TODAY'S TOPIC

KNIGHT ERRANT

Knight Errant Software Testing | Training |Project Consulting | Business Analysis

www.knighterrant.com.au

"The importance of testing in an AGILE development context"

Or

"Don't forget the testers"

PART 1

KNIGHT ERRANT

Knight Errant Software Testing | Training |Project Consulting | Business Analysis

www.knighterrant.com.au

The role of software testing in the software development lifecycle

The role of testing in the SDLC

The importance of testing

- Clients do not want a costly and stressful problem with software that doesn't work, isn't ready on time, does not meet specifications, impacts customers adversely, has bugs or is too complicated to use!

- The best results for a project occur when testing is planned and related activities commenced at the start of a project.

- Testing is not just finding bugs. It is the systematic assessment of an application’s fitness for the purpose it has been designed for.

- Testing should be conducted throughout the lifecycle for maximum project benefit

The role of testing in the SDLC

Types of testing

Reference: ISTQB Glossary of testing terms

- **Unit Testing** – The testing of individual software components.
- **Functional/System Testing** - Testing based on an analysis of the specification of the functionality of a component or system.
- **User Acceptance Testing** - Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.
- **Integration Testing** – Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.
- **Performance Testing** - The process of testing to determine the performance of a software product.
- **Load Testing** - To evaluate the behaviour of a component or system with increasing load, e.g. numbers of parallel users and/or numbers of transactions
- **Last minute frantic testing** – As usually happens

The role of testing in the SDLC

ISTQB Testing Principle 3: Early Testing is vital

- Industry Research (such as Gartner, Sticky minds) has highlighted approximately 50% of defects can be traced to requirements.

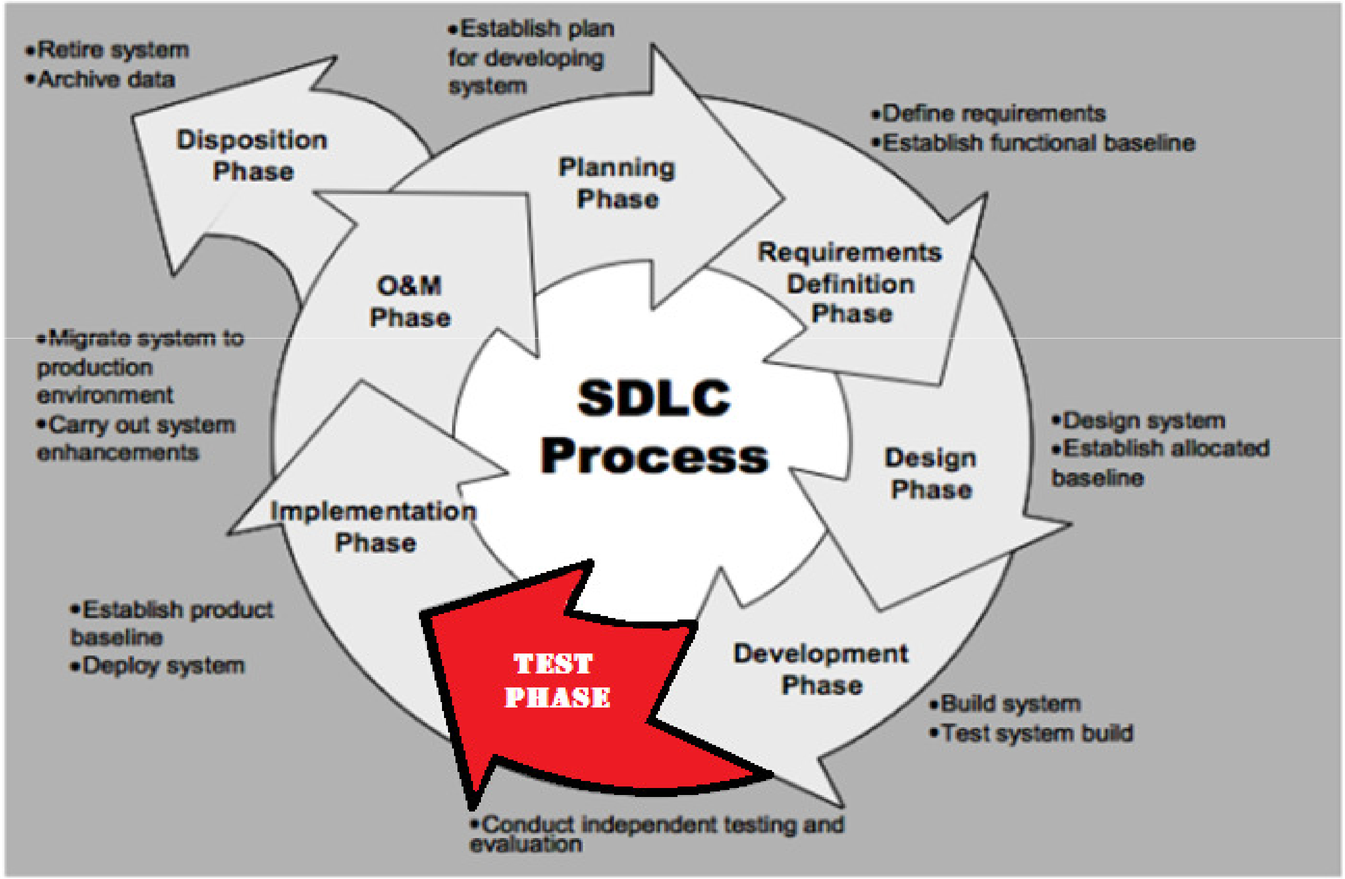
Stage of the SDLC	Cost to Fix Defects
Requirements	1x
Design	2x
Coding	5x
Testing	10x
Production	30x

