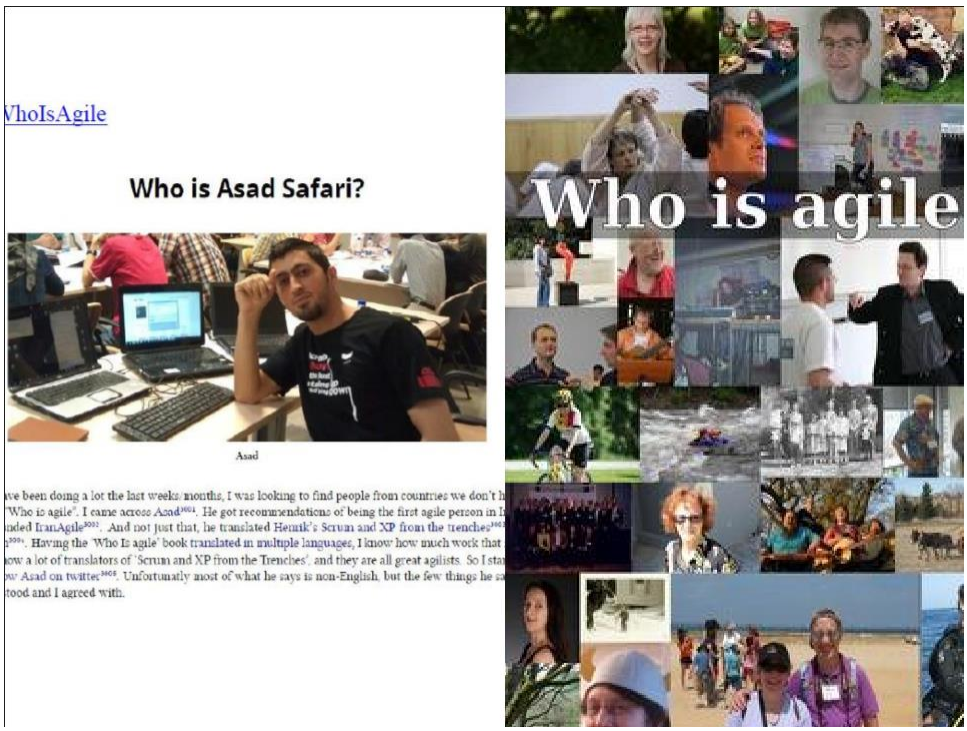
Practical Scrum

Asad Safari

CSP – CSM - CDA – PSM – PSPO – Management 3.0

- Founder of Iran Community and Iran Scrum Institute
- Member of Agile Alliance & Scrum Alliance
- +3 books in Agile mindset
- Coaching/Training Agile mindset over +7 years
- Agile Transformation in different industries: Banking, Ecommerce, Travel, Petrochemical, …
- Agile webinars and training courses for first time in Iran
- Agile world blog(more than 300 articles)



VhoIsAgile

Who is Asad Safari?

Asad

ve been doing a lot the last weeks/months, I was looking to find people from countries we don't h
"Who is agile". I came across Asad[900]. He got recommendations of being the first agile person in I
nded IranAgile[901]. And not just that, he translated Henrik's Scrum and XP from the trenches[902]
s[903]. Having the 'Who Is agile' book translated in multiple languages, I know how much work that
ow a lot of translators of 'Scrum and XP from the Trenches', and they are all great agilists. So I sta
ow Asad on twitter[904]. Unfortunatly most of what he says is non-English, but the few things he s
ttood and I agreed with.

Why are you in this class?

Goal of this course

Understand Scrum well enough to be able to apply it successfully & continuously improve.

Team Forming

10 MINS

Ensure you are in roughly even-sized teams of 5-6 members

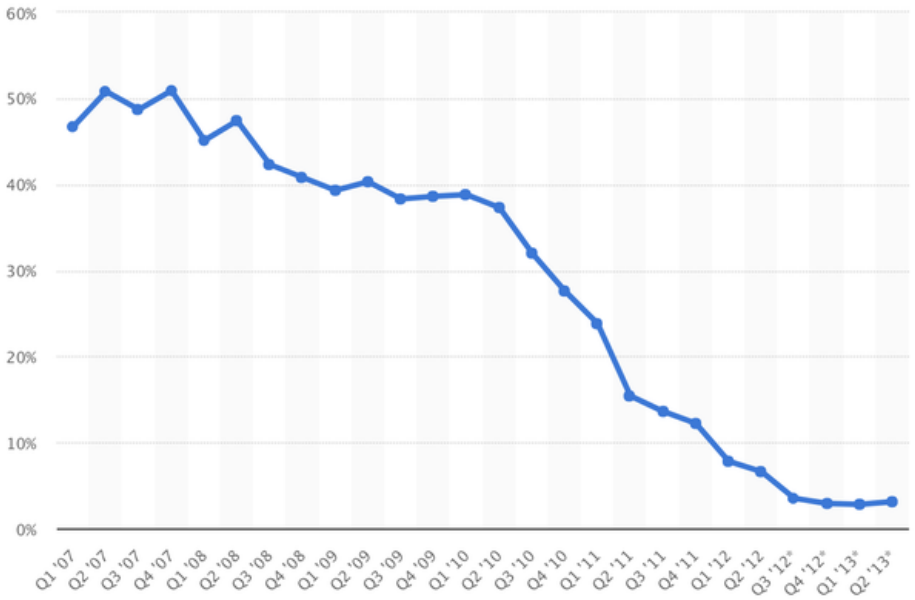
Post for all to see:

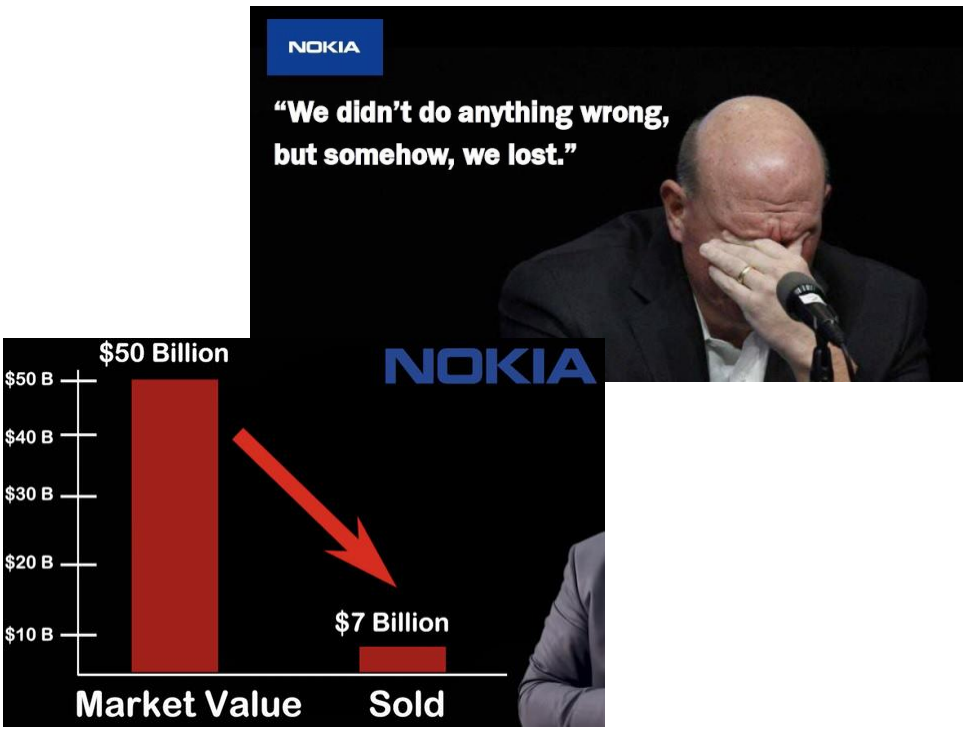
- Your team name
- A definition of a Scrum
- 3 things you want to learn in this class

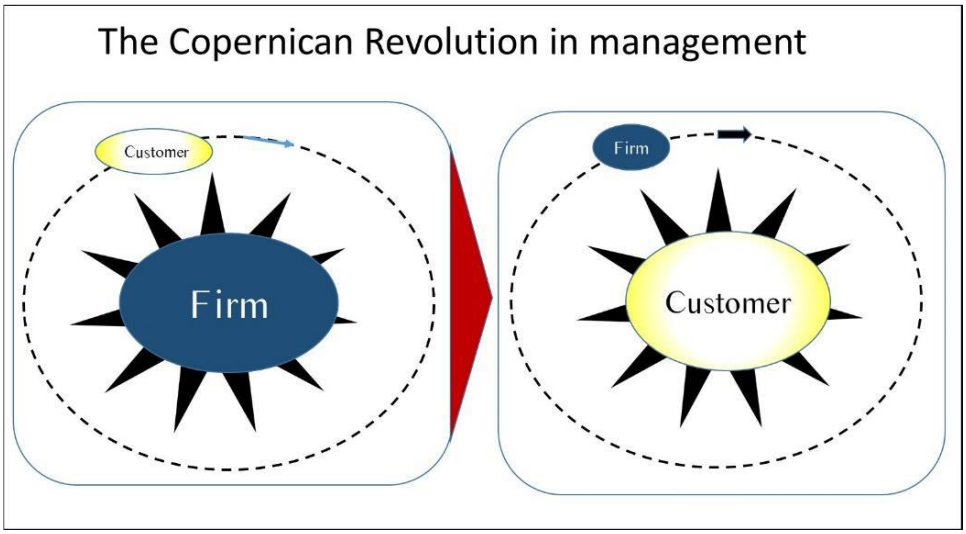
AHA!

- Write down your AHAs on sticky notes
- Put them somewhere on the wall

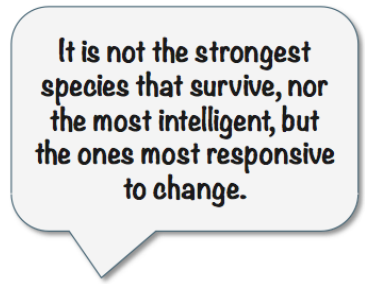
Why did Nokia fail?







Everything is changing so fast:
Technology
Requirements
Markets
Tools
Minds



Charles Darwin

Most IT projects don't succeed

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

IT project success rate 1994: 15% (success = supplier satisfied & customer satisfied)

