

Agile Methodologies - Scrum

Metodologias de Trabalho em Equipa

P.PORTO

What is Agile?



- Agile is an **iterative approach** to project management and software development that helps teams deliver value to their customers faster and with fewer headaches.
- Instead of betting everything on a "big bang" launch, an agile team delivers work in **small, but consumable, increments**.
- Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly.

Agile Manifesto

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

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

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

(<https://agilemanifesto.org/>)

Principles behind the Agile Manifesto

1. Our highest priority is to **satisfy the customer** through **early and continuous delivery** of valuable software.
2. Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the **shorter timescale**.
4. Business people and developers must **work together daily** throughout the project.
(..)

Principles behind the Agile Manifesto



(..)

5. Build projects around **motivated individuals**. Give them the environment and support they need and **trust** them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
7. **Working software** is the primary measure of progress.
8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to **maintain a constant pace indefinitely**.

(..)

Principles behind the Agile Manifesto

(..)

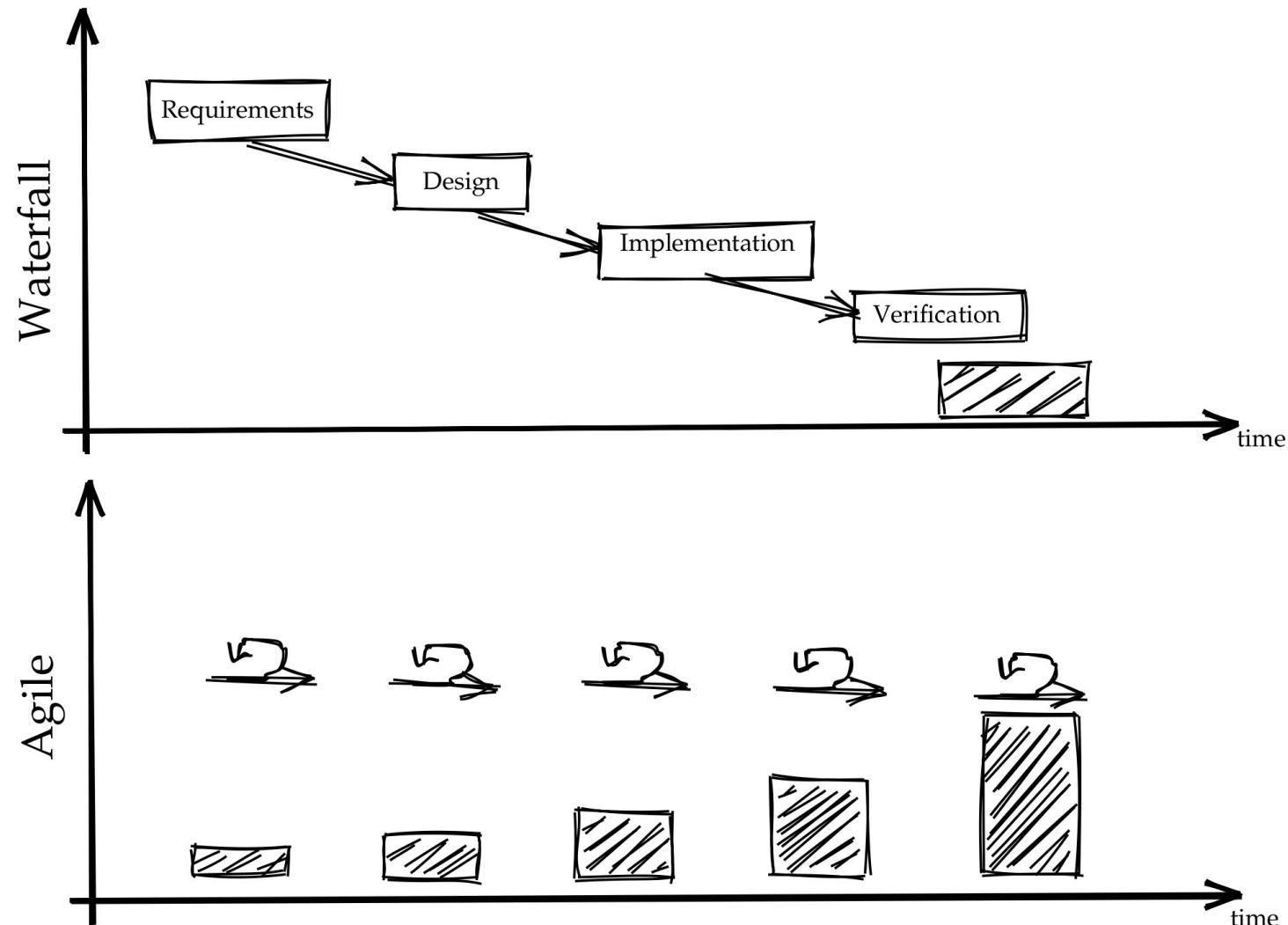
9. Continuous attention to **technical excellence** and **good design** enhances agility.
10. **Simplicity** – the art of maximizing the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from **self-organizing teams**.
12. At **regular intervals**, the team reflects on how to become **more effective**, then tunes and adjusts its behavior accordingly.

