

Adopting Agile Practices

Ian Charlton

Managing Consultant

ReleasePoint Software Testing Solutions

ANZTB SIGIST (Perth) – 30 November 2010

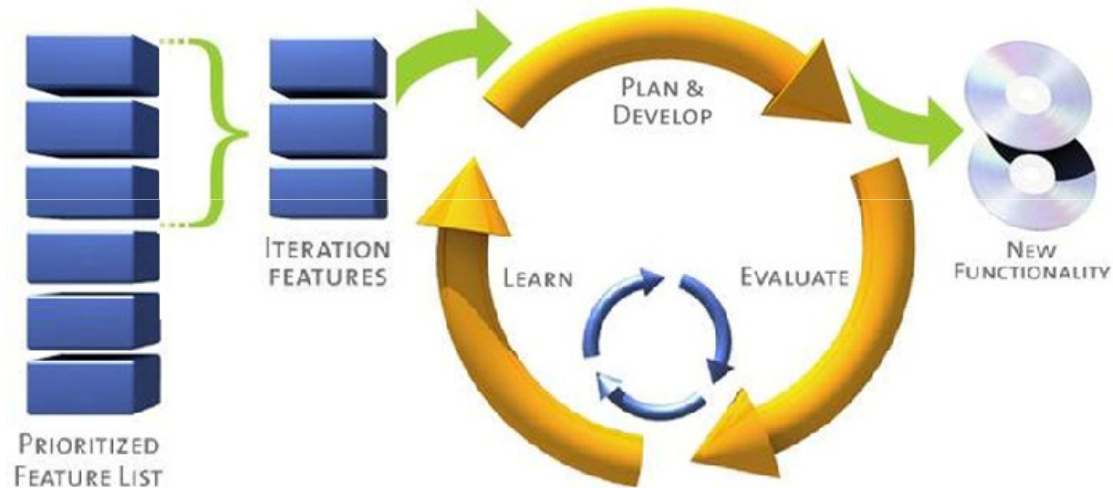


Tonight's Agenda

- What is Agile?
- Why is Agile Important to Testers?
- Scrum – An Agile Methodology
- Baseline - Our Typical Test Management Process
- Adopting Agile Practices – Our Dilemmas
- Adopting Agile Practices – Our Solution?
- Questions?
- Further Information

What is Agile?

- Agile is a methodology used primarily in software development initiatives.
- Based on 12 principles, it enables teams to respond to the unpredictability of building software through incremental, iterative work cadences, known as sprints.



- Scrum, Adaptive Software Development (ASD), and Lean Development focus more extensively on project management and collaboration practices.
- Extreme Programming (XP), Agile Modelling (AM), and Feature-driven Development (FDD) concentrate heavily on software implementation practices.

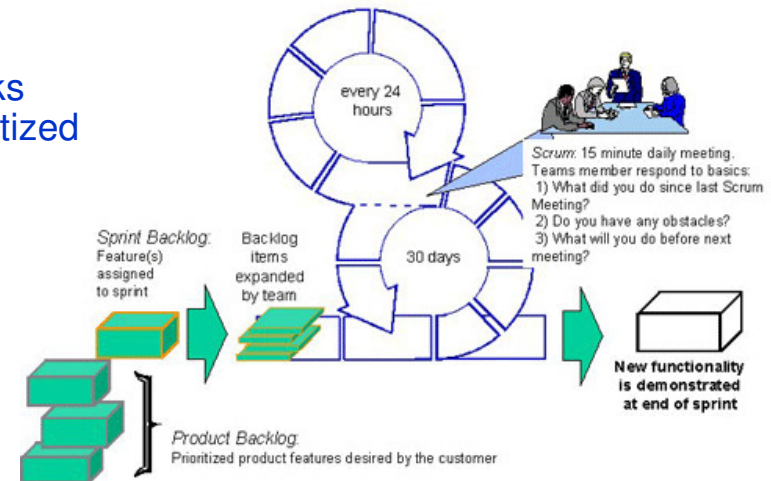
Agile – The Principles

Manifesto for Agile Software Development:

1. Satisfy customer through early and continuous increments;
2. Deploy first increment within couple of weeks and the whole software within couple of months;
3. Customer and agile teams must work jointly daily throughout the project;
4. Agile team and customer must have face-to-face meetings;
5. Welcome requirements even in late phases of the system development;
6. Trust and respect must be maintained amongst Agile team members;
7. Velocity of the project must be measured after delivery of each increment;
8. Emphasis should be on good design to increase agility;
9. Best architecture and design always come out from self-organisation;
10. Adjust and tune according to the situation;
11. Whole development process must follow keep it simple (KIS) principle; and
12. Agile project needs consistent work until completion.

