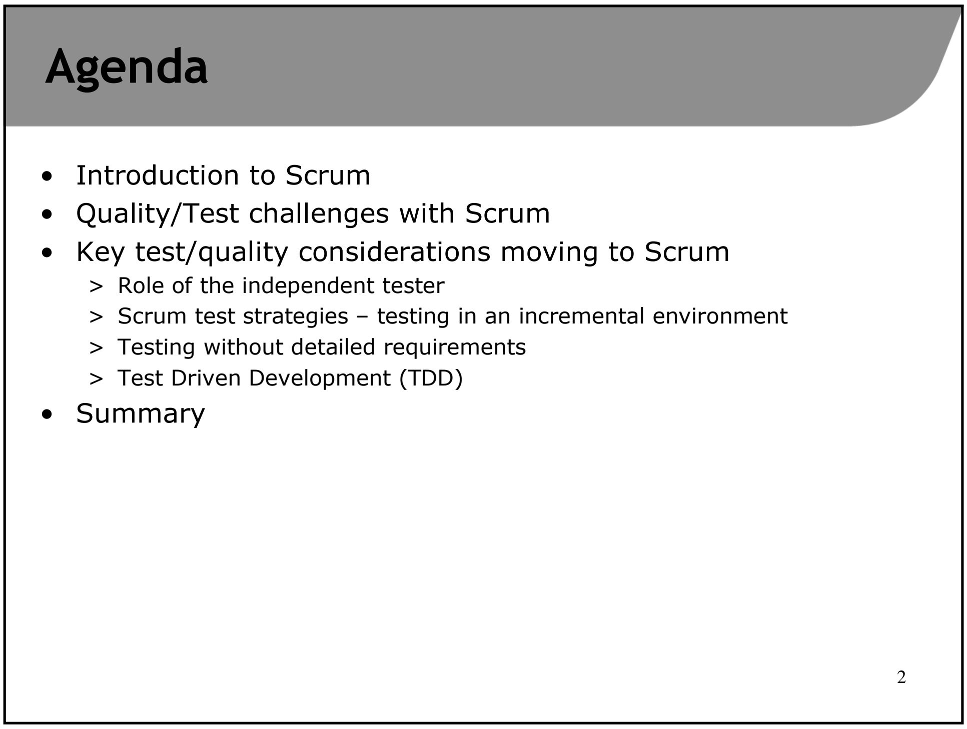
SOGETI

Scrum – a tester's perspective

Presentation to Agile Tour Belfast

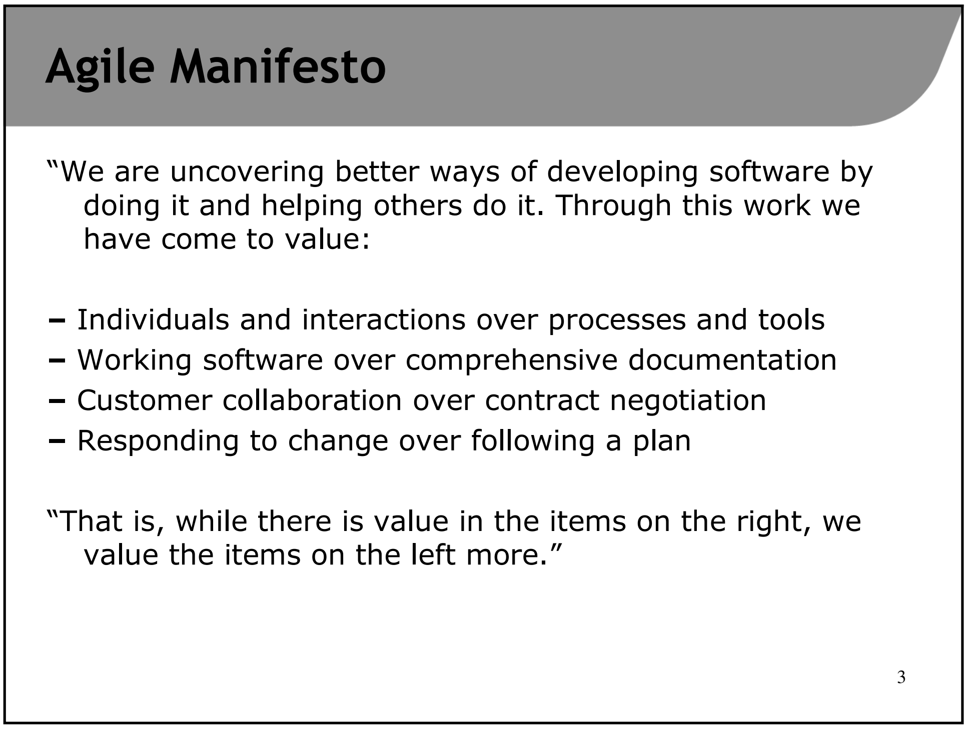
Fran O'Hara, Practice Manager, Sogeti Ireland

www.sogeti.ie
www.uk.sogeti.com



Agenda

- Introduction to Scrum
- Quality/Test challenges with Scrum
- Key test/quality considerations moving to Scrum
  - > Role of the independent tester
  - > Scrum test strategies – testing in an incremental environment
  - > Testing without detailed requirements
  - > Test Driven Development (TDD)
- Summary



