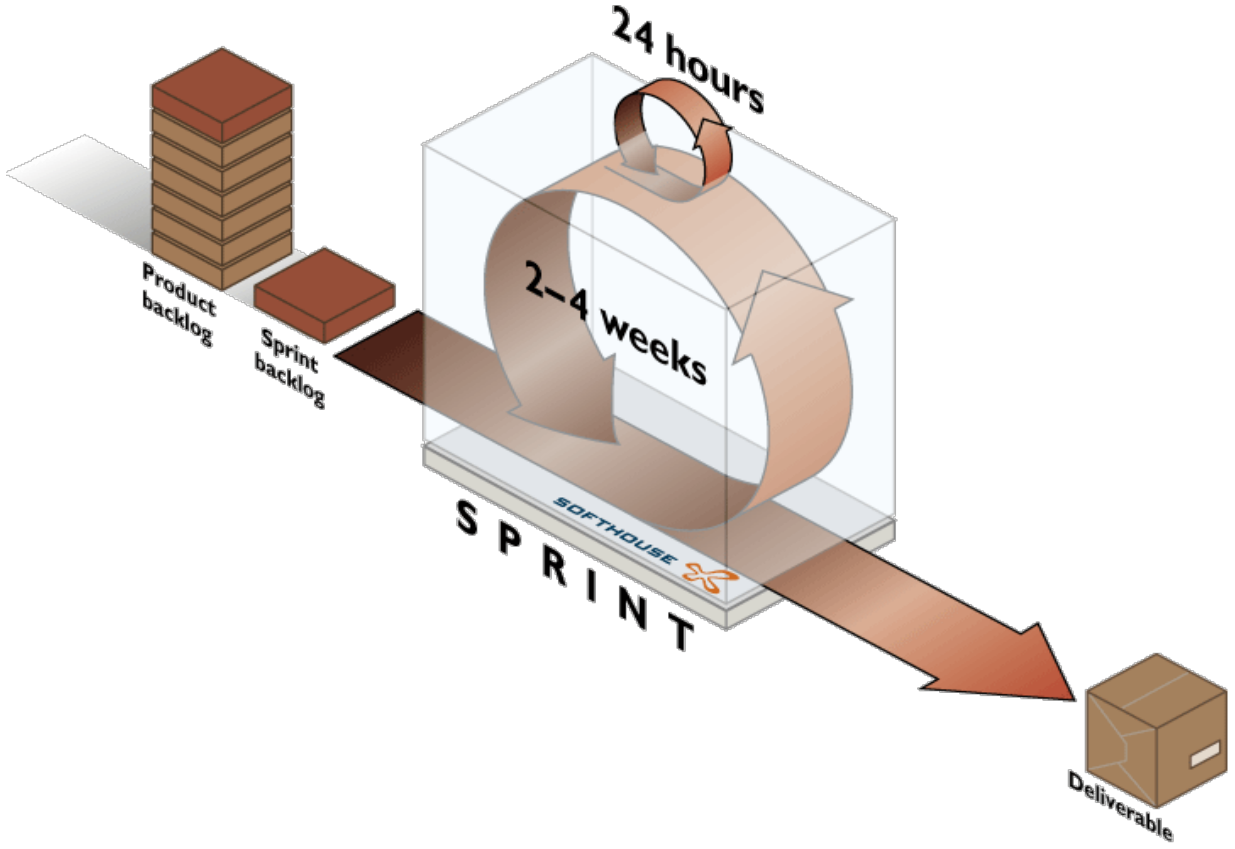
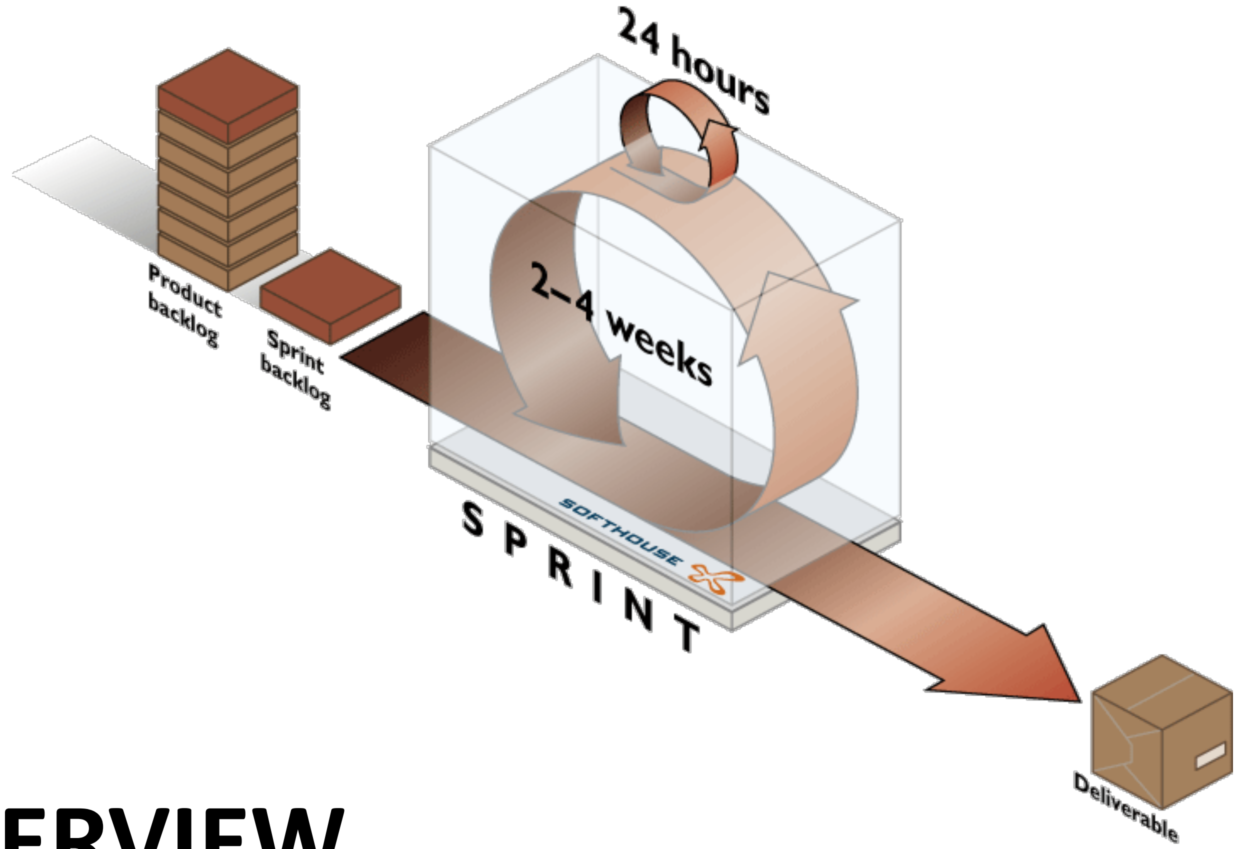
Scrum



Honesty
Visibility
Common Sense

Arne Åhlander
Ver: 3.1

Content

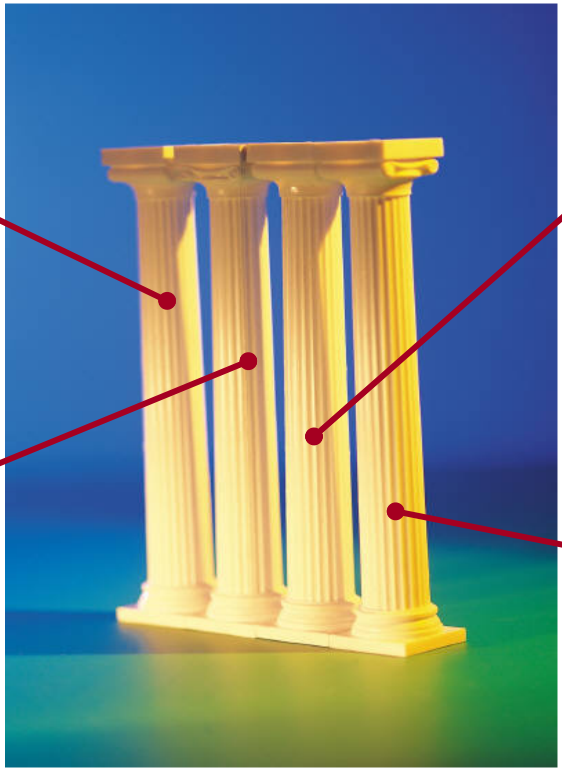


SCRUM OVERVIEW

What is Scrum?

- Empirical framework to manage and control development and deployment of complex products
  - Empiricism is dependent on frequent inspection and adaptation to reach goal
  - Inspection is dependent on transparency
  - Scrum, since it is lightweight, is fairly easy to understand, and **very** hard to implement

Four pillars of Scrum



Iterative development

High priority functionality

Incremental delivery

Team
- **Self organized**
- **Cross functional**

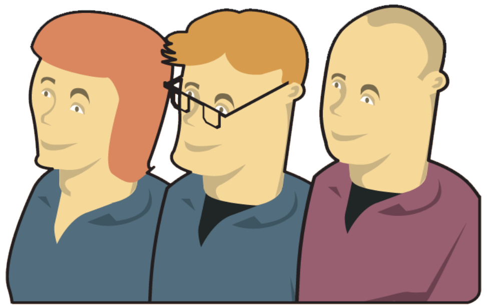
Scrum Values

- **Commitment**
- **Focus**
- **Openness**
- **Respect**
- **Courage**

Scrum Roles

- Product Owner

- ScrumMaster



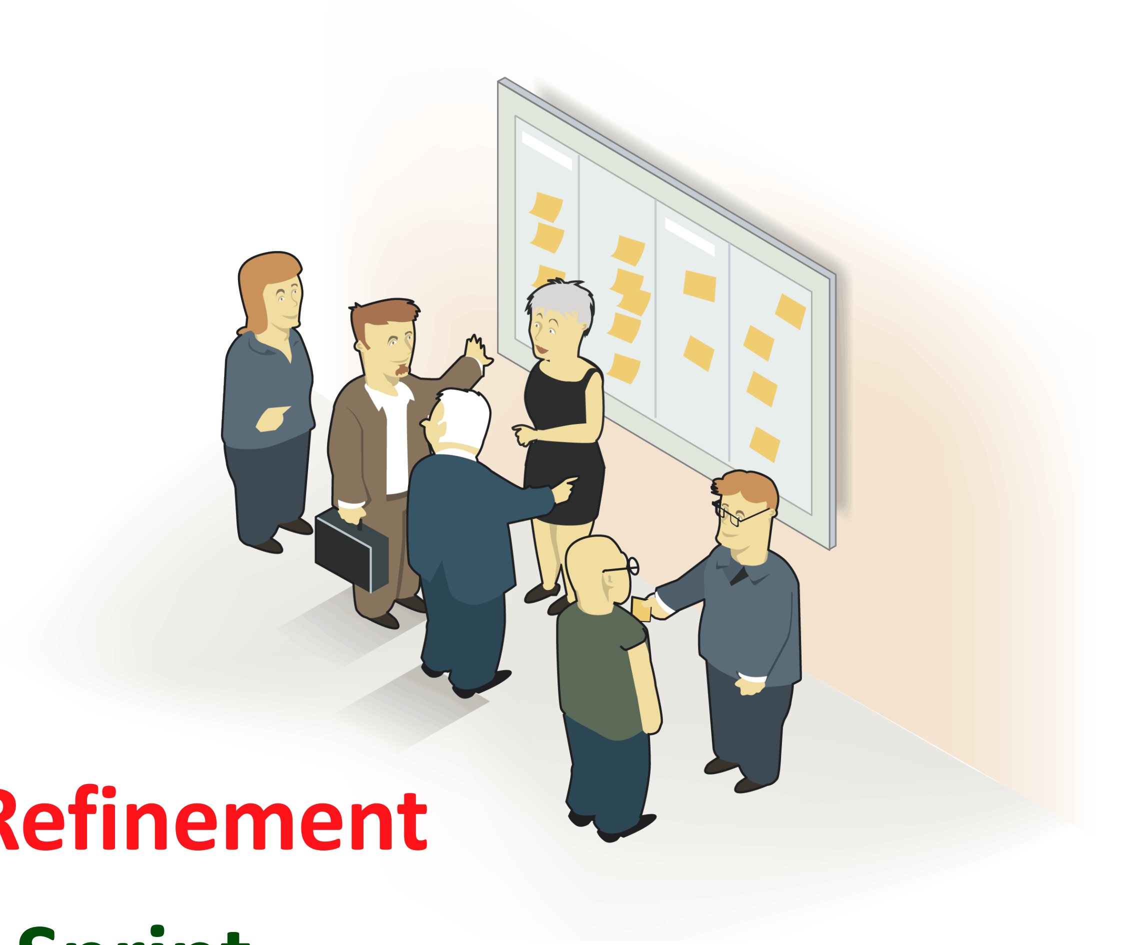
- Development Team

Scrum Meetings (Events)

- **Sprint Planning**
- **Daily Scrum**
- **Sprint Review**
- **Retrospective**

  and an activity

- **Product Backlog Refinement**
- **All contained in a Sprint**



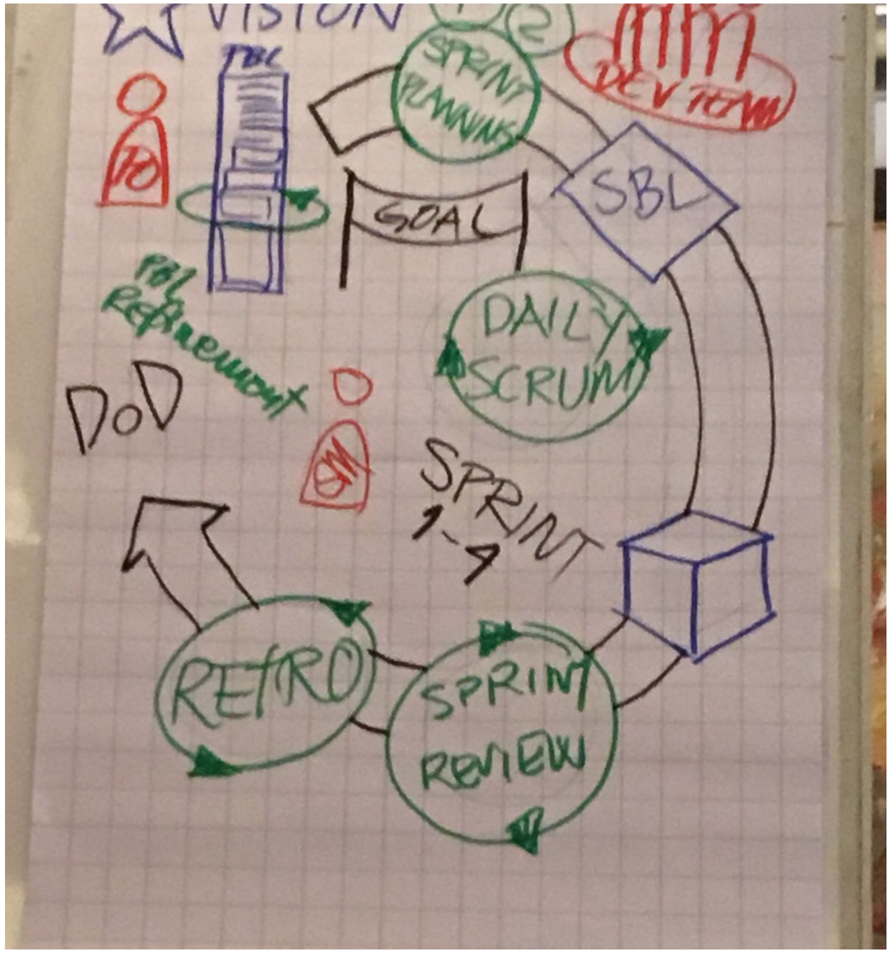
Scrum Artifacts

- Product Backlog
- Sprint Backlog
- Product Increment

Required

- Product Vision
- Sprint Burndown Chart
- Etc. (as long as you need it to support your Scrum implementation)

