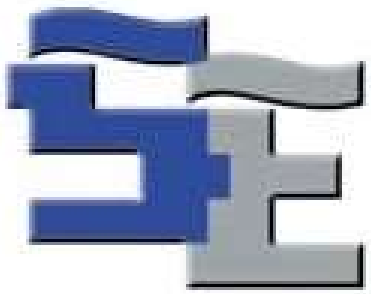


Software Engineering

SCRUM

SCRUM – Agile Project Management

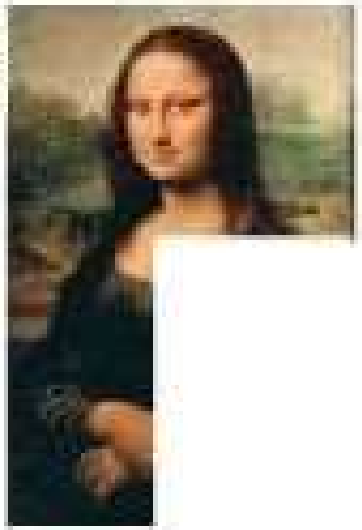
- SCRUM is an agile, lightweight process for managing and controlling software and product development in rapidly changing environments.

  - Iterative, incremental process
  - Team-based approach
  - developing products with rapidly changing requirements
  - Controls the chaos of conflicting interest and needs
