COMP9900 Information Technology Project
COMP3900 Computer Science Project

Course Overview

Teamwork Strategies

Software Tools

Week 1

School of Computer Science and Engineering (CSE)

Lecturer: Dr. Tatjana Zrimec

Course Email: cs9900@cse.unsw.edu.au

Course Website:

http://www.cse.unsw.edu.au/~cs9900

or http://www.cse.unsw.edu.au/~cs3900

Course Information

Lectures: **Tuesday 12:00 - 14:00**

Location: Science Theatre (K–F13–G09)

Lectures: **Thursday 16:00 - 18:00**

Location: Keith Burrows Theatre (K–J14–G5)

(Weeks 1, 2, 3 and 10)

Consultation**: Monday or Tuesday**

Location and time: will be announced

Course Information

- *Labs*: **Weeks 2 to 10, (11) - 2 hours per week in labs**

Note: Monday Labs Week 4 (public holiday) will be held in Week 11 at the same time and location

- *Labs – Progress demo*: **Weeks 4, 6, 8, 10, (11) - 2 hours per week in labs**

COMP9900 – IT Project
COMP3900 – CS Project

- A capstone software project
- Students work in teams
  - To define
  - To implement
  - To evaluate
    - A real-world software system

  - A brief and generic description of the project with some requirements will be provided

Course Summary

A capstone software project. Students work in teams to define, implement and evaluate a real-world software system. Most of the work in this course is team-based project work, although there are some introductory lectures on software project management and teamwork strategies. Project teams meet **weekly/fortnightly** with project mentors to report on the progress of the project. Assessment is based on a project proposal, a final project demonstration and report, and on the quality of the software system itself. Students are also required to reflect on their work and to provide peer assessment of their team-mates' contributions to the project.

Labs

- Regular group meetings (more than once per week) among team members
- Weekly/Fortnightly meeting with the mentor
- Fortnightly sprint demos to mentor
- Attendance to labs is assumed

Lectures

- A few introduction/kickstart lectures at the beginning
- One wrap-up lecture at the end

Readings

- No textbook
- Slides will be provided / linked from the course homepage
- Online resources may be provided from time to time (e.g., in the class, or via the tutor/mentor in the lab)

Assumed Knowledge

Before commencing this course, students should be able to:

- produce correct software programs in Python, Java or C/ C++, i.e., compilation, running, testing, debugging, etc.
- produce readable code with clear documentation.
- have basic knowledge of database programming, Web programming and/or script programming (such as Python, PHP, JavaScript).

Assumed knowledge

For COMP9900, students must:

- Be in their final semester of study, and
- Have completed at least 72 UOC towards MIT program 8543.

Assumed knowledge

For COMP3900, students must:

- Have successfully completed COMP1531 and (COMP2521 or COMP1927)
- Be enrolled in a BSc Computer Science major
- Have completed at least 120 UOC (to make sure that the student is in her/his final year)

Learning outcomes

- work as an effective member of a team to develop a software project (with a real-world application in mind)
- develop strategies leading to the development of high-performing, self-managing team
- develop skills to handle team issues during different phases of software development
- communicate effectively, orally and in writing, with peers, mentor (as supervisor) and lecturer (as stakeholder)
- appreciate and be familiar with the role of project management in software development
- produce clear and comprehensive software project proposal and report
- participate in a group presentation, demonstration, and liaising with mentor and stakeholder
- use relevant software tools to support effective software management.

Assessment

- No written exam

Assessment – ONE project

Item	Due	Mark
Proposal	Friday Week 3 @ 23.59	10%
Project Demo	During Week 10 Lab	15%
Project Report	Friday Week 10 @ 23.59	15%
Software Quality	Friday Week 10 @ 23.59	40%
Peer Assessment	Sunday Week 10 @ 23.59	20%

The project

- Team-based project
- Each team has ideally four (4) members

An Introduction to

Agile SCRUM Methodology

Modified for COMP3900/9900

Introduction

Classical methods of software development have many disadvantages:

- *huge effort during the planning phase*
- *poor requirements conversion in a rapid changing environment*
- *treatment of staff as a* `factor of production'