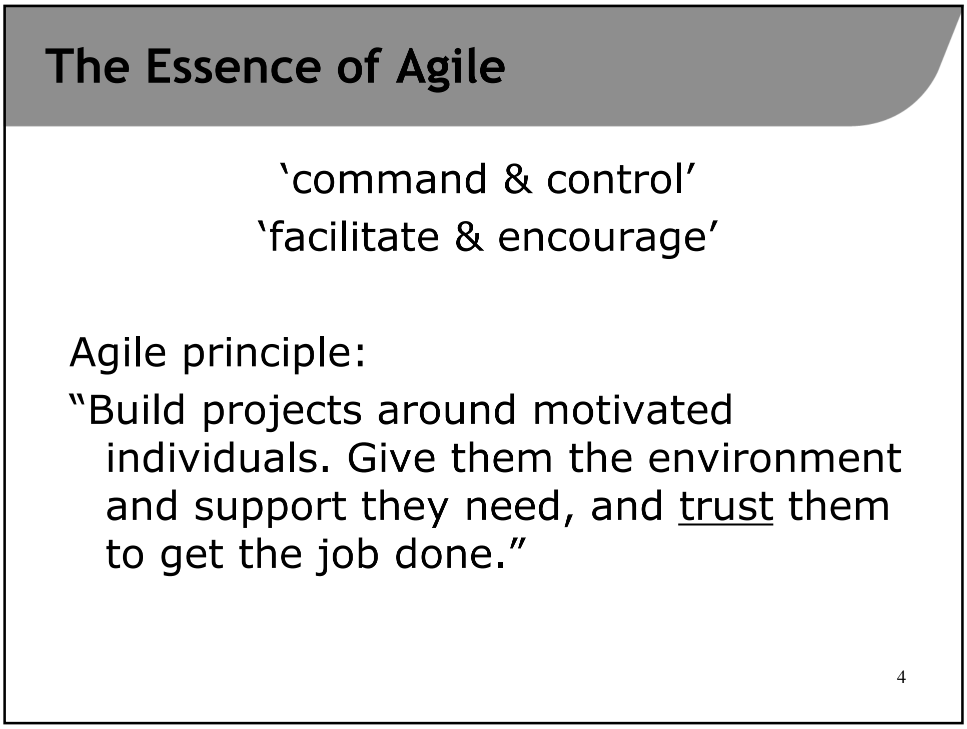
Agile Manifesto

“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.”

3



The Essence of Agile

‘command & control’

‘facilitate & encourage’

Agile principle:

“Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.”

4

Six principles of Agile Development

1: Customer lists known requirements (to a high level), then prioritises them.

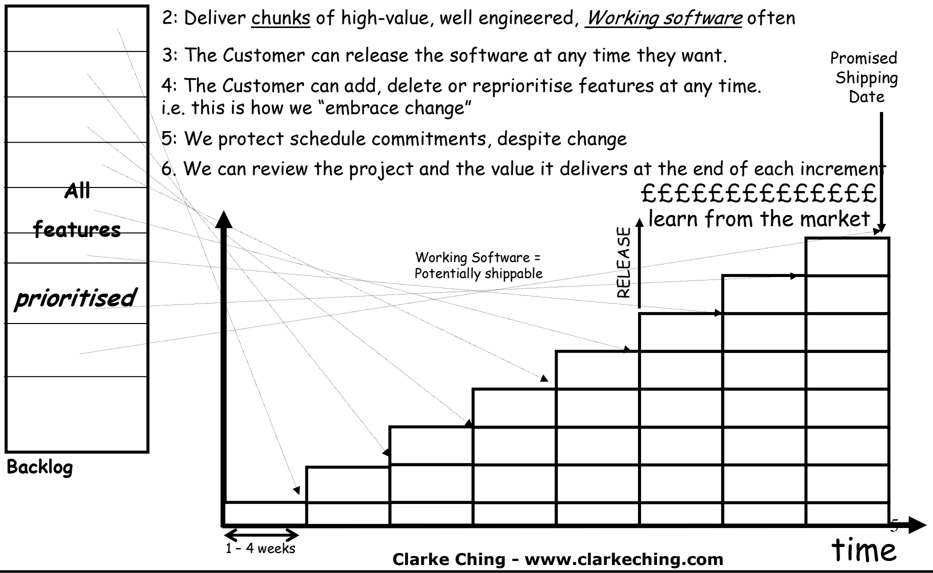
2: Deliver <u>chunks</u> of high-value, well engineered, *<u>Working software</u>* often

3: The Customer can release the software at any time they want.

4: The Customer can add, delete or reprioritise features at any time. i.e. this is how we "embrace change"

5: We protect schedule commitments, despite change

6. We can review the project and the value it delivers at the end of each increment



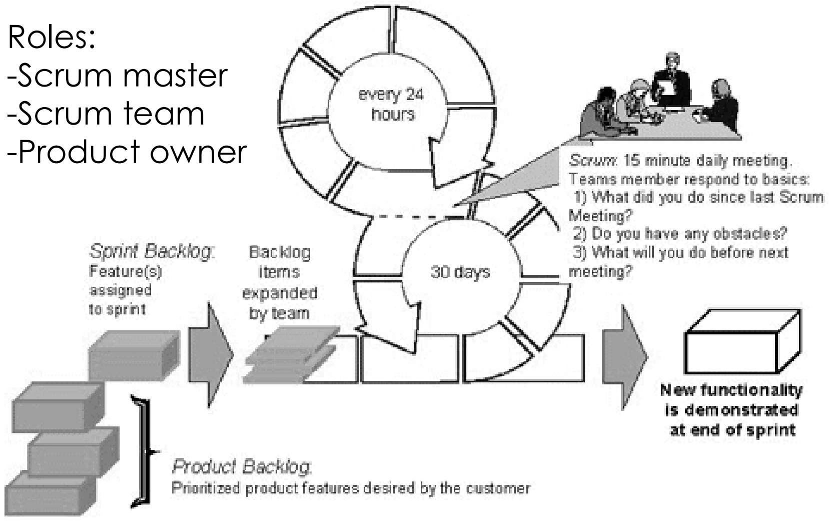
Clarke Ching - www.clarkeching.com

Scrum in 100 words

- Scrum is an agile process framework that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

Scrum

Roles:
-Scrum master
-Scrum team
-Product owner


every 24 hours
Scrum: 15 minute daily meeting.
Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog: Feature(s) assigned to sprint
Backlog items expanded by team
30 days
New functionality is demonstrated at end of sprint
Product Backlog: Prioritized product features desired by the customer

