# Agile Quality Management

# Agile Manifesto Values

| | | |
|---|---|---|
| individuals and interactions | over | processes and tools |
| working software | over | comprehensive documentation |
| customer collaboration | over | contract negotiation |
| responding to change | over | following a plan |

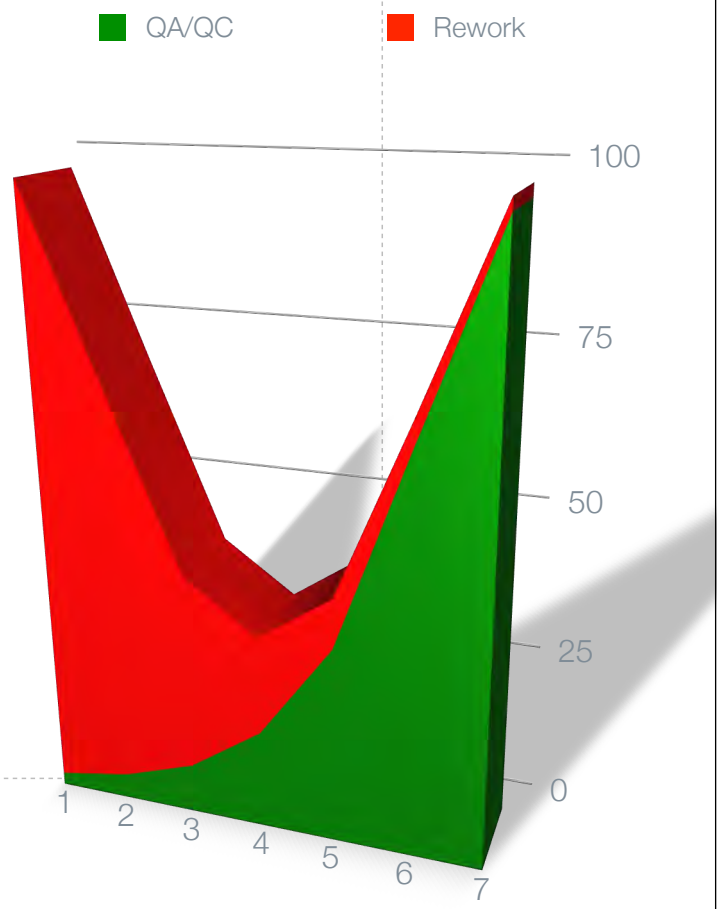
# Quality Management Goals

- Customer Satisfaction
- Leadership
- Involvement of People
- Process Approach
- System Approach to Management
- Continual Improvement
- Factual Approach to Decision Making
- Mutually Beneficial Supplier Relationships

ISO 9001:2000

# Quality is…

- The totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs (ISO 8402)
- when the customer returns – and not the product… (corporate slogan)
- free (Philip B. Crosby)



# What has Changed?

| Traditional Wisdom | Agile Perspectives |
|---|---|
| Strict change management | Change is inevitable |
| Comprehensive documentation | Working software is more important. And face-to-face communication is even better |
| Up front planning | Plan to the next iteration |
| Formal entrance and exit criteria with signoffs | Collaborate, don't hand-off |
| Comprehensive system-level regression tests | Detect defects earlier with automated unit tests and continuous integration |

Elisabeth Hendrickson: Agility For Testers

# The Rhythm of Agile Projects

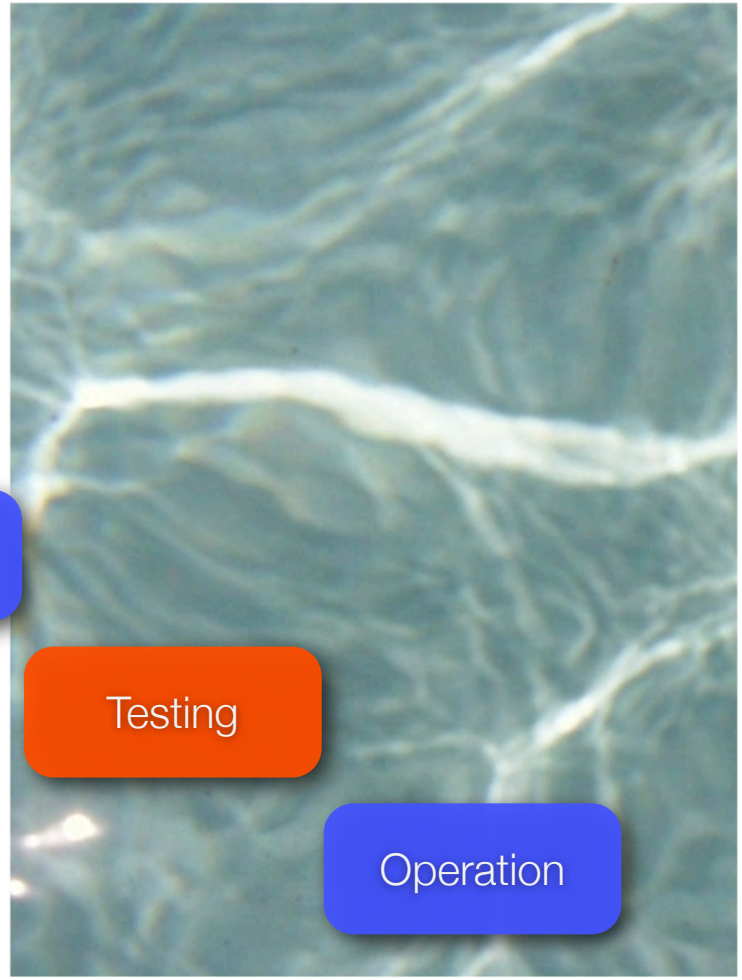
# Down the Waterfall?