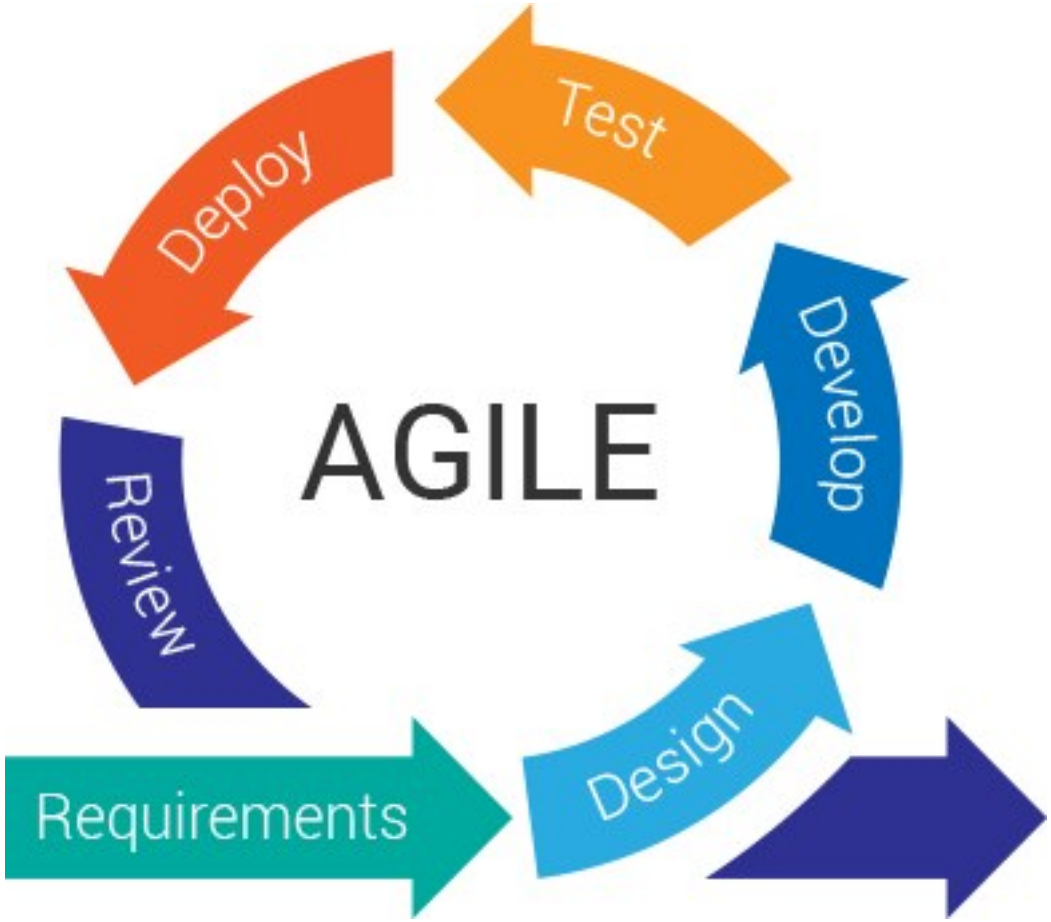
➢ *New methods:*

Agile Software Development Methodology

What is Agile ?

- Agile proponents believe
  - Current software development processes are too heavyweight or cumbersome
    - Too many things are done that are not directly related to software product being produced
  - Current software development is too rigid
    - Difficulty with incomplete or changing requirements
    - Short development cycles (Internet applications)
  - More active customer involvement needed

What is Agile ?



https://www.travancoreanalytics.com/why-agile-is-a-popular-choice/

Contd…

- Agile methods are considered
  - Lightweight
  - People-based rather than Plan-based
- Several agile methods
  - No single agile method
  - XP, SCRUM, …
- No single definition
- Agile Manifesto closest to a definition
  - Set of principles
  - Developed by Agile Alliance

Agile Manifesto

A Statement of 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

Agile Methods

Agile methods:

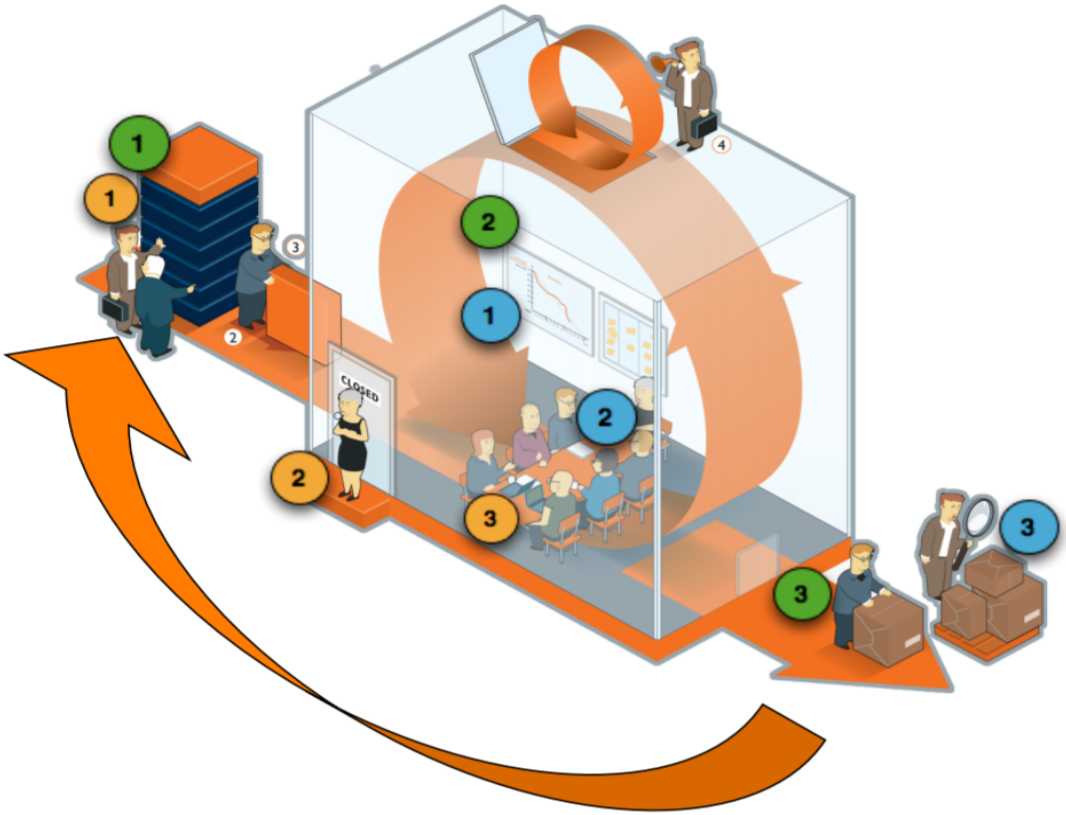
- Scrum
- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dynamic System Development Method (DSDM)
- …

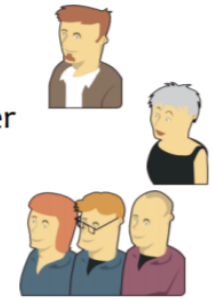
Agile Alliance (www.agilealliance.org)

- A non-profit organization promotes agile development

Scrum

Scrum in one Minute





Roles

1. Product Owner
2. Scrum Master
3. Team

Artefacts

1. Product Backlog
2. Sprint Backlog
   Burndown chart
   Impediment list
3. Product Increment

Ceremonies

1. Sprint Planning
2. Daily Scrum
3. Sprint Review

[Scrum process]

[Created by Rebecka Hall on Oct 11, 2017]

https://documentation.cochrane.org/display/WWIRPT/Scrum+process

Scrum in 100 words

Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.

It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).

The business sets the priorities. Our teams self-manage to determine the best way to deliver the highest priority features.

Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance for another iteration.

History of Scrum

1993:

analysis of common software development processes → not suitable for empirical, unpredictable and non-repeatable processes

Design of a new method: Scrum by Jeff Sutherland & Ken Schwaber

Enhancement of Scrum by Mike Beedle & combination of Scrum with Extreme Programming

1995:

Jeff Sutherland & Ken Schwaber :"The Scrum Development process" paper

introduction of Scrum at Object-Oriented Programming, Systems, Languages & Applications (OOPSLA) conference

2001:

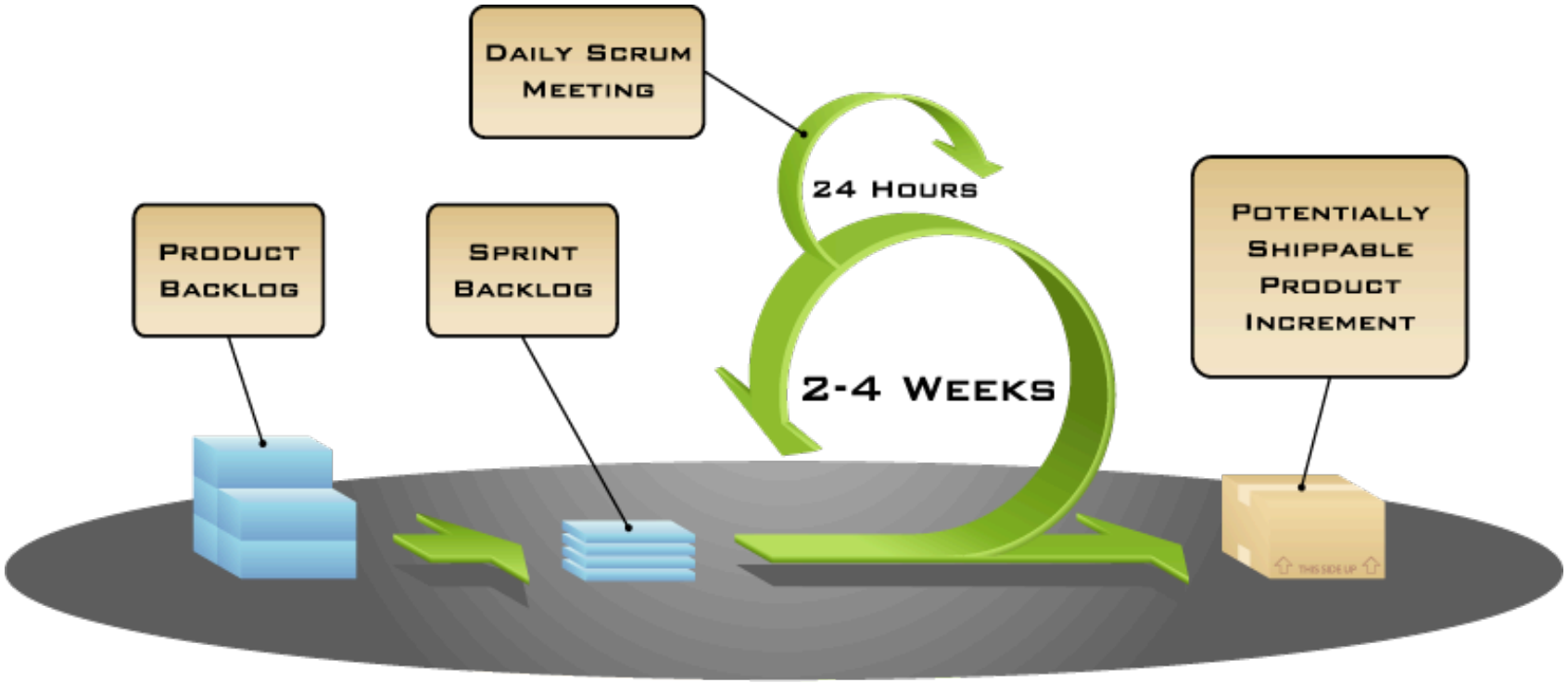
publication "Agile Software Development with Scrum" by Ken Schwaber & Mike Beedle

→ Successful application of Scrum in over 50 companies

Founders are members in the Agile Alliance

https://www.knowledgehut.com/tutorials/scrum-tutorial/scrum-history

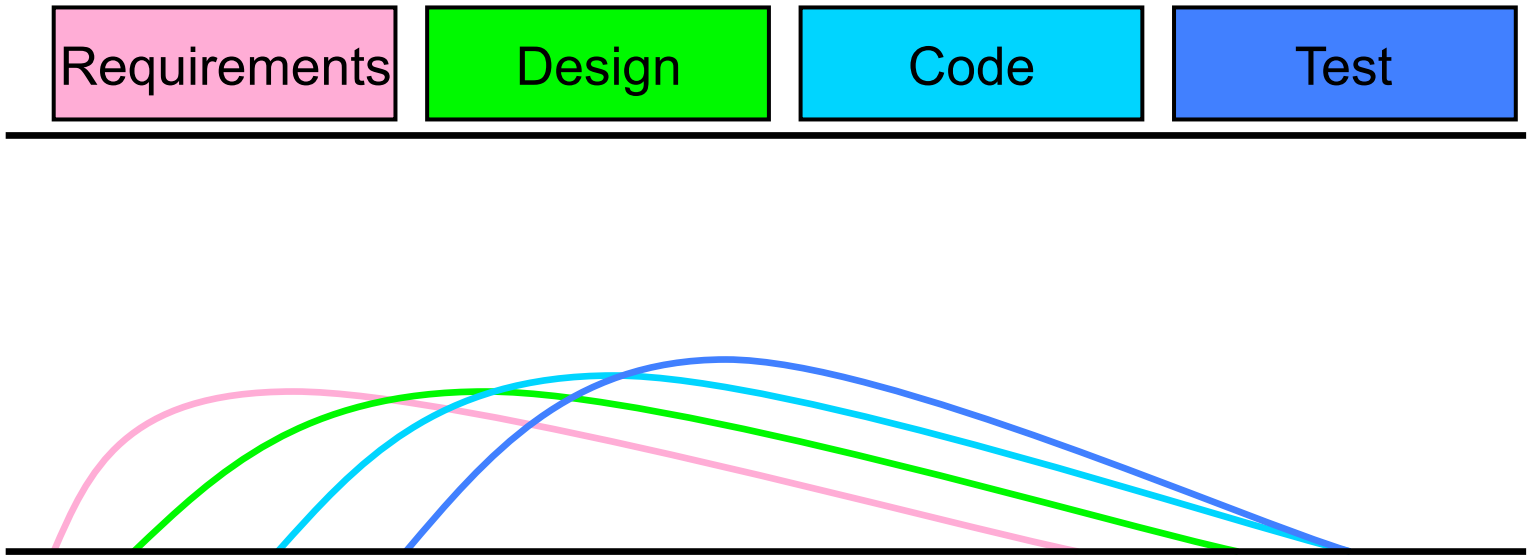
How Scrum Works?