Average cost & time overrun: ≈170%

Plan: €1,000,000

Actual: €2,700,000

IT project success rate 2004: 34%

Average cost & time overrun: ≈70%

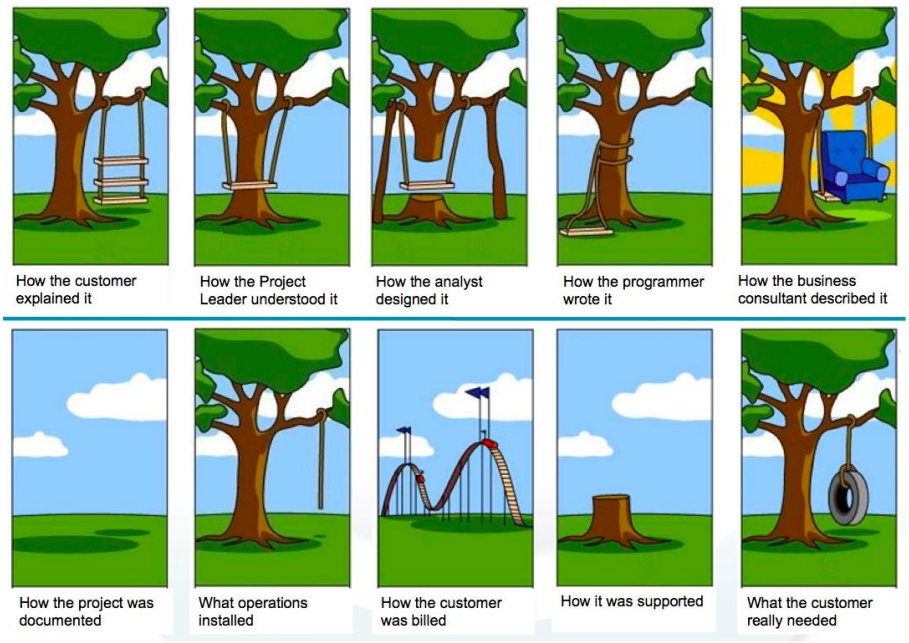
Plan: €1,000,000

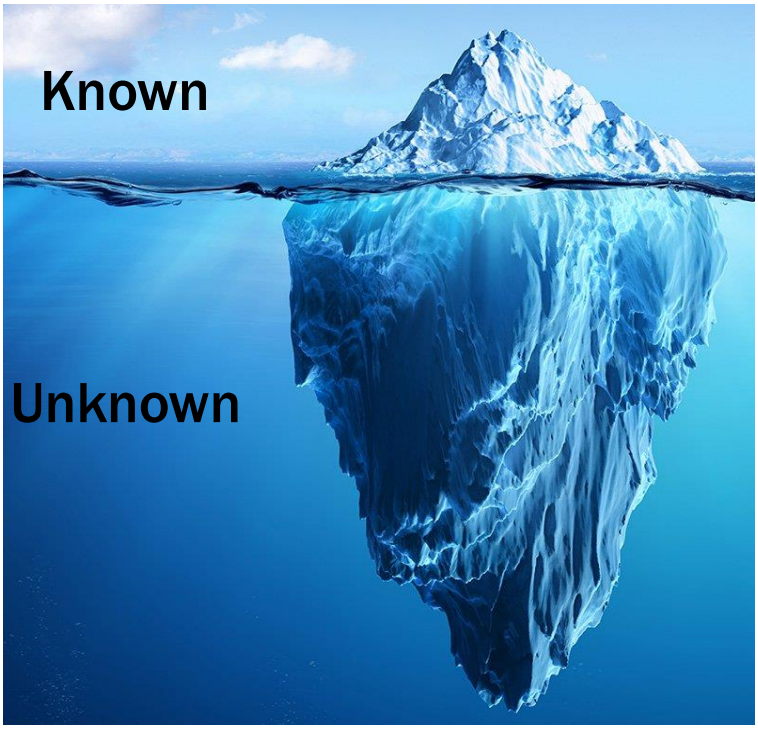
Actual: €1,700,000

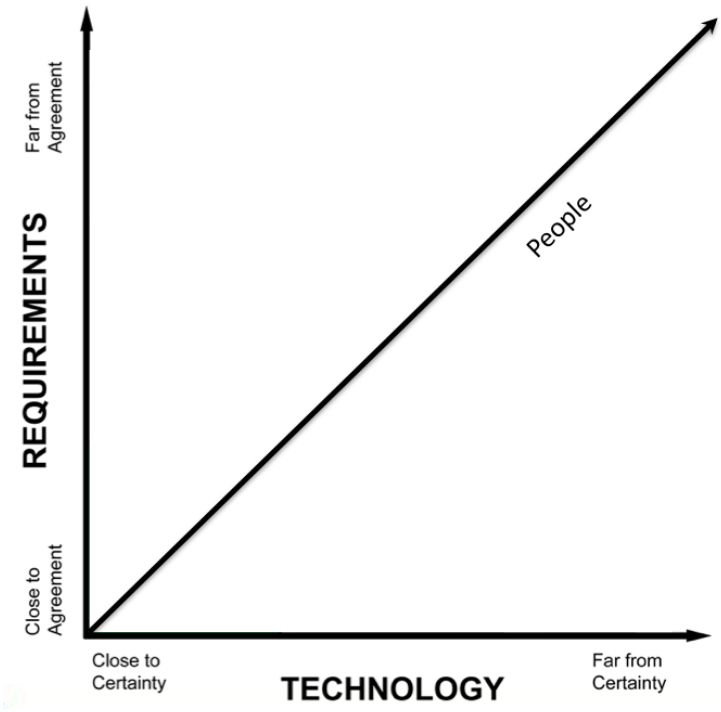
Sources:

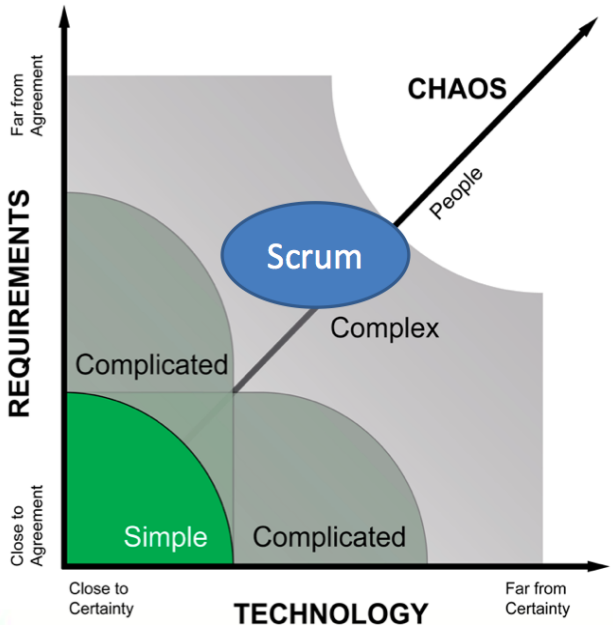
http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish
http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS

http://www.jacobsen.no/anders/blog/archives/images/project-thumb.jpg




Known
Unknown


REQUIREMENTS
Far from Agreement
Close to Agreement
People
Close to Certainty
TECHNOLOGY
Far from Certainty



Simple

Everything is known

Complicated

More is known than unknown

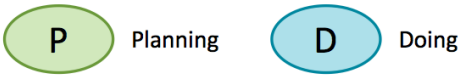
Complex

More is unknown than known

Chaotic

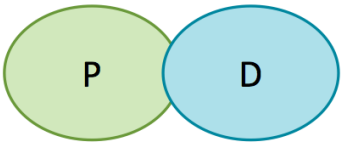
Very little is known

Empirical processes enable frequent planning



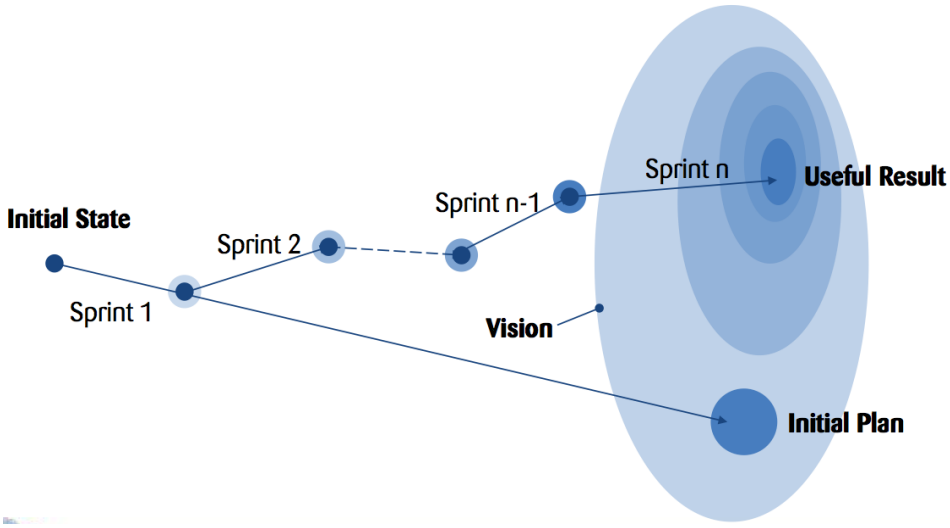
Defined (Predictive)

All planning is done at beginning



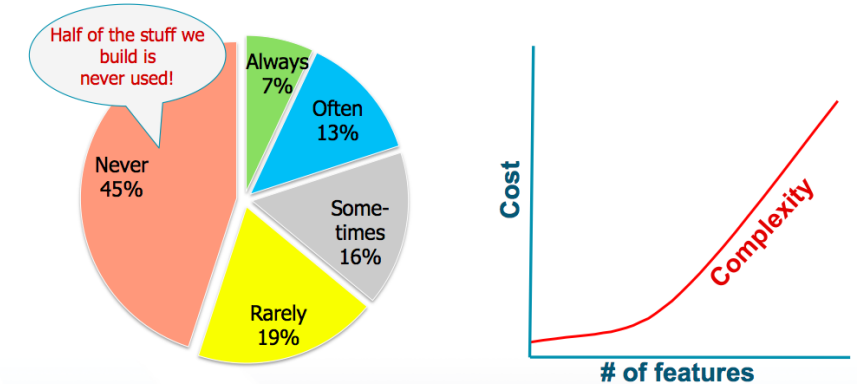
Empirical

Just-in-time planning and re-planning based on frequent inspection



We tend to build the wrong thing

Features and functions used in a typical system

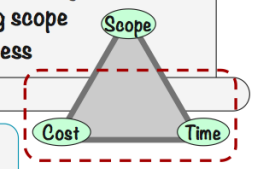


Source:
Standish group study reported at XP2002
by Jim Johnson, Chairman

What have we learned?

IT project success rate 1994: 15%
Average cost & time overrun: 170%
IT project success rate 2004: 34%
Average cost & time overrun: 70%

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



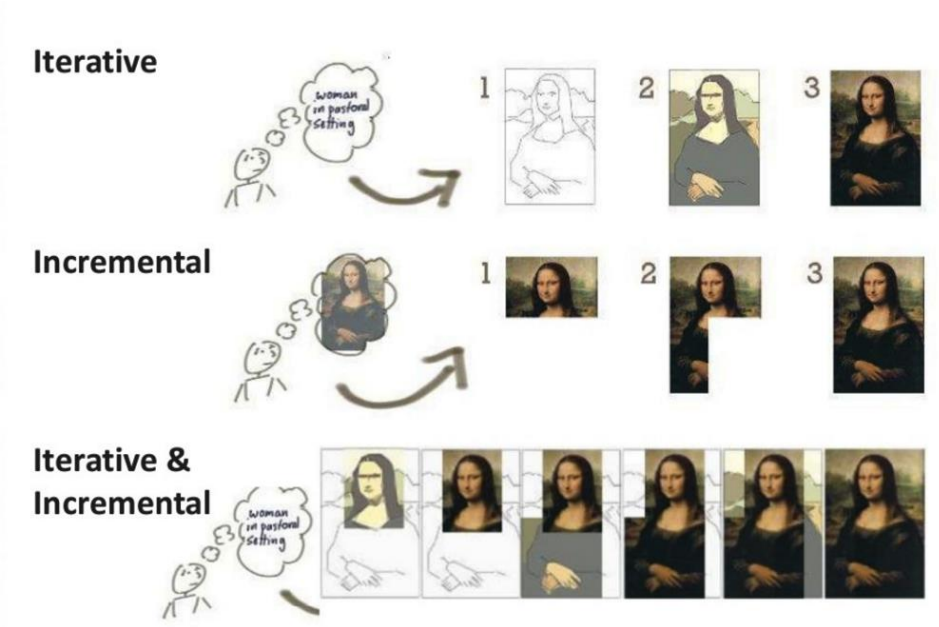
"The primary reason [for the improvement] is that projects have gotten a lot smaller."

Jim Johnson
Chairman of
Standish Group

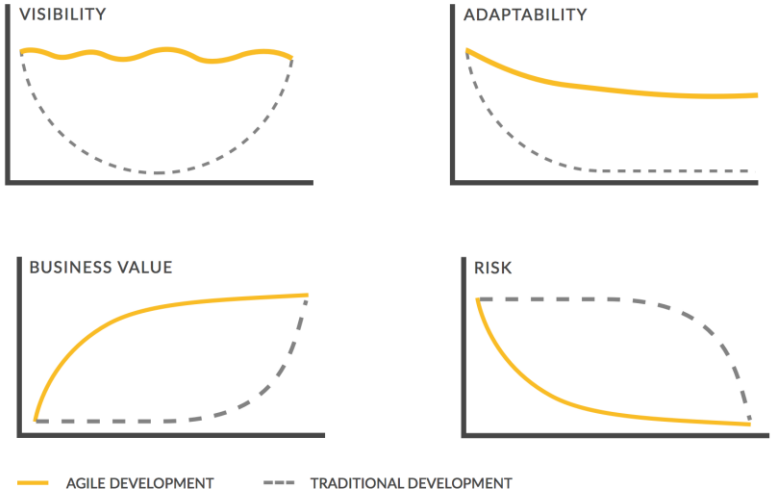
"Doing projects with iterative processes as opposed to the waterfall method, which called for all project requirements to be defined up front, is a major step forward."

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

Henrik Kniberg



AGILE DEVELOPMENT
VALUE PROPOSITION



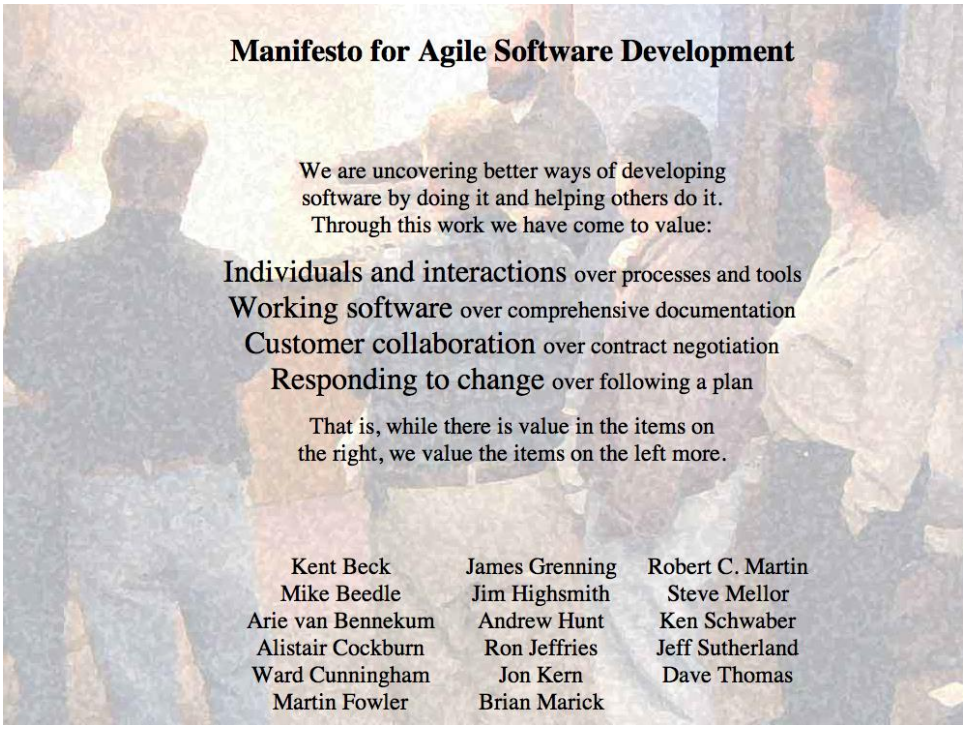
Agility (**a·gil·i·ty**)

–noun

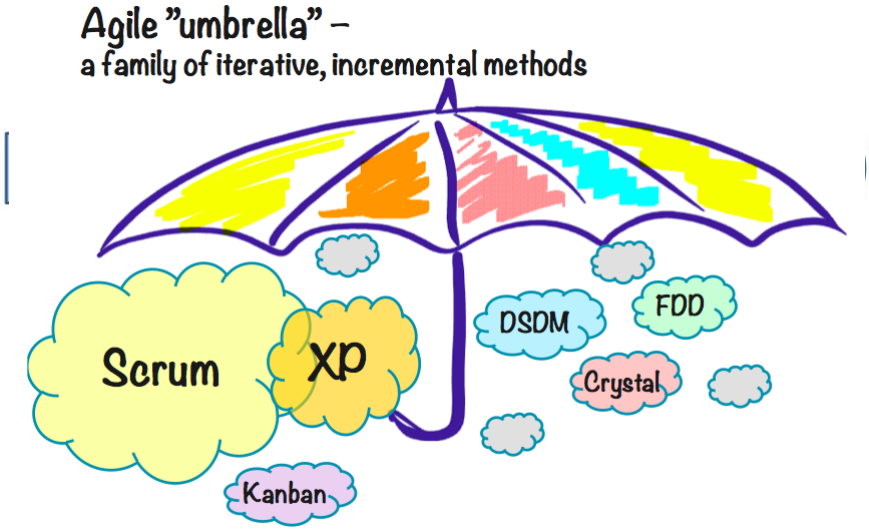
1. The ability to rapidly and deliberately respond to changing demand, while controlling risk.
2. flexibility, the capacity and capability of rapidly and efficiently adapting.

