Testing in Agile methodologies – easier or more difficult?

Lucjan Stapp
Warsaw University of Technology
Stowarzyszenie Jakości Systemów Informatycznych
L.Stapp@mini.pw.edu.pl
L.Stapp@sjsi.org

Professor in Warsaw University of Technology

Author of more than 40 papers, more then 10 are connected with testing;

Ex - believer of Agile methodologies, especially in testing;

Acting vice-president of Stowarzyszenia Jakości Systemów Informatycznych (Polish Testing Board);

Member of ISTQB Dictionary Working Group.

Agile manifesto

In February 2001, 17 software developers met at the Snowbird, Utah resort, to discuss lightweight development methods.

They published the **Agile Manifesto** (*Manifesto for Agile Software Development*) to define the approach now known as agile software development.

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:

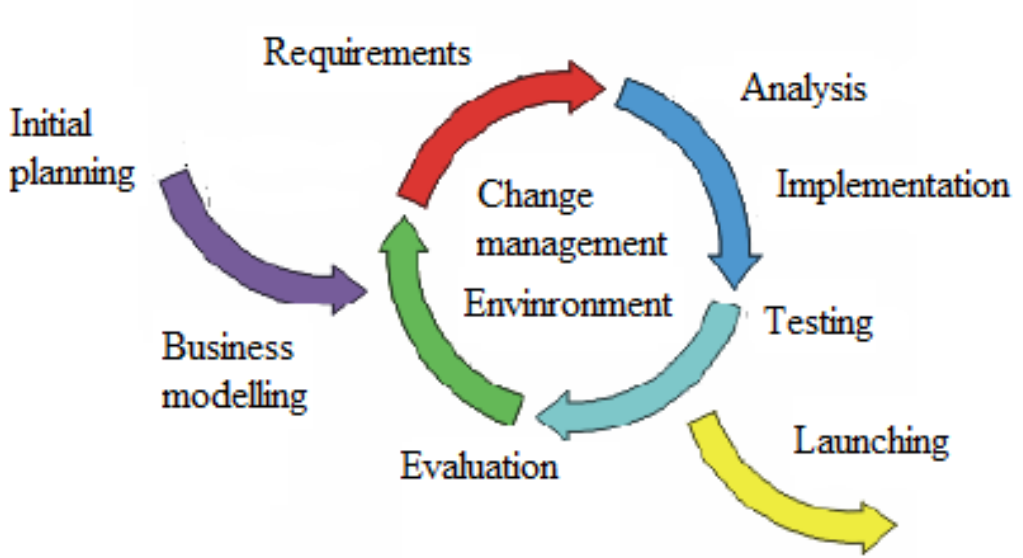
Individuals and interactions	**over**	processes and tools
Working software	**over**	comprehensive documentation
Customer collaboration	**over**	contract negotiation
Responding to change	**over**	following a plan

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

agilemanifesto.org

Agile manifesto

Only iterative incremental approach



As a result of each iteration we obtain the **working** software

Agile manifesto

Basic Agile principles:

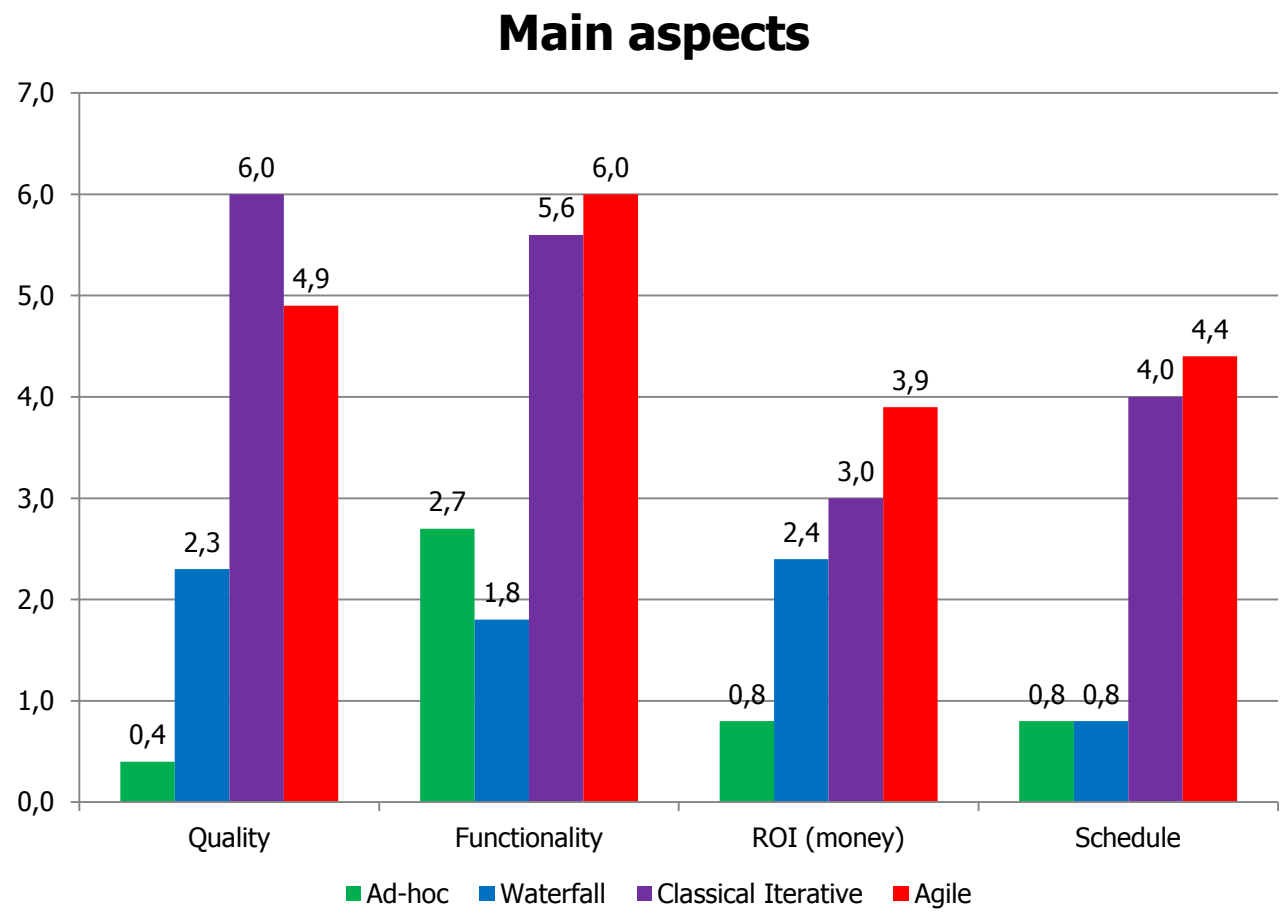
- **Working software**,
- Self-organization and motivation,
- **Informal** communication,
- Inspect and adapt,
- Customer collaboration:
  - Welcome changing requirements, even late in development.

Agile manifesto

Agile methodologies (between others):

- *eXtreme Programming XP,*
- *Scrum*,
- Kanban,
- Dynamic Systems Development Method,
- Adaptive Software Development,
- ……

How it works?



Agile teams are more effective in desired functionality then "traditional" teams. But less in quality.

Results of Dr. Dobbs Journal Investigations(2008)

How it works?

Higher ROI.