My illustration of Scrum



Possible Scrum Benefits

- Daily visibility and follow up
  - Everyone shall ALWAYS know what to do next!
  - When finished: PULL
  - Progress (how are we doing?)
  - Problems (Any Impediments?)
- Focus
  - We are working on the things most wanted by the customer
  - Deadline within 4 weeks!
  - Requirements stable for 4 weeks
- Deliveries
  - Working Software
  - Every 4 weeks (potientially)
- Planning
  - Clear goal
  - Detailed planning of what we know (this and next sprint)
  - Agile – customer can re-prioritize every 4th week

Challenges in Scrum

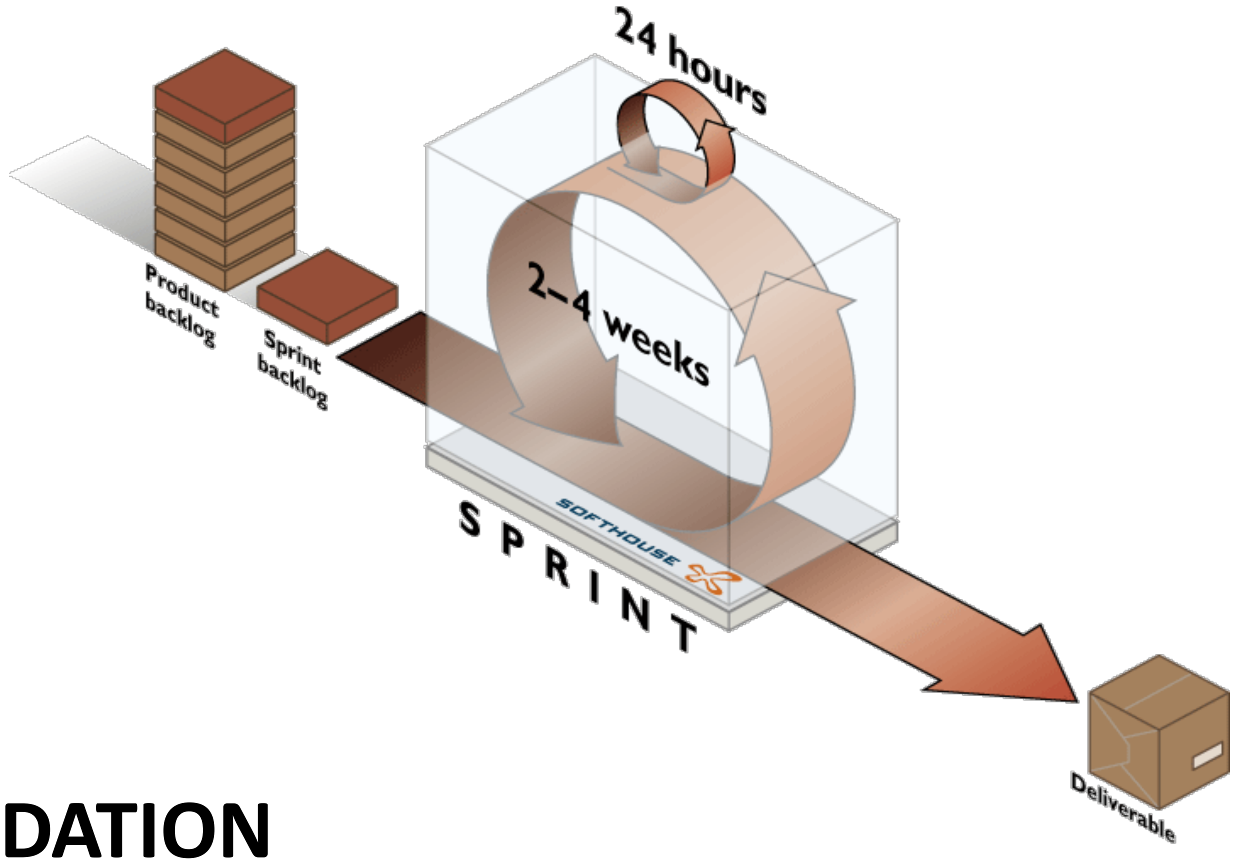
- Overwhelming details if not managed
- Cross functional team understanding
- Getting a product backlog
- Getting a product owner
- Non-dedicated roles
- Integrating support tasks
- Estimating/metric
- Daily estimates and decomposition of work
- Longer term planning
- Coordination with other teams – conflicting priorities
- Time for research/play/slack

Change!!!

- **Product Owner is used to "throwing the project over the wall" and holding engineering and development responsible for meeting needs.**

- **Scrum puts this responsibility back on the Product Owner and customers through the inspect and adapt and the Sprint Review.**

- **Developers are not used to inspecting each other's progress daily to adapt their work to optimize the chances of delivering (an increment) every Sprint.**



SCRUM FOUNDATION

Origins of Scrum

1

The New, New Product Development Game,

Harvard Business Review,

Jan. 1986,

Takeuchi and Nonaka

(LEAN)

2

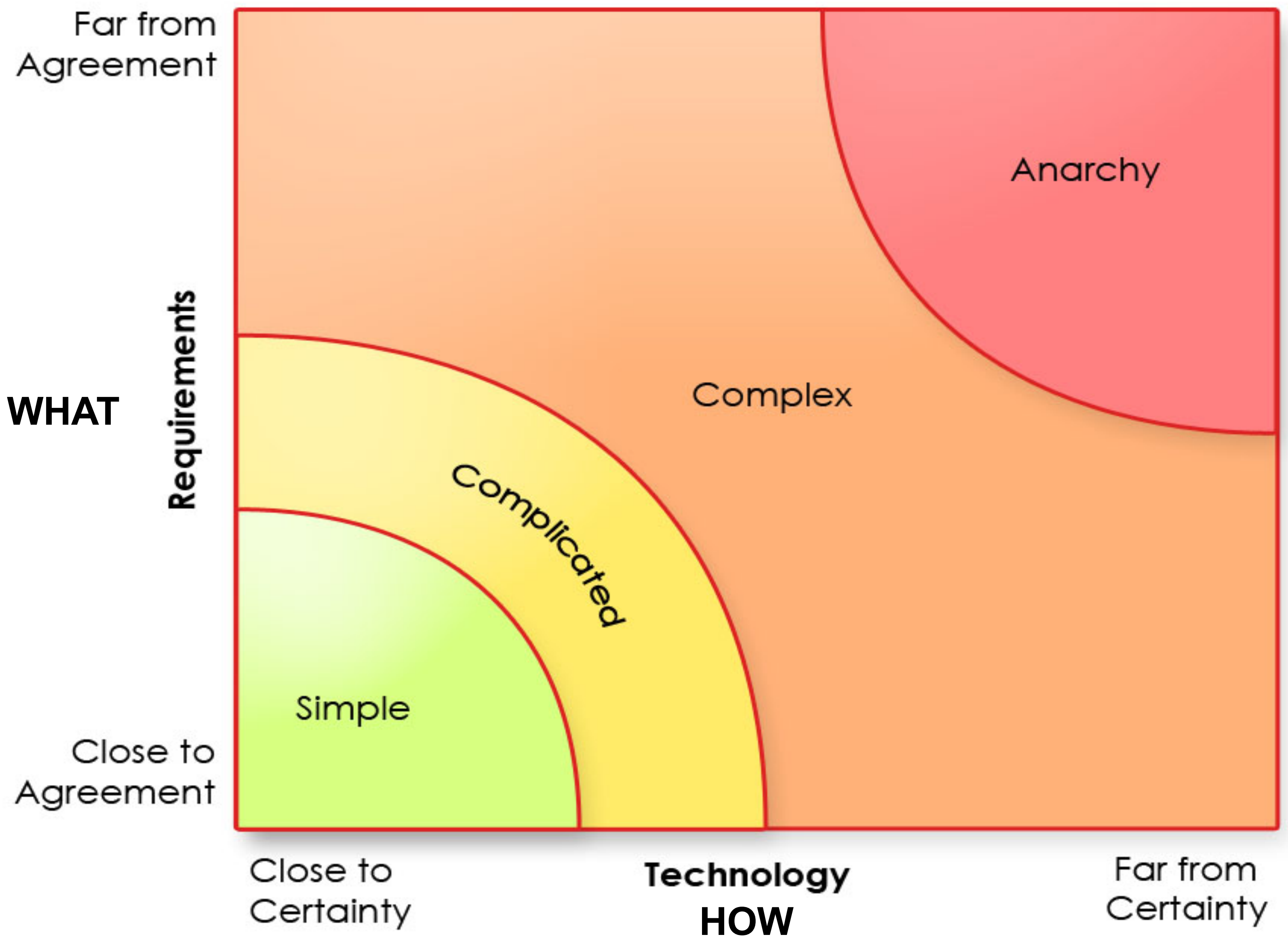
Borland

- Daily meeting
- Low number of roles

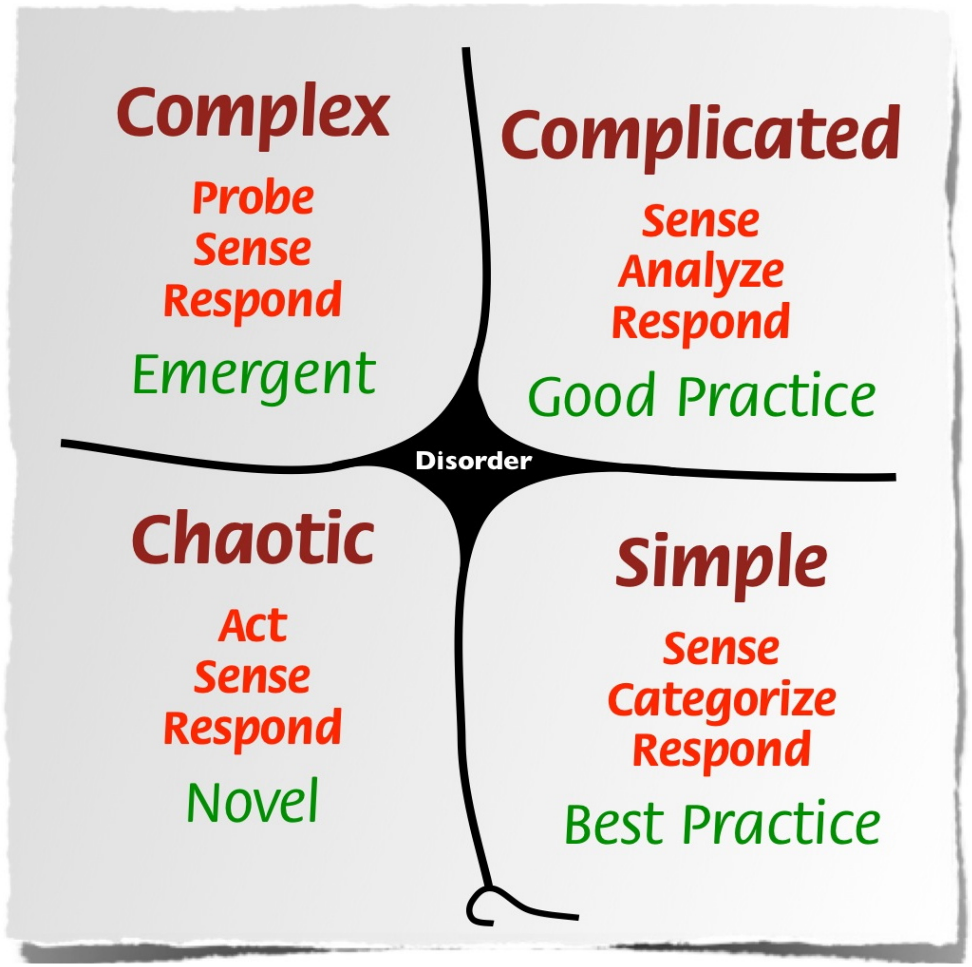
3

- Iterative development
- Incremental development
- Time boxes

The Spectrum of Process Complexity



Cynefin Framework



Dave Snowden, Cynefin framework with all five domains labeled
https://commons.wikimedia.org/wiki/File%3ACynefin_framework_Feb_2011.jpeg
By Snowden (Own creation, own work) [CC BY 3.0 (http://creativecommons.org/licenses/by/3.0)], via Wikimedia Commons

What does Emergence mean?

- Emergence
  - Impossible to know all requirements in advance
  - ”Thinking harder” and ”thinking longer” can uncover some requirements, but

EVERY PROJECT HAS SOME

EMERGENT REQUIREMENTS

  - Emergent requirements are those that we cannot identify in advance
- So what do we do
  - We talk more, write less
    - But write if you have to
  - Show product increment to Stakeholders (users, customer, business, etc.)
  - Acknowledge that requirements emerge
    - And all that this implies
  - Progressively refine our understanding of the product
  - Express this progressive refinement in the Product Backlog

Processes

Defined (Predictive)

Start with:

Plan and
all requirements

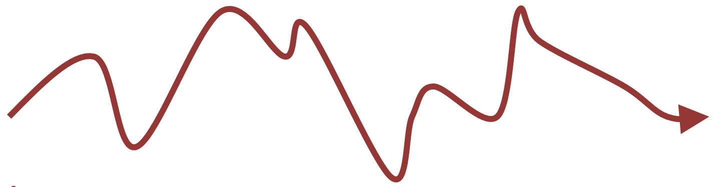
End with:

All requirements
completed

Scrum – Empirical (Adaptive)

Start with:

Goals
and
some priority
requirements



End with:

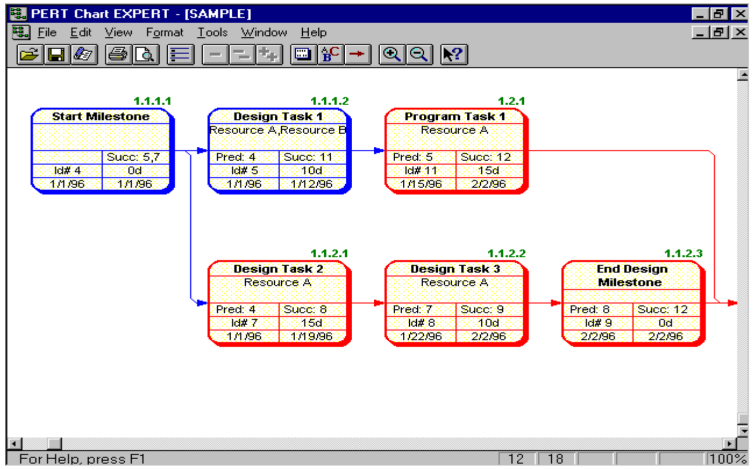
Goals met

Defined Processes

– Command and Control for simple projects

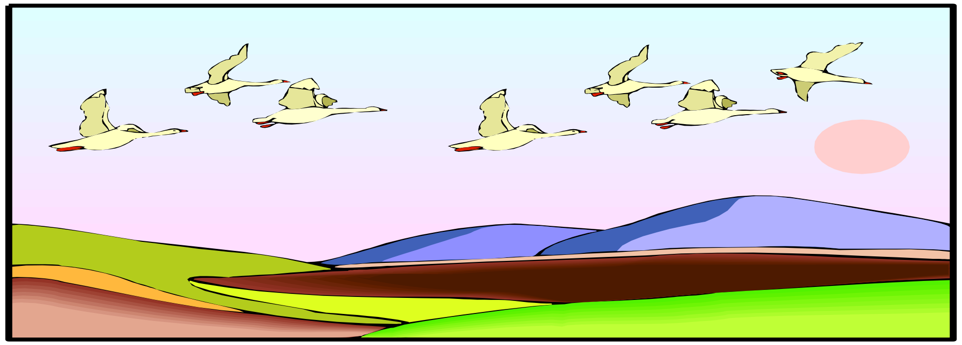
– Enforce that what happens is the same as what is planned

– Use change control to manage change



Empirical Processes

- When you can’t define things enough so that they run unattended and produce repeatable, acceptable quality output;
- Empirical models **are used when** the activities are not predictable, are non-linear, and are too complex to define in repeatable detail; and
- Control is through inspection and adaptation.



"It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood.

When the process is too complicated for the defined approach, the empirical approach is the appropriate choice."