  - Improve communication and maximize cooperation
  - Protecting the team form disruptions and impediments
  - A way to maximize productivity

Incremental

1

2



3

Iterative

1

2

3

History of Scrum

- 1995:
  - 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

- 1996: introduction of Scrum at OOPSLA conference

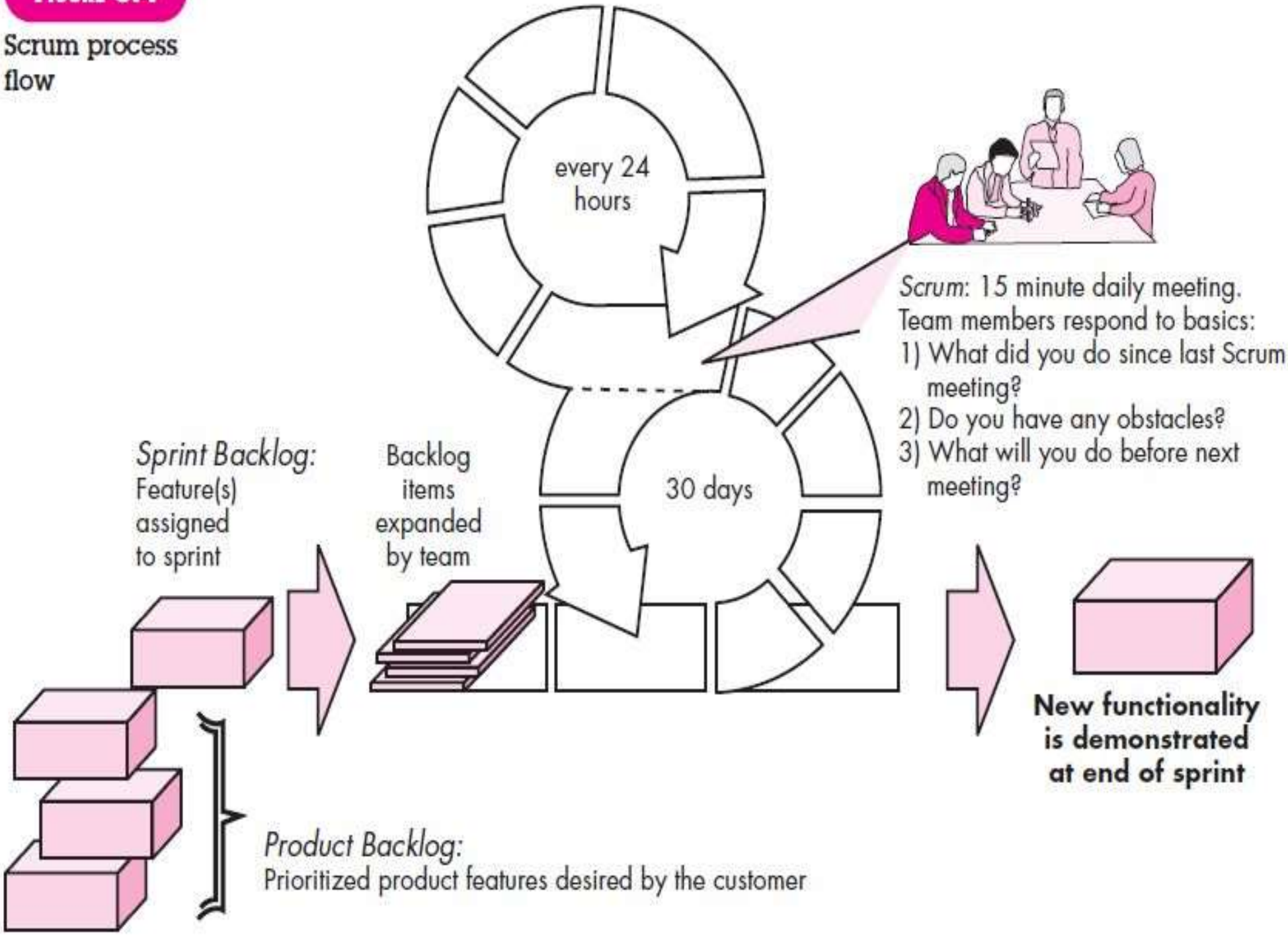
- 2001: publication “Agile Software Development with Scrum” by Ken Schwaber & Mike Beedle
- Successful appliance of Scrum in over 50 companies