See www.controlchaos.com

Scrum framework

Roles

•Product owner
•ScrumMaster
•Team

Ceremonies

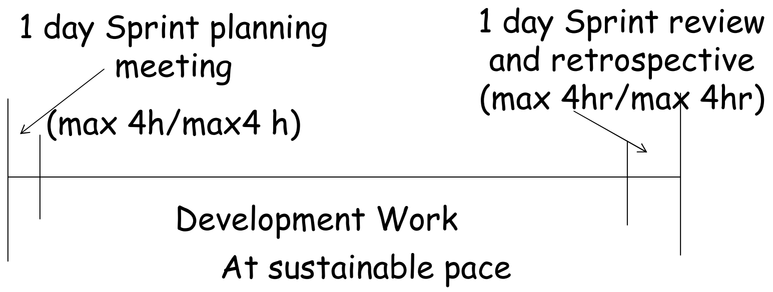
•Sprint planning
•Sprint review
•Sprint retrospective
•Daily scrum meeting

Artifacts

•Product backlog
•Sprint backlog
•Burndown charts

This and a number of other Scrum slides from Martine Devos: mmdevos@acm.org

Sprint/Iteration



1 day Sprint planning meeting

(max 4h/max4 h)

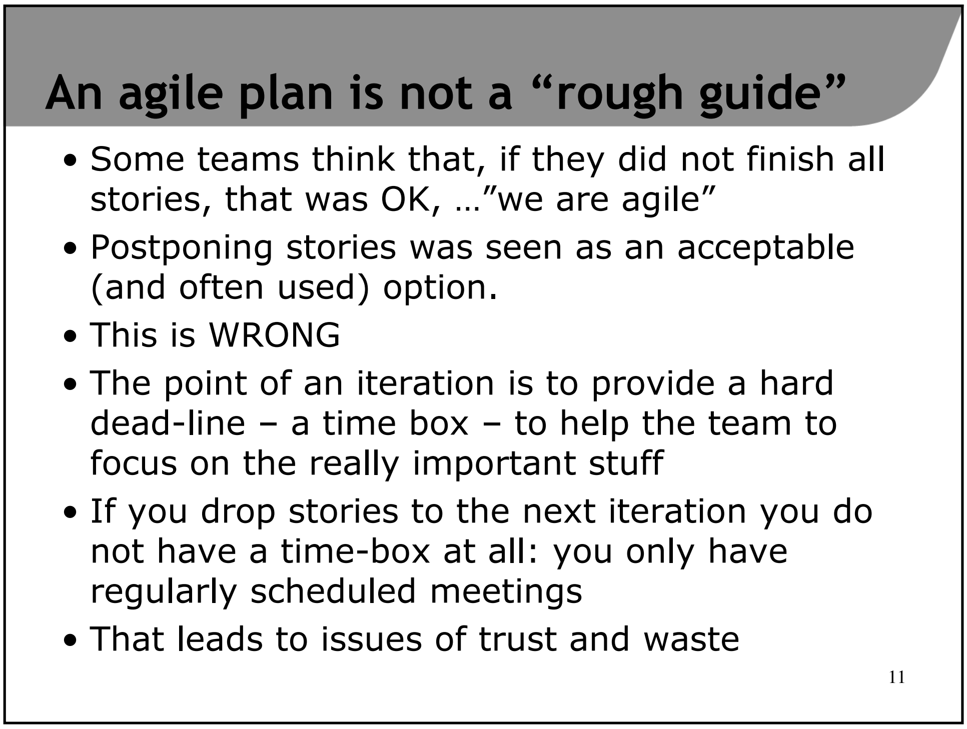
1 day Sprint review and retrospective (max 4hr/max 4hr)

Development Work

At sustainable pace

The Three Levels of Planning

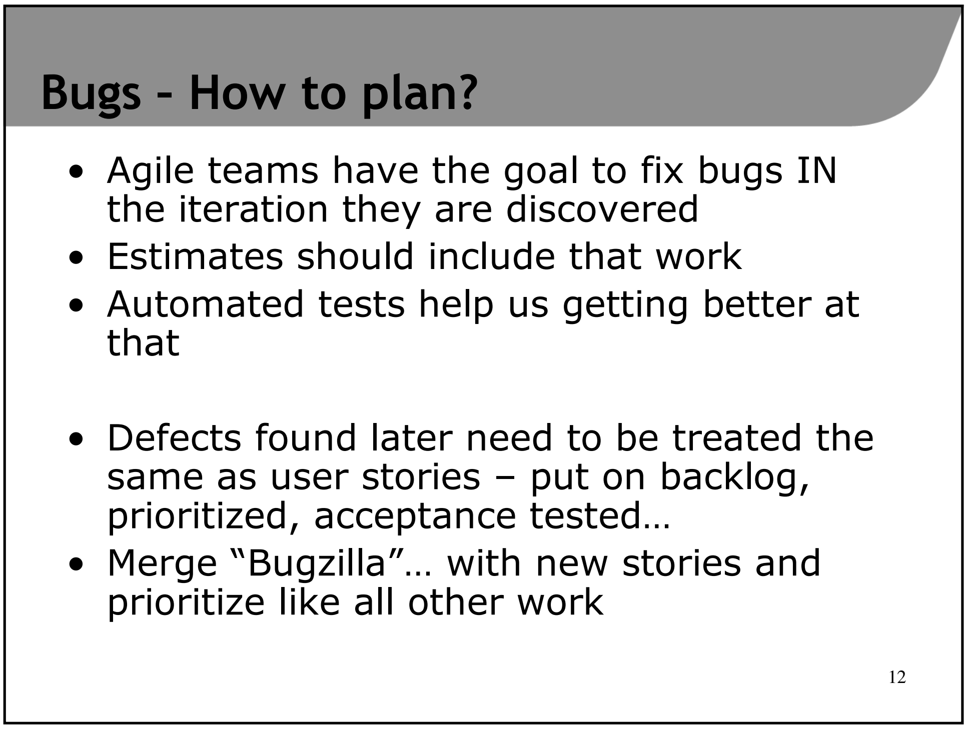
- Release plan
  - > Coarse-grained; forecast
  - > Balances date, scope and budget
  - > States anticipated order of story delivery
- Look-ahead plan
  - > Medium-grained; forecast
  - > Anticipates dependencies between team and levels the workload across all teams for the next 2-3 sprints
- Sprint plan
  - > Fine-grained; definite commitment
  - > Organises delivery of sprint goal
  - > States tasks and their duration in hours