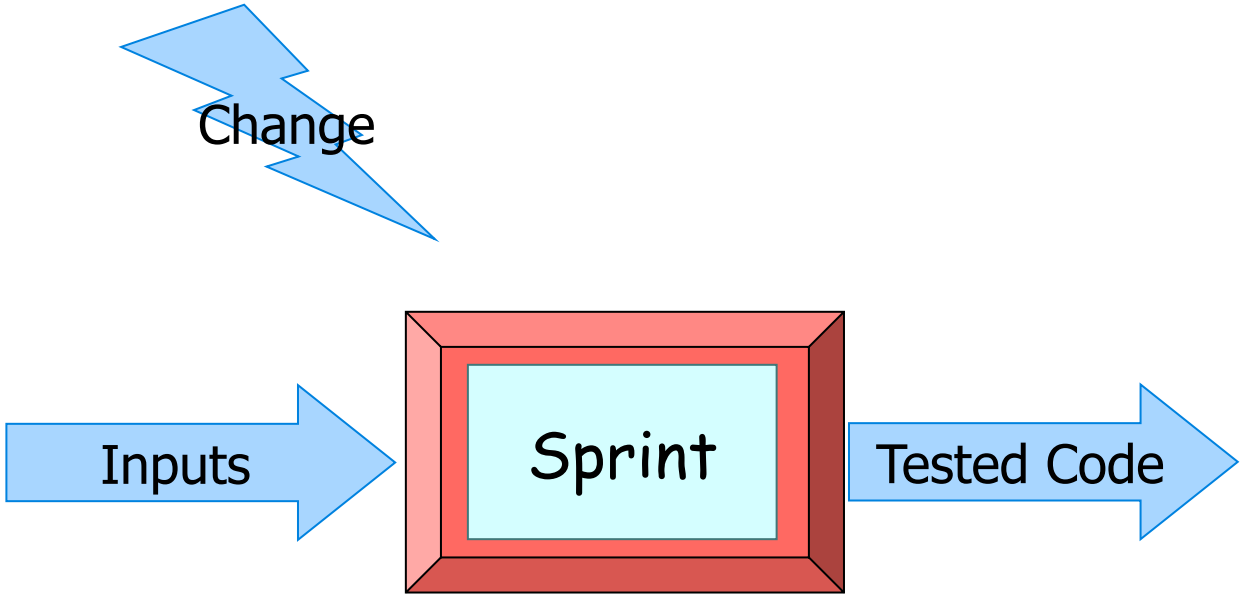
Sprints

- Scrum projects make progress in a series of "sprints"
- Target duration is two weeks
  - +/- a week or two
    - But, a constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

Sequential vs. Overlapping Dev.

