the business process being automated is ***not*** the ***only*** source of requirements

Business process model

- An SOA-based solution will typically automate part of a business process, or the entire business process.
- This business process model will need to be captured
- Provides at least part of the required functionality of the solution
 - ▶ Automation or partial automation of processes
 - ▶ Also contains metrics, rules, and other information



Business rules

- “...a business rule is guidance that there is an obligation concerning conduct, action, practice, or procedure within a particular activity or sphere.” [Business Rules Group]
- For IT systems
 - ▶ “...a business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behaviour of the business.” [Business Rules Group]
- Will need to be encoded in the SOA-based solutions
 - ▶ Rules engines
 - ▶ Workflow engine
- Typically externalised from applications
- These will be constraints on the solution



Agenda

- Introduction
- What is Service-oriented Architecture?
- BIG SOA requirements
- **Use cases in SOA-based solutions**
- little soa requirements
- Other important concepts
- Summary



Point of debate

What are use cases?

Only joking....



What are use cases?

- “Use cases represent things of value that the system performs for its actors. Use cases are not function or features, and they cannot be decomposed.” [Bittner, Spence, 2003]
- A **conversation** between the outside world and the system or business
- Use cases are still very relevant to capturing requirements for SOA-based solutions, but not quite the way we use Use Cases in application development

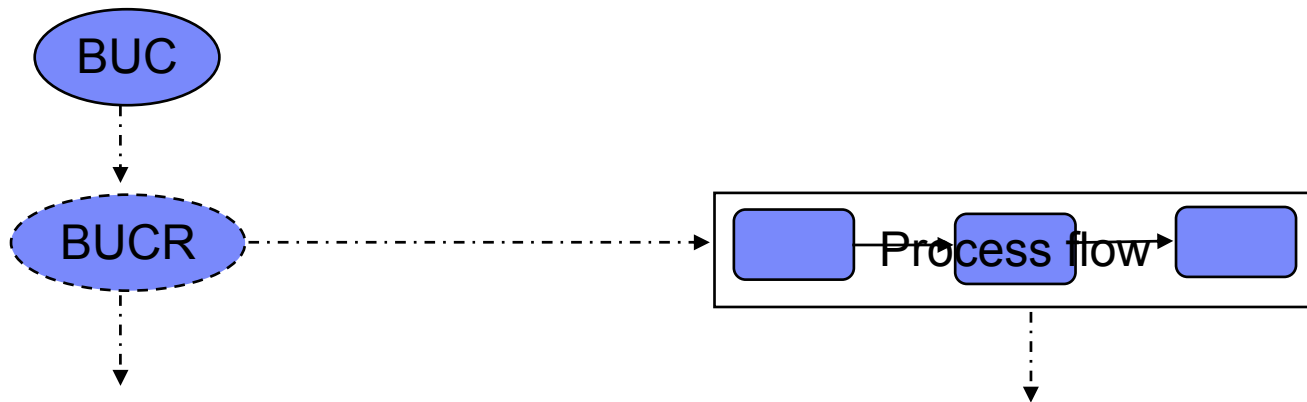
Business Use Cases

- Focus on the functional requirements of the business
- “A business use case defines a set of business use-case instances, where each instance is a sequence of actions a business performs that yields an observable result of value to a particular business actor. A business use-case class contains all main, alternate workflows related to producing the "observable result of value“ [RUP v7, 2007].
- *a sequence of actions a business performs that yields an observable result of value to a particular business actor*



Business Use Cases and SOA

- The business use case gives a compact overview of what the business is trying to achieve
- Process flows are really business use case realisations
- Can continue with RUP business use case realisations to help identify services



Aside: Next step - Identify Services

- The next step in building SOA-based solutions will be to identify candidate services
- RUP Task – Identify services
- Approaches
 - ▶ Top down - Business processes driven
 - ▶ Top down – (Business) use case driven
 - ▶ Bottom-up – Exposing existing assets
 - ▶ Data-driven
 - ▶ Rule-driven



Use case realisations and service identification

- “A use-case realization describes how a particular use case is realized within the design model, in terms of collaborating objects.” [RUP, 2007]
- Similar definition for Business Use Case Realisations
- Interactions between the outside world, and internals of the solution
 - ▶ The interactions are potential services!
- Business use case should “work” in isolation
 - ▶ Which implies that the realisations should also be independent of the other realisations
 - ▶ Which implies we will discover the services we need for each business use case
 - ▶ Which implies we discovered the “Really required services” (RRS)
 - ▶ Later we can rationalise the list of services and look for commonality and reuse