Scrum – An Agile Methodology

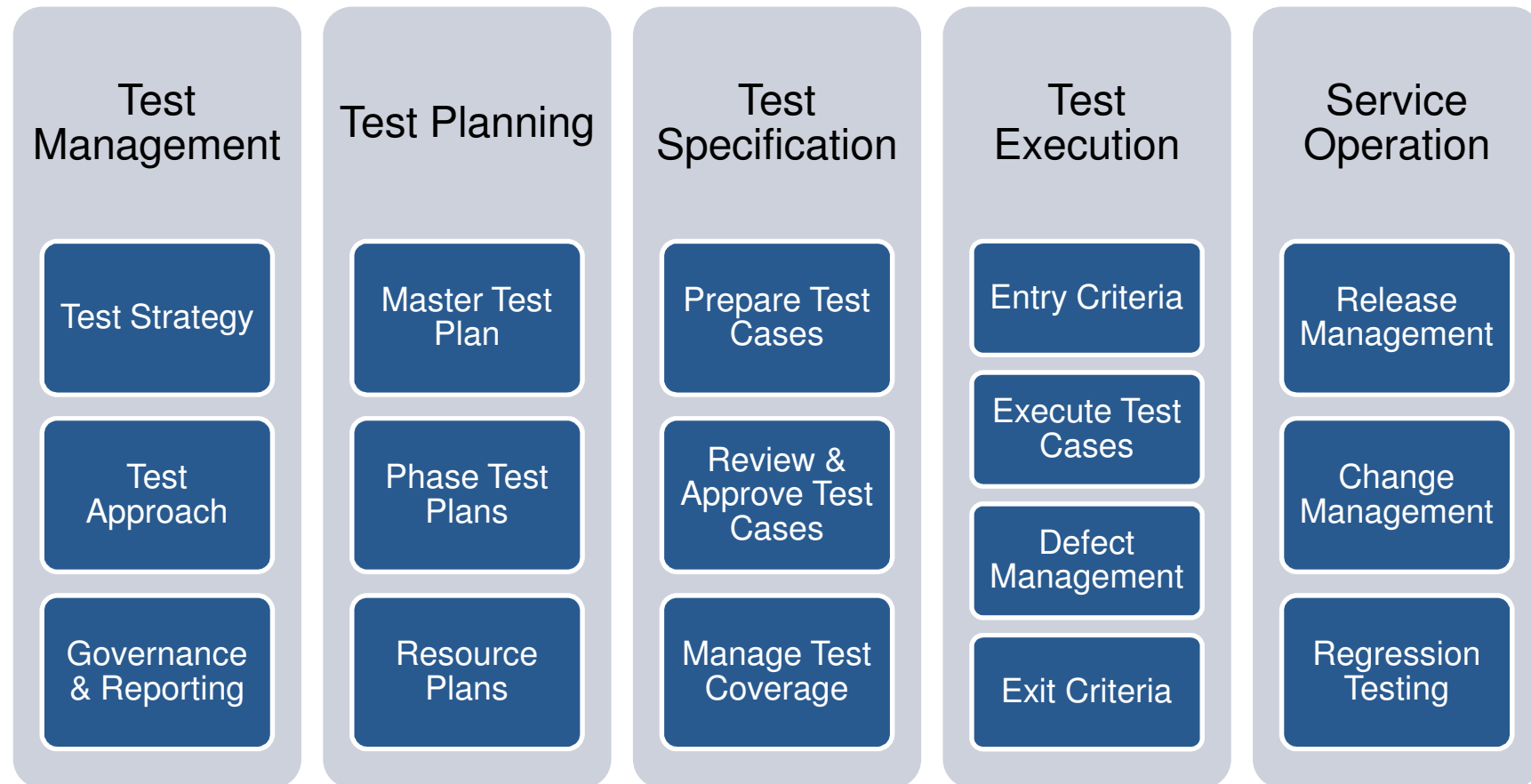
- History:
 - Scrum was applied in 1990s by Ken Schwaber and Mike Beedle. It is an agile, iterative, incremental developing method which assumes that changes and chaos exist through entire life-circle of the project and attempt to solve these problems.
- Objective:
 - Scrum naturally focuses an entire organization on building successful products.
 - Scrum is designed to add energy, focus, clarity and transparency to project teams development software systems allowing teams to operate in close proximity.
- Scrum Process:
 - Work is structured in cycles of work called sprints, iterations of work that are typically two to four weeks in duration. During a sprint, teams pull from a prioritized list of customer requirements, called user stories, so features developed first are of highest value.
 - A product is delivered at the end of each sprint.
 - Without major changes - often within thirty days – teams are building useful and demonstrable product functionality.



