



Agile Quality Management

Panagiotis Sfetsos, PhD
Assistant Professor,
Department of Informatics,
Alexander Technological Educational Institution
E - mail: sfetsos@it.teithe.gr
Web Page: <http://aetos.it.teithe.gr/~sfetsos/>



Objectives:

- ✓ Agile Manifesto Values - Agile Development
- ✓ What is quality?
- ✓ Quality Management Goals (ISO 9001:2008, ISO/IEC 9126, ISO/IEC 12207)
- ✓ Does Quality Management have a place in agile methods?
- ✓ Deming's Quality Improvement Cycles
- ✓ Best Agile practices
- ✓ Empirical findings on Quality in Agile Practices
- ✓ Human issues in Agile QM
- ✓ Key success factors



The emergence of the Agile Methods (1/2)

The Standish group, 1994:

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

The emergence of the Agile Methods (2/2)

The Standish group, 1994:

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
Other	13.9%


Total:

Success rate = 16.2%

Challenged projects = 52.7%

Impaired (cancelled) = 31.1%

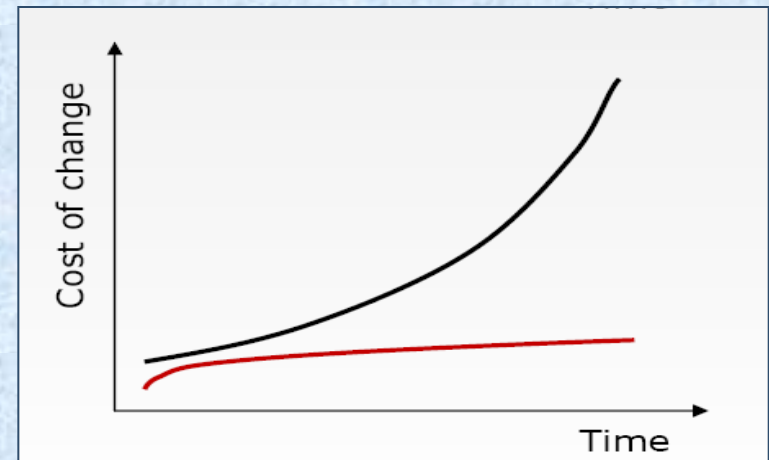
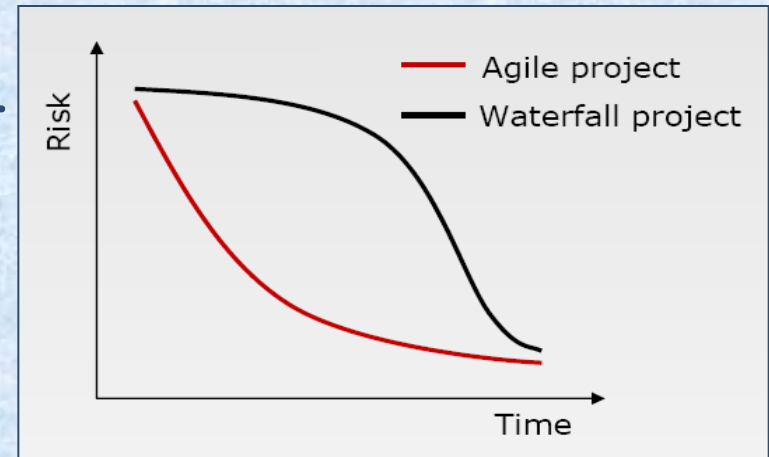
Agile Manifesto values in practice

- 
- A blurred background image showing a group of people in a meeting room. Several individuals are standing and talking, while others are seated at a table. The scene is brightly lit, possibly by a large window in the background, creating a collaborative and active atmosphere.
- ❖ Individuals and interactions **over** processes and tools
 - ❖ Working software **over** comprehensive documentation
 - ❖ Customer collaboration **over** contract negotiation
 - ❖ Responding to change **over** following a plan

Agile Software Development (1/3)

What does Agile development mean in practice?

- ❖ Iterative and incremental development
- ❖ “Fits just right” process
- ❖ Continuous testing and validation
- ❖ Simple planning and design
- ❖ Rapid response to change
- ❖ Close collaboration
- ❖ Empowered and self-organized teams
- ❖ Frequent delivery of working software
measure of progress



Agile Software Development (2/3)

Agile means re-evaluate your best practices in a rapid fashion...

- ❖ Test Driven Development (TDD) or Test First Development + Refactoring
- ❖ Continuous Integration
- ❖ Customer participation
- ❖ Pair Programming
- ❖ Shared Code Ownership

Agile Software Development (3/3)

	Agile Methods	"Traditional" Methods
Project size	Small, Medium	Large
Approach	Adaptive	Predictive
Success Measurement	Business Value	Conformation to plan
Management Style	Decentralized	Autocratic
Perspective to Change	Change Adaptability	Change Sustainability
Culture	Leadership-Collaboration	Command-Control
Documentation	Low	Heavy
Emphasis	People-Oriented	Process-Oriented
Cycles	Numerous	Limited
Domain	Unpredictable/Exploratory Predictable	Predictable
Upfront Planning	Minimal	Comprehensive
Team Size	Small/Creative	Large
Quality	Built-in from the beginning	Split in the devel. phases
Quality Assurance	Upstream in the SDP	End Product
Return on Investment	Early in Project	End of Project

What is Quality?

