





De transitie naar beter Agile testen

Jeroen Mengerink
Jeroen.Mengerink@polteq.com
@AngusVB



Agenda

- Introduction Agile and SCRUM
- Improvement model
- Key areas
 - Introduction
 - Levels
 - Checkpoints (first level)
 - Good practices

© 2015 2 

Agenda

- **Introduction Agile and SCRUM**
- Improvement model
- Key areas
 - Introduction
 - Levels
 - Checkpoints (first level)
 - Good practices

© 2015

3



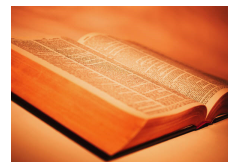
What is Agile?

agile /'ædʒaɪl; NAmE 'ædʒɪl/ adj.

1 able to move quickly and easily SYN nimble: a strong and agile athlete

2 able to think quickly and in an intelligent way: an agile mind / brain

(source: Oxford Advanced Learner's dictionary)



© 2015

4



Agile Manifesto

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.

(source: <http://www.agilemanifesto.org/>)

© 2015

5



SCRUM

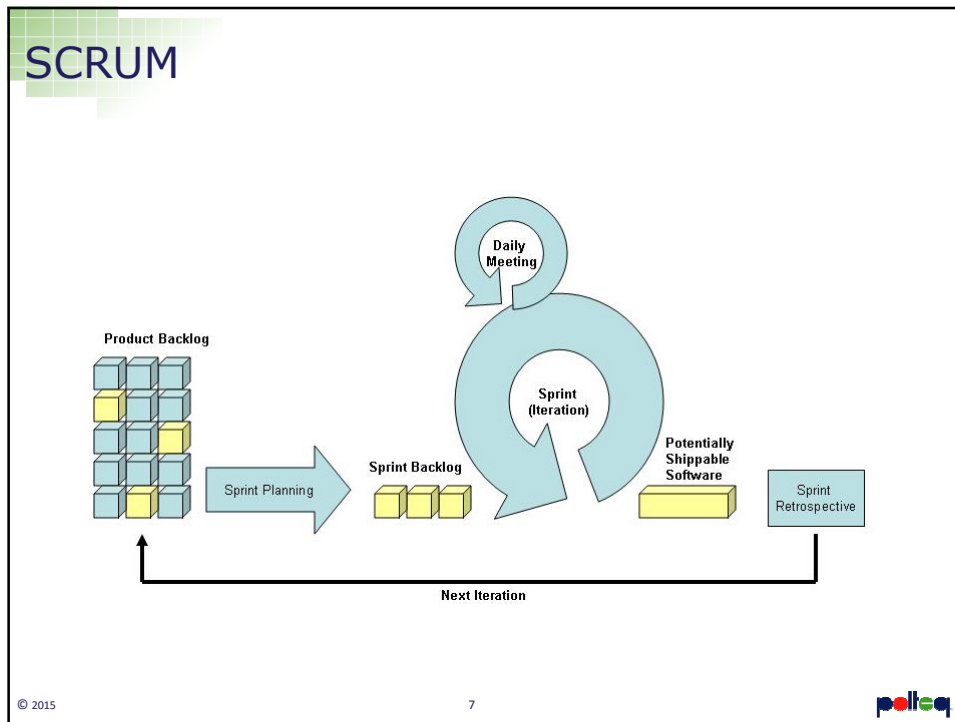
SCRUM is an iterative and incremental method for managing software projects and product or application development

- Raise the effectiveness of the team
- Guard the progress of the team
- Solve impediments
- Guard the project progress
- Minimizing of project risks

© 2015

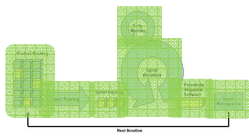
6






SCRUM

- Product Backlog – list of “requirements”
- Sprint Planning – to determine the content of the sprint backlog
- Sprint Backlog – list of detailed function descriptions split down into tasks for the current sprint
- Sprint – work period of 1 – 4 weeks
- SCRUM meeting – daily project status meeting
- Shippable code – usable, tested functionality
- Retrospective - demo to stakeholders, followed by a reflection



© 2015 8 

Agile/SCRUM

Agile

- Set of basic ideas
- Little overhead
- Focused on people

SCRUM

- Process management
- Short iterations
- Multidisciplinary teams

© 2015

9



Transitioning from traditional to agile

Process	Waterfall Development	Transition	Agile Development
Measure of Success	Conformance to plan	→	Response to change, working code
Management Culture	Command & control	→	Leadership, self managing teams
Requirements & Design	Big & upfront documentation	→	Continuous, just in time elaboration
Coding & Implementation	Code all features in parallel. Test later	→	Code & unit test, deliver serially
Test & Quality Assurance	Big, planned / test late	→	Continuous & concurrent testing starts early
Planning & Scheduling	detailed / fixed scope, estimate time & resources	→	Two-level plan / fix date, estimate scope

(Scaling Software Agility: Best Practices for large Organisations – DeanLeffingwell 2007)

© 2015

10



Agenda

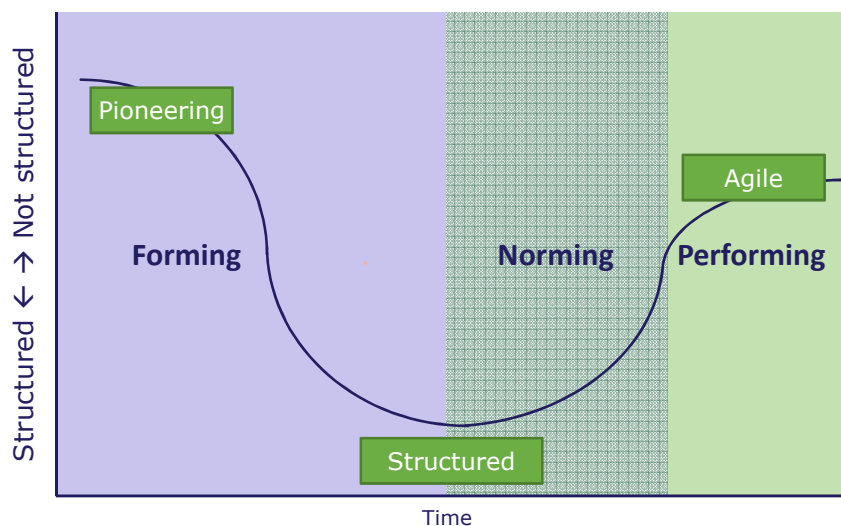
- Introduction Agile and SCRUM
- **Improvement model**
- Key areas
 - Introduction
 - Levels
 - Checkpoints (first level)
 - Good practices