Requirements

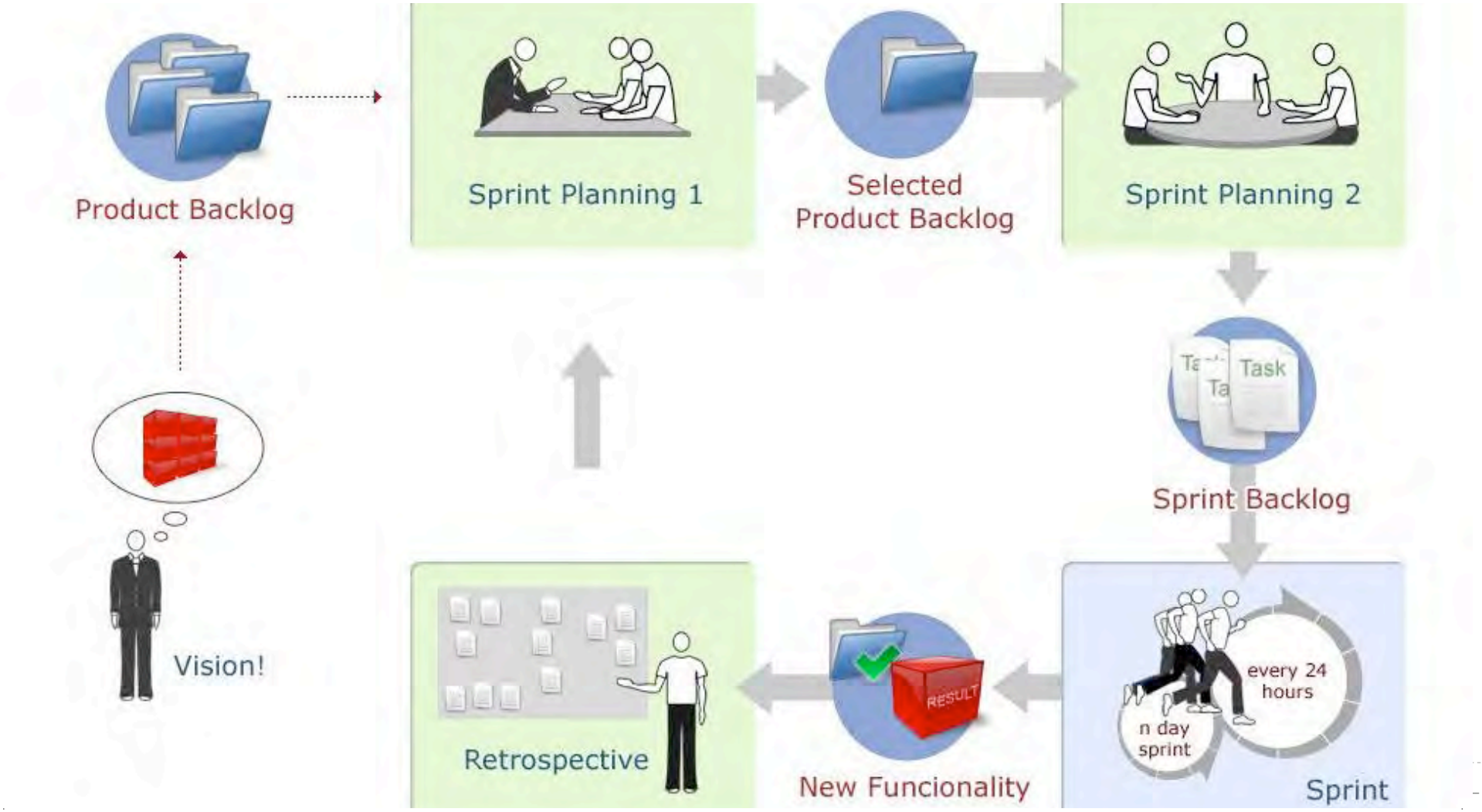
Analysis

Design

Coding



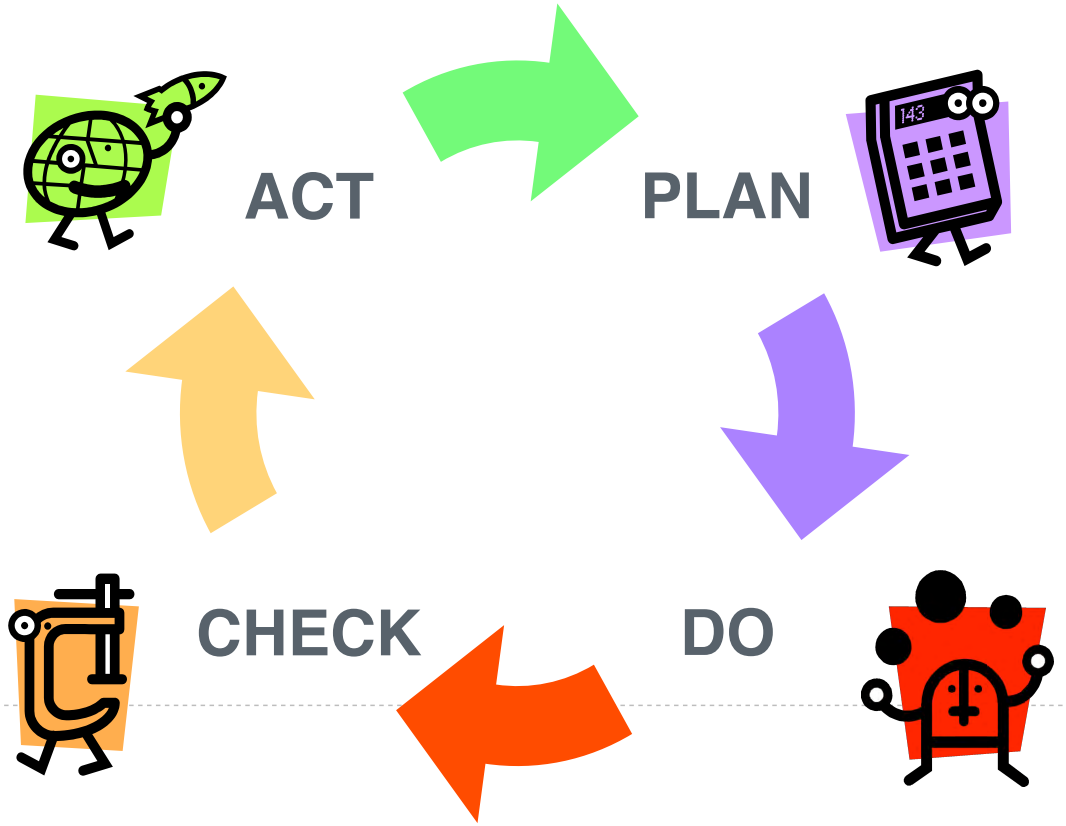
# Scrum Flow


Product Backlog
Sprint Planning 1
Selected
Product Backlog
Sprint Planning 2
Task
Sprint Backlog
every 24
hours
n day
sprint
Sprint
RESULT
New Funcionality
Retrospective
Vision!

# Agile Development



# The Deming Cycle

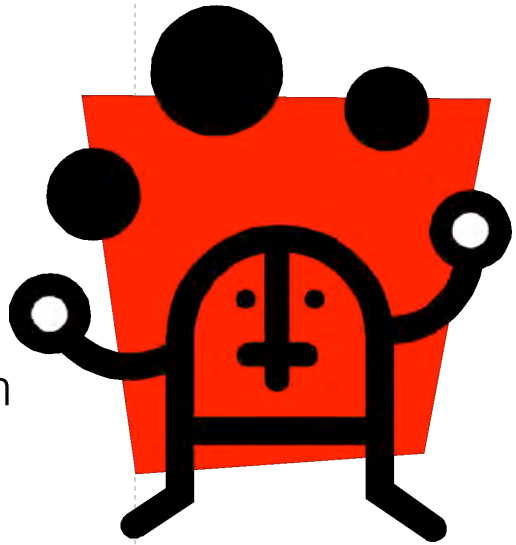


# ***Plan*** for Quality

- What is your definition of quality?
- How do you know you have reached it?
- What has to be tested when?
- How can the tests be performed?

