cibit academy

inspearit training services

# Marshmallow Challenge

# Marshmallow Challenge

- Goal
  - Build the Tallest Freestanding Structure
- Building kit
  - 1 meter string (may be cut to pieces)
  - 1 meter masking tape (may be cut to pieces)
  - 20 spaghetti sticks (may be broken to pieces)
  - 1 marshmallow (may not be reduced in size)
- Guidelines
  - Measurement will take place 18 minutes after the start
  - The height will be measured from the base till the top op de marshmallow
  - It is not allow to support the structure with something higher than the base

18:00

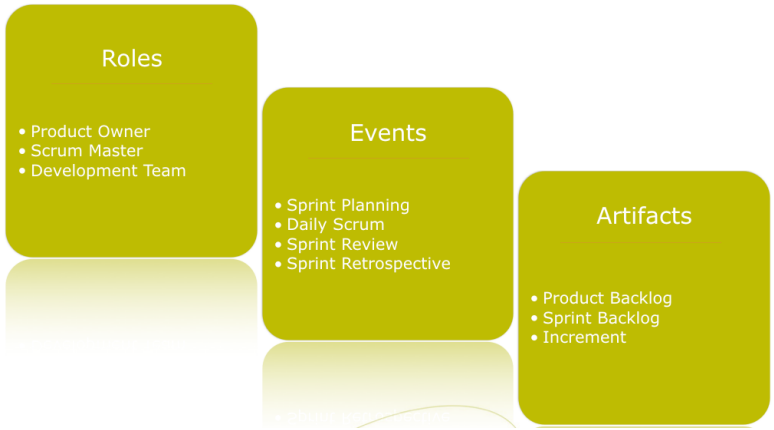
# Shuhari

cibit academy
inspearit training services

# Professional Scrum Master

## Welcome to day 2

# Recap day 1





| Day 1 |
| --- |
| Introduction |
| Scrum process & Scrum roles |
| Scrum events & Scrum artifacts |
| **Lunch** |
| Scrum simulation |
| Scrum Master profile |
| Coaching the Product Owner |
| Retrospective day 1 |

# Planning day 2

**Day 2**

Recap day 1 & Planning day 2

Coaching the Development Team

Effective retrospectives

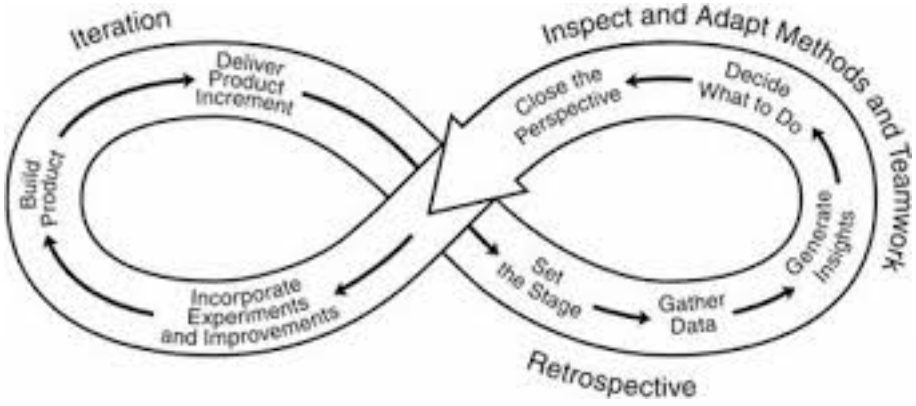
**Lunch**

Coaching the Organization

Removing impediments

Relation between Agile & Scrum

Open for questions
Retrospective day 2 & course evaluation




cibit academy
inspearit training services

# Coaching the Development Team

# Team dynamics

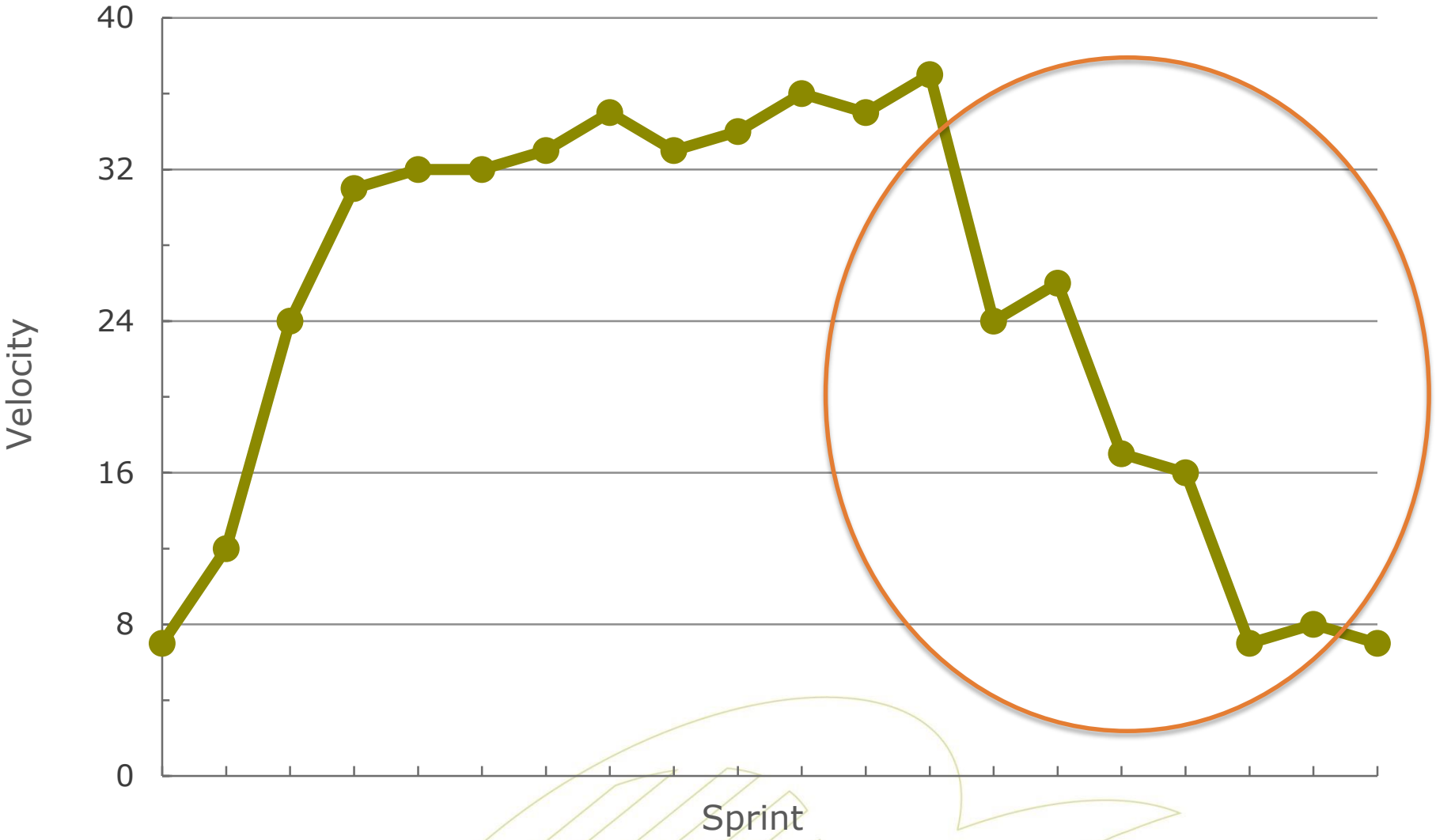
# Penny game (with cards !)

cibit academy
inspearit training services
workshop

- Make group of 4 workers and 1 manager
- The group of 4 workers passes along cards to each other in batches. Each worker adds value by turning the card over
- The manager time how long for the first coin to reach the customer, and how long for the whole set of cards to reach the customer

- Round one: batch size is 25
- Round two: batch size is 5
- Round three: batch size is 1

