



MALMÖ UNIVERSITY

INLEDANDE WEBBPROGRAMMERING MED JAVASCRIPT

INTRODUCTION TO WEB PROGRAMMING USING JAVASCRIPT

ME152A

L10: AGILE: SCRUM PROJECT DEVELOPMENT AND  
MANAGEMENT

# OUTLINE

- Agility
- Agile manifesto
- SCRUM
- SCRUM model
- SCRUM roles
- SCRUM process
- Our plan



# AGILITY

- Agile originated within IT and software development, and is well-known within these communities.
- The agile movement is a paradigm shift in how we deliver benefit that has been with us since the mid 1990's, even though the actual word “agile” was coined in 2001.

# AGILITY (CONT'D)

- Agile is about enabling constantly improving and self-managed knowledge worker teams with inner drive.
- Principles:
  - having the customer in the middle of delivery,
  - a focus on delivering value instead of delivering on goals, and
  - tools and techniques that enable the team to deliver credible, creative results in spite of starting with unclear, fuzzy goals and requirements.

# AGILITY (CONT'D)

In agile teams, we don't visit a document to find out how to meet customer needs; we visit a real live human being.

The agile organisation  
**§ has the living customer perspective in the middle of delivery**  
to ensure immediate feedback and correction

# AGILITY: THE VALUE (CONT'D)

## Project problems in the CHAOS report

1. Lack of User Input	12.8%
2. Incomplete requirements	12.3%
3. Changing Requirements	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

# AGILITY: THE VALUE (CONT'D)

- Being on time, budget and specification can be seen as the perfect project, but does it infer value?
  - The answer to the question above must be no.
  - There is not necessarily a relationship between time, budget, specification and value.
  - This is because the plan, budget and specification are often wrong.
  - **To deliver optimal value:**
    - We may need to deliver outside of time and budget.
    - We may also discover the idea was not quite as good as we first envisioned and, because of this, we may end up using very little of the original budget.
    - To deliver on value we may also have to deliver something very different from the original idea, if we discover that the basic idea needs improvement.

# AGILITY: THE VALUE (CONT'D)

*Getting the result on time and budget is a good start, and agile can deliver on time and budget, but if you stop there, you will easily lose what is most important—the actual value.*

Term

Value

$$\text{Benefits} - \text{Costs} = \text{Value}$$

NewBenefits - CostOfDeliveringNewGreatProductOrIdea => WeGetValue

# TRADITIONAL METHODS

- Are based on some presumptions:
  - Requirements are stable
  - Technology is well known to the team and mature in its implementation
  - There will be no surprises, no changes, no deviations
  - Everyone on the team will have a consistent skill level
  - Product Management has patience
  - Senior Management has patience
  - Customers have patience

# REALITY

- 31% of the software development projects are cancelled
- 75% are considered failures from those who initiated them
- Every second project exceeds the envisioned budget by 200%

Source: Standish group “CHAOS reports” and Gartner Group

# THE NEED FOR AGILITY

- The need to respond to change:
  - Changes in user requirements
  - Changes in user needs
  - Changes in priorities
  - Changes in design
  - Changes in technologies
- 
- While the real questions to address are:
  - How do we address the change when it occurs?
  - How do we minimize impact and cost?

# **WHAT IS AGILE (AGILITY)?**

From the Dictionary:

- Be able to move quickly and easily
- Be able to think and understand quickly

In practice

- Ability to swiftly respond to changes
- Quick reprioritization and fast respond
- Interactive, iterative and incremental
- Value based rather than a plan based

# THE BIRTH OF AGILE

- Industry leaders discovered similarities between their methodologies
  - XP (eXtreme Programming) – Kent Beck
  - Scrum – Ken Schwaber
  - Lean Software Development – Mary Poppendieck
  - Crystal family – Alistair Cockburn
  - Feature Driven Development – Peter Coad
- They met in 2001 – and decided to name the “family” of methodologies **Agile**
- They also created the Agile Manifesto, and defining **Values** and **Principles**

[www.agilemanifesto.org](http://www.agilemanifesto.org)

# THE AGILE MANIFESTO

1. Satisfying customer is a top priority
2. Welcome changing requirements
3. Deliver working software frequently
4. Business people and developers must work together
5. Build projects around motivated individuals.
6. Effective methods for conveying information
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development.
9. Continuous attention to technical excellence and good design
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. Regular reflection on how to become more effective

