

Scrum



Manifesto for Agile Software Development

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

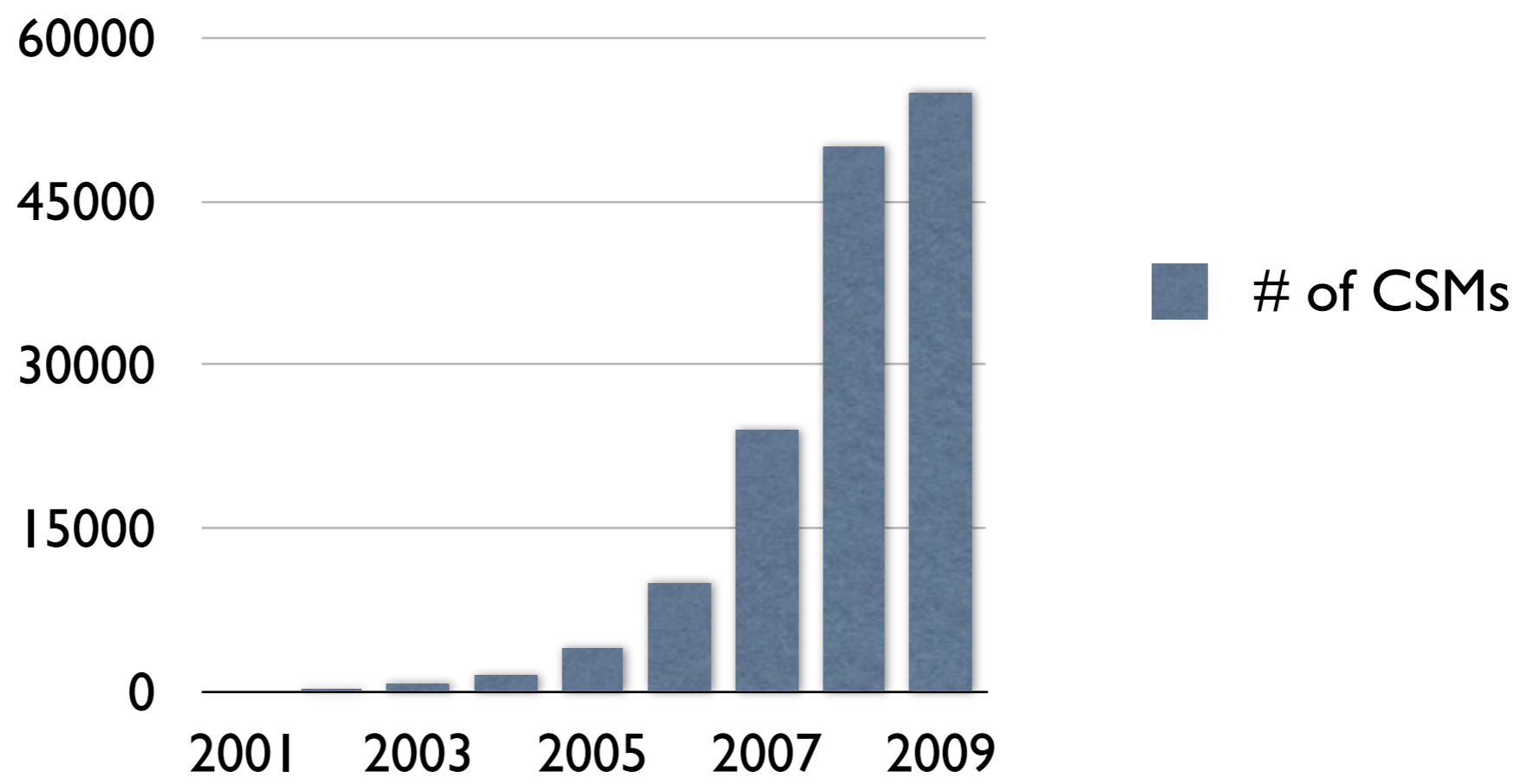
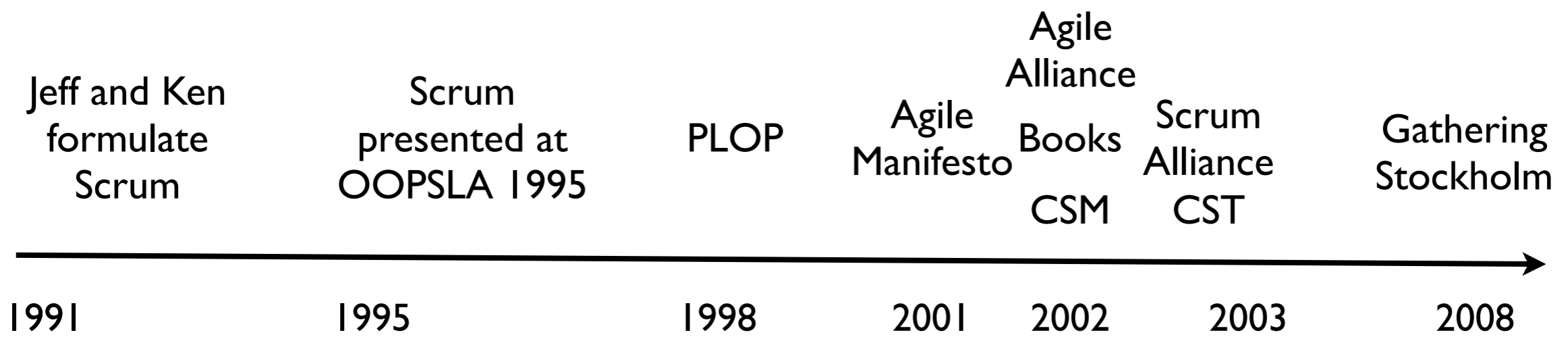
Responding to change over following a plan