© 2015

11



Evolution of Agile testing

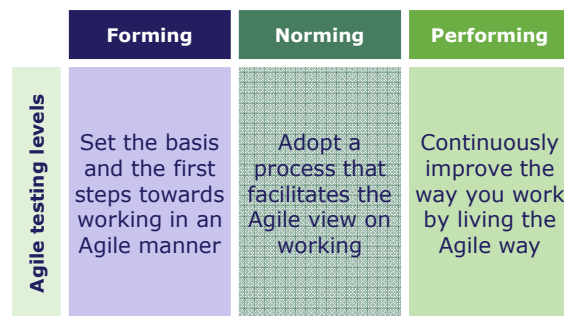


© 2015

12



Agile testing maturity levels



© 2015

13



Discussion

Which areas would you investigate if you want to know how mature a company is with Agile testing?

Assessment model

Key area	Forming				Norming				Performing			
1 Stakeholder commitment	1	2	3	4	1	2	3	4	1	2	3	
2 Planning & Estimation	1	2	3	4	1	2	3	4	1	2	3	4
3 People	1	2	3	4	1	2	3	4	1	2	3	
4 Interaction	1	2	3	4	1	2	3	4	1	2	3	4
5 Teamwork	1	2	3		1	2	3	4	1	2	3	4
6 Test process	1	2	3		1	2	3	4	1	2	3	
7 Test management	1	2	3	4	1	2	3		1	2	3	
8 Test profession	1	2	3	4	1	2	3	4	1	2	3	
9 Test automation	1	2	3		1	2	3	4	1	2	3	4
10 Regression & E2E testing	1	2	3	4	1	2	3		1	2	3	4
11 Defect management	1	2	3		1	2	3	4	1	2	3	
12 Test environment	1	2	3		1	2	3		1	2	3	4

© 2015

15



Assessment model

- There is no initial level like in other models
- Each key area has a number of checkpoints
- The levels are continuous
 - Next maturity level builds on previous level

© 2015

16



Assessment model: example result

Key area	Forming				Norming				Performing			
1 Stakeholder commitment	1	2	3	4	1	2	3	4	1	2	3	
2 Planning & Estimation	1	2	3	4	1	2	3	4	1	2	3	4
3 People	1	2	3	4	1	2	3	4	1	2	3	
4 Interaction	1	2	3	4	1	2	3	4	1	2	3	4
5 Teamwork	1	2	3		1	2	3	4	1	2	3	4
6 Test process	1	2	3		1	2	3	4	1	2	3	
7 Test management	1	2	3	4	1	2	3		1	2	3	
8 Test profession	1	2	3	4	1	2	3	4	1	2	3	
9 Test automation	1	2	3		1	2	3	4	1	2	3	4
10 Regression & E2E testing	1	2	3	4	1	2	3		1	2	3	4
11 Defect management	1	2	3		1	2	3	4	1	2	3	
12 Test environment	1	2	3		1	2	3		1	2	3	4

© 2015

17



Agenda

- Introduction Agile and SCRUM
- Improvement model
- **Key areas**
 - Introduction
 - Levels
 - Checkpoints (first level)
 - Good practices

© 2015

18



01. Stakeholder commitment – Introduction

- Recognise the value of a team and of each role in the team
- Create an environment in which the team can work effectively
- Deliver acceptance criteria
- Participate in acceptance
- Prioritise backlog, risks and defects
- Deliver context for user stories
- Allow freedom of choice
- Facilitate Agile/SCRUM

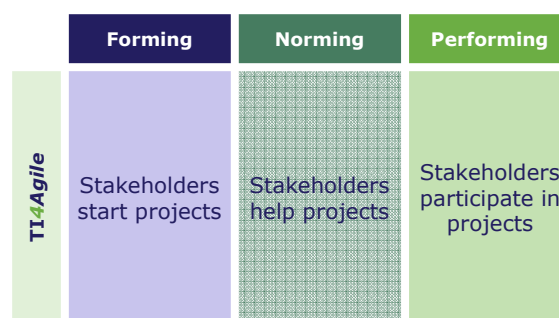


© 2015

19



01. Stakeholder commitment – Levels



© 2015

20



01. Stakeholder commitment – Forming

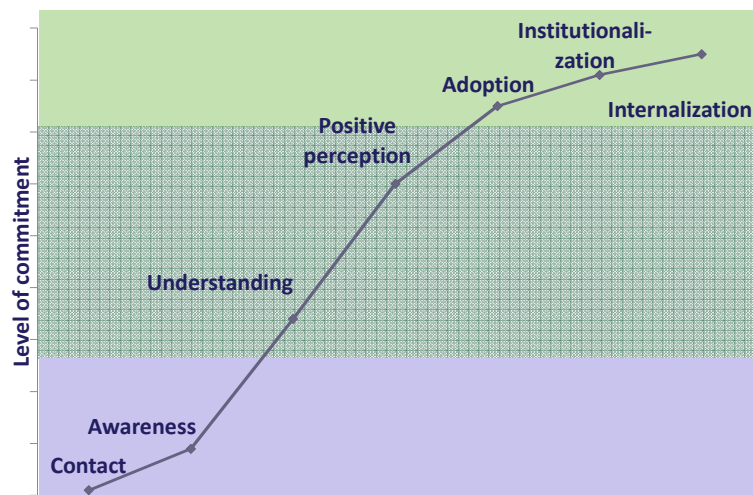
1. The principal stakeholder is defined and known by the testers
2. Stakeholders deliver the committed resources
3. Stakeholders actively acquire information on the progress of the project
4. Stakeholders are willing to adapt their way of working to the test process