- *Process Dynamics, Modeling*, and Control, Ogunnaike and Ray, Oxford University Press, 1992

Agile Values

Individuals and interactions over **processes and tools**

Working software over **comprehensive documentation**

Customer collaboration over **contract negotiation**

Responding to change over **following a plan**

http://www.agilemanifesto.org

12 Agile Principles

- 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

http://www.agilemanifesto.org/principles.html

Agile Practices

Agile lays out a vision and then nurtures project resources to do the best possible to achieve the plan.

Agile is the “art of the possible.”

Which of the following are common practices in Agile?

- ☑ **Frequent inspection and adaptation**
- ☐ ~~Detailed planning~~
- ☑ **Emergence of requirements, understanding, technology & team capabilities**
- ☑ **Self-organization**
- ☐ ~~Big up front design~~
- ☑ **Dealing with reality**
- ☐ ~~Dealing with artifacts of reality~~
- ☑ **Collaboration**

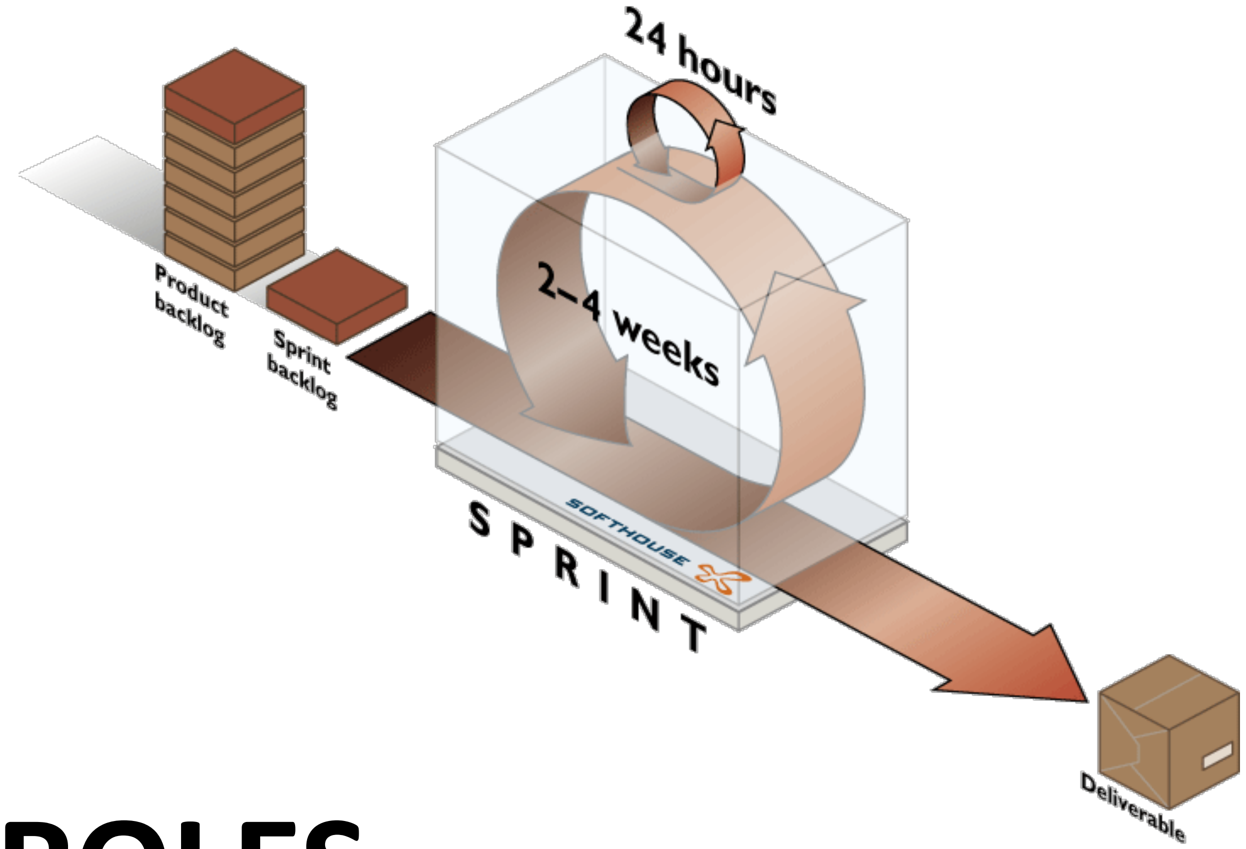
Agile vs Scrum

What is the relationship between Agile and Scrum?

Scrum is one implementation of the Agile values and principles.

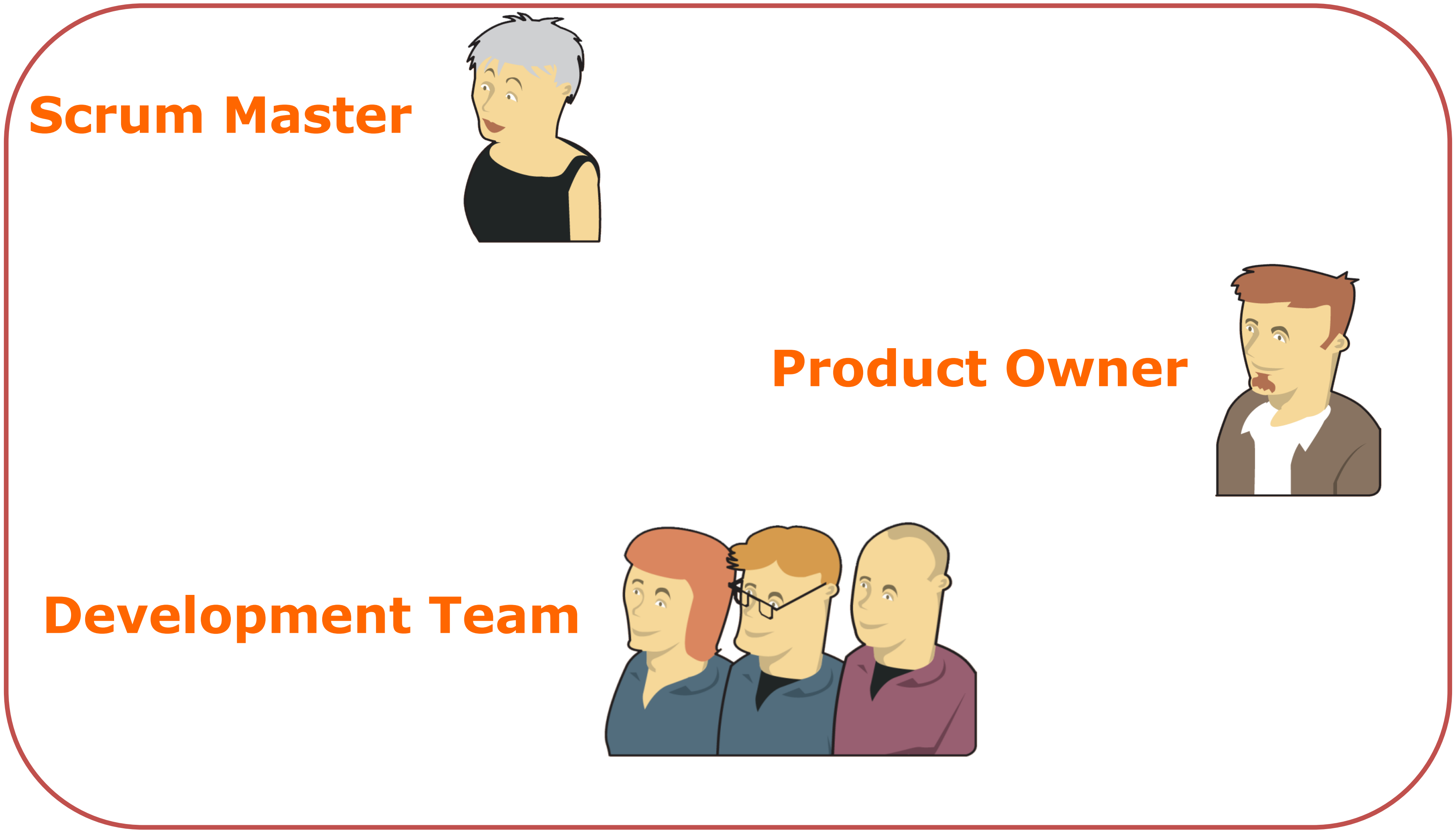
Other implementations are called Xp (eXtreme programming), DSDM (Dynamic System Development Method), FDD (Feature Driven Development), and Crystal.

They are all under the Agile “umbrella”

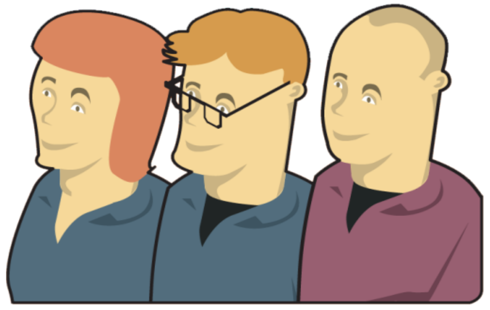

24 hours
2–4 weeks
Product backlog
Sprint backlog
SPRINT
SOFTHOUSE
Deliverable

SCRUM ROLES

Scrum Team

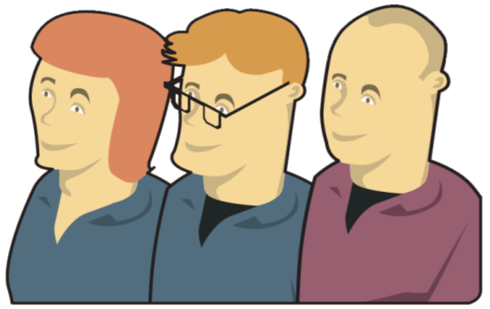


Development Team Rights



- Produce quality work
- Provide their own estimates (of product backlog items)
- Sign up for work rather than be assigned work (commit to work)
- Manage their own work (self-organized)
- Authority to do whatever is needed to meet commitment

Development Team



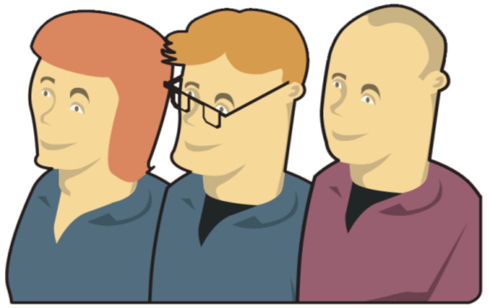
Responsibilities

- Committing to work;
- Develop Product Increment;
- Develop their ways of working;
- Authority to do whatever is needed to meet commitment;
- Manage the Sprint (Full autonomy and authority during a Sprint.);
- Identify impediments;
- Support PO;
- Estimate PBIs

Characteristics

- Self-organizing;
- Cross-functional with no roles or titles;
- 3-9 team members;
- Mutual accountability
- No subteams
- Best team available;
- Has the business and technical domain skills to build an increment of functionality;
- Not for everyone

Development Team Practices



Scrum does not tell how to implement or build a product. Instead we are encouraged to use practices that we have tried out before and to try out new practices. Within eXtreme Programming (XP) twelve engineering practices are prescribed. They are good candidates to try out for the development team in Scrum.

Possible practices to try out (you may try others as well):

- Continuous Integration
- Test Driven Development
- Collective Ownership
- Pair programming (or work)
- Coding Standard
- Simple Design

Self Organization

- **What does self organization mean?**
  - That all decisions about how, by whom and in what order are taken by the development team
  - That the decisions are taken by the development team in it's entirety and not by someone outside of the team (not by the Product Owner, not by the ScrumMaster)
- **How is it done?**
  - Each team needs to decide how they should self organize
- **When is it done?**
  - All the time
  - When it is required
  - The development team self organizes in connection with Sprint Planning and each Daily Scrum. At least.

Product Owner Rights

- Independent authority
  (decides what is the right product to build and why)

- Defining scope for the Development Team (the Product Backlog)

- Deciding what and when to release

- Order the Product Backlog

- Terminate Sprint

Product Owner

- Manages the vision, ROI, and releases
- Develops and maintains the Product Backlog
- Order and Prioritizes the Product Backlog
- Empowered to make decisions for all customers and users
- Attends meetings
- Presents and explains Product Backlog to dev team
- Available to the development team during Sprint

Product Owner in other words

- **Represent** stakeholders
- Create and present **Product Vision**
- Maximize **ROI**
- Create, maintain, order and prioritize **Product Backlog**
- Decide what and when to **release**
- **Available** to development team
- **Inspect** the results
- Decide what is **accepted**
- **Clarify** Product backlog Items
- Agree on **Definition of Done** with development team
- Define **Sprint Goal** together with development team

Product Owner Characteristics

- Understands customer needs thoroughly
- Able to create and communicate the product vision
- Empowered to make decisions,
  is decisive and know when to say no
- Has good working conditions with
  - the stakeholders
  - the team/s
- Understands value creation
- Is available to the development team

Product Owner

May

- clarify
- deliver the latest business value
- inspire the team value
- talk at Daily Scrum (if invited to do so by the development team)

Should avoid

- acting as a Project Manager
- demotivating the team
- telling what and when something needs to be done

Scrum Master
Rights

– Experiment with new ideas

– Have access to stakeholders and decision makers

– Openly address issues, problems and impediments

Scrum Master

- Removing the barriers between development and the customer so the customer directly drives development
- Teaching the customer how to maximize ROI and meet their objectives through Scrum
- Improving the lives of the development team by facilitating creativity and empowerment
- Improving the productivity of the development team in any way possible and,
- Improving the engineering practices and tools so each increment of functionality is potentially shippable.

Scrum Master in other words

- Teaches and coaches the Scrum framework
- Facilitates collaboration between PO and dev team
- Observes and reflects upon what is ongoing
- Removes impediments
- Gives feedback to roles and individuals
- Improves the implementation of Scrum

Disadvantages of combined roles

- ScrumMaster & member of Development Team
  - Not enough time to carry out both roles good
  - Different focus of the roles
  - Hard to coach and facilitate a team you are in
  - ScrumMaster responsibilities will suffer
- Dev Team member & Product Owner
  - Hard to separate what and how

Adviced against

- Product Owner & ScrumMaster
  - Conflict of interest
  - Is like a traditional Project Manager

Avoid!

Scrum means tranformation

Most projects deliver software every 6 to 18 months. Scrum reduces this to many 1 month deliveries to increase control via inspect/adapt

This puts stress on the team and organization, exposing underlying problems and limitations

The Scrum Master's job is to prioritize these problems and help the organization overcome them to get better at software development, managing software investments, and becoming a community to work in

When you don’t know what to do

- Ask the team!

- Examples:
  - I noticed <situation>. What shall we do?
  - I observed <situation>. Is that important?
  - I feel <feeling>. What is your take on it?
  - How can we find out why <situation>?
  - What do you think we should do?
  - Who has any idea about …?
  - In what way is this useful?
  - What have you decided?
  - What should I do?

Scrum and the ScrumMaster

- Scrum is a simple iterative, incremental skeleton with some rules

- Equipped with a resolute, patient ScrumMaster, Scrum can be used to transform
  - software development into a profession,
  - projects into Valuable endeavors,
  - development organizations into communities that people look forward to working in.

- It takes time
  Remember: a dead ScrumMaster is a useless ScrumMaster

- Remember that Scrum is just a framework.
  It doesn't fail.

- Sometimes people can't stand what it exposes.

- ScrumMasters are the key to its degree of success in transforming organizations.

- Teaching
  - To help someone gain knowledge
- Coaching
  - To help someone improve their knowledge or use it in better ways
- Facilitating
  - To make things easier (meetings, collaboration, communication, and decision making)

ScrumMaster as a Facilitator

The ScrumMaster facilitates by:
- making sure meetings are planned and carried out in a constructive way
- observing and asking questions
- offering techniques to support decision making

Team decision making can be facilitated by:
- Dot voting
- Fist of five
- Roman evaluation (thumb voting)

Communication can be obstructed.

Team members and Scrummaster need to look out for:
- sarcasm,
- irony,
- aggressiveness,
- defensiveness,
- misdirection,
- etc.

Teams may avoid the above by establishing ground rules to foster clear communication.

ScrumMaster as a Coach

The ScrumMaster coaches:

– the Development Team

– the Product Owner

As a coach it is beneficial for the ScrumMaster to be aware of challenges that self organized teams are facing.