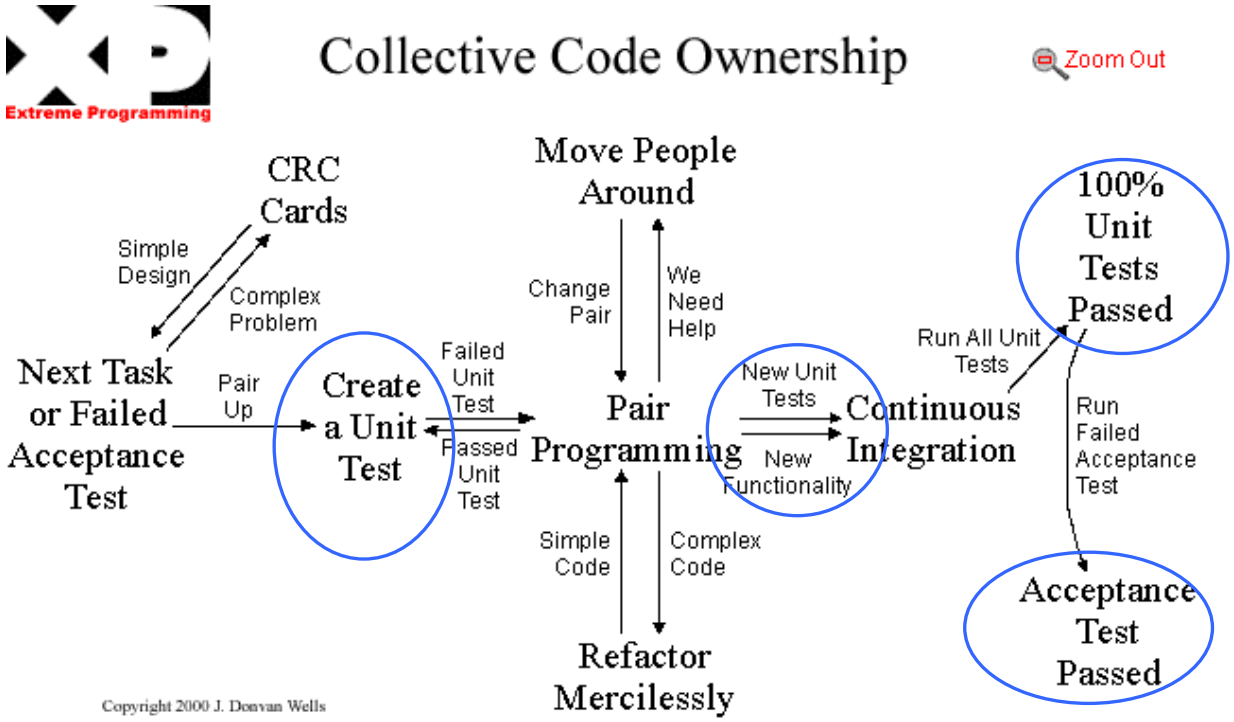
Agile teams work in more effective way (but not harder) and should propose needed functionality earlier, hence there is a shorter path to the market and greater profit.

Faith

DDJ found also that people believe that Agile teams product higher quality (however it is **NOT** true) then traditional teams, hence one can observe higher satisfaction of stakeholders.

eXtreme Programming

"First" popular Agile methodology is **X**treme **P**rogramming (XP).



eXtreme Programming

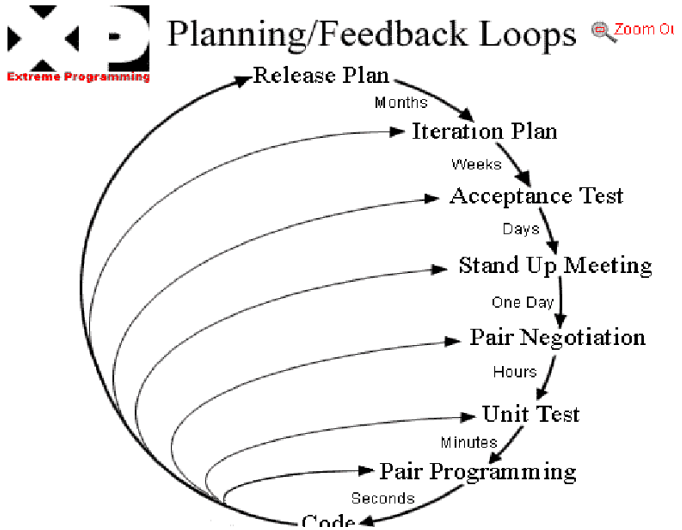
Basic XP testing assumptions.