An agile plan is not a “rough guide”

- Some teams think that, if they did not finish all stories, that was OK, …”we are agile”
- Postponing stories was seen as an acceptable (and often used) option.
- This is WRONG
- The point of an iteration is to provide a hard dead-line – a time box – to help the team to focus on the really important stuff
- If you drop stories to the next iteration you do not have a time-box at all: you only have regularly scheduled meetings
- That leads to issues of trust and waste

11



Bugs - How to plan?

- Agile teams have the goal to fix bugs IN the iteration they are discovered
- Estimates should include that work
- Automated tests help us getting better at that

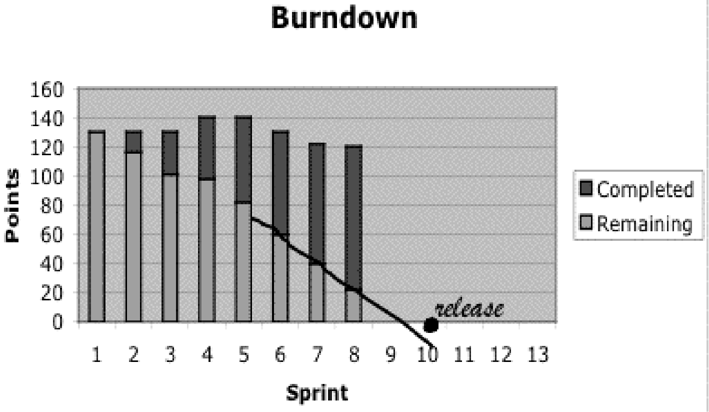
- Defects found later need to be treated the same as user stories – put on backlog, prioritized, acceptance tested…
- Merge “Bugzilla”… with new stories and prioritize like all other work

12

Done

- What does “Done” mean for the project?
  - > Spec written
  - > Code checked in
  - > Builds
  - > Unit tests complete successfully
  - > 80% code coverage on unit tests
  - > Accepted by testers
  - > Within acceptable defect levels
  - > Non functionally tested (performance, security…?)
  - > Etc.
  - > Accepted by product owner
- If stories not ‘done’ by end of Sprint then they are put back on backlog for the next Sprint planning
- Can there be more than one level of ‘Done’?