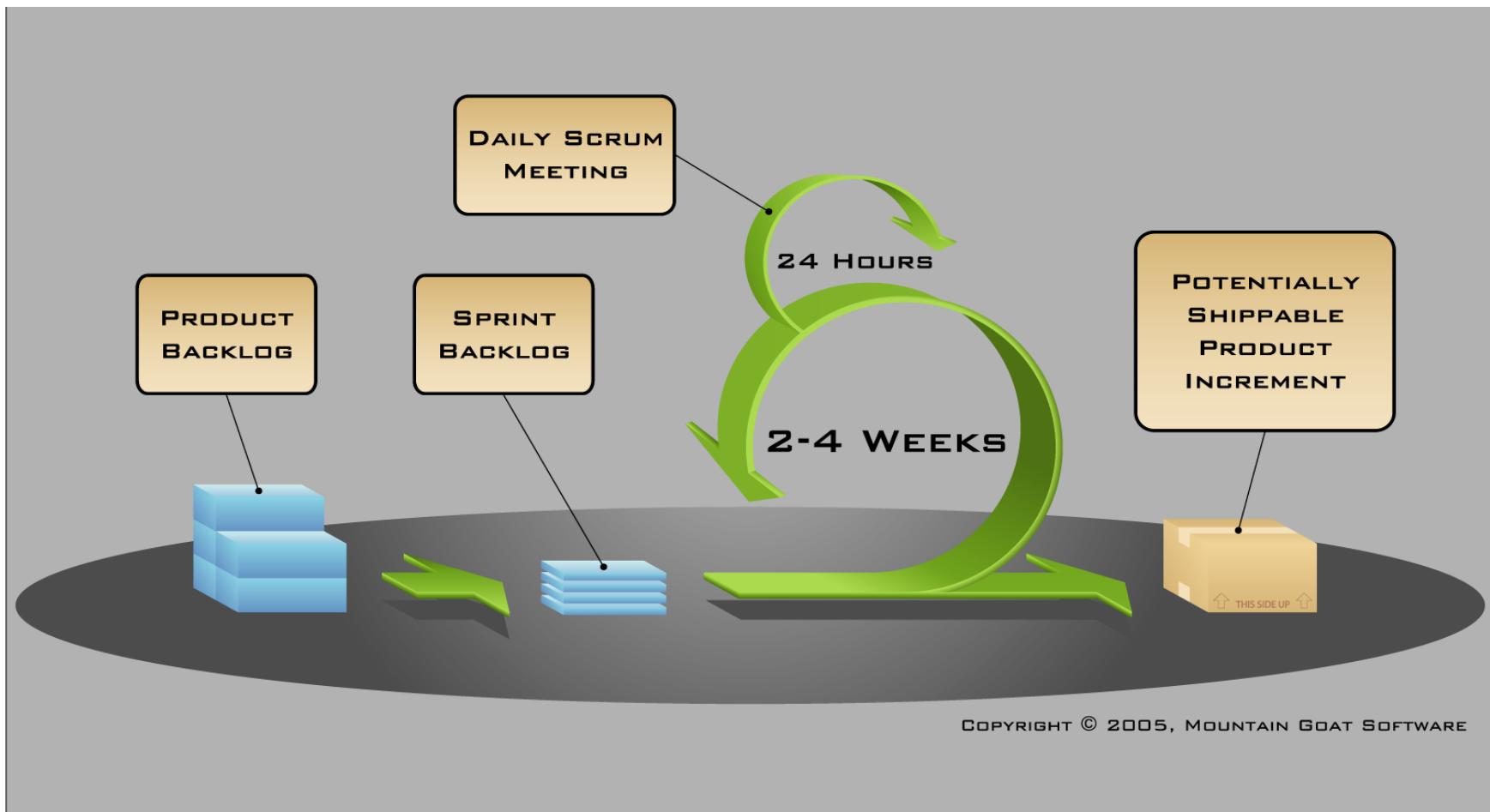
# THE THREE “I” KEYWORDS

1. Iterative
2. Incremental
3. Interaction

# SCRUM: A MANAGEMENT FRAMEWORK

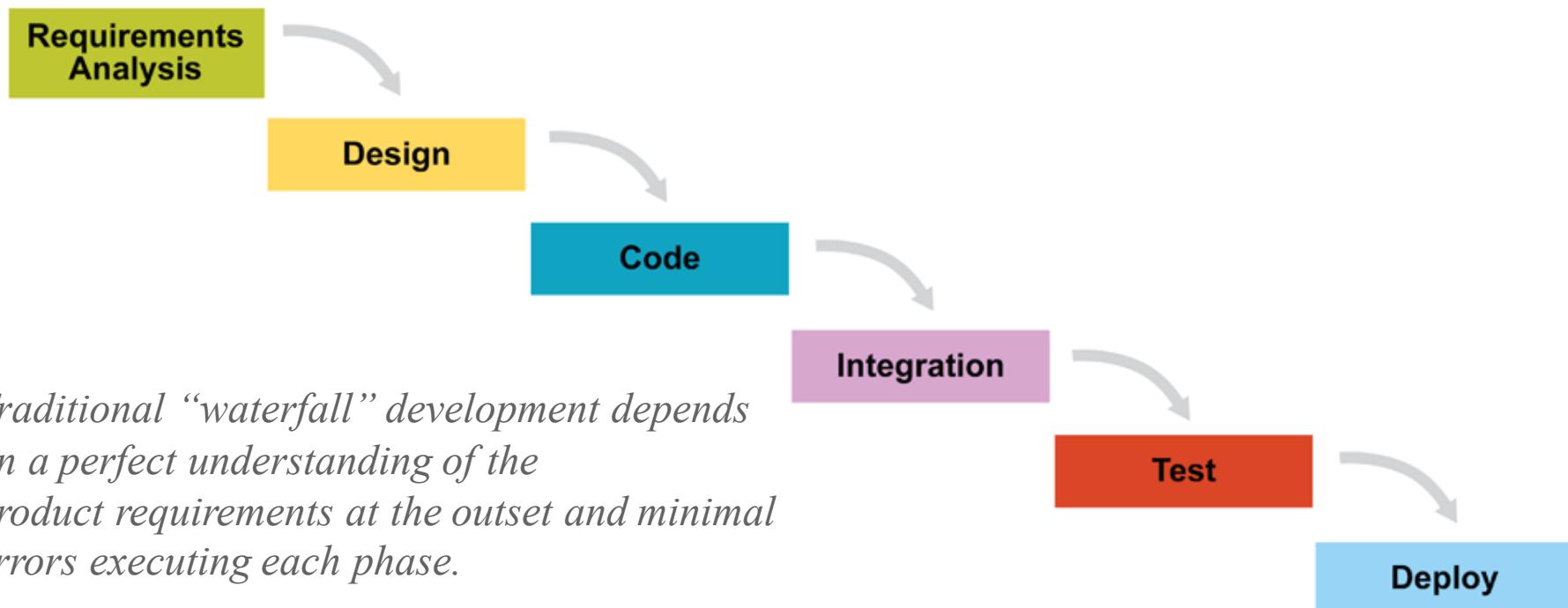
- Scrum is a management framework for incremental product development using one or more cross-functional, self-organizing teams of about seven people each.
- It provides a structure of roles, meetings, rules, and artifacts. Teams are responsible for creating and adapting their processes within this framework.
- Scrum uses fixed-length iterations, called Sprints, which are typically 1-2 weeks long (never more than 30 days). Scrum teams attempt to build a potentially shippable (properly tested) product increment every iteration.