Source: US National Institute of Standards & Technology

The earlier a defect is identified the cheaper it is to fix!

The role of testing in the SDLC

Where testing usually occurs


SDLC Process
Planning Phase
•Establish plan for developing system
Requirements Definition Phase
•Define requirements
•Establish functional baseline
Design Phase
•Design system
•Establish allocated baseline
Development Phase
•Build system
•Test system build
TEST PHASE
•Conduct independent testing and evaluation
Implementation Phase
•Establish product baseline
•Deploy system
O&M Phase
•Migrate system to production environment
•Carry out system enhancements
Disposition Phase
•Retire system
•Archive data

Testing must be integrated

- In the previous slide, testing is represented as a discreet activity late in the development cycle
- In reality, testing should be integrated to each stage of the lifecycle
  - Static testing of requirements
  - Dynamic testing of code
- This is equally valid regardless of the development methodology
  - WATERFALL benefits from static testing of detailed requirements prior to issuing to development
  - AGILE benefits from static testing of requirements and Dynamic testing of SPRINT deliverables against business expectations

The role of testing in the SDLC

Summary

- Testing is a major part of the SDLC but it has long been recognised that it doesn’t always have the appropriate visibility and emphasis in project planning
- Early involvement of testing expertise leads to fewer defects
- Requirements can (and should) be tested
- You can’t leave testing until the end of the project and expect the best result
- Testing needs to be integrated into the process regardless of the development approach