Characteristics of Agile Software Development



- Light weight Methodology
- Small to medium sized teams
- Vague and/or changing requirements
- Vague and/or changing techniques
- Simple design
- Minimal system into production

Traditional vs. Agile Delivery



Why we user (or should use) it?



- Reduced risk
- Earlier Return on Investment (ROI)/value
- Increased visibility of progress
- Increased predictability
- Increased productivity
- Reduced waste
- More productive & happy teams

What is an Agile Framework?



- Frameworks are **methodologies** or even **processes**
- Most agile teams use frameworks only as a starting point for their agile transformation, eventually customizing elements to meet their unique needs.
- There are many popular agile frameworks used by various organizations. Often these organizations **modify parts of the frameworks as they see fit and as they iterate on their own agile processes.**

Popular Agile Frameworks



- Scrum
- eXtreme Programming (XP)
- Dynamic Systems Development Method (DDSM)
- Feature Driven Development (FDD)
- Adaptive Software Development (ASD)
- Lean Software Development (LSD)
- Disciplined Agile (DA)
- Scaled Agile Framework (SAFe)
- Rapid Application Development (RAD)

Which Framework is Best?



- Unfortunately, there is no one-size-fits-all way to practice agile software development.
- There are many factors that may influence which framework you choose to work with. Such as:
 - Company size
 - Team structure
 - Available resources
 - Needs of stakeholders
 - Structure/size of your product portfolio
- Each framework has its own unique set of strengths and weaknesses



Scrum

Metodologias de Trabalho em Equipa

P.PORTO

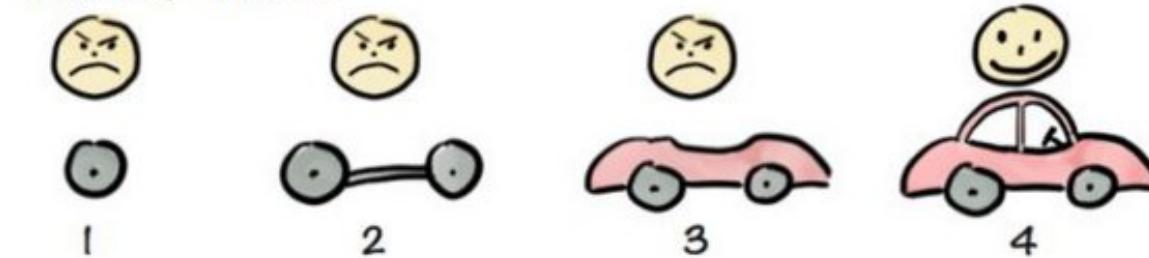


What is Scrum?

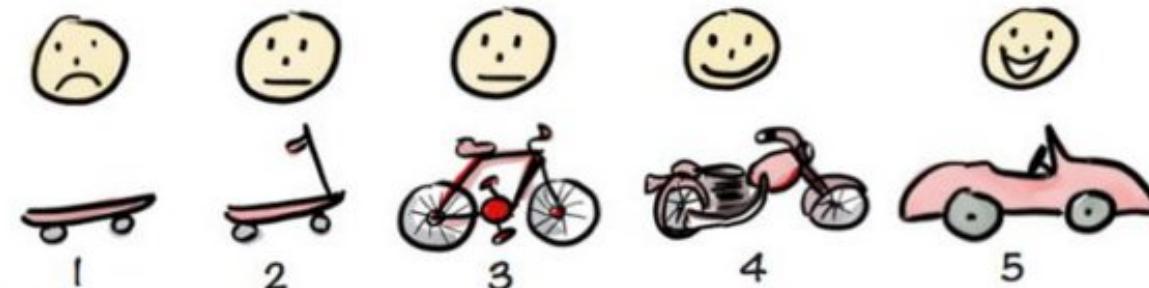


Incremental ≠ Iterative

Not like this....



Like this!



Henrik Kniberg

<http://softwaredevelopmenttoday.com/2015/09/how-to-explain-agile-and-incremental-delivery-to-anyone/>

Scrum in a nutshell

- 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**. Teams self-organize 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 it for another sprint.

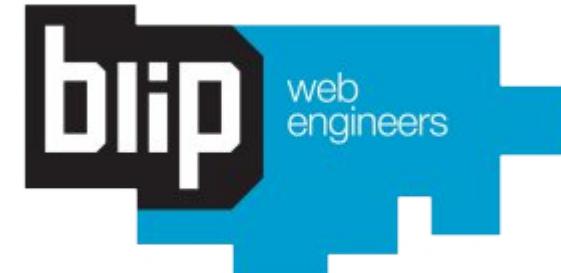
Scrum origins



- Jeff Sutherland
 - Initial scrums at Easel Corp in 1993
 - IDX and 500+ people doing Scrum
- Ken Schwaber
 - Scrum presented at OOPSLA 96 with Sutherland
 - Author of three books on Scrum
- Mike Beedle
 - Scrum patterns in PLOPD4
- Ken Schwaber and Mike Cohn
 - Co-founded Scrum Alliance in 2002, initially within the Agile Alliance