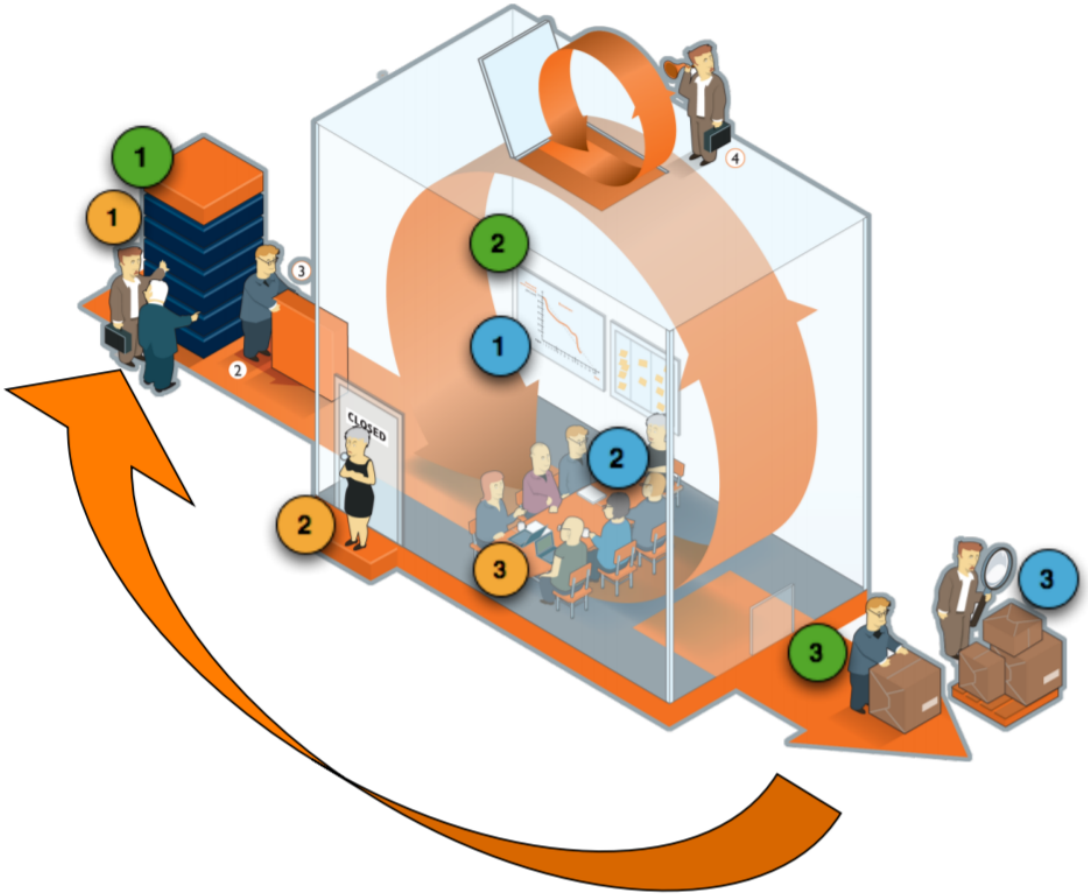


No changes during the sprint



Plan sprint durations around how long you can commit to keeping change out of the sprint

Scrum in one Minute



Roles

1. Product Owner
2. Scrum Master
3. Team

Artefacts

1. Product Backlog
2. Sprint Backlog
   Burndown chart
   Impediment list
3. Product Increment

Ceremonies

1. Sprint Planning
2. Daily Scrum
3. Sprint Review

Scrum Framework

Roles : Product Owner, Scrum Master, Scrum Team

Ceremonies : Sprint Planning, Sprint Review, Sprint Retrospective, and Daily Scrum Meeting

Artifacts : Product Backlog, Sprint Backlog, and Burndown Chart

Product Owner

Define the features of the product

Decide on release date and content

Be responsible for the profitability of the product (ROI)

Prioritize features according to market value

Adjust features and priority every iteration, as needed

Accept or reject work results.

The Scrum Master

The *Scrum master* removes obstacles that keep the development team from working at their highest capacity

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

Scrum Team

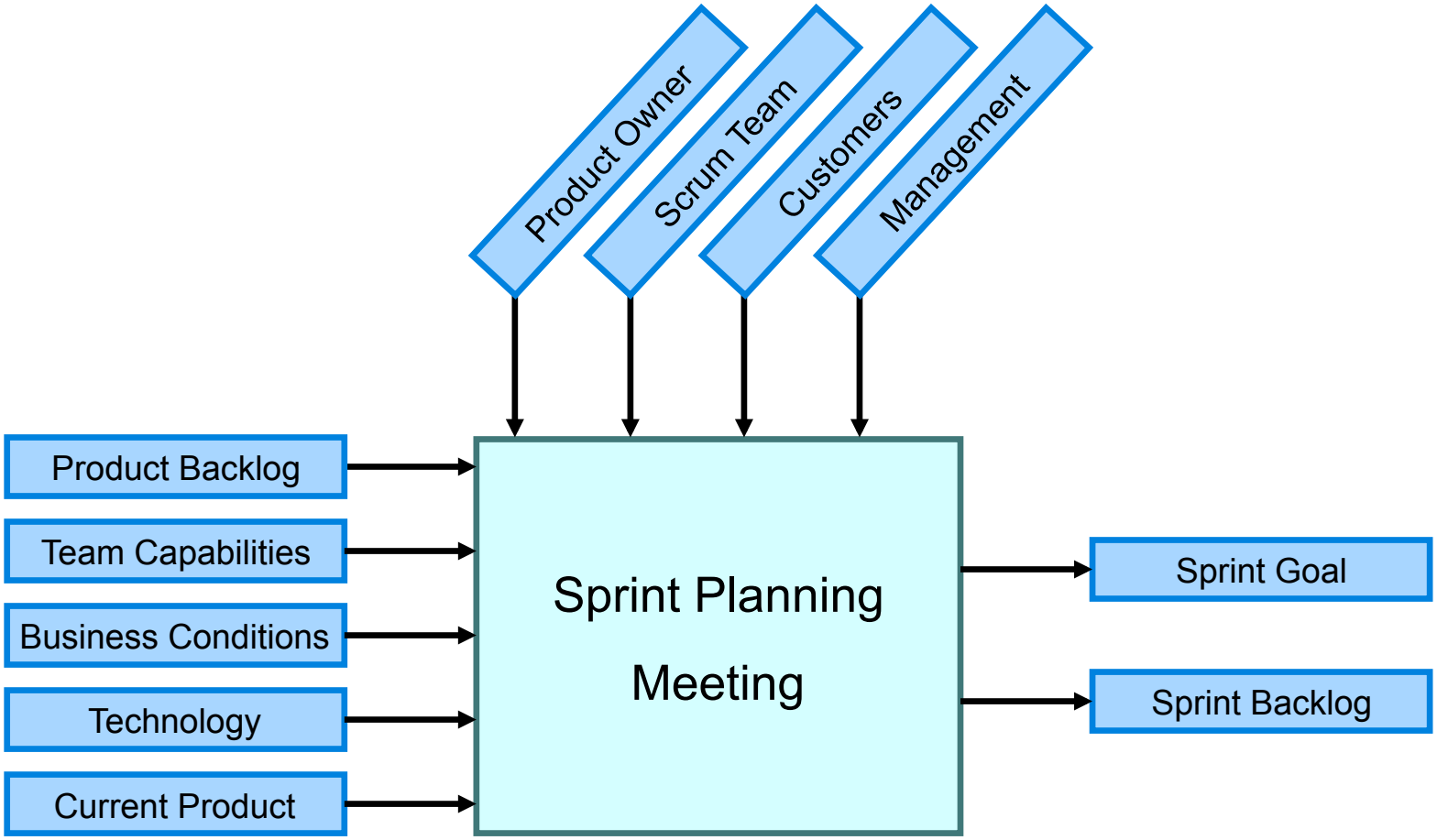
- Typically 5-10 people
- Cross-functional
  - QA, Programmers, UI Designers, etc.
- Members should be full-time
  - May be exceptions (e.g., System Admin, etc.)
- Teams are self-organizing
  - What to do if a team self-organizes someone off the team??
  - Ideally, no titles but rarely a possibility
- Membership can change only between sprints