WATERFALL v AGILE

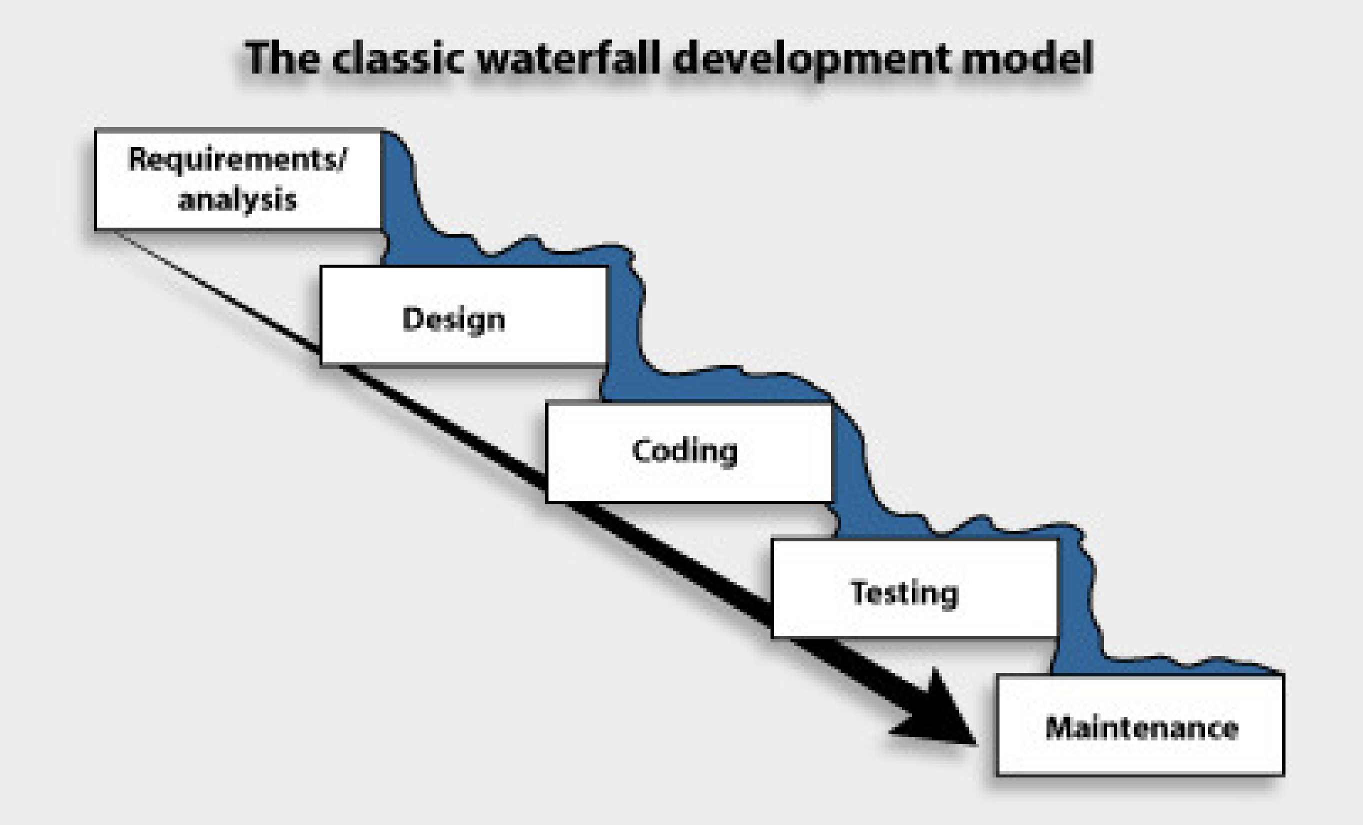
What is WATERFALL

Follows a distinct set of steps through the life cycle of your project. The steps are usually similar to this:

- Requirements Analysis
- Design
- Implementation (or Build)
- Testing
- Installation
- Maintenance