# Excellence is not an option

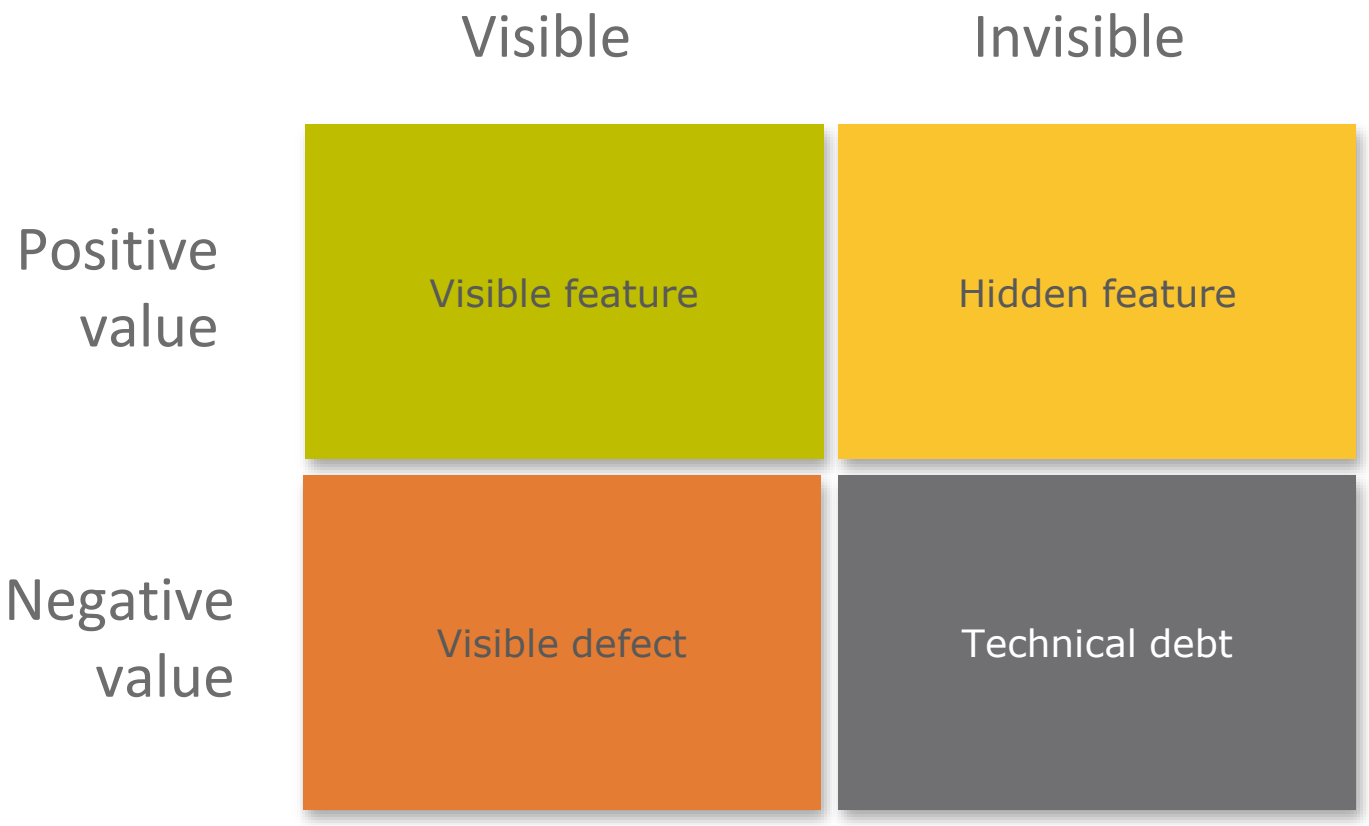


cibit academy

inspearit training services

# Technical Debt

# Technical debt

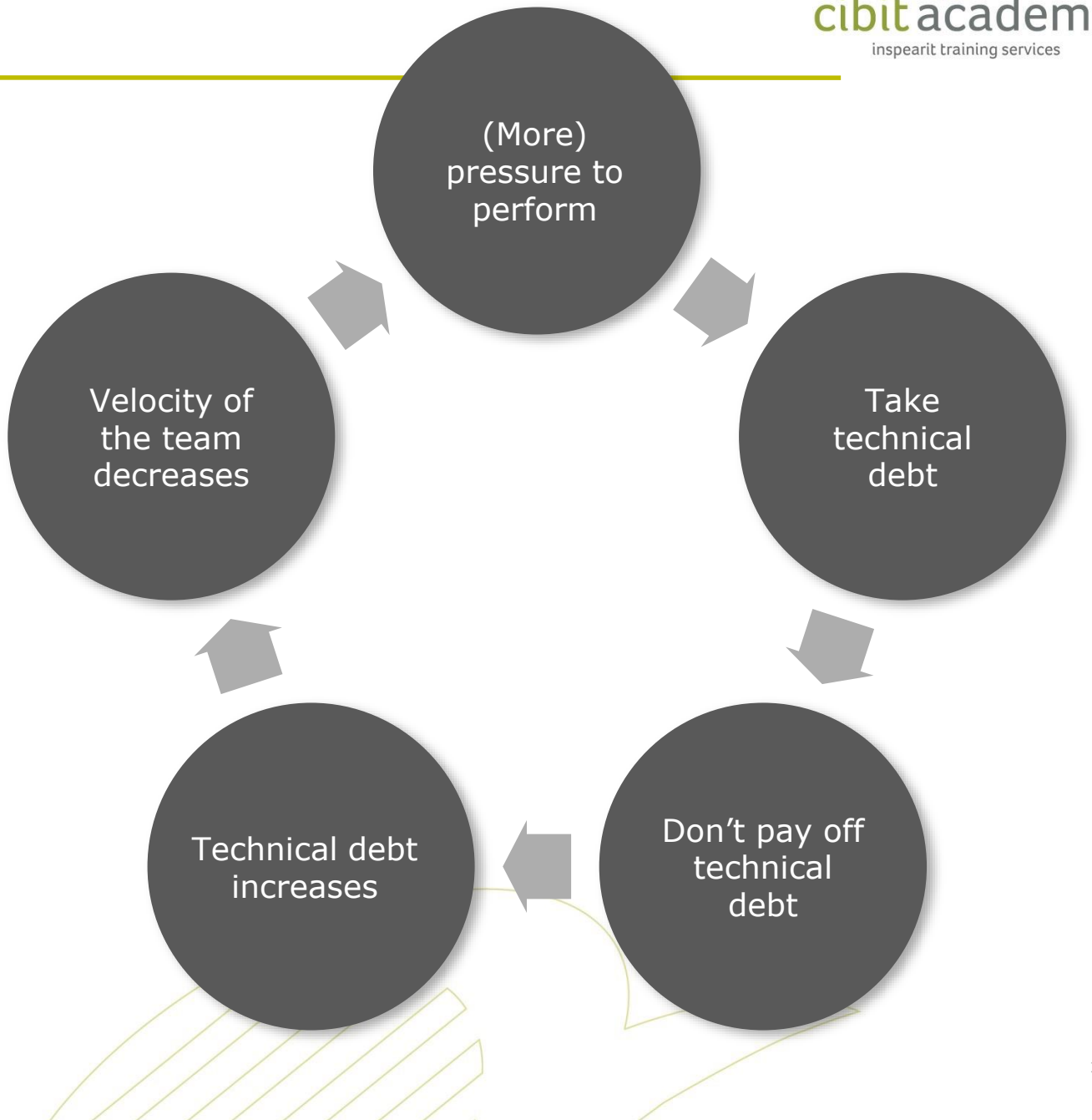


# Kinds of technical debt

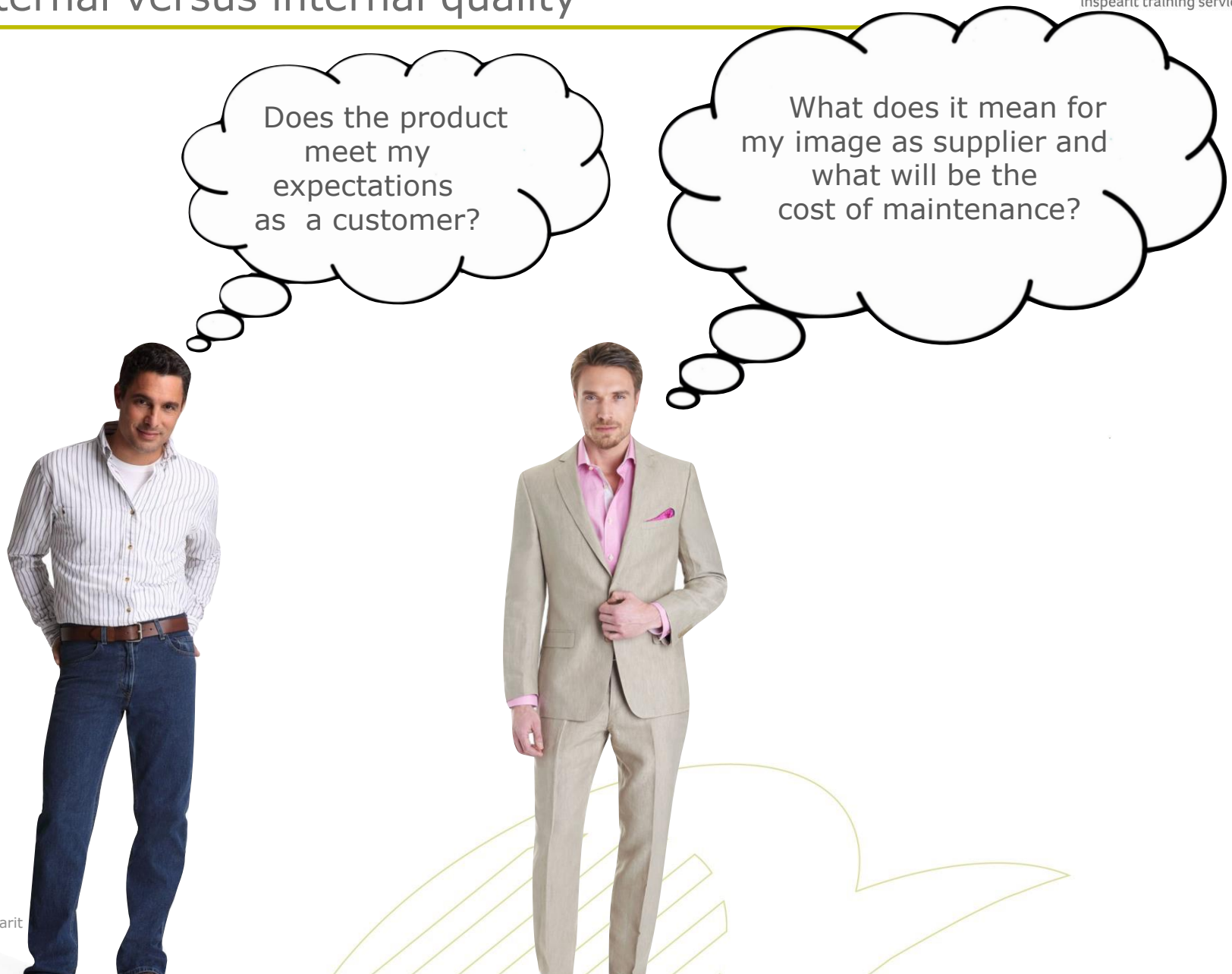
| | Reckless | Prudent |
|---|---|---|
| Deliberate | *"We don't have time for design"* | *"We must ship now and deal with consequences"* |
| Inadvertent | *"What's Layering?"* | *"Now we know how we should have done it"* |

# Technical debt

… and we are the root cause!



# External versus internal quality



# Internal quality is not negotiable

My code base must be stable and remain so for the long run to prevent technical debt.

That's why I don't negotiate about internal quality: it must be good, because that's what my customer expects!

# Definition of Done

- Define a Shared Understanding between PO and development teams of what it means for a PB item to be Done


## Team "Done" list

**… with a story**

- All code (test and mainline) checked in
- All unit test passing
- All acceptance tests identified, written & passing
- Help file auto generated
- Functional tests passing

**… with a Sprint**

All story criteria, plus…

- Product backup updated
- Performance testing
- Package, Class & Architecture diagrams updated
- All bugs closed or postponed
- Code coverage for all unit tests at > 80%

**… release to INT**

All Sprint criteria, plus…

- Installation packages created
- MOM packages created
- Operations guide updated
- Disaster recovery plan updated
- All test suites passing

**… release to PROD**

All INT criteria, plus…

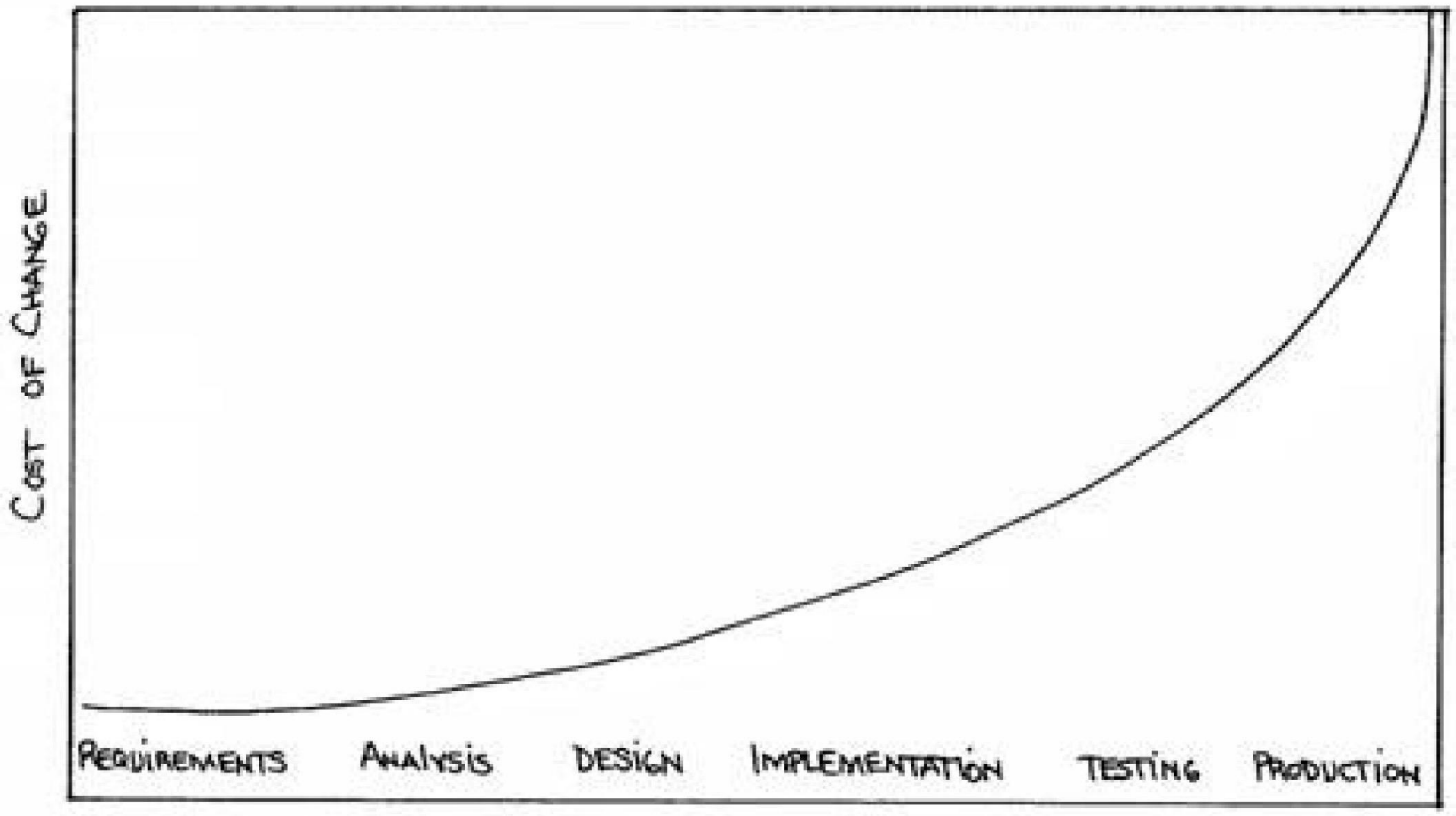
- Stress testing
- Performance tuning
- Network diagram updated
- Security pass validated
- Threat modeling pass validated
- Disaster recover plan tested

# How to prevent technical debt

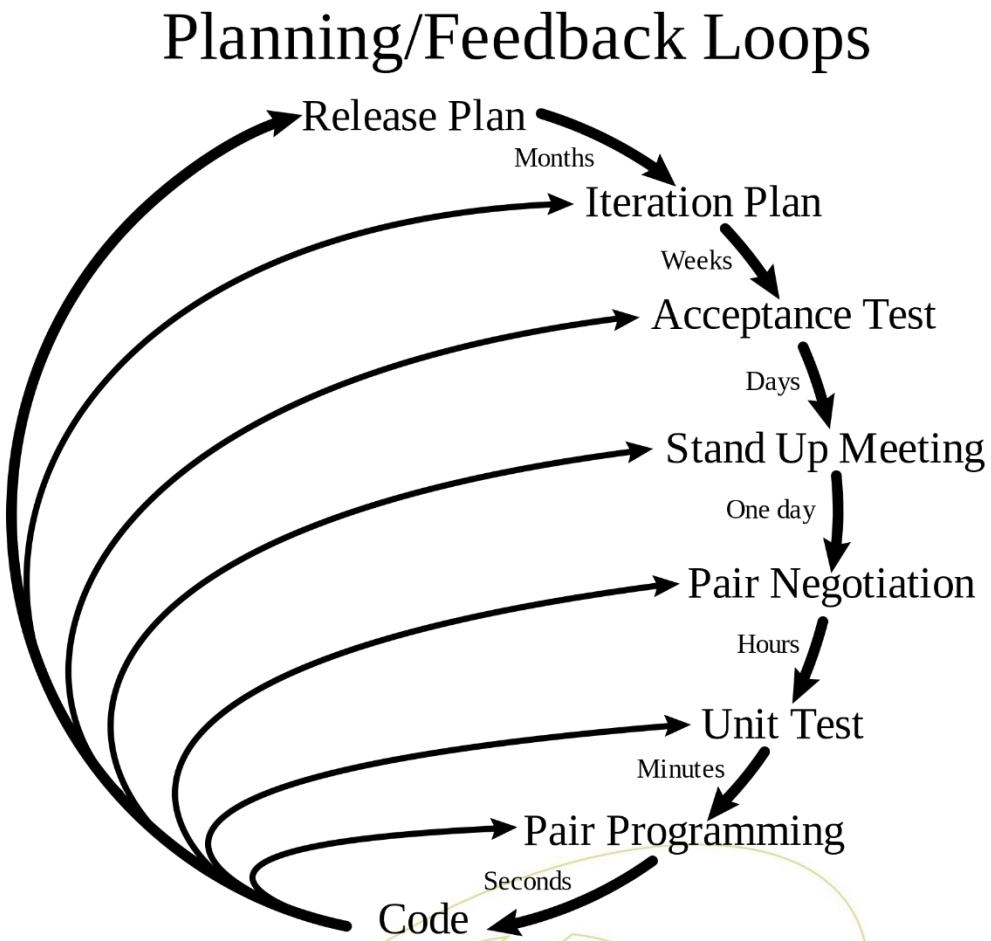
cibit academy
inspearit training services

- Gain knowledge about technical debt and it’s cost
- Stimulate refactoring
- Use Test Driven Development (TDD)
- Continuous integration
- Just in Time design
- Shared ownership of code
- Multidisciplinaire development teams
- Involvement of architects
- Give room to work on internal quality