Ceremonies

During every sprint cycle there are re-occurring ceremonies:

- Sprint Planning Meeting
- Sprint
- Daily Scrum
- Sprint Review Meeting

Sprint Planning Meeting



Parts of Sprint Planning Meeting

1st Part:

- Creating Product Backlog
- Determining the Sprint Goal
- Participants: Product Owner, Scrum Master, Scrum Team

2nd Part:

- Participants: Scrum Master, Scrum Team
- Creating Sprint Backlog

Pre-Project/Kickoff Meeting

A special form of Sprint Planning Meeting

Meeting before the beginning of the Project

Sprint

A month-long iteration, during which a product functionality is incremented

NO outside influence can interfere with the Scrum team during the Sprint

Each Sprint begins with the Daily Scrum Meeting

Daily Scrum

Parameters

- Daily
- 15-minutes
- Stand-up
- Not for problem solving

Three questions:

1. What did you do yesterday
2. What will you do today?
3. What obstacles are in your way?

Daily Scrum

- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a Scrum Master to track the progress of the Team

Scrum FAQs

Why daily?

> "How does a project get to be a year late?"
>
> > "One day at a time."
> >
> > > Fred Brooks, The Mythical Man-Month.

Can Scrum meetings be replaced by emailed status reports?

No

- Entire team sees the whole picture every day
- Create peer pressure to do what you say you'll do

Sprint Review Meeting

Team presents what it accomplished during the sprint

Typically takes the form of a demo of new features or underlying architecture

Informal

- 2-hour prep time rule

Participants

- Customers
- Management
- Product Owner
- Other engineers

Sprint Retrospective Meeting

Scrum Team only

Feedback meeting

Three questions

- Start (ideas for next sprint)
- Stop (what didn’t go well)
- Continue (what went well)

Don’t skip !!!

Product Backlog

- A list of all desired work on the project
  - Usually a combination of
    - story-based work (“let user search and replace”)
    - task-based work (“improve exception handling”)
- List is prioritized by the Product Owner (sometimes with the Scrum Master together)
  - Typically a Product Manager, Marketing, Internal Customer, etc.

Product Backlog

Requirements for a system, expressed as a prioritized list of Backlog Items

Is managed and owned by a Product Owner

Spreadsheet / JIRA (typically)

Usually is created during the Sprint Planning Meeting

Can be changed and re-prioritized before each sprint planning meeting

Sample Product Backlog

	Item #	Description	Est	By
Very High				
	1	**Finish database versioning**	16	KH
	2	**Get rid of unneeded shared Java in database**	8	KH
	-	**Add licensing**	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
	-	**Enforce unique names**	-	-
	7	In main application	24	KH
	8	In import	24	AM
	-	**Admin Program**	-	-
	9	Delete users	4	JM
	-	**Analysis Manager**	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
	-	**Query**	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
	-	**Population Genetics**	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	**Add icons for v1.1 or 2.0**	-	-
	-	**Pedigree Manager**	-	-
	20	Validate Derived kindred	4	KH
Medium				
	-	**Explorer**	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

From Sprint Goal to Sprint Backlog

Scrum team takes the Sprint Goal and decides what tasks are necessary

Team self-organizes around how they'll meet the Sprint Goal

- Manager doesn't assign tasks to individuals

Managers don't make decisions for the team (but they can decide priority)

Sprint Backlog is created

Sprint Backlog during the Sprint

- Changes
  - Team adds new tasks whenever they need to in order to meet the Sprint Goal
  - Team can remove unnecessary tasks
  - Sprint Backlog can only be updated by the team
- Estimates are updated whenever there's new information

Sprint Backlog

A subset of Product Backlog Items, which define the work for a Sprint

Is created ONLY by Team members

Each Item has it’s own status

Should be updated every day

Sprint Backlog

No more than 300 tasks in the list

If a task requires say more than 21 points (to be precise, a small number of points, to be discussed), it should be broken down

Team can add or subtract items from the list. Product Owner is not allowed to do it

Sample Sprint Backlog

	Days Left in Sprint	15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
	Total Estimated Hours:	**554**	**458**	**362**	**270**	**0**
-	**User's Guide**	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
	Misc. Small Bugs					
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
	Environment					
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
	Database					
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