- × The totality of characteristics of a product or service that bear on its ability to satisfy stated and implied needs (ISO 8402)
- × The degree to which a system, component, or process meets specified requirements and customer/user needs or expectations (IEEE)
- × Both definitions are focused on **satisfying the customer's need for the software product**



Why is Quality Important?

❖ Customers expect it

- Focus on the Customer (expectations, requirements)

❖ We need to get better to compete

- Teamwork (Working together to reach goals and solve problems)
- Continuous Improvement

❖ Need to be able to measure our progress

- Measuring Success



Quality Management Goals (1/3)

The 8 principles of **ISO 9001:2000**...

- ❖ Customer satisfaction
- ❖ Leadership
- ❖ Involvement of People
- ❖ Process approach
- ❖ Systemic approach
- ❖ Continuous improvement
- ❖ Decisions based on facts
- ❖ Relationship with suppliers mutually beneficiary



Quality Management Goals (2/3)

The ISO/IEC 12207

Agile Methods and the **Development Process Area...**

Planning game or sprint planning



Requirements definition activities

Test Driven Development, Continuous Integration,
Customer involvement, Pair Programming



Testing and Implementation activities



Quality Management Goals (3/3)

The ISO/IEC 9126

Software Product in the Agile Methods...

Quality Model

External
Metrics

Internal
Metrics

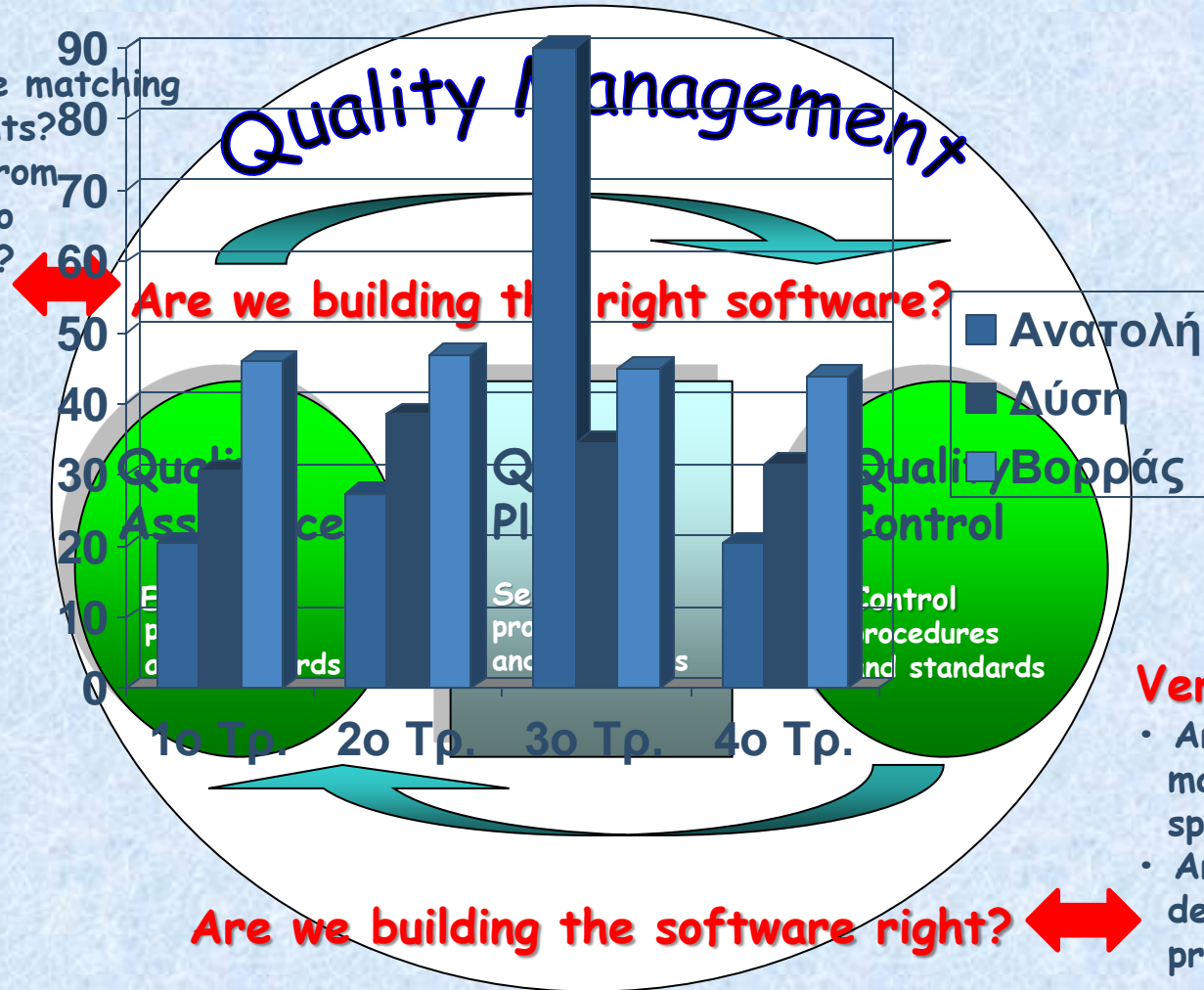
Quality in
Use Metrics



Does Quality Management, as in "traditional", have a place in agile methods?

