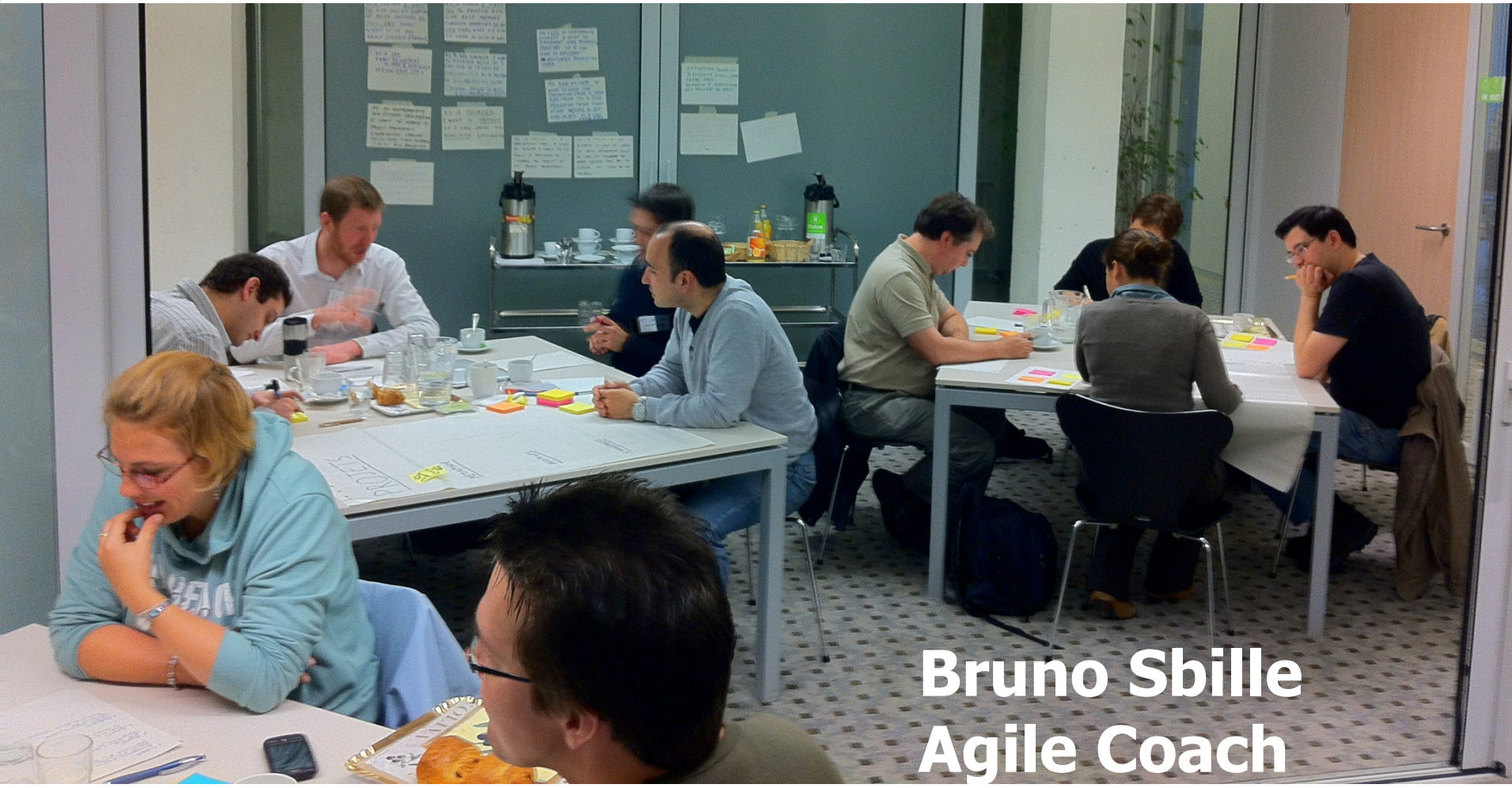


Workshop: How can agile methods accelerate your growth?

May 10th 2012 from 8:30 to 17:00 - Brussels



**Bruno Sbille
Agile Coach**

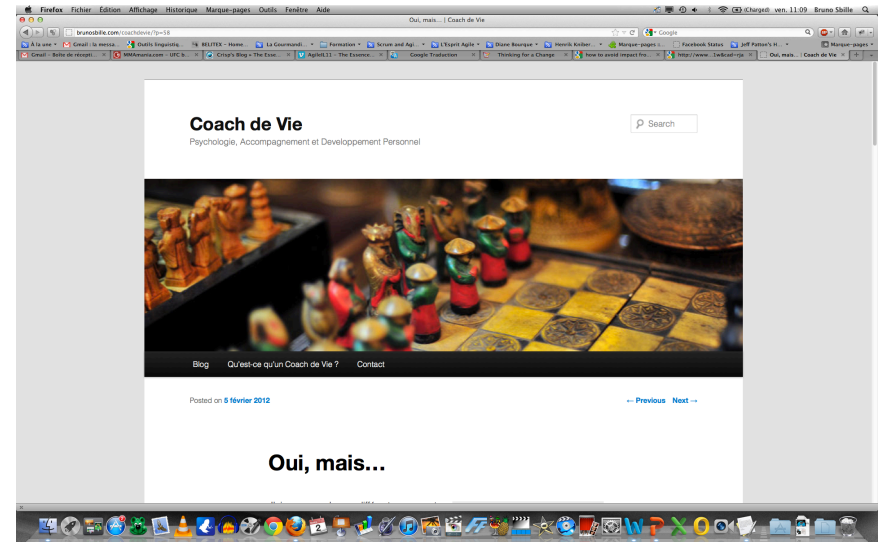
Bruno Sbille

Blog Agile: <http://brunosbille.com>



Coach & Formateur

Blog Coaching <http://brunosbille.com/coachdevie>



- **Coach & Trainer Agile Methods**
- **Trainer“Soft Skills” (Communication, Problem Solving, Management, Coaching, People Management,...)**
- **Personal Coach/ Life Coach**

bruno.sbille@gmail.com

+32 491 05 05 59

@BrunoSbille

Objectives



- **Give you an overview of Agile Methods**



- **Practice and « Live » Agile Methods**



- **Give you some concrete tools to use**

Alternative
methods

Yes, but...

Participation



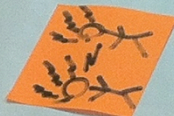
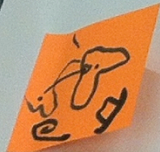
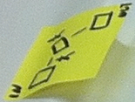
Why are you here ?

- **User Story: un outil “Agile” pour faciliter les requirements**
- **En tant que <user> je veux <action> ainsi je peux <raison>**

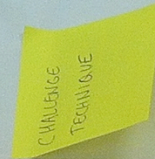
C'est quoi un projet ?

PROJETS

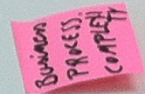
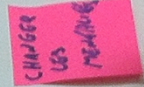
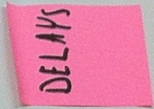
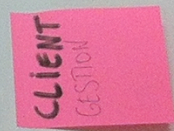
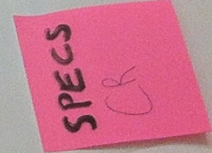
METAPHOR