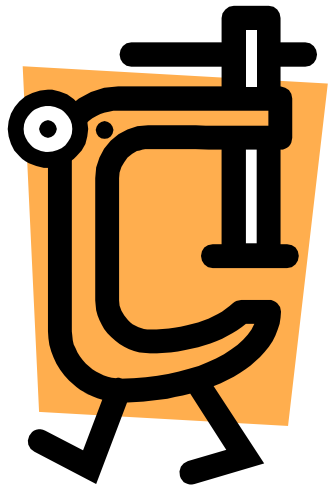
# ***Do*** the Thing Right

- Quality awareness
- Customer Collaboration
- S.M.A.R.T requirements
- User acceptance tests vs. system tests
- Continuous Integration
- Test Driven Development
- Coding Standards



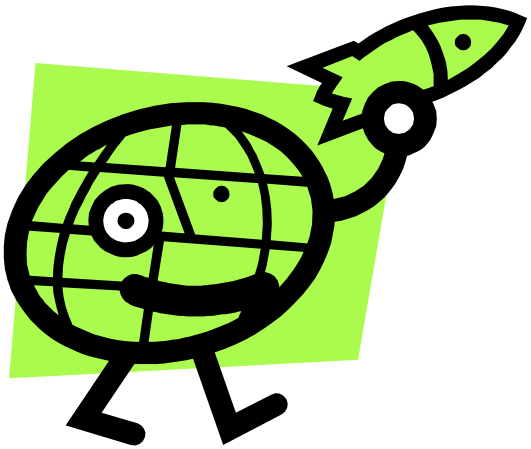
# ***Check*** Where You Are

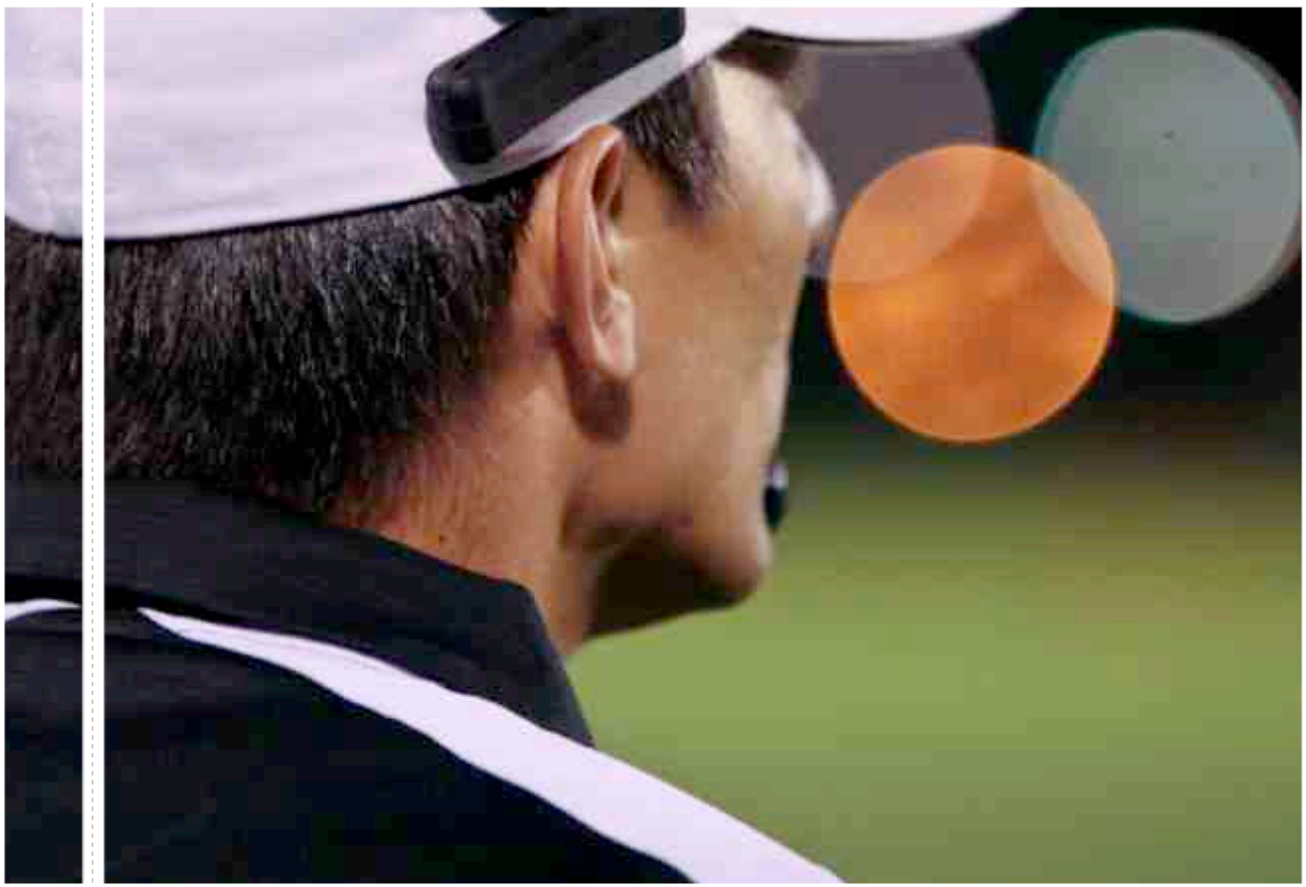
- Test Results
- Code Quality
- Progress
- Team Capability
- Team Velocity
- Customer Satisfaction
- Yourself



# *Act* Immediately

- Establish process framework
- Teach quality consciousness
- Enhance communication
- Reduce defect count
- Institutionalize Change





# Quality Coaching

# ScrumMaster vs Quality Manager

| ScrumMaster | Quality Manager |
|---|---|
| takes care about quality | takes care about quality |
| facilitates process | facilitates process |
| result oriented | result oriented |
| protects the team | protects the quality |
| solve impediments | observe issue management |
| increase performance | decrease defects |
| guide and coach | observer and controller |

# First steps for Agile QM

- Learn agile methods
- Identify problems
- Establish practices
- Inspect and adapt

# Process Improvement

# Retrospectives

- What went well and should be kept?
- What went wrong and should be dropped?
- What did we learn?
- What do we have to analyze?
- What do we have to discuss?

Norm Kerth, Project Retrospectives

# Impediment Backlog

- Team Level
- Group / Program Level
- Organizational Level

# Requirements, Estimations and Acceptance Criteria

# Agile Specification

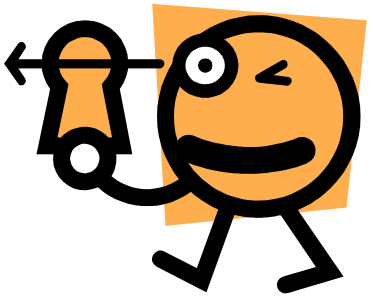
- An agile specification is a prioritized list of …
  - User Stories
  - Use Cases
  - Feature Description
  - Bugs...
- In Scrum known as Product Backlog
- The Product Backlog is a living document

# Non Functional Requirements

- Non functional requirements can be
- … defined in the product backlog
- … defined in the project vision
- … part of the company standards