Sprint Burndown Chart

Depicts the total Sprint Backlog hours (i.e., points) remaining per day

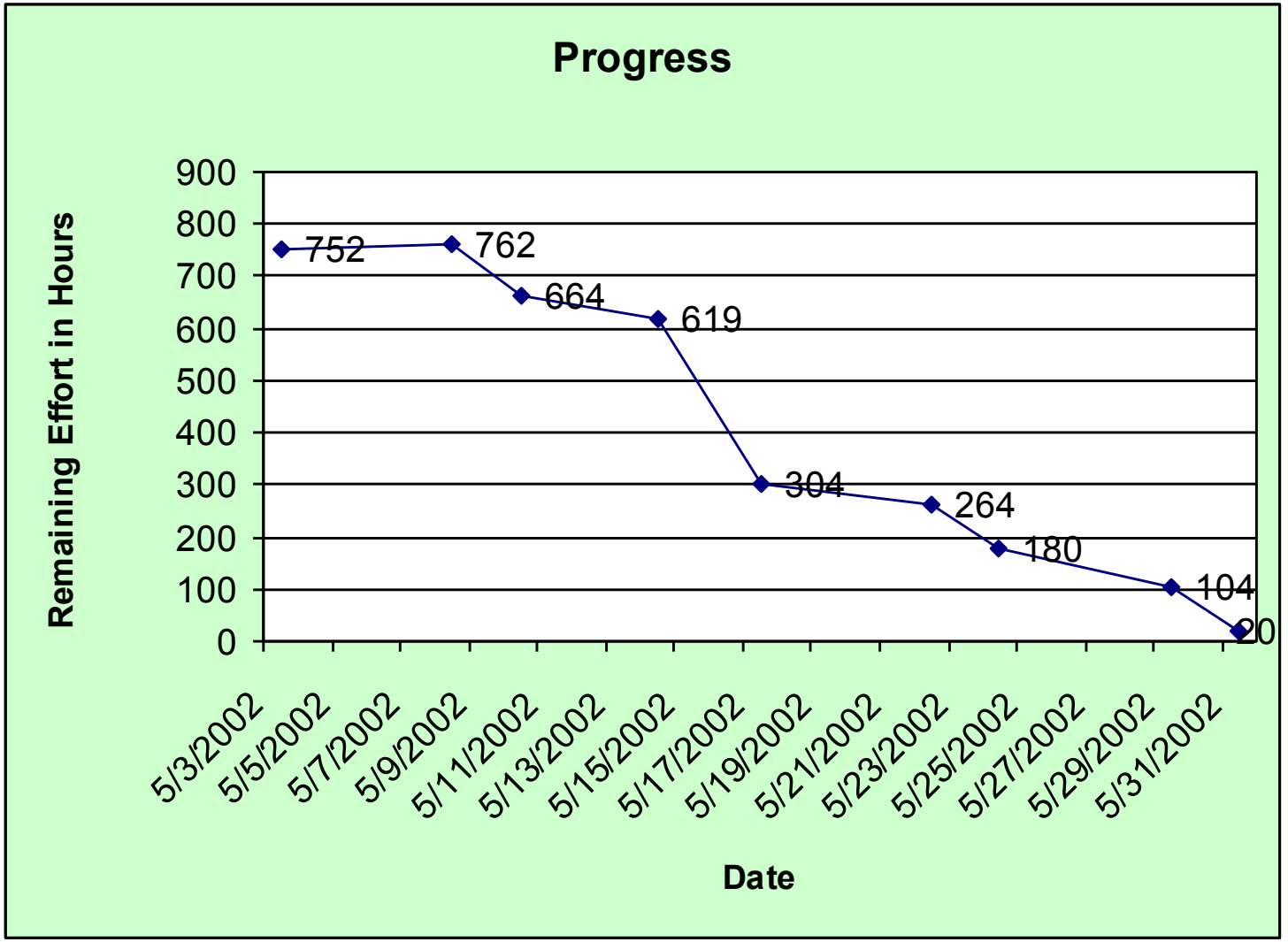
Shows the estimated amount of time to release

Ideally should burn down to zero to the end of the Sprint

Actually is not a straight line

Can bump UP

Sprint Burndown Chart



Release Burndown Chart

Will the release be done on right time?

X-axis: sprints

Y-axis: amount of hours remaining

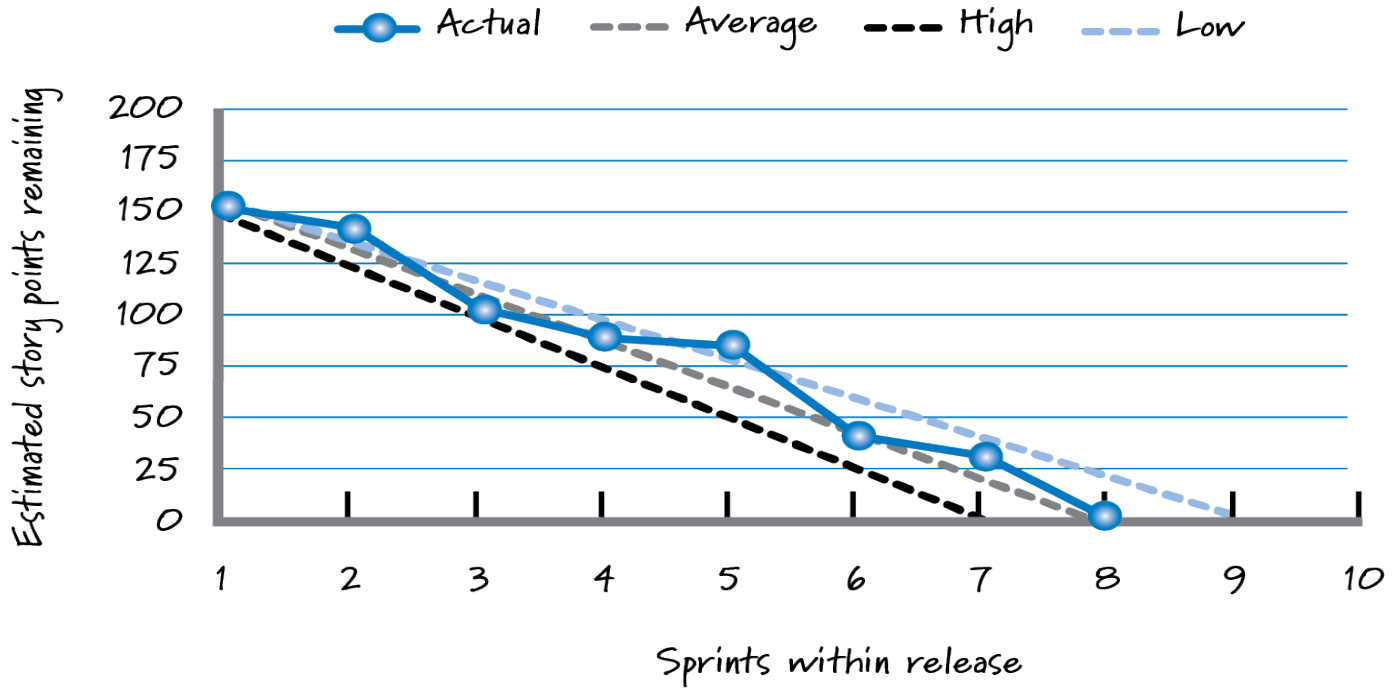
The estimated work remaining can also burn up