© 2015

21



01. Stakeholder commitment – The path to commitment

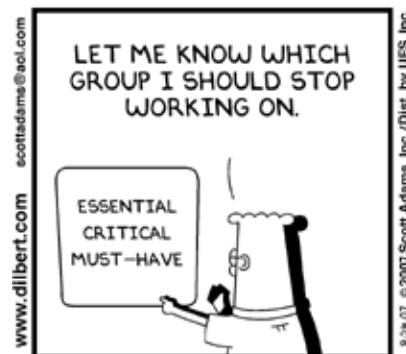


© 2015

22



Prioritize backlog



© 2015

23



02. Planning & Estimation – Introduction

- Work breakdown
- Plan for all activities
 - Preparation
 - Review
 - Execution
 - Retest
 - Unit test
 - Bug fixing
- Team effort is important

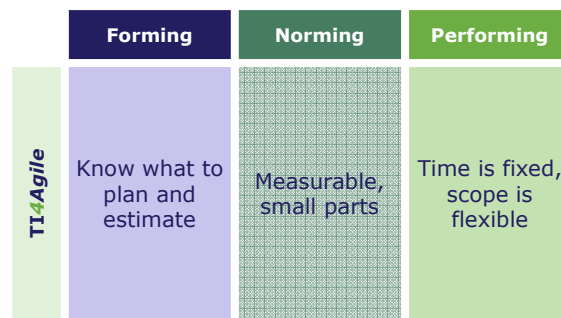
PLAN FIRST!

© 2015

24



02. Planning & Estimation – Levels



© 2015

25



02. Planning & Estimation – Forming

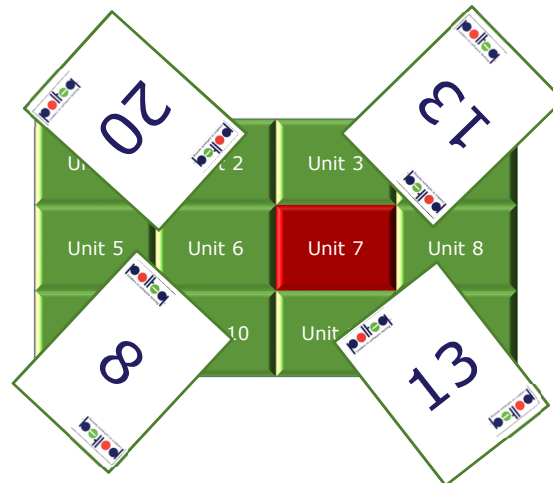
1. Each test activity is planned (prepare - define - execute)
2. Test levels are defined and overlap is minimized (UT – ST - AT)
3. A technique for test effort estimation is applied
4. Planning is agreed with the stakeholders

© 2015

26



02. Planning & Estimation – Planning poker

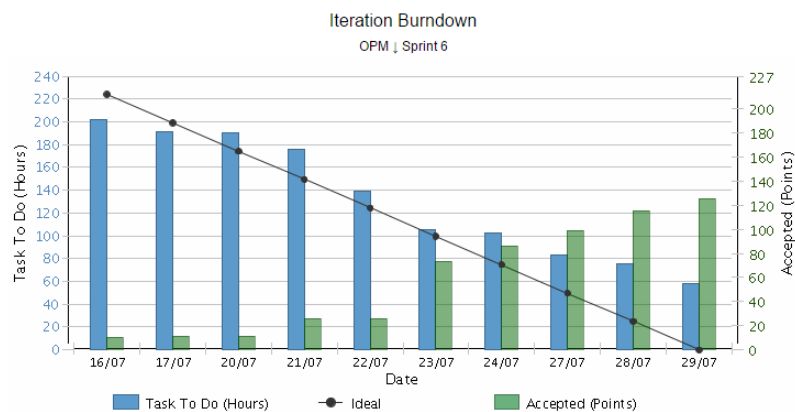


© 2015

27



02. Planning & Estimation – Burn-down chart



© 2015

28






Discussion

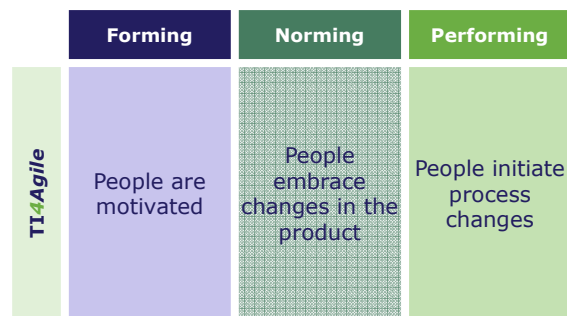
What would you conclude from the given burn-down chart?

03. People – Introduction



© 2015 30 

03. People – Levels



© 2015

31



03. People – Forming

1. People are well trained and/or experienced in their functions
2. People are willing to put in extra effort when needed (commitment)
3. People can explain their value in the project context
4. People take full responsibility for their work

© 2015

32



03. People – Feedback

- State something positive related to the subject
- State your criticism objectively
- Don't use the word "but"
- State the effect
- Suggest an improvement



Thank you for delivering a lot of information in the meeting. I noticed that you were talking a lot, this provided me little room to give my opinion. You might want to ask others for their input in the future.