POSITIVES



CHALLENGES





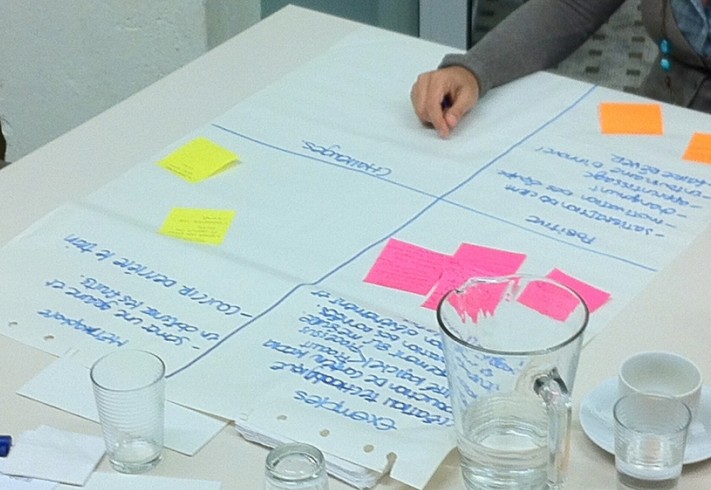


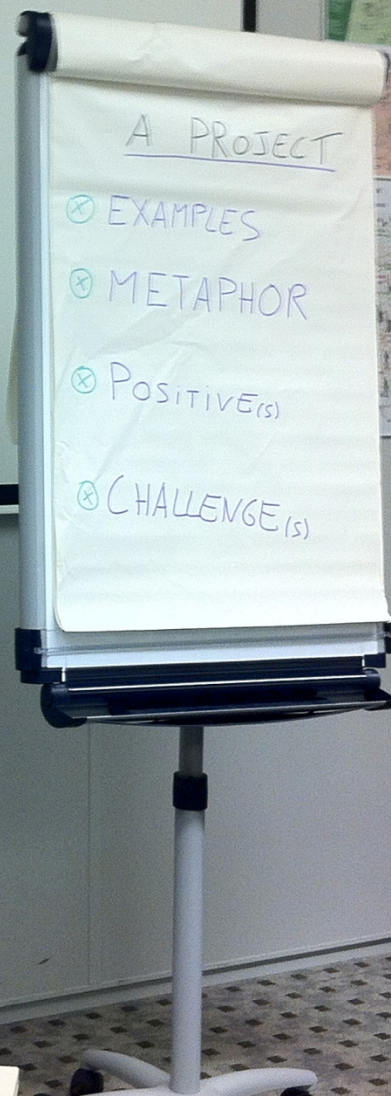
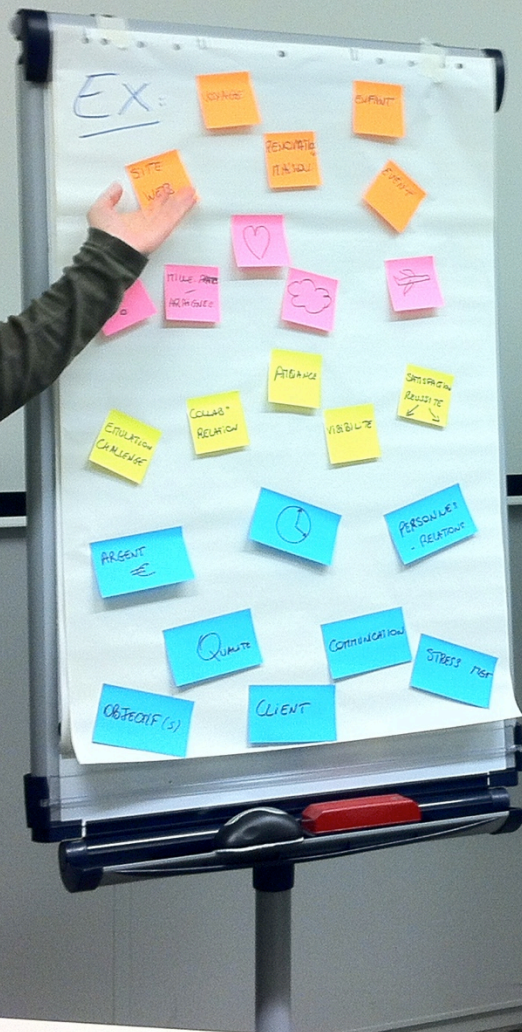
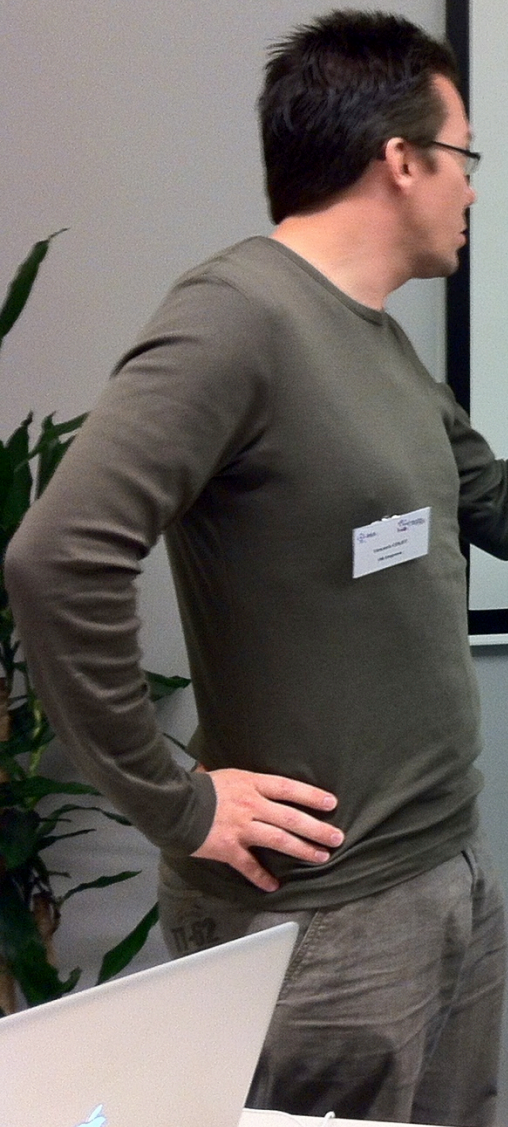
Discover Software in Brussels...

We are a cluster of companies, universities, incubators and public support organizations dedicated to the Brussels software industry.

s.com

AS A <USER> I
WANT <FEATURE>
SO I CAN <REASON>



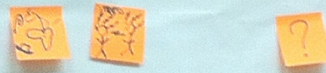




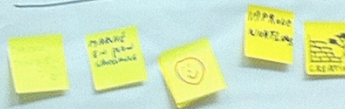
PROJETS



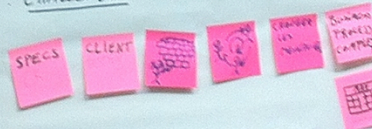
METAPHOR



POSITIVES



CHALLENGES



A PROJECT

⊗ EXAMPLES

⊗ METAPHOR

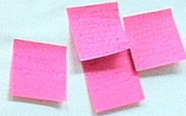
⊗ POSITIVE(S)

⊗ CHALLENGE(S)



Exemples

- création technologique
- Production de contenu média
- Qualité logicielle
- Développement sur mesure et migration des données
- organisation d'événement et mise en relation



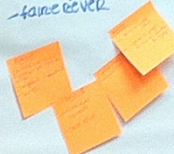
Métaphore

- Semer une graine et en observer les fruits.
- Courir derrière le train



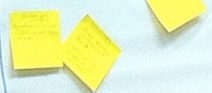
Positive

- satisfaction client
- motivation des équipes
- changement
- apprentissage
- enthousiasme d'innover
- faire rêver



Challenges

- difficulté à convaincre des bénéfices
- maîtrise des risques
- timing / deadlines
- gestion du stress
- maintenir la motivation



A PROJECT

⊗ EXAMPLES

⊗ METAPHOR

⊗ POSITIVE(S)

⊗ CHALLENGE(S)

What is all this stuff?

TDD

Scrum

XP

Kanban

Refactoring

Lean

RUP

Continuous
Integration

Pair
programming

Agile

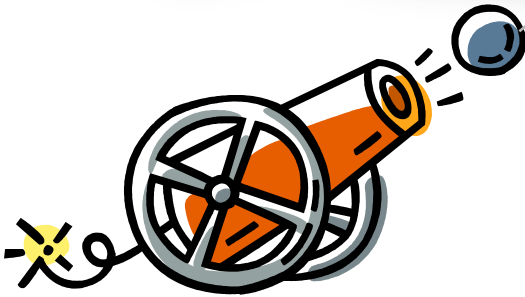
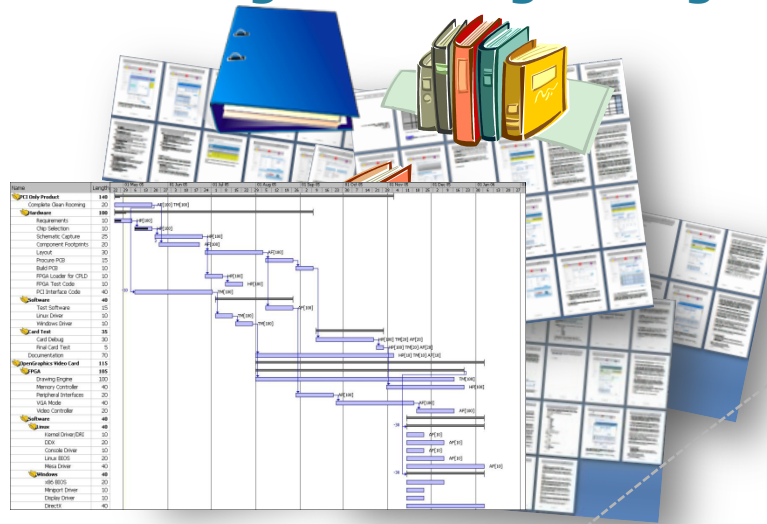


What have we learned?

Traditional SW projects are like a cannon ball

Assumptions:

- The customer knows what he wants
- The developers know how to build it
- Nothing will change along the way

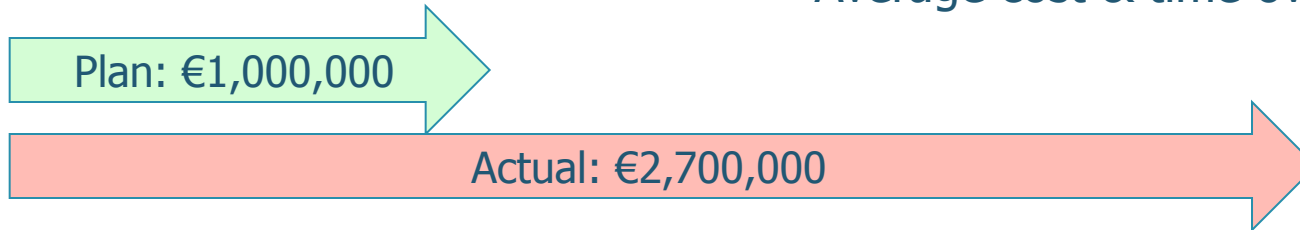


Most IT projects don't succeed

The Standish Group has studied over 40,000 projects in 10 years.