# SCRUM (VISUALLY)



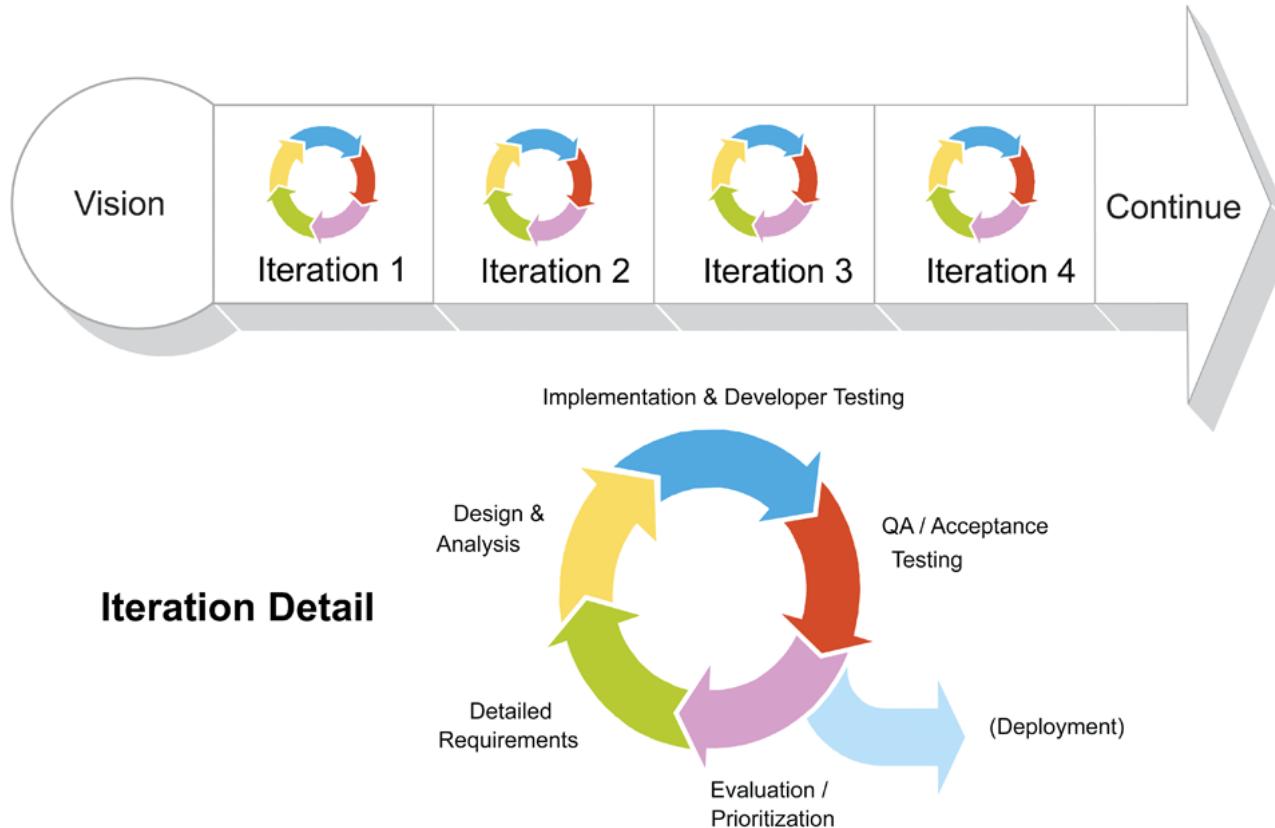
# **SCRUM AS AN ALTERNATIVE TO WATERFALL**

Scrum's incremental, iterative approach trades the traditional phases of "waterfall" development for the ability to develop a subset of high-value features first, incorporating feedback sooner.



*Traditional “waterfall” development depends on a perfect understanding of the product requirements at the outset and minimal errors executing each phase.*

# SCRUM MODEL



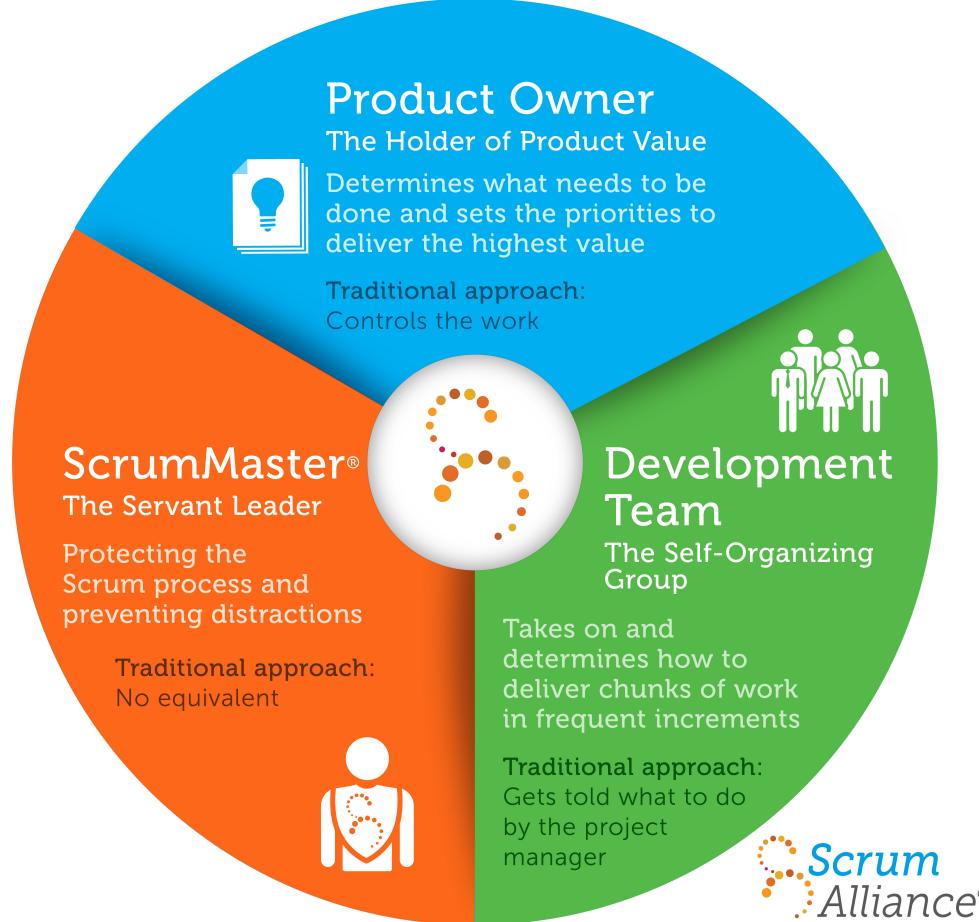
*Scrum blends all development activities  
into each iteration, adapting to  
discovered realities at fixed intervals*

# SCRUM ROLES: PRODUCT OWNER, SCRUM MASTER AND DEVELOPMENT TEAM

Scrum Roles: A different way of thinking, a better way to drive success

Scrum roles differ from traditional project roles.