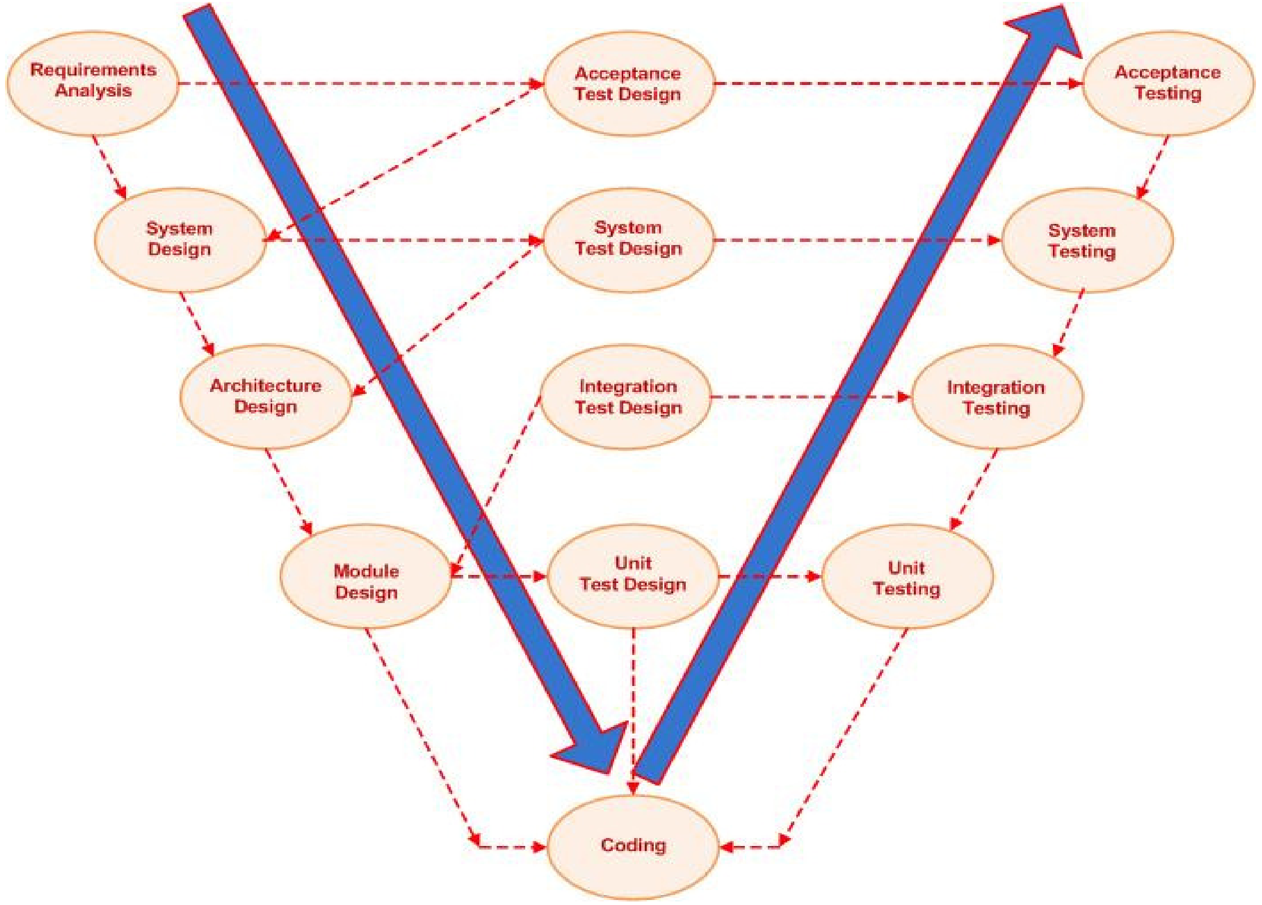
Testing in WATERFALL

Traditional approach – distinct testing phase



Testing in WATERFALL

Adapting WATERFALL to allow testing at each stage



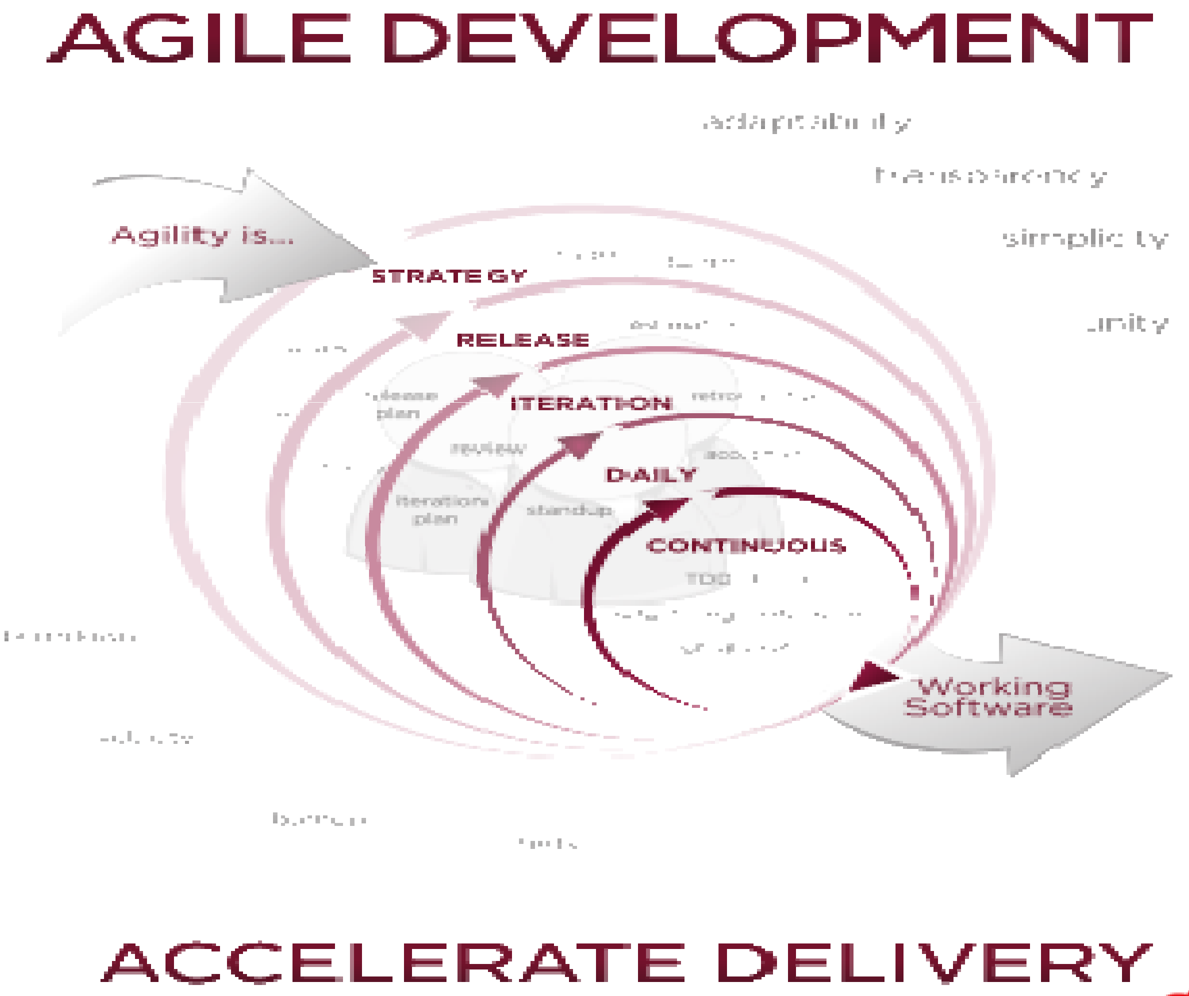
What is WATERFALL

Good model for well defined deliverables, or using offsite development resources.

But can be difficult when dealing with internal clients with “flexible” requirements.

Pros and cons of the Waterfall method.	
Pros	**Cons**
Detailed documentation.	Slow start.
Agreed and signed off requirements.	Fixed requirements difficult to change.
Can be delivered using developers with a lower skill set, or working remotely	No customer visibility of software until the development has been completed.
Reduced number of defects through thorough design planning.	Lack of flexibility making it difficult to change direction.
Defined start and end point for each phase, allowing progress to be easily measured.	Customers often unclear about their requirements initially.

What is AGILE



What is AGILE

From the AGILE manifesto:

- **Individuals and Interactions** – in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- **Working software** – working software will be more useful and welcome than just presenting documents to clients in meetings.
- **Customer collaboration** – requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important.
- **Responding to change** – agile development is focused on quick responses to change and continuous development.

So what does this mean for testing?

PART 2

KNIGHT ERRANT

Knight Errant Software Testing | Training |Project Consulting | Business Analysis

www.knighterrant.com.au

What AGILE means for Testing

Question:

When and how are requirements statically tested in an agile development environment

Individuals and Interactions

- Approach requires a lesser degree of separation of the test team than other methodologies
- High level of interaction between testers, BA, developers and customers
  - Regular consultation as opposed to formal review and sign off
- Focus on manual confirmation testing as opposed to rigidly scripted functional tests
  - Focus is on regular feedback and review

Question:

What expertise does a test analyst bring to an agile development team

Working software over comprehensive documentation

- Tests are often designed directly from the software
  - Reading unit tests
  - Exploring streams and analysing output
- Test team are responsible for not only assessing the systems, but selecting the best method for performing the assessment
- Discussions with other members of the team replace some documentation
- Test team is not the only ones responsible for testing
  - Developers for unit tests
  - Customer for acceptance tests

Question:

Do testers play a role in the fast turnaround of issues / problem resolution

Customer collaboration over contract negotiation

- Testers may:
  - Take on the role of the customer when designing and executing acceptance tests
  - Be a customer advocate when the customer is unavailable during decision points
  - Take on the role of a “Bad” customer to extend acceptable testing to the limit of the systems
- By collaborating with the customer, and acting as proxy as appropriate, the project benefits by:
  - Defects are discovered and raised regardless of their relationship to fixed requirements
  - Acceptance of the product as a whole is measured throughout the lifecycle, as opposed to at fixed points toward the end of the project

Question:

How different does a test plan look in an agile development context

Value responding to change over the following of a plan

- Testers need to make good use of unit tests for regression/stability so they can keep pace with testing iteration development
  - This helps give testers some ownership of overall product performance
- A risk of AGILE – some people expect testers to have detailed plan for testing of overall product
  - This is not possible due to the nature of the iterative development
  - It is possible to have detailed plans at iteration level however this is contrary to the principle of “working software over comprehensive documentation”
  - Any iteration test plans need to consider overall software performance not just current iteration where possible
    - Or detail the need to look at separate test cycle for product delivery (typically after 6 iterations)