Validation

- Is the software matching the requirements?
- Can we trace from requirements to implementation?



Deming's Quality Improvement Cycles (1/2)

PLAN

-**Establish** the objectives and **processes** necessary to deliver results in accordance with the expected output.

DO

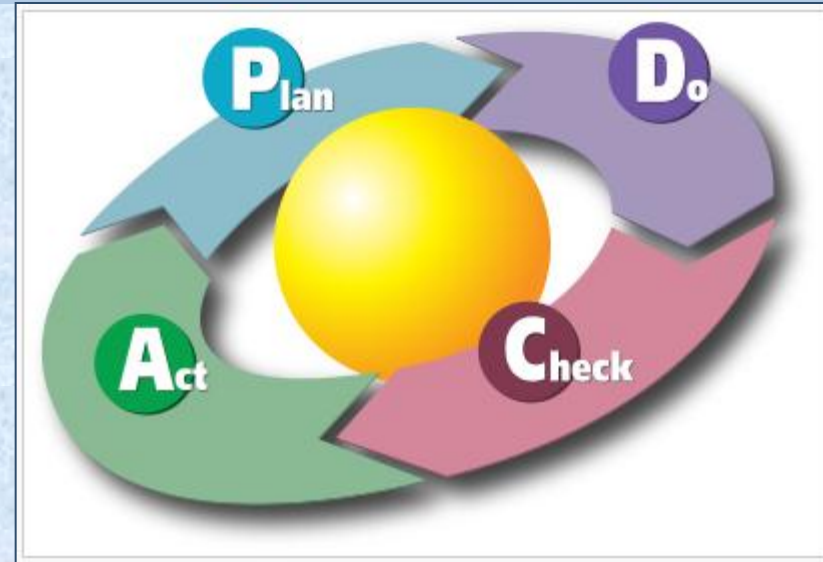
-**Implement** the new **processes**. Often on a small scale if possible.

CHECK / STUDY

-**Measure** the new **processes** and compare the results against the expected results to ascertain any differences.

ACT

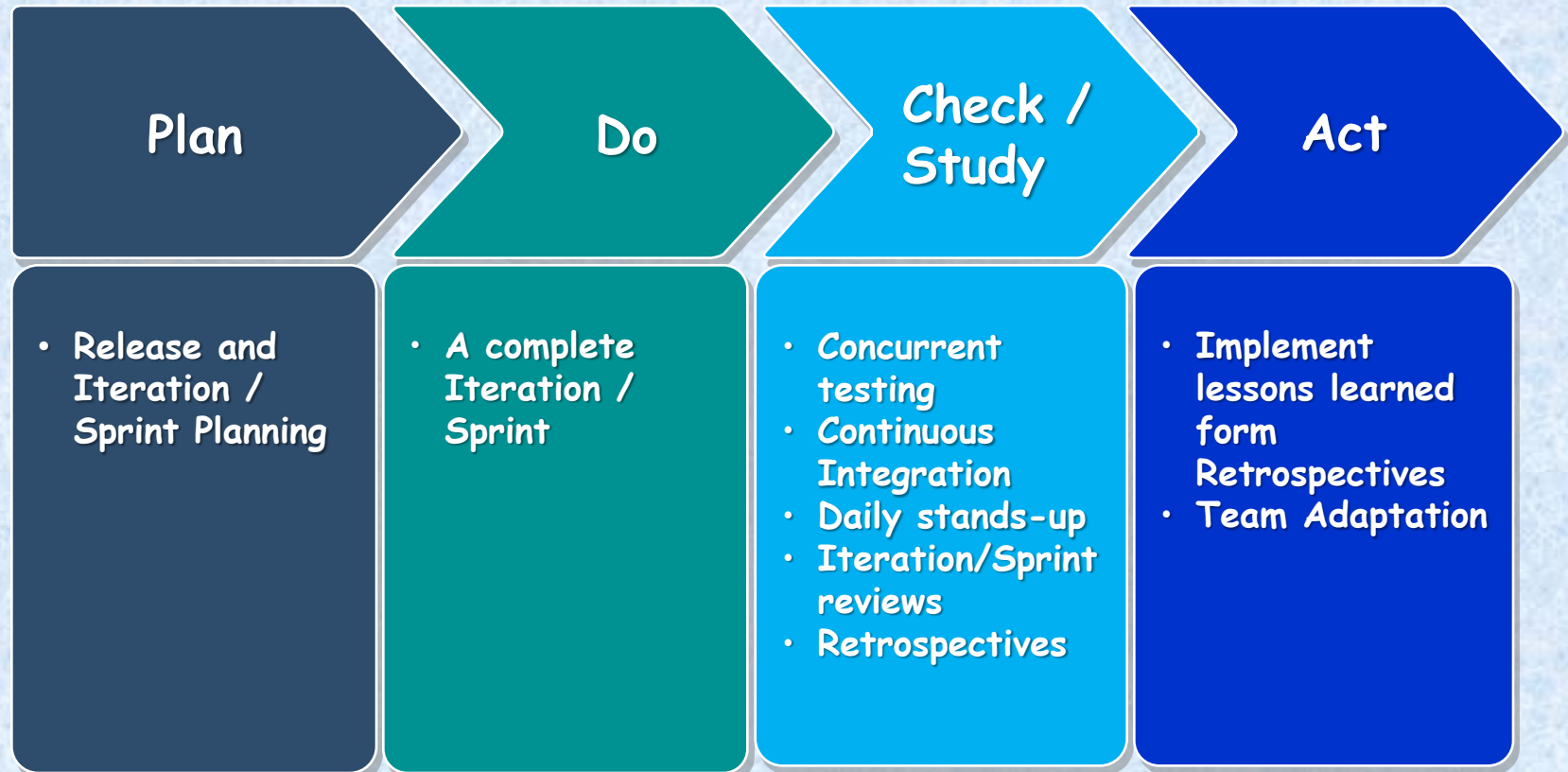
-Analyze the differences to determine their cause. Determine where to apply changes that will include **improvement**.