© 2015

33



03. People – T-shaping

- Know the context
- Know your own strengths
- Assist others
- Leave your comfort zone



© 2015

34

34



04. Interaction – Introduction

- Actively share information
- Open and honest
- Use meetings for their purpose
 - Understandable for all attendees

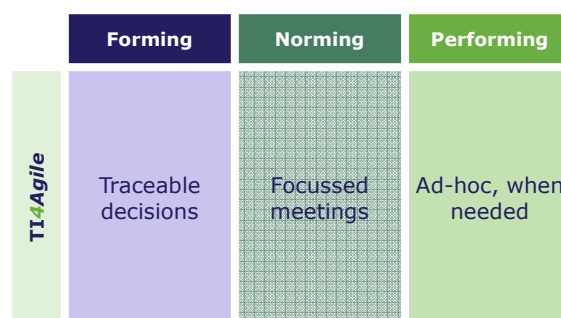


© 2015

35



04. Interaction – Levels



© 2015

36



04. Interaction – Forming

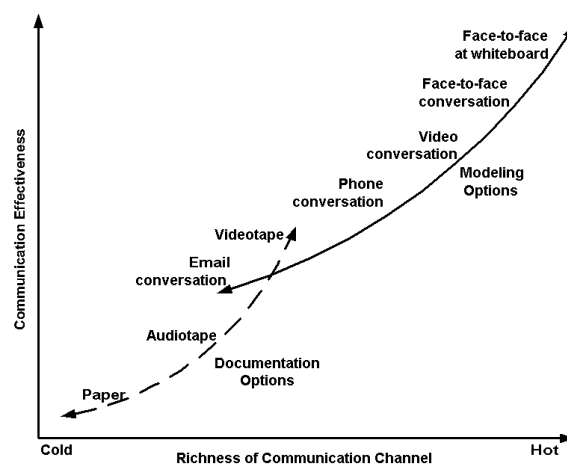
1. It is possible to trace back points of action, agreements and decisions
2. Progress, product quality and risks are communicated
3. Meetings are only attended by relevant people
4. Tooling to support the necessary communication within the project is supplied

© 2015

37



04. Interaction – Communication = essential



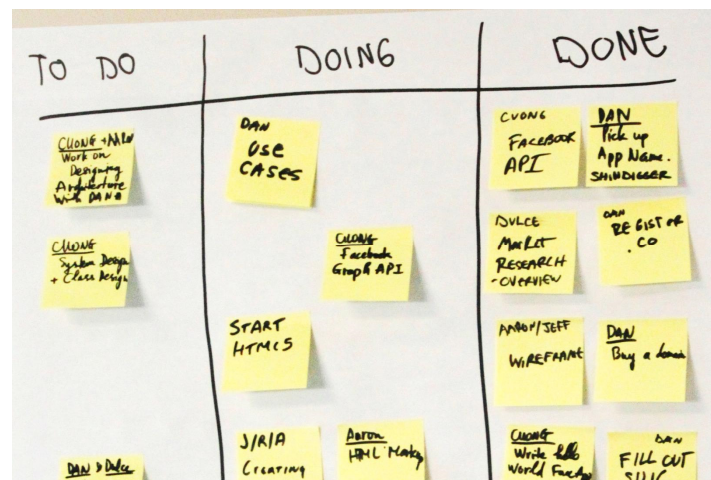
Original Diagram Copyright 2002 Alistair Cockburn, Modified Version Scott Ambler 2002

© 2015

38



04. Interaction – SCRUM board



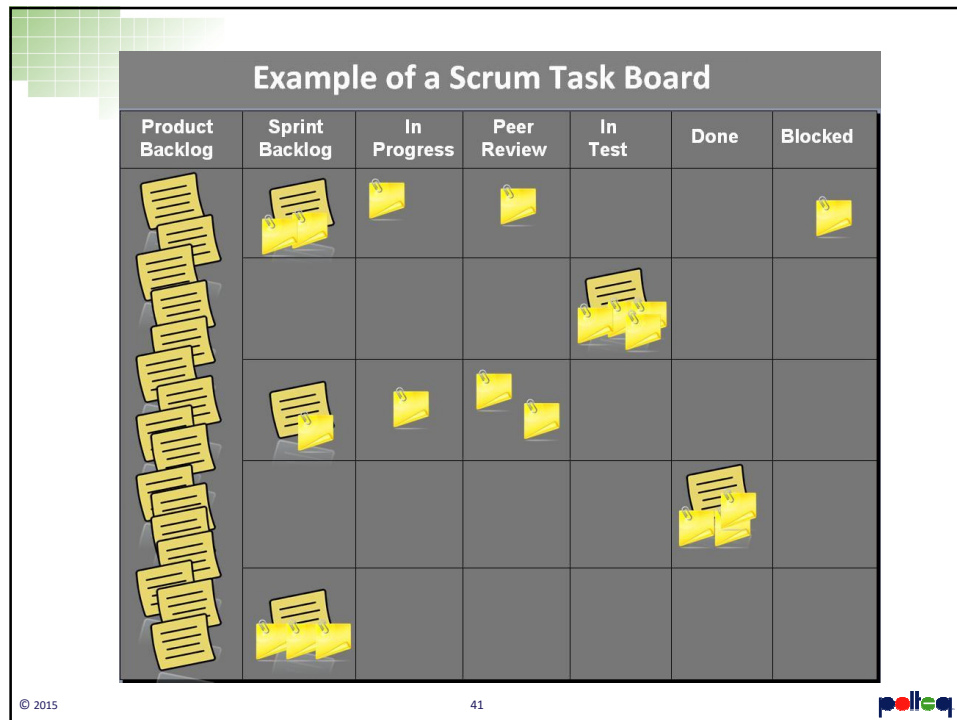
© 2015

39



Discussion

**Pro's and con's of different SCRUM
board layouts**



04. Interaction – Timeboxing



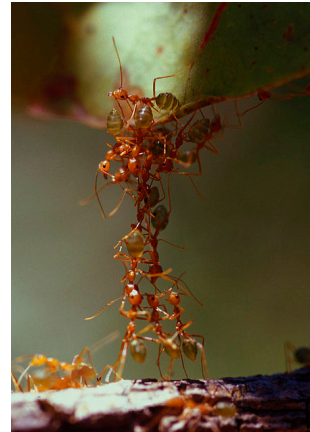
© 2015

42



05. Teamwork – Introduction

- Respect and trust each other
- Commitment as a team
 - Work towards a team goal
- Help each other where possible
 - Multi disciplined