  Founders are members in the Agile Alliance

Functionality of Scrum

FIGURE 3.4

Scrum process flow



Components of Scrum

Roles

- Product owner
- Scrum Master

Team

Process

- Sprint planning
- Sprint review
- Sprint retrospective

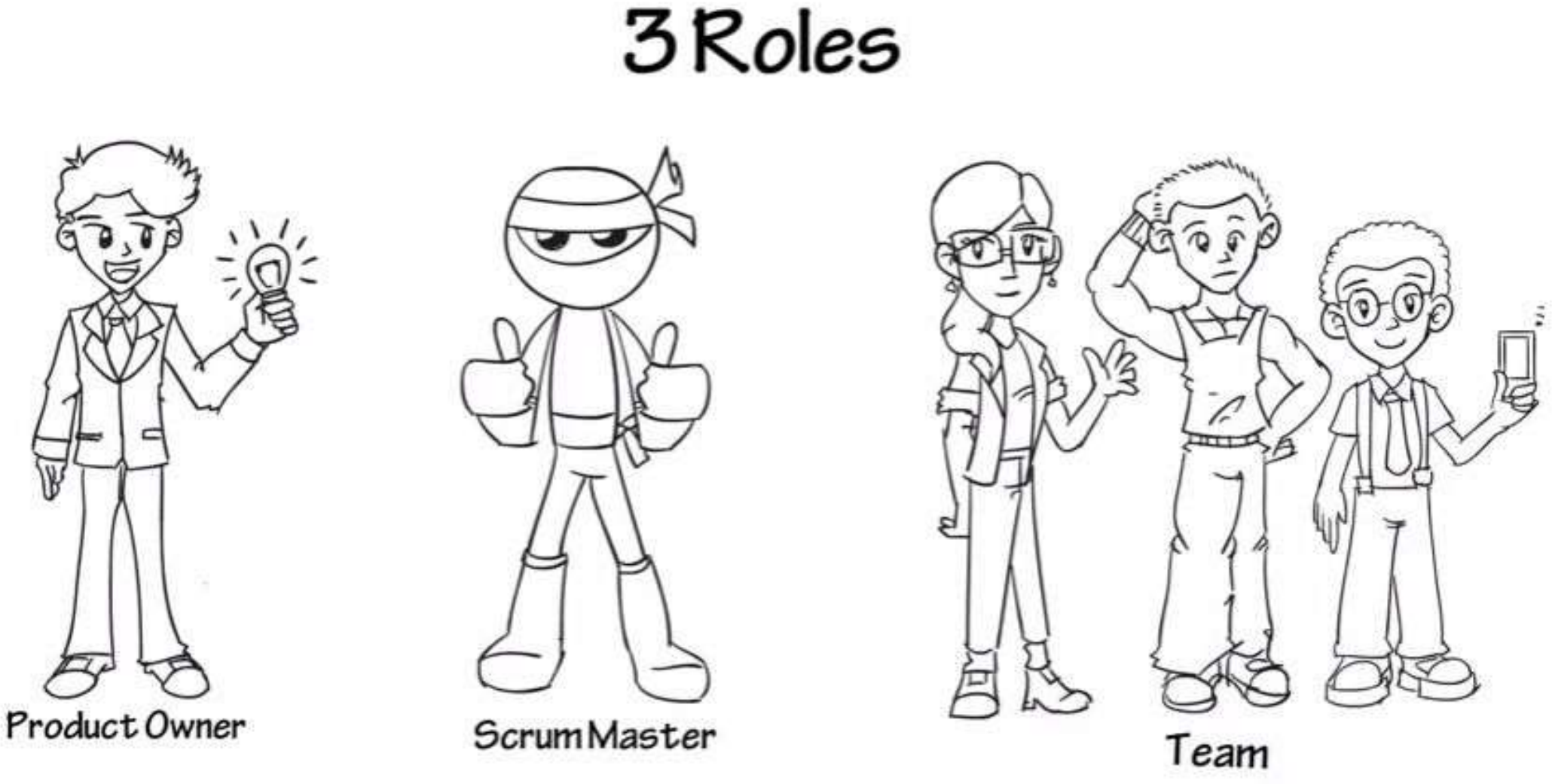
Daily scrum

Artifacts

- Product backlog
- Sprint backlog

Burndown charts

Scrum Roles



Product Owner

- Typically a product manager
- Represents the customers, users, and other stakeholders.
- Defines and documents the product features called “user stories”
- Prioritizes the features to be developed
- Controls the product backlog (prioritized list of product features)
- Plans and announces releases
- Provides a point of contact between the stakeholders and developers

Scrum Master

- Typically a Project Manager or Team Leader
- Responsible for the success of scrum
- Scrum values, practices and rules are enacted and enforced
- Scrum Master isn't a project manager because the team is self-organizing, so it guides itself.
- Conducts all daily scrums
- Main job is to remove impediments

The Scrum Team

- Typically 5-10 people
- Cross-functional (QA, Programmers, UI Designers, etc.)
- Members should be full-time
- Team is self-organizing
- Membership can change only between sprints