- **All code must have unit (module) tests,**
- **All code must pass all module tests before it can be released,**
- **When a failure is found tests are created before the failure is addressed ,**
- **Acceptance tests are run often and the results are published.**

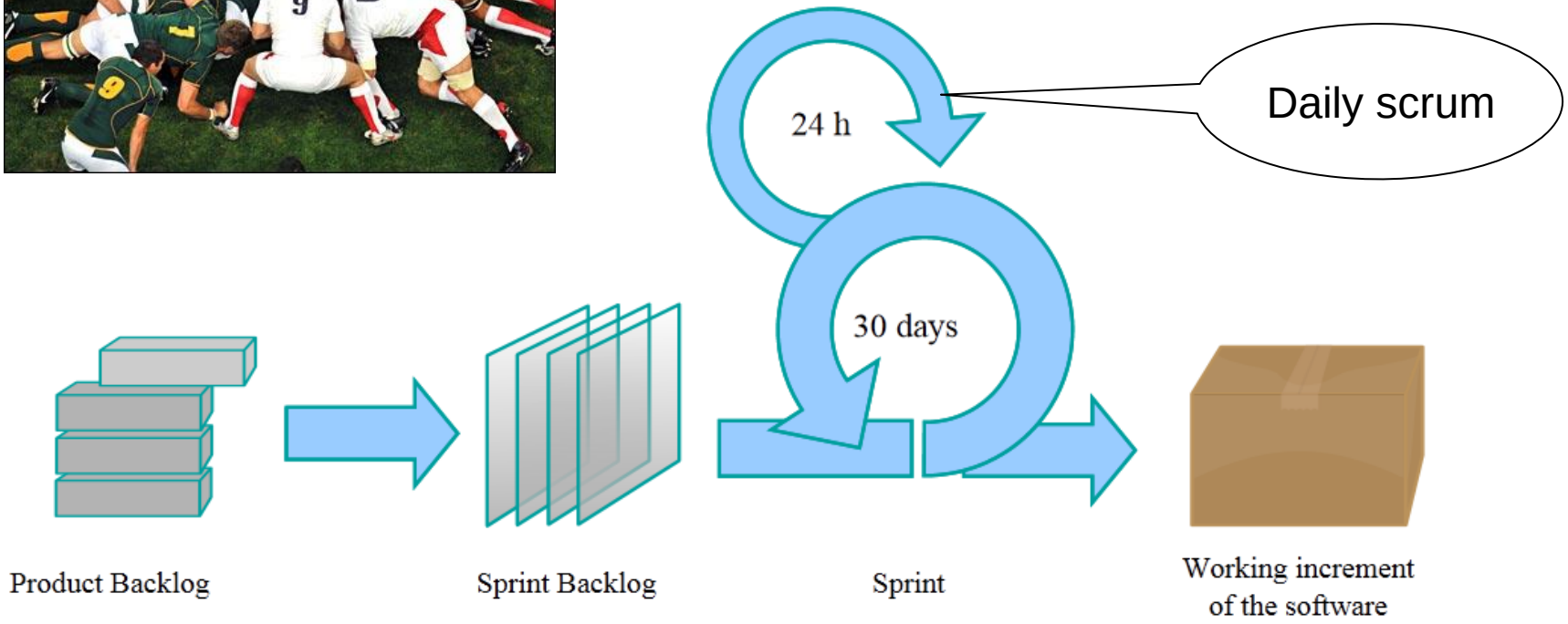
eXtreme Programming

But:

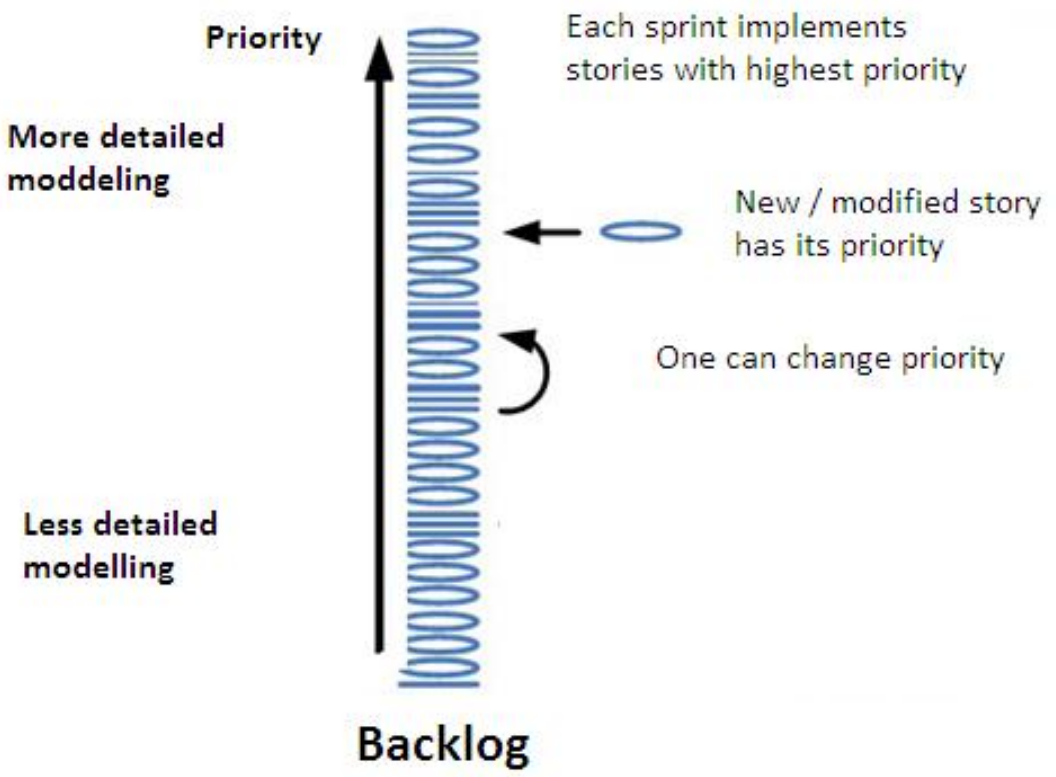
- Tests are taken on module level (XUnit tests),
- 2 builds a week – strong automation is needed (rather expensive),
- Frequent changes – even on accepted by testers modules,
- There is no place for "business" acceptance tests,
- Systems of systems problem – external integration tests.



Scrum


24 h
Daily scrum
30 days
Product Backlog
Sprint Backlog
Sprint
Working increment
of the software

Scrum


Priority
More detailed
moddeling
Less detailed
modelling
Each sprint implements
stories with highest priority
New / modified story
has its priority
One can change priority
Backlog

Scrum

User Story

- Elements of User Story
  - As… (concrete user)
  - I want… (problem to be solved)
  - because… (desired results).
- Definition of user satisfaction criteria
  - Typically as acceptance tests description

Scrum

INVEST

A good user story is:

- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimable
- **S**ized Appropriately
- **T**estable

Scrum

INVEST –Testable

- Tests concentrate on simple problems

Scrum team does not expect very specific complicated conditions.

World wide transaction system for an international bank

A fish trade company in Japan makes a payment to a vendor on Iceland. It should have been a payment in Icelandic Kronur, but it was done in Yen instead. The error is discovered after 9 days and the payment is revised and corrected, however, the interest calculation (value dating)…

From a talk by Hans Buwalda

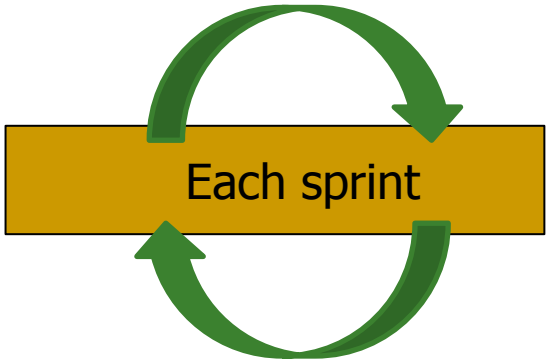
Scrum

Testing in Scrum

Testing in Scrum should be iterative.
Testers in Scrum must work without complete documentation.
Testers in Scrum should be **flexible**.

Scrum

Beginning of the project	Understanding of the project principles
Release planning	Stories estimation; questions: *„What happens, if…?"*
Sprint planning	Validation of **satisfaction conditions** , adding new ones



Creation and **testing in pairs**:
Developer and tester

- Write and execute tests for each story,
- Write and execute **functional** tests,
- Automate test,
- Make exploration tests.