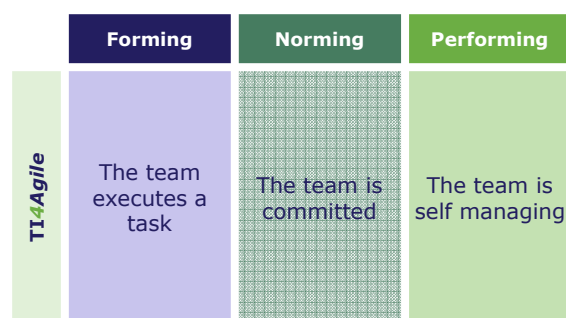


© 2015

43



05. Teamwork – Levels



© 2015

44



05. Teamwork – Forming

1. The team is co-located
2. Each team member is involved in important decisions
3. The team understands its goal

© 2015

45



05. Teamwork – Keep the goal visible

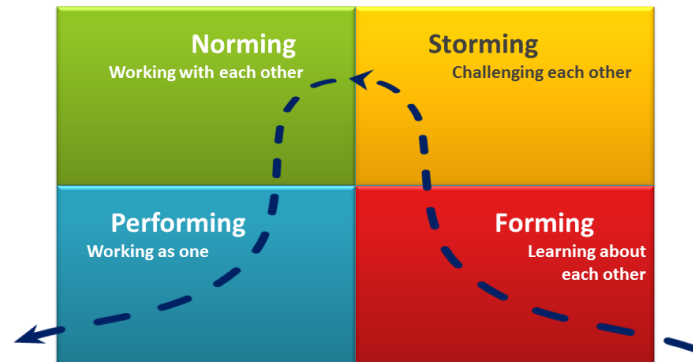


© 2015

46



05. Teamwork – Team development



(Bruce W. Tuckman – Stages of group development)

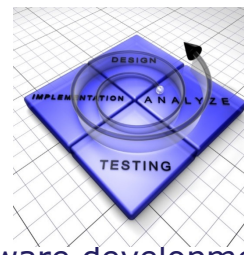
© 2015

47



06. Test process – Introduction

- What and when to test?
- Finish what you start
- Testing is fully integrated in the software development process
- Everybody tests

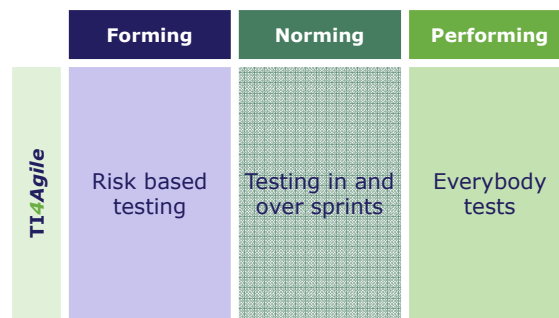


© 2015

48



06. Test process – Levels



© 2015

49




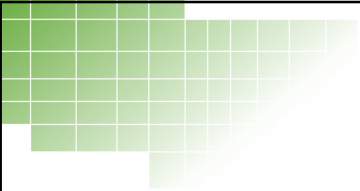
06. Test process – Forming

1. The effort for test tasks is based on product risks
2. Test design techniques are applied when creating test cases
3. Testing considers different quality characteristics

© 2015

50



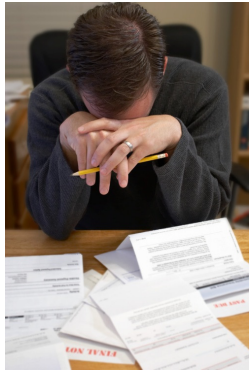



Discussion

**When do you do a PRA?
(product risk assessment)**

06. Test process – Testing debt

- Automated unit tests
- Not all regression testing completed
- Non-functional testing
- Coverage of automation for acceptance tests
- Test data



© 2015 52 

06. Test process – Pair testing



© 2015

53



07. Test management – Intro

- People management
- Generic test approach / strategy
- Risk analysis
- Release planning
- Keep the big picture (birds eye view)
 - Cross teams
 - Cross sprints
 - Cross projects

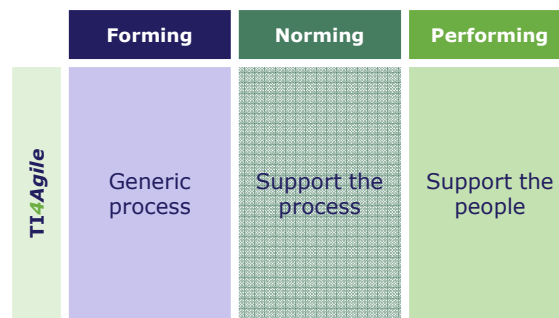


© 2015

54



07. Test management – Levels



© 2015

55



07. Test management – Forming

1. Test management allocates the proper test staff
2. A generic test approach is available
3. Test management provides birds eye view on quality and risks in the project
4. Test management contributes to the release planning by adding a quality view to high level estimation

© 2015

56



07. Test management – Generic test approach / strategy

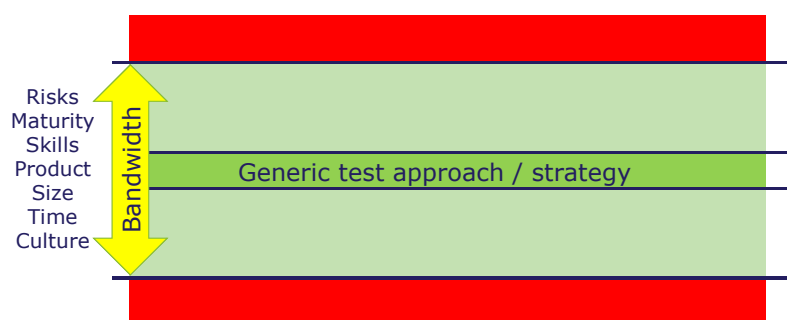
- Training and coaching
- Defect management
- Risk analysis
- Non-functionals
- E2E testing
- Tooling and automation
- Environment management
- Metrics and reporting
- ...