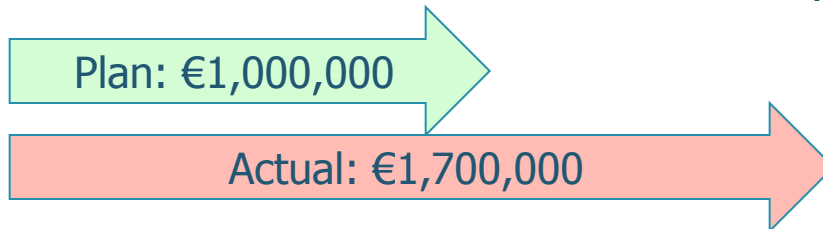
IT project success rate 1994: **15%**

Average cost & time overrun: **≈170%**



IT project success rate 2004: **34%**

Average cost & time overrun: **≈70%**

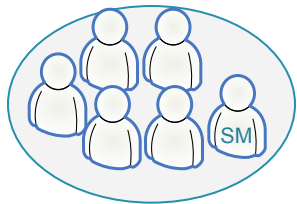


Sources:

<http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish>
<http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>

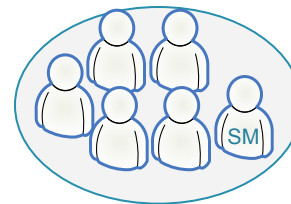
How estimates are affected by specification length

Spec



117 hrs

Same spec – more pages

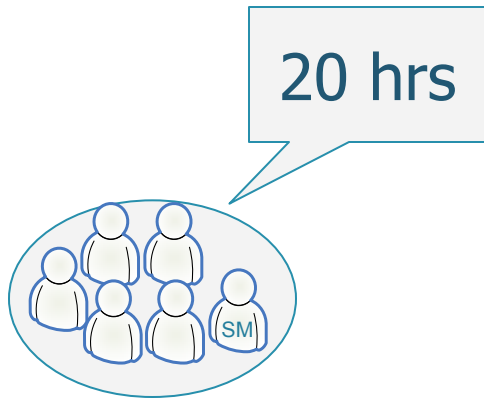
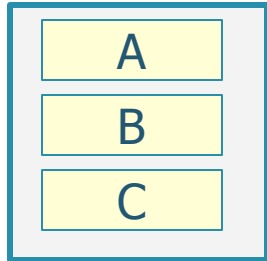


173 hrs

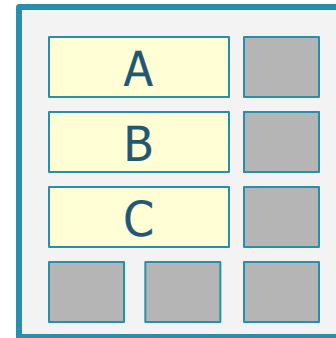
Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

How estimates are affected by irrelevant information

Spec 1



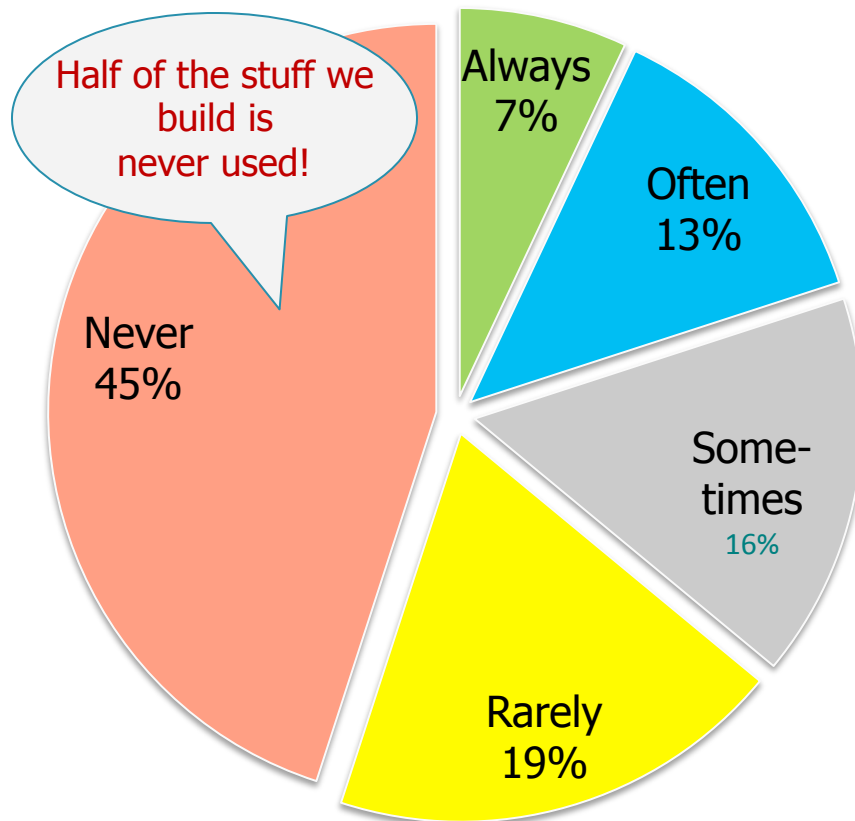
Same spec
+ irrelevant details



Source: How to avoid impact from irrelevant and misleading info on your cost estimates, Simula research labs estimation seminar, Oslo, Norway, 2006

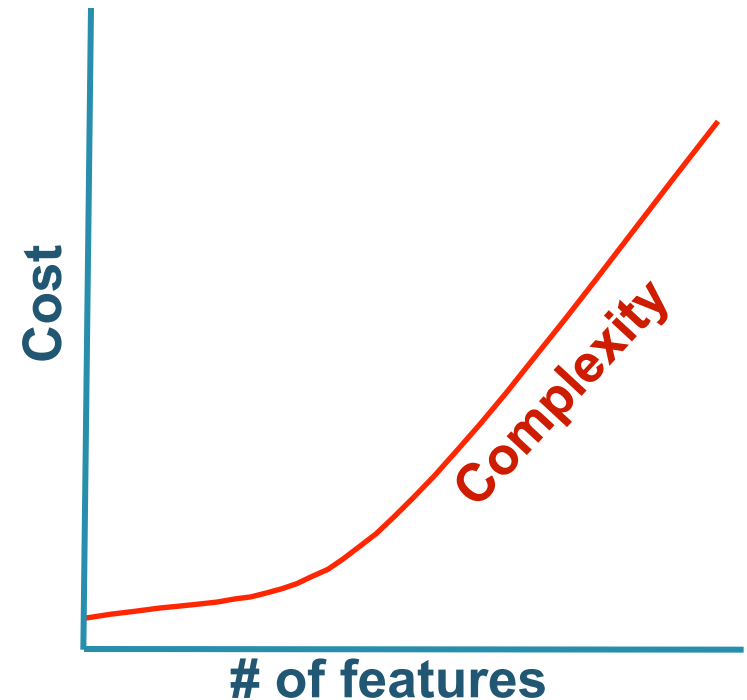
We tend to build the wrong thing

Features and functions used in a typical system



Sources:

Standish group study reported at XP2002 by Jim Johnson, Chairman



This graph courtesy of Mary Poppendieck

What have we learned?

IT project success rate 1994: 15%

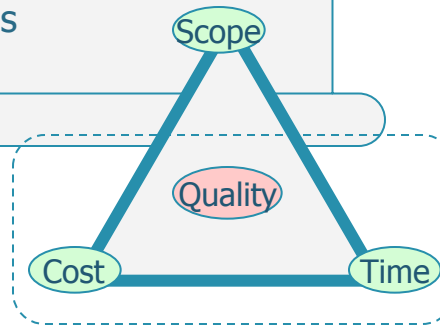
Average cost & time overrun: $\approx 170\%$

IT project success rate 2004: 34%

Average cost & time overrun: $\approx 70\%$

Top 5 reasons for success

1. User involvement
2. Executive management support
3. Clear business objectives
4. Optimizing scope
5. Agile process



"The primary reason [for the improvement] is that **projects have gotten a lot smaller.**"

"Doing projects with **iterative processes** as opposed to the waterfall method, which called for all project requirements to be defined up front, is a major step forward."

"There is no silver bullet but **agile methods** come very close"



Jim Johnson
Chairman of
Standish Group

Sources:

<http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish>

<http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>

"My Life is Failure", Jim Johnson's book

Interview: Jim Johnson of the Standish Group

Posted by [Deborah Hartmann Preuss](#) on Aug 25, 2006

Sections [Process & Practices](#), [Architecture & Design](#) Topics [Delivering Value](#), [Waterfall](#), [XP](#), [Project Management](#), [Methodologies](#), [Pair Programming](#), [Software Craftsmanship](#), [Scrum](#),
FEATURED [Agile](#), [Programming](#), [Statistics](#), [Interviews](#)

Share  |      

 [Bookmark this!](#)



CHAOS

Jim Johnson is the founder and chairman of the [Standish Group](#), a globally respected source of independent primary research and analysis of IT project performance. He is best known for his research on *why projects fail*, as well as on system costs and availability. He is also a pioneer of modern research techniques such as virtual focus groups and case-based analytical technology.

The Standish Group's claim to fame is, of course, from its interviews, on project performance of over 100,000 projects. From project failures, the major factors for failure were identified and wasted on software development for project

Taking time out from his vacation, Jim Johnson was interviewed and joined by Gordon Divitt, Executive Vice President



Top Ten Reasons for Success

- ☒ 1. User Involvement
- ☒ 2. Executive Management Support
- ☒ 3. Clear Business Objectives
- ☒ 4. Optimizing Scope
- ☒ 5. Agile Process

Agile in a nutshell

The collage features a variety of programming and software development books. The books are arranged in a grid-like fashion, with some overlapping. The background is a light gray with a faint grid pattern.