# S.M.A.R.T.

- Specific
- Measurable
- Achievable
- Relevant
- Timed



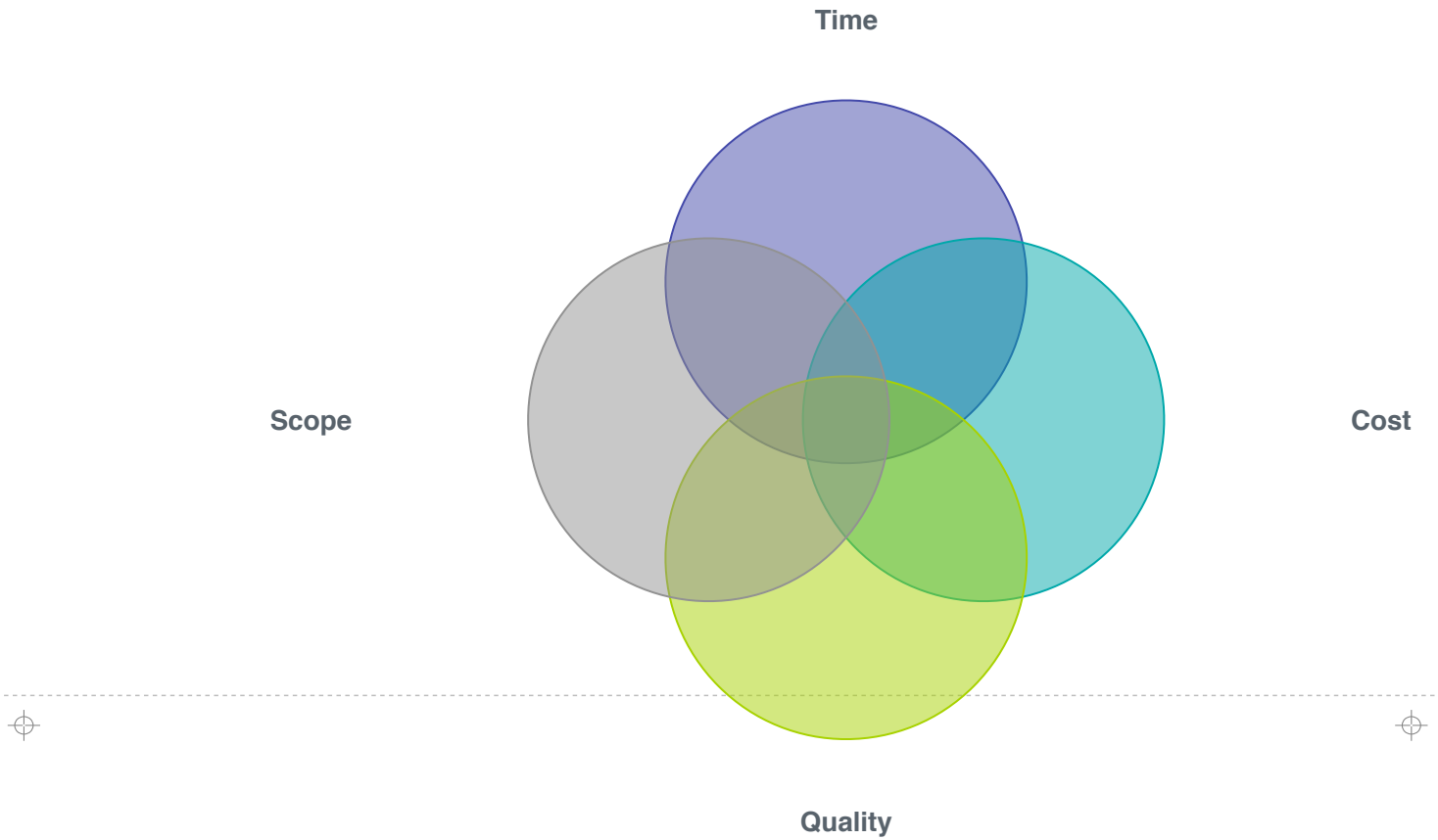
# User Stories

… promise for a future conversation.

# Acceptance Tests

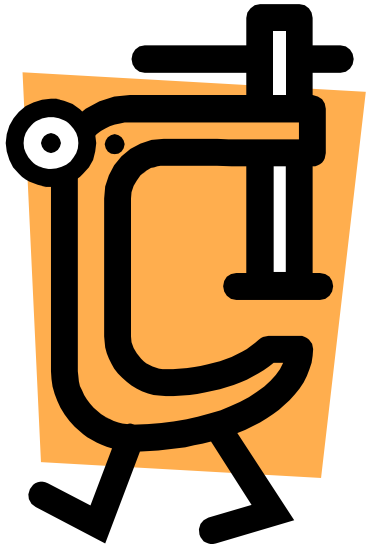
- Owned by customer / Product Owner
- Written together with the customer / user, Product Owner, developer and tester
- About the what and not the how
- Expressed in the language of the problem domain
- S.M.A.R.T. …

# Metric Dimensions



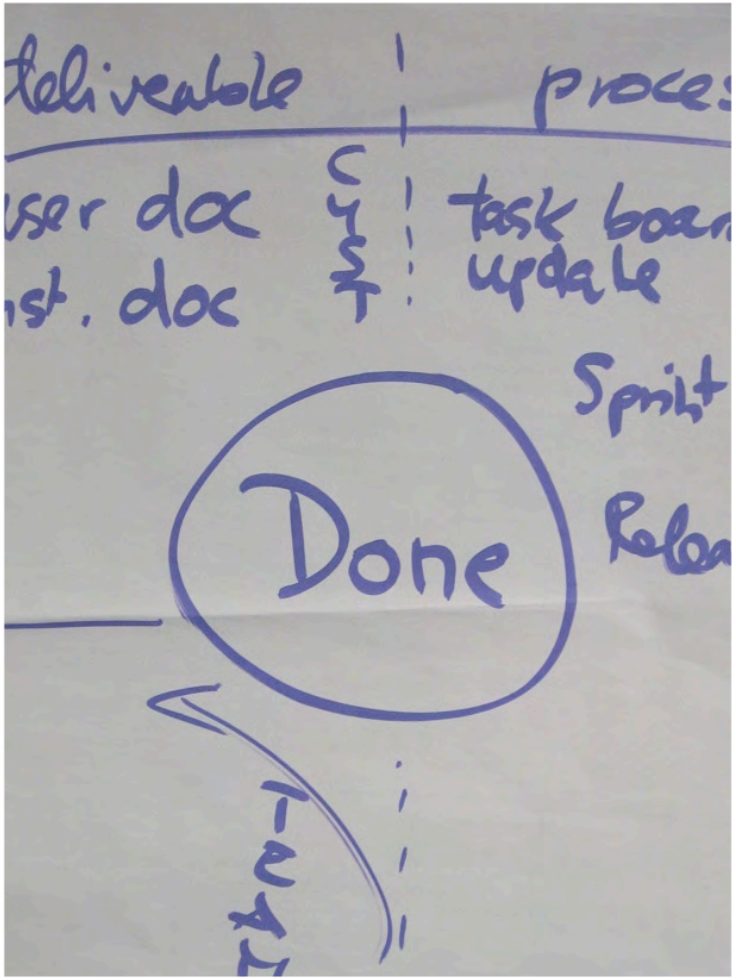
# Basic AQM metric toolbox

- Business Value
  - –Build up charts
  - –Non functional BV
- Process Quality
  - –Acceptance %
  - –Velocity
- Engineering Quality
  - –Code Metrics