Basic testers activities in Scrum

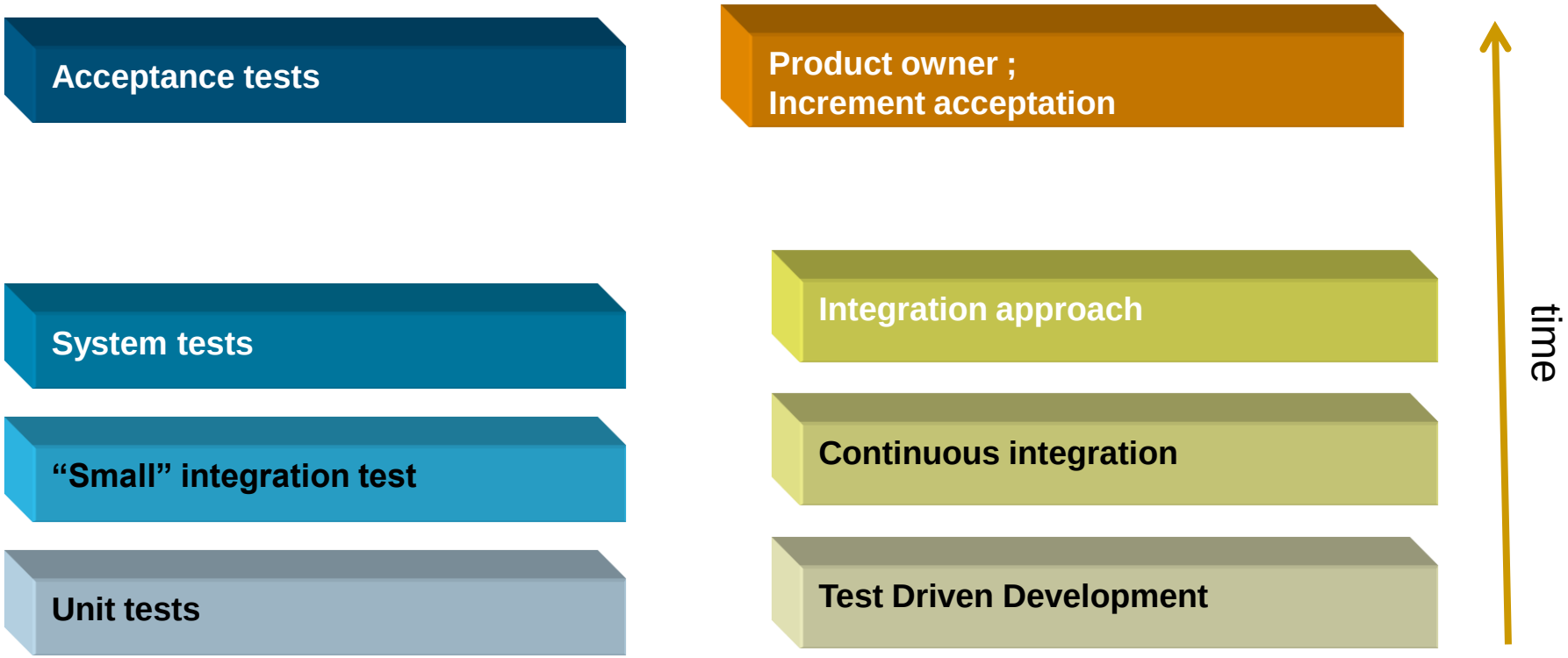
Scrum

Team complement
There is no division on developers and testers.
Testers should **be build in team.**

But:

- How many team members do concentrate on quality problems? (1/10, 2/10 ???);
- "Group thinking" – only positive;
- Not enough "business knowledge"
  - product owner problem.

Classic approach vs. Agile



Classic approach vs. Agile

Ken Beck[1] :

> *Programmers need their own tests to maintain speed and flexibility. These tests may improve quality enough that QA is no longer needed as a Great Wall to protect the customers against the depredations of the Mongrel Programmer Hordes*.