Scrum has been used by:



Scrum has been used for:



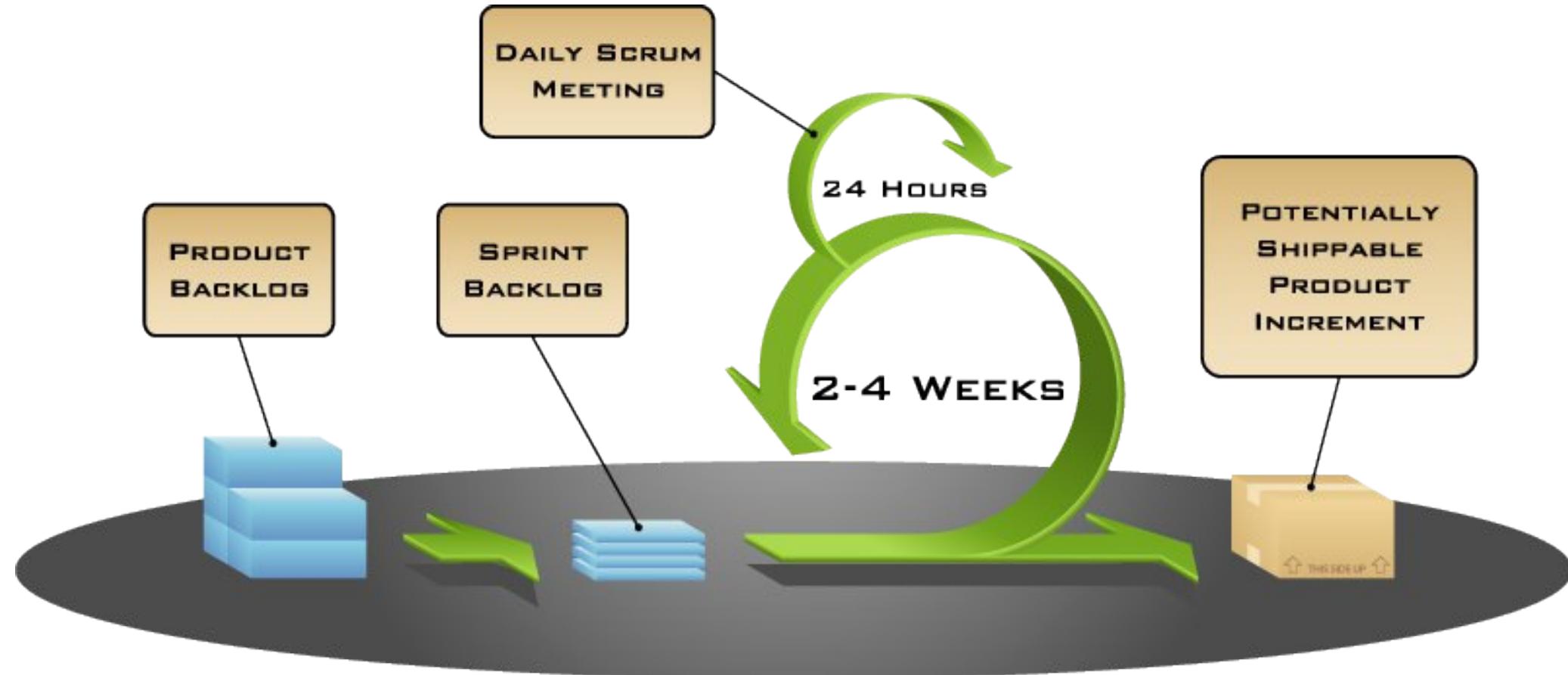
- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

Characteristics



- Self-organizing teams
- Product progresses in a series of *sprints*
- Requirements are captured as items in a list of *product backlog*
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects

Scrum framework



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

[\(www.mountaingoatsoftware.com/scrum\)](http://www.mountaingoatsoftware.com/scrum)

Sprints



- Scrum projects make progress in a series of *sprints*
- Typical duration is **2–4 weeks** or a calendar month at most
- A **constant duration** leads to a better rhythm
- Product is designed, coded, and tested during the sprint

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

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

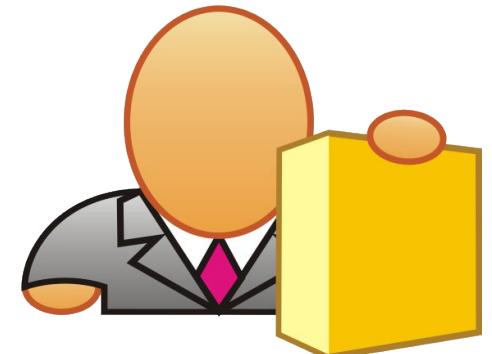
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