By collaborating, a Scrum team delivers more business value, faster.



Scrum  
Alliance®

<https://www.scrumalliance.org/scrum/files/03/03df6a07-956c-4ced-82aa-5aec91a88e62.jpg>



MALMÖ UNIVERSITY

# PRODUCT OWNER

- Single person responsible for maximizing the return on investment (ROI) of the development effort
- Responsible for product vision
- Constantly re-prioritizes the Product Backlog, adjusting any longterm expectations such as release plans
- Final arbiter of requirements questions
- Accepts or rejects each product increment
- Decides whether to ship
- Decides whether to continue development
- Considers stakeholder interests
- May contribute as a team member

# SCRUM DEVELOPMENT TEAM

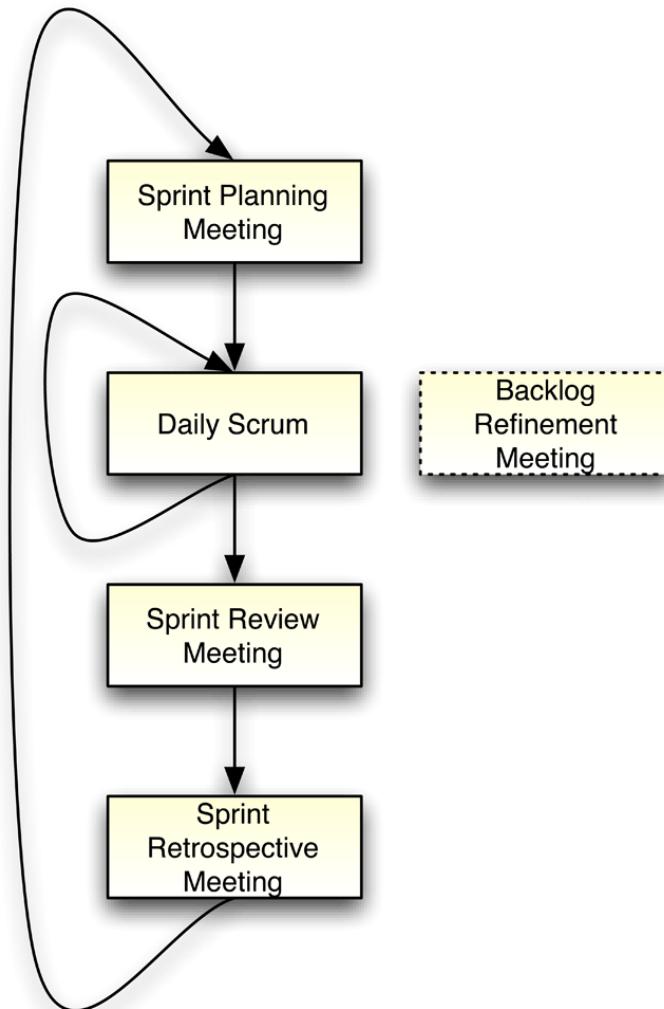
- Cross-functional (e.g., includes members with testing skills, and often others not traditionally called developers: business analysts, domain experts, etc.) Self-organizing / self-managing, without externally assigned roles
- Negotiates commitments with the Product Owner, one Sprint at a time
- Has autonomy regarding how to reach commitments
- Intensely collaborative
- Most successful when located in one team room, particularly for the first few Sprints
- Most successful with long-term, full-time membership. Scrum moves work to a flexible learning team and avoids moving people or splitting them between teams.
- 3-9 members (originally  $7 \pm 2$  members)

# SCRUMMASTER

- Facilitates the Scrum process
- Helps resolve impediments
- Creates an environment conducive to team self-organization
- Captures empirical data to adjust forecasts
- Shields the team from external interference and distractions to keep it in group flow (a.k.a. the zone)
- Enforces timeboxes
- Keeps Scrum artifacts visible
- Promotes improved engineering practices
- Has no management authority over the team (anyone with authority over the team is by definition not its ScrumMaster)
- Not a coordinator

# SCRUM MEETINGS: THE FLOW

*All Scrum Meetings are facilitated by the ScrumMaster, who has no decision-making authority at these meetings.*