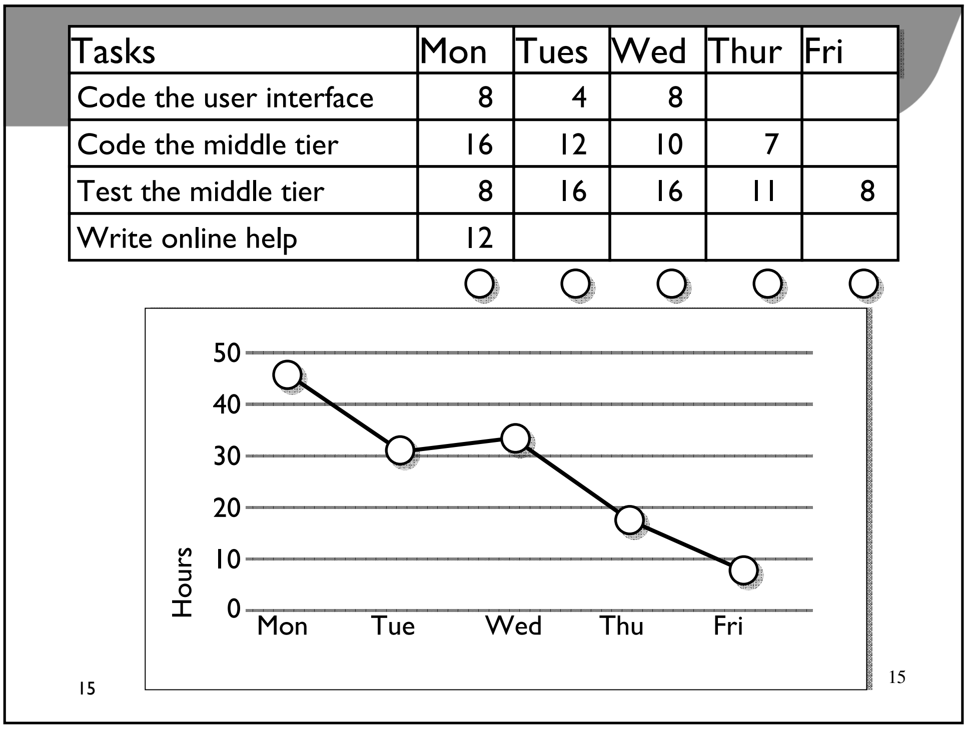
SCRUM tools & techniques - example



Sample burn-down chart – product, release, sprint backlogs used

Product Backlog typically a combination of

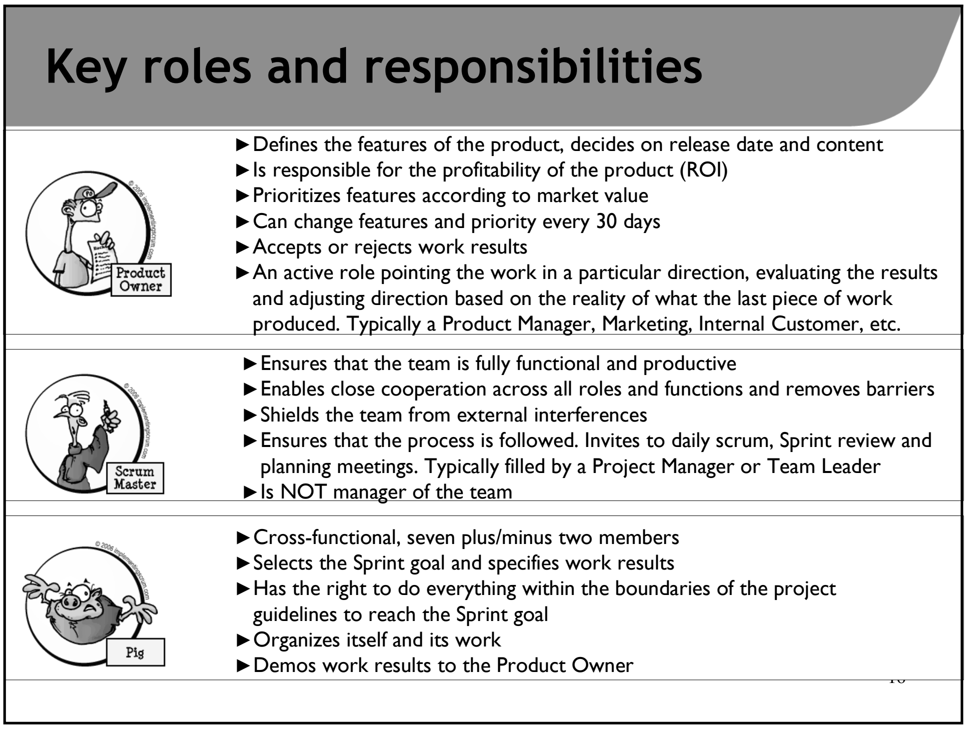
- story-based work (“let user search and replace”)
- task-based work (“improve exception handling”)
- bugs



Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				

15

15



Key roles and responsibilities

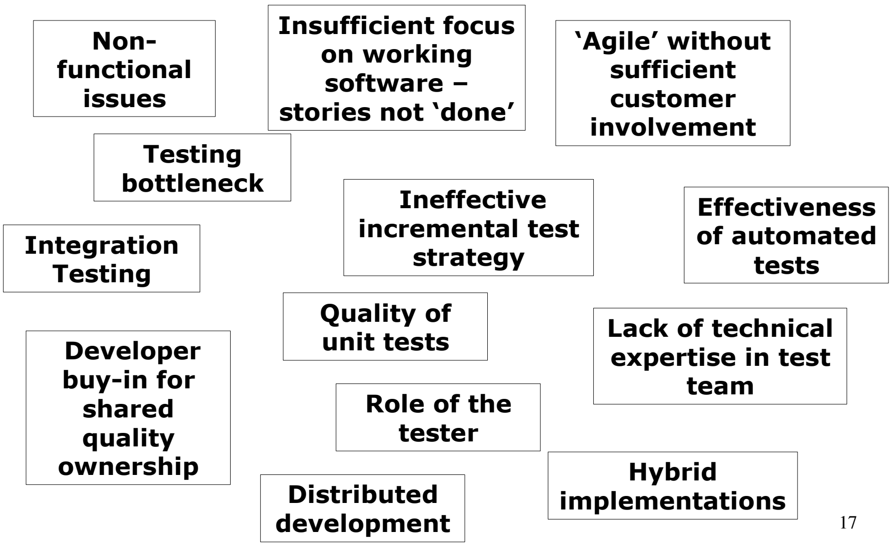
Product Owner

- ►Defines the features of the product, decides on release date and content
- ►Is responsible for the profitability of the product (ROI)
- ►Prioritizes features according to market value
- ►Can change features and priority every 30 days
- ►Accepts or rejects work results
- ►An active role pointing the work in a particular direction, evaluating the results and adjusting direction based on the reality of what the last piece of work produced. Typically a Product Manager, Marketing, Internal Customer, etc.

- ►Ensures that the team is fully functional and productive
- ►Enables close cooperation across all roles and functions and removes barriers
- ►Shields the team from external interferences
- ►Ensures that the process is followed. Invites to daily scrum, Sprint review and planning meetings. Typically filled by a Project Manager or Team Leader
- ►Is NOT manager of the team

- ►Cross-functional, seven plus/minus two members
- ►Selects the Sprint goal and specifies work results
- ►Has the right to do everything within the boundaries of the project guidelines to reach the Sprint goal
- ►Organizes itself and its work
- ►Demos work results to the Product Owner

Typical quality related challenges


Non-functional issues
Insufficient focus on working software – stories not 'done'
'Agile' without sufficient customer involvement
Testing bottleneck
Ineffective incremental test strategy
Effectiveness of automated tests
Integration Testing
Quality of unit tests
Lack of technical expertise in test team
Developer buy-in for shared quality ownership
Role of the tester
Hybrid implementations
Distributed development