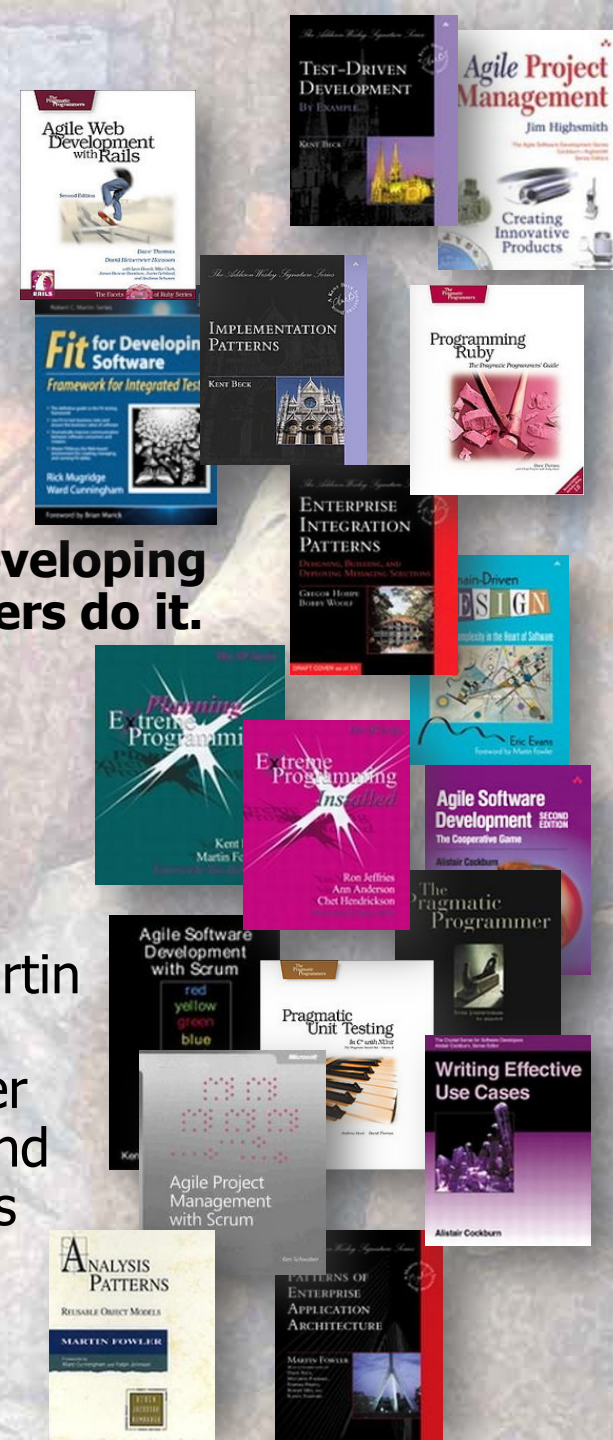
Books visible in the collage include:

- Surviving Object-Oriented Projects** by Robert C. Martin
- Agile Software Development: Principles, Patterns, and Practices** by Robert C. Martin
- Extreme Programming Adventures in C#** by Robert C. Martin
- Crystal Clear: A Human-Powered Methodology for Small Teams** by Robert C. Martin
- Pragmatic Unit Testing in Ruby** by Robert C. Martin
- Everyday Scripting in Ruby** by Robert C. Martin
- Refactoring Databases: Evolutionary Database Design** by Robert C. Martin
- Extreme Programming Explained: Embrace Change** by Robert C. Martin
- JUnit: A Pocket Guide** by Robert C. Martin
- Test-Driven Development: By Example** by Robert C. Martin
- The Wiki Way: Quick Collaboration on the Web** by Robert C. Martin
- UML Distilled: Third Edition: A Brief Guide to the Standard Object Modeling Language** by Robert C. Martin
- Refactoring: Improving the Design of Existing Code** by Robert C. Martin

We are uncovering better ways of developing software by doing it and helping others do it.

Snowbird ski resort, Utah

Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas



Agile Manifesto

www.agilemanifesto.org

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**

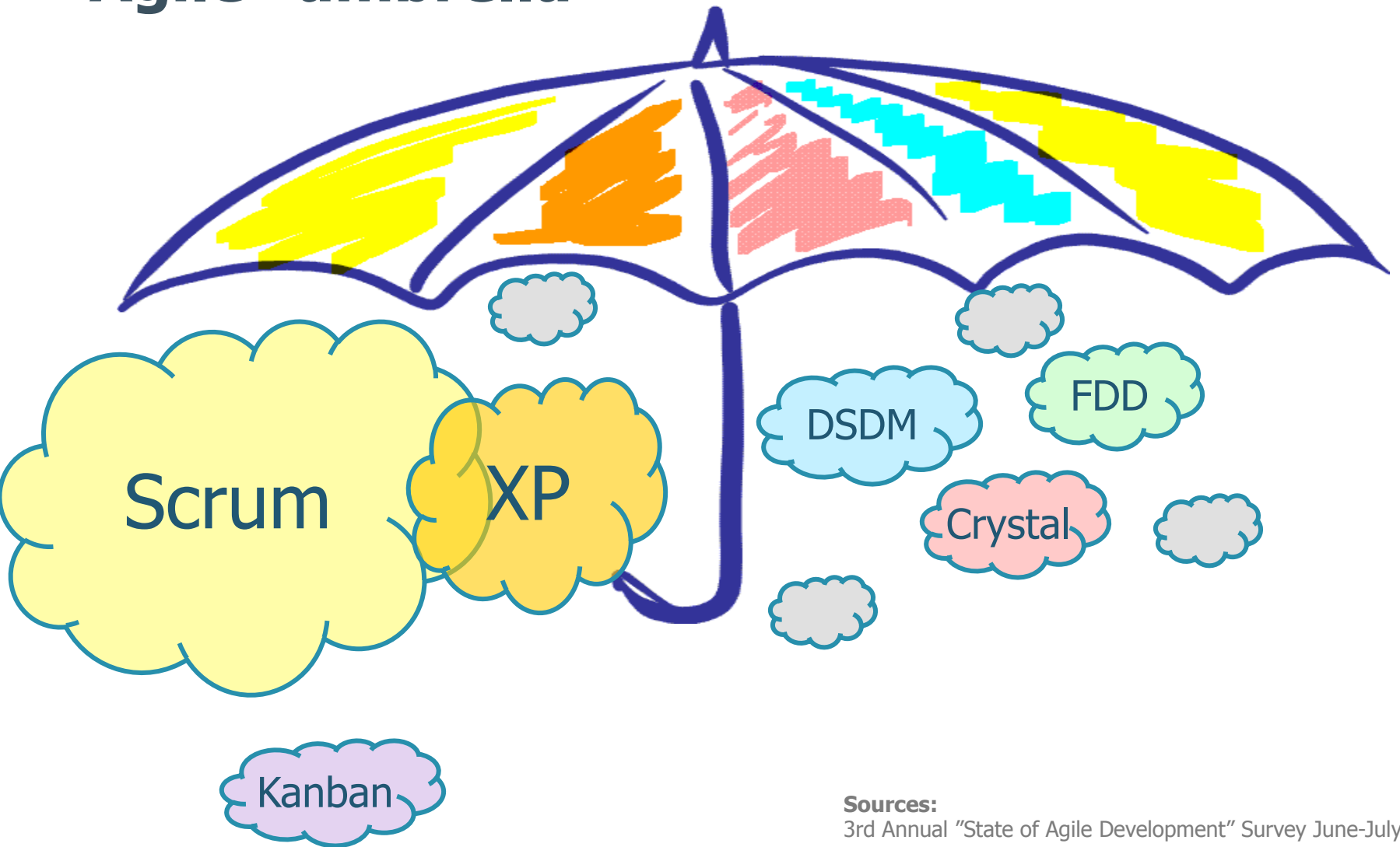
Responding to change over **following a plan**

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

Principles behind the Agile Manifesto

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

Agile "umbrella"



Sources:

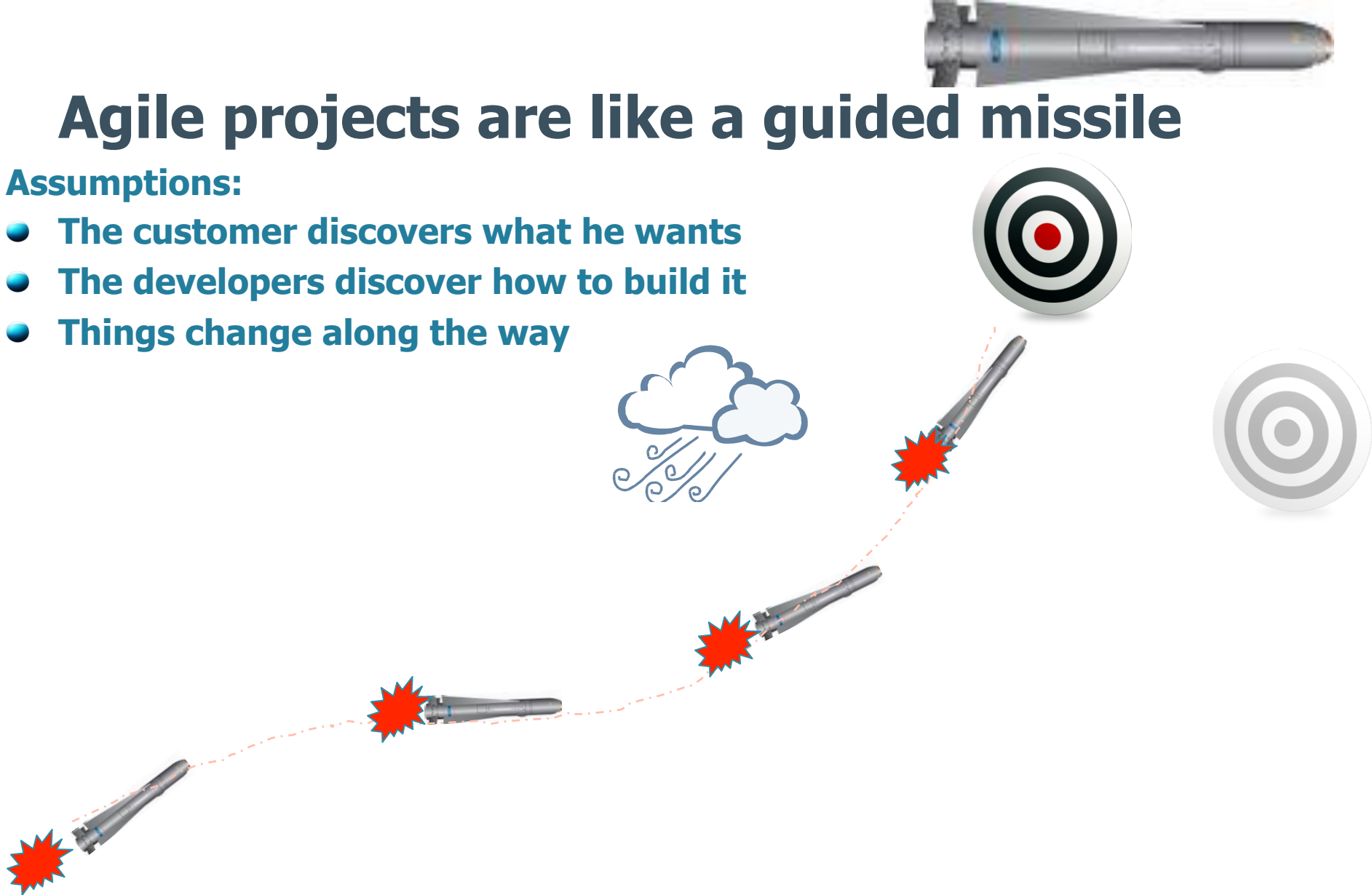
3rd Annual "State of Agile Development" Survey June-July 2008