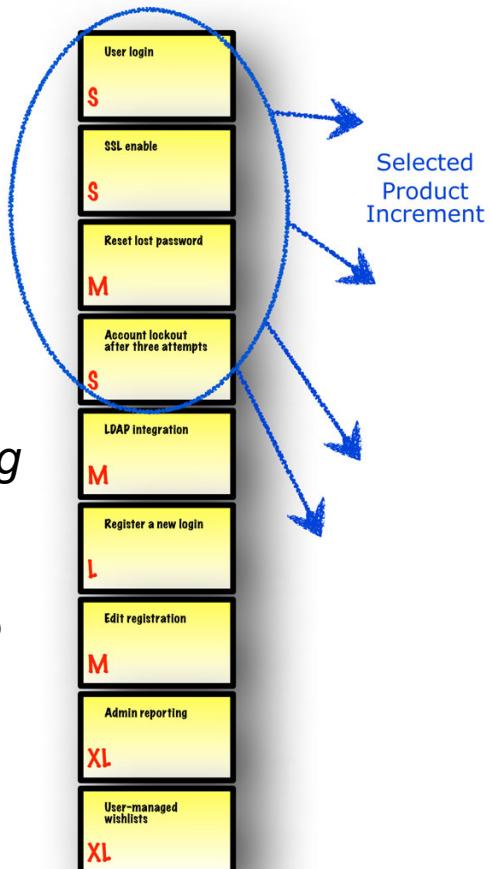


# SPRINT PLANNING MEETING

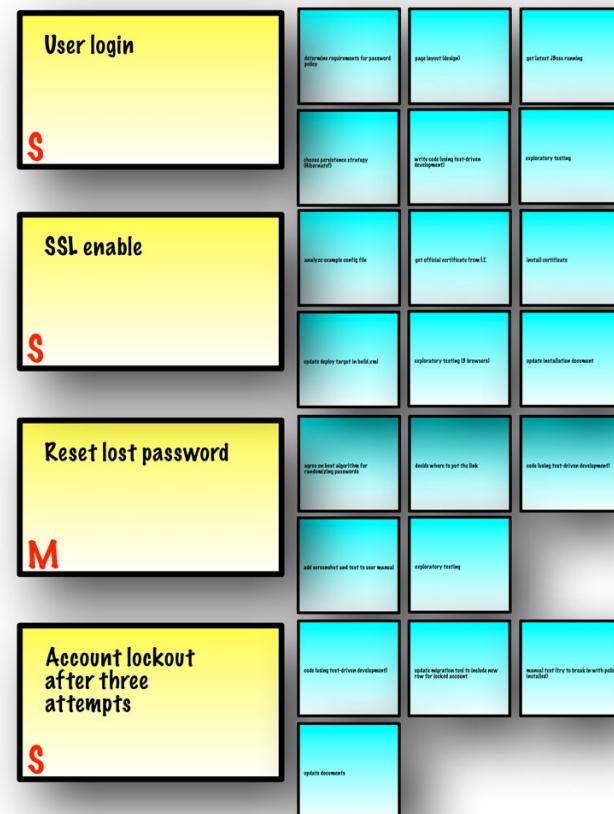
- At the beginning of each Sprint, the Product Owner and team hold a Sprint Planning Meeting to negotiate which Product Backlog Items they will attempt to convert to working product during the Sprint.
- The Product Owner is responsible for declaring which items are the most important to the business.
- The team is responsible for selecting the amount of work they feel they can implement without accruing technical debt.
- The team “pulls” work from the Product Backlog to the Sprint Backlog.
- When teams are given complex work that has inherent uncertainty, they must work together to intuitively gauge their capacity to commit to items, while learning from previous Sprints.

# FROM PRODUCT BACKLOG TO SPRINT BACKLOG

Product Backlog



Sprint Backlog



*Sprint Planning Meeting outcome is committed Product Backlog Items (PBIs) and subordinate Sprint Tasks.*

# DAILY SCRUM AND SPRINT EXECUTION

- Every day at the same time and place, the Scrum Development Team members spend a total of 15 minutes reporting to each other.
- Each team member summarizes (the “W” questions):
  - **what** he did the previous day,
  - **what** he will do today, and
  - **what** impediments he faces.
- Standing up at the Daily Scrum will help keep it short. Topics that require additional attention may be discussed by whomever is interested after every team member has reported.
- The team may find it useful to maintain a current Sprint Task List, a Sprint Burndown Chart, and an Impediments List.
- During Sprint execution it is common to discover additional tasks necessary to achieve the Sprint goals.

# SPRINT REVIEW MEETING

- After Sprint execution, the team holds a Sprint Review Meeting to demonstrate a working product increment to the Product Owner and everyone else who is interested.
- The meeting should feature a live demonstration, not a report
- After the demonstration, the Product Owner reviews the commitments made at the Sprint Planning Meeting and declares which items he now considers done.
- For example, a software item that is merely “code complete” is considered not done, because untested software isn’t shippable.

## **SPRINT REVIEW MEETING (CONT'D)**

- Incomplete items are returned to the Product Backlog and ranked according to the Product Owner's revised priorities as candidates for future Sprints.
- The ScrumMaster helps the Product Owner and stakeholders convert their feedback to new Product Backlog Items for prioritization by the Product Owner.
- If the Product Owner feels that the newly discovered scope is more important than the original expectations, new scope displaces old scope in the Product Backlog.

# SPRINT RETROSPECTIVE

- Each Sprint ends with a retrospective. At this meeting, the team reflects on its own process. They inspect their behavior and take action to adapt it for future Sprints.
- A common impediment to full transparency on the team is the presence of people who conduct performance appraisals.
- Another impediment to an insightful retrospective is the human tendency to jump to conclusions and propose actions too quickly.
- A third impediment to psychological safety is geographic distribution. Geographically dispersed teams usually do not collaborate as well as those in team rooms.
- Retrospectives often expose organizational impediments.
- ScrumMasters should use a variety of techniques to facilitate retrospectives, including silent writing, timelines, and satisfaction histograms. In all cases, the goals are to gain a common understanding of multiple perspectives and to develop actions that will take the team to the next level.

# BACKLOG REFINEMENT MEETING

- Most Product Backlog Items (PBIs) initially need refinement because they are too large and poorly understood.
- Teams have found it useful to take a little time out of Sprint Execution — every Sprint — to help prepare the Product Backlog for the next Sprint Planning Meeting.
- In the Backlog Refinement Meeting, the team considers the effort they would expend to complete items in the Product Backlog and provides other technical information to help the Product Owner prioritize them.
- Large vague items are split and clarified, considering both business and technical concerns.
- Sometimes a subset of the team, in conjunction with the Product Owner and other stakeholders, will compose and split Product Backlog Items before involving the entire team in estimation.