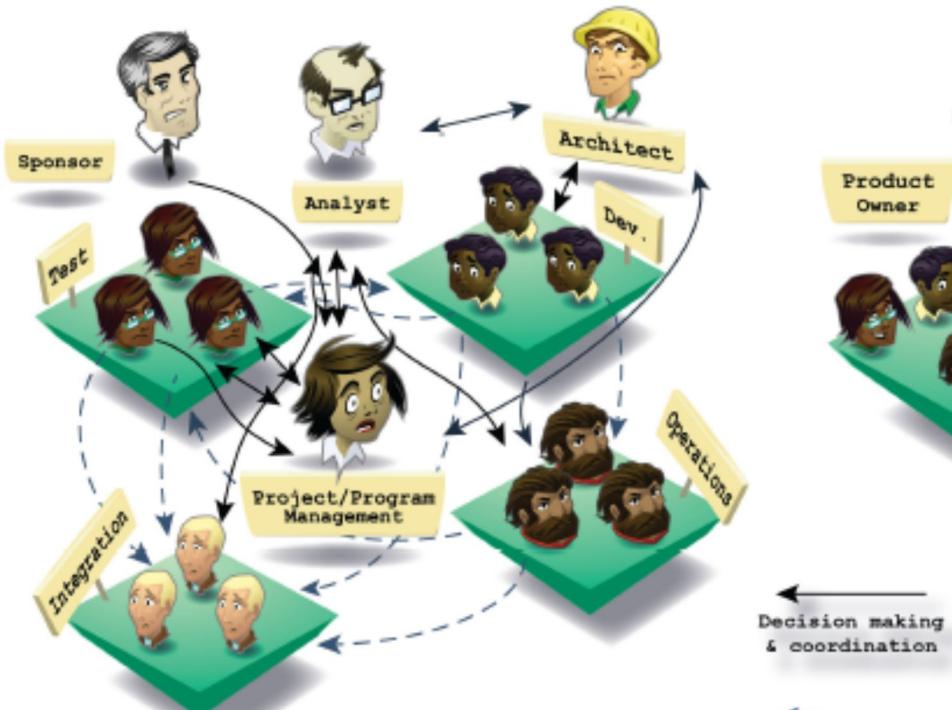
Product owner

- Define the features of the product (as User Stories)
- Decide on release date and content
- Be responsible for the profitability of the product
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

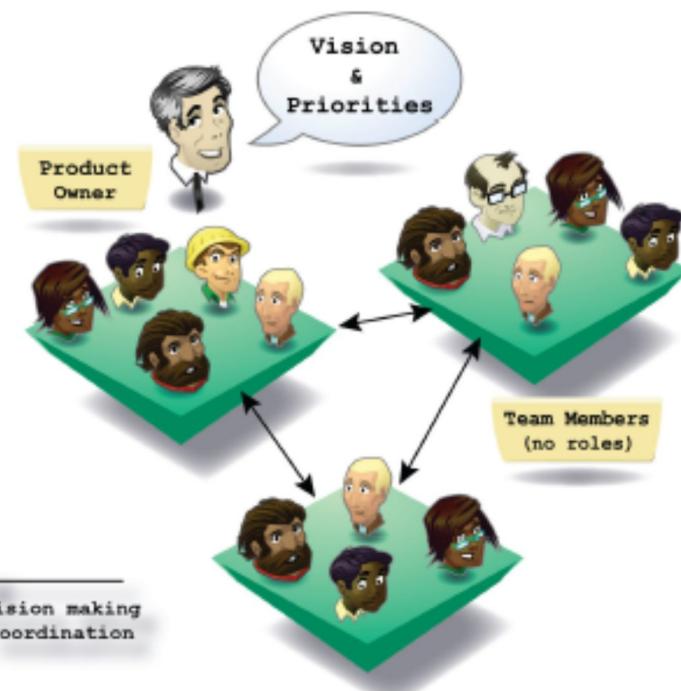


Product owner

Traditional Coordination



Agile Coordination



The ScrumMaster

- 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



The team

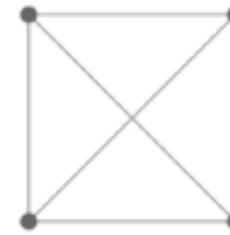
- Typically, 5–9 people
- Cross-functional:
 - Programmers, testers, user experience designers, etc.
- Members should be full-time
 - May be exceptions (e.g., database administrator)
- Teams are self-organizing
 - Ideally, no titles but rarely a possibility
- Membership should change only between sprints