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



In 2008, 84% of all Agile projects used Scrum

Source: December 2008 Global Agile Company Online Survey



Great News

Time Boxes being
widely used

Waterfall being
used less

Agile values are
public

Not So Great News⁵ (underneath waterfall we found)

Developers have trouble with
increments

Customers would rather throw
the responsibility over the
wall

IT middle-management is
resistant

Transparency is avoided

Command and control is
prevalent

Scrum

Empirical process for managing the development and deployment of complex products.

Empiricism is dependent on frequent **inspection and adaptation** to reach goal.

Inspection is dependent on **transparency**.

Scrum rests on the four legs of **iterative** development that generates done **increments** of functionality using **self-managing teams** that are **cross-functional**.

Scrum Values

Transparency

Empiricism

Self-Organization

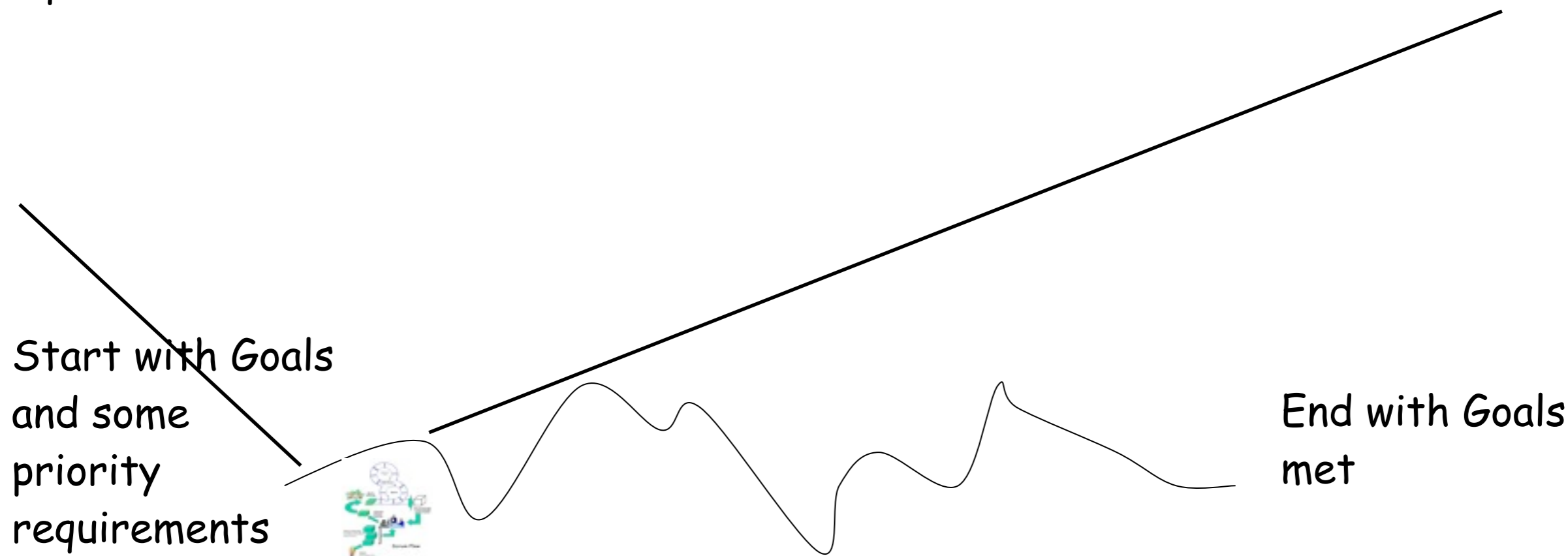
Integrity

Delivery

Defined, Predictive