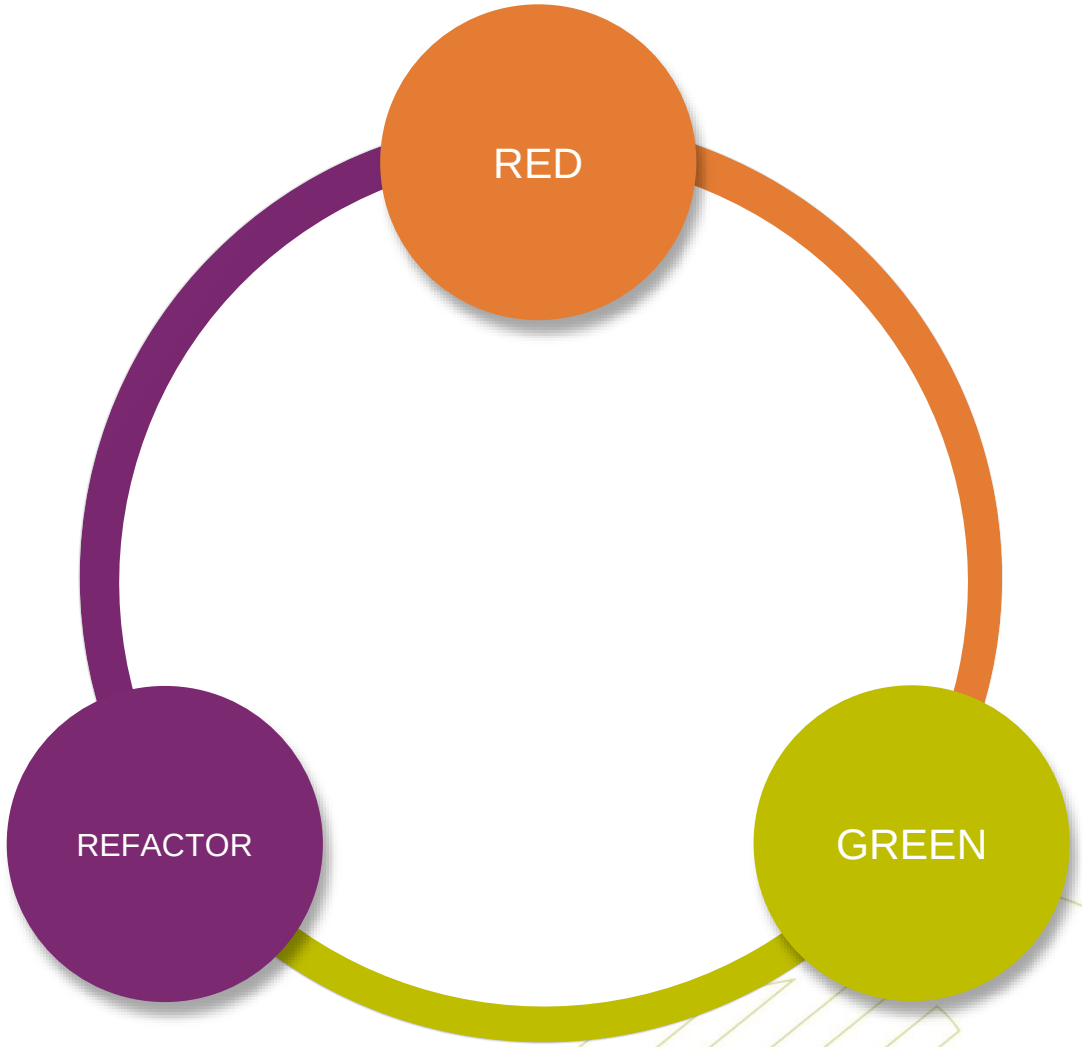
# Cost of Delay in finding defects

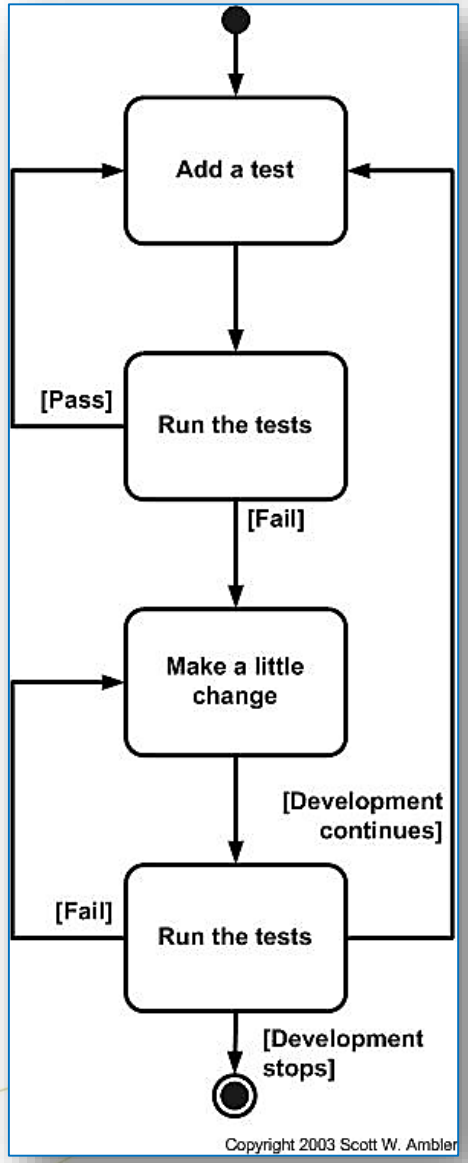


# Feedback loops



# Test Driven Development



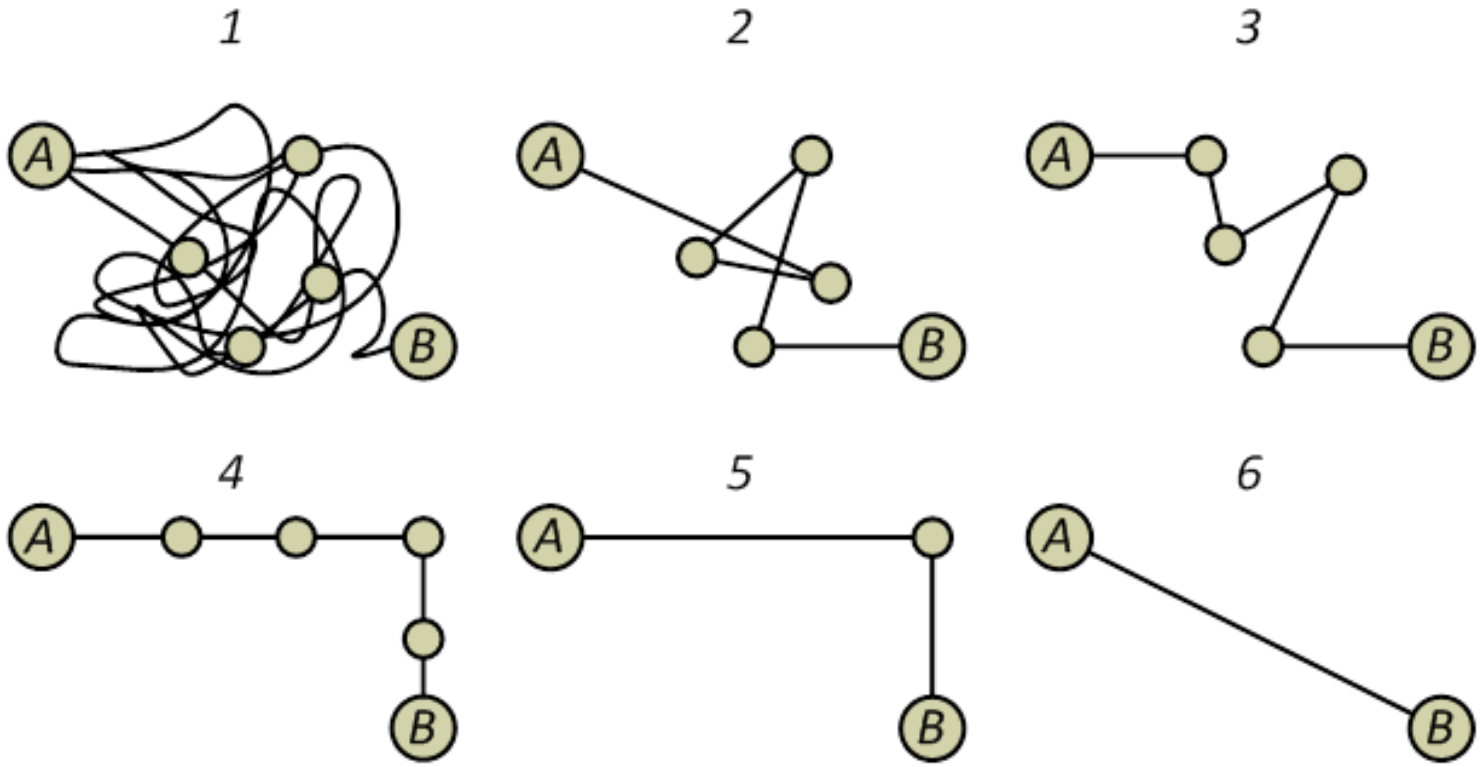



# Laws of Test Driven Development

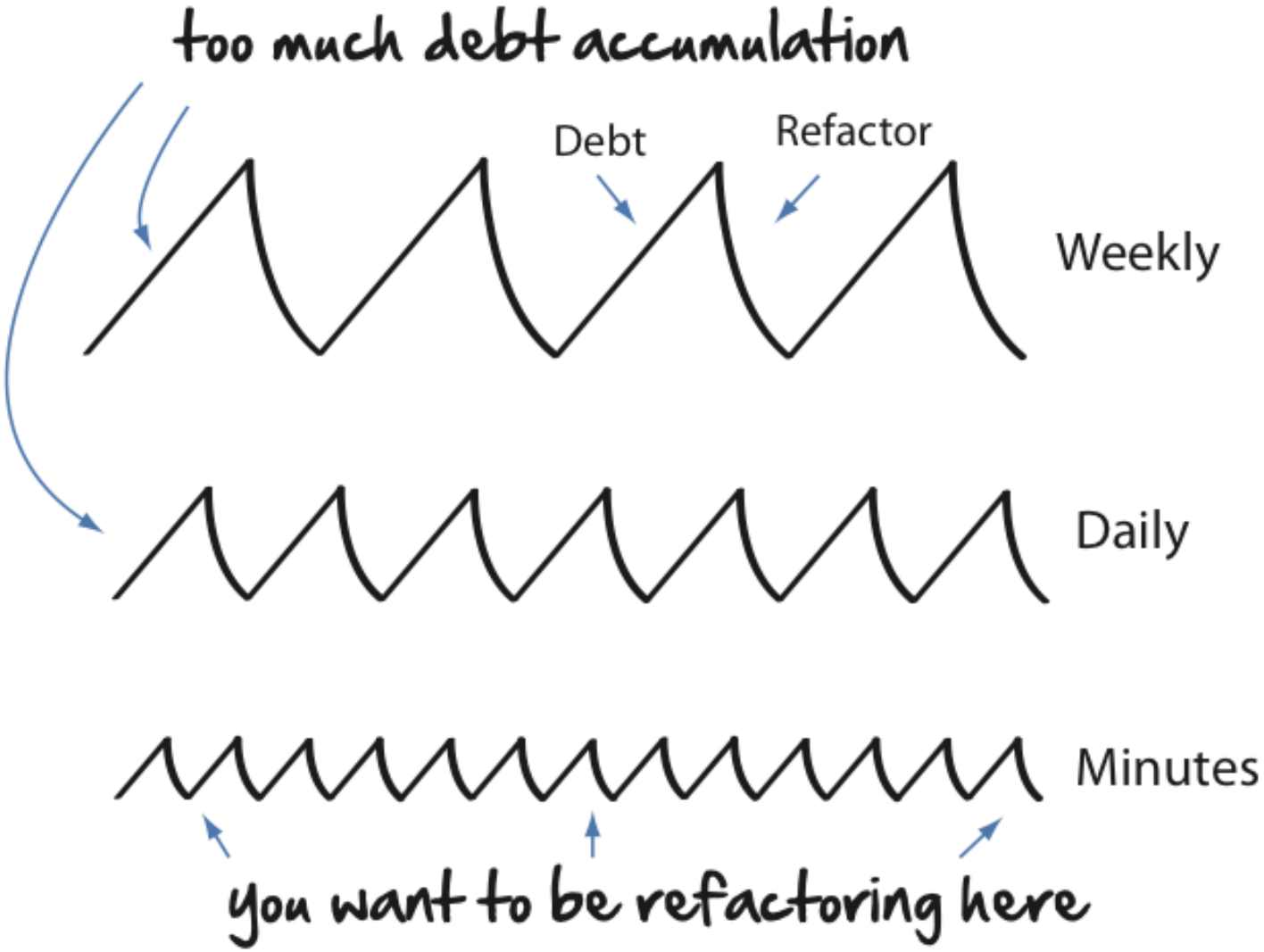
- You are not allowed to write any production code unless it is to make a failing unit test pass.
- You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
- You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

- Refactor is not optional.

# Refactoring

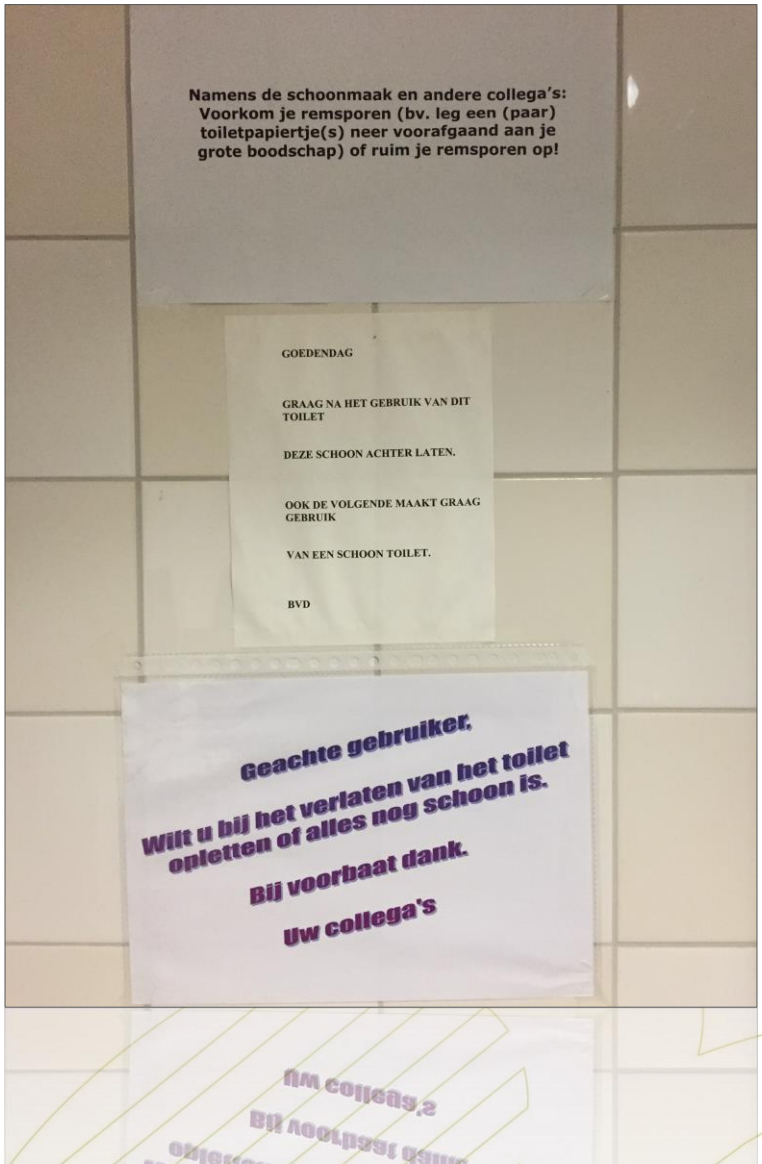


# Refactoring




# Visual Management

# We present a lot of instructions…



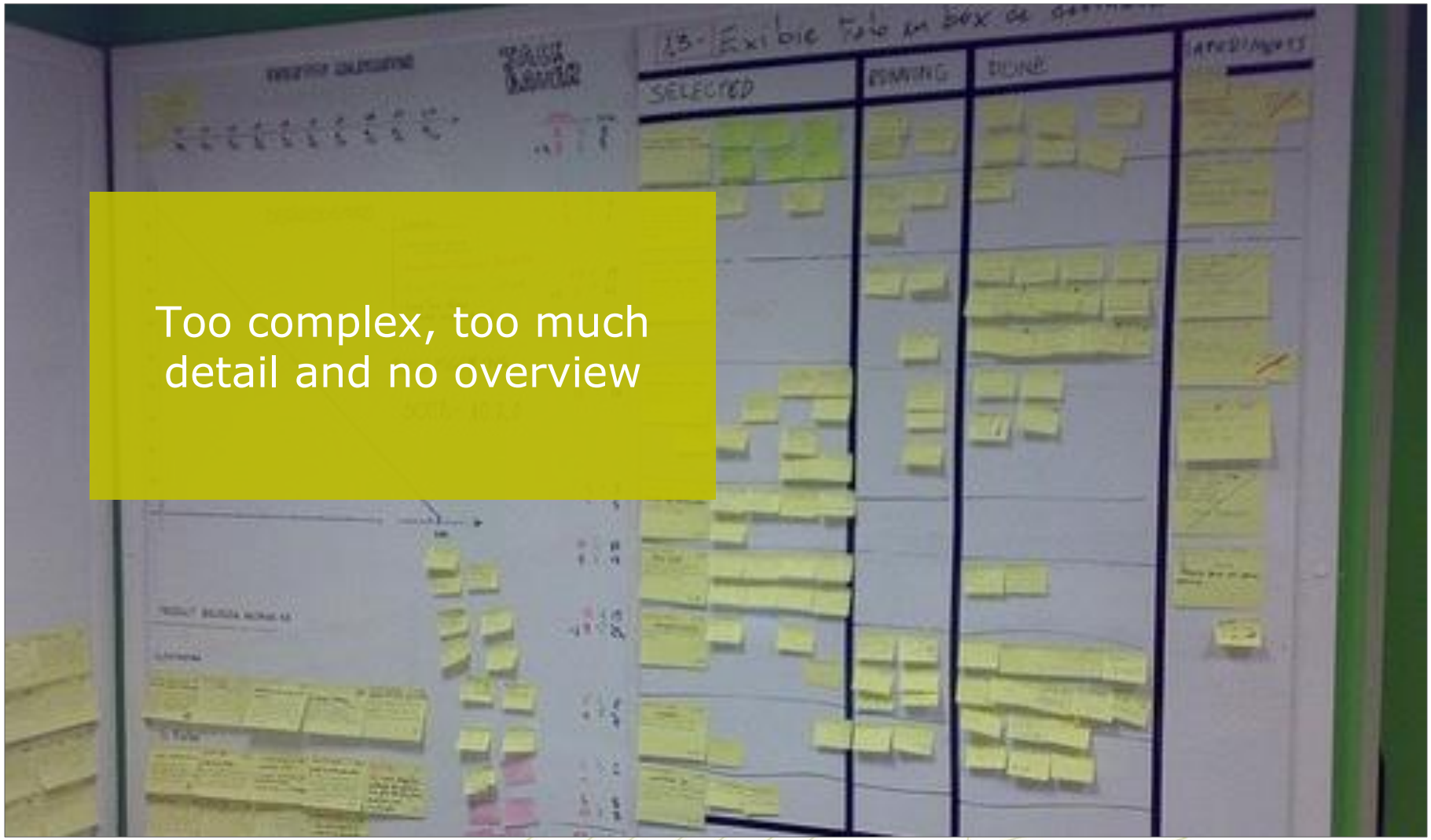

… but hanging information is not information radiation

# Teams create information radiators

- To be able to plan effectively as a team, and to develop and take over each others work when needed.
- To be able to involve stakeholders, provide overview in what the team does and what the status is, and to create enthusiasm.
- Together with working software (Demo's) meant to eliminate the need of traditional progress reports.