© 2015

57



07. Test management – Generic test approach / strategy

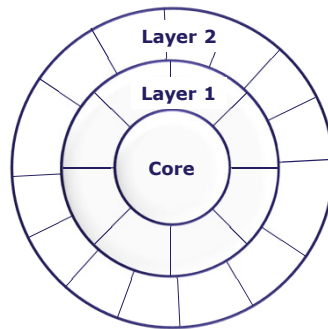


© 2015

58



07. Test management – The risk wheel



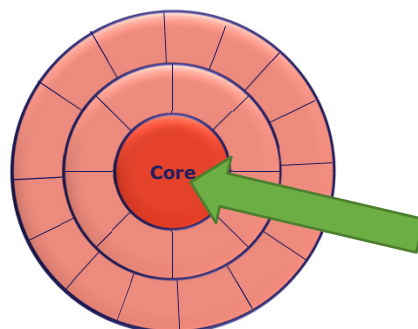
**Not every adjustment
has an equal impact**

© 2015

59



07. Test management – The risk wheel



Core:

Technical:

- Database interaction
- Most used DLL's

Functional:

- Most important proces
- Most used functionality

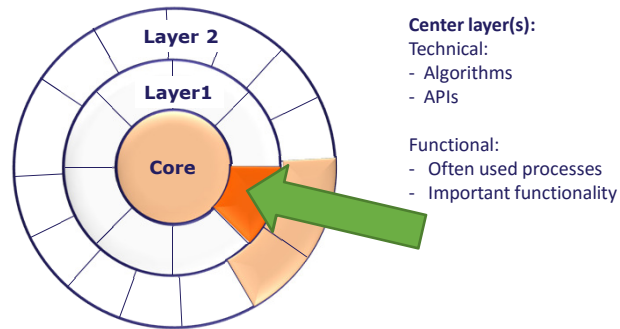
Impact : Huge

© 2015

60



07. Test management – The risk wheel



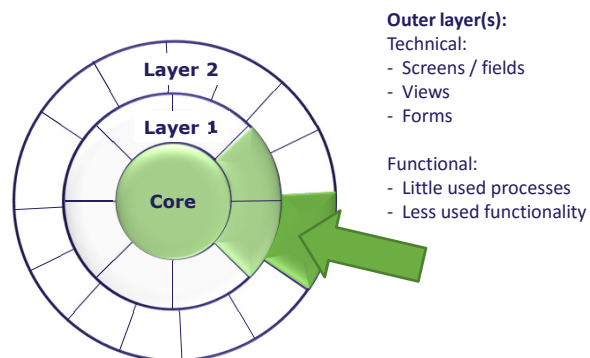
Impact : *Middle*

© 2015

61



07. Test management – The risk wheel



Impact : *Small*

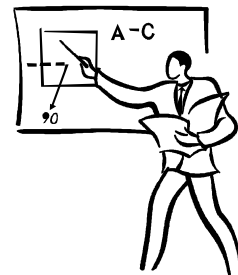
© 2015

62



08. Test profession – Introduction

- More focus on soft skills
- Technical aspects (drivers, stubs, logging, ...) are part of the job
- Share test knowledge
 - Within profession, over teams
 - Within team, over professions
- It is more than just testing

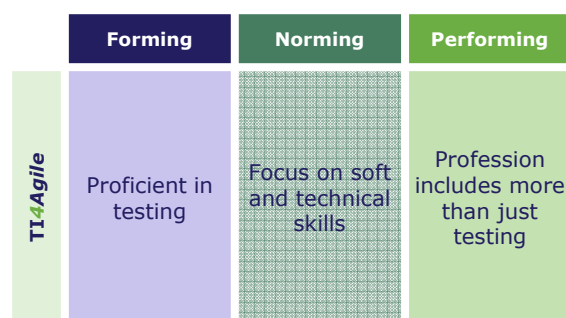


© 2015

63



08. Test profession – Levels



© 2015

64



08. Test profession – Forming

1. Testers have received specific test training and/or have sufficient experience in the field of structured testing
2. Testers can explain the rationale behind chosen techniques that have been applied
3. Test functions are part of the organizations career development
4. Testers can explain their added value

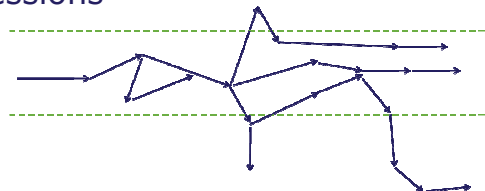
© 2015

65



08. Test profession – Exploratory testing

- The tester simultaneously:
 - learns about the product and its defects
 - plans the testing work to be done
 - designs and executes the tests
 - and reports the results
- Charters
- Fixed time sessions



© 2015

66



08. Test profession – Support others in testing activities

- Introduce boundary value testing to developers
- Introduce what-if thinking to designers
- Suggest scenarios to the product owner

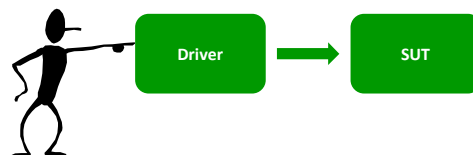


© 2015



08. Test profession – Driver

- A driver allows you to call functionality in the system under test
- After executing the driver, the tester checks the response of the system under test



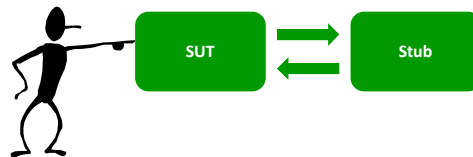
© 2015

68



08. Test profession – Stub

- Stubs are used to simulate unexisting parts that are necessary for the system under test
- The tester checks the output of the system under test, knowing that simulated data was used during processing