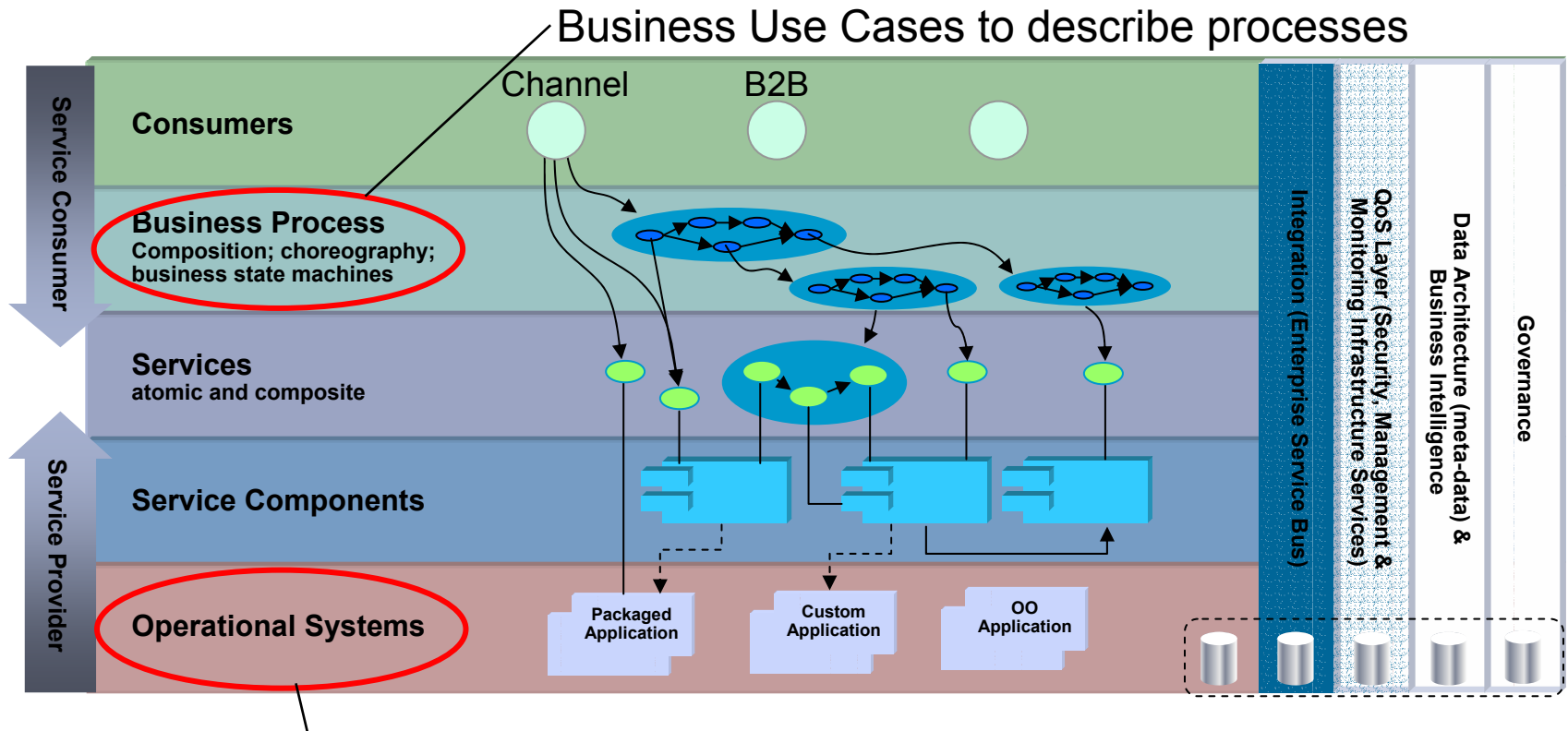


So how should we utilise Use Cases in SOA?

- Top-down SOA
 - ▶ Define business use cases to discover and document the intent of the business
 - ▶ Use the interactions inside business use case realisations to help identify candidate services
- Bottom-up SOA
 - ▶ Existing Asset Analysis
 - ▶ To describe the behaviour of systems that provide the services
 - ▶ A use case model describes the complete system, our service or services may only use an aspect of that system



A Service-Oriented Architecture-based solution



System Use Cases to describe apps



Agenda

- Introduction
- What is Service-oriented Architecture?
- BIG SOA requirements
- Use cases in SOA-based solutions
- **little soa requirements**
- Other important concepts
- Summary



little soa Requirements: Service Creation

- Look at creating services first
- We need the following information about a service
 - ▶ What it should do?
 - ▶ Quality of Service (QoS)
 - ▶ Service Level Agreement (SLA)
- These are similar to **any** project's requirements
- Need to focus on reuse
 - ▶ If not – why bother making this a service?



little soa Requirements: Service Consumption

- This is similar to service creation requirements
 - ▶ What the service should do
 - ▶ What qualities of service are required
 - ▶ What SLA exists
- But also
 - ▶ Who provides the service?
 - ▶ What is the cost of the service?
 - ▶ How long can we rely on this service?