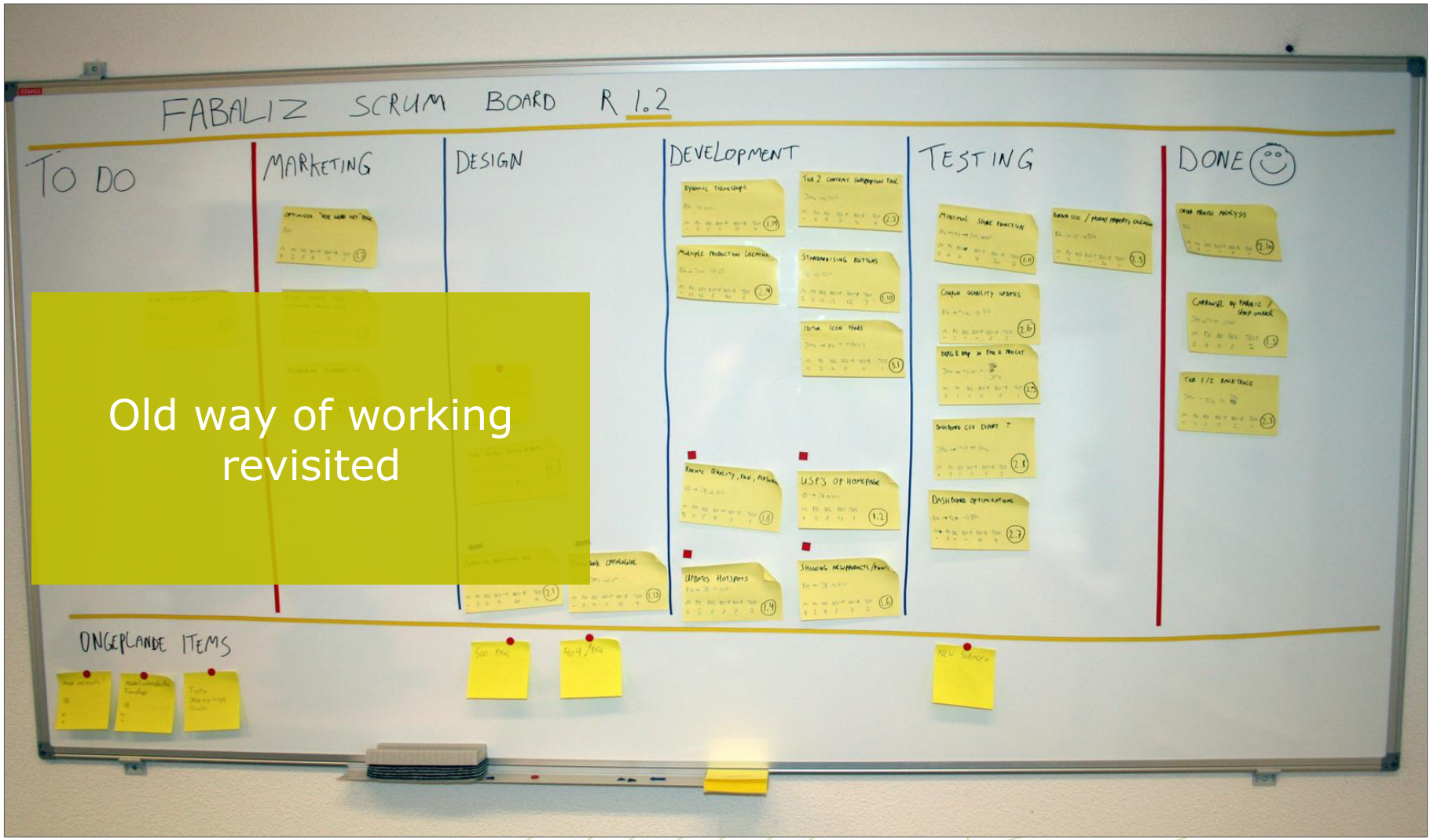
Challenge the team to work visually!

# Smell 1

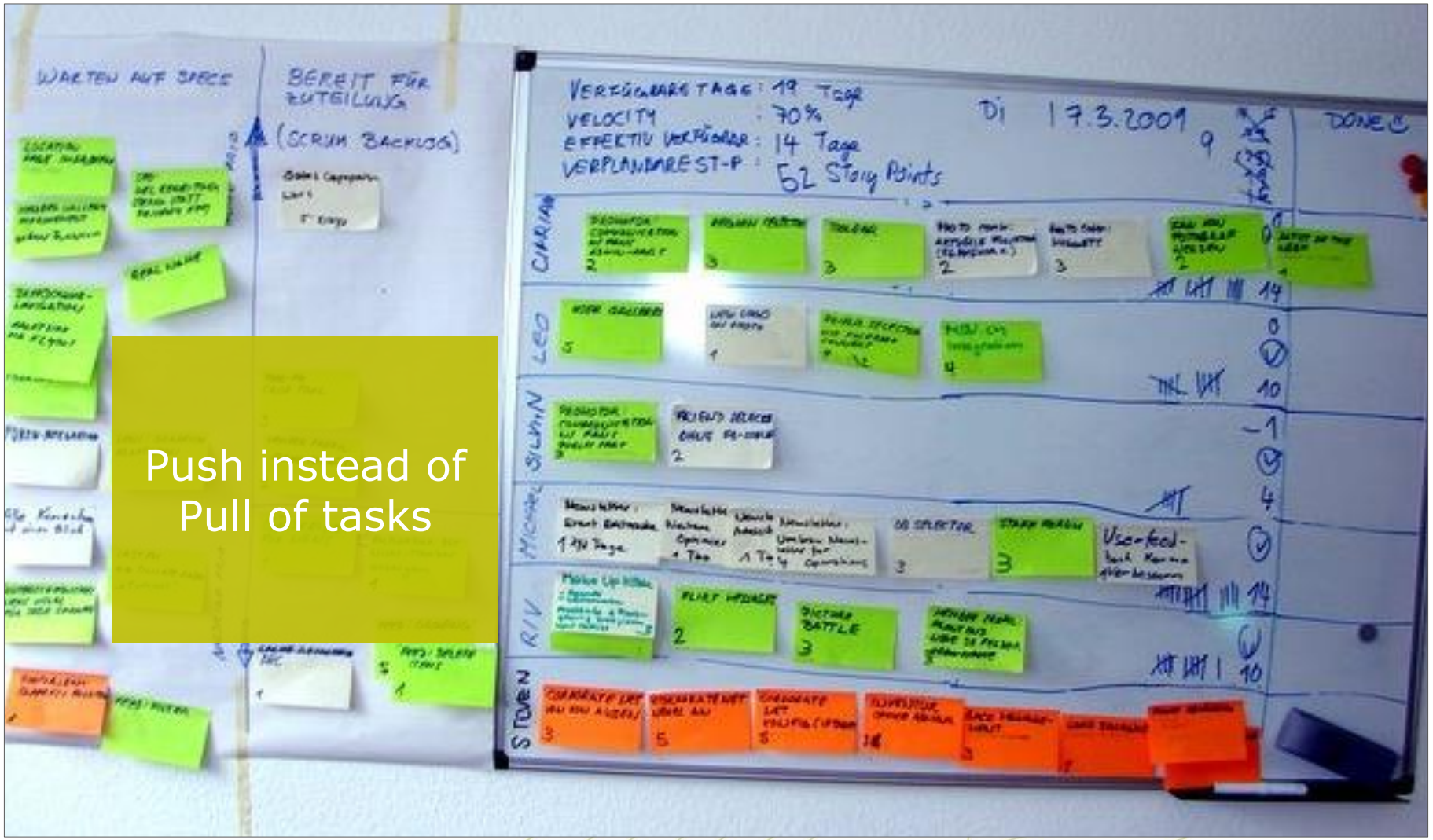



Too complex, too much detail and no overview

# Smell 2


FABALIZ SCRUM BOARD R 1.2
TO DO
MARKETING
DESIGN
DEVELOPMENT
TESTING
DONE
ONGEPLANDE ITEMS
Old way of working revisited

# Smell 3

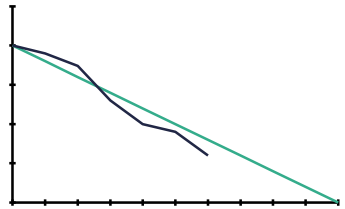

WARTEN AUF SPRINT
BEREIT FÜR ZUTEILUNG
(SCRUM BACKLOG)
VERFÜGBARE TAGE: 19 Tage
VELOCITY: 70%
EFFEKTIV VERFÜGBAR: 14 Tage
VERPLANBARE ST-P: 52 Story Points
Di 17.3.2009
DONE
CIARIAN
LEO
SILVAN
MICHAEL
RIV
STEVEN
Push instead of Pull of tasks