The courage to be honest enough to admit that building software is complex and it can't be perfectly planned since requirements change.

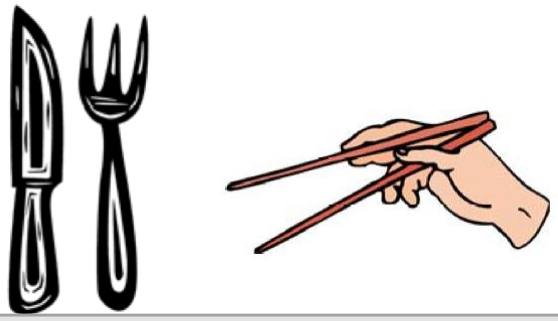
موسسه اسکرام ایران


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.
Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

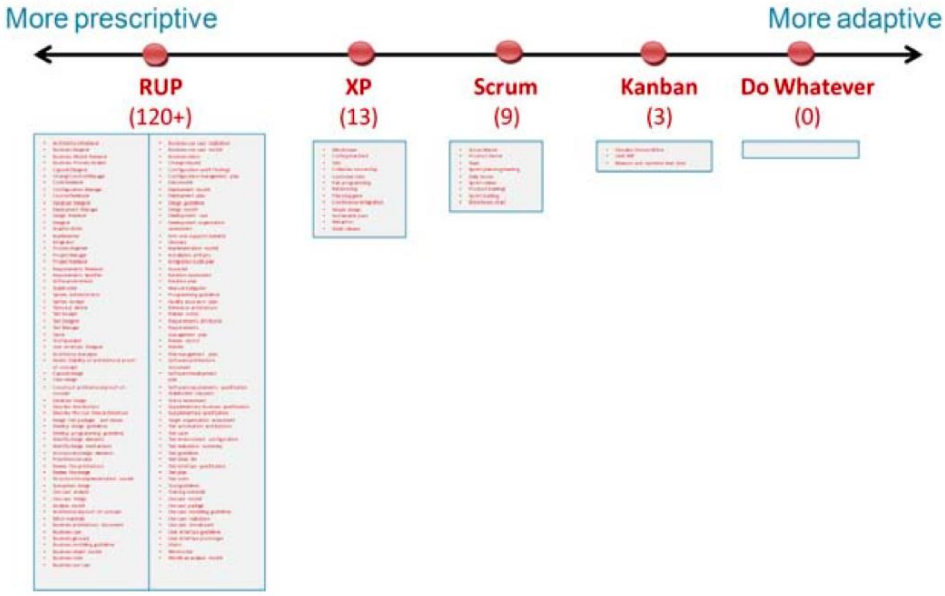

Satisfy the Customer
Welcome Change
Deliver Frequently
Collaborate Daily
Support & Trust Motivated Teams
Promote Face-to-Face Conversations
Deliver Working Software
Promote Sustainable Pace
Promote Technical Excellence
Maximize Through Simplicity
Have Self-Organized Teams
Reflect & Adjust Regularly
Source: www.agilemanifesto.org



Knife or fork – which tool is better?

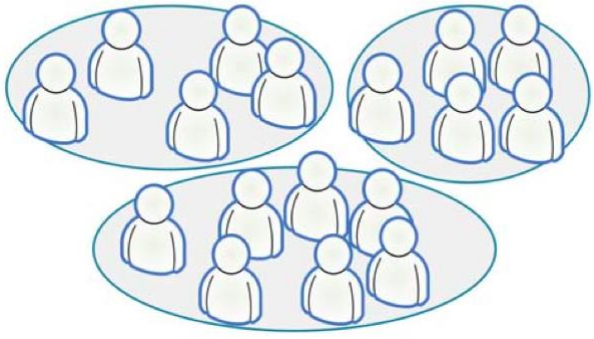


Compare for understanding,
not for judgment.

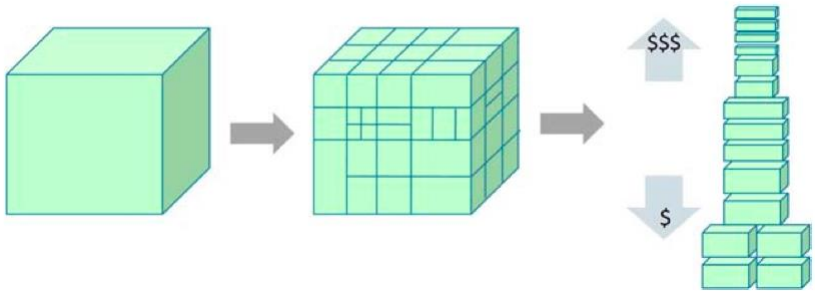


Scrum in a Nutshell

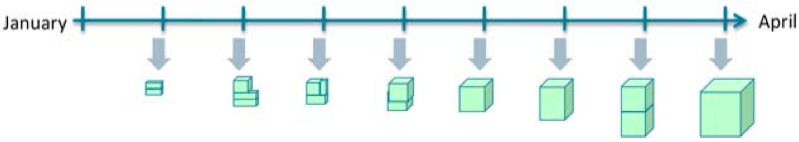
- **Split your organization** into small, cross-functional, self-organizing teams.



- **Split your work** into a list of small, concrete deliverables. Sort the list by priority and estimate the relative effort of each item.



- **Split time** into short fixed-length iterations (usually 1 – 4 weeks), with potentially shippable code demonstrated after each iteration.



- **Optimize the release plan** and update priorities in collaboration with the customer, based on insights gained by inspecting the release after each iteration.
- **Optimize the process** by having a retrospective after each iteration.

Scrum in a Nutshell

- So instead of a **large group** spending a **long time** building a **big thing**, we have a **small team** spending a **short time** building a **small thing**.

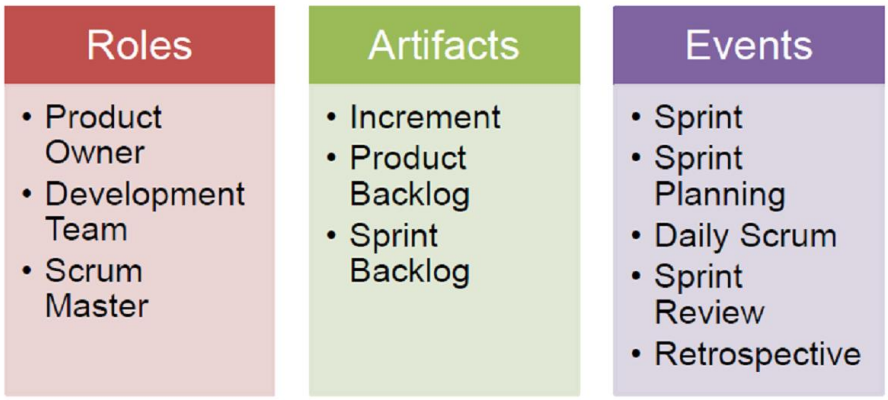
- But integrating regularly to see the whole.

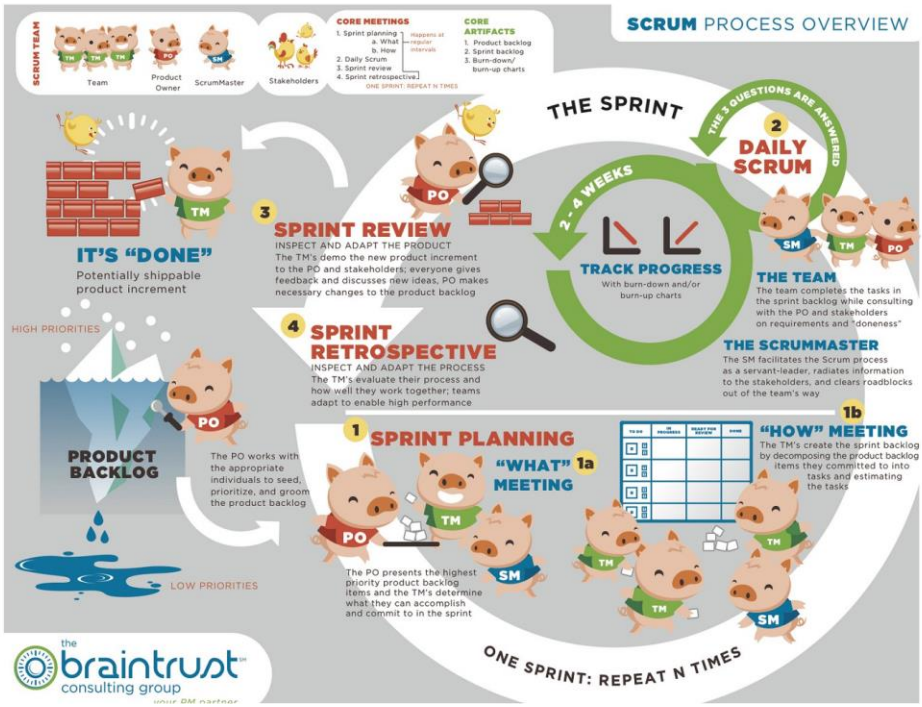
Scrum and Agile Basics

- **Scrum (n):** A framework within which people can address complex problems, and productively and creatively deliver products of the highest possible value.

- Lightweight
- Extremely simple to understand
- Extremely difficult to master

Scrum





Roles and Responsibilities

موسسه اسکرام ایران

Scrum Roles

موسسه اسکرام ایران

Product Owner

- Optimizes the value of the Product
- Creates and maintains the Product Backlog
- Chooses what and when to release
- Represents stakeholders and customers to the Development Team

موسسه اسکرام ایران

Product Owner

May Be	May Not Be
• A Product Manager • An executive • A personnel manager • A customer – When a customer understands Scrum	• A committee – They might chair one – They might represent one • The Scrum Master – This is a direct conflict of interest

The Development Team

- Creates the product Increment
- Operates in a series of Sprints
- Organizes itself and its work
- Collaborates with Product Owner to optimize value

Development Team Members

May Be	May Also Be
• Software Developer • Engineer • Tester • Architect • Graphic Designer • Technical Writer	• Business Analyst • Database Specialist • User Interaction Designer • Requirements Engineer

The Development Team

Ideally has every competency it needs to deliver a done increment

Scrum Master

- Enacts Scrum values, practices, and rules throughout the organization
- Ensures the Scrum Team is functional and Productive
- Provides guidance and support for the Scrum Team

Scrum Master

May Be	May Not Be
• A manager with appropriate servant-leader skills	• A Product Owner
• A member of the Development Team	*This is a direct conflict of interest*
• Selected by the Development Team	

Roles Review

Scrum Master

Product Owner

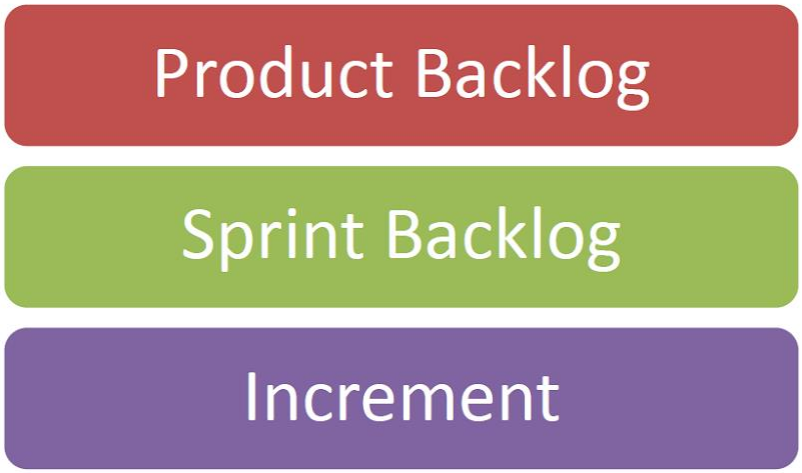
Development Team

موسسه اسکرام ایران

Scrum Artifacts

موسسه اسکرام ایران

Artifacts



موسسه اسکرام ایران

Product Backlog

- An ordered list of desirements
- Potential features of the product
- The single source of truth for what is planned in the product
- Public and available