- 3061 respondents
- 80 countries

Agile projects are like a guided missile

Assumptions:

- The customer discovers what he wants
- The developers discover how to build it
- Things change along the way



Timeboxing

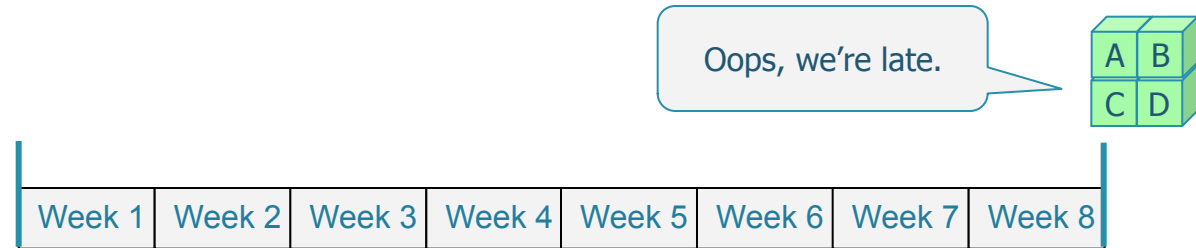
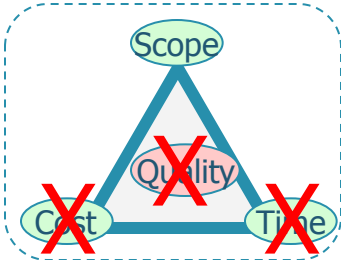
Plan

(doomed to fail, but we don't know it yet)



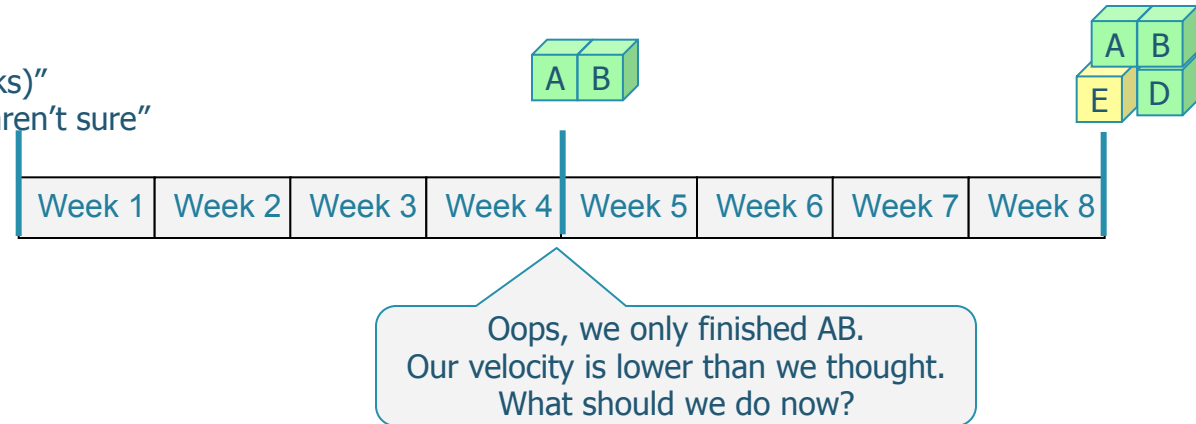
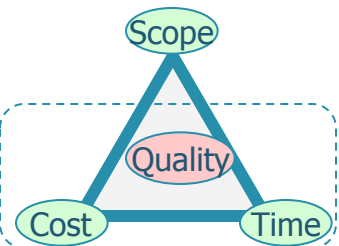
Traditional scenario

"We will deliver ABCD in 4 weeks"



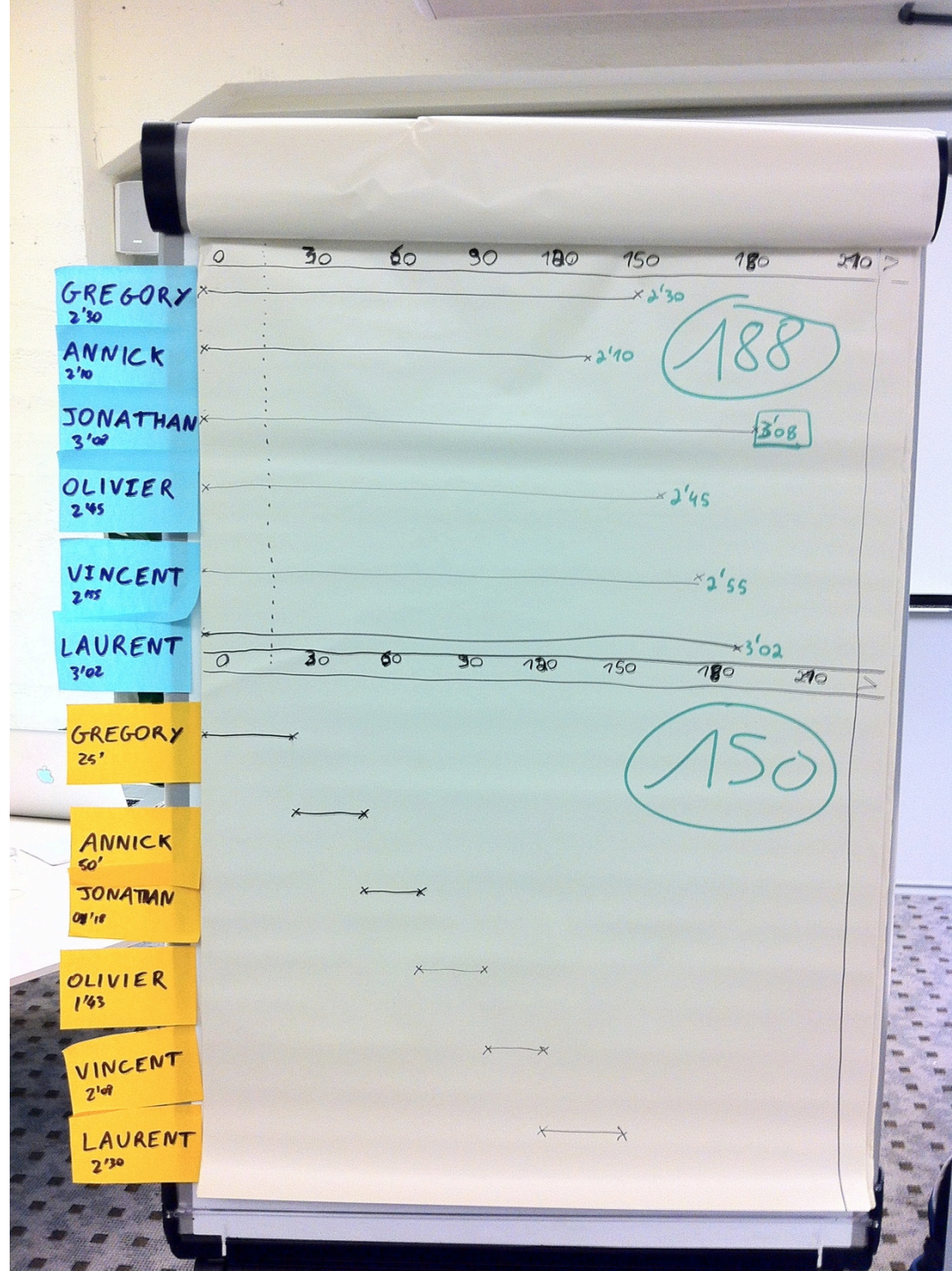
Agile scenario

"We always deliver something every sprint (4 weeks)"
"We *think* we can finish ABCD in 1 sprint, but we aren't sure"
"We always deliver the most important items first"



The Name Game

- Introduction to KanBan
- <http://www.crisp.se/henrik.kniberg/multitasking-name-game>



Planning is easier with frequent releases



Scrum in a nutshell

Who is using Scrum ?