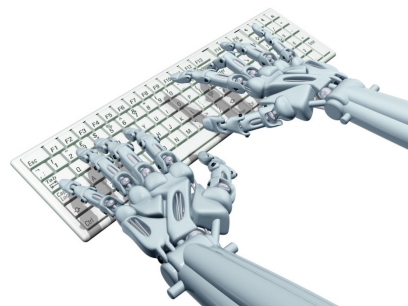
© 2015

69



09. Test automation – Introduction

- Continuous integration
- At every level
- Important part of the sprint
- Risk based
- Maintainable

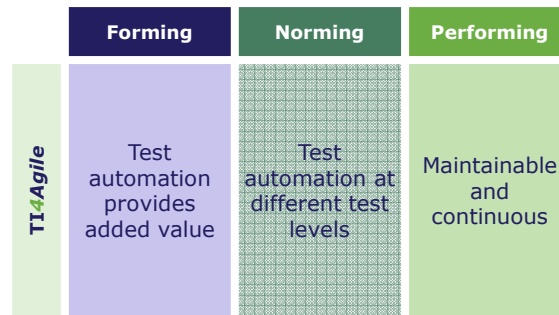


© 2015

70



09. Test automation – Levels



© 2015

71



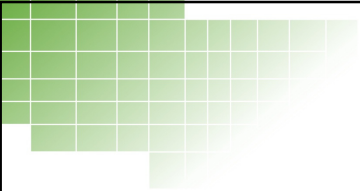

09. Test automation – Forming

1. The organization acknowledges the importance of test tooling and allocates sufficient budget
2. Test cases are automatable by describing them using 'inputs - actions - expected results'
3. Decisions on what to automate are based on risks and return on investment

© 2015

72

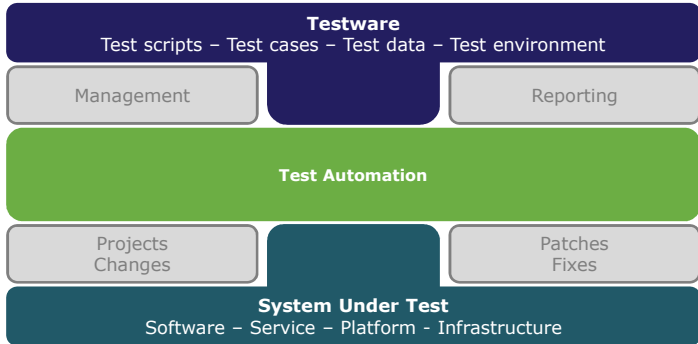


Discussion

Do we need to optimize the test automation at team level or at organizational level?

09. Test automation – Test Automation Context



```

graph TD
    subgraph Testware
        TS[Test scripts]
        TC[Test cases]
        TD[Test data]
        TE[Test environment]
    end
    subgraph TestAutomation [Test Automation]
        M[Management]
        R[Reporting]
    end
    subgraph SystemUnderTest [System Under Test]
        PC[Projects Changes]
        PF[Patches Fixes]
    end
    Testware --- TestAutomation
    TestAutomation --- SystemUnderTest
    
```


Testware
Test scripts – Test cases – Test data – Test environment

Management Reporting

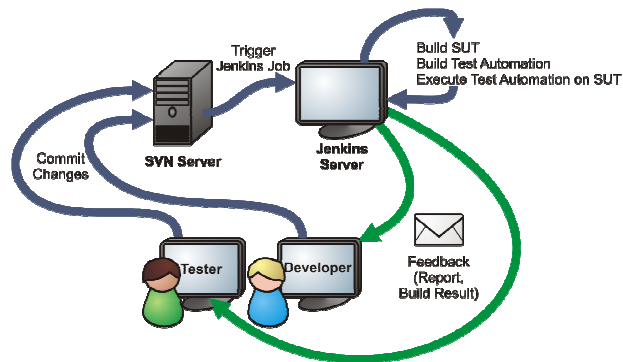
Test Automation

Projects Changes Patches Fixes

System Under Test
Software – Service – Platform – Infrastructure

© 2015 74 

09. Test automation – Continuous integration



© 2015

75



10. Regression & E2E testing – Introduction

- Focus wider than current sprint
 - Probably wider than the team
- Assign where and when
- Business value

Regression:
"when you fix one bug, you
introduce several newer bugs."

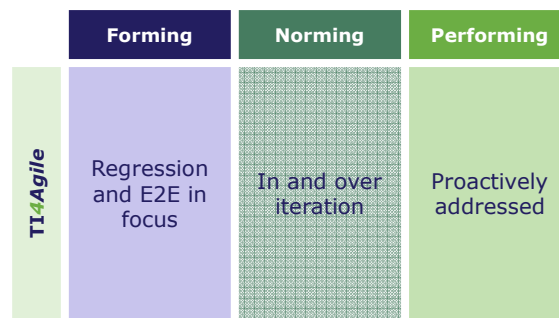


© 2015

76



10. Regression & E2E testing – Levels



© 2015

77



10. Regression & E2E testing – Forming

1. There is a basic strategy for regression testing
2. There is a basic strategy for E2E testing
3. Test sets are maintained and updated
4. Responsibilities for E2E and regression testing are clear

© 2015

78



10. Regression & E2E testing – Mind mapping

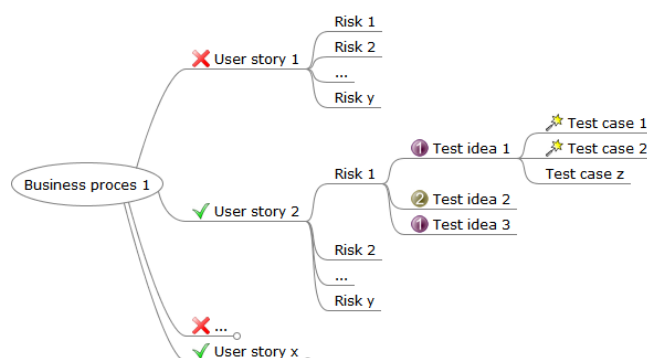
- Visualize information in a structured way
- Create a clear view on E2E information
 - Which parts are done?
 - Which risks are involved?
 - How to test?
- Understandable for different types of people