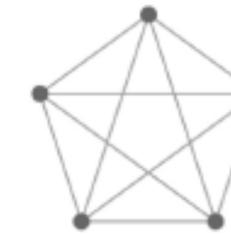
The team size



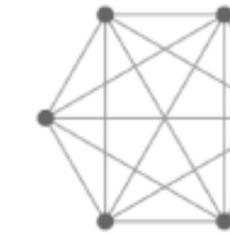
3 people, 3 lines



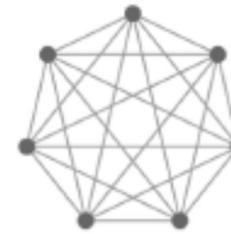
4 people, 6 lines



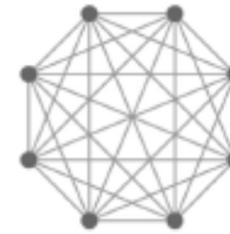
5 people, 10 lines



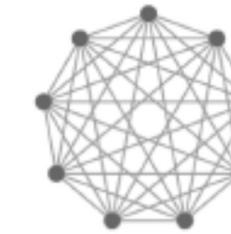
6 people, 15 lines



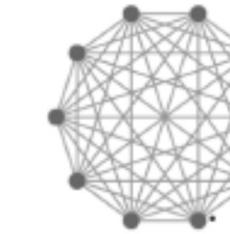
7 people, 21 lines



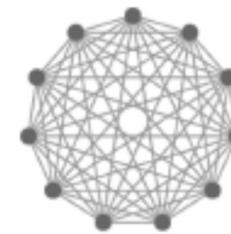
8 people, 28 lines



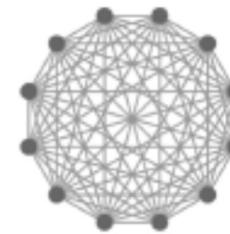
9 people, 36 lines



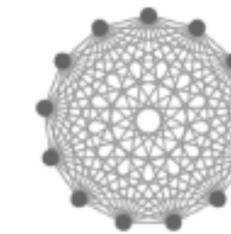
10 people, 45 lines



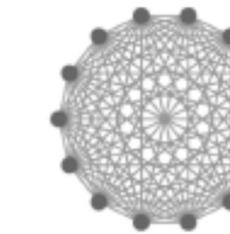
11 people, 55 lines



12 people, 66 lines



13 people, 78 lines



14 people, 91 lines

High-performing teams are ...



STABLE

Members of the team aren't changed frequently

LONG-LIVED

Products will come and go but the team remain together for years

SMALL

Small enough to avoid silos and maximize Collaboration

DEDICATED

Each person is a Member of one and only one team

CROSS- FUNCTIONAL

The team has the functional skills to archive done

STEADILY IMPROVES

Continuous improvement helps the team to be constantly evolving



Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Sprint planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
 - Tasks are identified and each is estimated (1-16 hours)
 - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a vacation planner, I want to see photos of the hotels.



Code the middle tier (8h)
Code the user interface (4h)
Write test fixtures (4h)
Code the foo class (6h)
Update performance tests (4h)

The daily scrum

- Parameters:
 - Daily
 - 15-minutes
 - Stand-up
- Not for problem solving
 - Whole world is invited
 - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings



The daily scrum

- Everyone answers 3 questions:

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are **not** status for the Scrum Master
 - They are commitments in front of peers

The sprint review

- 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
 - No slides
- Whole team participates
- Invite the world



Sprint retrospective

- Periodically look at what is and is not working
- Typically, 15–30 minutes
- Done after every sprint
- Whole team participates
 - ScrumMaster
 - Product owner
 - Team
 - Possibly customers and others



Sprint retrospective



- Whole team gathers and discusses what they'd like to:
 - Start doing
 - Stop doing
 - Continue doing

This is just one of many ways to do a sprint retrospective.

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog

- The requirements (user stories, bugs, tech tasks, study)
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product.
- Prioritized by the product owner.
- Reprioritized at the start of each sprint.



User Stories Context

As a ... *(user of the system)*
I want ... *(feature or problem to be solved)*
So that ... *(benefit of story being completed)*

Writing User Stories is
a **Product Owner Job**.

The “so that” part is
incredibly valuable as
it **focuses people on**
the real reason
behind this story.

AND EACH FEATURE
NEEDS TO HAVE
WHAT WE CALL A
“USER STORY.”



INVEST Acronym



- Set of criteria to assess the quality of a User Story

Independent

Negotiable

Valuable

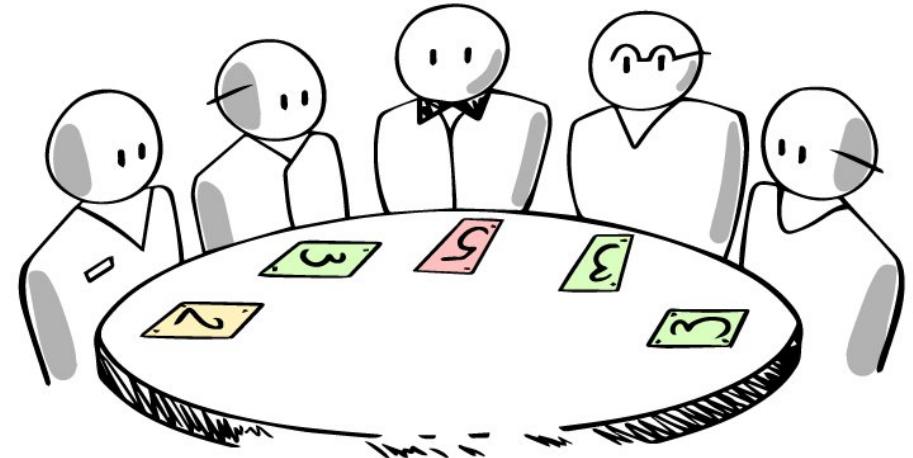
Estimable

Small (sized appropriately)