Deming's Quality Improvement Cycles (2/2)

The Agile version..

Iterative and Incremental Quality Improvement Cycles



Agile Quality Assurance

QA in the Agile Software Development Life Cycle..

	Techniques / Practices
Requirement & Design QA	<ul style="list-style-type: none">- Simple planning and design- User Stories- Acceptance Tests and UAT- Release/Iteration planning- Architectural spike- System Metaphor- On-site customer feedback
Implementation and Code QA	<ul style="list-style-type: none">- Unit tests- Refactoring- Acceptance Tests and UAT- CRC-cards- Pair Programming
Integration and System QA	<ul style="list-style-type: none">- Unit tests coverage- Acceptance Tests and UAT- Continuous Integration- Common Codebase- Customer Feedback

"Traditional" vs. Agile (1/2)

QA - key differences...

- ❖ On-time system delivery, to spec, and within budget and schedule

A working product increment

- ❖ QA team and PM responsible for "quality"

"Whole team" approach - "Built-in"

- ❖ QA team engaged in the testing phase (...end of product)

Development team (testers included), upstream in the SDP



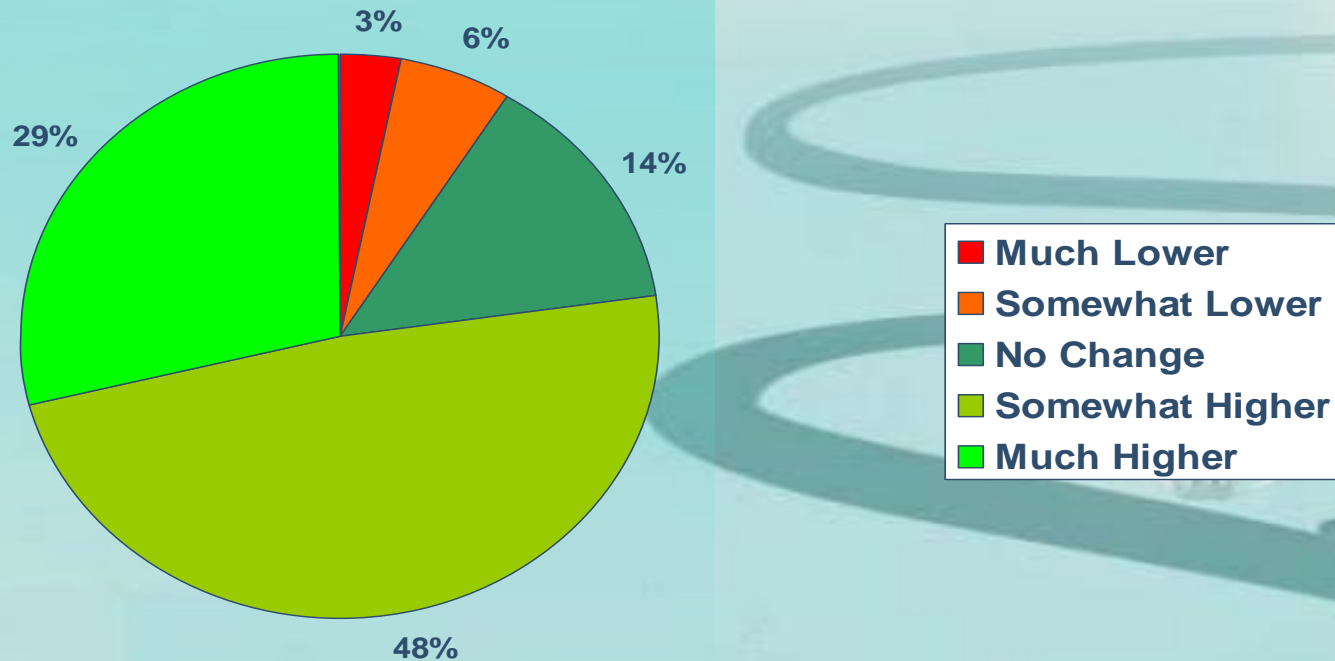
"Traditional" vs. Agile (2/2)