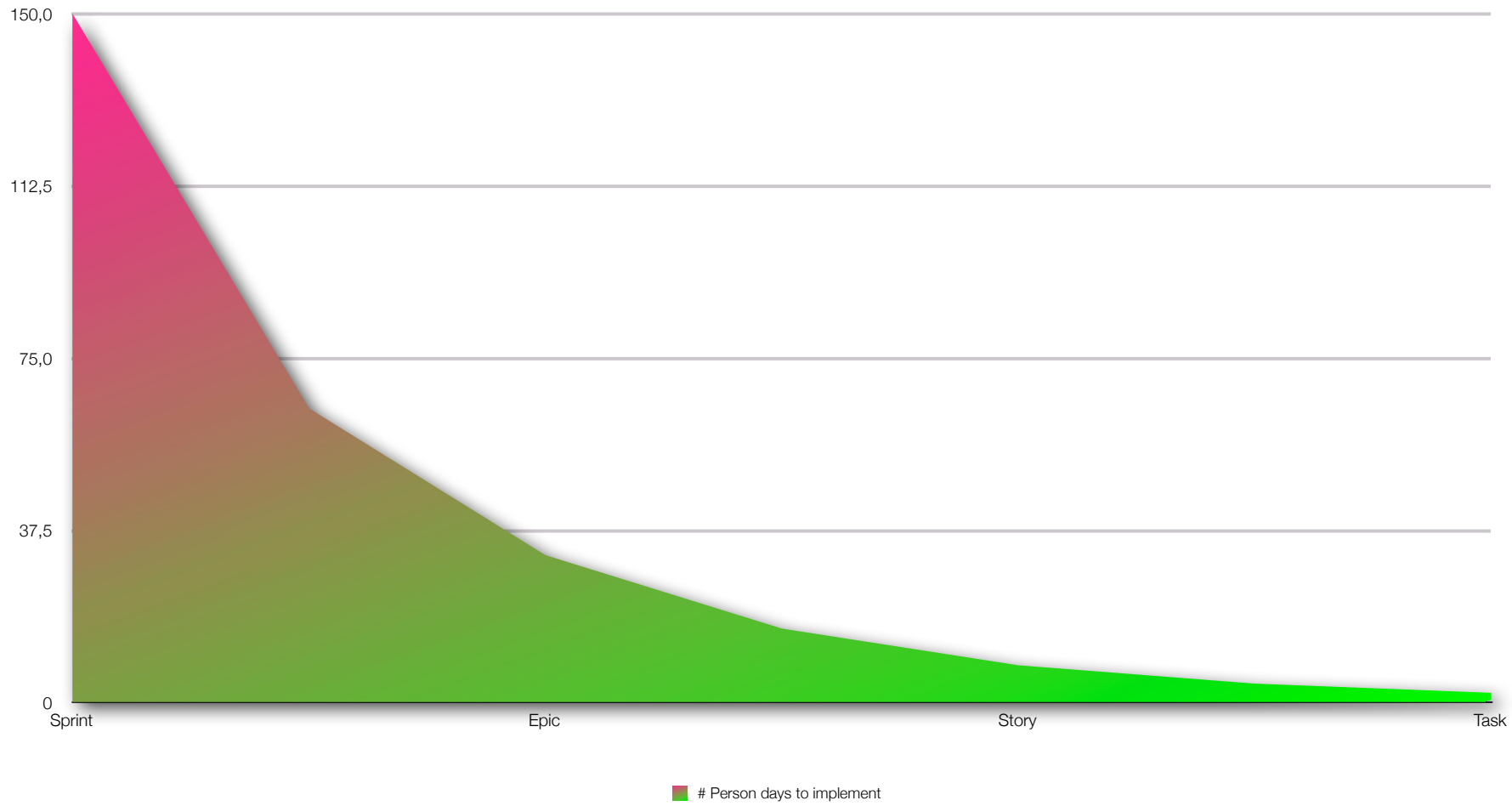


# Potentially shippable

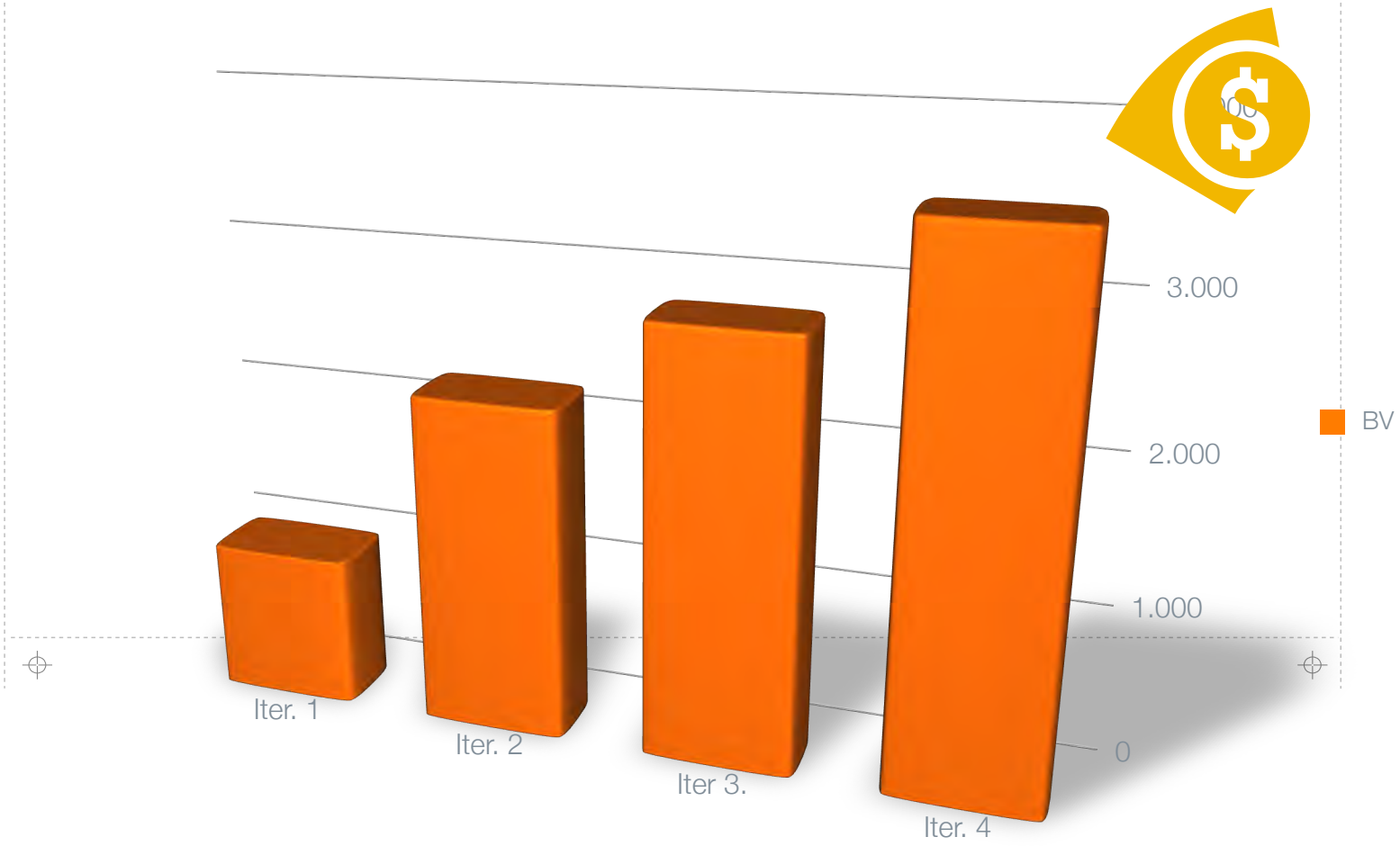
- Developed
- Built
- Integrated
- Tested
- Documented
- ?