موسسه اسکرام ایران

Product Backlog Item

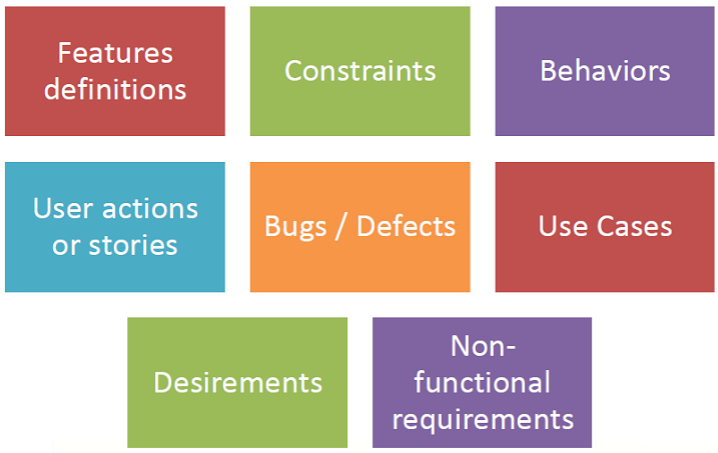
- The unit of deliverable work

- Contains clear acceptance criteria
  - Criteria for successful completion
  - Answering what will be true when this works

- May reference other artifacts like:
  - Specifications, Mockups, Architecture Models

- Sized appropriately
  - May be completed within a single Sprint
  - Typically with a few other PBIs

Valid Product Backlog Items



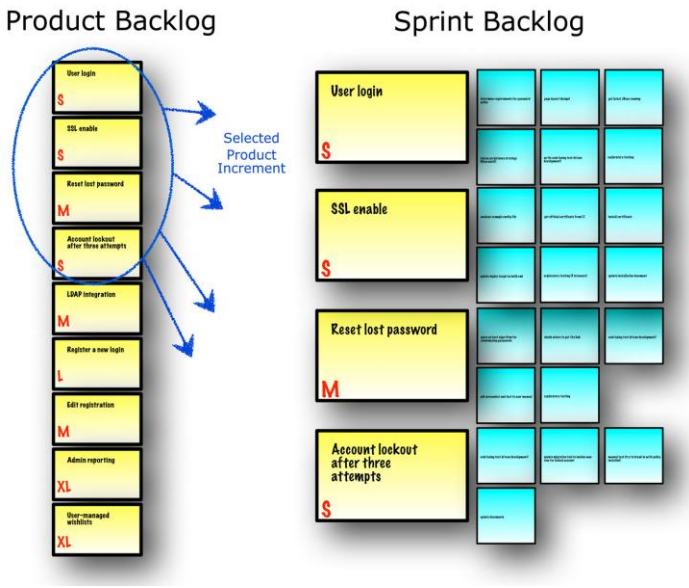
موسسه اسکرام ایران

Sprint Backlog

- Created by the Development Team during Sprint Planning

- Often derived by examining and decomposing Product Backlog items

- Might be a simple To Do list



Increment

- The software created in the Sprint
- Is *usable and it works*
- Is *potentially shippable*
- Must be DONE
  - As per Scrum Team standards
  - With no work remaining

Artifacts Review

Product Backlog

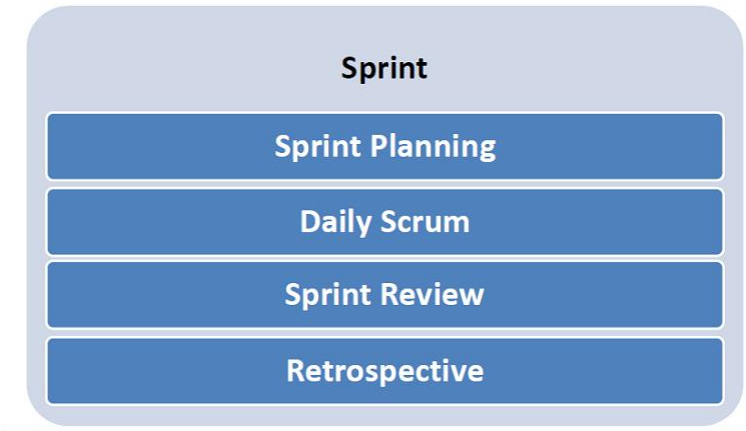
Sprint Backlog

Increment

موسسه اسکرام ایران

Events and Time Boxes

Scrum Events and Time Boxes



موسسه اسکرام ایران

Sprint Planning Meeting

- A Sprint Goal is created

- The Sprint Backlog is created
– The Development Team selects Product Backlog items for the Sprint
– The Development Team forecasts what will be completed this Sprint

- The entire Scrum Team attends

Sprint

- The time the Scrum Team works on creating the next product Increment

- Bounded by Sprint Planning and Sprint Retrospective

- One month or less

Daily Scrum

A daily meeting for the Development Team to:

- Synchronize activities
- Create a plan for the next 24 hours
- Assess progress toward the Sprint Goal

The Scrum Master may or may not facilitate

A Simple Scrum Board

موسسه اسکرام ایران

Sprint Review

The Scrum Team shows the product increment

- The Product Increment is inspected

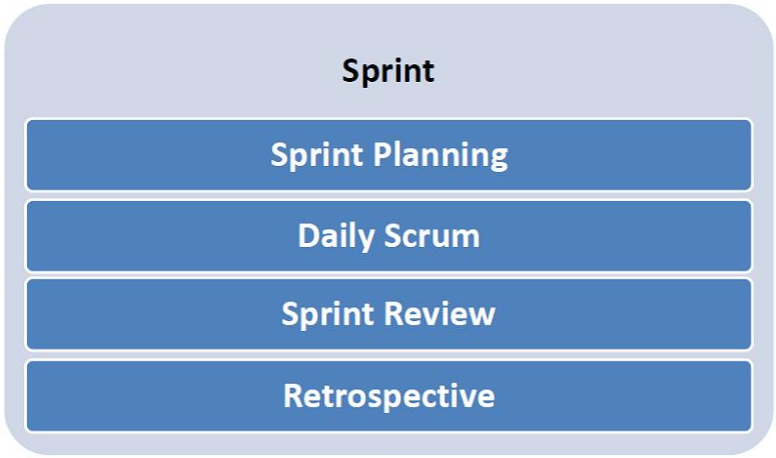
- Stakeholders are encouraged to provide feedback

Sprint Retrospective

- An inspect / adapt opportunity for the Scrum Team

- The Scrum Team discusses
  - What went well in the Sprint
  - What could be improved
  - What will we commit to for the next Sprint

Concept Review



موسسه اسکرام ایران

Section Summary

- Scrum Defined
- Roles and Responsibilities
- Artifacts of the Framework
- Events and Time Boxes

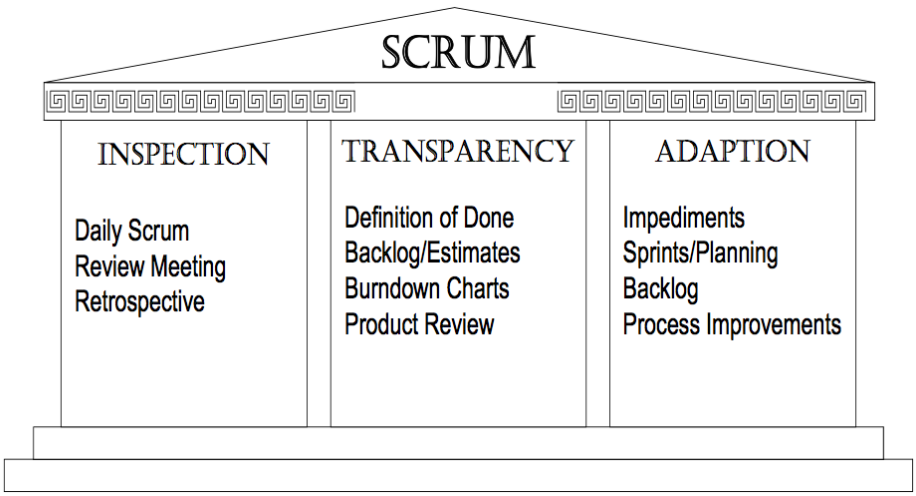
موسسه اسکرام ایران

Section References

- Scrum Guide from Scrum.org
- Agile Project Management with Scrum by Ken Schwaber
- Agile Software Development with Scrum by Ken Schwaber and Mike Beedle
- The Enterprise and Scrum by Ken Schwaber
- The Fifth Discipline: The art and practice of the learning organization by Peter Senge
- martinfowler.com/articles/itsNotJustStandingUp.html
- Agile Estimating and Planning by Mike Cohn
- Scrum and XP from the Trenches by Henrik Kniberg

موسسه اسکرام ایران

Theoretical Foundation



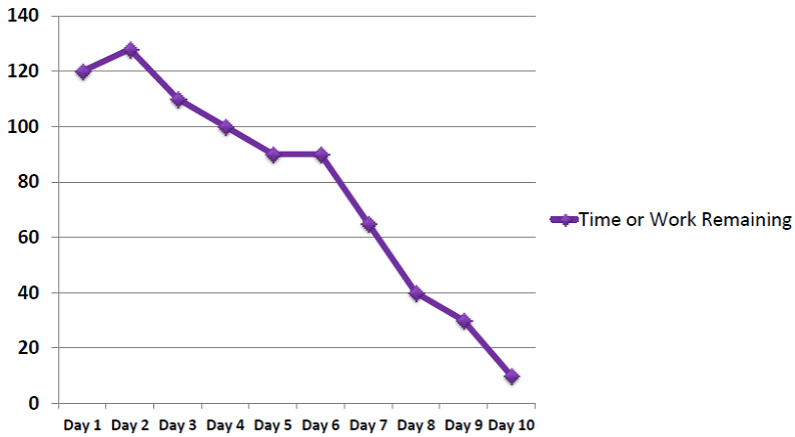
The Scrum Framework Part 2

In This Module

- Scrum Measurements
- More on Roles, Artifacts, and Events
- Self Organization

Scrum Measurements

Sprint Burndown – One Tool for Sprint Monitoring



Sprint Progress

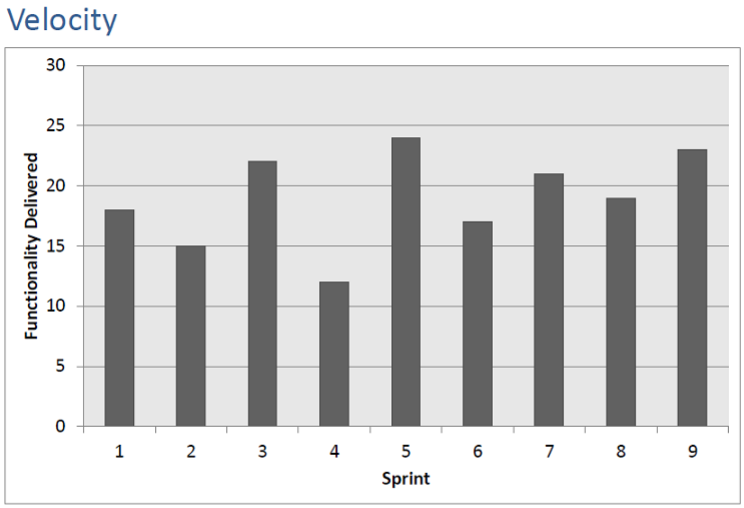
- This measurement is for
  - The Development Team
  - No one else

- Used to see
  - How we are progressing in the Sprint
  - If scope should be discussed with the Product Owner

Sprint Progress Monitoring

- May change abruptly when
  - New work is added or removed during the Sprint
  - Scope is renegotiated with the Product Owner
  - New things are learned about the work of the Sprint

- Can easily be used
  - To micromanage the Development Team
  - To demonstrate false progress



Velocity

- A measure of features or functionality delivered per Sprint
- Used by the Product Owner to provide forecasts
- Used by the Development Team to gauge how much work to pull in a Sprint Planning meeting

Understanding Velocity

Purpose: Consider the nature of velocity

A Development Team's Velocity may vary dramatically from one Sprint to the next.

Question: Why? Is this good or bad?