Differences in Quality

- | | |
|--|---|
| <ul style="list-style-type: none">❖ Split quality (<i>in each development phase</i>)❖ QA team responsible for "quality"<ul style="list-style-type: none">- Formal methods- Quality Police- No specs, can't test❖ Quality Assurance - end product | <ul style="list-style-type: none">❖ Built-in quality (<i>from the beginning</i>)❖ "whole team" approach<ul style="list-style-type: none">- Self-directed team responsible for quality, testers included- Pair/team collaboration,- Day-to-day activities❖ Quality Assurance - upstream in the SDP (<i>from the beginning</i>) |
|--|---|

Agile Techniques (1/2)

How Have Agile Approaches Affected the Quality of Systems Deployed?

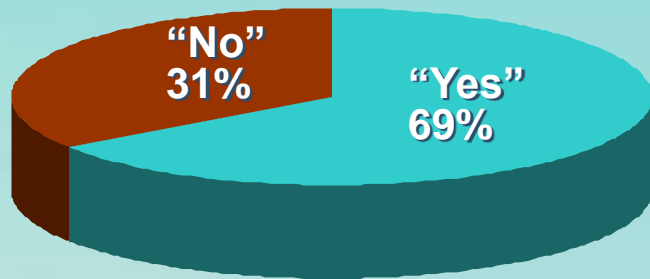


Source: Ambler 'Agile Adoption Rate Survey' of over 4200 Dr. Dobb's subscribers, March 2008

Agile Techniques (2/2)

Third-party research suggests even wider adoption..

Have you adopted any Agile techniques?



Source: Ambler 'Agile Adoption Rate Survey' of over 4200 Dr. Dobb's subscribers, March 2008

In a recent study conducted by the Agile Journal, it was determined that **88%** of companies, many with over 10,000 employees, are using or evaluating Agile practices on their projects.

Which of the agile practices should be implemented to improve software quality?



Test Driven Development (1/4)

What is TDD?

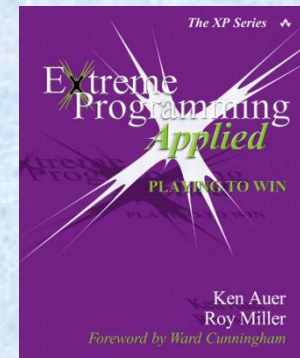
TDD isn't about testing...

...it's about programming! (specification, design, testing and coding)

Writing simpler, clearer and more robust
and more maintainable code!

"Before you write code, think about what it will do. Write a test
that will use the methods you haven't even written yet."

Extreme Programming Applied: Playing To Win
Ken Auer, Roy Miller
"The Purple Book"



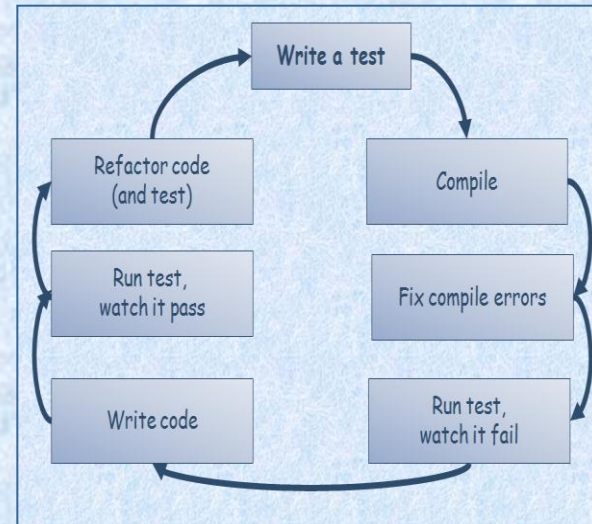
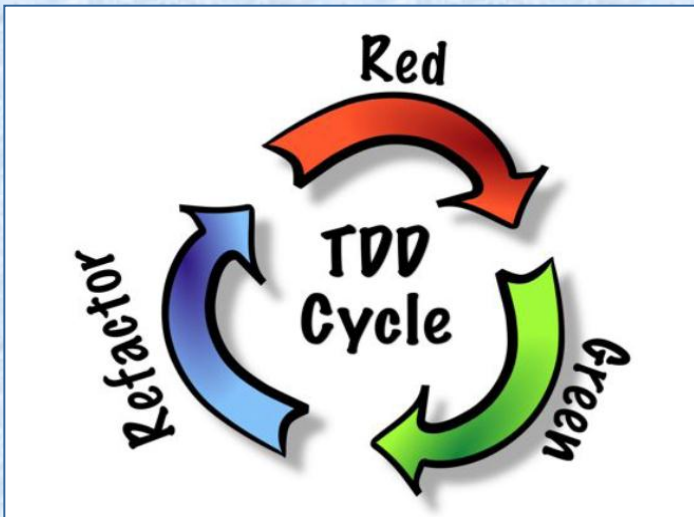
No code should go into production unless it has associated tests...
...Catch bugs before they are shipped to your customer

Test First Development + Refactoring (2/4)

A test is not something you “do”, it is something you “write” and run once, twice, etc.