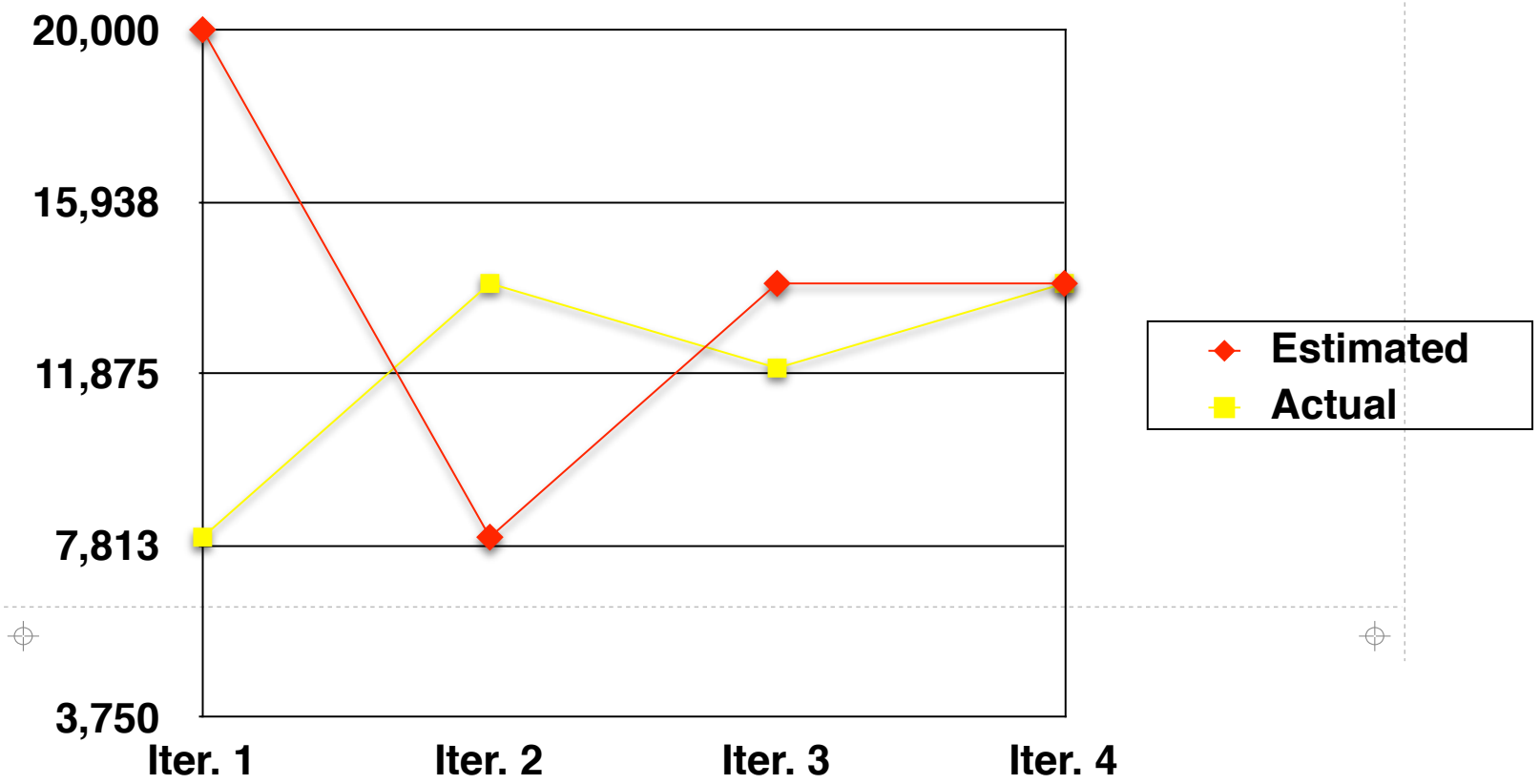
# Story Size


150,0
112,5
75,0
37,5
0
Sprint
Epic
Story
Task
# Person days to implement

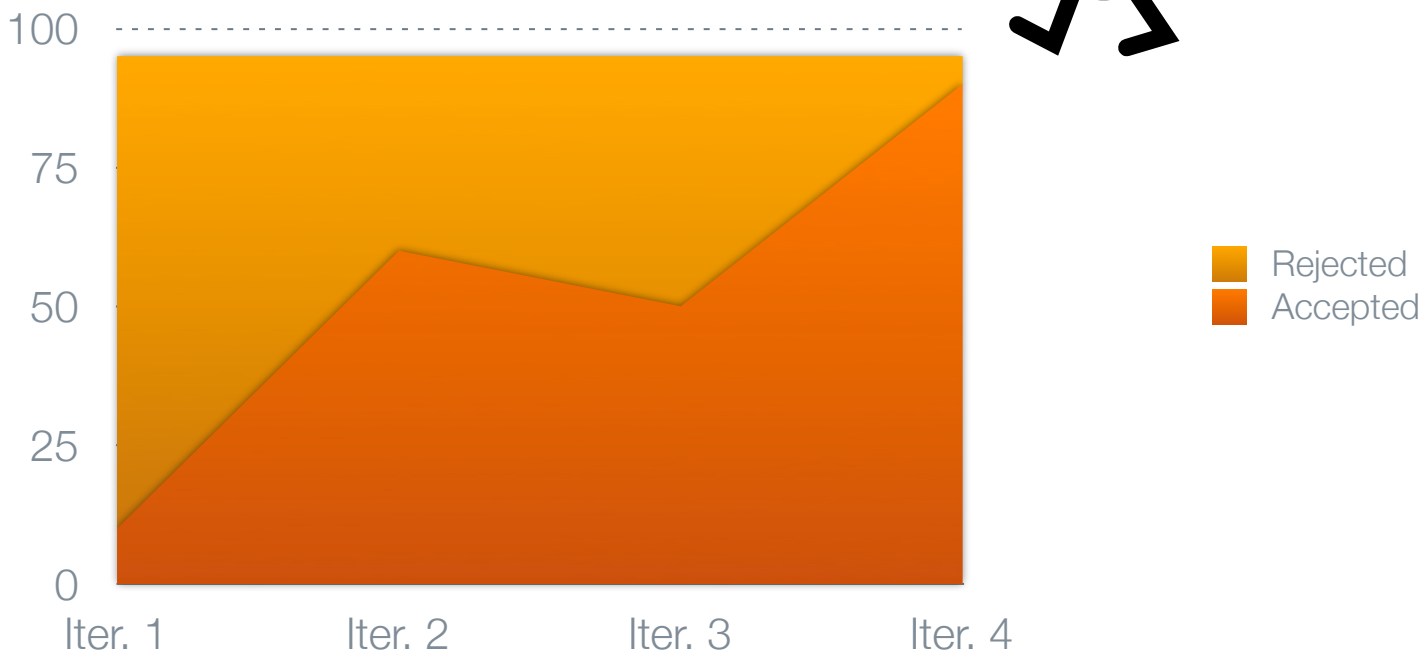
# Business Value Build Up



# Team Velocity

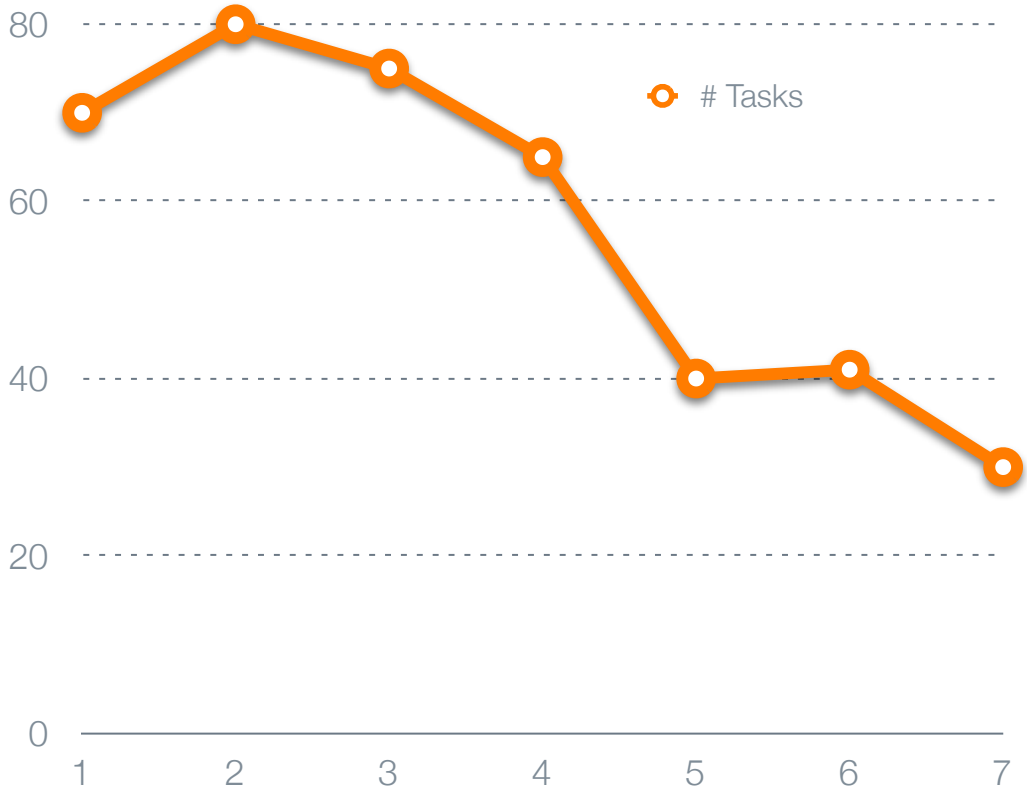


# Acceptance %

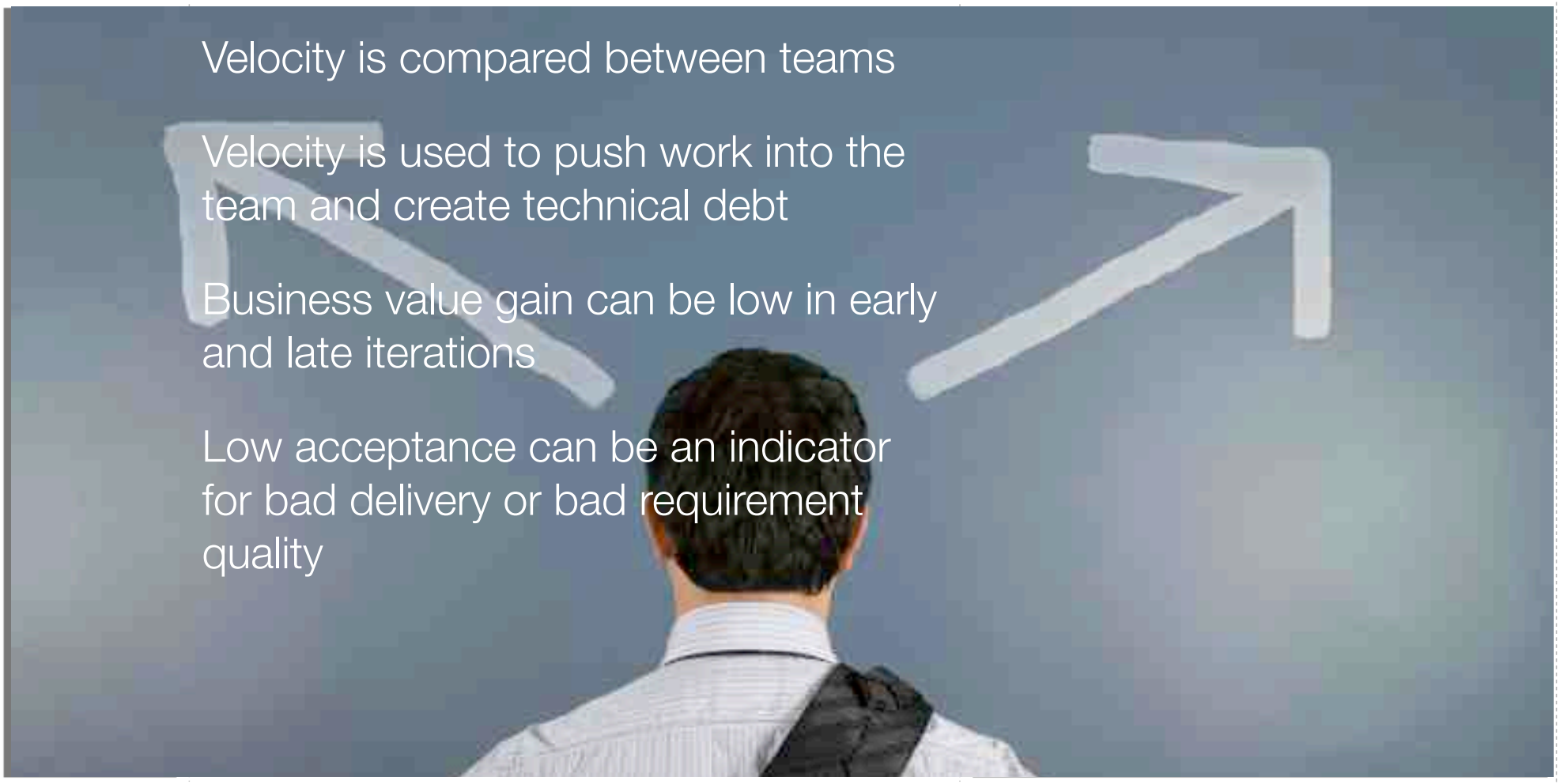