Question:

What does a test analyst do within a SPRINT

TESTING WITHIN A SPRINT

- Testing activities are done in parallel with the design and development work.
- The testers are involved, both passively and actively, in each step to gain an understanding of the customer’s requirements and thus identify what needs to be tested.
- The planned tests should include, as a bare minimum, the requirements being implemented for that sprint.
- The two most popular methods for testing within a sprint are:
    - Execute tests upon completion of the designated sprint scope items. That is, schedule time within the sprint to allow test execution to occur.
    - Execute tests as code is completed. That is, tests are executed on the software as scope items are completed.
- Regression can be included within the test activities for a sprint but this is usually only feasible on a new product that has had less than four previous sprints. A regression test can be scheduled after several sprints to ensure the software continues to function as expected.

Question:

What can go wrong or cause delays from a test analysts point of view

Testing within an iteration

In order to confirm a software meets customer needs, and to provide timely feedback to developers on unexpected behaviour, software needs to be in a workable state

This objective cannot properly be achieved if software is not available until the end of an iteration. Possible contributors to this include:

- The iteration scope is too large causing development to be squeezed for time
- Decisions are not being made as a team and shared ownership is lost resulting in delays when designing the implementation

Question:

What happens if testing is staggered to occur in a subsequent SPRINT

Testing within an iteration

A natural temptation is to “move” testing to the next iteration. This goes against the AGILE methodology because:

- It creates separation between the test team and the rest of the project
- Reduces communication
- Delays feedback on iteration code. If problems are found the developers have already moved on to new functionality
- Can create a focus on the volume of code written as opposed to quality
- Can create idle time for both developers and testers if they are not working in parallel

PART 3

KNIGHT ERRANT

Knight Errant Software Testing | Training |Project Consulting | Business Analysis

www.knighterrant.com.au

The role of testers in AGILE

Role of an AGILE Test Manager

The role of the test manager is largely to take responsibility for:

- Non-Unit Testing tools
- Test environments
- Test data

Choices such as testing “bleeding edge” vs. “system as a whole” become more important due to the shortened timescale of iterations.

Reliance on unit test for coverage and an understanding of the impact of new functionality on existing systems is essential.

Factors to consider when planning an iteration test cycle are:

- Finding different problems
- Improving diagnosis of existing problems
- Setting criteria so testers know when to move to another area of the system when problems are found.

The role of an AGILE tester

A tester can expect to:

- Communicate more with developers and customers
- Gain a greater understanding of the technologies and code
- Are no longer the final-say or “last defender of quality”
- Share more of the risk
- Gain a greater sense of ownership

The main functions a tester will perform are:

- Enhance the confirmation test at both a Unit and Acceptance level
- Identify, diagnose and exploit unexpected system behaviour to ensure the product is fit for purpose

PART 4

KNIGHT ERRANT

Knight Errant Software Testing | Training |Project Consulting | Business Analysis
www.knighterrant.com.au

The Key Points

Waterfall v AGILE - Summary

- Accept the chosen approach may often be a Hybrid of methodologies to suit business
    - Need to adapt to client needs as opposed to a purest approach to methodology
- Both have their pro’s and cons
    - The appropriate methodology depends on the client and product
    - Both are valid
- This presentation focusses on how the role of testing changes for each methodology
- Both approaches have merit, we do not advocate one over the other
- Testing should be factored into plan up front, with everyone clear on the approach.
- Neither approach requires more or less testing, just a different approach at the various stages
- Testing can be tailored to suit (more documentation, less documentation etc)

Key Differences of AGILE testing

- Difference in specifications
  - AGILE specifications less formal, and focussed on small parcels of development
- Focus on communications over documentation
- Fast turnaround and correction of issues
  - Importance of testing individual sprints
- Removal of mgt/control structure (safety net)– focus back on individual roles and deliverables
  - Need for collaboration and shared responsibility
- Feedback loop
  - Testers feeding in to development iterations
- Defect Mgt
  - Focus on changing software to suit needs instead of meeting pre-determined requirements

I love it when a Test Plan comes together

K E KNIGHT ERRANT

Knight Errant Software Testing | Training |Project Consulting | Business Analysis

www.knighterrant.com.au

Questions?