



SOGETI

# Practices of Quality and Testing when Implementing Agile

*October 14<sup>th</sup> 2010*

*Ken Brennock, Principle Consultant Sogeti Ireland*



# Abstract

Agile implementations often struggle to implement effective approaches to testing that achieve productivity enhancements with the required level of quality. The aim of this presentation is to provide insights into the various approaches and practices of quality and testing when implementing Agile methodologies. The presentation will examine past experiences where quality/testing have often not been successful, to the present where testers are starting to understand their role in a wide variety of agile team implementations and will conclude with considerations of what the future might hold for role of testing in agile projects.

# Agenda

- **Introductions**
- **As an attendee I would like a brief review of what is Agile**
- **As an attendee I would like to know testing issues in Agile**
- **As an attendee I would like to see some solutions to these issues**



WHO  
ARE

YOU?



# **User Story #1**

**As an attendee I would like a brief review of what is Agile**

**How will we test this?**

# Agile Manifesto

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

“That is, while there is value in the items on the right, we value the items on the left more.”

# Agile Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

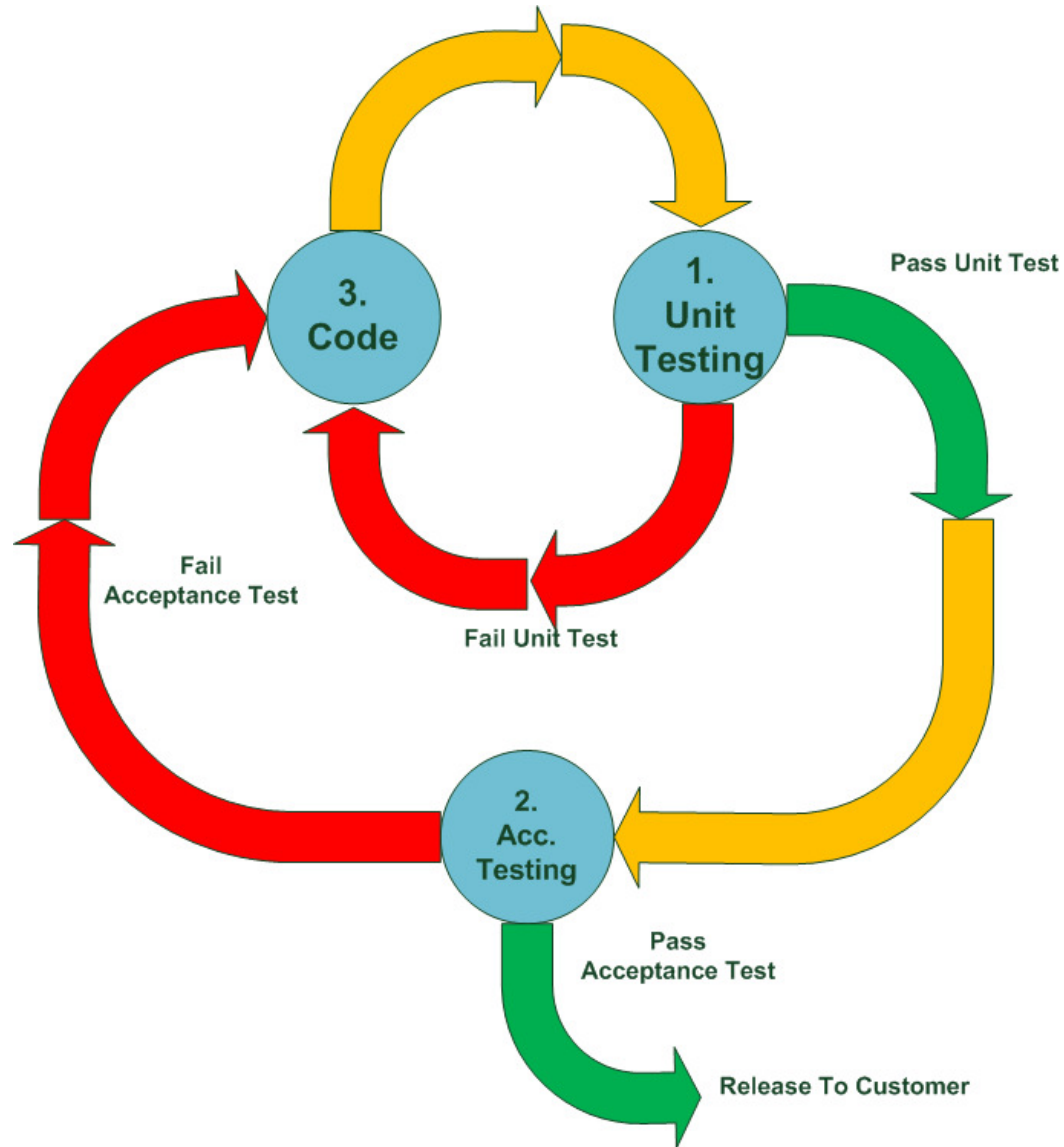
# Some differences between Agile Quality and Plan-driven QA