More on Roles, Artifacts, and Events

Product Owner

- Has the final word on what the Development Team is working on

- Not the Development Team's assistant
  - May not ever write a User Story
  - Typically spends 20-30% of their time with the Development Team

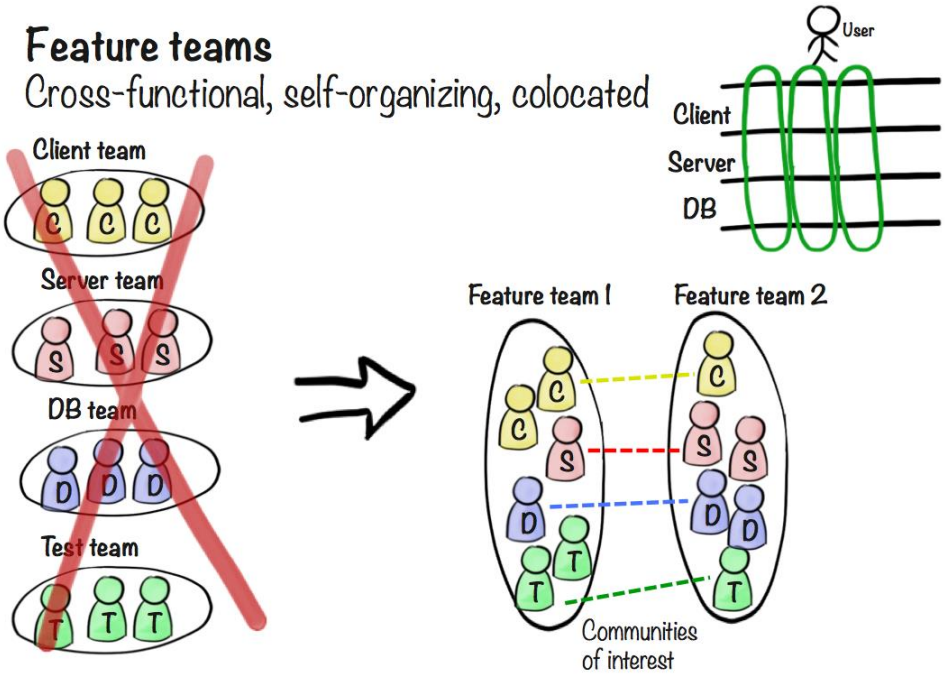
- Defines features and functionality
  - The level of this will vary
  - Some Product Owners will work closer to implementation details than others

The Development Team

- Composition is constant throughout a Sprint

- Typically has 6 +/- 3 members

- May have partially allocated members
  - Often considered an impediment
  - Ex: Database Administrators, User Interface Design experts, Technical Writers



Scrum Master

- Helps the Scrum Team
  - Adhere to Scrum values, practices, and rules
  - Understand and use self-organization
  - Be more productive
  - Increase quality of Increments

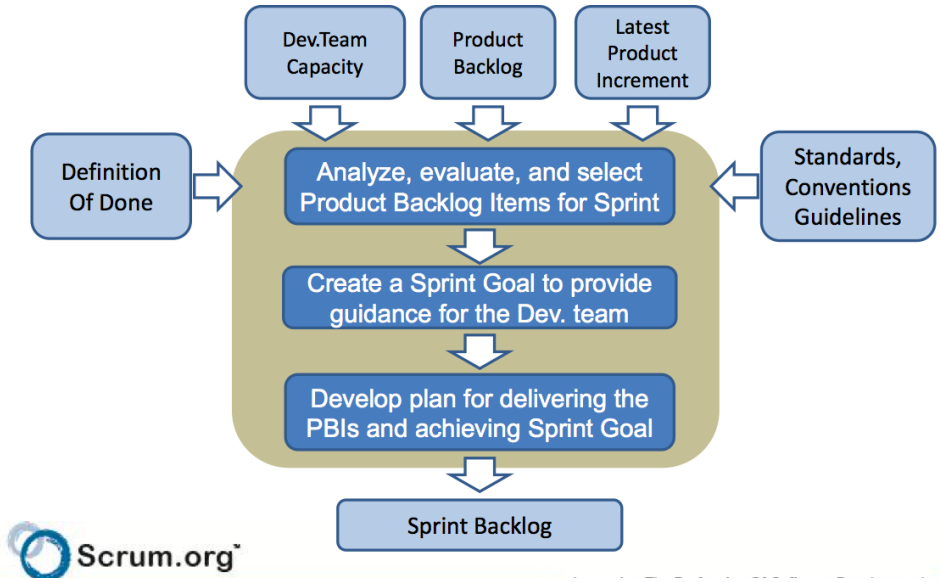
- Removes Impediments to the Development Team's success

Sprint Planning Meeting

- Typical duration is 5% of total Sprint length
  - 8 hours for 4 week Sprint
  - 4 hours for 2 week Sprint

- Has two phases
  - What will we do?
  - How will we do it?
  - Some Development Teams do this all at once

Flow of Sprint Planning Meeting



Sprint Planning Meeting Attendees

What might each be responsible for in this meeting?

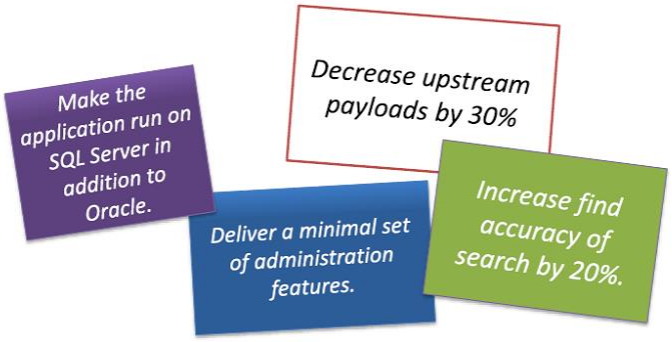
- Product Owner
- Development Team
- Scrum Master

Sprint Goals

An objective to be met in the Sprint	• Through the implementation of the PBIs selected in Sprint Planning • Providing guidance to the Development Team
Allows flexibility in delivering the increment	• Allows wiggle room for exact implementation of PBIs • Although the Sprint Goal is fixed
Are sacrosanct throughout the Sprint	• As the Development Team works, it keeps this goal in mind • Each Daily Scrum assesses the Team's progress toward meeting the Sprint Goal

موسسه اسکرام ایران

Some Sprint Goals



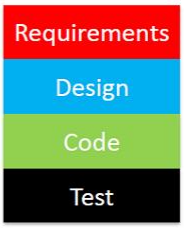
The Sprint

- All Sprints have consistent duration
- Starts right after the previous one
- Scope is negotiated constantly throughout
  – Between Development Team and Product Owner
  – This recognizes uncertainty even within the Sprint

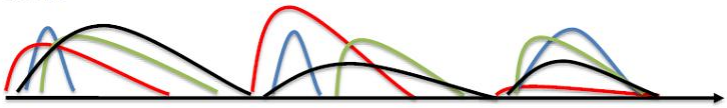
موسسه اسکرام ایران

The Development Team

- Non Sequential execution is key
- Everyone pitches in regardless of individual skill specialty



- The Development Team is held to account as a unit



The Daily Scrum

- 15 minute Time Box
- Create a plan for the next 24 hours
- By the Development Team, for the Development Team
  - Scrum Master ensures it occurs
  - The Development Team's opportunity to inspect/adapt daily
  - NOT a status report for management

Why a Daily Scrum?

- Share commitments
- Identify Impediments
- Create focus
- Increases situational awareness

Sprint Review

- 2.5% of Sprint Duration
  - 4 hours for a 4 week Sprint
  - 2 hours for a 2 week Sprint

The entire Scrum Team is present

+

All who care to attend

The Sprint Retrospective

- The Scrum Team's opportunity to improve
- Typically 3 hours for a 4 week Sprint
  - Proportionally less for shorter Sprints
- After every Sprint Review
- Whole Scrum Team participates
  - Scrum Master
  - Product Owner
  - Development Team

Kaizen - Continuous Improvement

موسسه اسکرام ایران

The Typical Sprint Retrospective Model

What worked well?

What could be improved?

What will we commit to doing in the next Sprint?

Scrum Team members make actionable commitments

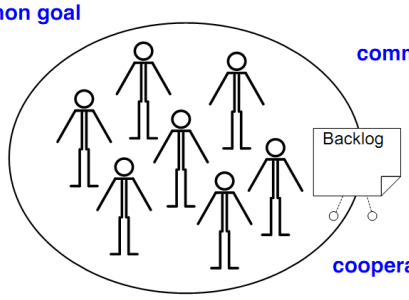
موسسه اسکرام ایران

Self Organization

Characteristics of (Scrum) Teams

have a challenging, common goal
make their own decisions
are autonomous

6 +- 3 persons
are cross functional
are multi-skilled workers



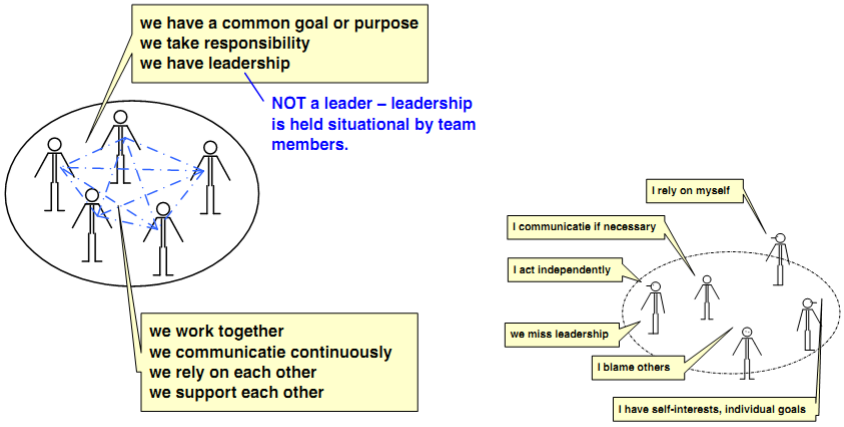
communicative

pull vs. push

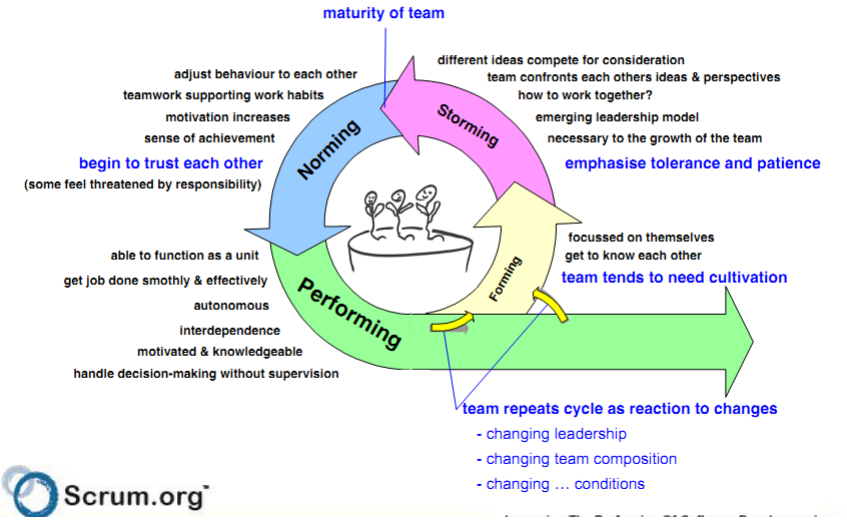
cooperation vs. competition

have the authority to manage their process and progress

Team vs. Group



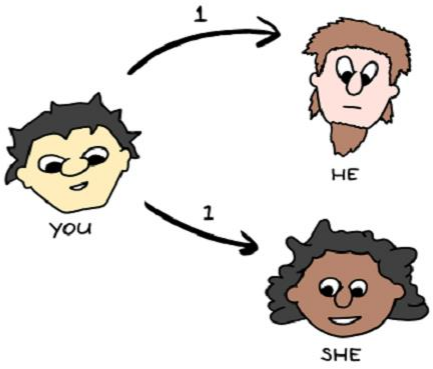
Forming – Storming – Norming – Performing



Managers are like gardeners

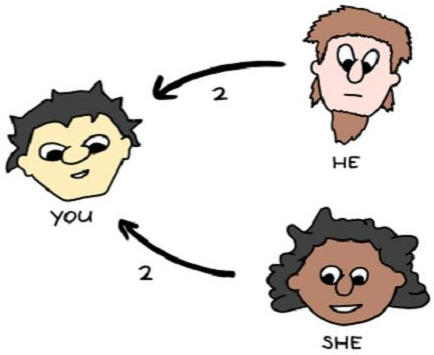
They let self-organization (anarchy) do useful work while steering the system toward valuable results

7



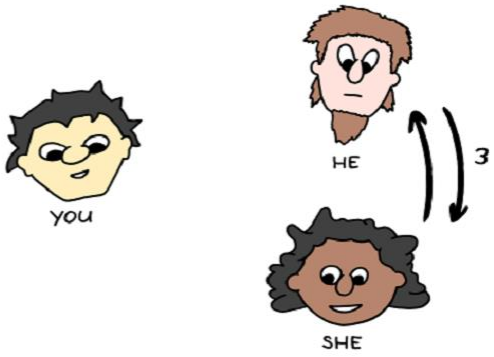
1) Trust your people (communicate this clearly)

20



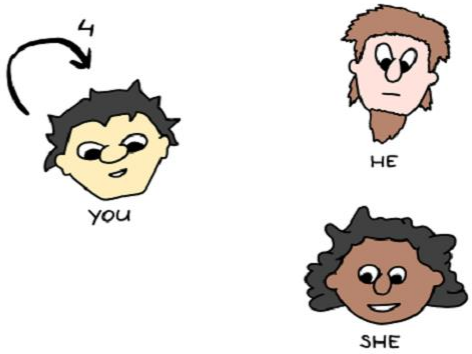
2) Earn trust from your people (consistent behavior)

21



3) Help people to trust each other (mingle, don't meddle)

22



4) Trust yourself (stay true to your own values)

23

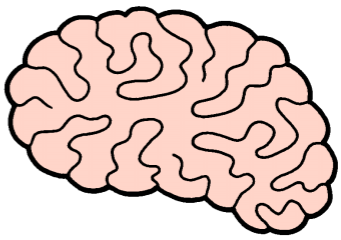
Distributed being

A complex system is more than the sum of its parts, and the "extra" stuff is distributed over the system. It cannot be attributed to any single authoritative part.

Control from the bottom up

In a complex system, everything happens at once, and problems ignore any central authority. Therefore overall **governance must be spread among all the parts**.