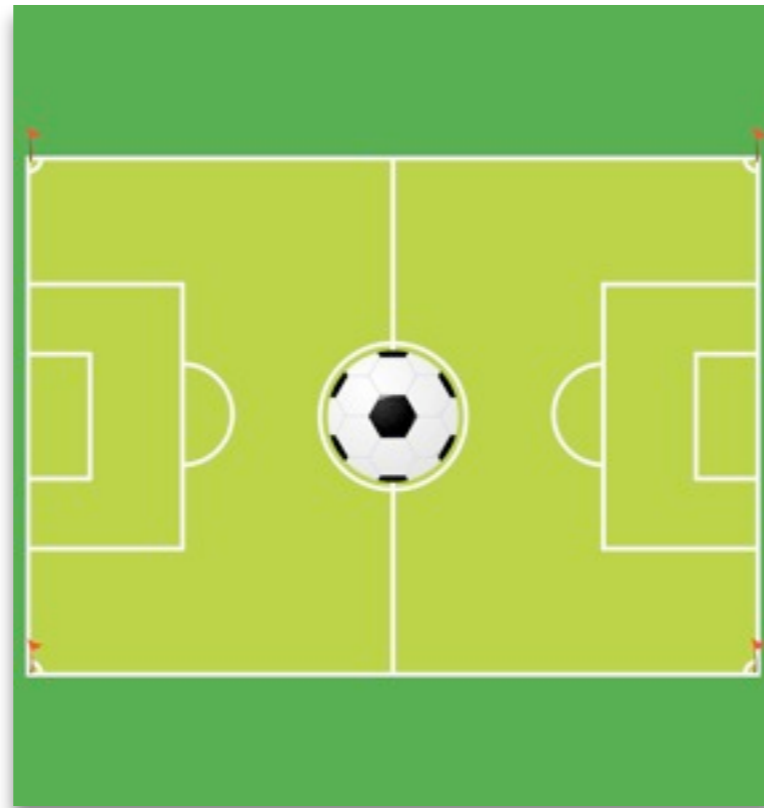
Start with Plan
and all
requirements

End with all requirements
completed



Scrum - Empirical

Scrum is not a methodology that will make you develop better products. Scrum does not provide the answers to how to build quality software faster.



Scrum is a tool you can use to find out what you need to do to build quality software faster.

Scrum

SCRUM GUIDE

This guide explains how to use Scrum to build products. In doing so, it will describe how the framework and its artifacts, time-boxes, roles and rules work together. Scrum does not include techniques and processes for building products; however, it will point out the efficacy and flaws of these techniques and processes.

Scrum is a framework for developing complex products and systems. It is grounded in empirical process control theory*. Scrum employs an iterative, incremental approach to optimize predictability and control risk. Within each iteration, Scrum employs self-organizing, cross-functional Teams to optimize flexibility and productivity.