	<b>Agile Quality</b>	<b>Plan-Driven QA</b>
<b>Who is responsible for software quality?</b>	All development team members	The QA team
<b>When are quality-related topics addressed?</b>	During the entire software development process; quality is one of the primary concerns of the development process	Mainly at the QA/testing stage
<b>Status of quality-related activities relatively to other software development activities</b>	Same as other activities	Low-Medium
<b>Work style</b>	Collaboration between all role holders	Developers and QA people might have conflicts

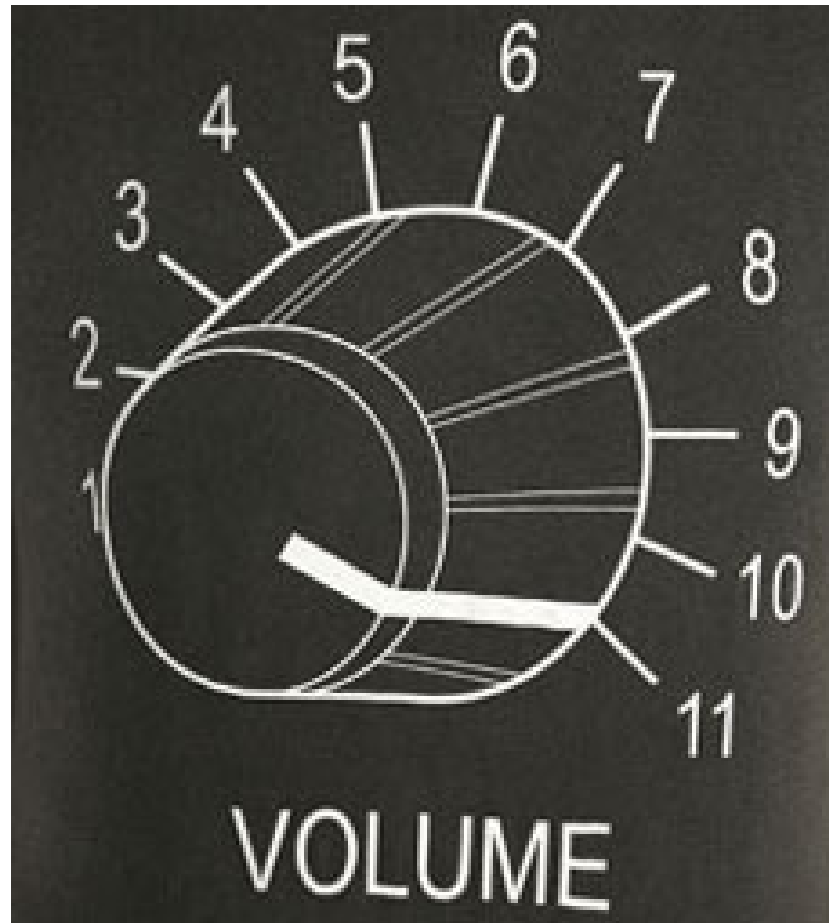




# Agile Development



Turning the knobs to 10...