Google™

Gartner®



Borland®
THE OPEN ALM COMPANY



TOYOTA



NOKIA
Connecting People

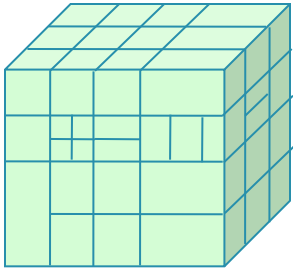


PHILIPS
sense and simplicity

SIEMENS

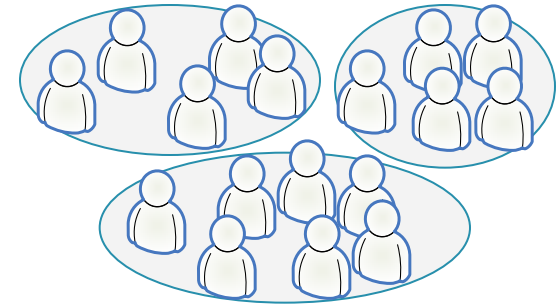
Scrum in a nutshell

Split your product

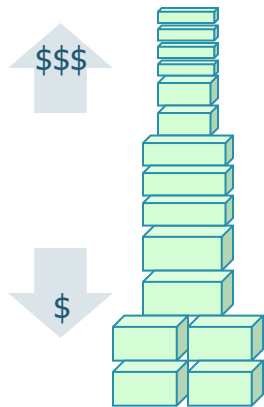


~~Large group~~ spending ~~a long time~~ building a ~~huge thing~~
Small team spending a little time building a small thing
... but integrating regularly to see the whole

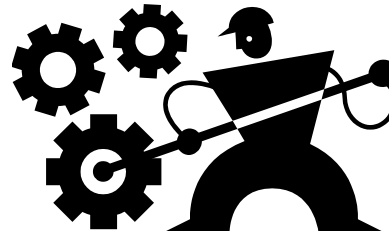
Split your organization



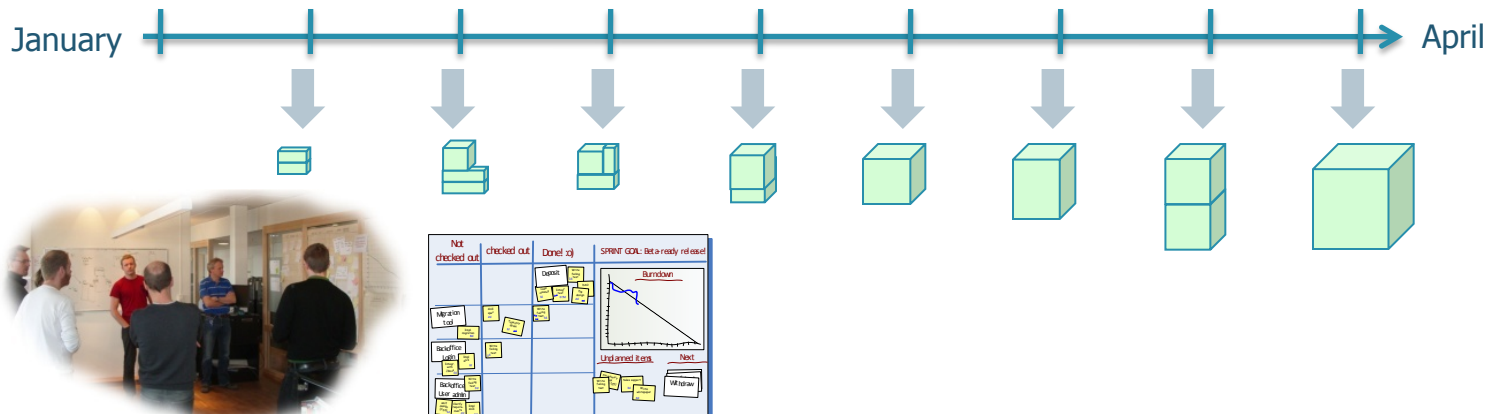
Optimize business value



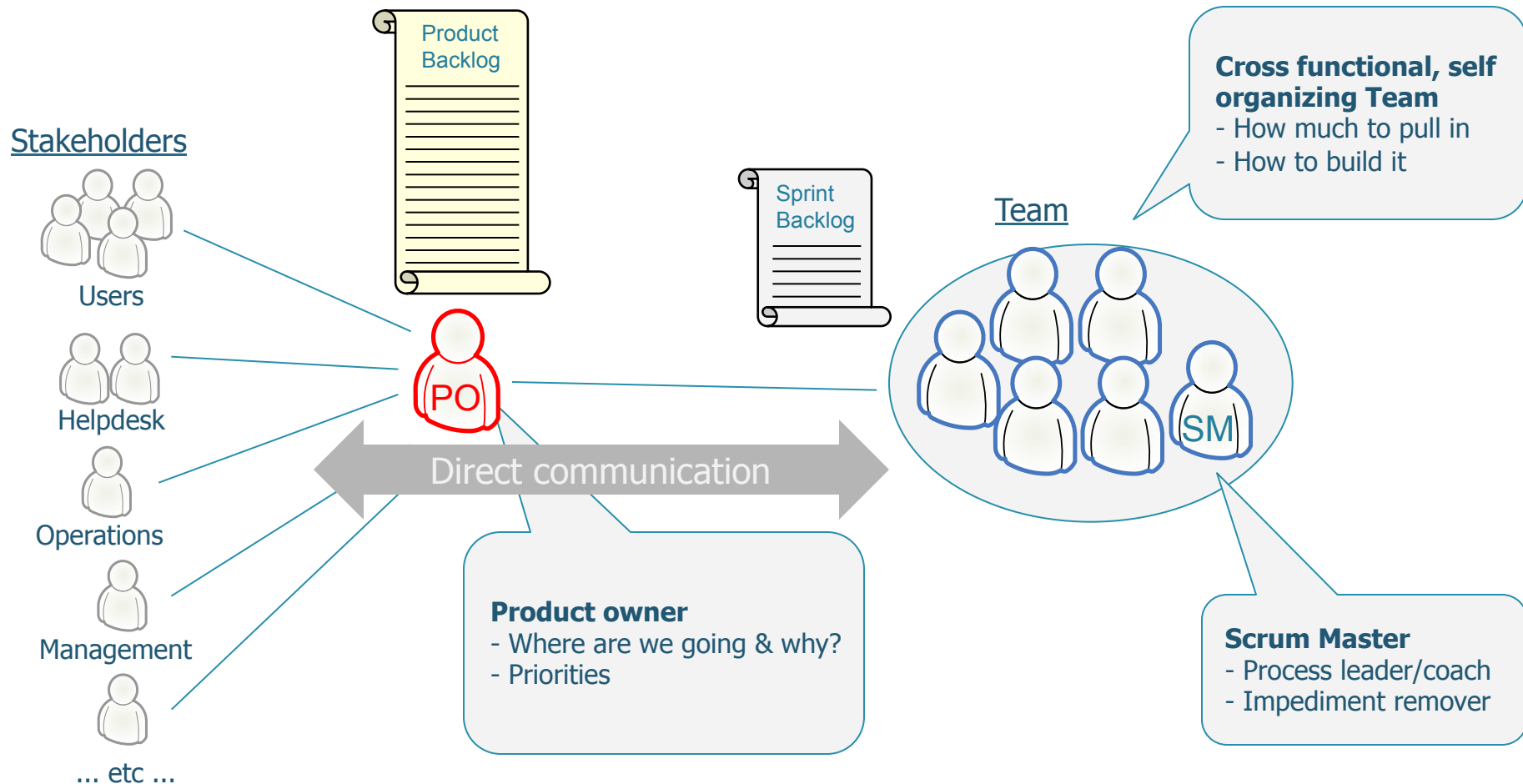
Optimize process



Split time



Scrum overview – structure



Product backlog

Product vision

As a <stakeholder>
I want <what>
so that <why>

Product Backlog

As a buyer
I want to **save my shopping cart**
so that I can continue shopping later

As a booker
I want to receive **notifications** when
new available slots appear in the
calendar
so that I don't have to keep checking
manually

(... etc ...)

Definition of Done

- Releasable
 - User Acceptance tested
 - Merged to Trunk
 - release notes written
 - No increased technical debt

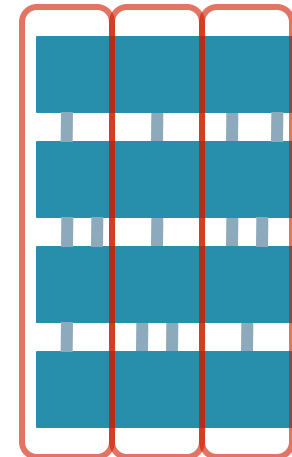
= I haven't messed up
the codebase

GUI

Client

Server

DB



Agile estimating strategy

- **Don't estimate time.**
 - Estimate *relative size* of stories.
 - Measure velocity per sprint.
 - *Derive* release plan.
- **Estimates done by the people who are going to *do the work*.**
 - Not by the people who want the work done.
- **Estimate continuously during project, not all up front.**
- **Prefer verbal communication over detailed, written specifications.**
- **Avoid false precision**
 - Better to be roughly right than precisely wrong

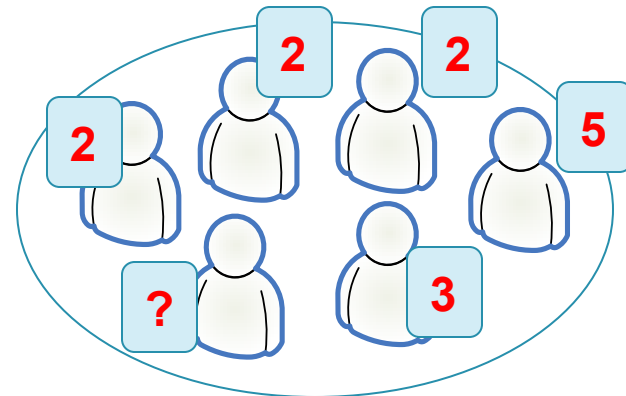
Planning poker

As a buyer
I want to **save my shopping cart**
so that I can continue shopping later