[1]www.soft.com/QualWeek/QW2001/papers/2Q.html

Classic approach vs. Agile

...

But .....

Module tests are limited in finding bugs:

Capers Jones[1] found that average effectiveness for unit test is between **25 - 30%**.

And Rex Black[2] (for American market) found that good system tests done by independent test team achieves **85%** effectiveness in founding bugs.

[1] *Capers Jones:* **MEASURING DEFECT POTENTIALS AND DEFECT REMOVAL EFFICIENCY http://www.rbcs-us.com/images/documents/Measuring-Defect-Potentials-and-Defect-Removal-Efficiency.pdf**

[2] **http://www.rbcs-us.com/images/documents/**

Solution

Solution:

Two levels of testing:

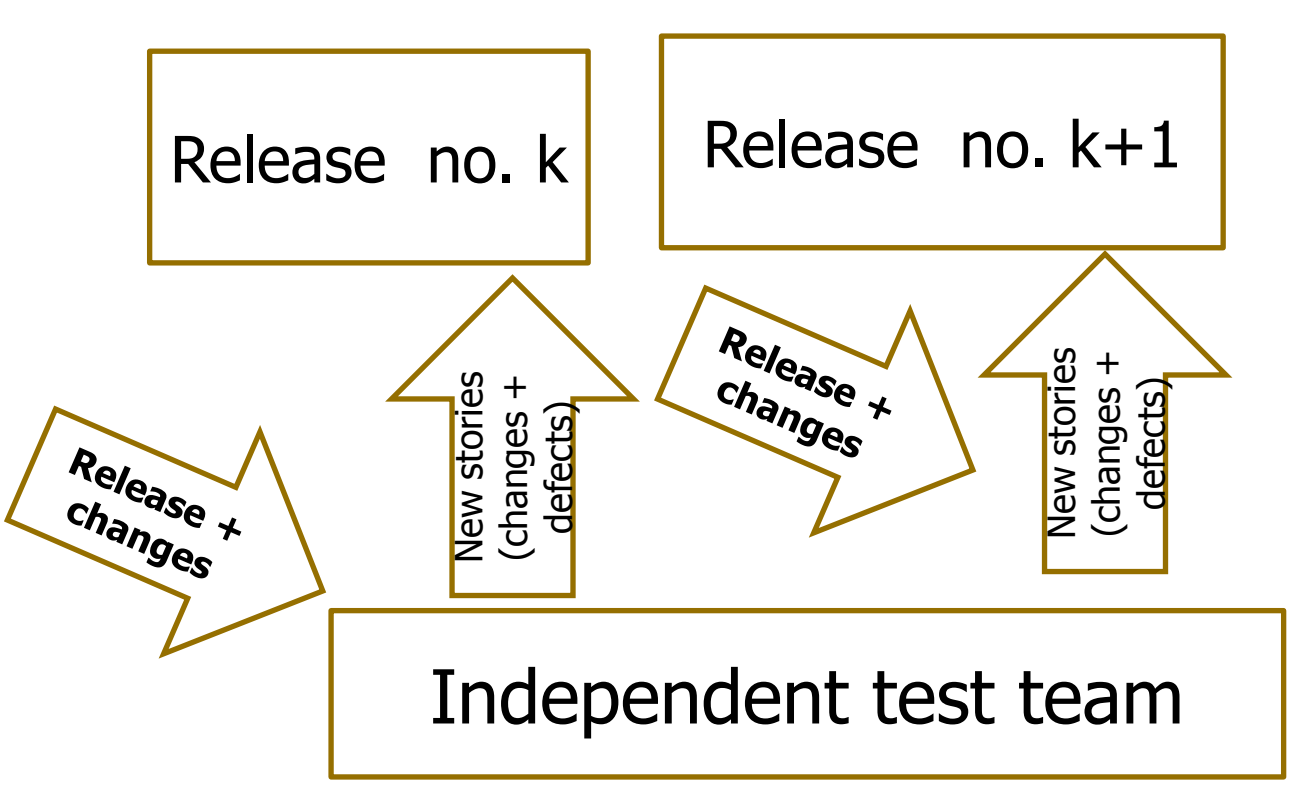
- Internal testing – in Agile team
- External testing – independent test team

Solution

Typical tasks of external testing:

- "business" acceptance tests.
- More complicated tests:
  - Nonfunctional tests.
- Ending game
  - User acceptance tests.