Role of tester

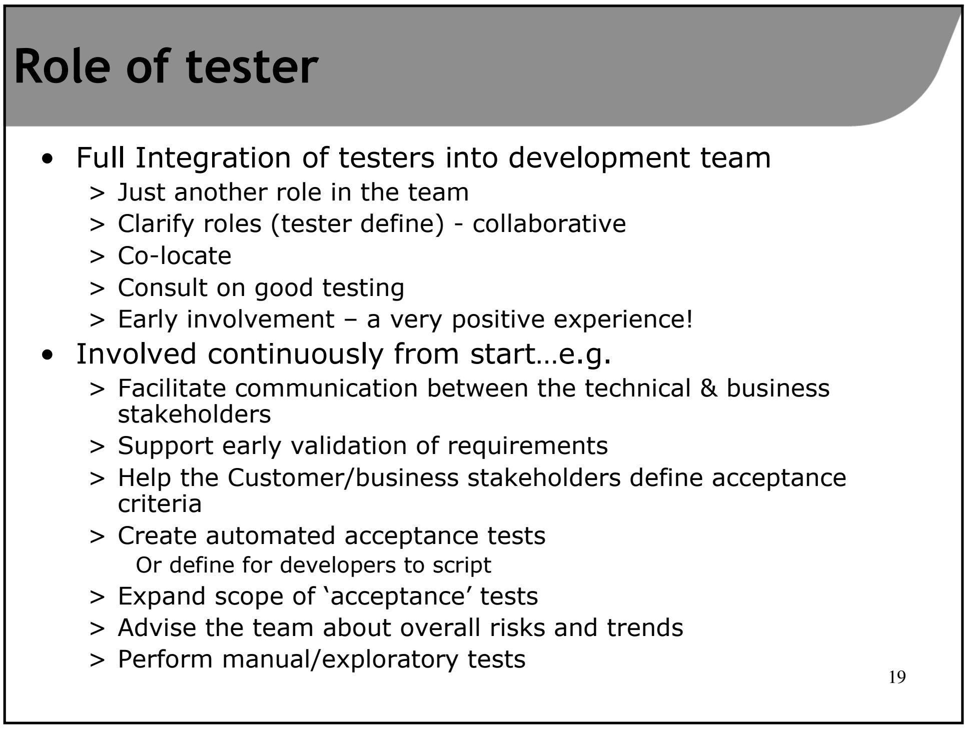
'The nature of the tester's role changes in iterative projects. We are no longer the high-profile victims, we are no longer the lonely advocates of quality, we are merely (!) competent service providers, collaborating with a group that wants to achieve high quality. '

Cem Kaner

Dedicated testers bring two benefits:

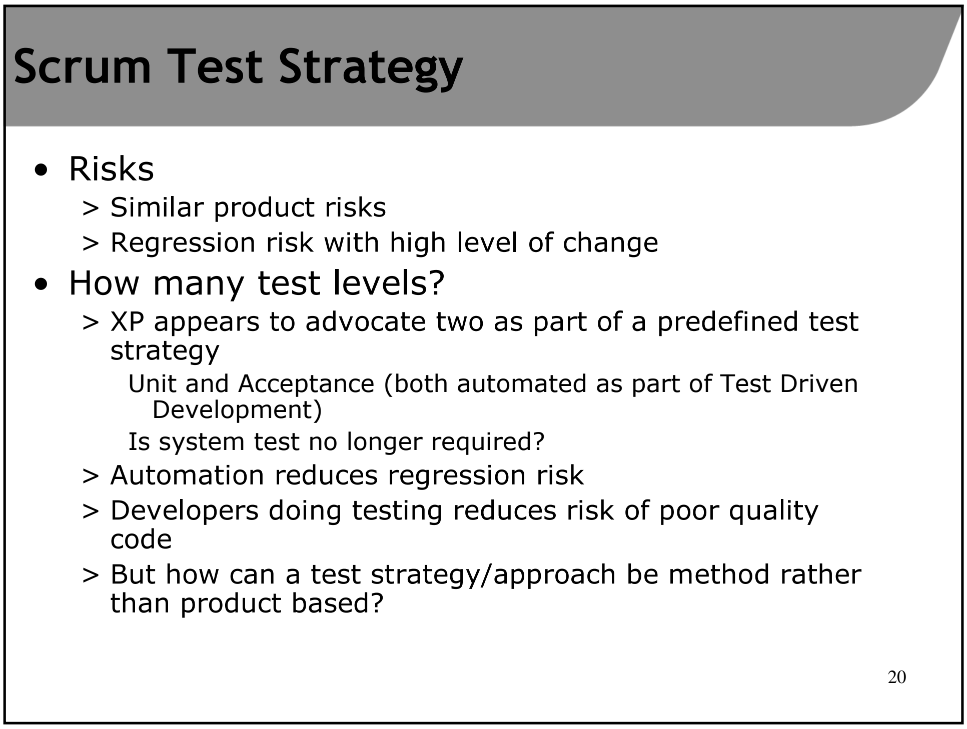
- Focus on customer usage over technical implementation
- Focus on uncovering flaws over confirming completeness

(Bret Pettichord)



Role of tester

- Full Integration of testers into development team
  - > Just another role in the team
  - > Clarify roles (tester define) - collaborative
  - > Co-locate
  - > Consult on good testing
  - > Early involvement – a very positive experience!
- Involved continuously from start…e.g.
  - > Facilitate communication between the technical & business stakeholders
  - > Support early validation of requirements
  - > Help the Customer/business stakeholders define acceptance criteria
  - > Create automated acceptance tests
    - Or define for developers to script
  - > Expand scope of 'acceptance' tests
  - > Advise the team about overall risks and trends
  - > Perform manual/exploratory tests