Common challenges are:

- when forecasts are bad
- dealing with technical debt
- when team composition changes, e.g. someone is leaving the team

ScrumMaster as a Servant Leader

By Servant leadership we mean a leader who wants to serve first and foremost and is a leader by choice

Servant leadership is different from authoritarian, top-down management.

Servant Leader	Authoritarian Manager
1. Involving	1. Telling
2. Supporting	2. Directing
3. Morale oriented	3. Result oriented

Service to the Development Team

The Scrum Master serves the Development Team in several ways, including:

- Coaching the Development Team in self-organization and cross-functionality
- Helping the Development Team to create high-value products
- Removing impediments to the Development Team’s progress
- Facilitating Scrum events as requested or needed
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood

Service to the Product Owner

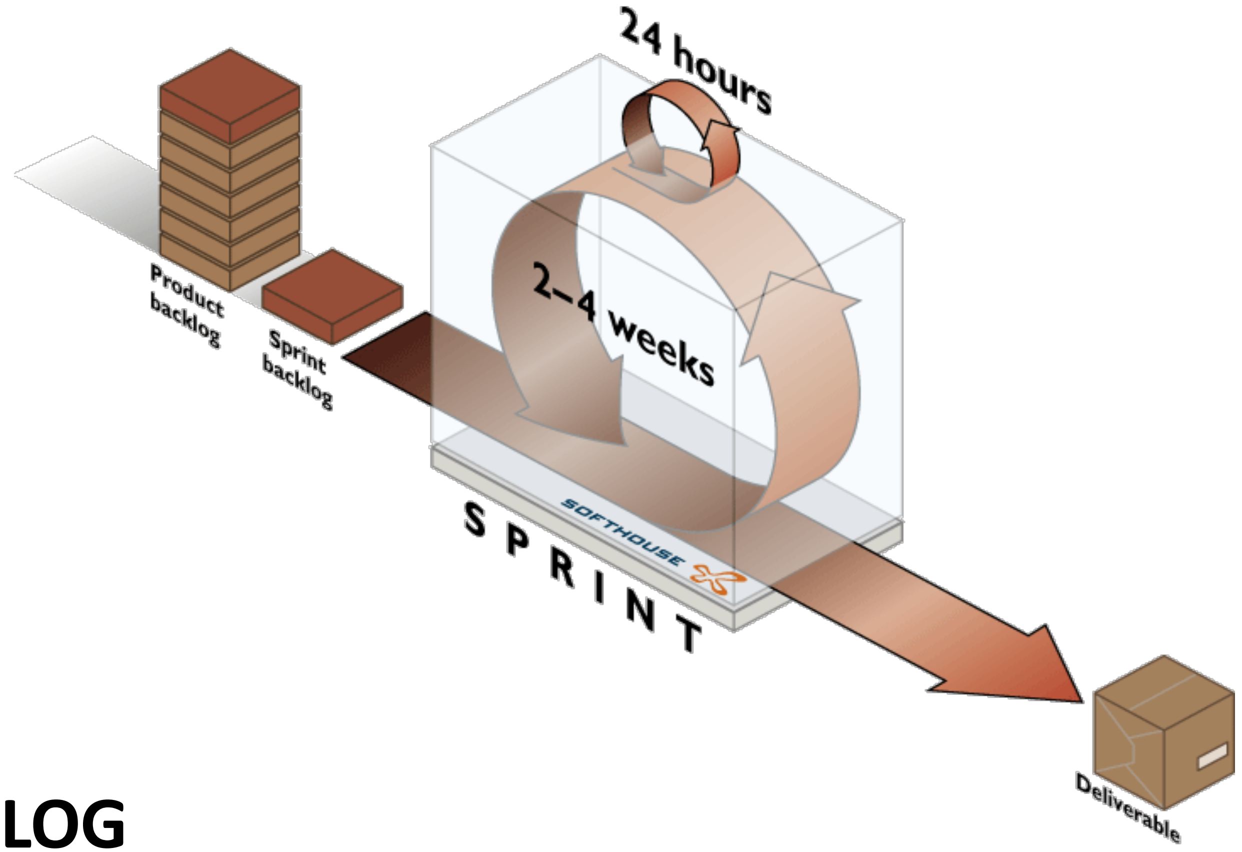
The Scrum Master serves the Product Owner in several ways, including:

- Finding techniques for effective Product Backlog management
- Helping the Scrum Team understand the need for clear and concise Product Backlog items
- Understanding product planning in an empirical environment
- Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value
- Facilitating Scrum events as requested or needed

Service to the Organization

The Scrum Master serves the organization in several ways, including:

- Leading and coaching the organization in its Scrum adoption
- Planning Scrum implementations within the organization
- Helping employees and stakeholders understand and enact Scrum and empirical product development
- Causing change that increases the productivity of the Scrum Team
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization



PRODUCT BACKLOG

Product Backlog

- Ordered list of functionality, technology, issues
- Issues are placeholders that are later defined as work
- Emergent, prioritized, estimated
- More detail on higher priority backlog
- One list for multiple teams
- PO responsible for priority
- Anyone can contribute
- Maintained and posted visibly
- Derived from Business Plan or Vision Statement, which sometimes have to be created with customer

Role Responsibilities wrt PBL

Dev Team	PO	SM
Estimate PBIs	Owns PBL	Support PO and Dev Team
Support PO	Decide value	
Participate in PBL refinement	Prioritize (order) PBIs	
	Keep PBL up to date	

Rules of Memory

Product Backlog

- **D**etailed enough
- **E**mergent or Evolving
- **E**stimated
- **P**rioritized

Product Backlog Item

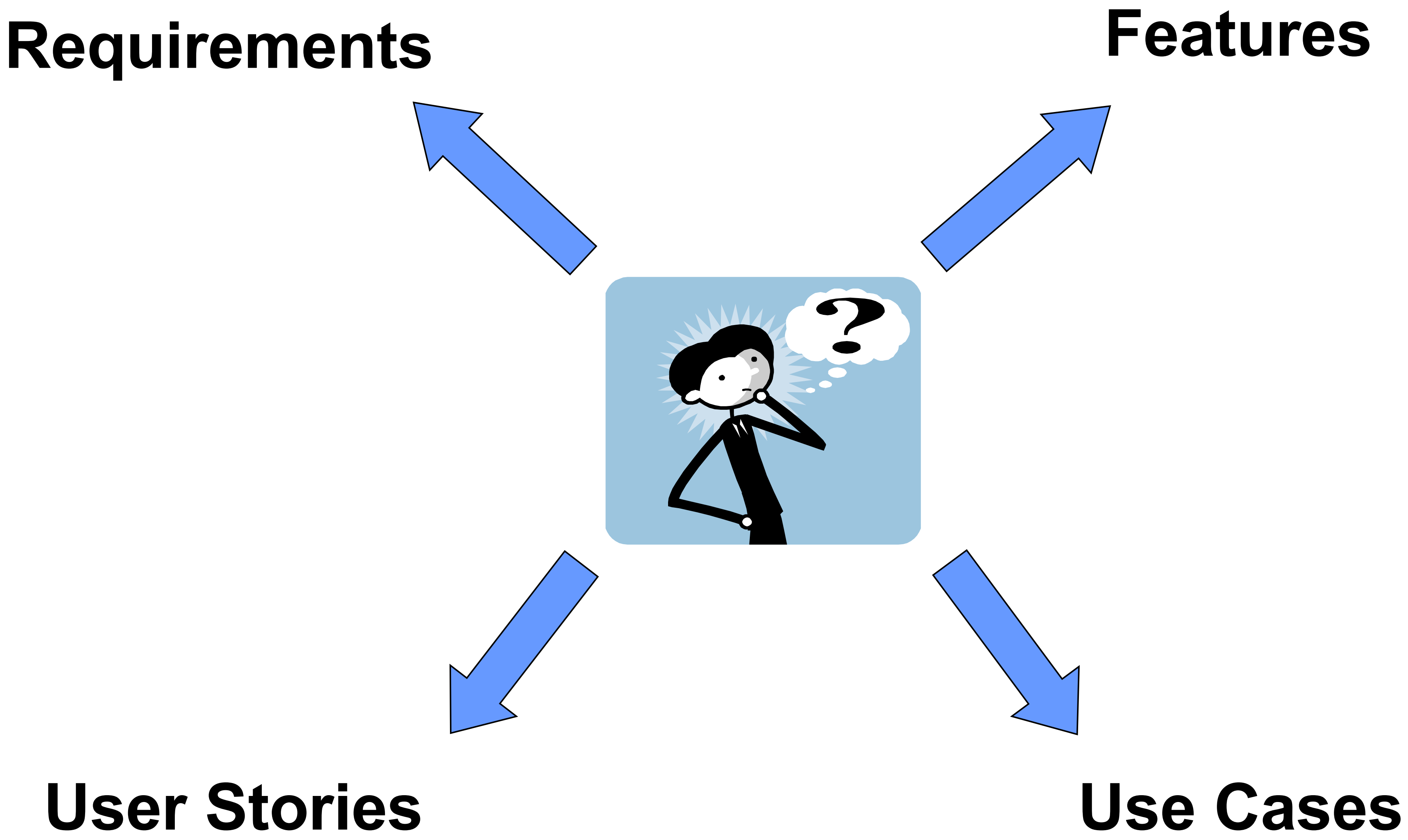
- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimatable
- **S**mall enough
- **T**estable

Attributes of a Product Backlog Item

- The items of the PBL have different attributes
- Product Owner decides which are needed
- The following are examples:
  - Description
  - Order
  - Estimate
  - Value

Product Backlog: Support Communication

- A good Product Backlog supports communication between the Team, the Product Owner and other Stakeholders
- Efficient communication happens face-to-face not in writing
- The form of the Product Backlog Items can support more or less structured communication
- The way the Product Backlog is made has a big influence on the team’s velocity



Courtesy Gertrud Bjørnvig

Architecture/Infrastructure

- Allocate time for architecture and infrastructure development
- Scrum frontloads this development onto highest priority Sprints
- First Sprint develop business functionality that keeps customer engaged and tests architecture and infrastructure
- Do a general architecture and infrastructure; implement specific code for the user stories

Product Backlog Refinement

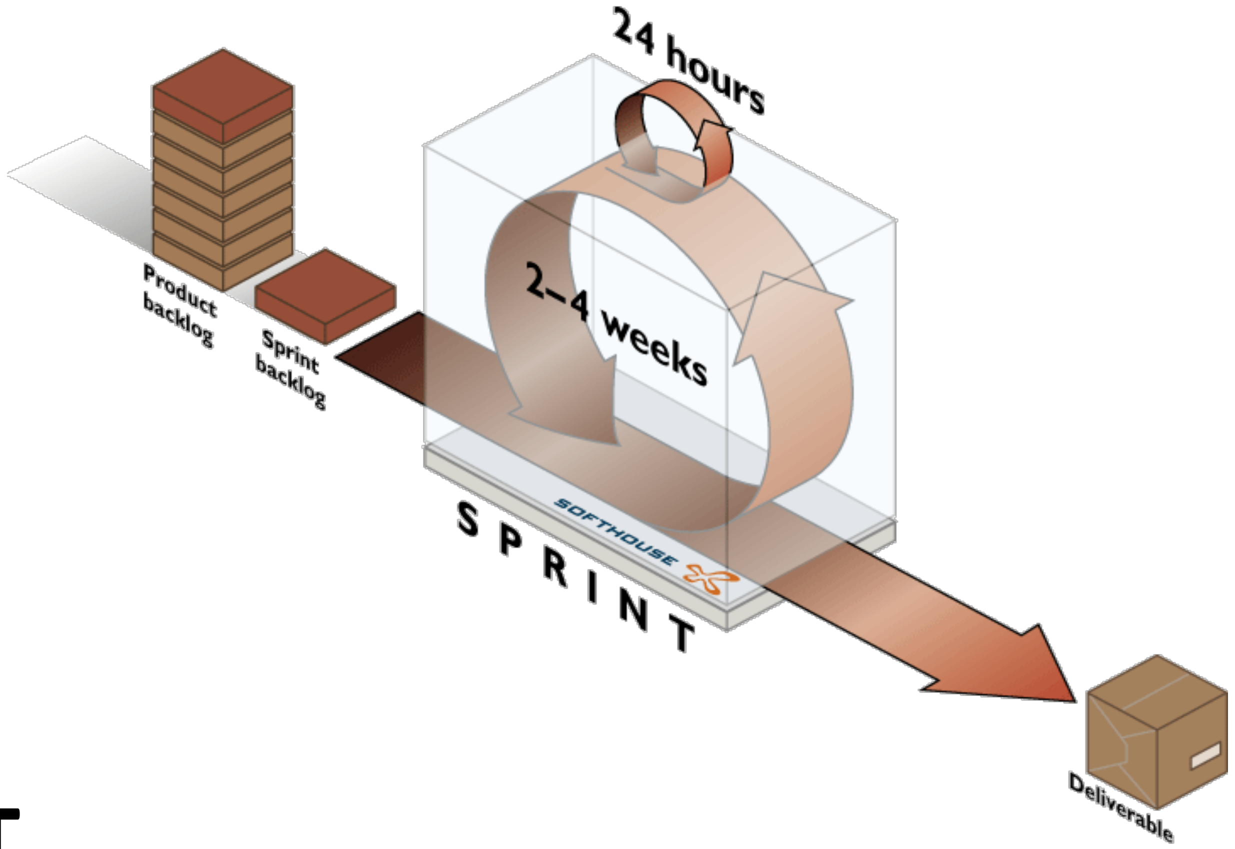
Since we start building the product before all requirements are detailed, we need to continuously refine the PBL

Product Backlog Refinement is done prior to next Sprint Planning. PO and Dev team decide how often

PO and Dev Team collaborates and participates in PBL refinement

If we don’t refine the PBL we may not be ready for next Sprint Planning and the productivity of the the Dev team risk going down

Dev team should spend maximum 10% of available time in a Sprint on refinement. If more time is spent, productivity risk going down


24 hours
2–4 weeks
Product backlog
Sprint backlog
SPRINT
SOFTHOUSE
Deliverable

THE SPRINT

The Sprint

- Team always produces:

  working product increment

- In first Sprint/s build:

  Architecture sceleton

- Sprints are guided by:

  Sprint Goal

Sprint Scope and Duration

- In Scrum both the scope and the duration of sprints are fixed.
  - Good reasons for the above are:
    - Facilitate empiricial process control so that we continuously can inspect and adapt
    - To reach some degree of predictability
    - To limit the amount of work in progress
    - To limit risk

Sprint Goal

A Sprint Goal is an objective of the Sprint.

The sprint goal should answer the question “Why are we doing this sprint?