# Scrum board

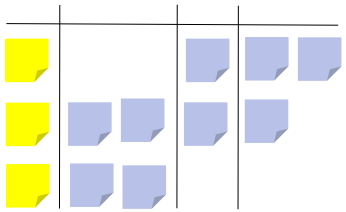
# Visual Management & Indicators

cibit academy
inspearit training services

**Sprint Burndown**

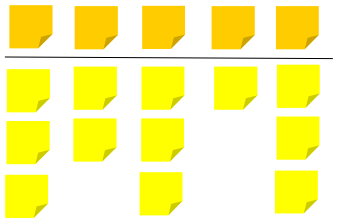


**Sprint Backlog**

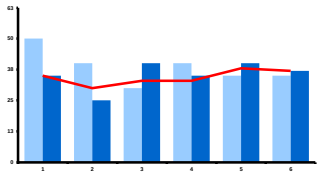


Sprint level

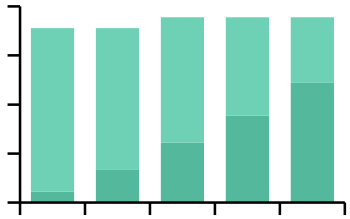
**Product Backlog**



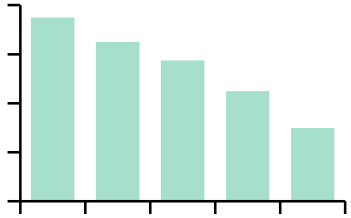
**Velocity**



**Release Burnup**

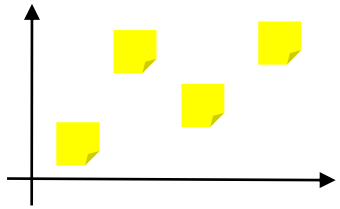


**Release Burndown**

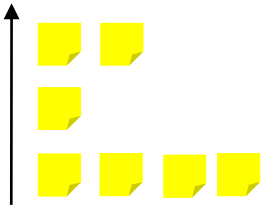


Project / release level

**Risks**



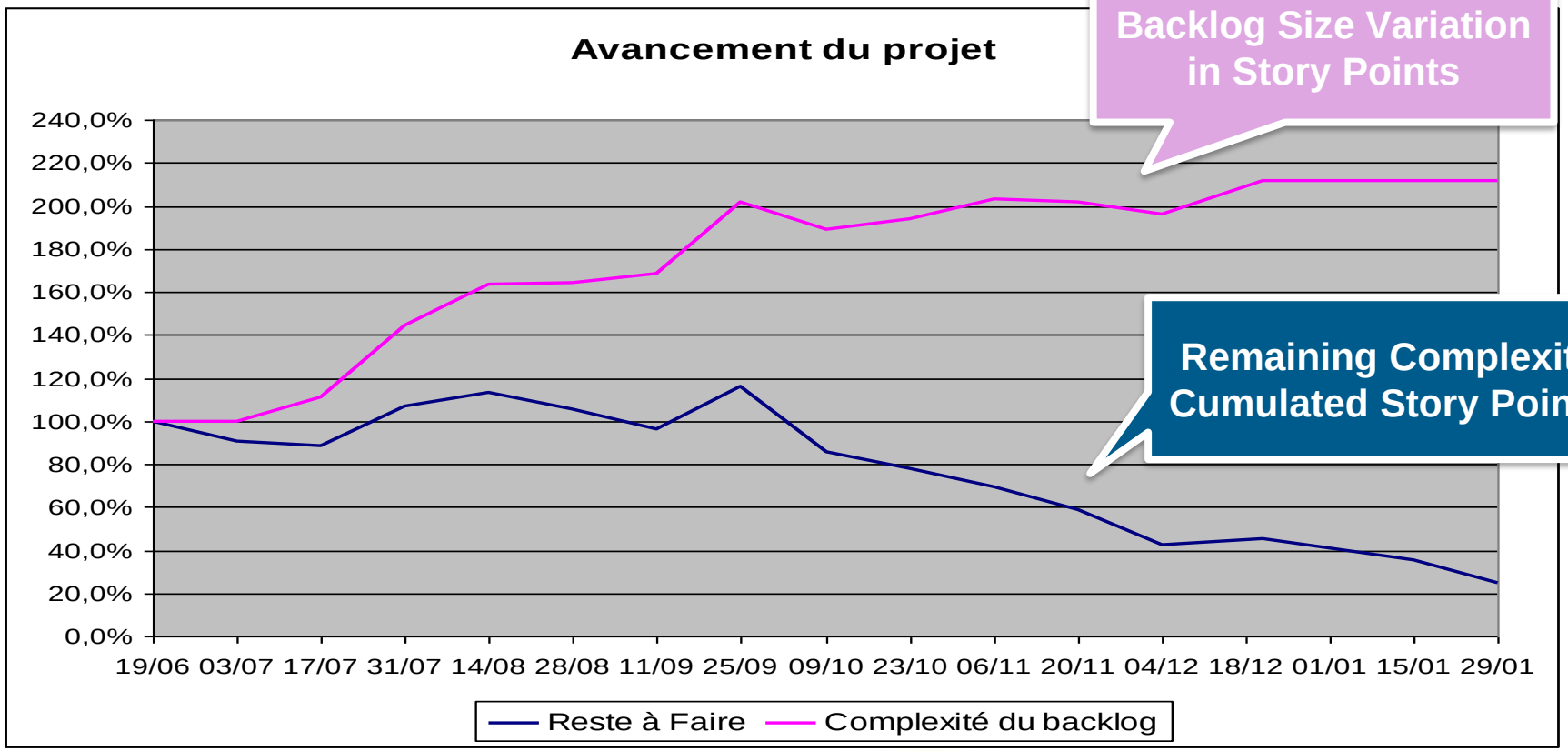
**Impediments**



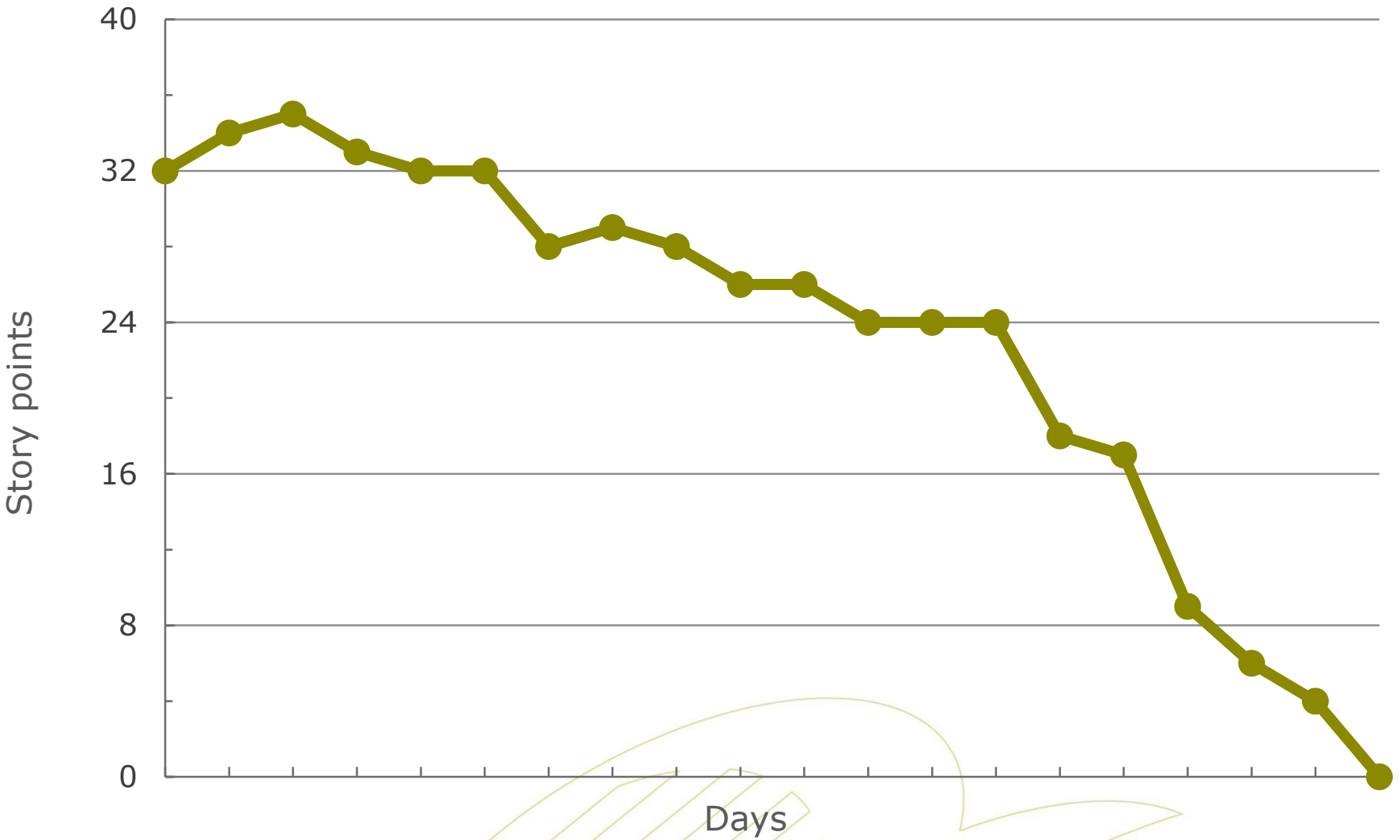
Follow up

# Velocity

# Burn down chart



# Sprint burndown chart




# Documentation in an Agile environment

# Rule of thumb

If you "need" a lot of documentation…

… you most probably will have a communication problem!

# The pitfalls of documentation

- Gives a false (but comfortable) feeling of productivity.
  - Working software is the only measure of progress, not documentation.
- Documentation provides piece of mind when all the thinking is on paper, regardless if anyone will read it.
  - Documents are hardly tuned to the target audience, if the target audience is even known
  - Gives a false (but comfortable) feeling of building collective knowledge.
- Only documentation that is fully understood and accepted contributes to building on collective knowledge.
  - People are not good at throwing away documentation.
- This will not contribute to the maintainability of a product.

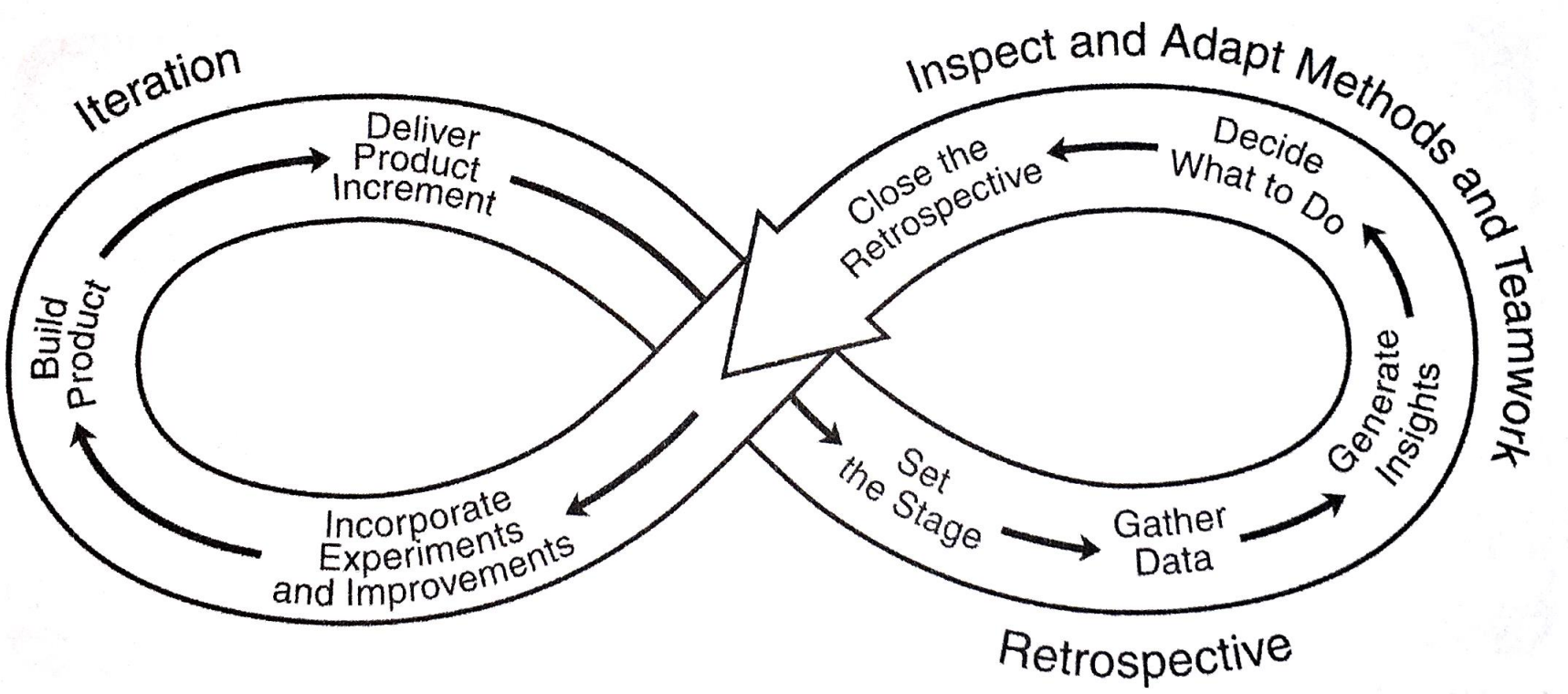
Maximize the amount of work not done is better!

cibit academy
inspearit training services

# Effective retrospectives

# The Scrum Master shines at the retrospective


# Retrospective process



# It’s more than just the retrospective activity

- Step 1 - provide context (why are we here?)
- Step 2 - prime directive (what do we want do achieve?)
- Step 3 - energize the group
- Step 4 - check in (provide safety)
- Step 5 - facilitate the retrospective activity
- Step 6 - filter the information gathered and put it into actions
- Step 7 - check out (e.g. ROTI-score)

# "Sailing Boat" exercise

cibit academy
inspearit training services
workshop

- Are we still on target to become a Professional Scrum Master ? (harbor)
- What are impediments for a perfect course? (anchors)
- What are enhancers for a perfect course? (winds)

cibit academy
inspearit training services

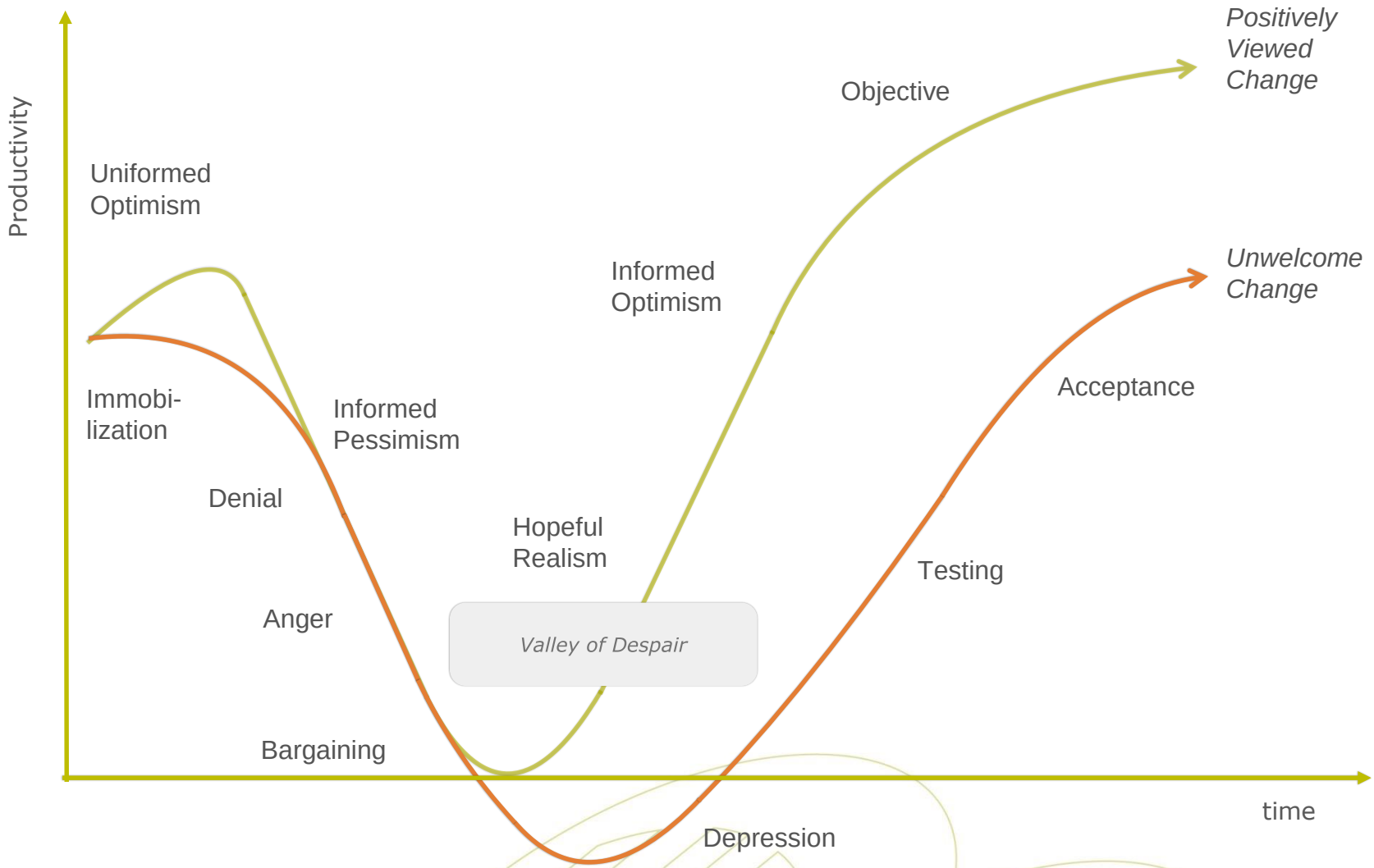
# Coaching the organization

# It’s all about change management


CHANGE !!!