Agenda

- Introduction
- What is Service-oriented Architecture?
- BIG SOA requirements
- Use cases in SOA-based solutions
- little soa requirements
- Other important concepts
- Summary



Reviewing SOA requirements

- BIG SOA
 - ▶ Business should validate goals, metrics, KPI's and processes
 - ▶ Architecture board should review to identify potential for reuse
- little soa
 - ▶ Service creation
 - Should this be a new service? Architecture board
 - (Potential) Service consumers – fit for intended purpose?
 - ▶ Service consumers
 - Are these requirements complete?
 - Will a service that meets these requirements meet our needs?
 - Confirm SLA's, QoS
 - Similar to package selection

SOA and testing

- Requirements drive test planning as per usual
- SOA-based solutions exposed to the outside world
 - ▶ Implies more rigorous testing!
 - ▶ External interfaces
 - Security
 - QoS
- Services
 - ▶ Creation
 - Non-functionals such as behaviour under load, failover, conformance to standards
 - ▶ Consumer
 - Behaviour as expected?
 - Error responses?



Process and automation

- Process
 - ▶ RUP with the latest SOMA content (v7.1 or greater)
- Automation
 - ▶ IBM Rational RequisitePro
 - Business modeling template
 - SOA Redbook examples
 - ▶ IBM WebSphere Business Modeler
 - Recent enhancement – WBM to RequisitePro integration
 - ▶ IBM WebSphere Service Registry and Repository
 - Service management
 - ▶ IBM Rational ClearQuest
 - Test management
 - Connection to requirements in RequisitePro



Agenda

- Introduction
- What is Service-oriented Architecture?
- BIG SOA requirements
- Use cases in SOA-based solutions
- little soa requirements
- Other important concepts
- Summary

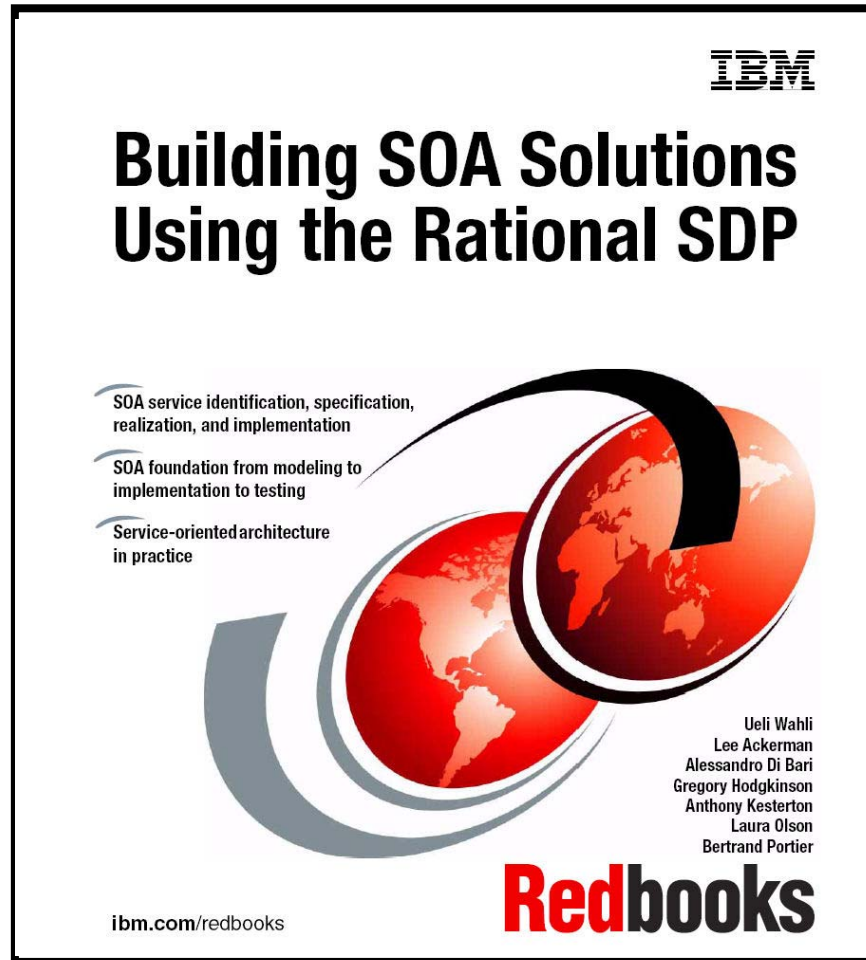


References

- Booch [2006], http://www-03.ibm.com/developerworks/blogs/page/gradybooch?entry=snake_oil_oriented_architecture
- Wahli, et al [2007], “Building SOA Solutions using the IBM Software Development Platform”
- [RUP v7, 2007] IBM Rational Method Composer v7.1, Large project RUP.
- Bittner, Spence [2003], “Use Case Modeling”, Addison-Wesley
- Oberg et al [2000], “Applying Requirements Management with Use Cases” [part of RUP 2007]



References



Summary

- Requirements are still required!
- Different requirements focus for SOA-based solutions
- Contrasts
 - ▶ Solution vs service requirements
 - ▶ Existing service vs new service requirements
- Non-functional requirements are critical
- Reviewing, testing, process and automation
- Redbook





Questions





Thank You

Anthony Kesterton
akesterton@uk.ibm.com