- Handle five times as many users as version 0.8
- Prove the concept that was delivered by Architects
- Allow frequent flyers to check in online.
- Enable use of Apple Pay in Sweden.
- Make the application run on new server

Benefits of having a Sprint Goal

- Guide the development team
- Help the development team keep focus
- Allow the development team to inspect progress towards an objective

Sprint Backlog

- Characteristics
  - Created by development team
  - Owed by the development team
  - Based upon pull rather than push
- Creation
  - Created during Sprint Planning
- Changes
  - Only Development team is allowed to change it
- Follow up
  - By development team on a daily basis

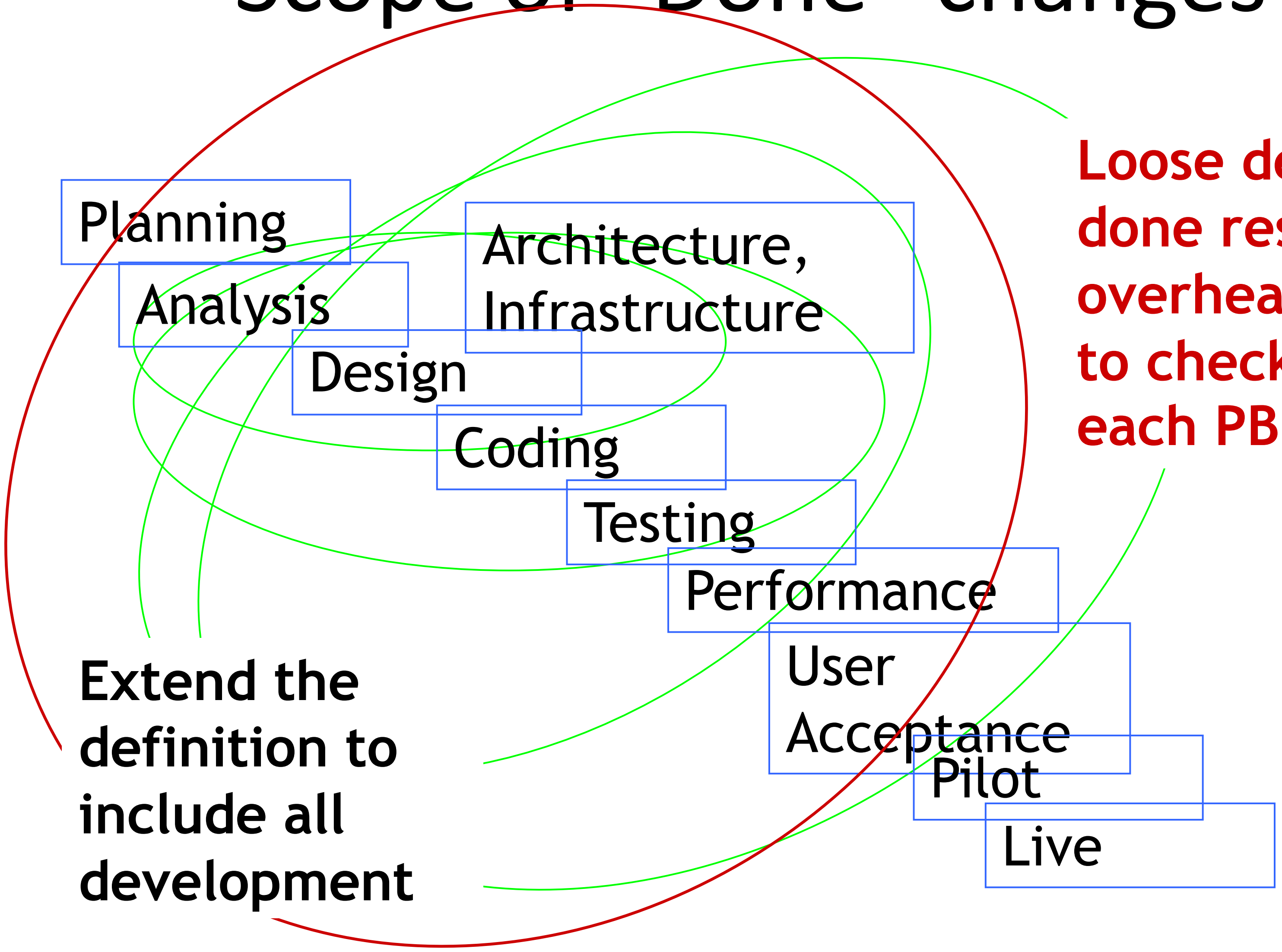
Sprint Backlog
True or False

- ❑ **Should be highly visible**
- ❑ **Should contain just enough detail**
- ❑ **Real-time snapshot of the Development Team's work for the sprint**
- ❑ **Estimated work remaining is updated daily**
- ~~❑ **Product Owner may add, delete or change the Sprint Backlog**~~
- ❑ **Development Team may add, delete or change the Sprint Backlog**

Definition of Done

- We want DoD to increase transparency
- A weak DoD may lead to misunderstanding low quality
- DoD should be extended over time as the capabilities of the development team increases
- Misunderstanding, low quality and difficulties to release, deliver or deploy may be signs of weaknesses in the DoD

Scope of ”Done” changes



Loose definition of done results in overhead needed to check state of each PBI

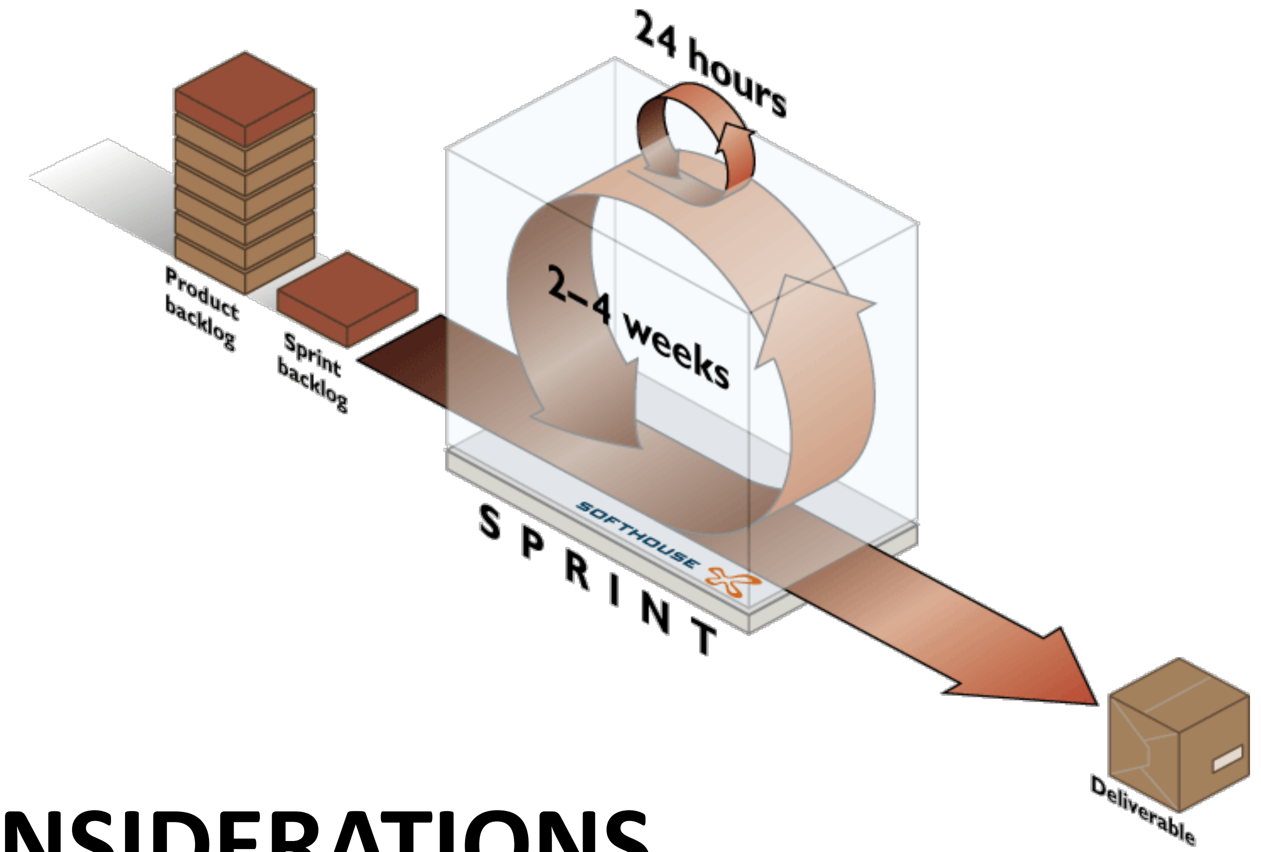
Extend the definition to include all development

”Done” in Large Projects

Done

- At a team or team of teams level, this means that common engineering infrastructure and tools are probably being employed, such as:
  - Version control
  - Build
  - Testing
  - QA Environments

- If the above is not true, we risk delays, integration difficulties, and low quality

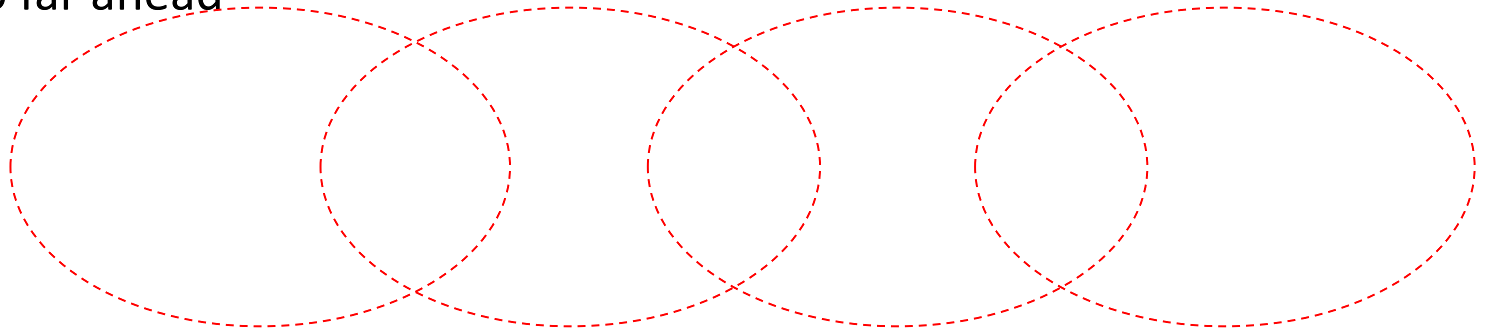


SPRINT CONSIDERATIONS

Overlapping Sprints

- Some Planning might be done for the next Sprint
- Some Design might be done for the next Sprint
- Some "Spikes" or research might be done for the next Sprint

A little is OK, as long as you collaborate and don't get too far ahead



Emergency Procedures

1. Do something different. I.e. Remove impediments
2. Get help
3. Reduce scope of the Sprint – Can only be decided by the PO
4. Terminate (cancel) Sprint – Can only be decided by the PO

Reference: https://www.scrumbook.org/product-organization-pattern-language/emergency-procedure.html

Sprint Abnormal Termination

Sprints can be cancelled before the allotted days are over

1. **Who can cancel a Sprint?**

   Product Owner

2. **What are valid reasons for cancelling a Sprint?**

   a. What the development team is working on has become obsolete

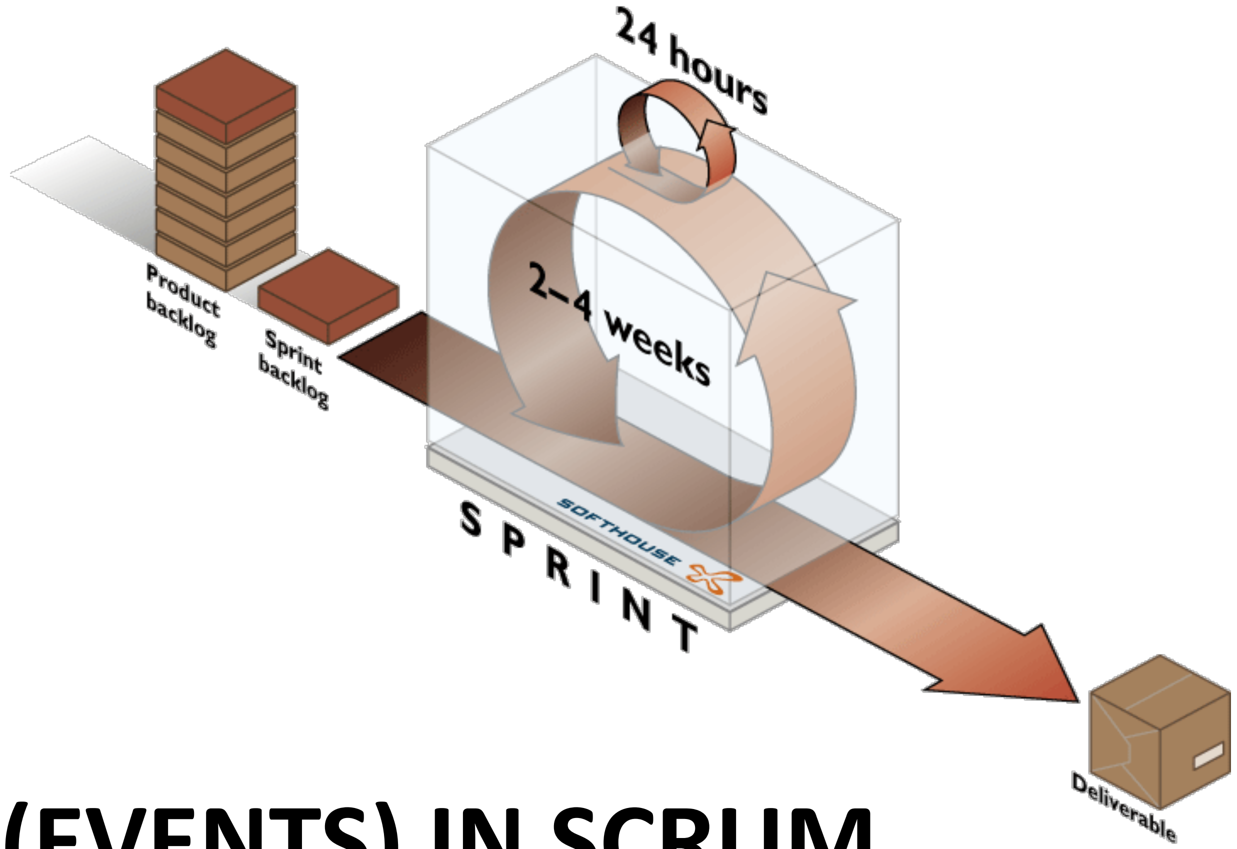
   b. The development team has identified an emergency and gone through the emergency procedures without resolving the problems.

3. **What are possible things to happen after a Sprint is cancelled?**

   a. Take care of the team members

   b. Retrospect on why this did happen and how we can avoid it happening in the future

   c. Plan a new Sprint



MEETINGS (EVENTS) IN SCRUM

Timeboxing

- Timeboxing is a time management technique where you allocate a fixed time period to a planned activity. You work on the activity during the fixed time period and stop working on it once the time is up - then, you assess whether you've reached your planned goals

- Timeboxing is used in Scrum to increase efficiency and minimize waste and risk. In Scrum, timeboxing is a critical component of all five events. Some Scrum teams also use timeboxing during a Sprint to concretely define open-ended tasks. An example of an open-ended task might be conducting research that is necessary for the team to reach a decision or to estimate the size and complexity of an upcoming piece of work.

- Possible benefits of timeboxing are:
  - Forces prioritization
  - Establishes a WIP limit
  - Demonstrates progress
  - Avoids unnecessary perfection

References:
https://www.scruminc.com/what-is-timeboxing/
https://innolution.com/blog/the-benefits-of-timeboxing

Sprint Planning Meeting

Input

- *Product Backlog and Product Vision*
- *Availability of Development Team members*
- *Result from last Sprint Review*
- *Result from last Retrospective*
- *DoD*

Output

- *Sprint Goal*
- *Sprint Backlog*

Purpose

To plan the Sprint and to identify a Sprint Goal

Participants

- *Development Team*
- *Product Owner*
- *ScrumMaster*

Timebox

Maximum 8 hours for
a one-month Sprint.
Shorter for shorter Sprints.

Sprint Planning Meeting Rules

- Sprint Planning is divided into two parts. Topic 1 and Topic 2
- Topic One:
  What can be done this Sprint?
- Topic Two:
  How will the chosen work get done?

- Possible Principles for Sprint Planning
  - Commitment based sprint planning
  - Velocity based sprint planning

Sprint Planning Discussions

- Role responsibilities for Sprint Planning

Product Owner:

  - Present PBL. Describe what

Development team:

  - Decide how much and how to build the product increment

ScrumMaster:

  - Facilitate the event

- Which negative aspects can occur if all elements of the Sprint Planning are not carried out?
  - The Scrum team my miss risks or bring in too much into the Sprint

- How can the Scrum Team use Sprint Planning to Inspect & Adapt?
  - To fully understand all aspects of the product being developed
  - To make sure that the Product vision is being met and is still relevant
  - Too minimize wasted work

- Transparency at Sprint Planning
  - Product Vision
  - Current state and progress

- What are the biggest challenges for Sprint Planning?
  - Bringing too much work into the Sprint
  - Not fully understanding what is brought into the Sprint

Daily Scrum Meeting

Input

- *Every team members knowledge of what happened since last Daily Scrum*

Output

- *Updated Sprint Backlog*
- *Updated Sprint plan to achieve Sprint Goal.*

Purpose

Inspect progress towards the Sprint Goal, and identify impediments

Participants

- *Development Team*
- *ScrumMaster*
- *(Product Owner)*

Timebox

15 minutes

Daily Scrum

- Guiding questions:
  - What did I do yesterday that helped the Development Team meet the Sprint Goal?
  - What will I do today to help the Development Team meet the Sprint Goal?
  - Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?