- It is a piece of code
- Testing is therefore “automated”
- Repeatedly executed, even after small changes

The cycle of **Red** – **Green** – **Refactor**



Test Driven Development (3/4)

Testing in all phases and levels...

- Unit testing
- Acceptance testing
- Integration testing
- Regression testing
- Functional testing
- System testing

Empirical findings

- The defects were decreased from 5% - 45%

Empirical Studies on Quality in Agile Practices: A Systematic Literature Review, P.Sfetsos, I. Stamelos. In QUATIC'2010.

Test Driven Development (4/4)

TDD benefits:

- shortens the programming feedback loop
- provides detailed specification (tests)
- improves quality (design and code)
- provides concrete evidence that your software works
- ensures that your design is clean
- provides an interface before algorithm
- supports evolutionary development
- encapsulates learning
- enhances intermediate stability
- enhances confidence in code changes

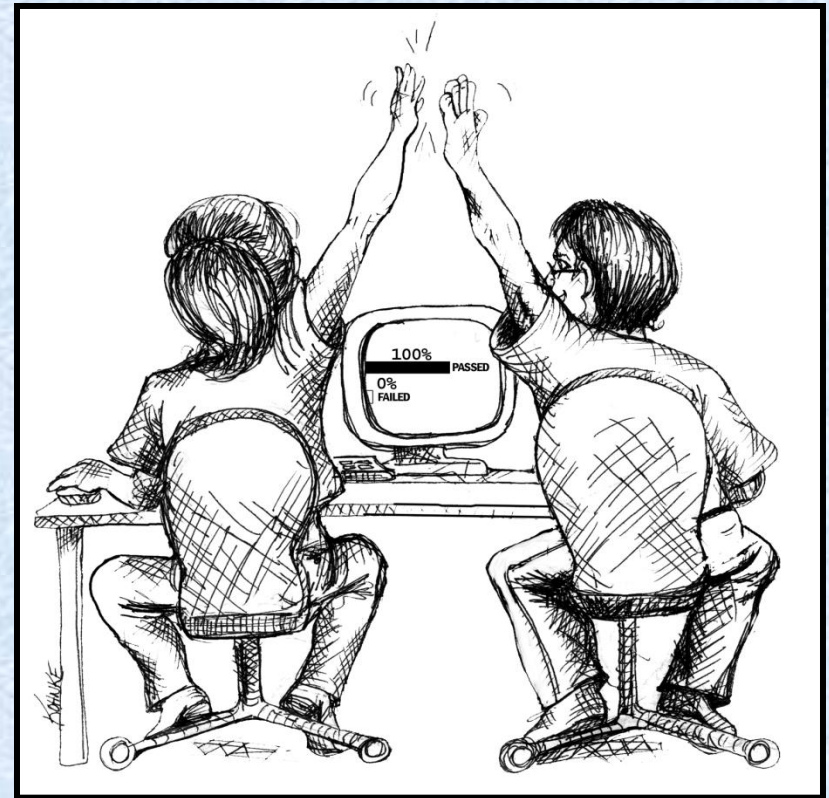
Pair Programming

Continual design and code review process

- Reduction of the defects
- Improvement of design and code quality
- Higher product quality
- Faster cycle time
- Enhanced learning - Knowledge transfer
- Enhanced trust/teamwork
- More communication
- Collective code ownership