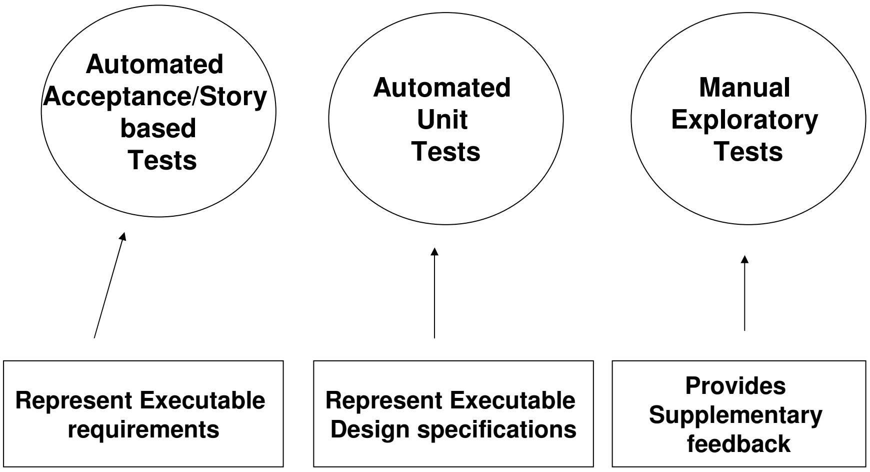
Scrum Test Strategy

- Risks
  - > Similar product risks
  - > Regression risk with high level of change
- How many test levels?
  - > XP appears to advocate two as part of a predefined test strategy
    - Unit and Acceptance (both automated as part of Test Driven Development)
    - Is system test no longer required?
  - > Automation reduces regression risk
  - > Developers doing testing reduces risk of poor quality code
  - > But how can a test strategy/approach be method rather than product based?

‘Acceptance’ Testing – is it enough?

- Usually not!…context/risk/strategy issue…
  - > May not be fully automated – partial regression strategy needed
  - > Expand to fuller ‘system’ tests
    - Functional testing
    - Non-functional testing – performance, usability, etc.
  - > May still need end-to-end business scenario focused User Acceptance test, user story interaction tests, etc.
  - > System integration testing issues
  - > Etc.
- Strategy and scheduling issue
  - > Adaptive, risk-driven

Testing within a Sprint



Automated Acceptance/Story based Tests

Automated Unit Tests

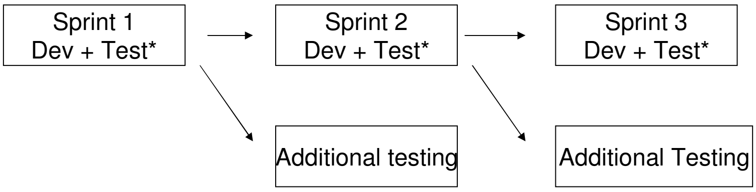
Manual Exploratory Tests

Represent Executable requirements

Represent Executable Design specifications

Provides Supplementary feedback

Sprints and Testing Strategy



***Sprint test = Automated Unit & Acceptance, Manual Exploratory**

At the end of a Sprint may need to perform additional testing as part of an adaptive testing strategy e.g.:

- **Additional exploratory testing**
- **Performance testing**
- **Usability testing**
- **Security testing**
- **System integration testing**
- **Combination/feature interaction testing**
- **Business cycle & end-to-end scenario testing – exercising multiple stories, end of month processing, etc.**

- Note**: Some of above may need to be included within a Sprint if needed for customer demo/release….define 'Done' for Sprint and also for Release**

Working software!!

Features, 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
  - User stories consist of a meaningful name, a brief description and conditions of satisfaction
  - At the deadline, a working (sub)system (set of features) is ready to potentially release
  - Any slippage results in low priority features dropped
- User stories have to fulfil the 3C criteria
  - Card: Small enough to fit on an index card
  - Conversation: Drives the conversation between product owner and team
  - Confirmation: Can be clearly validated/acceptance-tested

Sample User Story

Create a Meeting

As a Workgroup Member, I want to create a new meeting so that the meeting is saved to my calendar. I can state a meeting name and subject together with the date and start and end time.

Conditions of Satisfaction

Test that the Workgroup Member can select additional parameter such as conference reservation parameter if a voice conference is used.

Test supplying invalid data such as date is in the past, wrong date format, no meeting name, no start or end date, no subject.

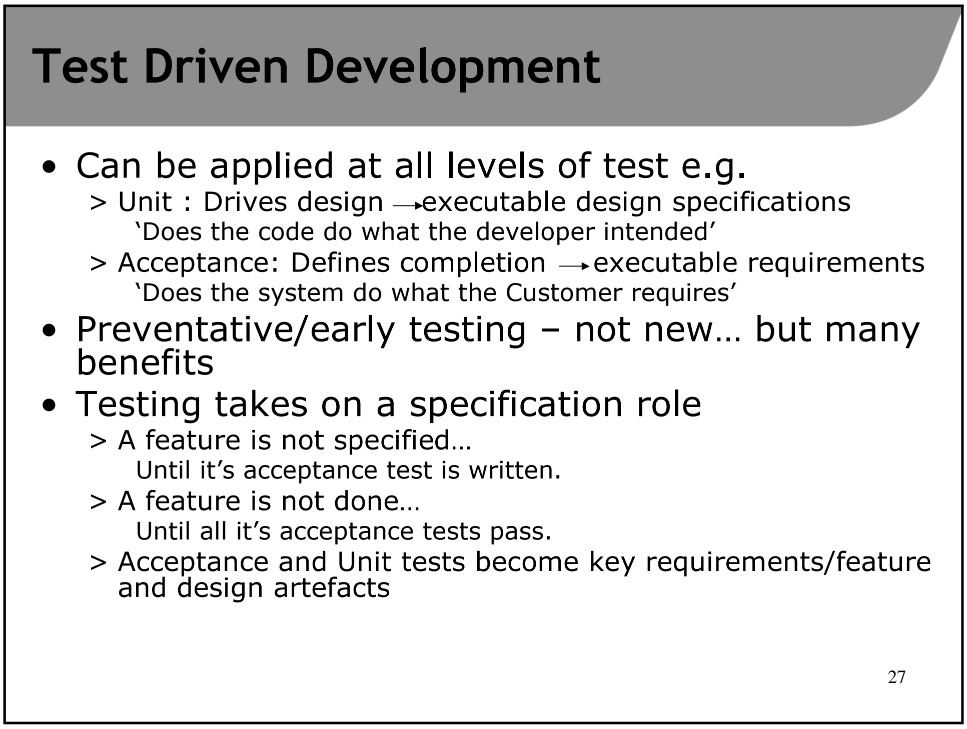
Test that the system offers the Workgroup Member the option to correct invalid entries.