- The guiding questions are linked to the Sprint Goal in order to enable inspection and adaption towards the Sprint Goal.

- Daily Scrum is different from a status meeting because it is
  - Forward looking
  - Shorter
  - Not about reporting to an external party

- The constraints (15 minutes, every day, only dev team) support the Scrum Team by allowing shorter time, better focus and to avoid wasting more than one day’s work

Daily Scrum Discussions

- Role responsibilities for Daily Scrum

Development Team:
  - Active
  - Inspecting and adapting towards Sprint Goal
  - Self organizing

ScrumMaster:
  - Present
  - Facilitating if asked or necessary

Product Owner:
  - Optional

- If all elements of the Daily Scrum are not carried out, the Development Team risk wasting time and making mistakes. Worst case: missing the Sprint Goal

- The Scrum Team can use the Daily Scrum to Inspect & Adapt towards the Sprint Goal

- We want full transparency at Daily Scrum. Both successes and impediments. This makes it is possible for the Scrum team to act upon impediments

- What are the biggest challenges for Daily Scrum?
  - Keeping the meeting short
  - Remembering the purpose
  - Avoiding solving impediments

Sprint Review Meeting

Input

- Product Increment
- Sprint Goal
- Sprint Backlog
- Definition of Done

Output

- Revised Product Backlog

Purpose

Inspect the Increment and adapt the Product Backlog if needed

Participants

- Development Team
- ScrumMaster
- Product Owner
- Stakeholders (invited by the PO)

Timebox

Maximum 4 hours for a one-month Sprint. Shorter for shorter Sprints

Sprint Review Meeting Rules

- In addition to reviewing the Product Increment, participants may also review timeline, budget, potential capabilities, and marketplace for the next anticipated releases of functionality or capability of the product
- All work done by the Development Team during a Sprint should be accepted before the Sprint Review. Only accepted product increment should be reviewed
- Possible outcomes of Sprint Review are new ideas and better understanding of what should remain on the Product Backlog

Reasons why the team have to present finished functionality

- To be able to inspect reality
- To get focused conversations
- To help everyone participating getting the same picture
- To be able to make release decisions

Sprint Review Discussions

- Role responsibilities for Sprint Review

Development Team:
  - Present the Product Increment
  - Receive feedback
  - Participate in discussions

ScrumMaster:
  - Facilitate

Product Owner:
  - Give feedback
  - Lead discussions

Stakeholders:
  - Give feedback
  - Participate in discussions

- If all elements of the Sprint Review are not carried out we risk missing feedback and in the long run, risk developing the wrong product

- The Scrum Team can use Sprint Review to Inspect & Adapt towards the Product vision

- Transparency at Sprint Review is of importance so that we are able to have the right conversations

- What are the biggest challenges for Sprint Review?
  - Not getting enough feedback
  - Not having the right Stakeholders present
  - Avoiding putting blame on Development Team

Sprint Retrospective Meeting

Input

- All knowledge and available data about what happened during the last Sprint

Output

- Actions for improvements

Purpose

Continuous improvement
Inspect our ways of working and create plan for what to improve next Sprint

Participants

- Development Team
- ScrumMaster
- Product Owner

I.e. The entire Scrum Team

Timebox

Maximum 3 hours for a one-month Sprint.
Shorter for shorter Sprints

Retrospective Discussions

- Role responsibilities for Retrospective

Development team:
  - Actively participating

Product Owner:
  - Actively participating

ScrumMaster:
  - Facilitating and participating

- If all elements of the Retrospective are not carried out the Scrum Team risk missing opportunities to improve their way of working

- The Scrum Team can use Retrospective to continuously Inspect & Adapt their ways of working and to solve long term impdiments

- Transparency at Retrospective will help the Scrum Team to identify good ways of working as well as opportunities for improvement

- What are the biggest challenges for Retrospective?
  - To create a safe environment for discussions
  - To have everyone participating on the same level
  - To identify all the good things we are able to do

Driving questions?

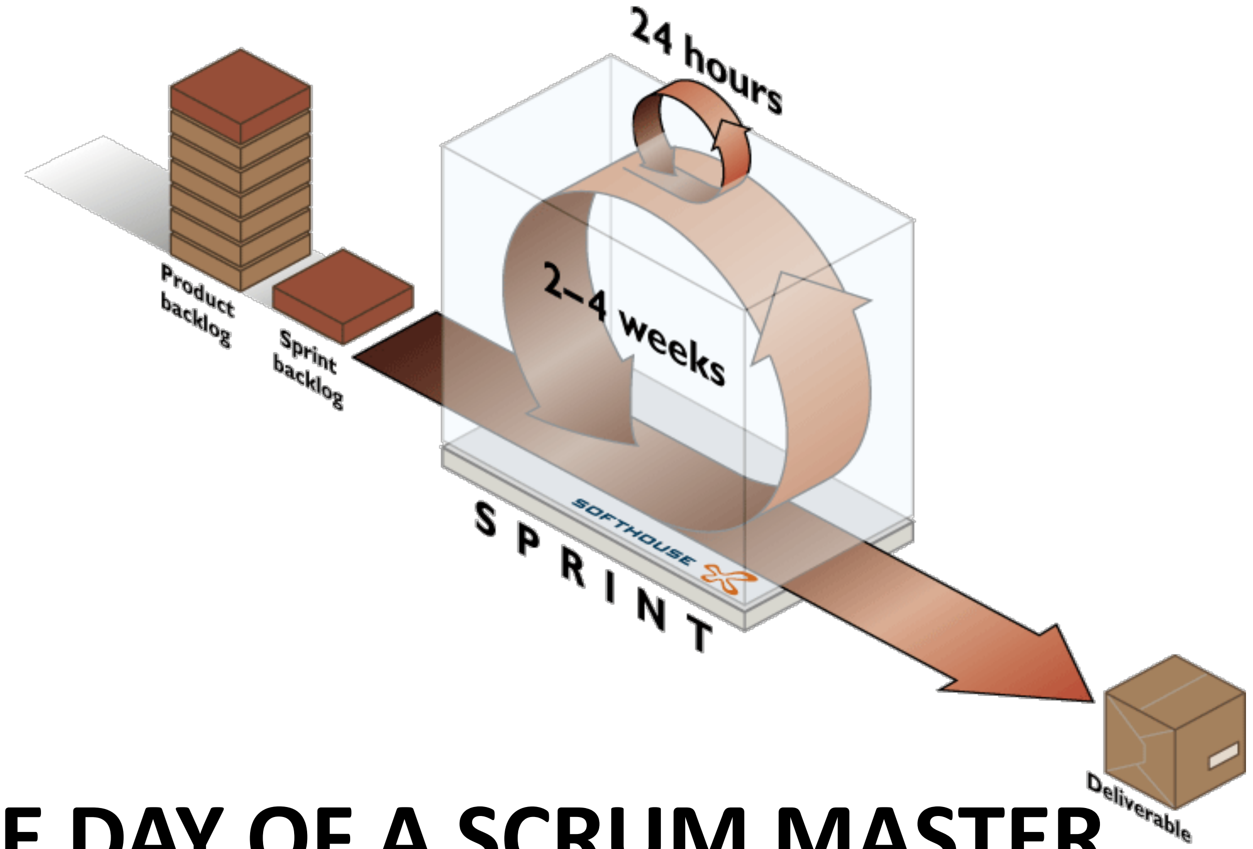
In Retrospectives it is good to ask questions. Four questions (not the only ones) that can be used to drive a Retrospective are:

1. What did we do good?
2. What can we change to become better?
3. What did we learn?
4. What is still puzzling us?

Structure

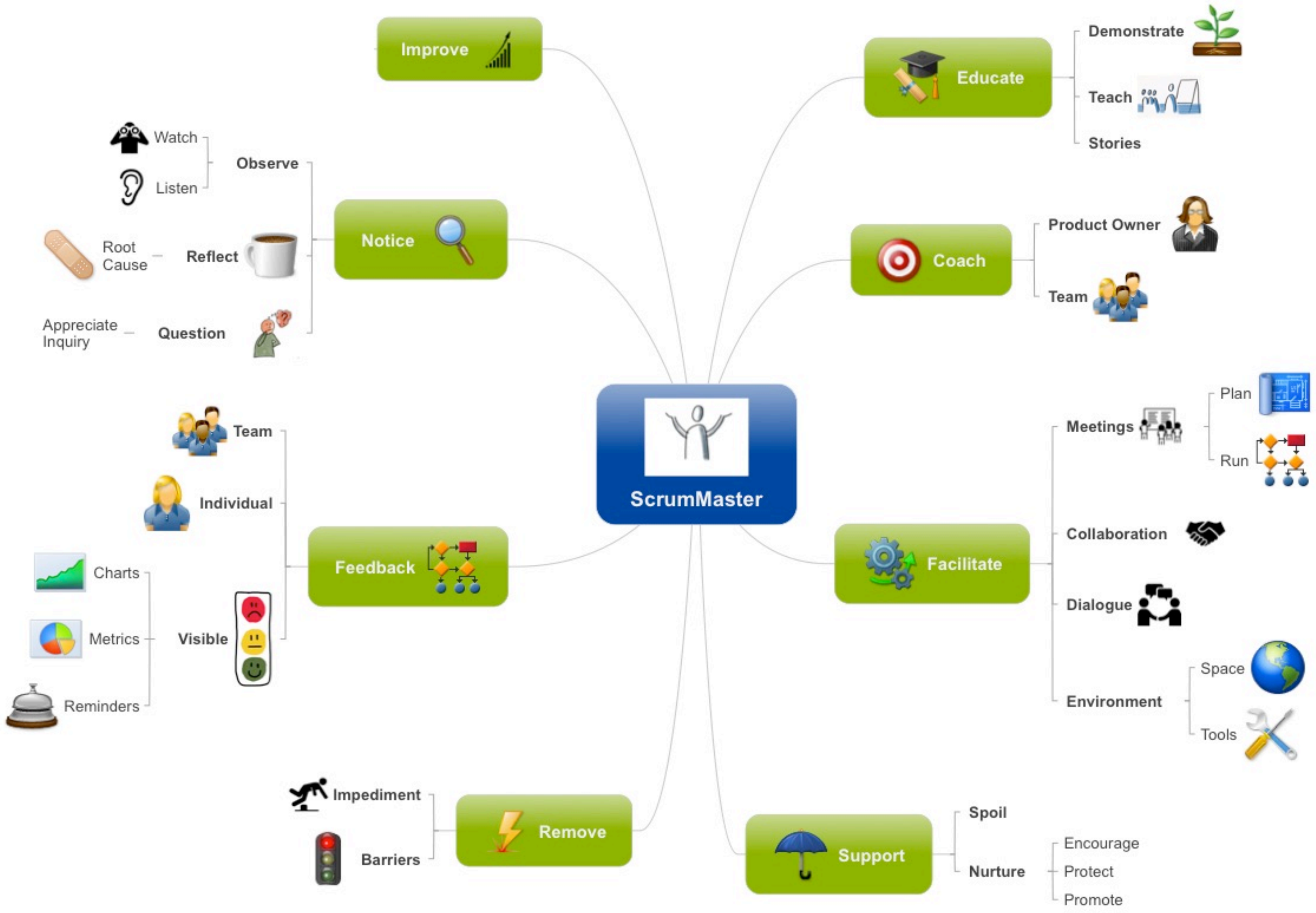
A possible structure for a Retrospective is outlined in
”*Agile Retrospectives*” by Esther Derby and Diana Larsen

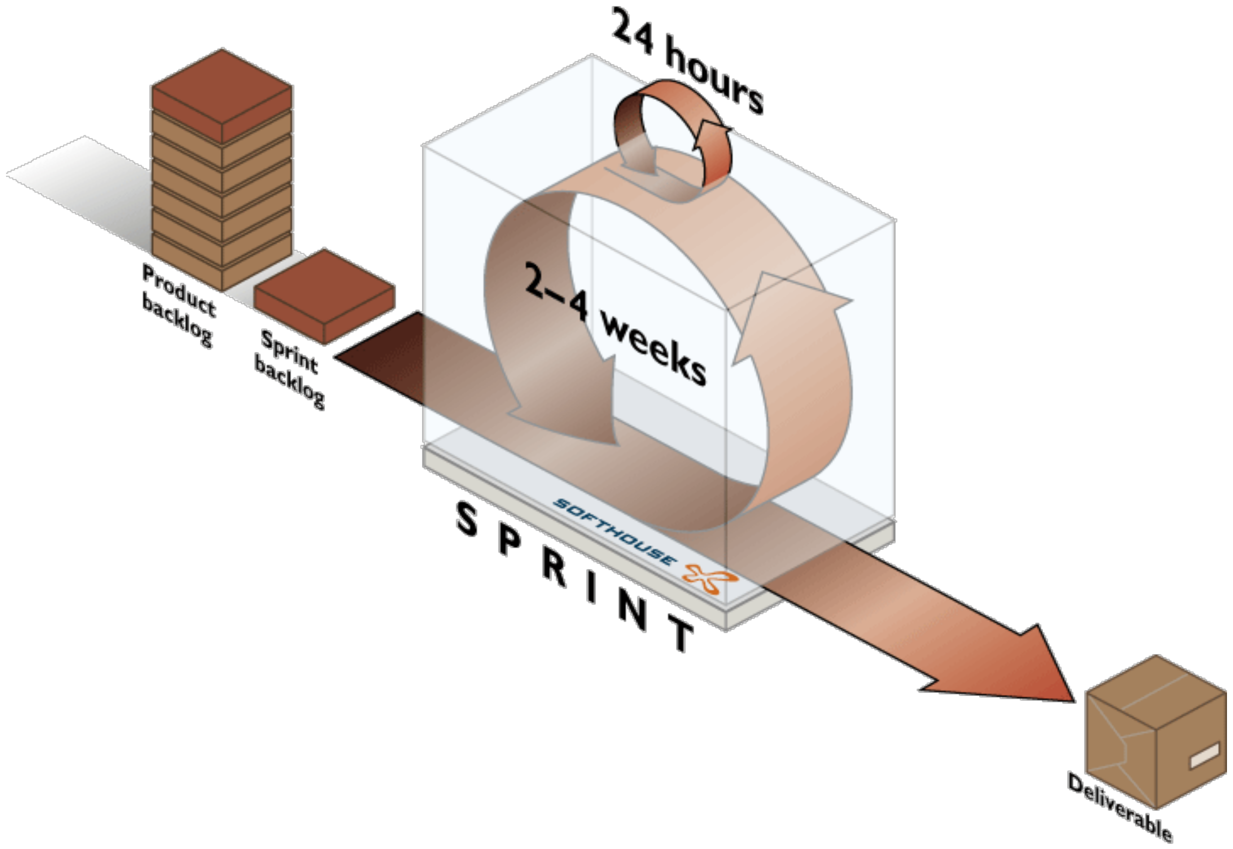
1. Set the stage
2. Gather data
3. Gain insight
4. Decide
5. Close retrospective