# Report Progress

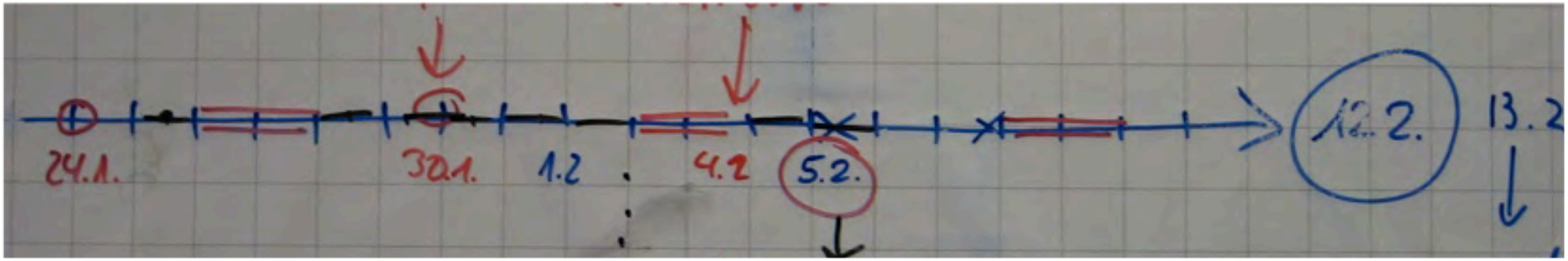
- Planned product backlog and releases.
- Revised product backlog and releases.
- Complete analysis of any changes in backlog, priorities, estimates
- Analysis of productivity
- Progress toward release
- Actions to improve