Empirical findings

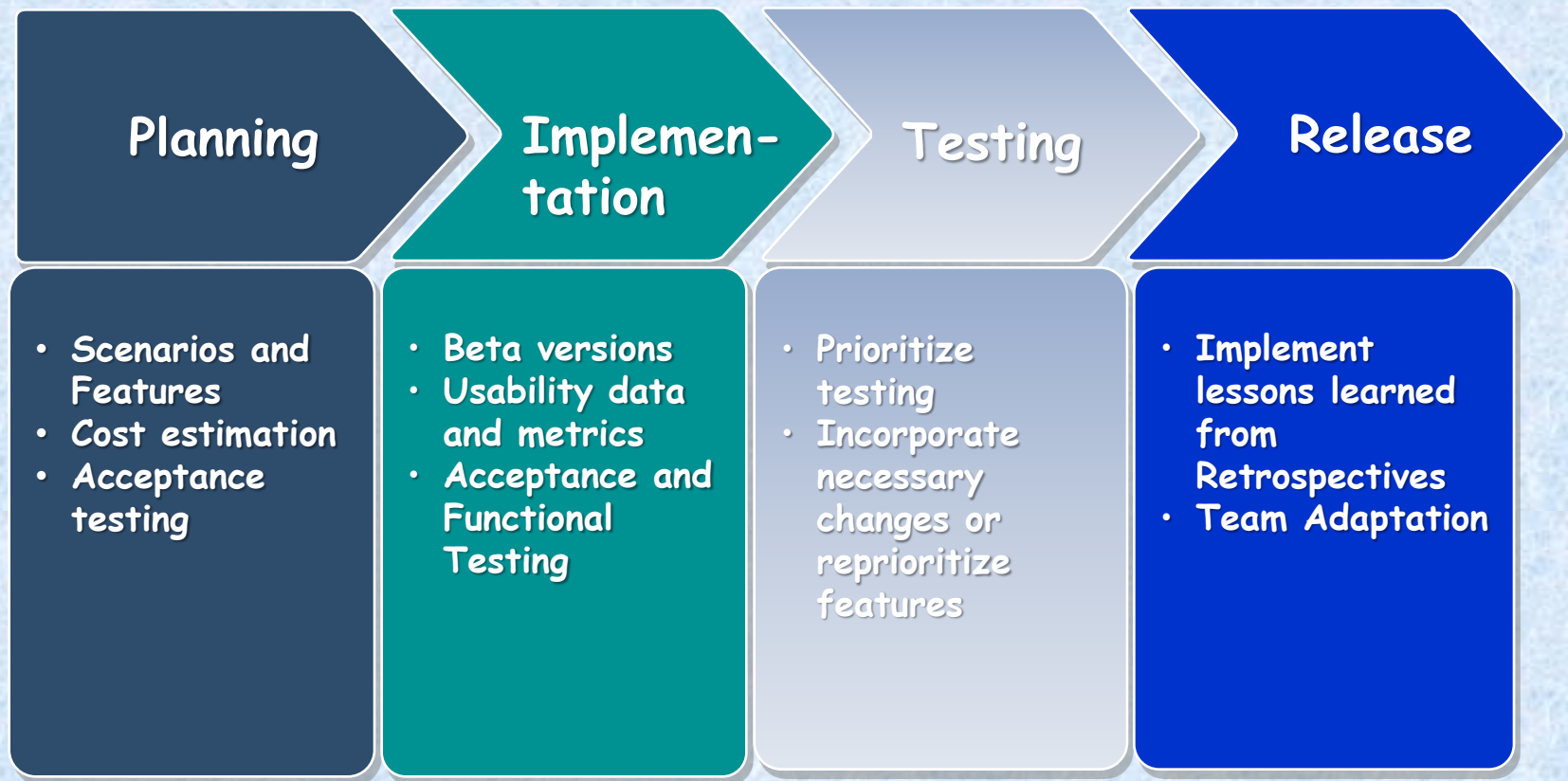
- Design and code quality improvement
varying from 15% up to 65%



Empirical Studies on Quality in Agile Practices: A Systematic Literature Review,
P.Sfetsos, I. Stamelos. In QUATIC'2010.

Customer Interaction and feedback

Customer involvement in the development phases...



Empirical Findings

A systematic literature review, evaluating quality approaches and metrics, in Agile practices, according to ISO/IEC 12207 and ISO/IEC 9126 standards:

TDD

- The defects were decreased from 5% - 45%
- Improvement of external quality (TFD + Refactoring)
- Decreased fault rates in Industry (than in Academia)

Pair Programming

- Design and code quality improvement varying from 15% up to 65%

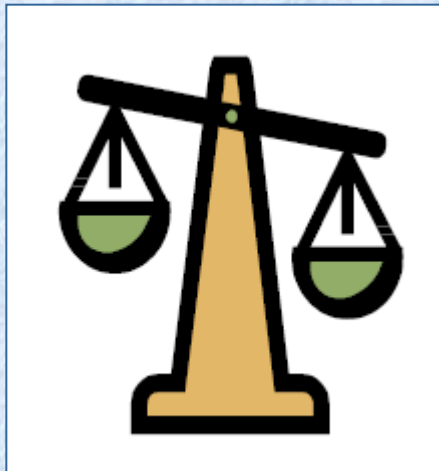
Other Practices

- Quality improvement for combined practices including Planning Game or Sprint Planning, Refactoring and On-site Customer.

Empirical Studies on Quality in Agile Practices: A Systematic Literature Review, P.Sfetsos, I. Stamelos. In QUATIC'2010.

Human issues in Agile QM

- Teams as **adaptive ecosystems** (roles and tasks depend on developers)
- **Communication** and **Collaboration** (most critical success factors)
- Human factors must be identified, understood, controlled, predicted or manipulated.
- Organizations and managers must capitalize on developers potential talents and strengths



◇ **Values**

◇ **People**

◇ **Teams**

◇ **Culture**

Key success factors (1/5)

Enhance:

× Agile Quality Planning & Execution

- “just enough, just-in-time (JIT)”
- **Planning levels (Quality criteria):** for User stories, Iteration and Release planning, End product
- **Visibility:** Release tracking, iteration tracking, and story tracking

× Code quality

- Test-Driven Development (TDD)
- Refactoring to improve existing design
- Coding with design patterns
- Pair programming
- Shared code responsibilities



Key success factors (2/5)

Enhance:

× Test-Driven Development, Not Defect-Driven

- TDD is a skill
- Efficiently and effectively testing
- Pairing and shared testing code responsibilities, refactoring, attention to design
- Add more tests if they add value
- Just Do It - Insist on achieving 100% coverage
- Measure and Review
- Make Results Visible

× Customer & User perspective

- The customer must be always available
- Business people and the project team must work together
- User involvement is necessary



Key success factors (3/5)

Enhance:

× Continuous Integration (maintaining Quality all the time)

- Test Driven Integrations
- Single source repository
- Frequent automated builds and tests



× Agile Verification and Validation

Validation (Are we building the right product?)