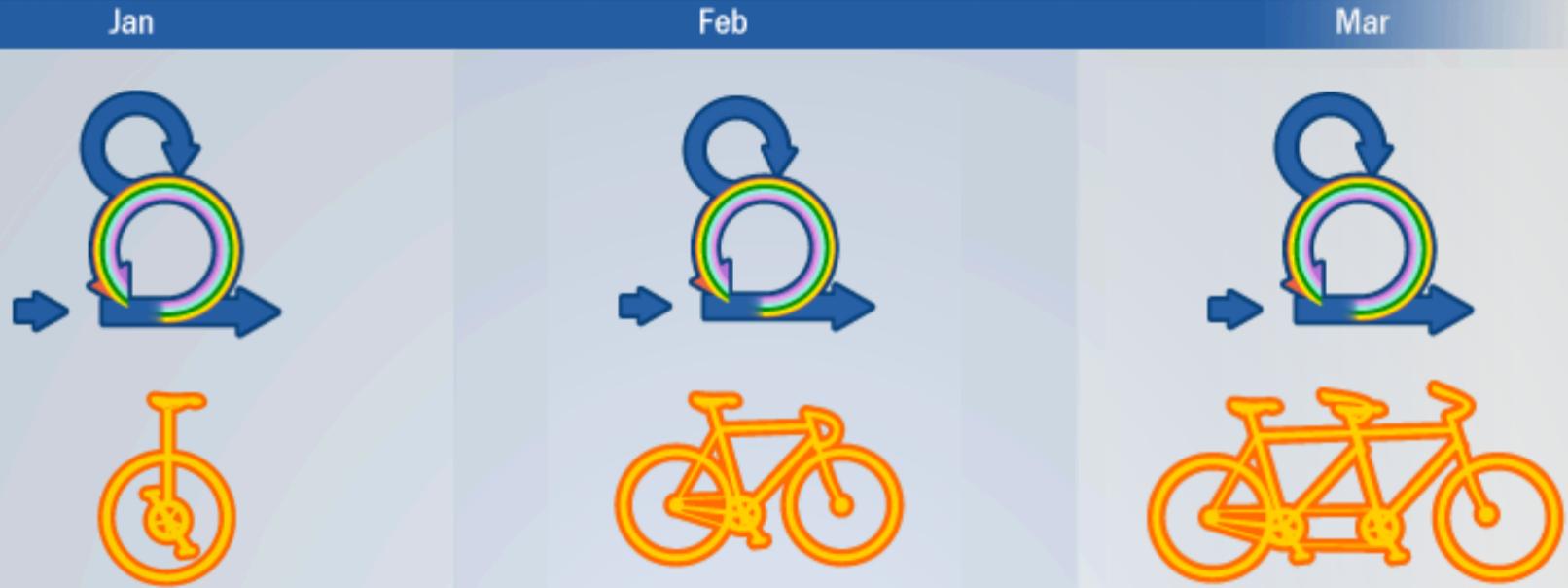
# OUR PLAN FOR THIS COURSE

<http://scrumtrainingseries.com/SprintPlanningMeeting/SprintPlanningMeeting.htm>

The diagram illustrates the Scrum framework components:

- PRODUCT BACKLOG:**
  - View Grades, current semester:** As a student I can see my grades online so that I don't have to wait until I get to school to know whether I'm passing.
  - Acceptance Criteria:** Columns align neatly on FingerFly 4.1 and iPhone.
  - EFFORT: SMALL.**
  - Update Grades, current semester:** As a teacher, I can update grades online so I don't depend on administrators to do it for me.
  - EFFORT: MEDIUM.**
  - View Grades, previous semester:** EFFORT: SMALL.
  - Attendance:** EFFORT: MEDIUM.
  - GPA:** EFFORT: SMALL.
  - Report Cards:** EFFORT: EXTRA LARGE.
  - Event Calendar:**
  - Alumni Archives:**
- SPRINT BACKLOG:**
  - COMMITTED BACKLOG ITEMS:** NOT STARTED, IN PROGRESS, COMPLETED
- SPRINT TIMEBOX:** Two weeks
- MEETING TIMEBOX:** A circular timer icon.
- CHARACTERS:** PRODUCT OWNER (chef), DEVELOPMENT TEAM (baker, pilot, painter, scientist), SCRUMMASTER (coach).

# SUMMARY OF SCRUM



# OUR PLAN

- **Stand-up (Daily SCRUM) meeting:** 15 min every working day
  - what you did the previous day,
  - what you will do today, and
  - what impediments you face.
- **Sprint planning:** 1h – once per week
  - In each Sprint, the Product Owner and team hold a Sprint Planning Meeting to negotiate which Product Backlog Items are most important
- **Sprint review:** 30 min – once per week
  - demonstrate a working product increment to the Product Owner
- **Sprint retrospective:** 30 min – once per week
  - the team reflects on its own process. They inspect their behavior and take action to adapt it for future Sprints.
- **Backlog Refinement Meeting:** the team
  - refinement because they are too large and poorly understood.
  - help prepare the Product Backlog for the next Sprint Planning Meeting.
  - provides other technical information to help the Product Owner prioritize them.  
Large vague items are split and clarified

All Scrum Meetings are facilitated by the ScrumMaster, has no decision-making authority at these meetings.

# OUR PLAN (CONT'D)

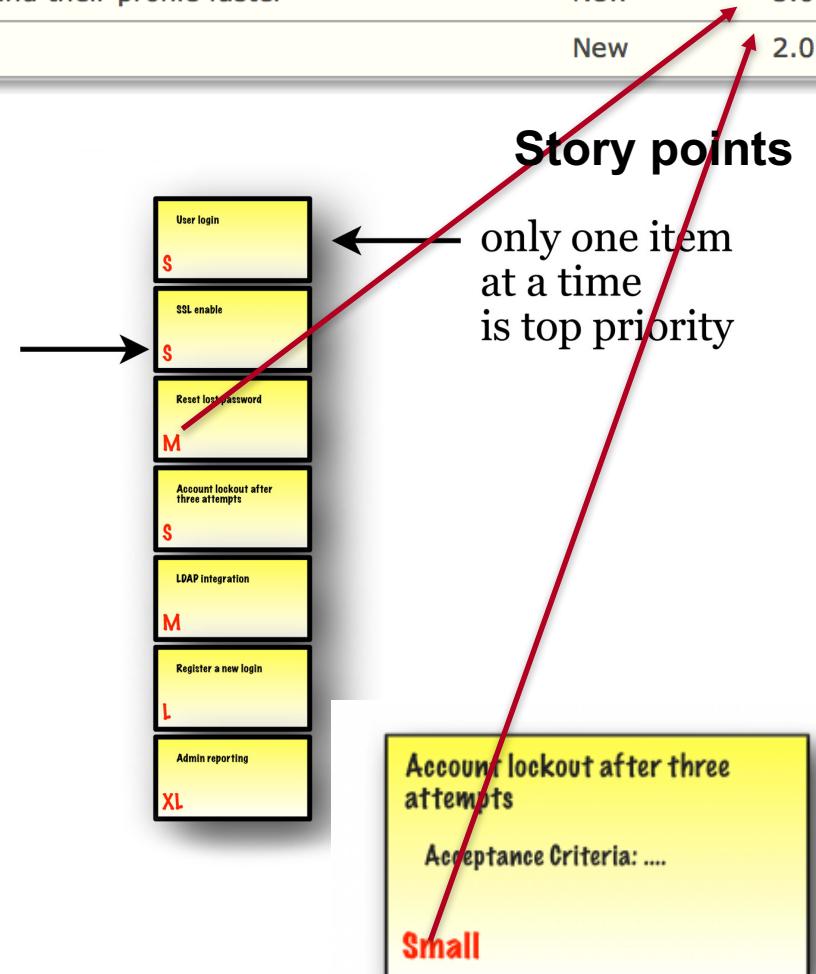
- **Projects:** PELARS and SMART HOME
- **Technology and Support:** Sebbe
- **Product owners:** Bato and Nils
- **Teams:** two groups including the two SCRUM Masters from the team
- **Technology for project management and tracking:**
  - Redmine: <http://139.59.136.105/redmine> (all of you need to register: <http://139.59.136.105/redmine/account/register>)
  - Slack: (Sebbe will provide group discussions for each group)
  - itslearning: for final submission of the project and grading
  - Github: online source code hosting and version control (Sebbe)
- **Note: YOU MUST USE THE TOOLS, OTHERWISE...?**