AN EXTREME DAY OF A SCRUM MASTER

Day in Life of ScrumMaster




24 hours
2–4 weeks
Product backlog
Sprint backlog
SPRINT
SOFTHOUSE
Deliverable

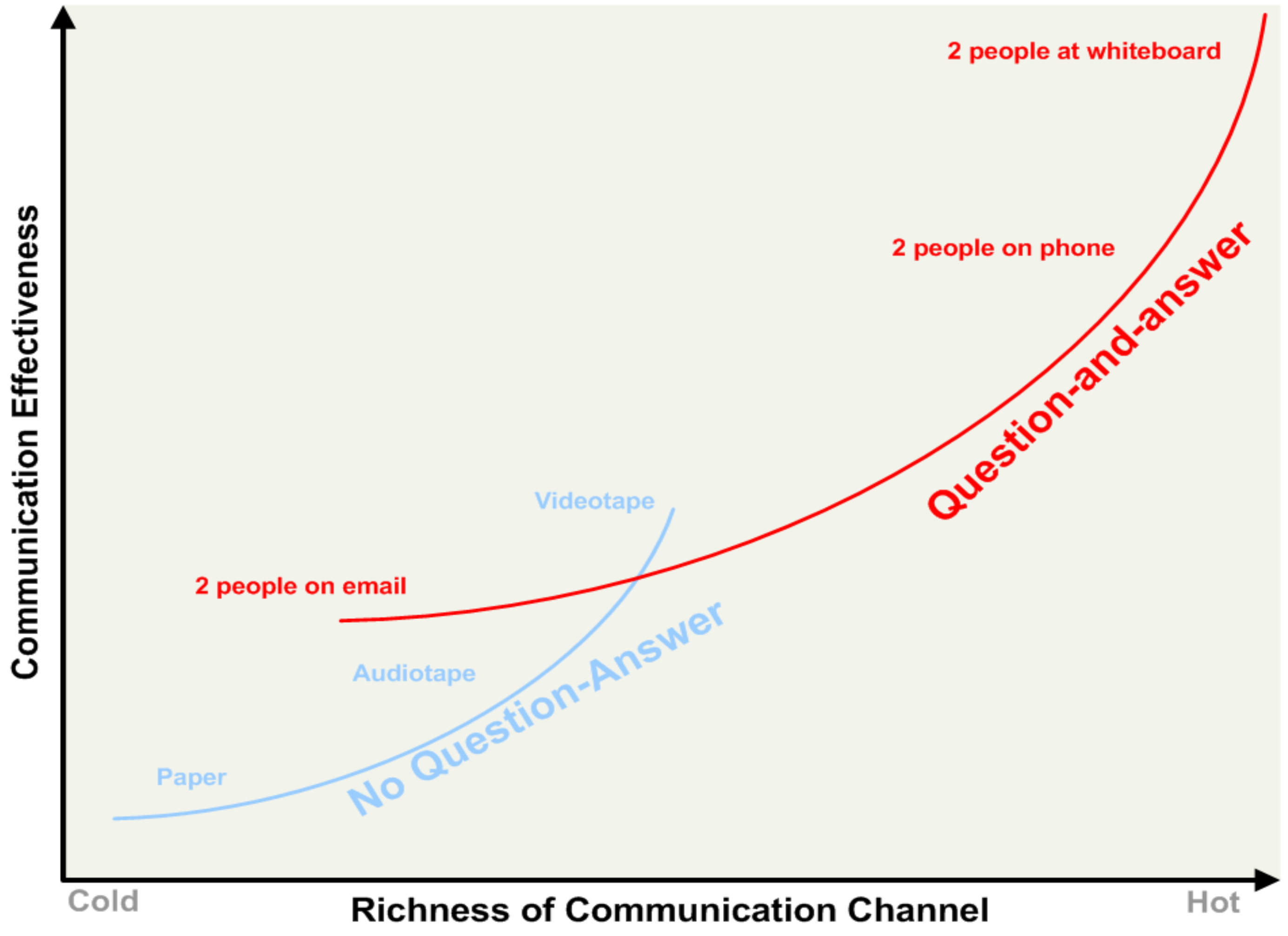
SUMMARY

Don’t forget

- Scrum works
  - But **not alone**. We need to add practices that fit our needs
- Scrum is simple
  - But **hard** to implement
- Scrum can be painful
  - It **exposes problems** and **limitations**
- Scrum is different
  - Be prepared for a **change of mindset**

PART 2
GOOD TO KNOW

Communication

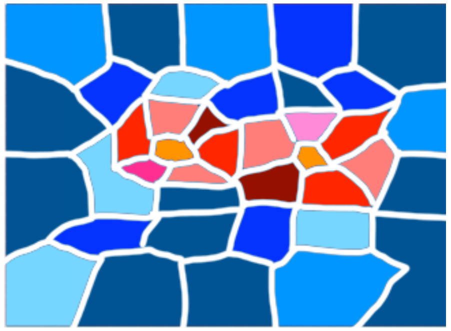


Alistair Cockburn

Architecture/Infrastructure

- Allocate time for architecture and infrastructure development
- Scrum frontloads this development onto highest priority Sprints
- First Sprint develop business functionality that keeps customer engaged and tests architecture and infrastructure
- Do a general architecture and infrastructure; implement specific code for the user stories

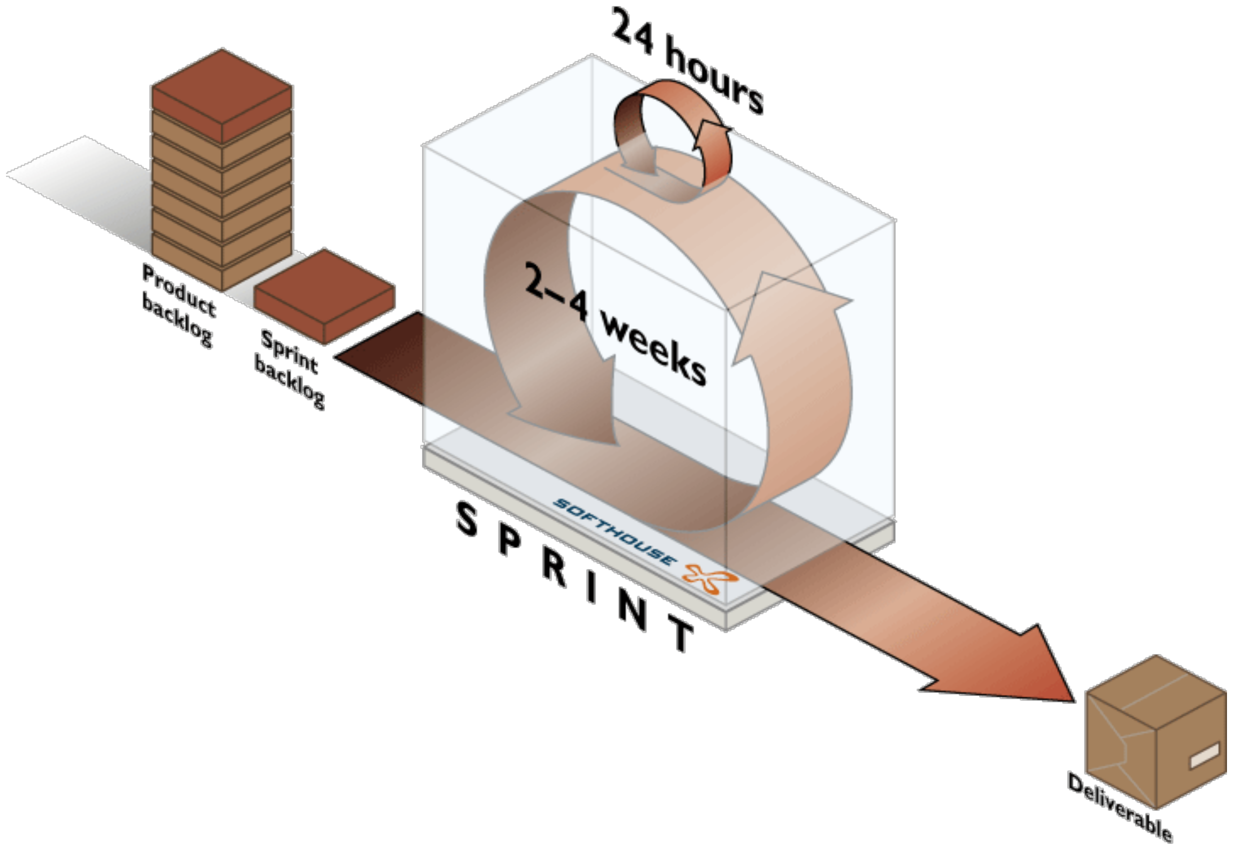
Scrum Patterns



Patterns for good implementations of Scrum are collected and presented at https://www.scrumbook.org/

Possible points of interest:

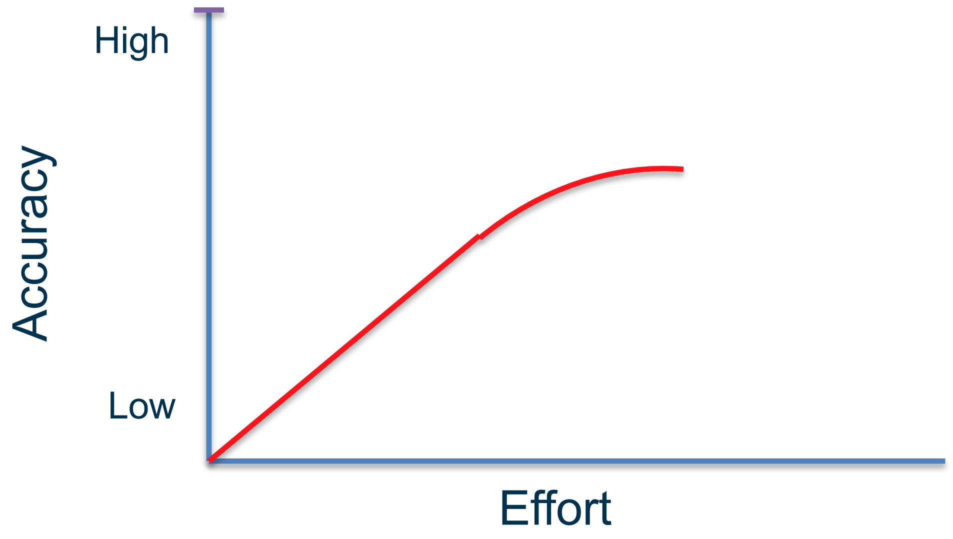
- The Spirit of the Game
- The core of Scrum presented as patterns (solutions to organizational, process, and business problems)


24 hours
2–4 weeks
Product backlog
Sprint backlog
SPRINT
SOFTHOUSE
Deliverable

ESTIMATING

Good Enough

It's better to be roughly right than precisely wrong. John Maynard Keynes 1883-1946



How does the amount of effort we put into estimation affect the accuracy?

Initially it may be a linear relationship. Not for long though.

Size matters

Establish a relative measure of size.

Obtain an adequate estimation.

But when can you deliver?

100 blocks	10 blocks per trip	10 trips
Size	Velocity	Duration

Duration is derived.

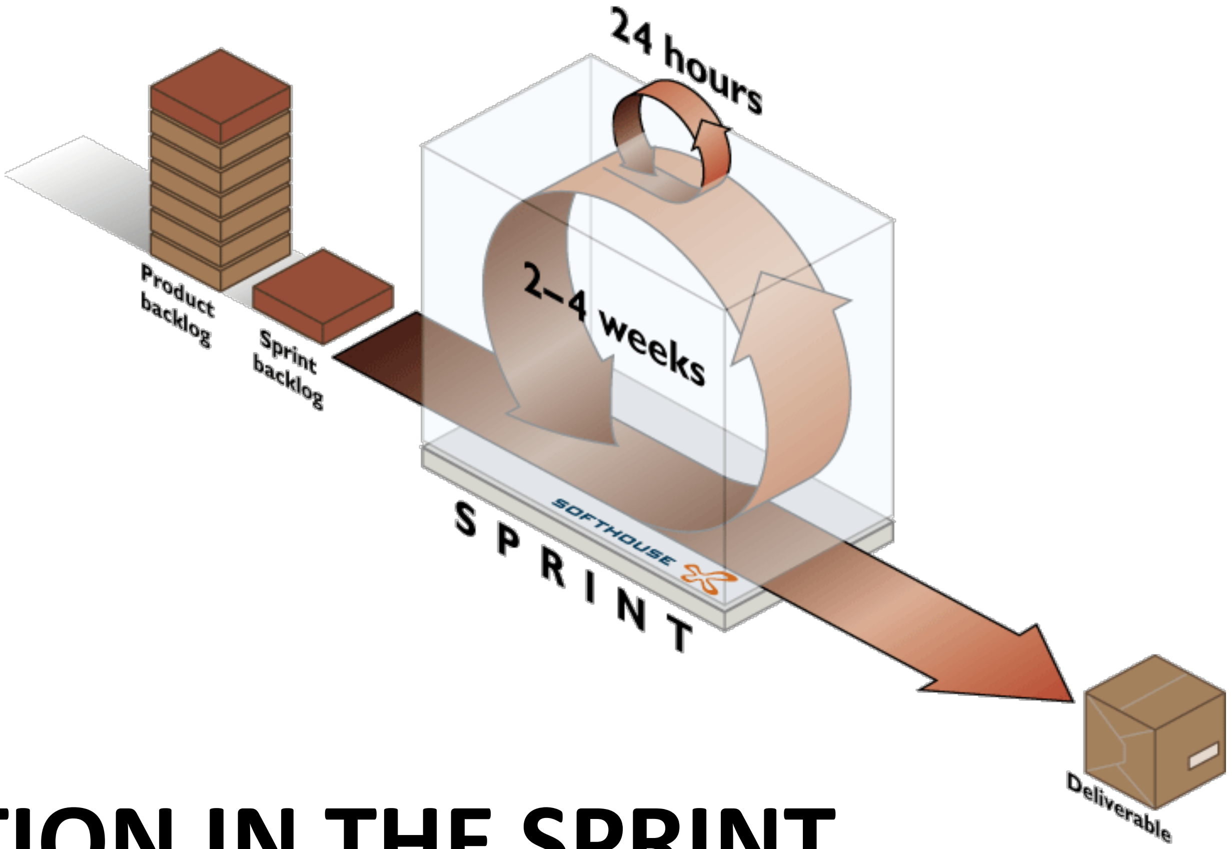
Points

- The ”size” of an item
- Influenced by
  - Effort
  - Team knowledge
- Points are unit-less and relative
  - A 5-point story is half the size of a 10-point
- Scale

As a user, I want to be able to have some but not all items in my cart gift wrapped.

5

0, 1, 2, 3, 5, 8, 13, 20, 40, 100



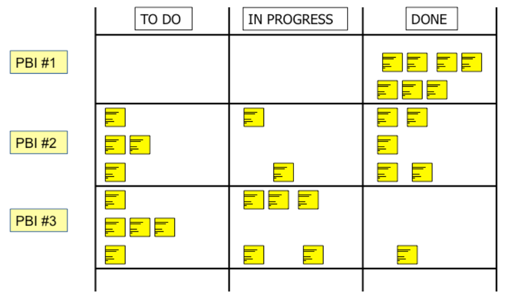
VISUALIZATION IN THE SPRINT

Information Radiators*

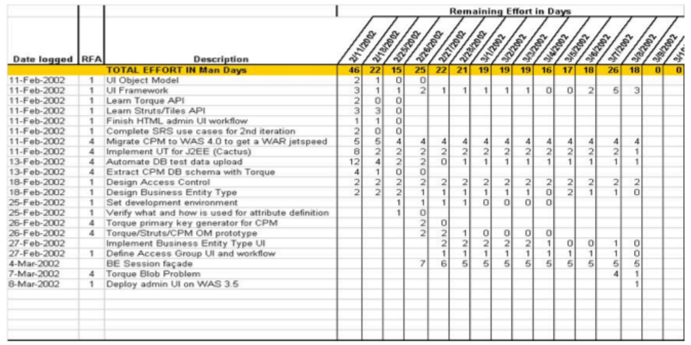
- Displays important information in a place where passersby can see it
- Increases communication to those outside the group without interference from outside
- Provides anchor points for the team