# Misconceptions about Agile

- **Agile teams don't plan**
  - > They plan continuously
- **Agile lacks process**
  - > Agile teams reject heavy predefined/standardised process
  - > They define and refine process that works best for them
- **Agile lacks discipline**
  - > Agile thrives on discipline (e.g. see XP): strict timekeeping and iteration plan adherence

# User Stories

- **A key change of approach is to divide the project into individual features prioritised by business value**
  - > **Each feature (or ‘story’) must be**
    - Stand alone
    - Have clear meaning and explicit value to customer
    - Possible to estimate for dev/test
    - Possible to prioritise by customer against other features
    - Possible to develop and test to completion
  - > **Features must be completed before another can start**
    - Analysed/Designed, coded, tested, integrated
  - > **At the deadline, a working (sub)system (set of features) is ready to release**
  - > **Any slippages results in low priority features dropped**

# Agile Methodologies

- SCRUM – lightweight work management framework
- eXtreme Programming (XP) – disciplined/rigorous set of practices for agile development
- Crystal – lightweight set of agile development practices
- Lean Development (from Lean Manufacturing)
- DSDM – RAD based approach
- Adaptive Software Development
- Feature Driven Development
- Context Driven Development

*In practice we often observe partial implementation and/or mixing with traditional plan driven approaches to create hybrid combinations*