Kelly, Kevin. *Out of Control*.
Boston: Addison-Wesley, 1994, page 469



10

Key Decision Areas

Make explicit list with "areas of authorization"

Prepare project schedules
Select key technologies
Set documentation standards
Etc…

People should not walk into "invisible electric fences"

Reinertsen, Donald. *Managing the Design Factory*. New York: Free Press, 1997, page 107.

27

Key Decision Areas

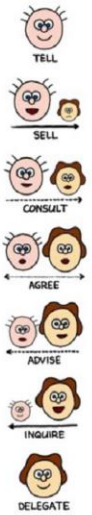
However…

Authorization per key decision area is not a "binary" thing

Reinertsen, Donald. *Managing the Design Factory*. New York: Free Press, 1997, page 107.

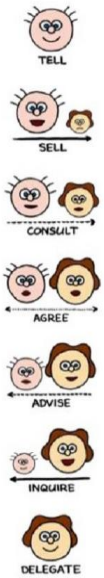
28

The Seven Levels of Authority



1. Tell: make decision as the manager
2. Sell: convince people about decision
3. Consult: get input from team before decision
4. Agree: make decision together with team
5. Advise: influence decision made by the team
6. Inquire: ask feedback after decision by team
7. Delegate: no influence, let team work it out

موسسه اسکرام ایران



1. Relocate to other office building
2. Replace waterfall with Scrum
3. Select new team members
4. Choose logo for business unit
5. Select architecture or component
6. Sprint length and deliveries
7. Coding guidelines and pairing

EXAMPLE

موسسه اسکرام ایران

1. **Create Design**: A new team consists of five developers, none of them with experience in Agile software development. They are going to build a small e-commerce web application. The team has learned about emergent design, and they tell you (their manager) that they want to make their own decisions about software design issues. You have a little experience with design yourself, but not much. What authority level will you give them?

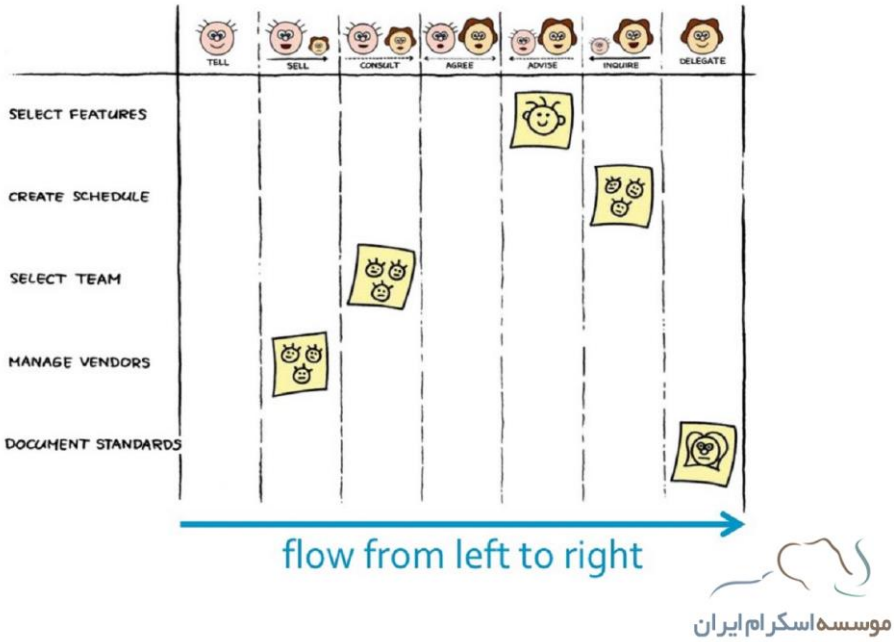
2. **Create Architecture**: An experienced software team is asked to build a component that monitors and regulates the flow of car traffic with smart traffic lights. There is nothing they can reuse, so they have to start from scratch. What authority level for creating the architecture will you give them now?

3. **Organize Entertainment**: A new team consists of a mix of experienced and inexperienced employees. They plan to organize a game night for all employees at the office on a Friday night. There is a fixed budget available. How will you authorize the team to organize this event?

4. **Form Teams**: A number of experienced agile employees claim that the traditional way of forming teams by management doesn't work well. They believe they are capable of forming their own teams around the projects that are handed to them. What will you do?

5. **Hire Employees**: You wish to involve existing team members in the recruitment and hiring of new employees. What authority level will you give them for decisions concerning various job candidates?

6. **Design Process**: One of your teams has delivered an application with major bugs in it, which cost their customer a lot of money. The team claims that the custom in-house process was to blame, and for their next project they want full authority over their own process, so that they can implement a bigger and stricter process. What will you do?



The optimal level of authority depends on people's
competence and the organizational impact of decisions

Command and Control

5 MINS

Beware the “productivity” illusion!

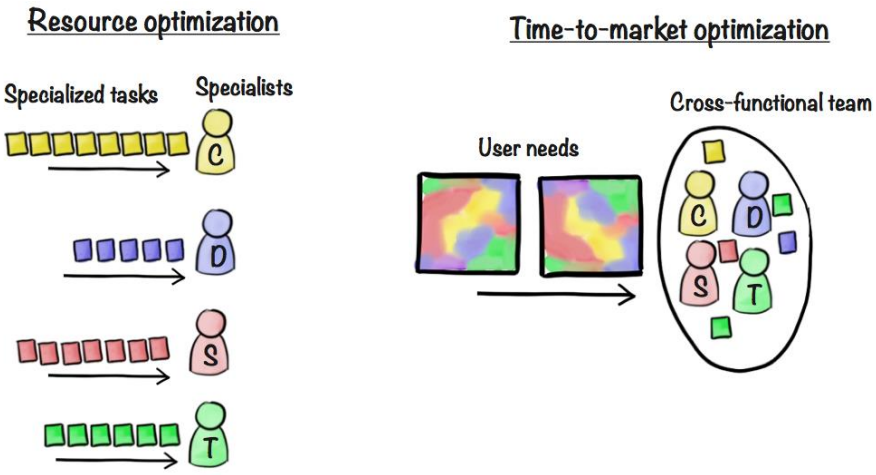
- Nobody actually wants software.
- Your job is to produce as LITTLE software as possible
- to solve the business problem.

100% resource utilization = 0% flow

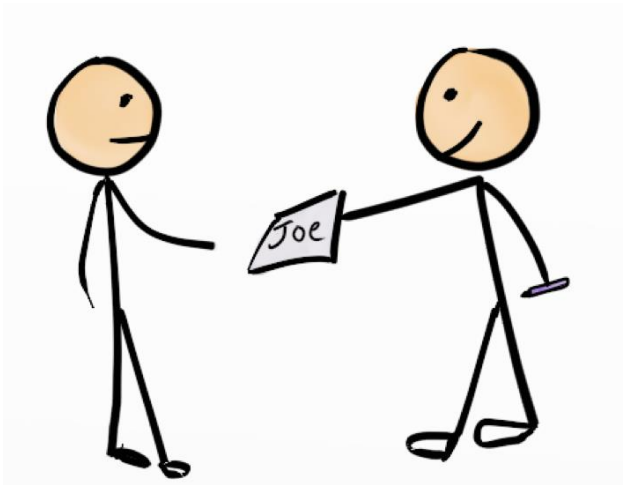
Max utilization, Slow flow

Low utilization, Fast flow

Resource optimization vs Time-to-market optimization



How long does it take to write a name?



10 MINS

Scrum Planning

Scrum Planning

- Plan constantly, not just in the beginning
- Planning is an activity, not a document
- Recognize, embrace, and support change rather than trying to control it

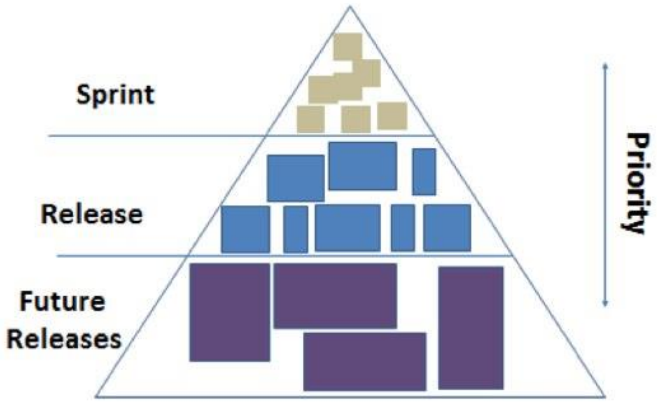
Scrum Planning

- Be constantly and consistently transparent

- Focus on historical performance,
not optimal scenarios of what might happen

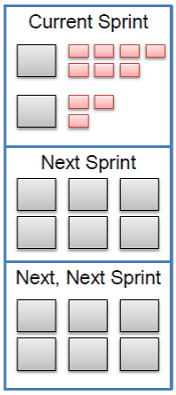
- Changing the plan doesn't mean changing timing

Backlog accuracy



Keep a Rolling Backlog Projection

- PBIs are estimated and ordered for approximately the next 3 Sprints at all times
- The current Sprint is detailed
  - Broken into Sprint Backlog Tasks
  - Very granular detail
- Next 2 Sprints are understood by the entire Scrum Team
  - Estimated
  - Valued
  - Ordered
  - Loosely planned

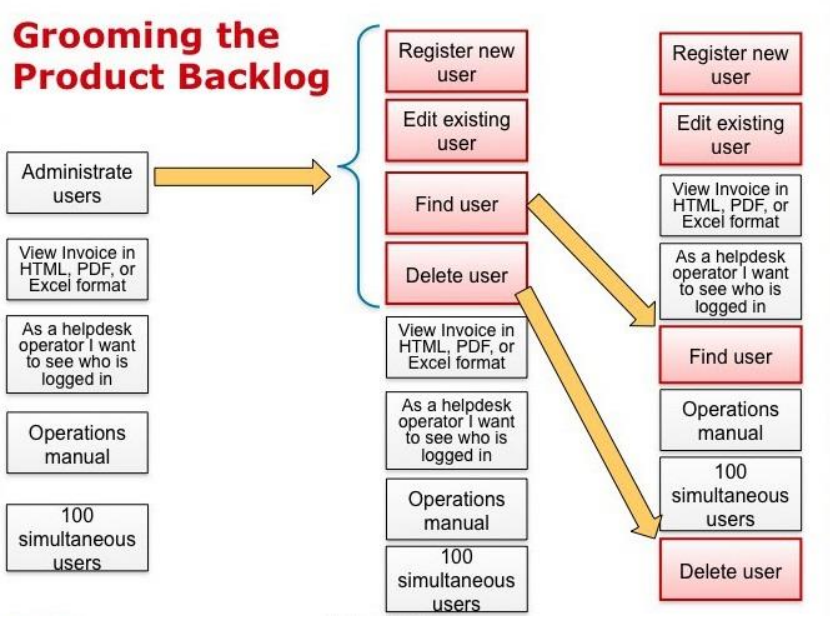


Backlog Grooming

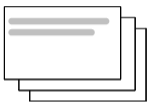
• Grooming means
– Planning the PBL to an actionable level of detail
– Maintaining a Rolling Backlog Projection

• Plan 10% of each Sprint to be spent grooming the Product Backlog

• Top ordered Product Backlog items are well understood

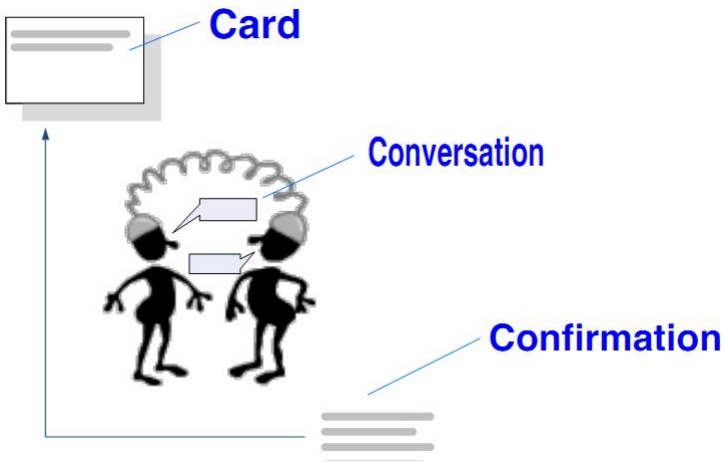


User Stories...



represent functionality
that is valuable to User or Customer

User Stories...



Product Backlog Item Meta Data

Complementary Practice

Title: . . .
As a . . .
I want . . .
So that . . .

Scenario: . . .
Given . . .
When . . .
Then . . .