© 2015

79



10. Regression & E2E testing – Mind mapping



© 2015

80



10. Regression & E2E testing – Coping with regression

- Hardening iteration
 - Iteration focused on improving the product
 - E2E
 - Regression
 - Loose ends
 - Shortcuts
 - Non-functionals



- Should be used as little as possible

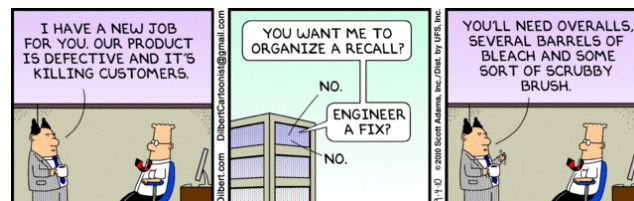
© 2015

81



11. Defect management – Introduction

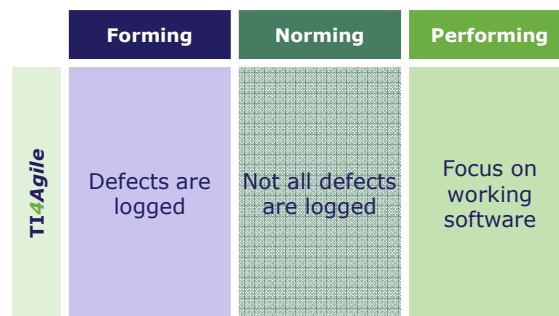
- Decide when to log a defect
- Decide what to log
- Pay attention to traceability
- Use one system for all defects



© 2015



11. Defect management – Levels



© 2015

83



Discussion

When do you log a defect, and what could be reasons not to log a defect?

11. Defect management – Forming

1. All persons involved in logging and/or tracking defects use the same defect tracking tool
2. Defects have a common set of characteristics (e.g. unique id, description, status)
3. All raised defects are followed up

© 2015

85



11. Defect management – Possibilities



- Log all defects
- Use a defect tracking system and apply rules on when to log a defect
- Create an automated test for the defect and let that be your defect documentation
- Create a card on your SCRUM board for every defect

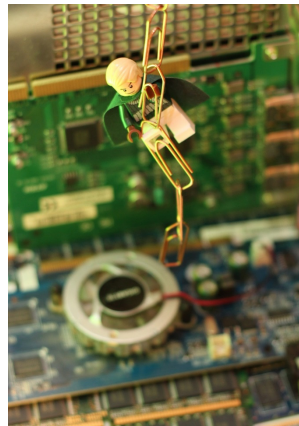
© 2015

86



12. Test environment – Introduction

- Self controlled
- Production like
- Version controlled
- Continuous integration

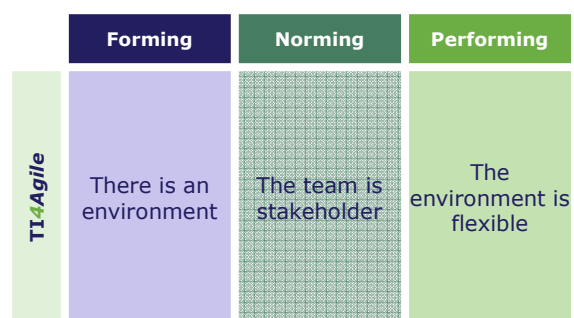


© 2015

87



12. Test environment – Levels



© 2015

88



12. Test environment – Forming

1. The environment is managed (setup, maintenance, version management, etc.)
2. The environment is sufficiently representative for the test
3. The data used in the environment can easily be refreshed

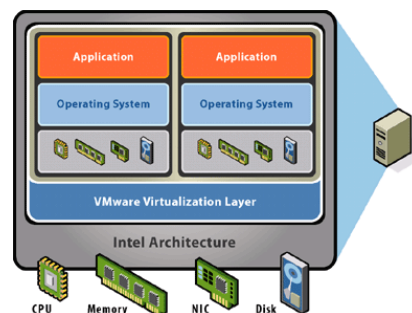
© 2015

89



12. Test environment – Virtualization

- Compatibility testing
 - Create a set of virtual machines
 - Test on different virtual machines

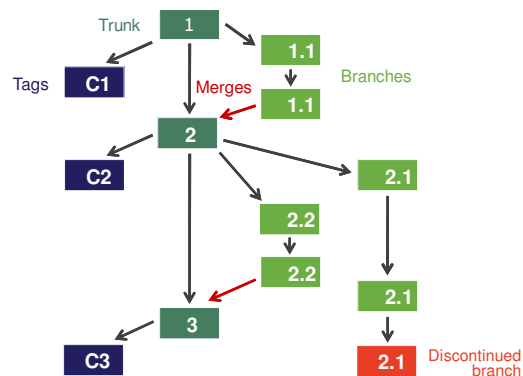


© 2015

90



12. Test environment – Version management



© 2015

91



Questions?

Key area	Forming				Norming				Performing			
1 Stakeholder commitment	1	2	3	4	1	2	3	4	1	2	3	4
2 Planning & Estimation	1	2	3	4	1	2	3	4	1	2	3	4
3 People	1	2	3	4	1	2	3	4	1	2	3	4
4 Interaction	1	2	3	4	1	2	3	4	1	2	3	4
5 Teamwork	1	2	3	4	1	2	3	4	1	2	3	4
6 Test process	1	2	3	4	1	2	3	4	1	2	3	4
7 Test management	1	2	3	4	1	2	3	4	1	2	3	4
8 Test profession	1	2	3	4	1	2	3	4	1	2	3	4
9 Test automation	1	2	3	4	1	2	3	4	1	2	3	4
10 Regression & E2E testing	1	2	3	4	1	2	3	4	1	2	3	4
11 Defect management	1	2	3	4	1	2	3	4	1	2	3	4
12 Test environment	1	2	3	4	1	2	3	4	1	2	3	4

<https://improvement.polteq.com/ti4agile/>

© 2015

92