# Metric Pitfalls

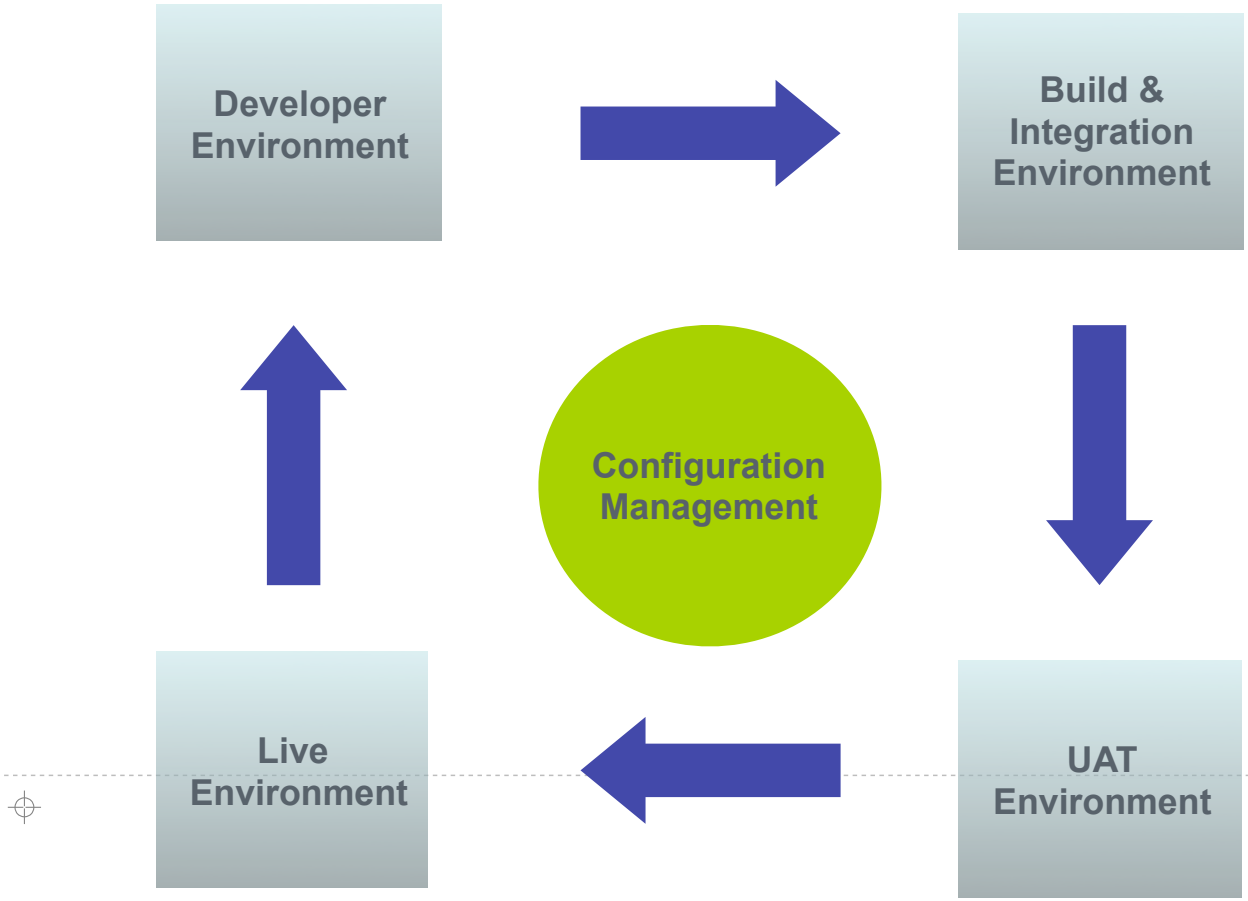


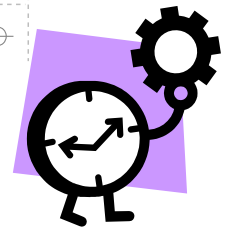
- Velocity is compared between teams
- Velocity is used to push work into the team and create technical debt
- Business value gain can be low in early and late iterations
- Low acceptance can be an indicator for bad delivery or bad requirement quality




# Engineering Practices

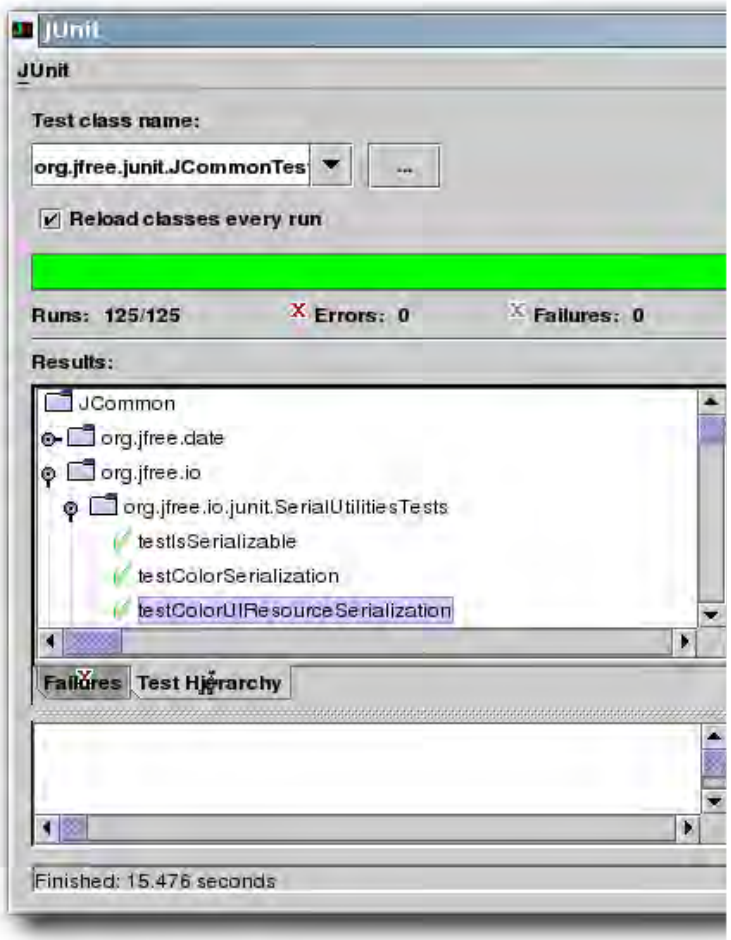
# Continuous Integration



# Testing

- Acceptance tests for stories
- Unit tests for components
- Automated integration testing
- Test to find bugs!



# Release Management

- Release Plan
- Responsible persons
- Rollout and roll-back plan
- Tagging, branching, berging
- Verification

# Collaboration

- Work on metrics
  - Apply any metrics that appear useful
- Establish continuous integration
  - Ideally a shared CI system
- Establish frequent reviews
  - Does the quality of the deliverable meet your requirements?
- Allow both sides to improve
  - perform frequent retrospectives

# Scenario: Game Portal

- Test automation developers, testers and developers form one team
- Initial resistance from developers to take testing tasks
- Testers closely integrated into team, automation developers work separately
- Cooperation is managed by the team in their Daily Standup sessions
- Testers manage the testcase batches for automation
- New functionality is still tested manually

# Exercise: Self

- Identify actions or activities that would improve the quality of the deliverables
- Think about the positive impact each activity would have
- Think about the effort each activity would take - S/M/L/XL/XXL
- Prioritize the activities
- How much could you reach in one month?

# Thank You


# Debrief

# Recommended Readings

- Beck
  eXtreme Programming explained
- Spillner/Linz
  Basiswissen Softwaretest – dpunkt 2005
- Hörmann/Dittmman/Hindel/Müller
  SPICE in der Praxis – dpunkt 2006
- Kneuper
  CMMI – dpunkt 2006
- Kerth
  Project Retrospectives – Dorset House 2001
- Schwaber/Beedle
  Agile Software Development with Scrum – Prentice Hall 2002
- Cohn
  User Stories Applied – Addison Wesley 2004
  Agile Planning and Estimation – Prentice Hall 2006