**User Story #1**  
**Done?**



# **User Story #2**

**As an attendee I would like to know testing issues in Agile**

**How will we test this?**



# Issues / Problems

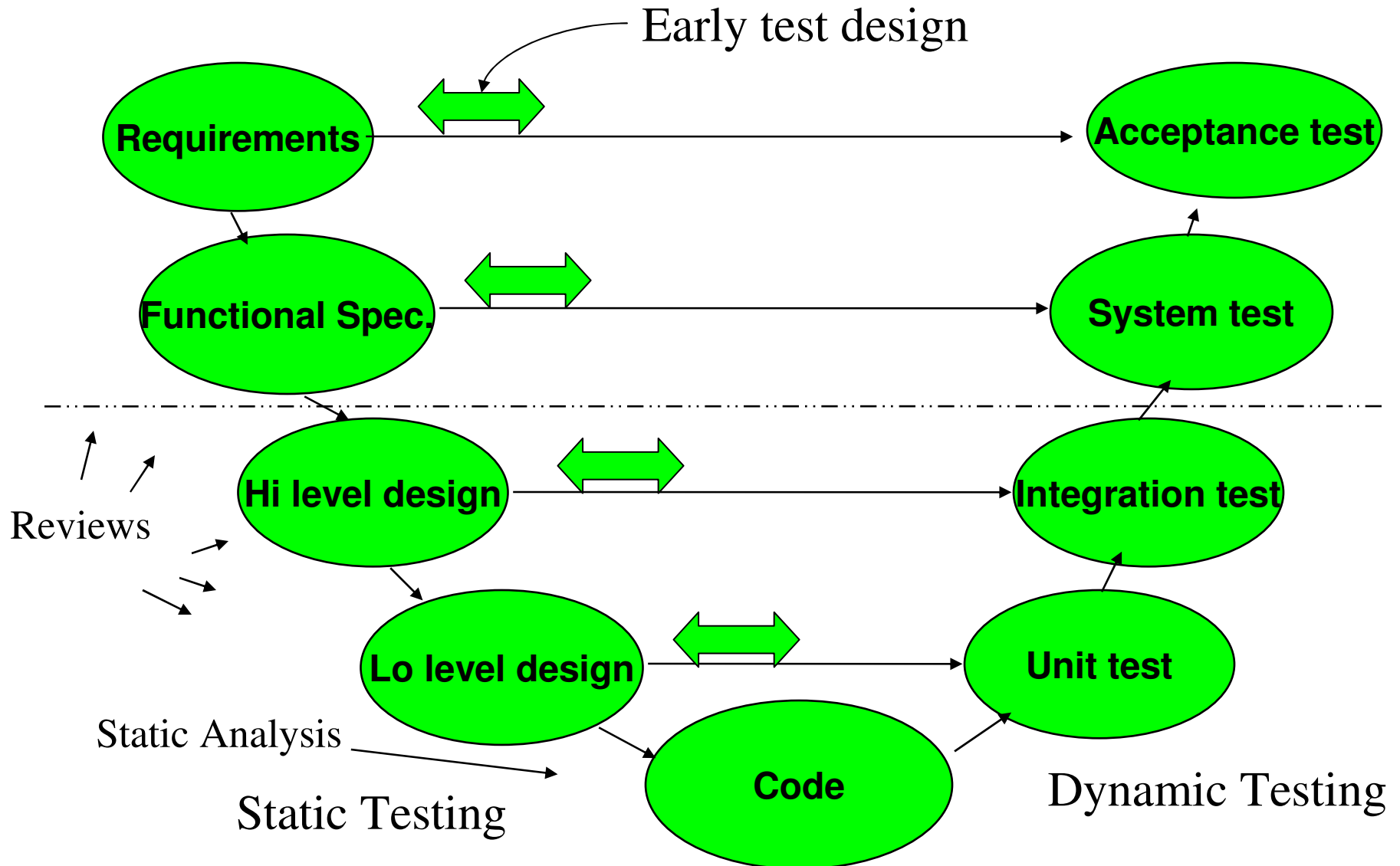
**What are your problems?**

# General Issues - In Practice

## Why it doesn't always happen

5. Skilled / Experienced Team
4. Working Software / Testing / Quality
3. Focus
2. Collaboration / Communication