Solution

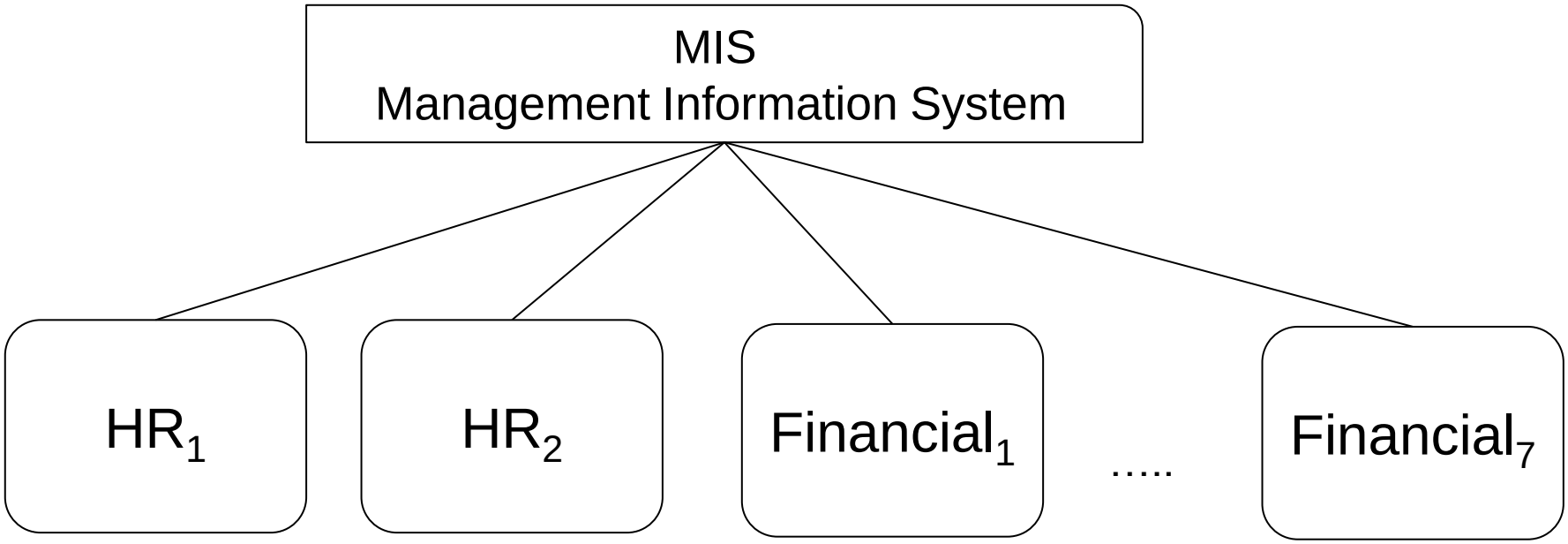


Internal tests
- TDD

External tests
- Acceptance
  - UAT
- Exploratory
- Nonfunctional
- Scenario based

3 examples

Example no.1



Examples

Example no.1

- Only f2f communication
  - No written requirements
    - Based on old existing systems
- Product owner
  - Concentrate on financial modules
- Tests
  - Only user acceptance
  - No written information about incidents

RESULT:

- 6 month delay
- Business problems
  - No working application

Examples

Example no. 2 NEW FUNCTIONALITY in bank system

- Only f2f communication
  - No written requirements
  - Frequent changes
  - Without info to test team
- Product owner
  - Less of business knowledge
  - Concentrate on security problems
- Tests
  - Long time before bugs closure

RESULT:

- disaster

Examples

Example no. 3 Telecom application connected with EURO 2012

- 90% written requirements
- Product owner
  - Concentrate on performance problems
- Tests
  - 2 performance testers in team
  - 60% time – performance tests

RESULT:

- In progress
- Deadline June 1st
- Success??

Thanks for your attention