- Acceptance Test-Driven Development (ATDD): to help define the design and code
- All User Acceptance Tests should be passing (accept the story, the iteration and the release - quality criteria).

Verification (Are we building the product correctly?)

- A quality development process
- Unit testing verifies at the code level
- Functional testing (and other tests) verifies product at higher level
- Frequent automated builds and tests

Key success factors (4/5)

Enhance:

× Team building

- Self Organizing vs. role or title based
- Empowered to make decisions vs. decisions by outside authority
- Decisions to be consensus-driven vs. leader-driven
- Committed to success as a team vs. success at any cost
- Motivated by trust vs. fear or anger
- Maintain constructive disagreement vs. damaging conflict



× Agile metrics

- Discuss the importance of quality with the team
- Measure results at the team level
- Identify simple metrics that can be collected early
- Track the value delivered to customer or business
- Track only as long as it adds value
- Ensuring that quality is not treated as an event, but has become a way for life for the team

Key success factors (5/5)

Agile Quality Assurance People must:

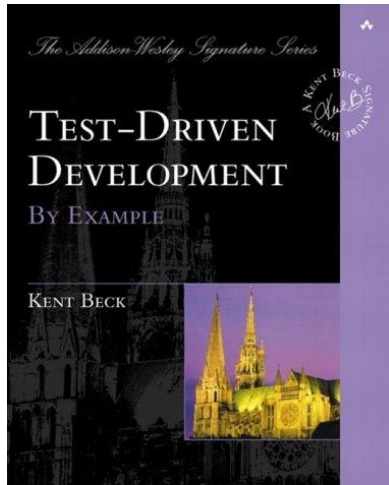
- Write automated specifications
- Collaborate with developers
- Continuously run all Acceptance Tests
- Be at the front of the Test Driven Development



Agile leadership

- Fix values and culture
- Embrace change - adjust methods to the projects
- Create best teams
- Influence team decisions - set movable boundaries
- Continuously reflect, inspect, and adapt

Books (1/2)

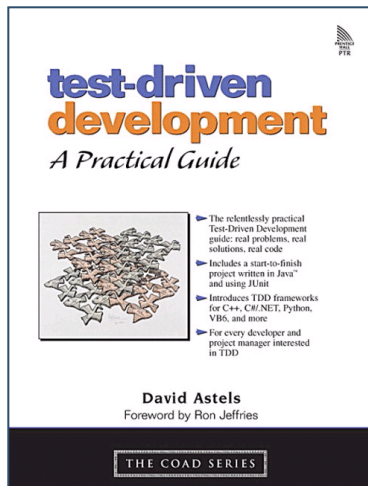


Test-Driven Development: By Example

Kent Beck

Addison-Wesley, 2003

ISBN 0-321-14653-0



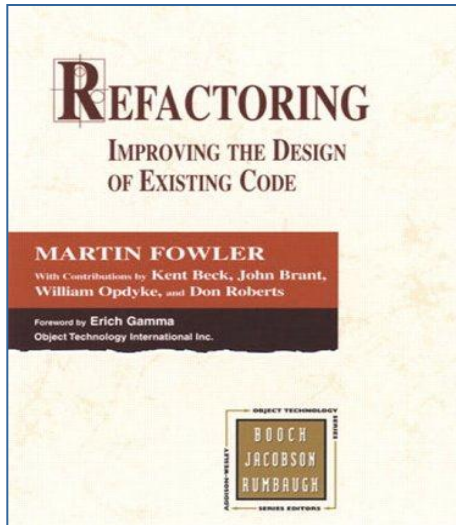
test-driven development: A Practical Guide

Dave Astels

Prentice-Hall/Pearson Education, 2003

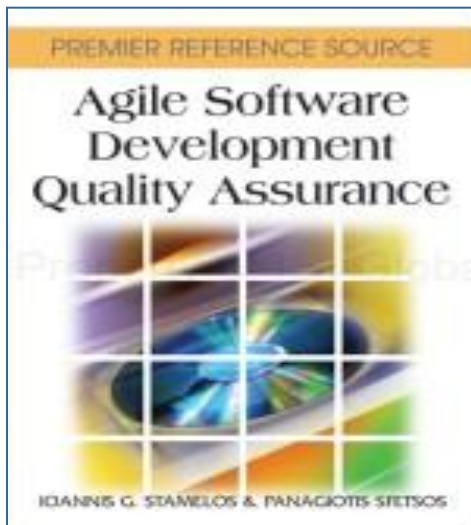
ISBN 0-13-101649-0

Books (2/2)



Refactoring: Improving the Design of Existing Code

Martin Fowler
Addison-Wesley, 1999
ISBN 0-201-48567-2



Agile Software Development Quality Assurance

Author(s)/Editor(s): I. Stamelos, P. Sfetsos
IGI Global, 2007
ISBN13: 9781599042169
ISBN10: 1599042169

The Quality Puzzle...



Thank you for your attention !