# PRODUCT BACKLOG

Product Backlog			Close completed Sprints	5
4	As a user I want to be able to find clients by last name, so that I can find their profile faster		New	3.0
13	As a (...) I want to be able to (...) so that (...)		New	2.0

- Specifies the *what* more than the *how* of a customer-centric feature
- Story point: is used to measure the effort required to implement a **story**. its a number that tells the team how hard the **story** is.
- Constantly re-prioritized by the Product Owner
- Any stakeholder (including the Team) can add items

top items  
are more  
granular

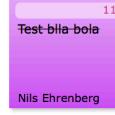
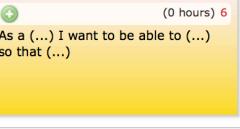


# SPRINT BACKLOG

Sprint one	2016-04-15	2016-04-22
5 As a system admin user I want to be able to configure user settings so that I can control access	Closed	3.0
Sprint two	2016-04-22	2016-05-02
10 As a system admin user I want to be able to add new users when required, so that I can...	New	8.0
6 As a (...) I want to be able to (...) so that (...)	In Progress	

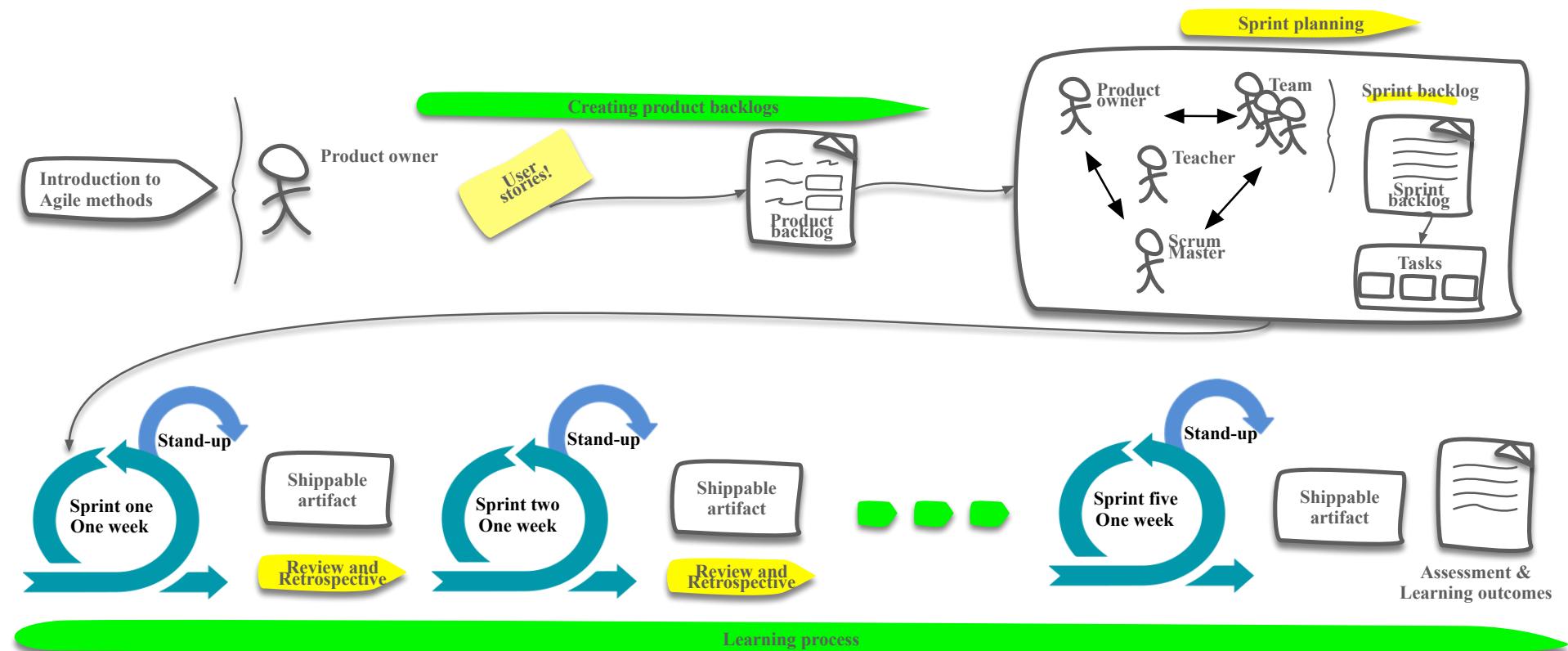
- Consists of committed user stories negotiated between the team and the Product Owner during the Sprint Planning Meeting
- Scope commitment is fixed during Sprint Execution
- **Initial tasks** are identified by the team during Sprint Planning Meeting
- Team will discover **additional tasks** needed to meet the fixed scope commitment during Sprint execution
- Visible to the team
- Referenced during the Daily Scrum Meeting