Why is Agile Important to Testers?

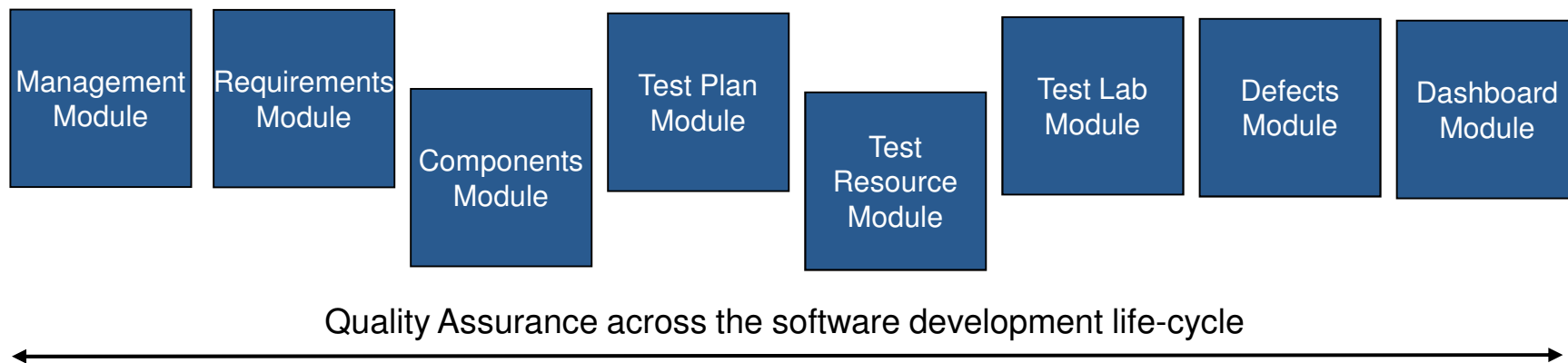
- Traditional development methods are (frequently) not delivering business advantages;
 - Projects are late, or even cancelled;
 - Functionality delivered is not what is really required; and
 - A lot of effort is spent delivering features that are not used.
- Consequently organisations are now using alternative development management strategies.
- Agile 'promises':
 - Reduce total cost of ownership;
 - Increase 'Quality'; and
 - Reduced time to market.
- **Agile has emerged as a leading software development methodology.**
 - If not 'The' leading methodology
- We cannot ignore this significant development in the way that software will be developed.

Our Typical Test Management Process



Our Tools Support This Process

Our leading test tools support this typical Test Management Process.