Product Owner	*Bob*
Development Team	*Alpha*
Business Value	*13*
Effort Estimate	*5*
ROI	*2.6*

موسسه اسکرام ایران

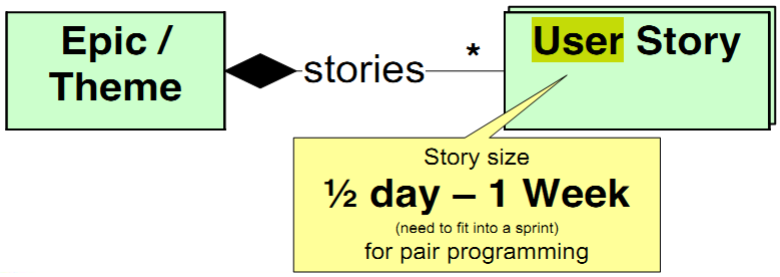
A Good User Story …

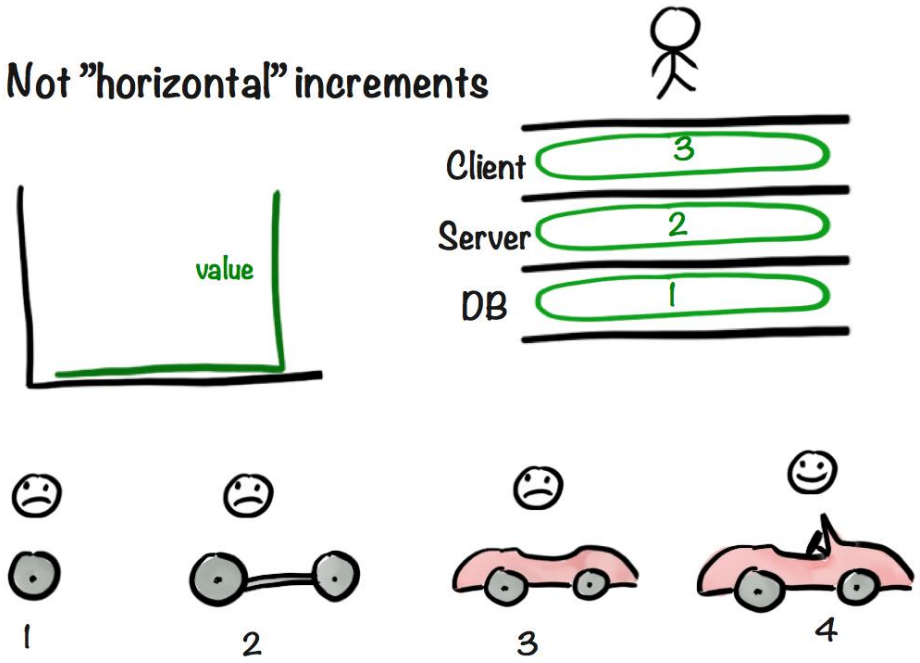
Independent
Negotiable
Valuable
Estimateble
Small
Testable

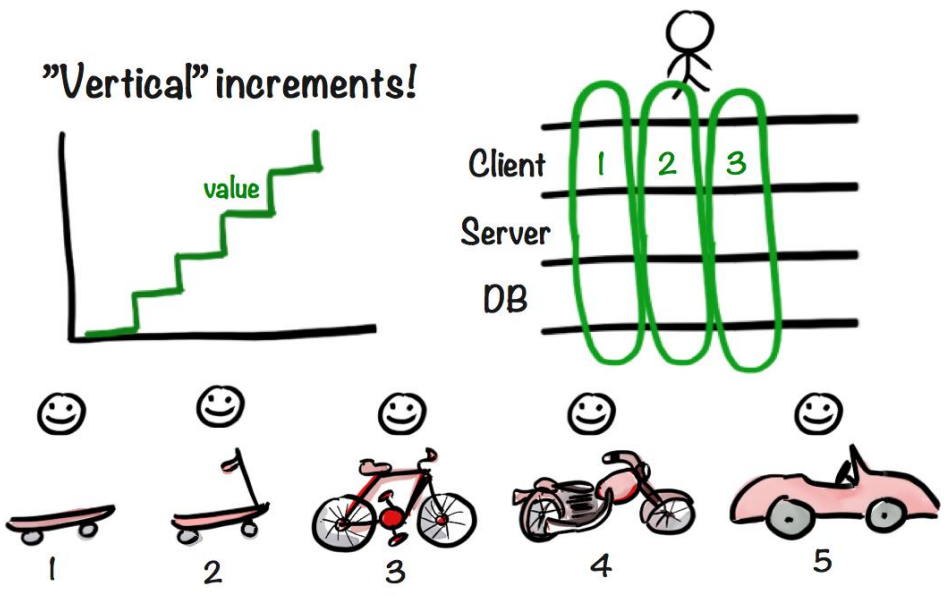
Size

Prefer many **short stories**

over big stories!




Not "horizontal" increments
value
Client
Server
DB
3
2
1
1
2
3
4


"Vertical" increments!
value
Client
Server
DB
1
2
3
1
2
3
4
5

Business Value

- The Product Owner is responsible for this
- It isn't always just revenue
- Can be estimated or calculated

Money to be earned

Clients to be retained

Delight to bring to users

Market share to capture

Money to be saved

Improvement to User Experience

Ordering the Product Backlog

- Risk
  - Identify risk for items in the Backlog
  - Do highest risk items first
- Return on Investment
  - Simple business value ranking system
  - This gives a single number by which to rank work
- Because the Product Owner says so

ROI Index Ordering

- Can see value at a glance
- Less subjective
- Not the last word
  - That's the Product Owner
  - ROI Index is a tool

موسسه اسکرام ایران

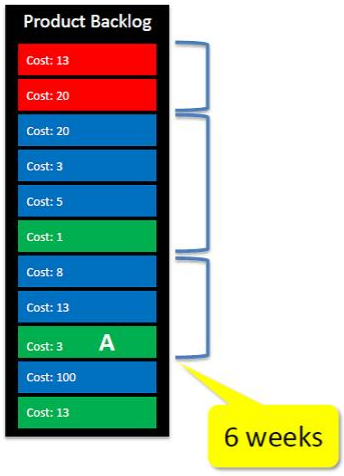
Rule #1

An accurate release plan requires an ordered and estimated backlog

موسسه اسکرام ایران

Rule #2

An accurate release plan requires known Velocity



When will constraint A likely ship?

- Average Team Velocity = 32
- Sprint Length = 2 weeks

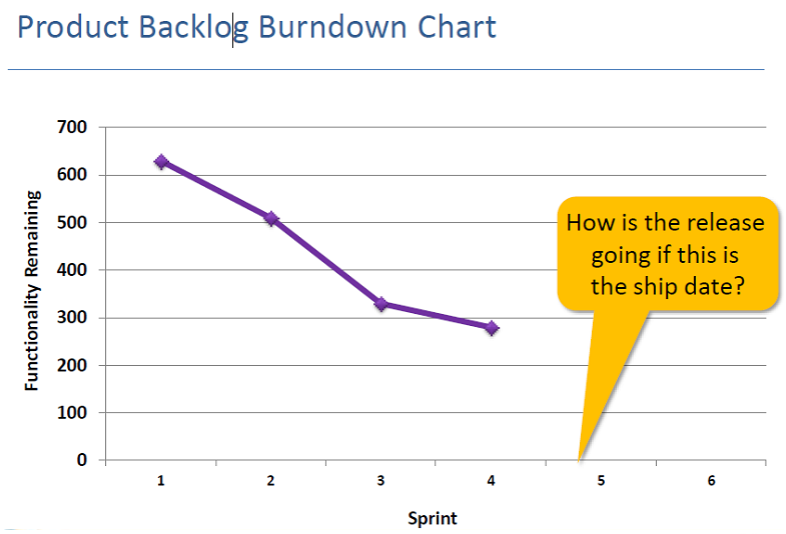
موسسه اسکرام ایران

What will be ready in 8 weeks?

- Average Team Velocity = 18
- Sprint Length = 2 weeks

2 Basic Types of Release Planning

Date Target Planning	Feature Target Planning
The product will release on a specific date	The product will release when specific features are ready
We Must Answer *How much of the backlog will be complete by a given date?*	We Must Answer *When will features A, B, and C be ready?*



موسسه اسکرام ایران

Product Backlog Burndown Chart

- Primarily used by the Product Owner

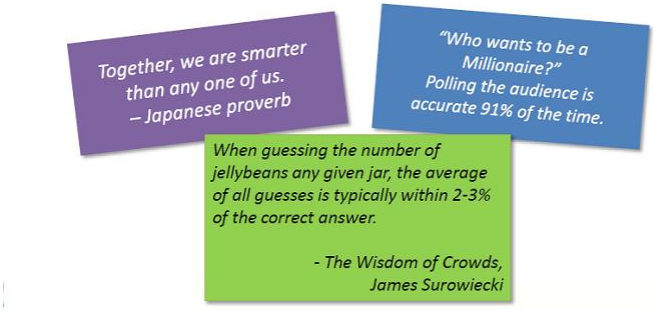
- Used to see
  - Progression toward a date
  - Progression toward a feature set
  - If scope should be discussed with the customer

موسسه اسکرام ایران

Estimating Software Development

Estimating with Groups

Group derived estimates are demonstrably more accurate than estimates by individuals

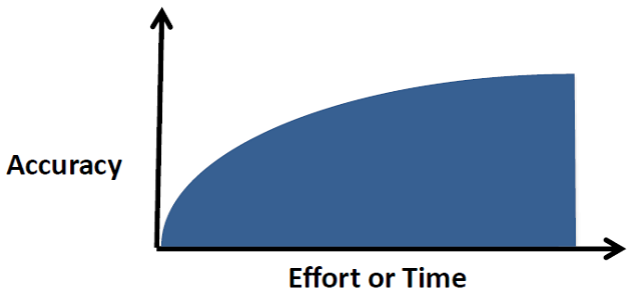


موسسه اسکرام ایران

Myth

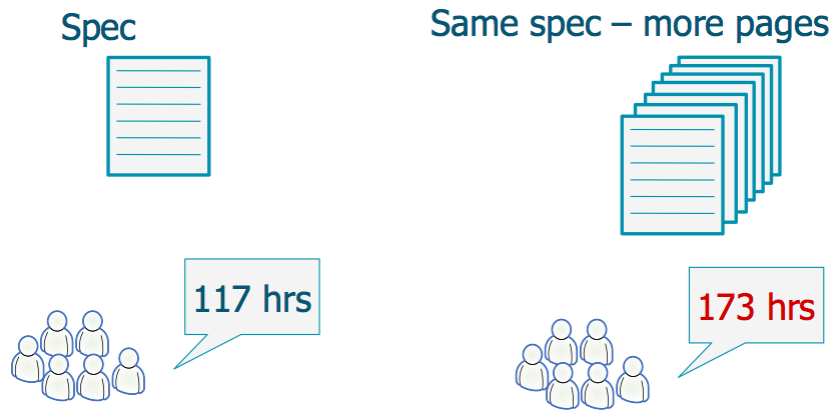
With more time, estimates get significantly more accurate.

Estimation is Often Expensive

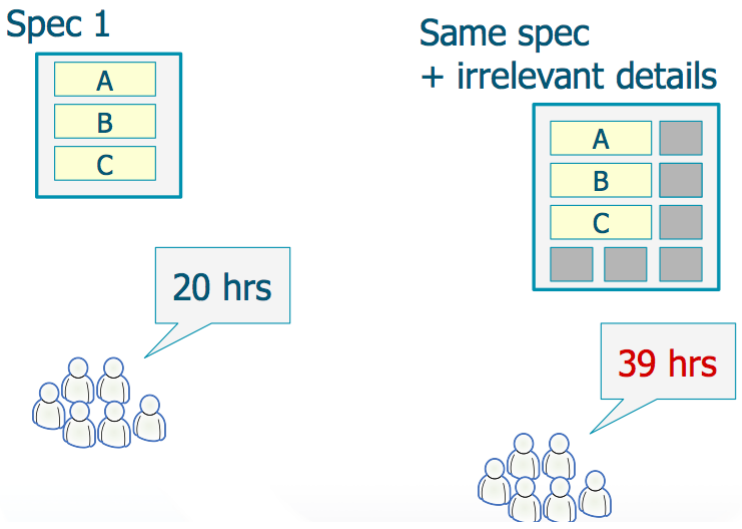


موسسه اسکرام ایران

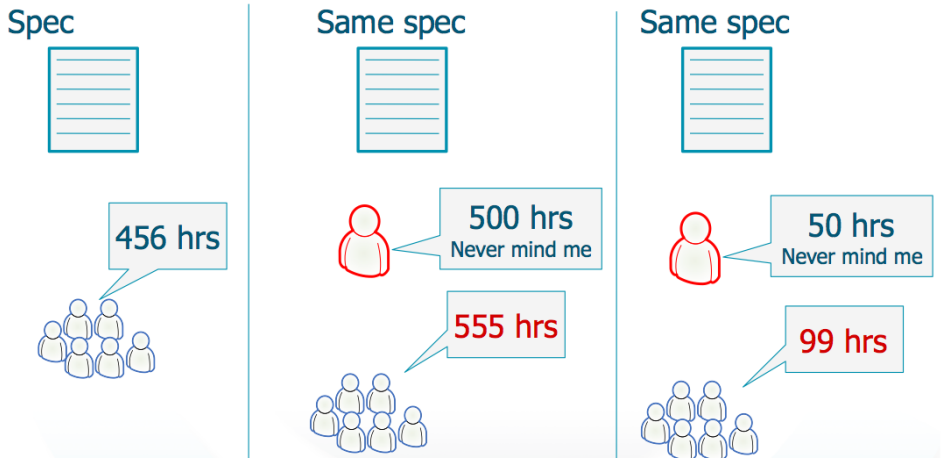
How estimates are affected by specification length



How estimates are affected by irrelevant information



How estimates are affected by anchoring



Story Points

- Very common way to estimate work
- Based on size and complexity, not duration
- Unitless and numerically relative
- Different for each team of estimators
- Points are additive
- Based on historical reality
- Easy to use and understand

Complementary Practice

موسسه اسکرام ایران

Story Point Values

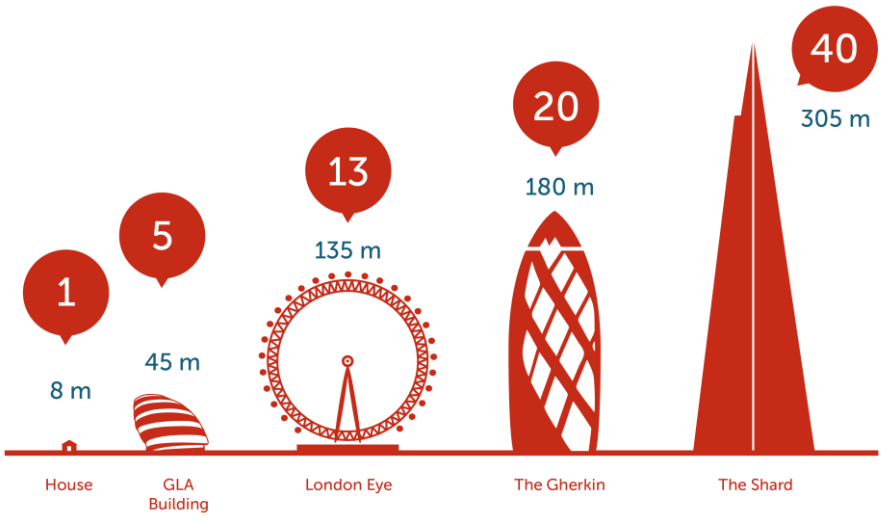
- Can you distinguish a 1-point story from a 2?
- Can you distinguish a 17 from an 18?
- How about a 99 from a 100?