# SPRINT TASKS

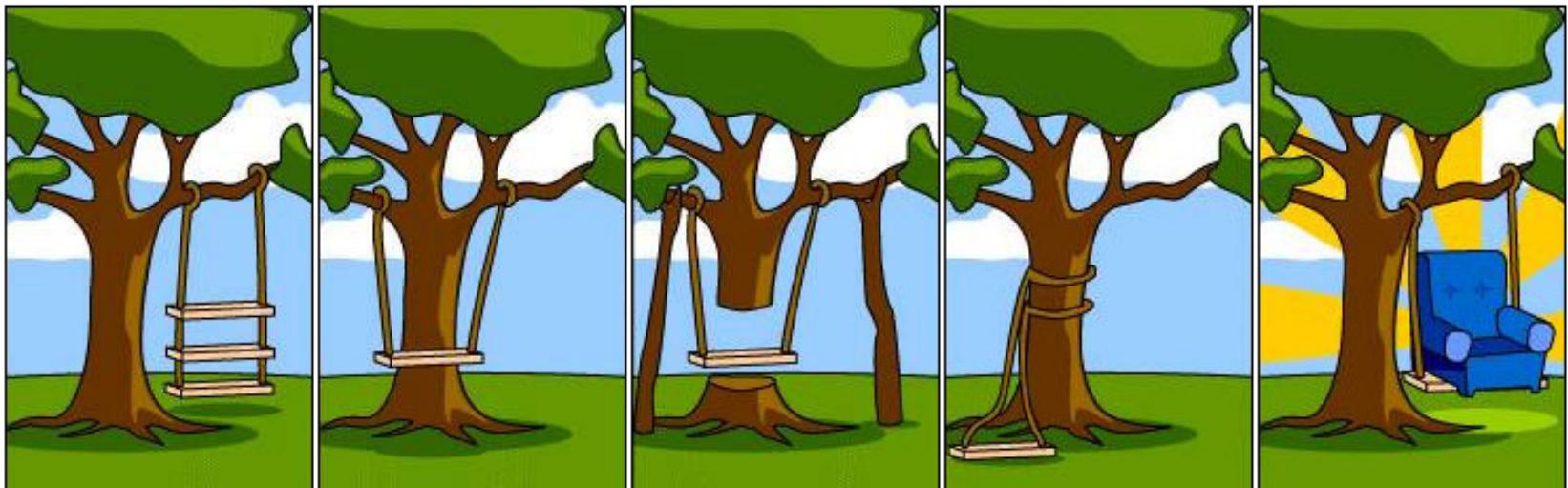
Story	New	In Progress	Resolved	Feedback	Closed	Rejected
 + Sprint Impediments						
 1 As a system admin user I want to be able to add new users when required, so that I can... (6 hours) 10 8.0	 Test implementation Bahtijar Vogel 6.0 12		 Check encryption Redmine Admin 6.0 15		 interface-mock Bahtijar Vogel 14	 Test-bla-bla Nils Ehrenberg 11
 + As a (...) I want to be able to (...) so that (...) (0 hours) 6		 Creating the interface Bahtijar Vogel 6.0 9				

- Specifies how to achieve the User stories – *what*
- Create and assign Tasks, as well the estimated hours for the task
- Requires one day or less of work
- Remaining effort is re-estimated daily, typically in hours
- During Sprint Execution, a point person may volunteer to be primarily responsible for a task
- Owned by the entire team; collaboration is expected

# SCRUM OVERVIEW: JAVASCRIPT COURSE



# WHY SCRUM?



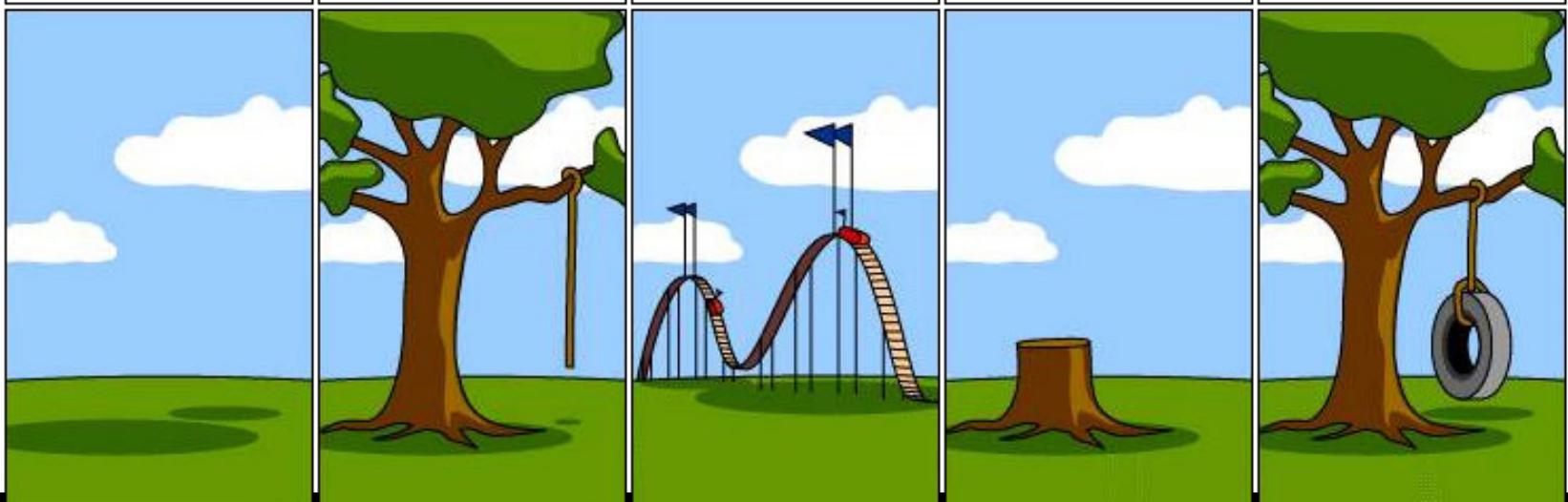
How the Customer explained it

How the Project Leader understood it

How the Analysts designed it

How the Programmer wrote it

How the Sales Rep described it



How the product was documented

How the product was tested

How the customer was billed

How the product was supported

What the customer really needed

**THANK YOU  
QUESTIONS?**