Product Burndown Chart

Is a “big picture” view of project’s progress (all the releases)

https://www.scrum-institute.org/Burndown_Chart.php

Product Burndown Chart



Notice that the release began with 150 points of estimated work to complete. After sprint 1, the team completed only 10 points, below even the lowest expected velocity. At the beginning of sprint 2, 140 points of work remained. At the end of sprint 2, however, the team had made up ground and was tracking closer to its highest expected velocity, with 100 points of work remaining. The team continued to plot the number of points until they had completed all of the work for the release.

https://innolution.com/blog/burn-charts-for-communicating-progress-through-a-scrum-release

Scalability of Scrum

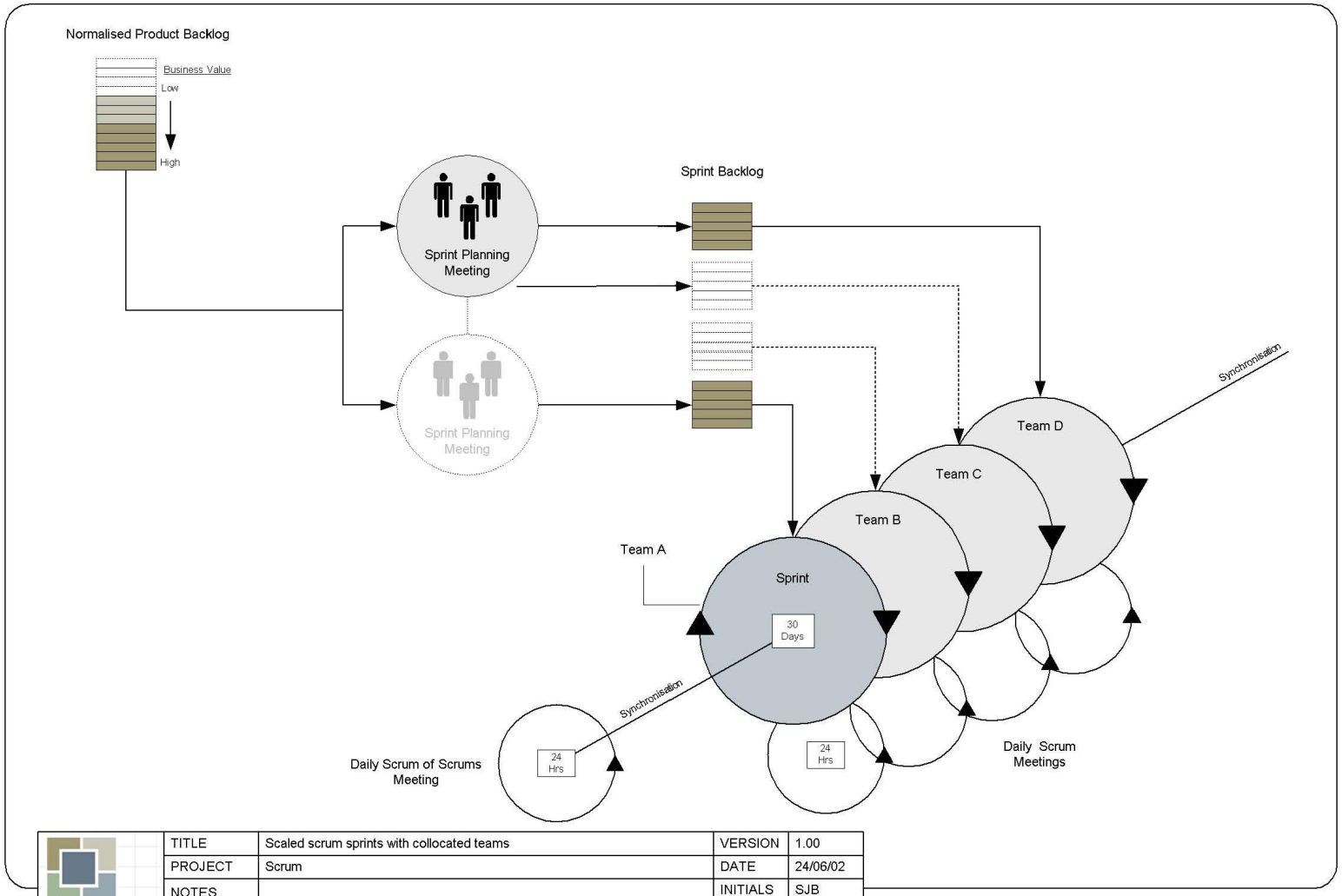
A typical Scrum team is 5-10 people

Jeff Sutherland - up to over 800 people

"Scrum of Scrums" or what called "Meta-Scrum“

Frequency of meetings is based on the degree of coupling between packets

Scalability of Scrum



TITLE	Scaled scrum sprints with collocated teams	VERSION	1.00
PROJECT	Scrum	DATE	24/06/02
NOTES		INITIALS	SJB
		PAGES	1 / 1

think-box
Thinking outside the box

Pros/Cons

- *Advantages*
  - *Completely developed and tested features in short iterations*
  - *Simplicity of the process*
  - *Clearly defined rules*
  - *Increasing productivity*
  - *Self-organizing*
  - *each team member carries a lot of responsibility*
  - *Improved communication*
- *Drawbacks*
  - *“Undisciplined hacking” (no written documentation)*
  - *Violation of responsibility*

References – Useful Links

- Scrum process, by Rebecka Hall on Oct 11, 2017

  https://documentation.cochrane.org/display/WWIRPT/Scrum+process
- Why Agile Is A Popular Choice

  https://www.travancoreanalytics.com/why-agile-is-a-popular-choice/
- Scrum History

  https://www.knowledgehut.com/tutorials/scrum-tutorial/scrum-history
- Manifesto for Agile Software Development

  http://www.agilemanifesto.org

References – Useful Links

- Advancing the Practice of Agile
  https://www.agilealliance.org

- Scrum Burndown Chart
  https://www.scrum-institute.org/Burndown_Chart.php
- Burn Charts for Communicating Progress Through a Scrum Release
  https://innolution.com/blog/burn-charts-for-communicating-progress-through-a-scrum-release