Testable

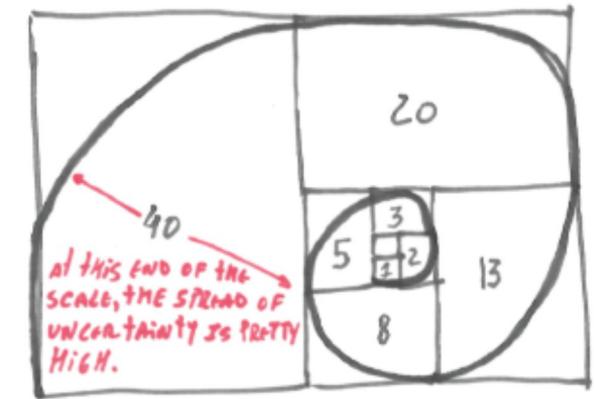
Some Agile Estimation Techniques

- Planning Poker
- T-Shirt Sizes
- Dot Voting
- The Bucket System
- Affinity Mapping
- ...



Planning Poker

- Makes use of story points to estimate the difficulty of the task at hand.
- Based on the Fibonacci sequence, the story point values that can be assigned are 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100.
- Each of these represent a different level of complexity for the overall project



Planning Poker

■ Procedure:

- Product Owner reads and explains the User Story
- The team discusses the requirements
- Each team member scores the US with points from the sequence
- If the story point estimations match up, that will be the final estimate.
- Otherwise, the team members who gave the lowest and highest points can voice their reasoning, and more discussion will ensue until there is a consensus.

■ Useful tools:

- Mobile Apps
- Add-ons for Slack

T-Shirt Sizes

- This technique uses T-Shirts sizes as story points for the size of the project.
- Is a useful method for being time-efficient. It can give a quick and rough estimate for how much work is expected for a project.
- The sizes can be converted into numbers at a later stage – when the team assigns a relative size to the project on hand.
- This is decided through discussion and collaborative efforts to understand everything that needs to be done.