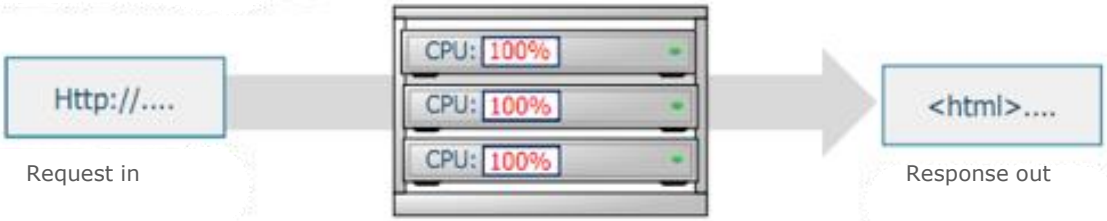
# 4 Levels of change

# Emotional cycles of change



# Beware of overload

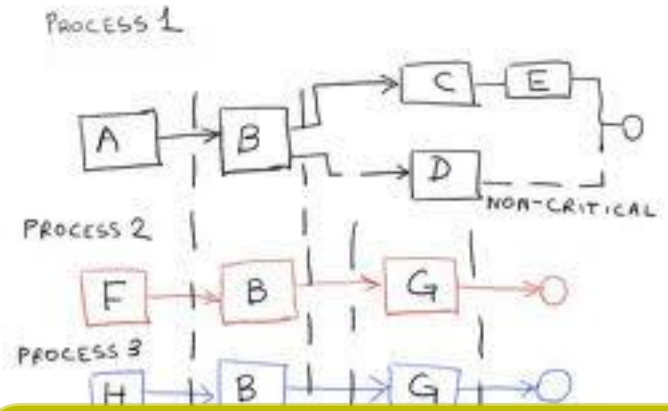

*100% road utilization = terrible throughput*



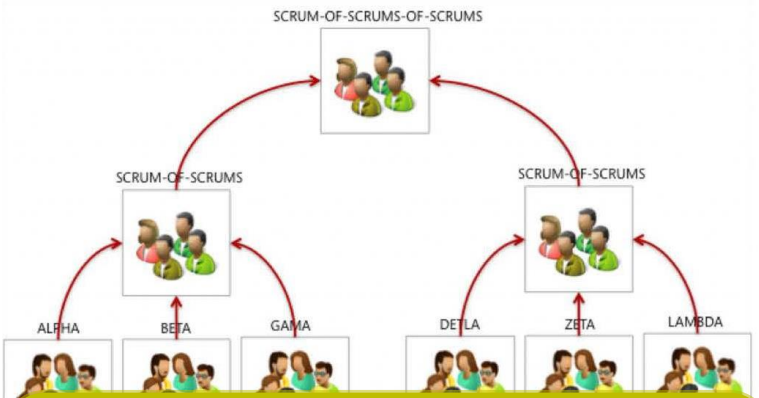

*100% server utilization = terrible throughput*


*100% team utilization = terrible throughput*

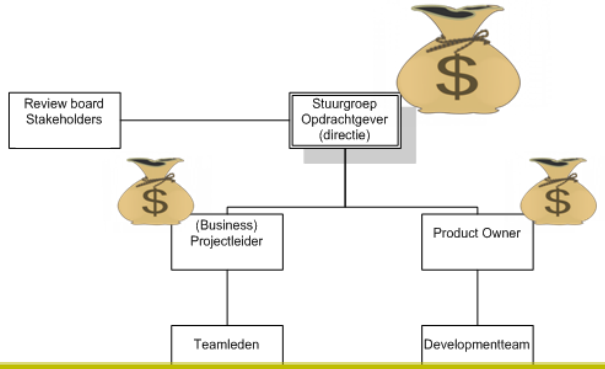
# Let’s talk about typical organization challenges