1. Trust

# Lack of Time for Testing



# New Culture

`The nature of the tester's role changes in iterative projects. We are no longer the high-profile victims, we are no longer the lonely advocates of quality, we are merely (!) competent service providers, collaborating with a group that wants to achieve high quality. `

**(Cem Kaner)**

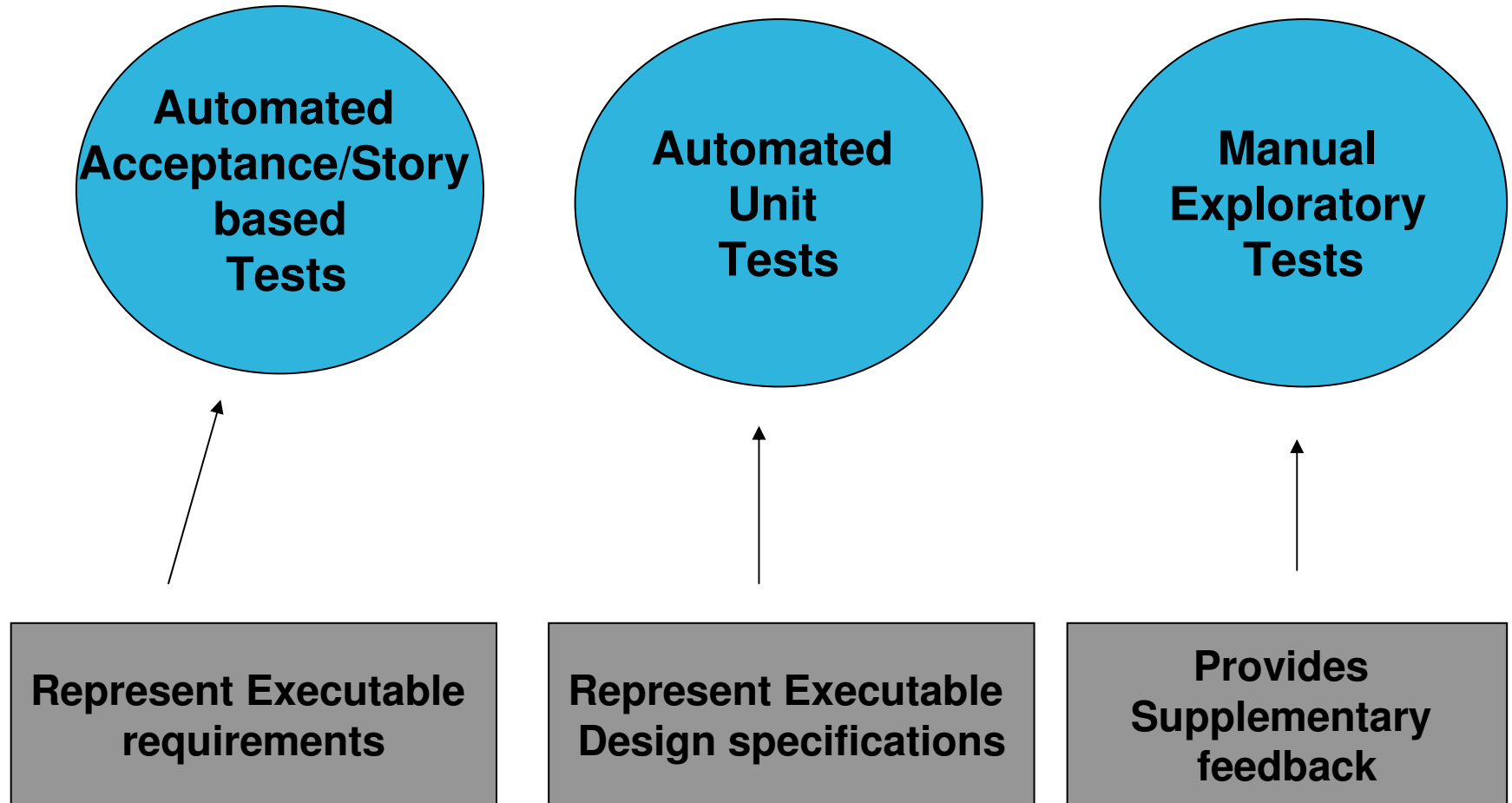
Dedicated testers bring two benefits:

Focus on customer usage over technical implementation

Focus on uncovering flaws over confirming completeness

**(Bret Pettichord)**

# Only Three Test Levels





**User Story #2**  
**Done?**



## **User Story #3**

**As an attendee I would like to see some solutions to these issues**

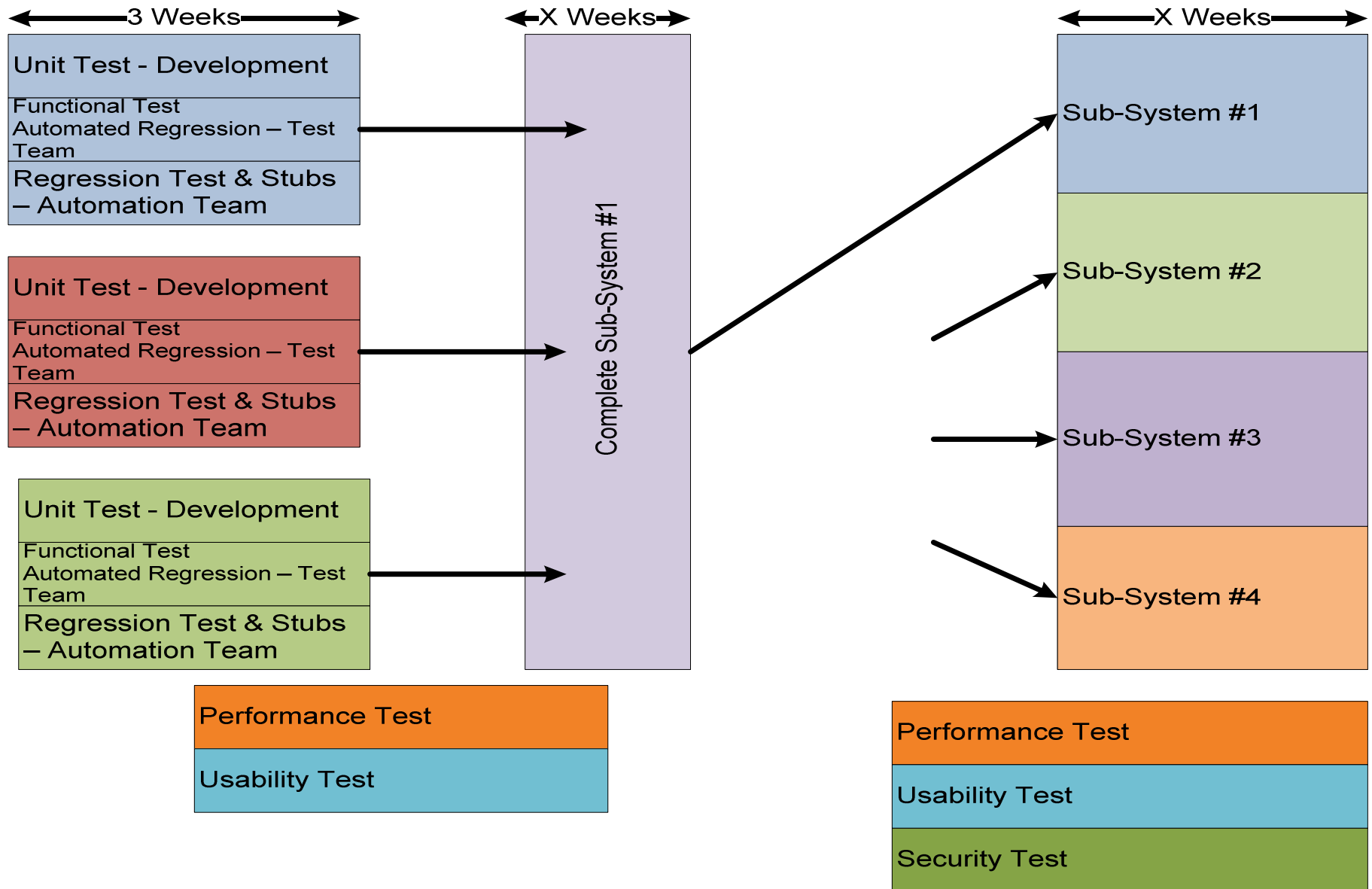
**How will we test this?**

# Time, Communication, Test Levels

- Test Strategy and Plan (Release)
  - > Plan as you go versus full upfront
    - Revise Plan at each iteration
    - Appropriate testing at each iteration
  - > Use to support communication
  - > Adaptive based on feedback
- Test Strategy and Plan (Iteration)
  - > Based on short term upcoming work
- Informal planning tools
  - > whiteboards, stickies, wikis, checklists, etc. (versus Gantt charts, comprehensive documents)
- Risk List



# Different Test Levels



# Time, Skill, Communication

## Exploratory Testing

“... agile programs are more subject to unintended consequences of choices simply because choices happen so much faster. This is where exploratory testing saves the day. Because the program always runs, it is always ready to be explored.”

Ward Cunningham on *Why should agile teams do exploratory testing?*:

# Skill, Culture

- Role of the Tester
  - > Full Integration of testers into development team
    - Co-locate**
    - Consult on good testing**
    - Early involvement – a very positive experience!**
  - > Working with little documentation
    - Communicate, Communicate, Communicate**
  - > Minimising test documentation
    - Lightweight, minimise unnecessary detail and duplication, multipurpose, checklists, risk lists, Wikis,**
  - > Tester skillset
    - Develop programming skills**

# Skill, Culture

- Training
  - > Agile Training
  - > Technical – Scripting, Coding Languages, Automation
  - > Reviews
- Plan for Process Change
  - > Allow time for change in process
  - > Adapt

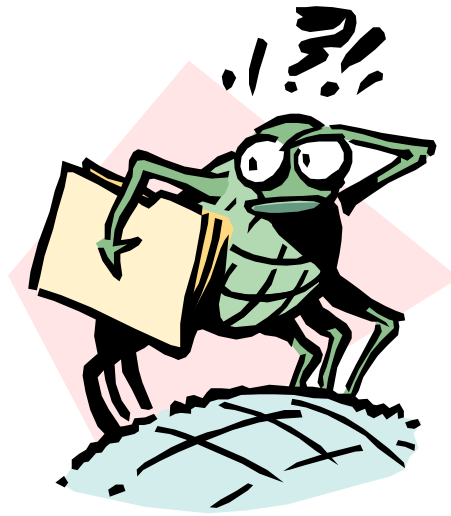
# Focus, Communication, Trust

- Daily Meetings
- Measure on working software
- Work as a team
- Product Reviews / Working Software
- Retrospectives



**User Story #3**  
**Done?**

# Questions



Contact Details:

[Ken.Brennock@sogeti.com](mailto:Ken.Brennock@sogeti.com)