The Process

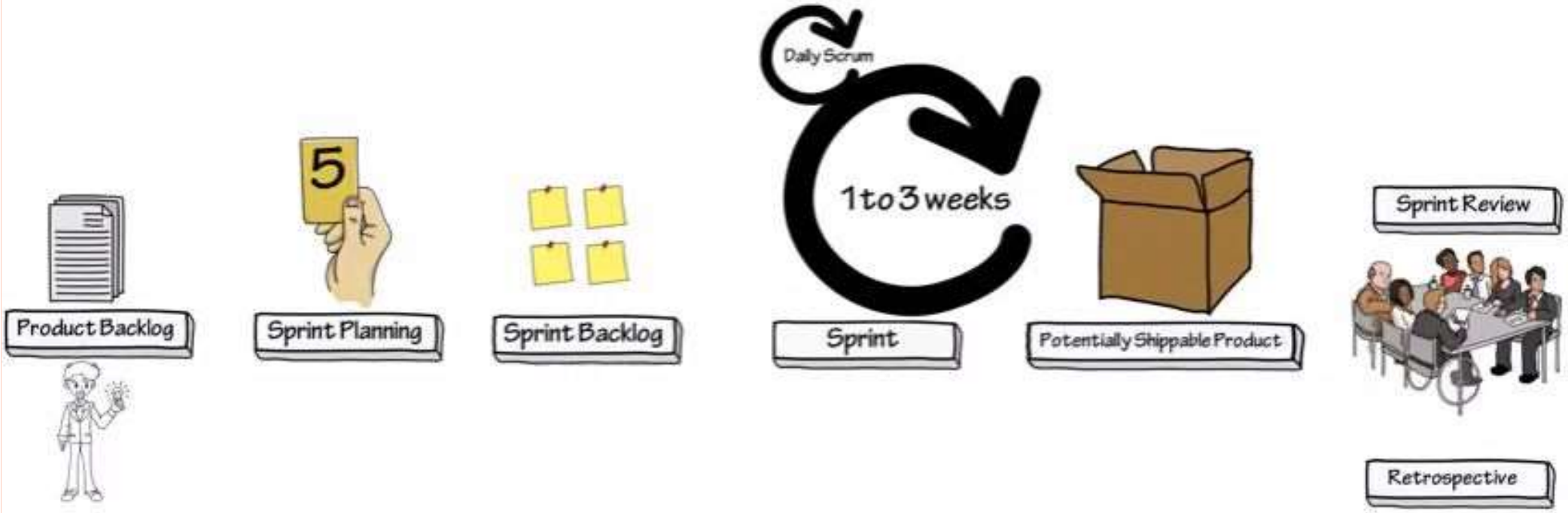
Sprint Planning Meeting

Sprint

Daily Scrum

Sprint Review Meeting

Sprint retrospective



Repeat this workflow for each sprint

Sprint Planning Meeting

- A collaborative meeting in the beginning of each Sprint between the Product Owner, the Scrum Master and the Team
- Takes 8 hours and consists of 2 parts
- 1st Part:
  - Creating Product Backlog
  - Determining the Sprint Goal.
  - Participants: Product Owner, Scrum Master, Scrum Team
- 2nd Part:
  - Creating Sprint Backlog
  - Participants: Scrum Master, Scrum Team

Pre-Project/Kickoff Meeting

- A special form of Sprint Planning Meeting
- Meeting before the begin of the Project

Sprint

- A month-long iteration, during which is incremented a product functionality
- NO outside influence can interference with the Scrum team during the Sprint
- Each Sprint begins with the Daily Scrum Meeting

Daily Scrum

- Is a short (15 minutes long) meeting, which is held every day before the Team starts working
- Participants: Scrum Master, Scrum Team
- 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
- Every Team member should answer on 3 questions
  - What did you do since the last scrum?
  - What do you hope to accomplish before the next scrum?
  - What obstacles do you see in your way?