Agile in a non-agile company



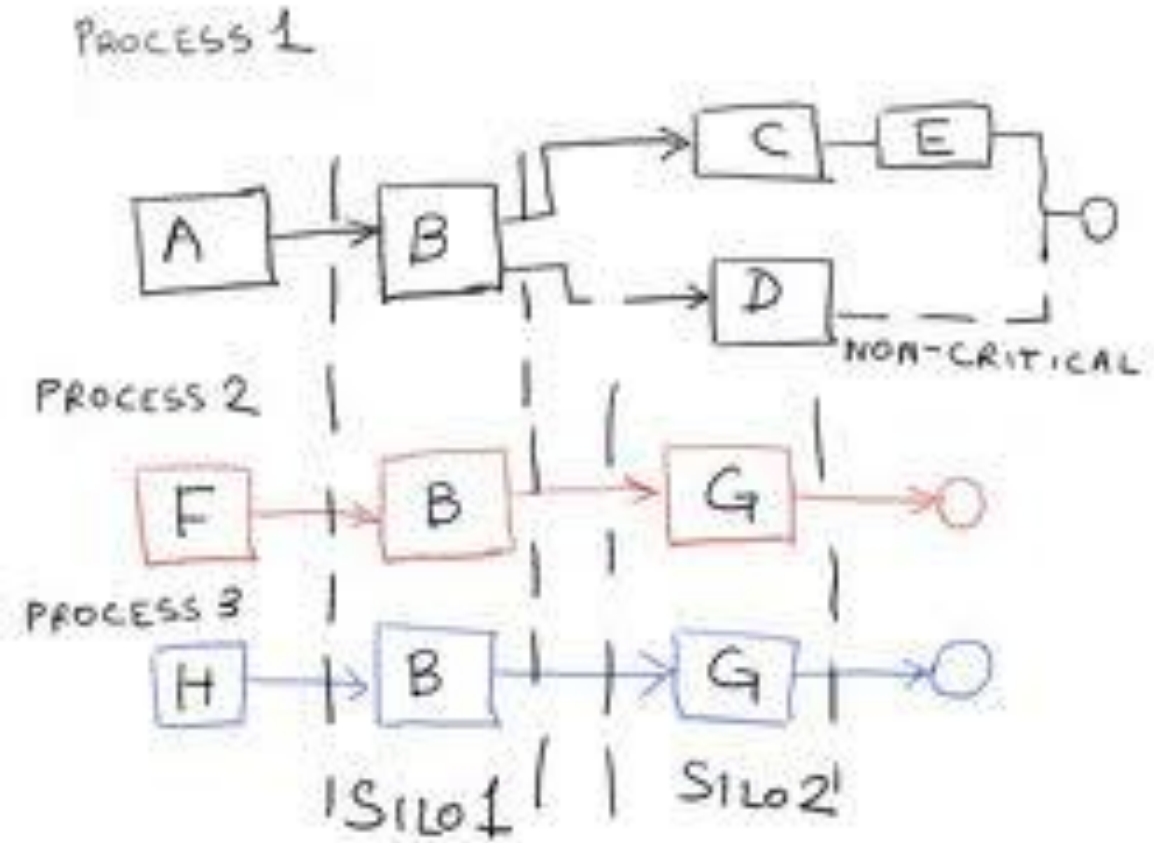

Dependencies with other teams



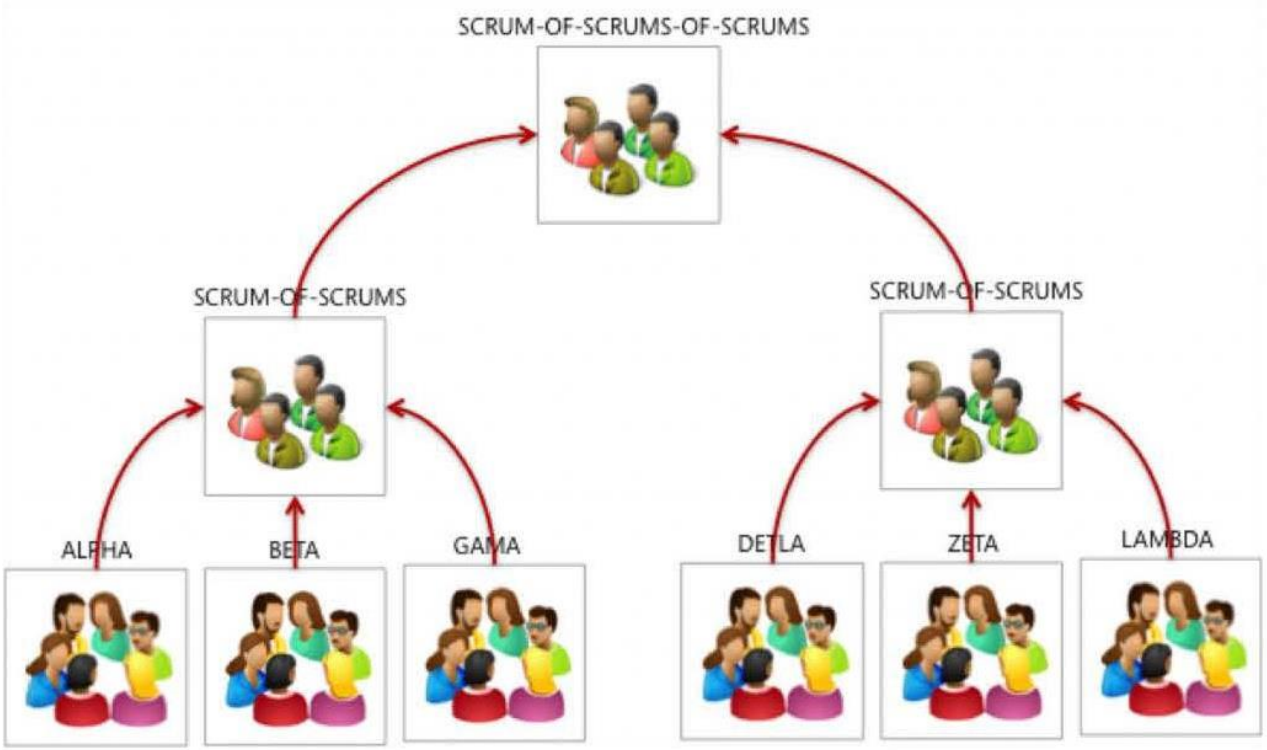

Agile in a project organization


Work together with management

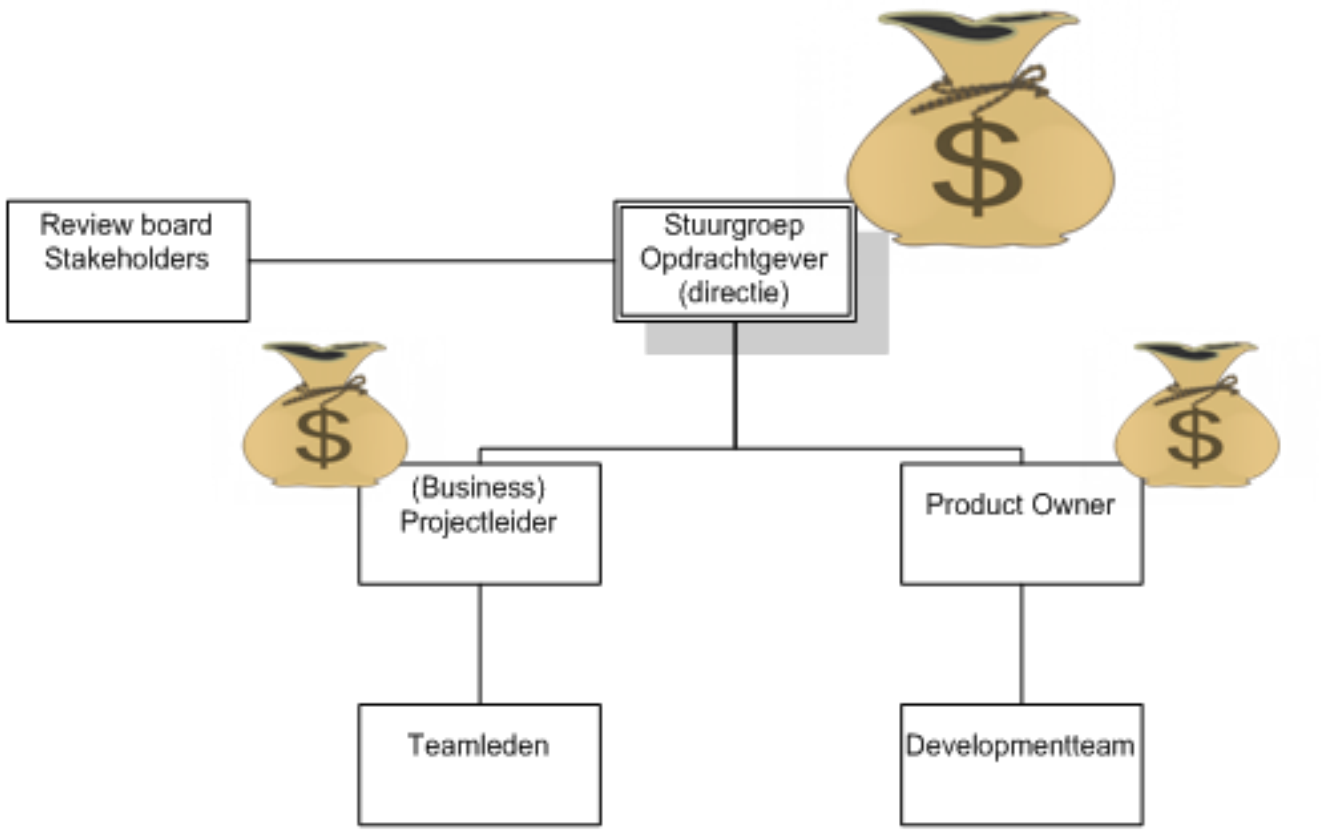
# Agile in a non-Agile organization



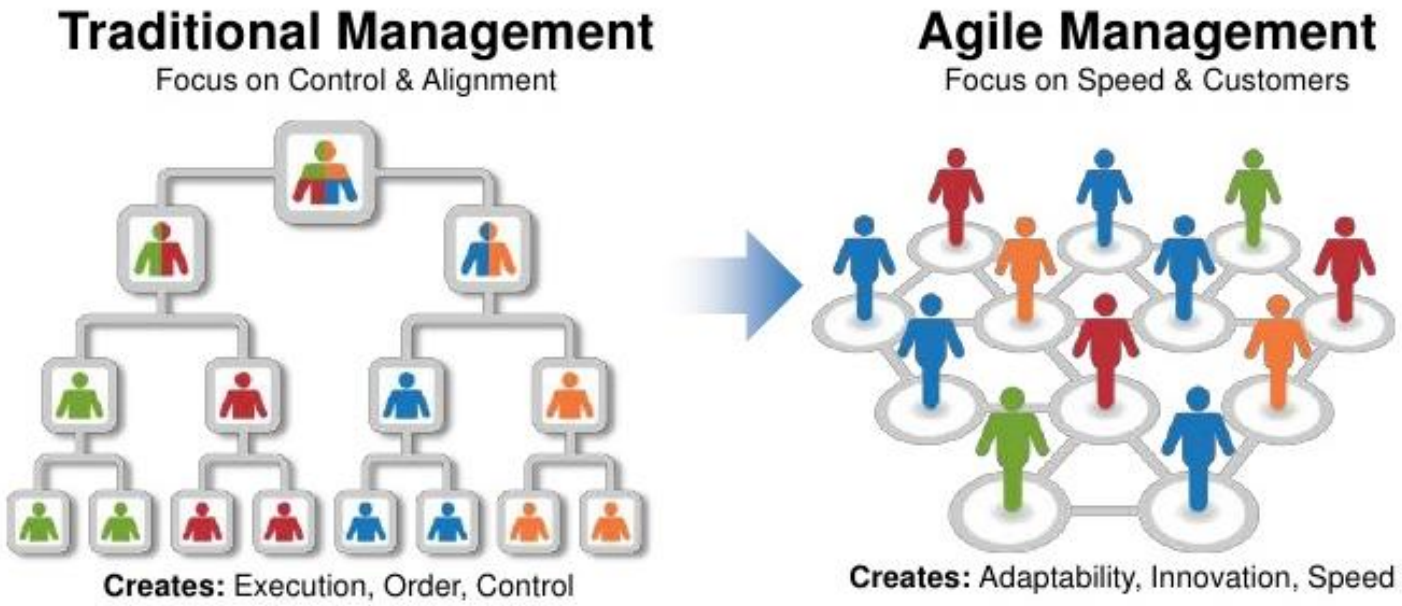
# Dependencies with other teams



# Agile in a project organization

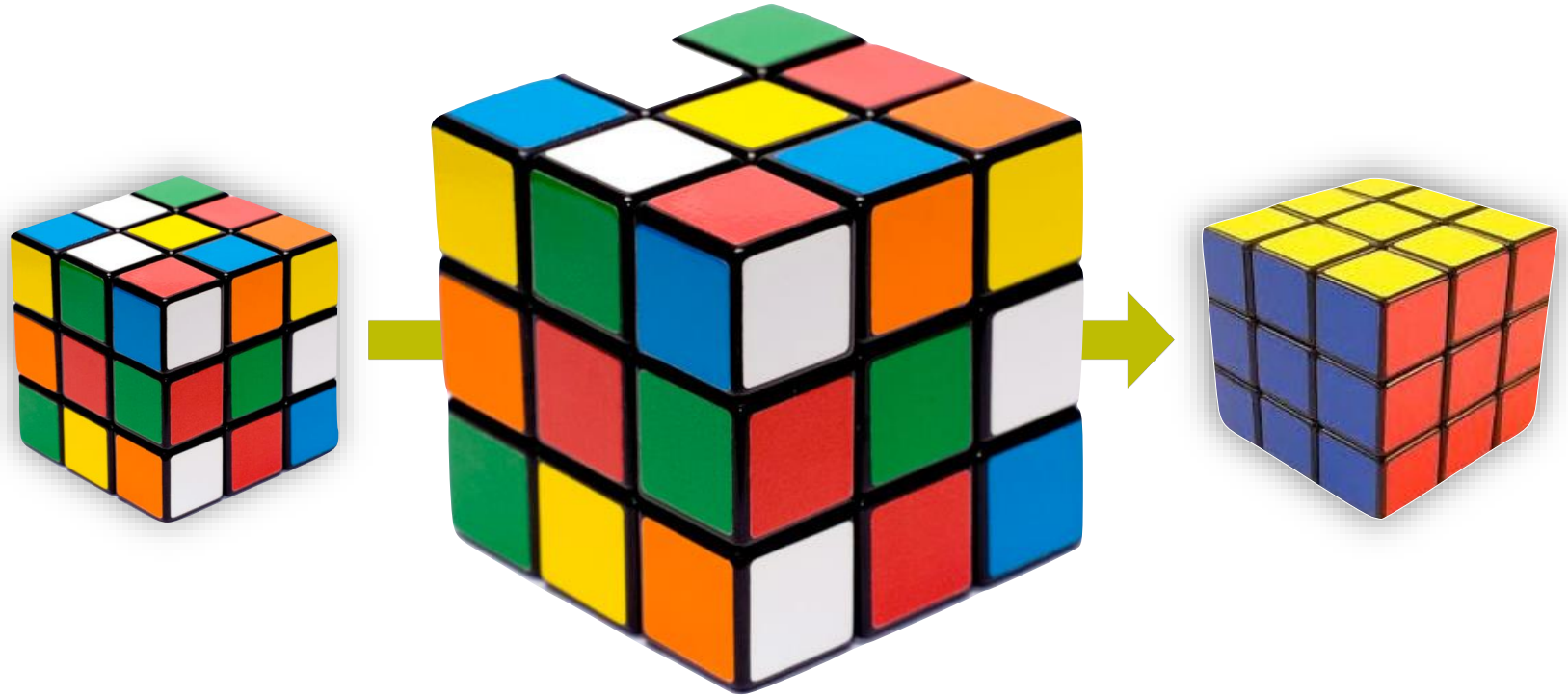


# Work together with management




# Removing impediments

# A3 Problem Solving



# A3 Problem Solving

cibit academy
inspearit training services

Detecting a problem → A3 process approval → Assigning A3 owner → Clarify context → Assemble a team



Select a solution → Implement solution → Evaluate solution → Continu or start new A3

Theme:

# Problem

To:
From:
Date:

## Background

## Current situation

## Target condition

## Root cause analysis

## Recommendations

## Implementation plan

## Result and follow-up

Theme:

_____
_____

# *Problem*

To: _____
From: _____
Date: _____

Current situation

Future situation

Theme:

_____
_____

# *Problem*

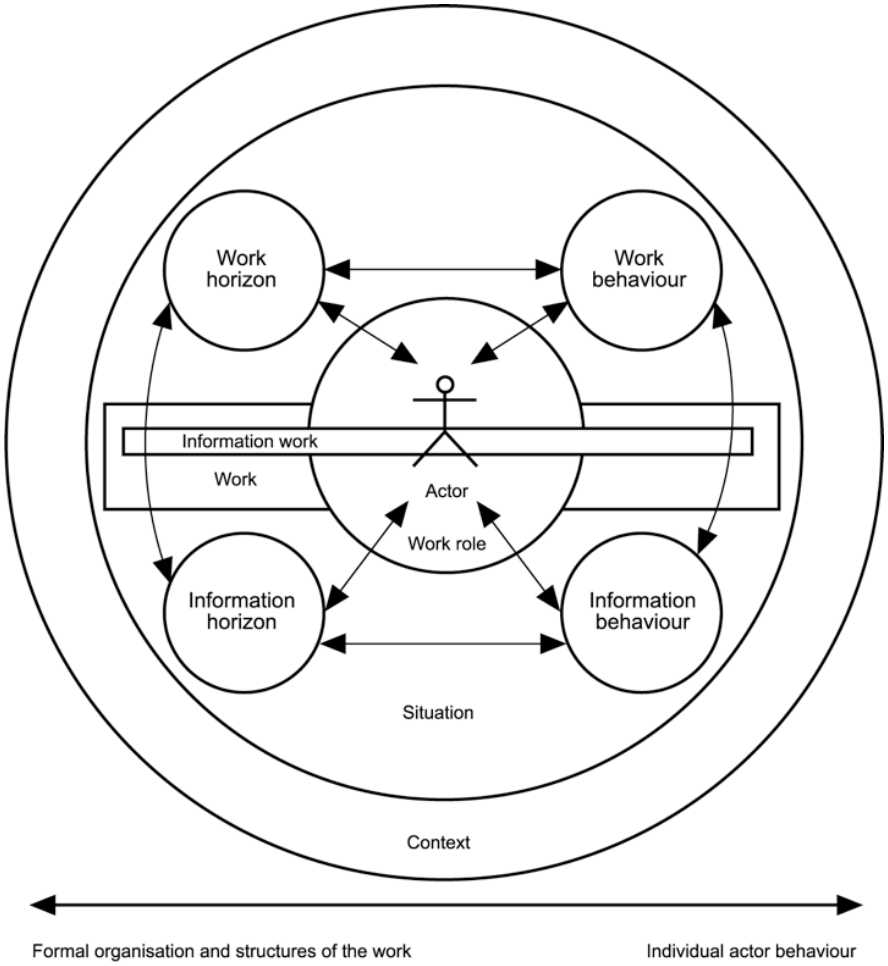
To: _____
From: _____
Date: _____

## Context

Current situation

Target condition

Root cause analysis



Theme:

# Problem

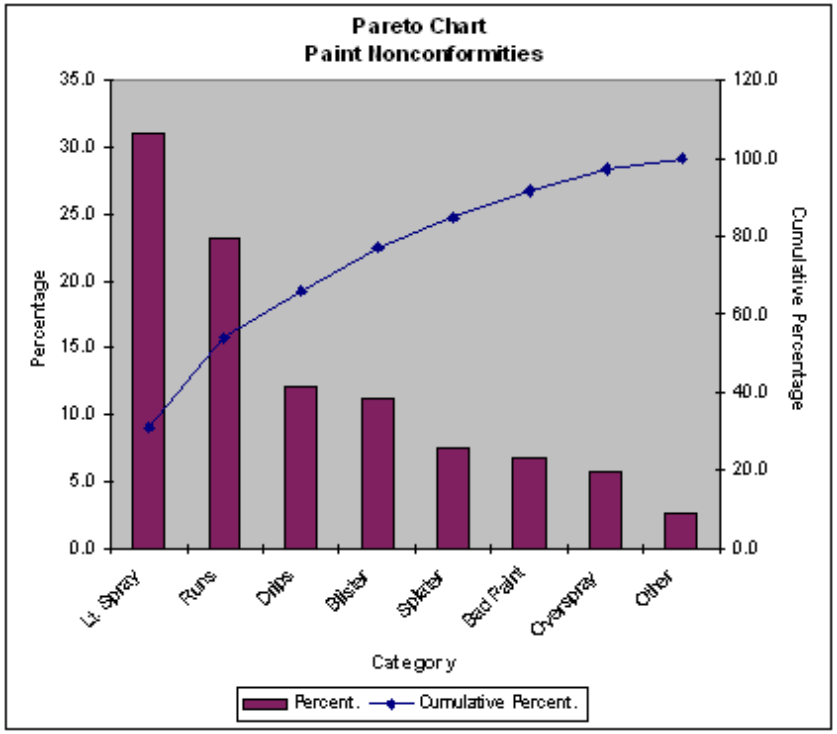
To:
From:
Date:

Background

## Current situation

Target condition

Root cause analysis



Theme:

_____
_____

# *Problem*

To: _____
From: _____
Date: _____

Background

Current situation

## Target condition

Root cause analysis

Theme:

_____

_____

# *Problem*

To: _____

From: _____

Date: _____

**Background**

**Current situation**

**Target condition**

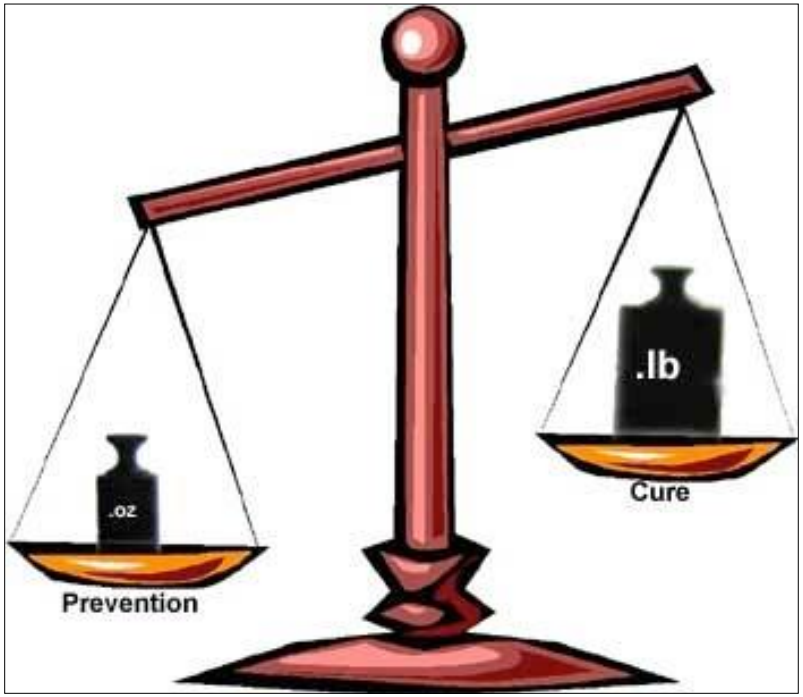

## Root cause analysis

Theme:

_____
_____

# *Problem*

To: _____
From: _____
Date: _____




## Recommendations

Implementation plan

Result and follow-up

Theme:

_____
_____

# *Problem*

To: _____
From: _____
Date: _____

Recommendations


## Implementation plan

Result and follow-up

Theme:

_____

_____

# *Problem*

To: _____

From: _____

Date: _____

Recommendations

Implementation plan

Measure that you have achieved the goal!

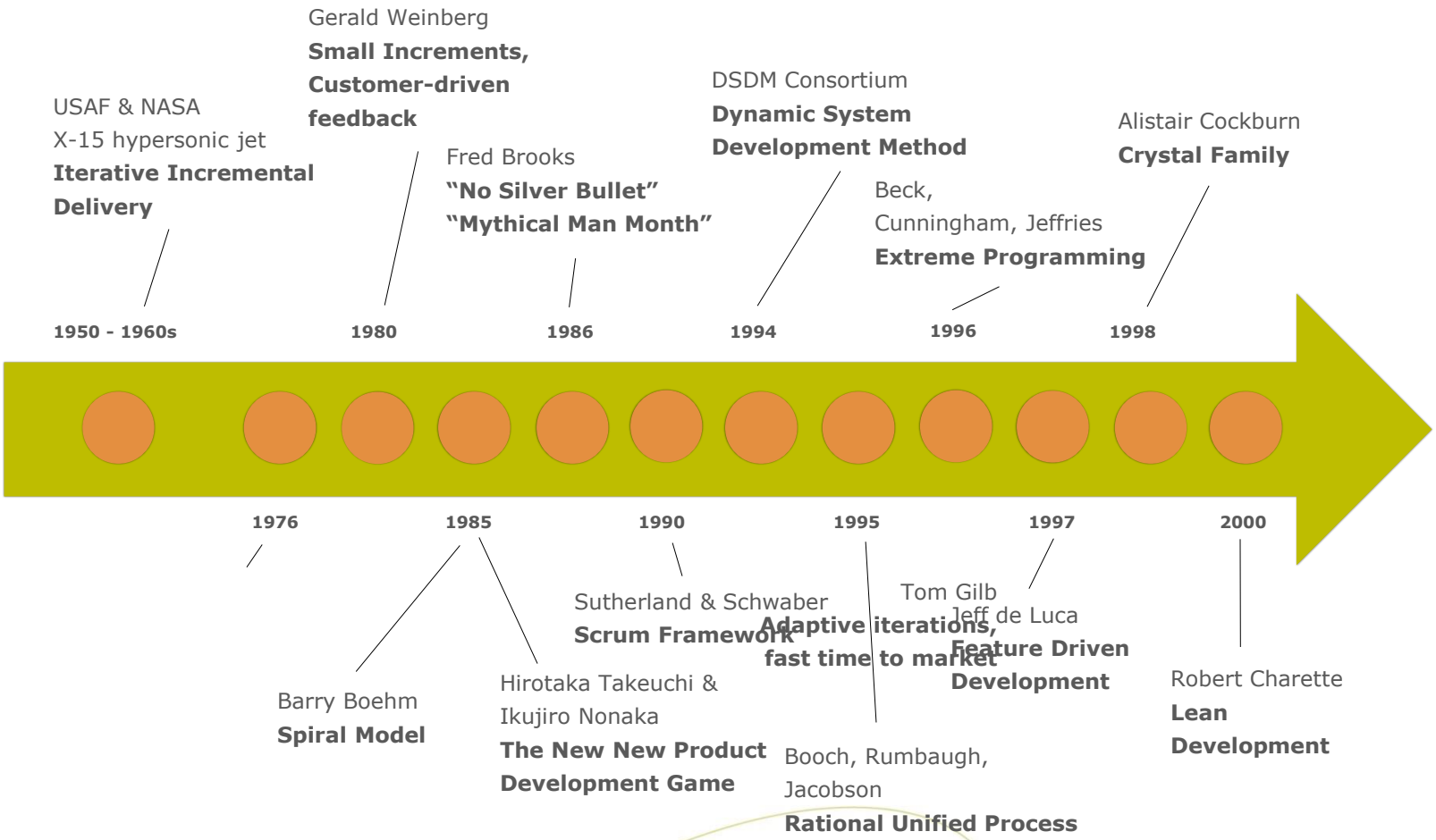
Communicate any follow-up activities.