The heart of Scrum is a ***Sprint***. A Sprint is one iteration of a month or less that is of consistent length throughout a development effort. All Sprints use the same Scrum framework, and all Sprints end with an increment of the end product that is potentially releasable. The increment is a complete slice, or piece, of the finished product or system that is developed by the end of an iteration, or Sprint. One Sprint starts immediately after the prior Sprint ends.

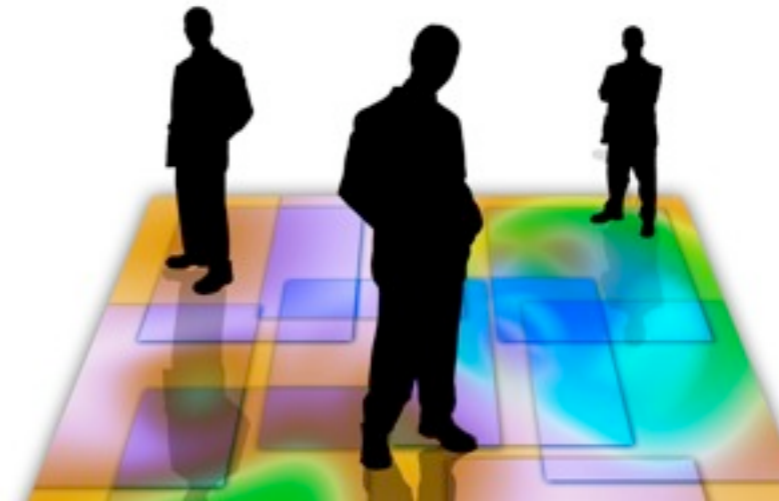
<http://www.scrumalliance.org/resources/>

Scrum Does Not Require Team Collocation

However, with Scrum,
you can measure the
productivity of
collocation



Self-Organizing
20 PBI/100k Euro



Not Self-Organizing
5 PBI/100k Euro

Scrum is a tool that you can use to:

- Increase productivity
- Increase predictability
- Increase risk management capabilities
- Increase the value of products and systems
- Increase quality
- Improving the morale and pleasure of the developers, product managers, customers and stakeholders

Scrum's use in spreading in product companies and engineering organizations.

Many of them suffer from one or more of the following:

- Release schedules slipping
- Stabilization at end of release taking longer and longer
- Releases are taking longer and longer
- Planning seems to take too long
- Changes are hard to introduce mid-release
- Quality is deteriorating
- Death marches are hurting morale

However,

When they use Scrum, they run into ScrumButs.

ScrumButs are reasons why they can't take full advantage of Scrum to solve the problems and realize the benefits.

A ScrumBut has a particular syntax:

(ScrumBut) (Reason)(Workaround)

An example ScrumBut is:

"We use Scrum, but (The Daily Scrum meetings are too much overhead) (so we only have them once a week, unless we need them more often)"

Dysfunction - perhaps only 25% of all developers are capable of creating a potentially shippable increment of functionality within a Sprint.

They don't have:

- the skills for acceptance test driven development, refactoring, test driven development.
- the followed practices of coding standards, coding reviews, and design reviews;
- the tooling for continuous builds with automated test harnesses for unit, regression, and performance testing.

The ScrumAlliance is starting a Certified Scrum Developer (CSD) program.

Teams that are able to select product backlog item(s) and turn them into an increment of potentially shippable product functionality will be certified.

CSD program will rollout this fall from various organizations based on several development infrastructures and engineering techniques.

The ScrumAlliance is starting a Certified Scrum Master - PMP program.

PMP's will learn how to apply the PMBOK processes and practices within a Scrum framework.

CSM - PMP program will begin rolling out this July. The ScrumAlliance will be partnering with PMI training organizations.

Prince2 is not yet planned.

The ScrumAlliance is going to tighten up and then expand the CST program.

CST's will receive further training and hone their training materials around the Scrum Guide.

CST's will work together more collaboratively.

CST's will have training domains.

This will begin in the fall of 2009 and be fully implemented by the start of 2010.



Questions?