- Use units that make sense
  - XS, S, M, L, XL, XXL
  - 1, 2, 3, 5, 8, 13, 21
  - 1, 2, 4, 8, 16, 32

Include big and small outliers if you want.
0, ½, 100, 300, ∞

موسسه اسکرام ایران

Relativity



Planning Poker Rules

1. Each estimator has a deck of estimation cards.
2. Customer/Product Owner reads a story and it's discussed briefly.
3. Each estimator selects a card that's his or her estimate.
4. Cards are turned over so all can see them (synchronously).
5. Discuss differences (especially outliers).
6. Re-estimate until estimates converge.

Avoid Anchoring

Please don't

- Broadcast opinions before estimations are made
- Show cards early

"This is an easy one"

"I am throwing a 3"

"This is huge"

"This will be a 5"

"I have no idea."

Because it

- Causes reactive estimates
- Shuts down discussion
- May leave important details undiscovered

Planning Poker

Planning Poker

Planning Poker

موسسه اسکرام ایران

Make Your Deck Now

5 MIN

Purpose: Each student create a Planning Poker deck

Create a deck of cards as follows:

1, 2, 3, 5, 8, 13, 21

? = I still have questions

∞ = Too big to estimate

* Note: The above sequence uses the actual Fibonacci sequence

موسسه اسکرام ایران

Planning Poker 1

Purpose: Practice using Planning Poker

1. Choose 3-5 PBIs of varying size your team has already delivered.
2. Choose 1 of medium size and label it as an 8.
3. Estimate the other completed PBIs

Use the estimated items as a comparison point to the items you are estimating.

Planning Poker 2

Purpose: Practice using Planning Poker

Using the already estimated and completed PBIs as reference points, estimate PBIs your team has not yet worked on.

Try and get 3-5 PBIs estimated.

Defining Done

When is Something Done?

- ❑When the Task is complete
- ❑When the Sprint is over
- ❑When the customer says so
- ☑When all agreed upon criteria for success are complete

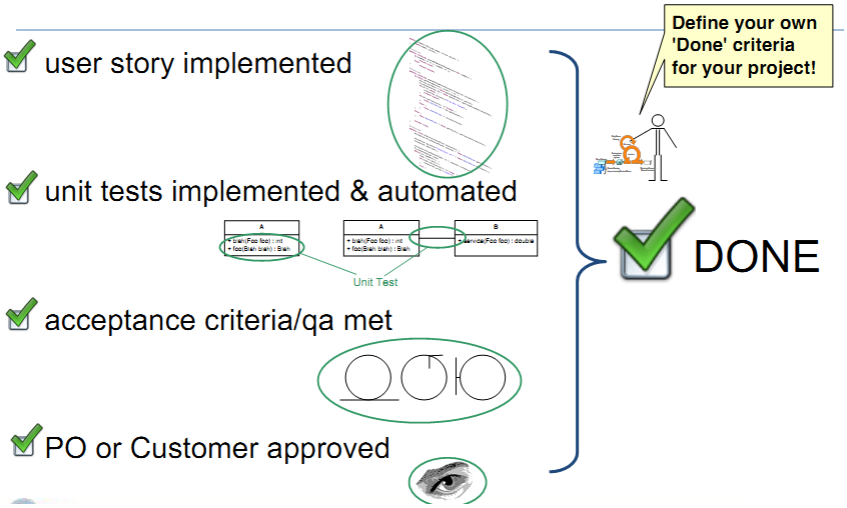
Definition of the Definition of Done

The Definition of Done (DoD) is a shared understanding of what it means for work to be complete.
This shared definition of completeness must be universally understood and agreed upon or transparency is lost.

DoD Tips

- Keep a checklist for DoD of the various levels
- Visit DoD in each Retrospective
- Evolve DoD for PBIs as they are executed
- DoD is defined by both the Product Owner and the Development Team

Definition of Done, an Example



Section Summary

- Planning Levels
- Estimating Software Development
- Owning a Product Backlog
- Defining Done

موسسه اسکرام ایران

Getting Started

In this Module

Getting Ready

Starting

Getting Ready

Agility Requires Organizational Change

Culture

"The way we do things here."

- Today's culture is finely tuned to produce current conditions
- Agility is an entirely new state
- Culture must change to achieve Agility
- Organizational change is a difficult multi-step process that requires leadership

موسسه اسکرام ایران

Each Team Answer

Should we use Scrum?

Why or why not?

Each Team Answer

What will happen if we don't change?

Each Team Answer

Why haven't previous efforts worked?

موسسه اسکرام ایران

Each Team Answer

Whose support do we need the most?

What do we need from them?

موسسه اسکرام ایران

Each Team Answer

If we change soon, what will be true:

- In 6 months?
- In 1 year?

Agile Transition Backlog

Create and order a backlog for your organization's transition to agility with Scrum.

Grooming the Scrum Implementation Backlog

- Who will be the Product Owner for this Scrum Implementation Backlog?
- Who will work with this Product Owner to groom this backlog?
  - Ideally, 1 person from each team in class
  - Plus the highest ranking person in class

Starting

موسسه اسکرام ایران

How many teams should we have?

- How many concurrent projects?
- How many people in each team?

Who Fills These Roles and Why?

Product Owner

Scrum Master

موسسه اسکرام ایران

Product Backlog

- Is there one today?
  - If not, when will one exist?

- How and where will the Product Backlog be managed and made visible?

- Are the PBIs estimated and ordered?
  - If not, when and where will that occur?
  - Who will schedule it?
  - How will PBIs be estimated?

The First Sprint

- How long will Sprints be and why?

- When will the first one begin and end?

- What might be a valid Sprint Goal for your first Sprint?

Information Radiators

- Where will they be?
- What will they show?

Schedule These Events

- Sprint Planning
- Daily Scrums
- Sprint Review
- Sprint Retrospective

ScrumBut (n): A practice present within a Scrum Team that impedes hyper-performance.

- Every Scrum Team will experience ScrumButs
- The Scrum Master's job is to mitigate them
- Doesn't necessarily mean you aren't using Scrum
- Means you aren't using Scrum to its potential

ScrumBut

We are doing Scrum, but . . .

Format: <ScrumBut> <Reason> <Workaround>

Example:

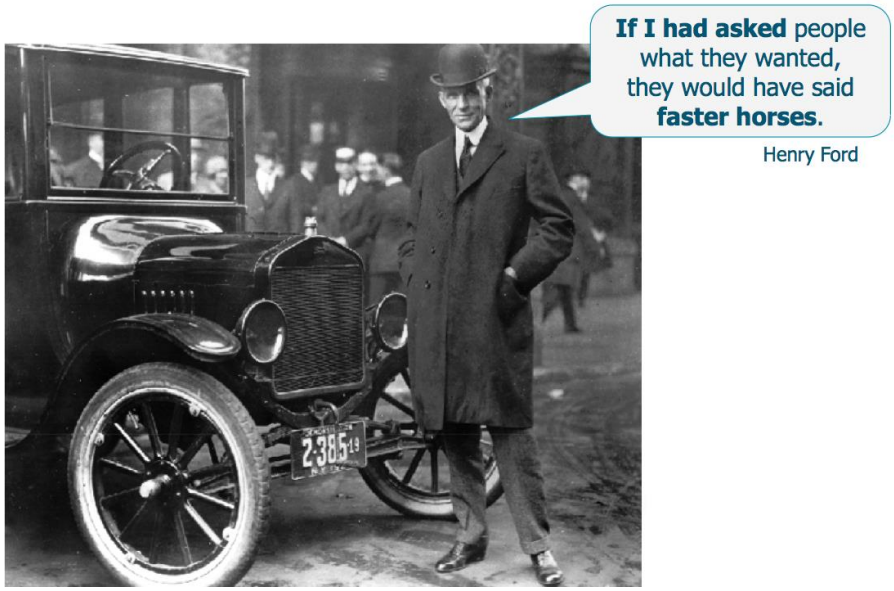
We use Scrum, but daily Scrum meetings aren't useful, so we don't do them.

A ScrumBut Mitigation Recipe

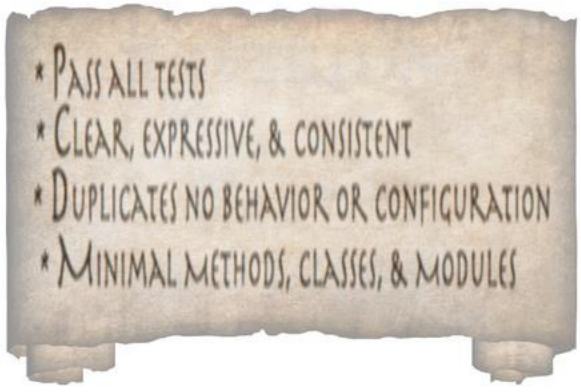
1. Identify the ScrumBut
2. State it clearly
3. Understand why it exists
4. Decide to change the causality
5. Observe the change
6. If ScrumBut still exists, repeat from 3

موسسه اسکرام ایران

Thanks



4 rules of Simple Code



Kent Beck
http://www.c2.com/cgi/wiki?XpSimplicityRules

What is technical debt anyway?

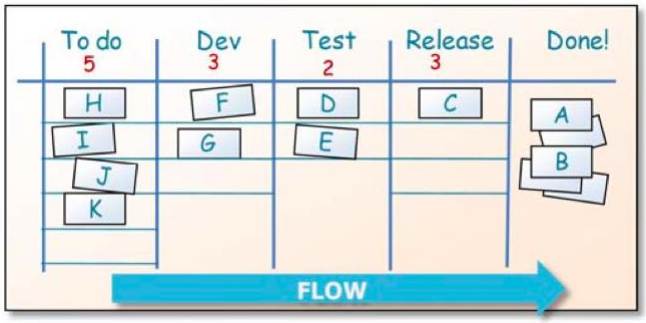
Anything about your code & development environment that slows you down. For example:

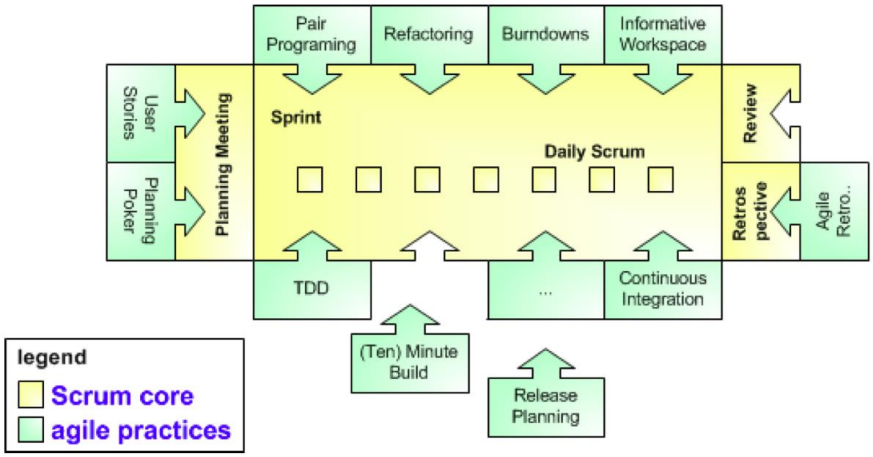
- Unclear, unreadable code.
- Lack of test automation, build automation, deployment automation, and anything else that could be automated that you do manually today.
- Duplicate code.
- Tangled architecture & unnecessarily complex dependencies.
- Slow, ineffective tools.
- Uncommitted code & long-lived branches (hides problems that will slow you down later).
- Important technical documentation that is missing or out-of-date.
- Unnecessary technical documentation that is being maintained and kept up-to-date.
- Lack of test environments.
- Long build-test cycle & lack of continuous integration.

Kanban in a Nutshell

- **Visualize the workflow**
  - Split the work into pieces, write each item on a card and put on the wall.
  - Use named columns to illustrate where each item is in the workflow.
- **Limit Work In Progress (WIP)** – assign explicit limits to how many items may be in progress at each workflow state.
- **Measure the lead time** (average time to complete one item, sometimes called "cycle time"), optimize the process to make lead time as small and predictable as possible.

Kanban in a Nutshell





Characteristics of Scrum

– time boxed

– self-managing teams

– cross functional

– work in iterations

– turn requirements into potentially shippable product