Results and follow-up

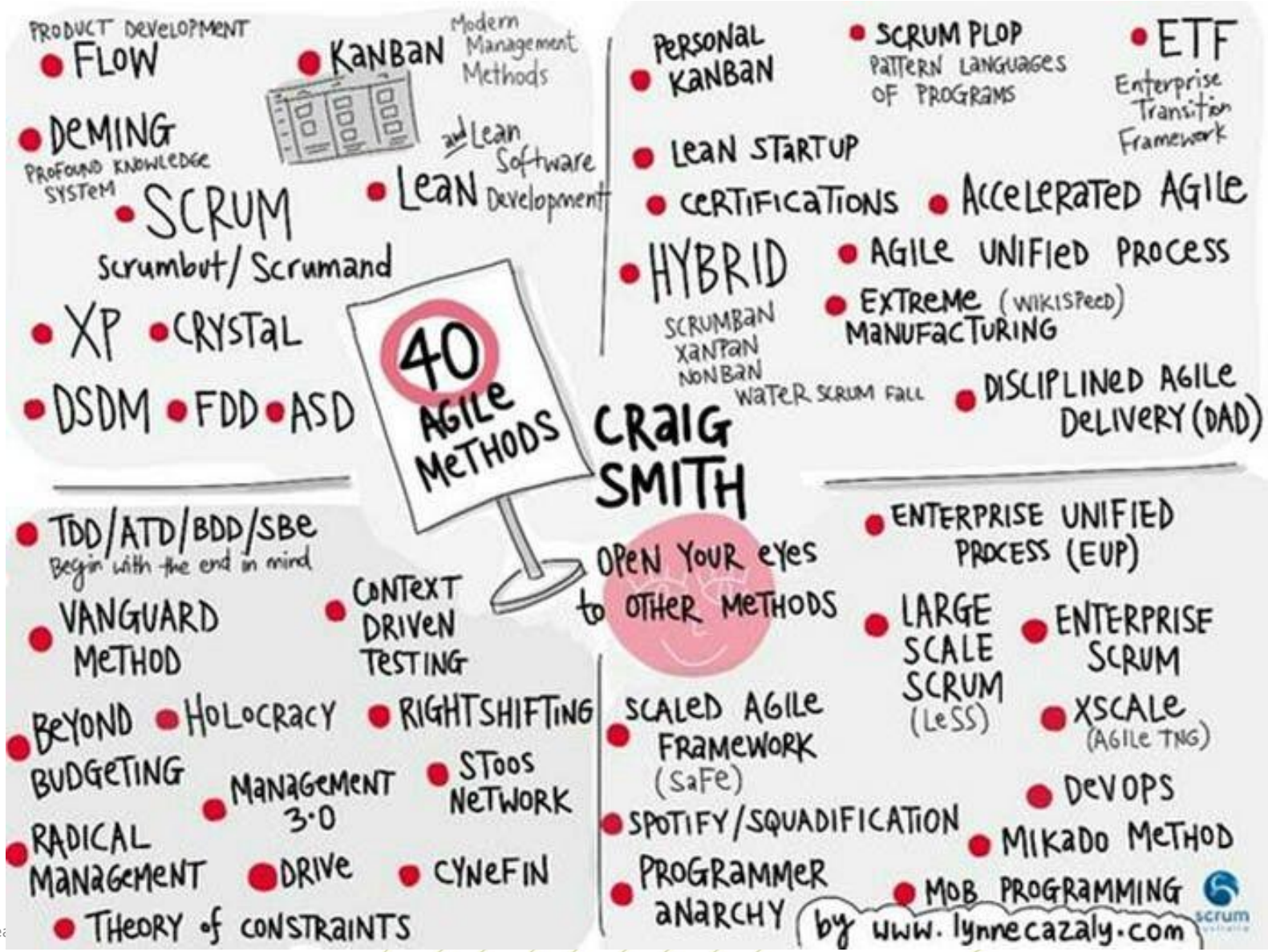

# Relation between Agile & Scrum

# Shuhari

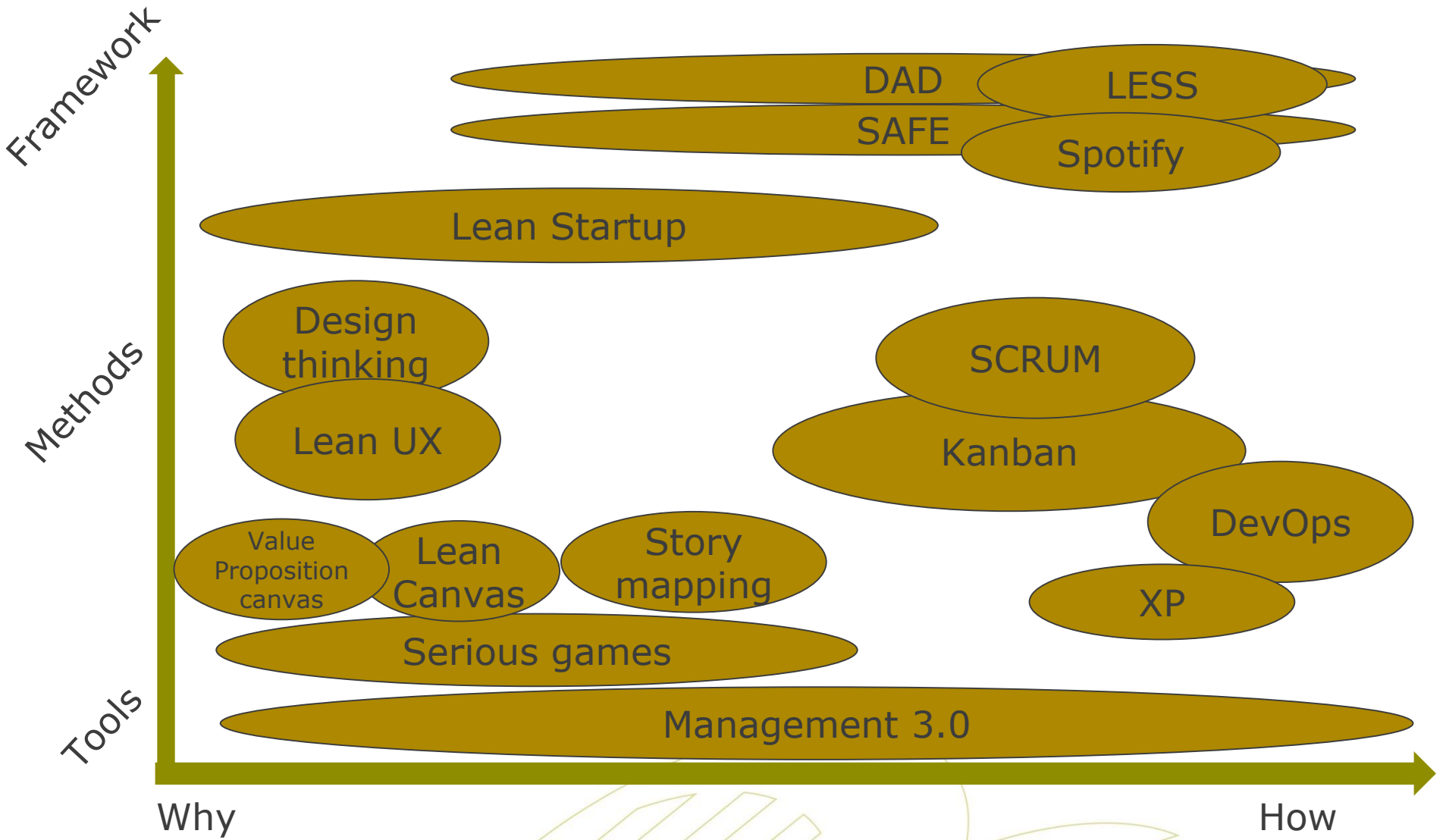
# A short history of Agility



# Many Agile Methods



# A map of Agility



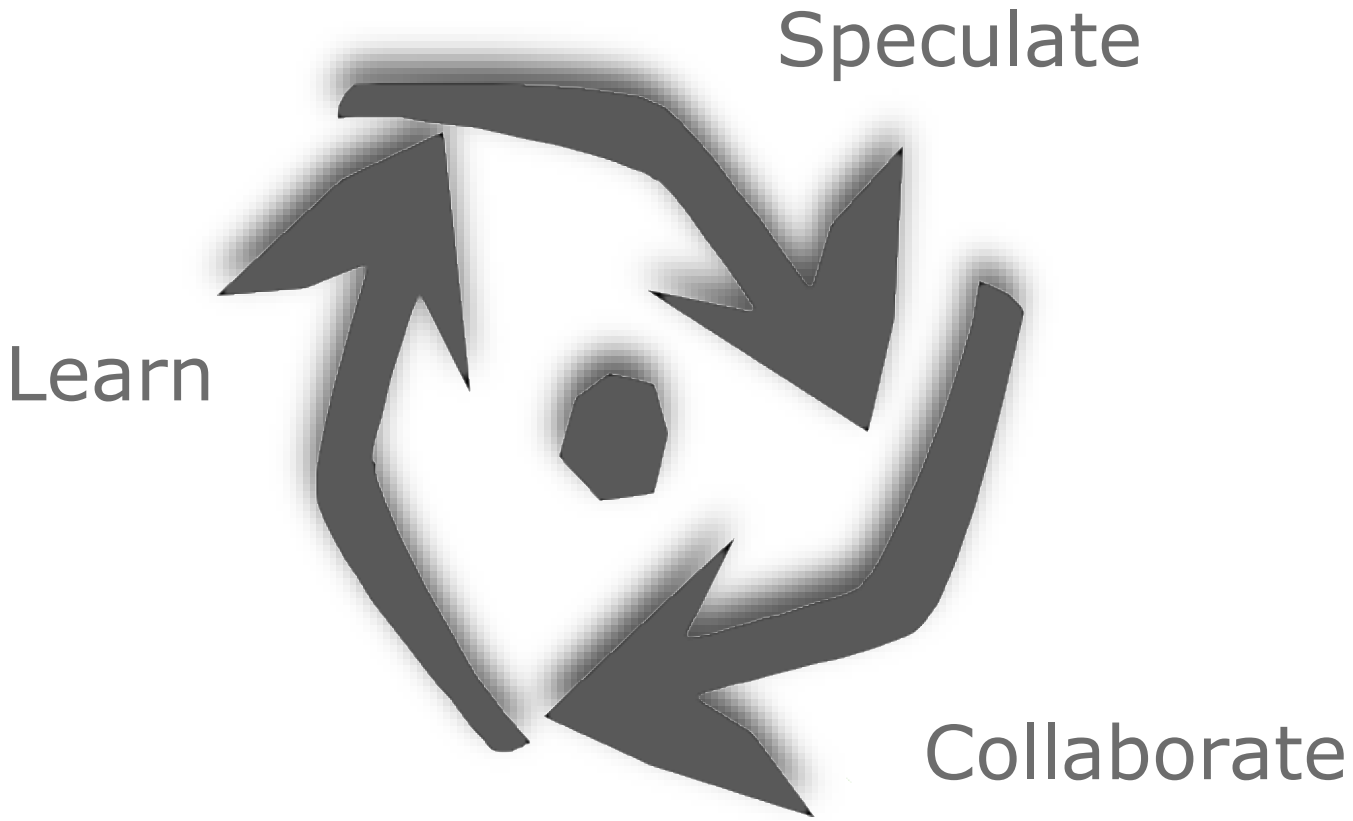
# The Agile Manifesto

# Principles behind the Agile Manifesto

cibit academy
inspearit training services

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

# The core of Agile



cibit academy
inspearit training services

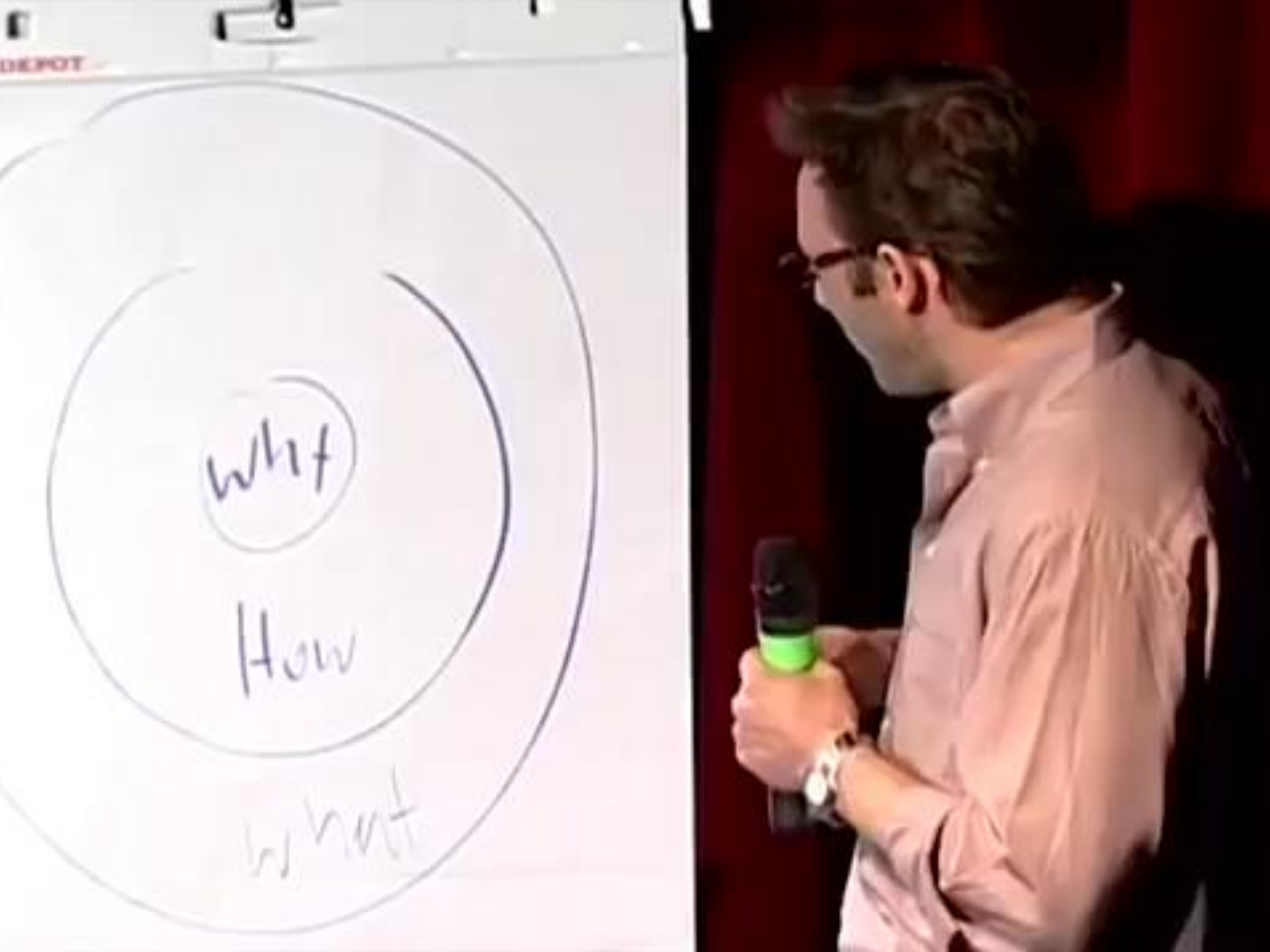
Bringing things together
and open for questions

# Beware of the Golden Circle theory

The "Why" - Everything we do, we believe in challenging the status quo, we believe in thinking differently.

The "How" - We make products that are beautifully designed and user friendly

"The What" - We just happen to make great computers - wanna buy one?


DEPOT
Why
How
What


time for questions

cibit academy
inspearit training services

# Retrospective day 2 and course evaluation

| | Strongly Agree | Agree | Di |
| --- | --- | --- | --- |
| he instructor had a nice haircut. | ○ | ○ | |
| The instructor knew how to cook. | ○ | ○ | |
| The instructor had a good singing voice. | ○ | ○ | |
| Shaving cream was used. | ○ | ○ | |
| Dogs were permitted in class. | ○ | ○ | |
| The instructor had a masculine nose. | ○ | ○ | |