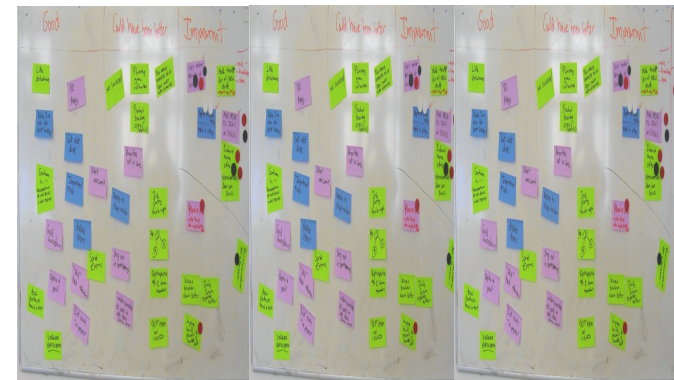
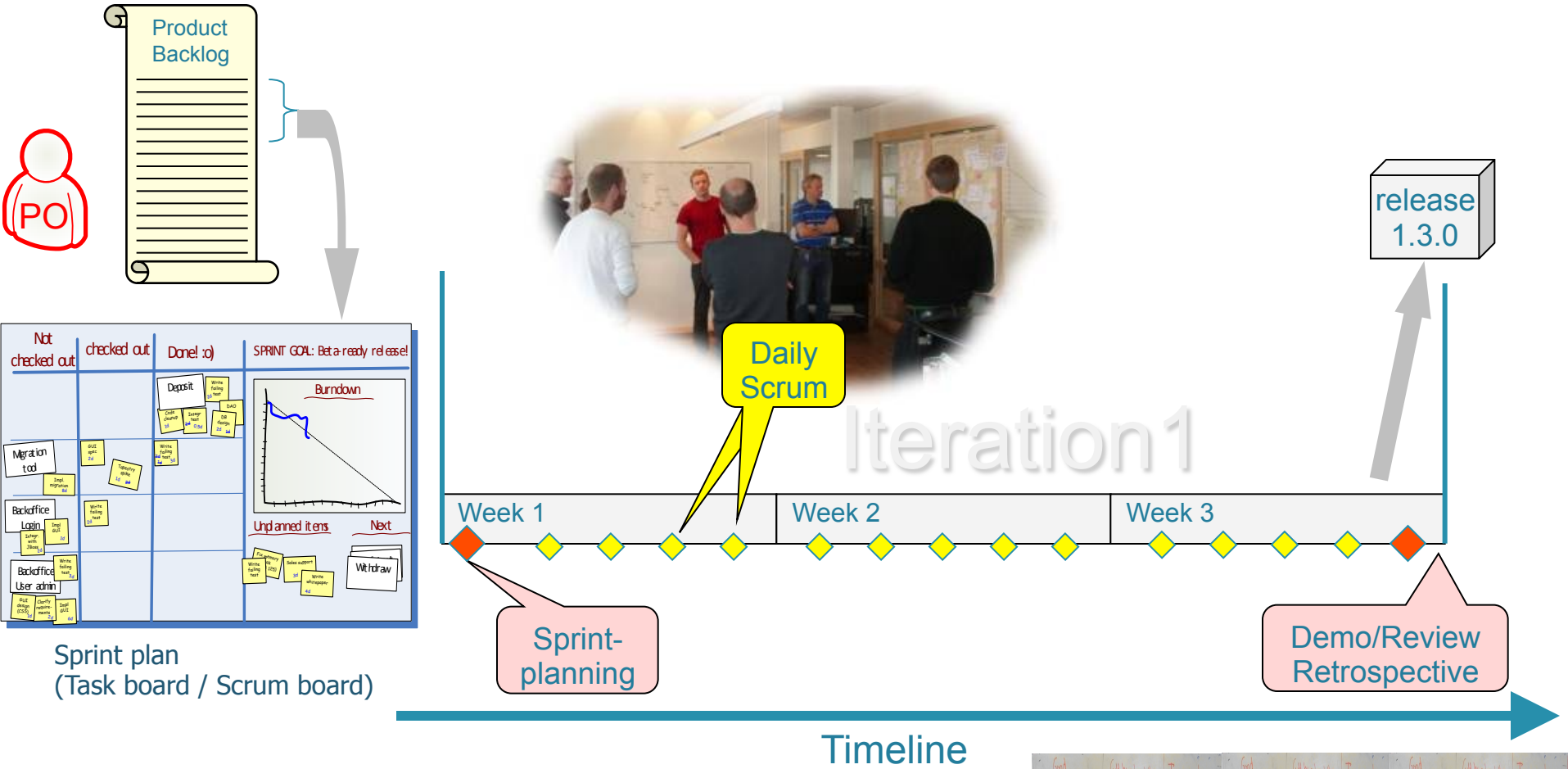
2

As a booker
I want to receive **notifications** when
new available slots appear in the
calendar
so that I don't have to keep checking
manually

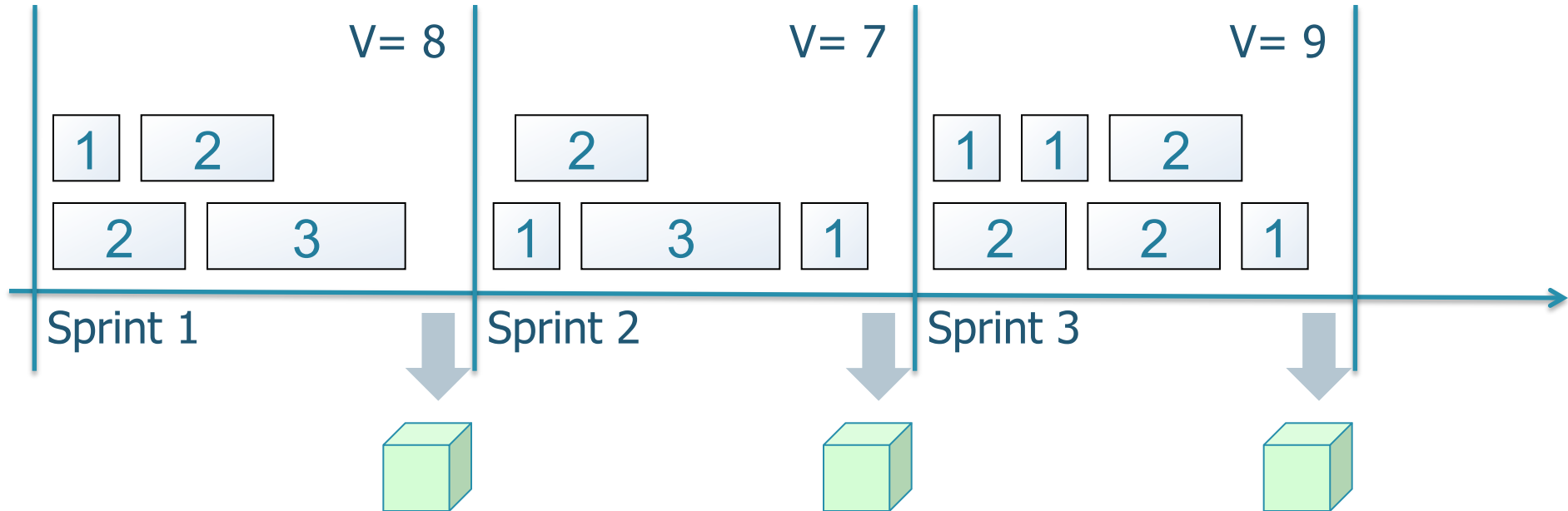
5



Typical sprint



Velocity



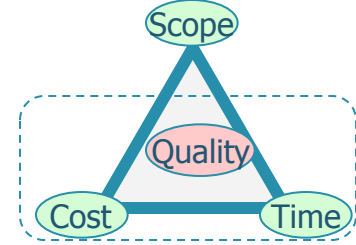
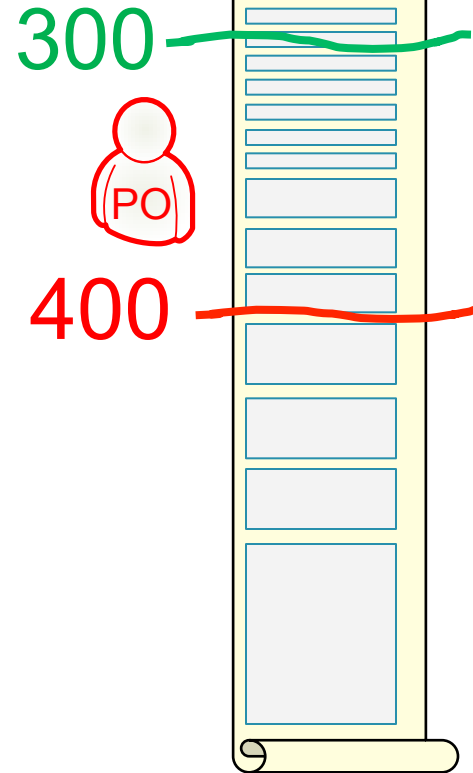
Likely future velocity:
7-9 per sprint

Release planning – fixed date

- Today is Aug 6
- Sprint length = 2 weeks
- Velocity = 30 - 40

What will be done
by X-mas?