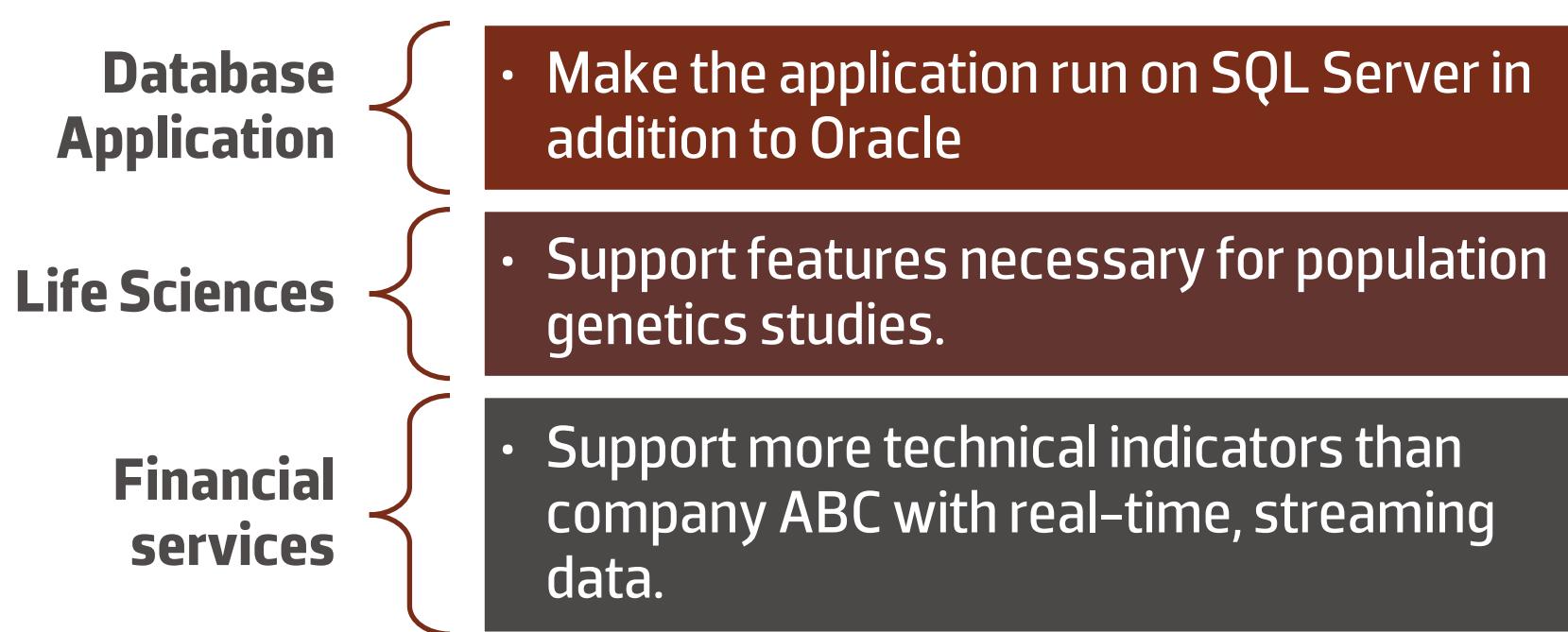
A sample product backlog



Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

The sprint goal

- A short statement of what the work will be focused on during the sprint



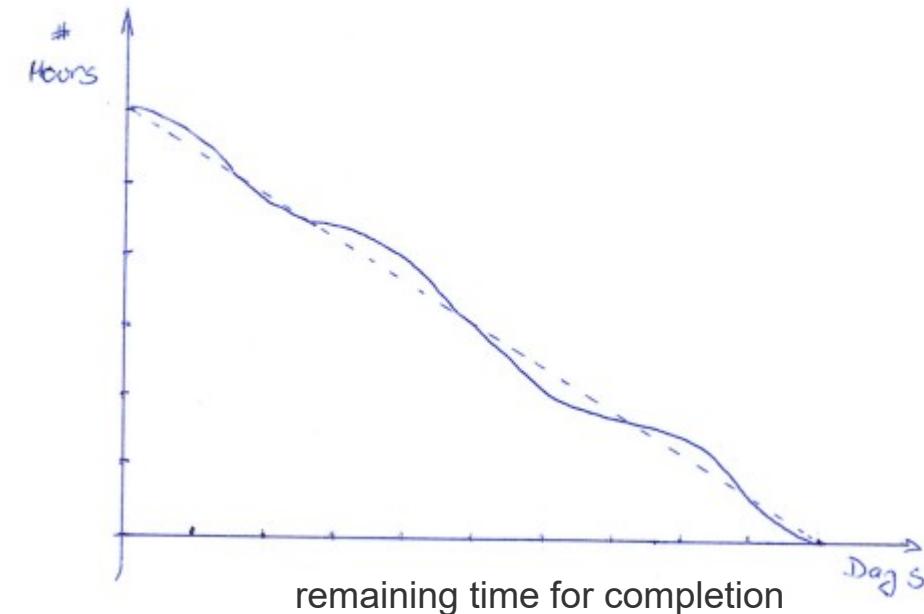
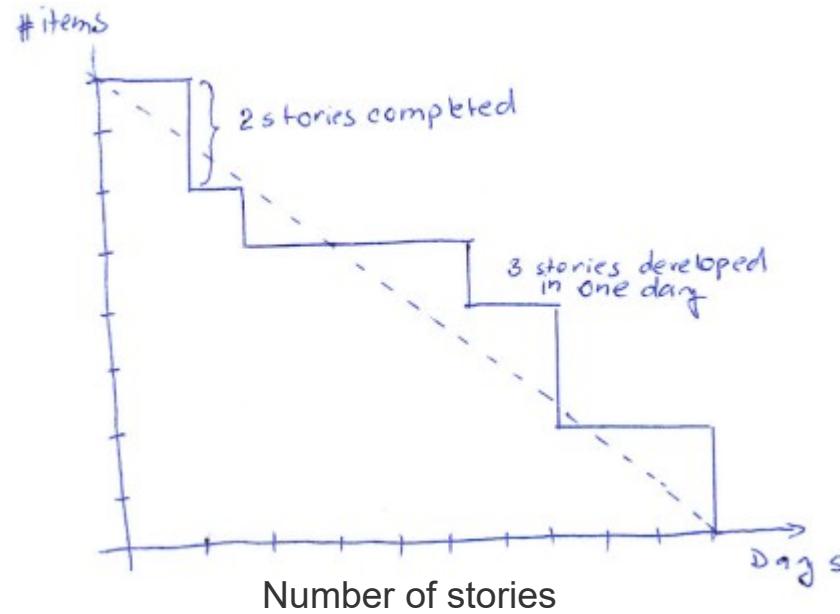
Managing the sprint backlog

- Individuals sign up for work of their own choosing
 - Work is never assigned
- Estimated work remaining is updated daily
- Any team member can change the sprint backlog tasks states.
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known

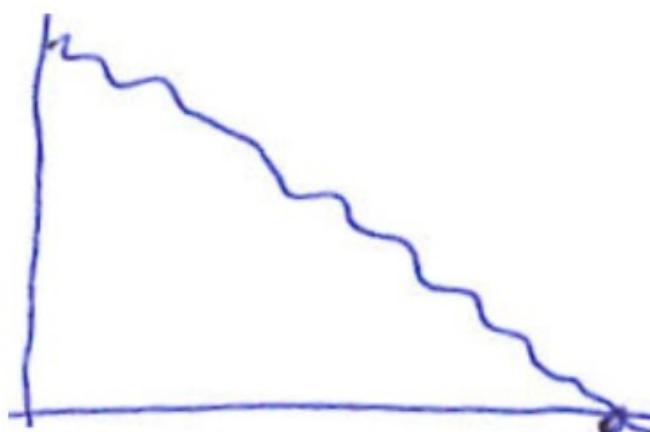
A sprint backlog

Stories	To Do	In Progr.	Testing	Done !
US N° ⑨ ⑩ ⑪ ⑫	US N° ⑦ US N° ⑧	US N° ⑥	US N° ④ US N° ⑤	US N° * ① US N° ② US N° ③ *