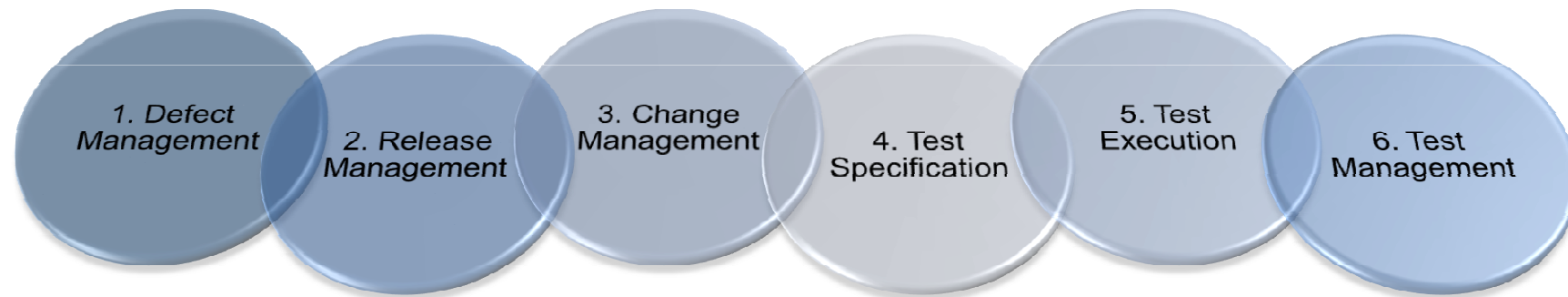
- They have been developed with structure and control in mind over many years.
- Significant financial investment in testing tools in terms of licence/maintenance costs, resources, training etc.
- Significant IPR investment in testware.

The Big Question is:

“Can they be successfully utilised for truly Agile projects?”

Adopting Agile Practices – Our Dilemmas

Looking at the our current 'typical' processes, we testers face a number of dilemmas in the following areas as we attempt to adopt Agile.



Dilemma 1 – Defect Management

- **Defect Management Dilemma**

- Defects require time to be fixed.
- Our current processes and tools expect defects to be investigated, fixed, retested and passed before the software is delivered.
- Our concepts of Severity and Priority don't make sense under Agile

Severity

1-Critical

2-High

3-Medium

4-Low

Priority

1-Urgent – Hot-Fix ASAP

2-Very High – Hot-Fix within 24 hours

3-High – Fix in next Release

4-Medium - Fix in next 2-3 Releases

5-Low – Fix when possible

- Agile assumes:
 - Delivery of working features at the end of the Sprint; and
 - Non-working features are excluded from the release.
- Therefore we need to be able to return a defect to the backlog, if it cannot be resolved within that Sprint.

Dilemma 2 – Release Management

- **Release Management Dilemma**
 - An Agile Release occurs at a pre-defined point in time (i.e. at the end of a sprint).
 - It only contains the features that work.
 - Features that don't work or were not even attempted are re-cycled to the backlog.
 - Our processes and tools are usually aligned to support a planned release to contain a defined set of features.
 - The concept that a feature will not be delivered if it is not working is not easily supported.
 - Adding and removing requirements to a release has an impact on the test cases that need to be executed and the corresponding defect management and retesting.
 - It takes time to use Risk-based approaches to assess requirement priorities, so it become difficult to constantly be reviewing the requirements.
 - Standard reporting and estimating are based on an approved 'baseline'.

Dilemma 3 – Change Management

- **Change Management Dilemma**
 - Change happens constantly to projects and operational systems.
 - Our processes and tools expect change to follow a procedure with impact analysis, estimates, and approvals.
 - Agile embraces change daily which compresses the time that our processes expect to take days into minutes.
 - Our documentation cannot be maintained within that frequency.
 - Our tools are not expecting to support rapid changes to entities across the software development life-cycle.
 - Version control becomes increasingly important to identify the latest instance.

Dilemma 4 – Test Specification

- **Test Specification Dilemma**

- As we have matured our testing activities, we have often taken advantage of creating test components for business process testing.
- We would also develop efficient test cases that could verify multiple requirements.
- Agile almost demands that the tests are modular to single (or a few) requirements, otherwise it would become necessary to run multiple instances of the same test case to test requirements delivered across multiple Sprints.
- It becomes difficult to prioritise test case preparation if requirements are constantly being reprioritised in the backlog.

Dilemma 5 – Test Execution

- **Test Execution Dilemma**

- Test execution does not commence based on typical test entry criteria.
- Tests are executed as soon as the developer has released that feature into the test environment.
- Test environment management becomes difficult.
- Planning test resources for testing is fluid and reactive.
- Increased likelihood that tests will be executed by someone other than the author.
- Extremely difficult to utilise automated testing (during a Sprint) because the system under test is constantly changing.

Automated testing is suited to regression testing.

- Test execution ends at a point in time rather than as defined by 'typical' test exit criteria.