Test Driven Development

- Probably the most effective single practice to improve quality
- Never write a single line of code unless you have a failing automated test
- Eliminate duplication

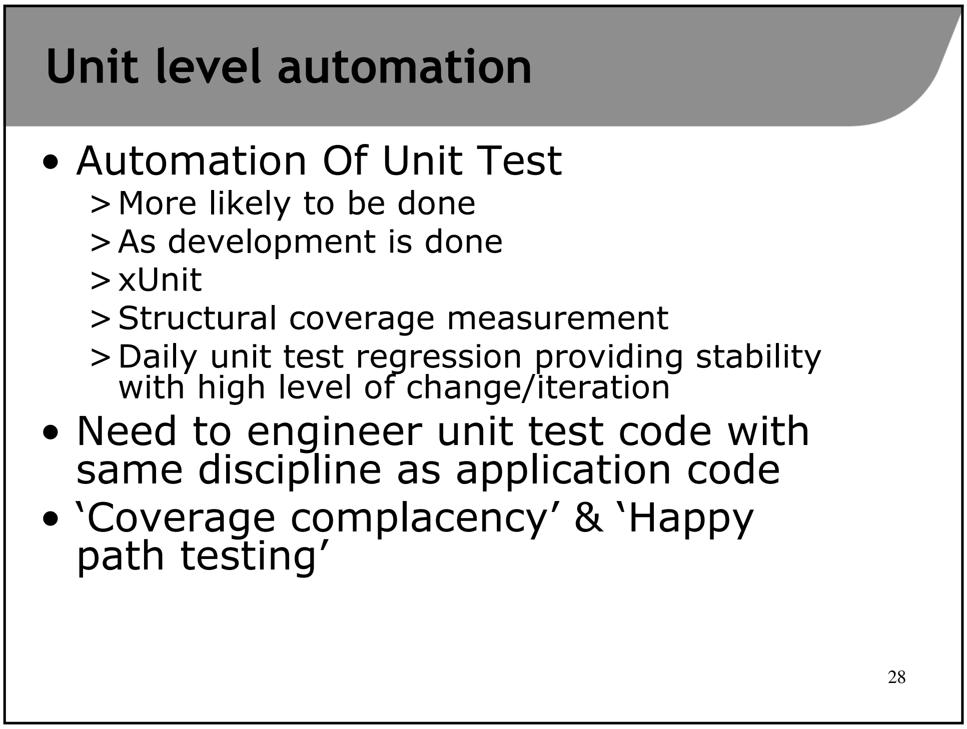
Kent Beck

- Write the test
- Write the code
- Refactor



Test Driven Development

- Can be applied at all levels of test e.g.
  - > Unit : Drives design → executable design specifications
    'Does the code do what the developer intended'
  - > Acceptance: Defines completion → executable requirements
    'Does the system do what the Customer requires'
- Preventative/early testing – not new… but many benefits
- Testing takes on a specification role
  - > A feature is not specified…
    Until it's acceptance test is written.
  - > A feature is not done…
    Until all it's acceptance tests pass.
  - > Acceptance and Unit tests become key requirements/feature and design artefacts

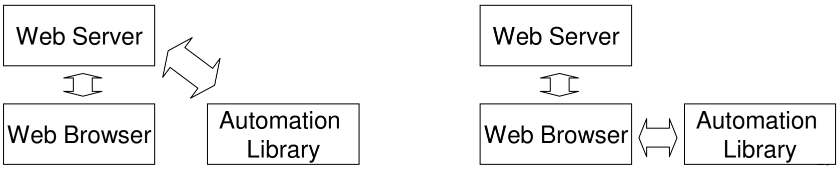


Unit level automation

- Automation Of Unit Test
  - > More likely to be done
  - > As development is done
  - > xUnit
  - > Structural coverage measurement
  - > Daily unit test regression providing stability with high level of change/iteration
- Need to engineer unit test code with same discipline as application code
- 'Coverage complacency' & 'Happy path testing'

Acceptance Testing

- Automated Acceptance Testing
  - > Design and Code for automation
  - > Automation Frameworks e.g.
    - Exactor, Brian Swan & Sean Hanly http://www.exoftware.com/xp_tools.htm
    - FIT/(Fitnesse), Ward Cunningham, http://fit.c2.com/
    - AXE, Odin Technology, http://www.odin.co.uk/
  - > Issues
    - Can be too low level or not business focused enough
    - Interface Drivers - Thin GUI layer issue….


Web Server
Web Browser
Automation Library
Web Server
Web Browser
Automation Library

Summary: Scrum - A New Way of Working

- **Incremental & Working Software!!!**
- Agility requires a particular mindset compared to traditional plan driven approaches – can be a significant cultural change
- Be proactive - Define your role and an appropriate test strategy
- Need highly automated regression testing
- Collaborate closely with developers and customers
- Business value drives priority
- Testing provides continuous feedback and decision support

Questions/discussion

Contact Details:

Fran.OHara@sogeti.com

Benefits Of Scrum

At its core, Scrum is an iterative, incremental process for developing any product or managing any work that produces a potentially shippable set of functionality at the end of each iteration

Scrum’s benefits are:

- Scrum is an agile process to manage and control development work.
- Scrum is a wrapper for (existing) engineering practices.
- Scrum is a team-based approach to developing systems when requirements are changing rapidly.
- Scrum controls the chaos of conflicting interests and needs.
- Scrum improves communication and maximises cooperation.
- Scrum detects and removes anything that gets in the way of developing and delivering products.
- Scrum is a way to maximise productivity.
- Scrum scales from single projects to entire organisations, and has managed development for multiple interrelated products and projects with over a thousand team members.
- Scrum is a way for everyone to feel good about their job, their contributions, and know they have done the very best they possibly could.