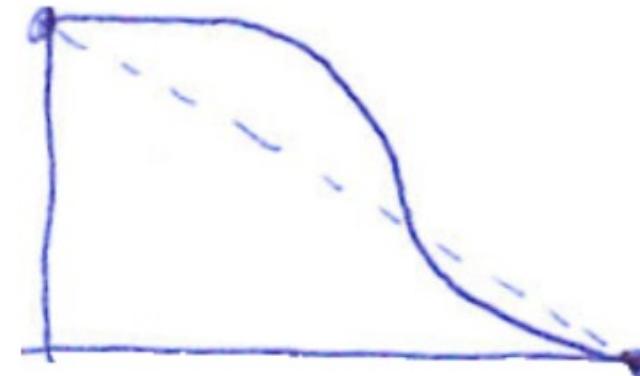
A sprint burndown charts



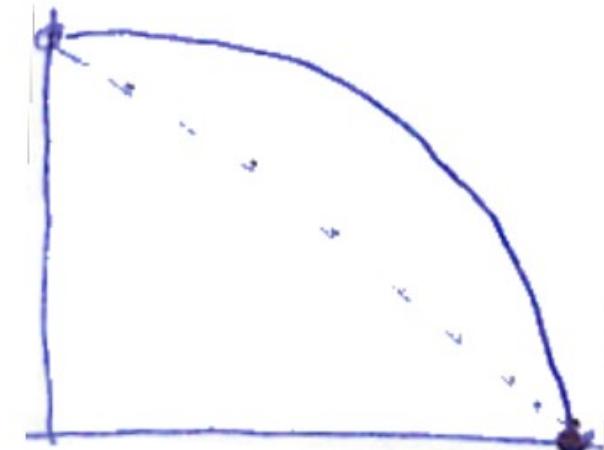
What a burndown chart can say about the team?



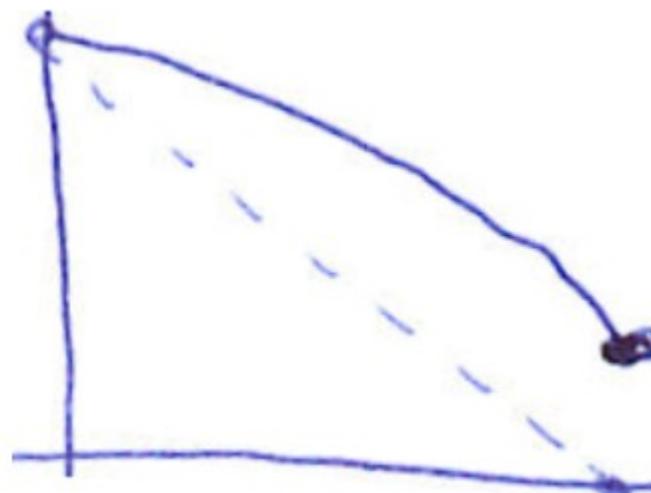
Ideal team



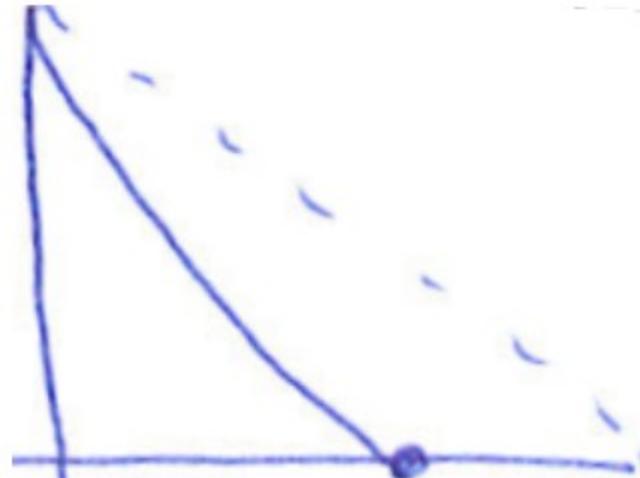
Great team



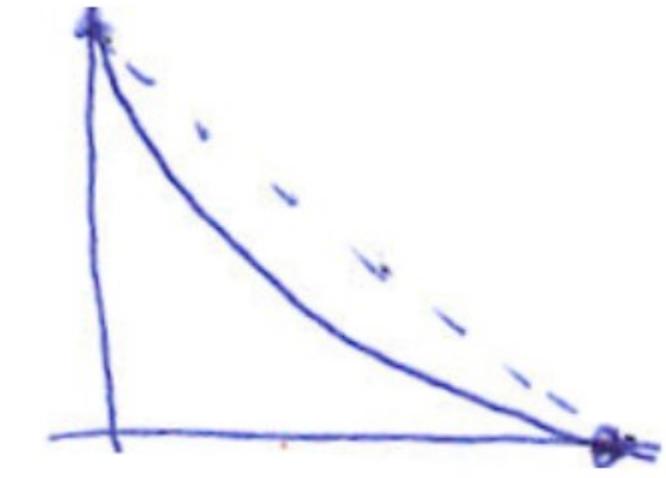
Nice team



Too late



Too early



They need to rest ...

Agile Methodologies - Scrum

Metodologias de Trabalho em Equipa

P.PORTO