* Cockburn, Agile Software Development

Task board

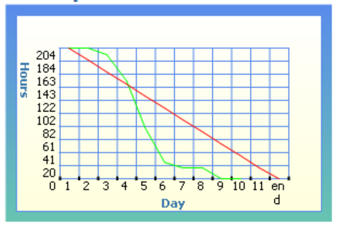


Sprint Backlog

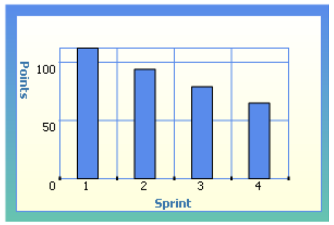


Date logged	RFA	Description	Remaining Effort in Days															
			2/11/2002	2/18/2002	2/25/2002	2/26/2002	2/27/2002	2/28/2002	3/1/2002	3/2/2002	3/3/2002	3/4/2002	3/5/2002	3/6/2002	3/7/2002	3/8/2002	3/9/2002	3/1
		TOTAL EFFORT IN Man Days	**46**	**22**	**15**	**25**	**22**	**21**	**19**	**19**	**19**	**16**	**17**	**18**	**26**	**18**	**0**	**0**
11-Feb-2002	1	UI Object Model	2	1	0	0												
11-Feb-2002	1	UI Framework	3	1	1	2	1	1	1	1	1	0	0	2	5	3		
11-Feb-2002	1	Learn Torque API	2	0	0													
11-Feb-2002	1	Learn Struts/Tiles API	3	3	0													
11-Feb-2002	1	Finish HTML admin UI workflow	1	1	0													
11-Feb-2002	1	Complete SRS use cases for 2nd iteration	2	0	0													
11-Feb-2002	4	Migrate CPM to WAS 4.0 to get a WAR jetspeed	5	5	4	4	4	4	4	4	4	4	4	4	4	4		
11-Feb-2002	4	Implement UT for J2EE (Cactus)	8	2	2	2	2	2	2	2	2	2	2	2	2	1		
13-Feb-2002	4	Automate DB test data upload	12	4	2	2	0	1	1	1	1	1	1	1	1	1		
13-Feb-2002	4	Extract CPM DB schema with Torque	4	1	0	0												
18-Feb-2002	1	Design Access Control	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
18-Feb-2002	1	Design Business Entity Type	2	2	2	1	1	1	1	1	1	0	2	1	1	0		
25-Feb-2002	1	Set development environment			1	1	1	1	0	0	0	0						
25-Feb-2002	1	Verify what and how is used for attribute definition			1	0												
26-Feb-2002	4	Torque primary key generator for CPM				2	0											
26-Feb-2002	4	Torque/Struts/CPM OM prototype				2	2	1	0	0	0	0						
27-Feb-2002		Implement Business Entity Type UI					2	2	2	2	2	1	0	0	1	0		
27-Feb-2002	1	Define Access Group UI and workflow					1	1	1	1	1	1	1	1	1	0		
4-Mar-2002		BE Session façade				7	6	5	5	5	5	5	5	5	5	5		
7-Mar-2002	4	Torque Blob Problem													4	1		
8-Mar-2002	1	Deploy admin UI on WAS 3.5														1		

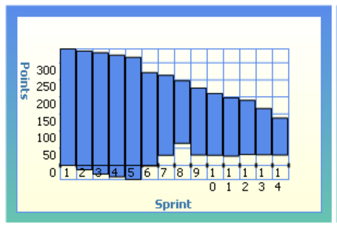
Sprint Burndown Chart



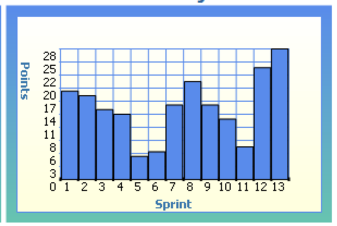
Release Burndown Chart



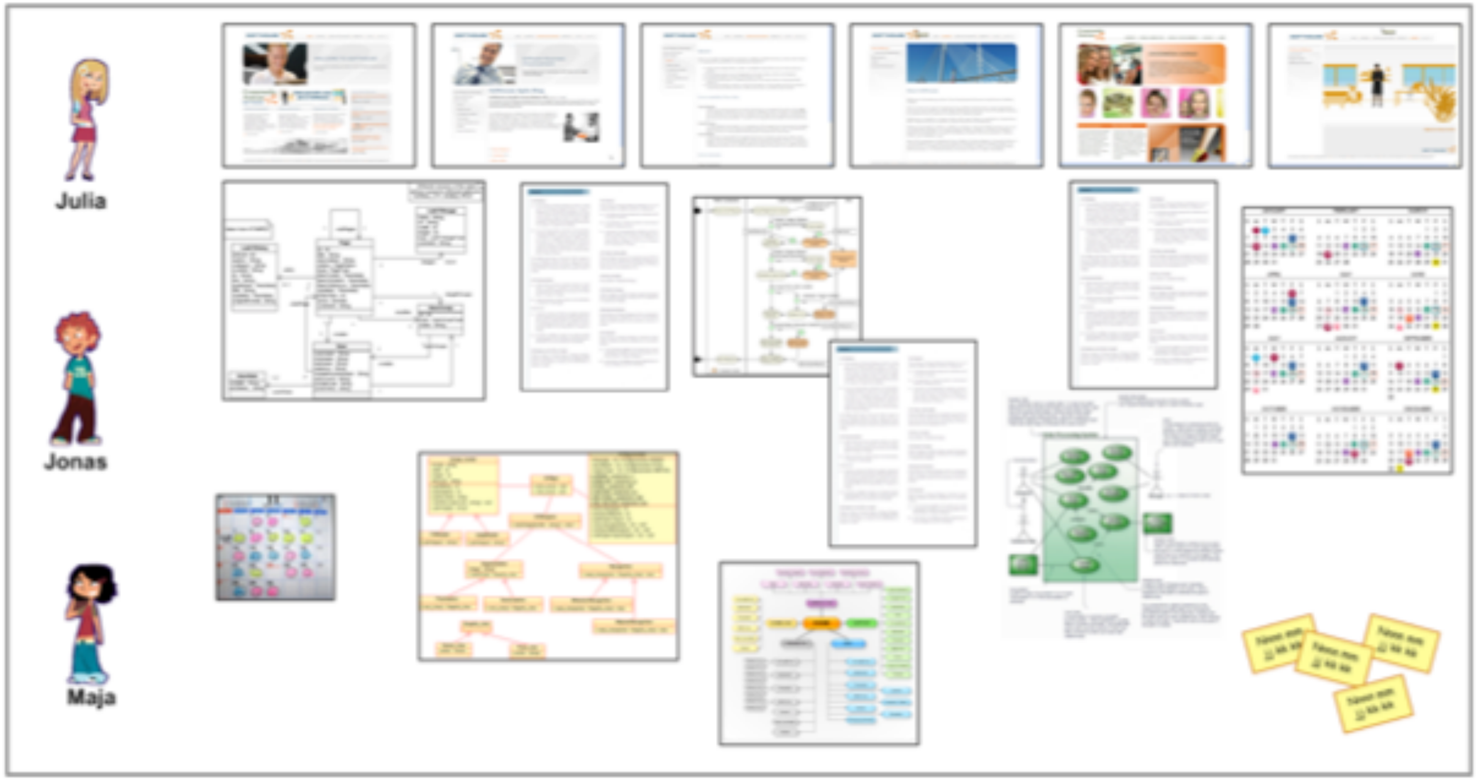
Product Burndown Chart



Velocity



Mean velocity: 15
Lowest velocity: 5
Highest velocity: 28



Scrum History

- 1990
  - Sutherland leads Guiness Peat Aviation IT innovations in Ireland. Project Leader/Gantt Chart problems identified.
    This lead to abandonment of Gantt Charts in Scrum.
    Also turns traditional project leader into team member and facilitator of self-organizing team.

- 1989-1993
  - Sutherland runs Object Databases building Matisse Object Database. Lots of pair programming, refactoring, AI, concurrent engineering
  - Sutherland leases space to iRobot startup. Learns subsumption architecture as basis for self-organizing systems. This feeds into Scrum.
  - Works extensively with Capers Jones, leading productivity author and researcher. For next seven years, Capers company benchmarks Scrum teams.
  - Sutherland on Advisory Board of Accion, largest microenterprise development non-profit in western hemisphere. Lots of small team research.
  - Small team formation concepts find their way into what becomes Scrum allowing teams to bootstrap economic performance.

Scrum History

- 1993
  - Sutherland goes to Easel Corporation as VP Object Technology to lead Object Studio Smalltalk project.
    Computer science and productivity research continues on how best to create Scrum.
    Takeuchi and Nonaka paper and Jim Coplien's review of Borland Quattro Pro project were key resources.
    Sutherland hires Jeff McKenna as consultant and John Scumniotales as first ScrumMaster. Together they form first Scrum team.

- 1994
  - Easel Object Studio Release 1 (6 Sprints) and Object Studio Release 2 (6 Sprints) using Scrum.

- 1995
  - Sutherland provides information on Scrum to Kent Beck to help him create XP

- 1995
  - Easel acquired by VMARK. Jeff invites Ken Schwaber to visit VMARK Object Studio Scrum team.
    Ken agrees to promote Scrum in the software industry as open source and publishes first Scrum paper at OOPSLA'95.

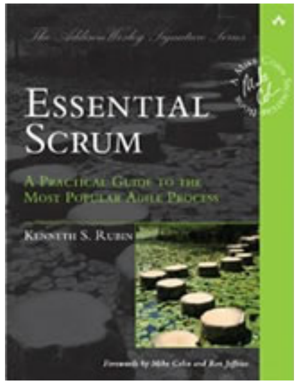
Scrum History

- 1996
  - Sutherland goes to Individual as VP of Engineering. Ken, as consultant, work together to bring up Scrum
  - Sutherland goes to IDX as SVP of Engineering/CTO. Ken helps bring up Scrum in many divisions. Scalability of Scrum demonstrated.
- 1999
  - Mike Beedle leads team effort to publish Scrum Pattern in Pattern Languages of Programming Design, Volume 4.
- 2000
  - Sutherland goes to PatientKeeper as CTO. Ken works with him to bring up Scrum at PatientKeeper. Sutherland evolves Type C Scrum.
- 2001
  - Agile Manifesto meeting - Ken, Jeff, and Mike Beedle represent Scrum as signatories.
  - Agile Alliance formed, Ken writes first book on Scrum and starts Certification Course in 2003.

Scrum Resources

- Aqqurite (http://aqqurite.se/material-from-certified-scrummaster-training)
- The Scrum Alliance (www.scrumalliance.org)
- Agile Alliance (www.agilealliance.org)

Literature


The Addison-Wesley Signature Series

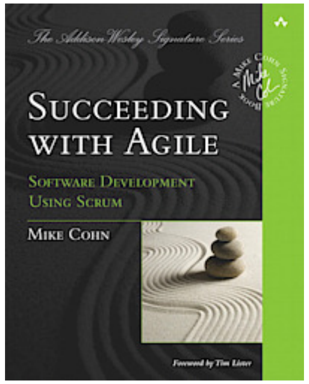
A Mike Cohn Signature Book

Essential Scrum

A Practical Guide to the Most Popular Agile Process

Kenneth S. Rubin

Forewords by Mike Cohn and Ron Jeffries


The Addison-Wesley Signature Series

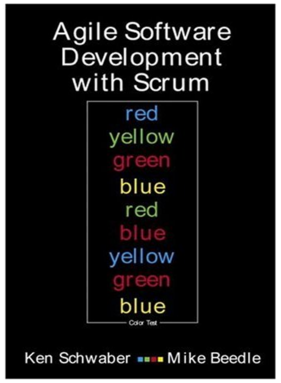
A Mike Cohn Signature Book

Succeeding with Agile

Software Development Using Scrum

Mike Cohn

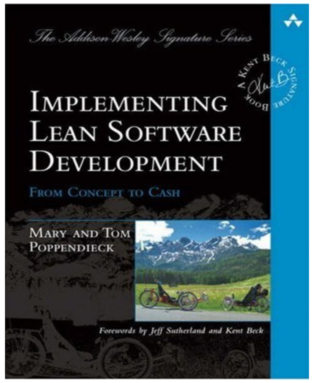
Foreword by Tim Lister


Agile Software Development with Scrum

red
yellow
green
blue
red
blue
yellow
green
blue

Color Test

Ken Schwaber Mike Beedle


The Addison-Wesley Signature Series

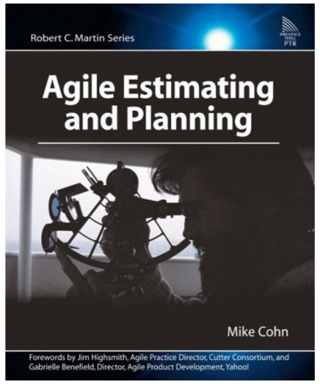
A Kent Beck Signature Book

Implementing Lean Software Development

From Concept to Cash

Mary and Tom Poppendieck

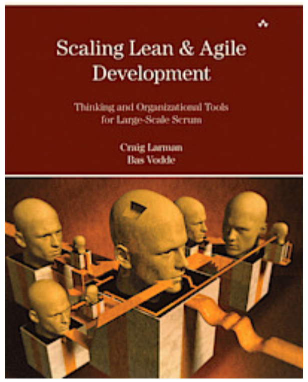
Forewords by Jeff Sutherland and Kent Beck


Robert C. Martin Series

Agile Estimating and Planning

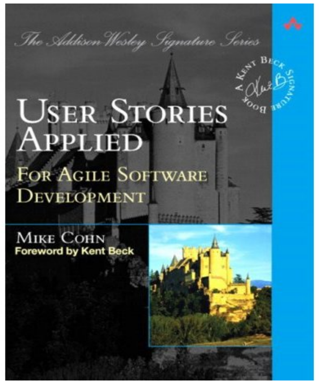
Mike Cohn

Forewords by Jim Highsmith, Agile Practice Director, Cutter Consortium, and Gabrielle Benefield, Director, Agile Product Development, Yahoo!


Scaling Lean & Agile Development

Thinking and Organizational Tools for Large-Scale Scrum

Craig Larman
Bas Vodde


The Addison-Wesley Signature Series

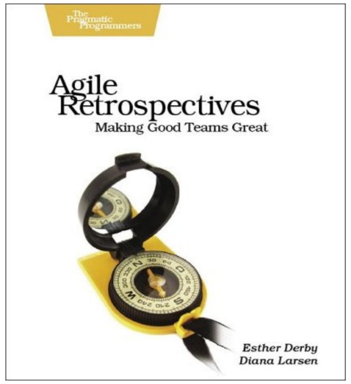
A Kent Beck Signature Book

User Stories Applied

For Agile Software Development

Mike Cohn

Foreword by Kent Beck


The Pragmatic Programmers

Agile Retrospectives

Making Good Teams Great

Esther Derby
Diana Larsen

Contacts

Arne Åhlander

- arne.ahlander@aqqurite.se
- www.aqqurite.se
- www.twitter.com/ArneAhl
- http://www.linkedin.com/in/arneahlander

aqqurite

PART 3
WORK MATERIAL & EXERCISES

Exercise: Great ScrumMasters

- Read the scenario
- Take a few minutes to discuss with your team what a great ScrumMaster would do in this situation
- Be ready to share your thoughts with the group.

SCENARIO 1

- *The CEO of the company appears in the middle of the Sprint, and says to you: one of our clients has a special request which if we can complete it, we will win $1 million in business.*

SCENARIO 2

- *The product owner says that he's not going to be available to attend the Sprint planning meeting, but he doesn't mind if the team goes ahead and does it without him.*

SCENARIO 3

- *In the middle of the Sprint, one of the team members manager comes and says she needs to pull him off the project for a couple days, to work on something else.*

SCENARIO 4

- *It's halfway through the Sprint, and the team is way behind on progress. It looks like there's no way it's going to finish what it committed to during the Sprint.*

SCENARIO 5

- *One member of the team looks really unhappy. He seems very distracted, and he's way behind on the tasks that he committed to complete.*

SCENARIO 6

- *One of the team members comes to you and tells you that the product owner just asked her to add an item to the current Sprint. Right now, the team is a third of the way through the Sprint.*

SCENARIO 7

- *One member of the team speaks up and says he thinks the retrospective is a waste of time; several other members of the team murmur in agreement, and someone else suggests that the team stop doing the retrospective.*

SCENARIO 8

- *The team appears to be very stressed out. They are having to work late most nights of the week, and they even have to work Saturdays every now and again, in order to meet their Sprint goals. You hear comments like scrum is awful it forces us to work so hard.*

SCENARIO 9

- *The Team has encountered problems three Sprints in a row and has been forced to decrease scope all three Sprints.*

SCENARIO 10

- *One team member does not want to sit with the rest of the team, but rather have a work place of his own.*

SCENARIO 11

- *The Team and the Product Owner can not agree upon definition of Done.*

SCENARIO 12

- *A team member refuses to go with a simpler solution that will meet the sprint goal because his line manager has told him that he wants a more advanced solution to enable the migration to a new version of a certain software.*