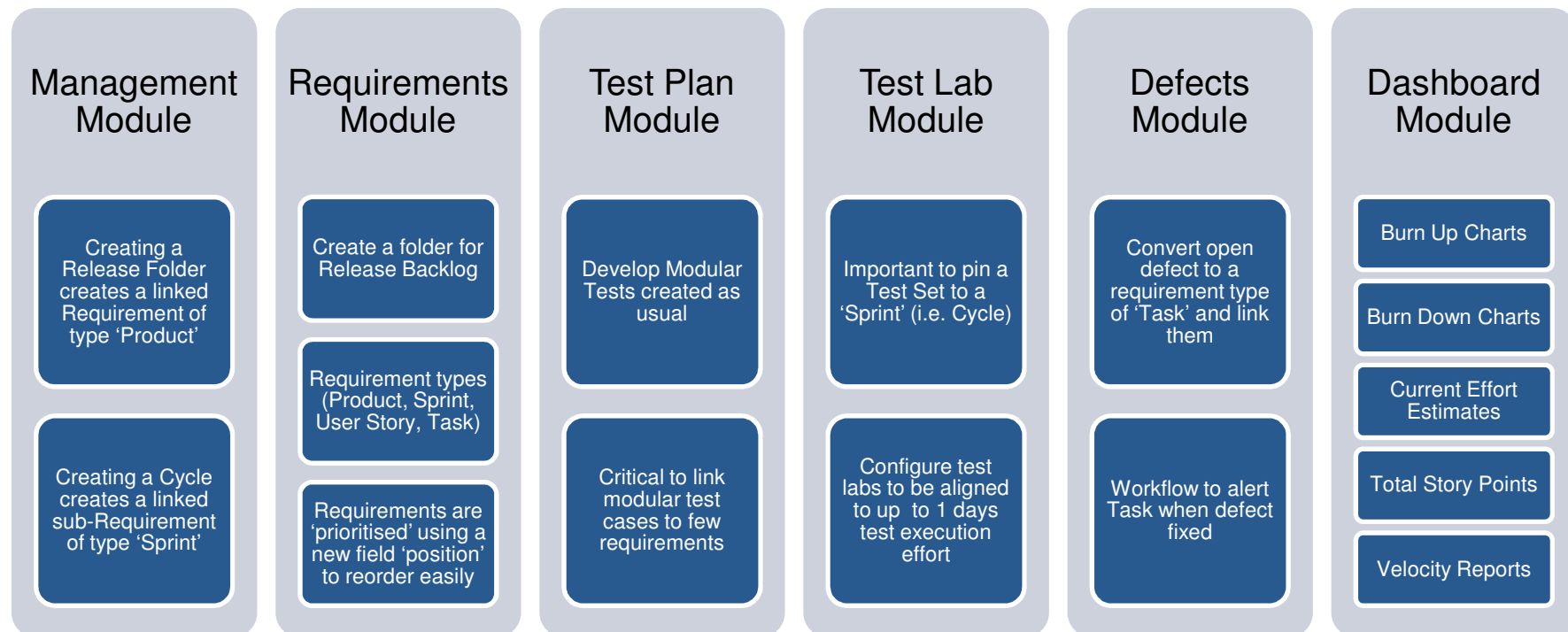
Dilemma 6 – Test Management

- **Test Management Dilemma**
 - Traditional Governance and Reporting cannot keep up with rapid change across the entire project scope.
 - Our typical metrics are not appropriate measures of a projects status or progress:
 - Requirements Coverage;
 - Tests completed;
 - Tests not run;
 - Open defects, etc.
 - Our usual entry and exit criteria are not applicable and the Test Manager is not in control of when testing commences or ends.
 - Our test tools require constant manipulation to keep the data current.
 - Need to reduce re-calculation effort including tasks, estimation, planning, and spent hours.

Adopting Agile Practices – Our Solution?

How can we adopt Agile practices within our existing processes and existing testing tools?

- Only way is to use the script editors to modify the workflow and configure our projects differently.
- Must adopt Agile terminology (e.g. User Roles, Requirement Types, etc.) to communicate.



Questions?



Further Information

- Further Details:
 - Chris Gray, WA Regional Manager
 - chris.gray@releasepoint.com.au
 - 0409 989 620
 - Ian Charlton, Managing Consultant
 - ian.charlton@releasepoint.com.au
 - 0425 417 668