(10 sprints)



The Ball Point Game



01:08

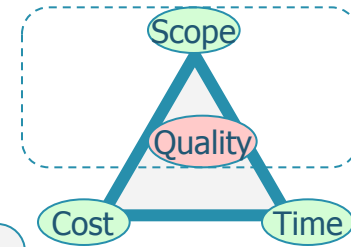
- AGILE VALUES
- ① INDIVIDUALS & INTERACTIONS
OVER PROCESSES & TOOLS
 - ② WORKING SOFTWARE
OVER COMPREHENSIVE DOCUMENTATION
 - ③ CUSTOMER COLLABORATION
OVER CONTRACT NEGOTIATION
 - ④ RESPONDING TO CHANGE
OVER FOLLOWING A PLAN



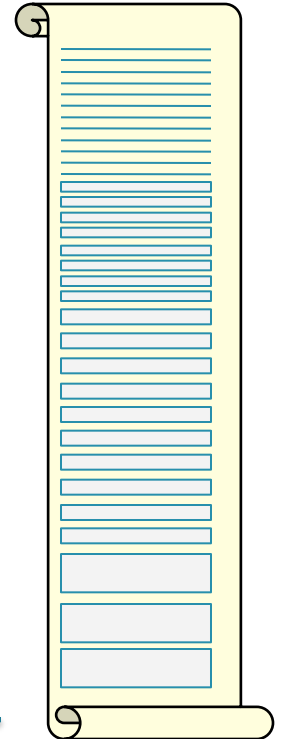
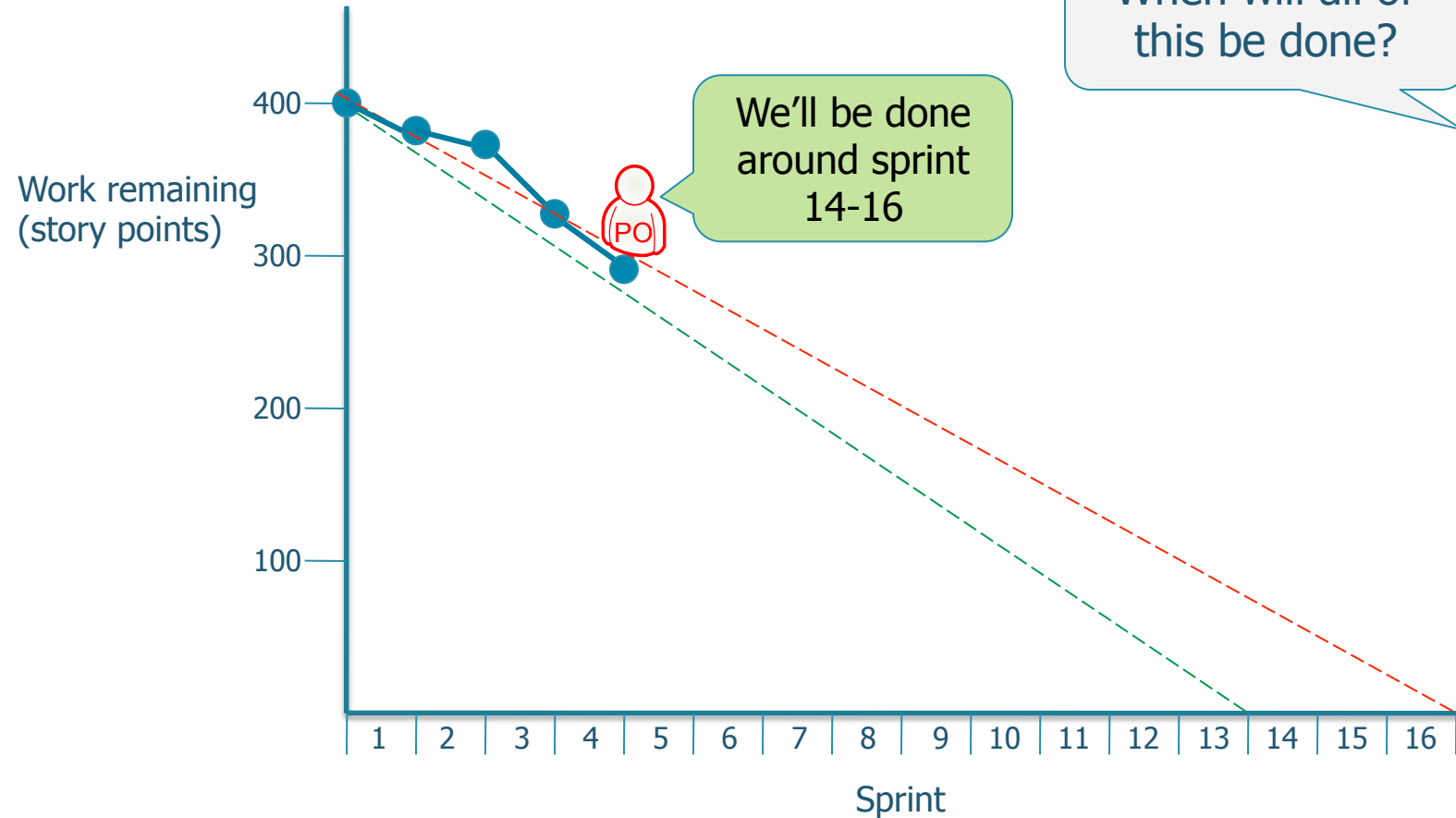
Le jeu des balles ce que ça nous apprend

- **Le Processus Scrum basé sur le cycle de Deming:**
http://fr.wikipedia.org/wiki/Roue_de_Deming
- **Le principe "Inspect and Adapt" Faites qqch puis, debriefing sur l'expérience.**
- **On a tendance à s'ajouter des contraintes**
- **Principe du bottleneck (**
http://fr.wikipedia.org/wiki/Th%C3%A9orie_des_contraintes**)**
- **Une équipe de 15 personnes, C'est 15 cerveaux plein d'idées**

Release planning – fixed scope



Release burndown chart



Benefits

- **Quickly on market**
- **Quality**
- **Flexibility**
- **The right product**
- **Visibility**
- **Cost control**
- **Less risks**
- **More motivations**
- **Bring quickly earnings**



Risks / Challenges

- **Management has an important role**
- **Agile methods require new or changed staffing and HR practices, including key business personnel participation on the team.**
- **The shift to agile development is more about culture than it is about process**
- **Change of culture**
- **Invest in trainings early**
- **Management understand and support Agile**
- **Bring a lot of transparency**
- **Support, Trust and Motivate the teams**

Take-away points

- **Waterfall is still waterfall even if you call it Scrum**
 - Know your tools, get training & coaching early.
- **Don't believe your plan**
 - There is no "the plan must be right because we promised".
 - Make sure you have reliable feedback loops & a changeable plan.
- **There is no "too low velocity"**
 - Just *actual* velocity, and a realistic or unreleastic plan.
- **Build quality in**
 - Don't postpone test & integration, that gives a false velocity.
- **Having good people isn't enough**
 - An inappropriate process can cause even a great team to fail.
- **Dramatic improvements can be made quickly**
 - With a strong management team that has access to empirical data and is willing to focus.

Si tu veux découvrir ce qu'est la Matrice, tu devras l'explorer toi même.

Choisis la pilule bleue et tout s'arrête : après tu pourras faire de beaux rêves et penser ce que tu veux.

Choisis la pilule rouge, tu restes au pays des merveilles, et on descend avec le lapin blanc au fond du gouffre. N'oublie pas : je ne t'offre que la vérité rien de plus.



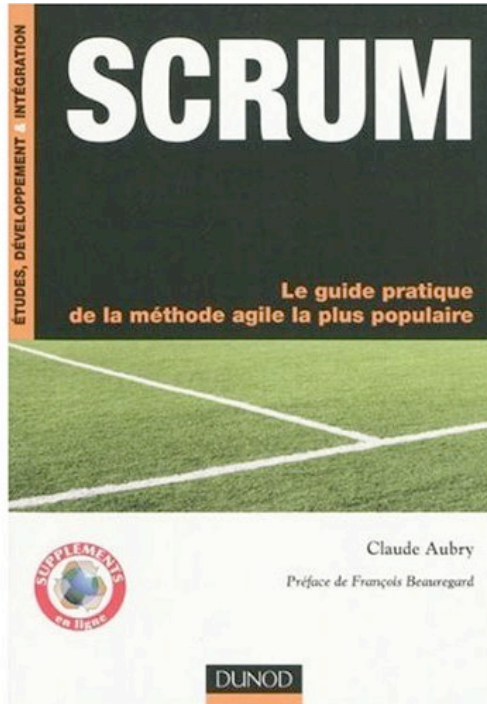
Sources

- **The Essence of Agile (Keynote by Henrik Kniberg)**
- <http://www.crisp.se/henrik.kniberg/multitasking-name-game>

En savoir plus

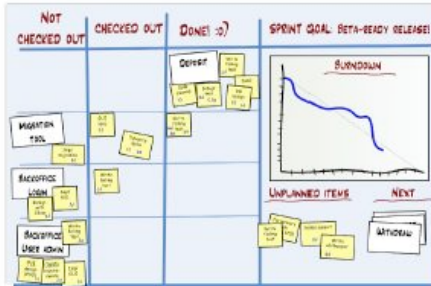
- www.mountangoatsoftware.com (EN)
- www.allaboutagile.com (EN)
- www.aubryconseil.com (FR)
- borisgloger.com/en/2008/07/01/product-backlog-templates-scrum-tools/ (EN - templates)
- www.xqa.com.ar/visualmanagement/ (EN)
- www.brunosbille.com (FR-EN)

En savoir plus



Scrum et XP depuis les Tranchées

Comment nous appliquons Scrum



Ecrit par :
Henrik Kniberg



Scrum methodology

by Soul'

10 months ago



For tasks estimation we used "planning poker" method.

<http://vimeo.com/4587652>

CONTACT

- bruno.sbille@gmail.com
- **+32 491 05 05 59**
- **@BrunoSbille**