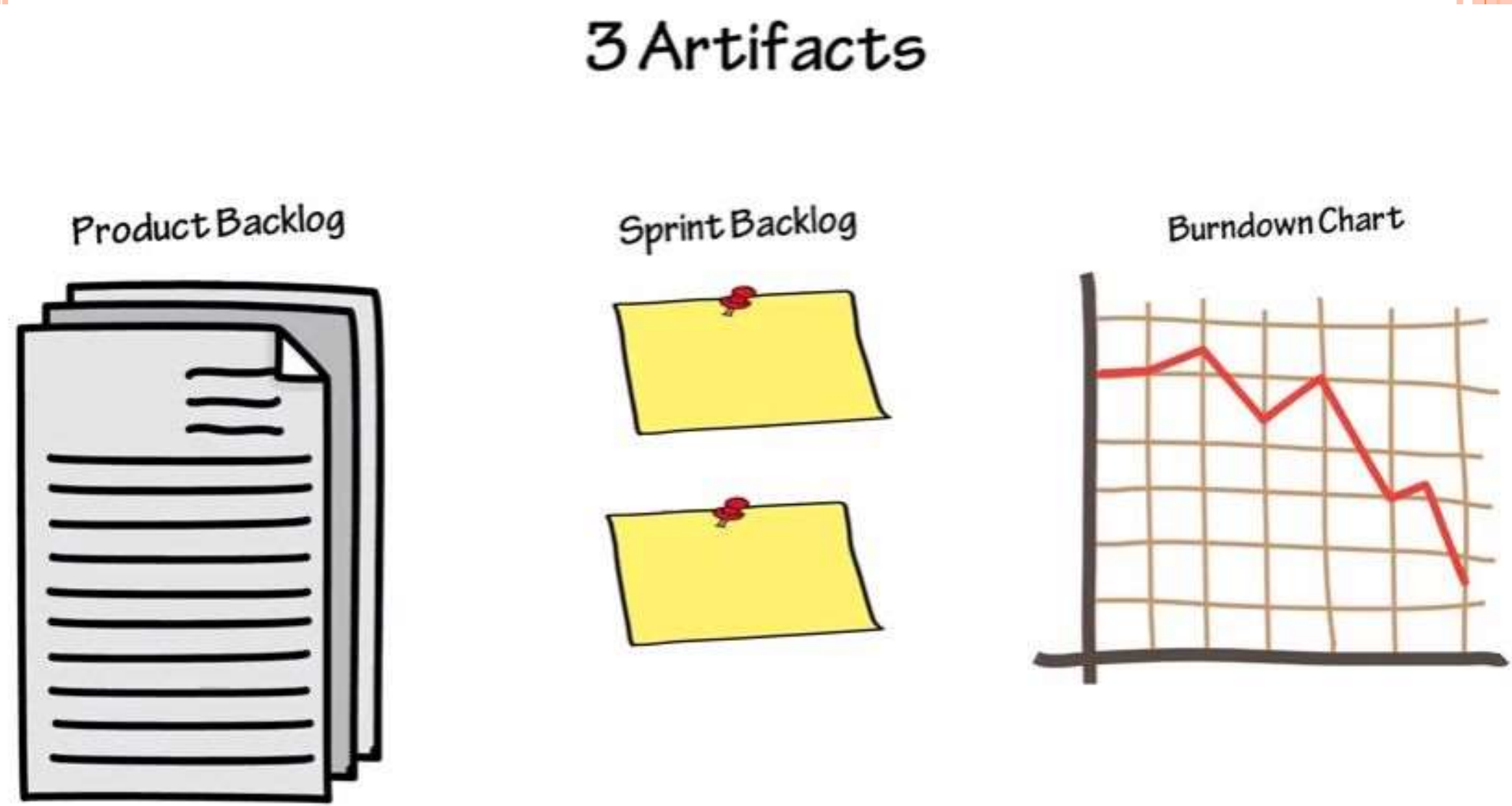
Sprint Review Meeting

- Is held at the end of each Sprint
- Business functionality which was created during the Sprint is demonstrated to the Product Owner
- Informal, should not distract Team members of doing their work

Sprint retrospective

- It occurs after the Sprint Review and prior to the next Sprint Planning.
- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - Scrum Master
  - Product owner
  - Team
  - Possibly customers and others

Scrum Artifacts



Product Backlog

- Requirements for a system, expressed as a prioritized list of Backlog Items
- Is managed and owned by a Product Owner
- Spreadsheet (typically)
- Usually is created during the Sprint Planning Meeting
- Can be changed and re-prioritized before each Sprint

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

Estimation of Product Backlog Items

- Establishes team's velocity
  - How much Effort a Team can handle in one Sprint
  - Velocity in a scrum is a measurement of how much the team gets work done in an iterations or sprint.
  - It is measured by

    Determining units of complexity.

    - Size-category
    - Story points
    - Work days/work hours
- Methods of estimation:
  - Expert Review

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
- No more then 300 tasks in the list
- If a task requires more than 16 hours, it should be broken down
- Team can add or subtract items from the list.
- Product Owner is not allowed to do it

A SPRINT BACKLOG

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Burn down Charts

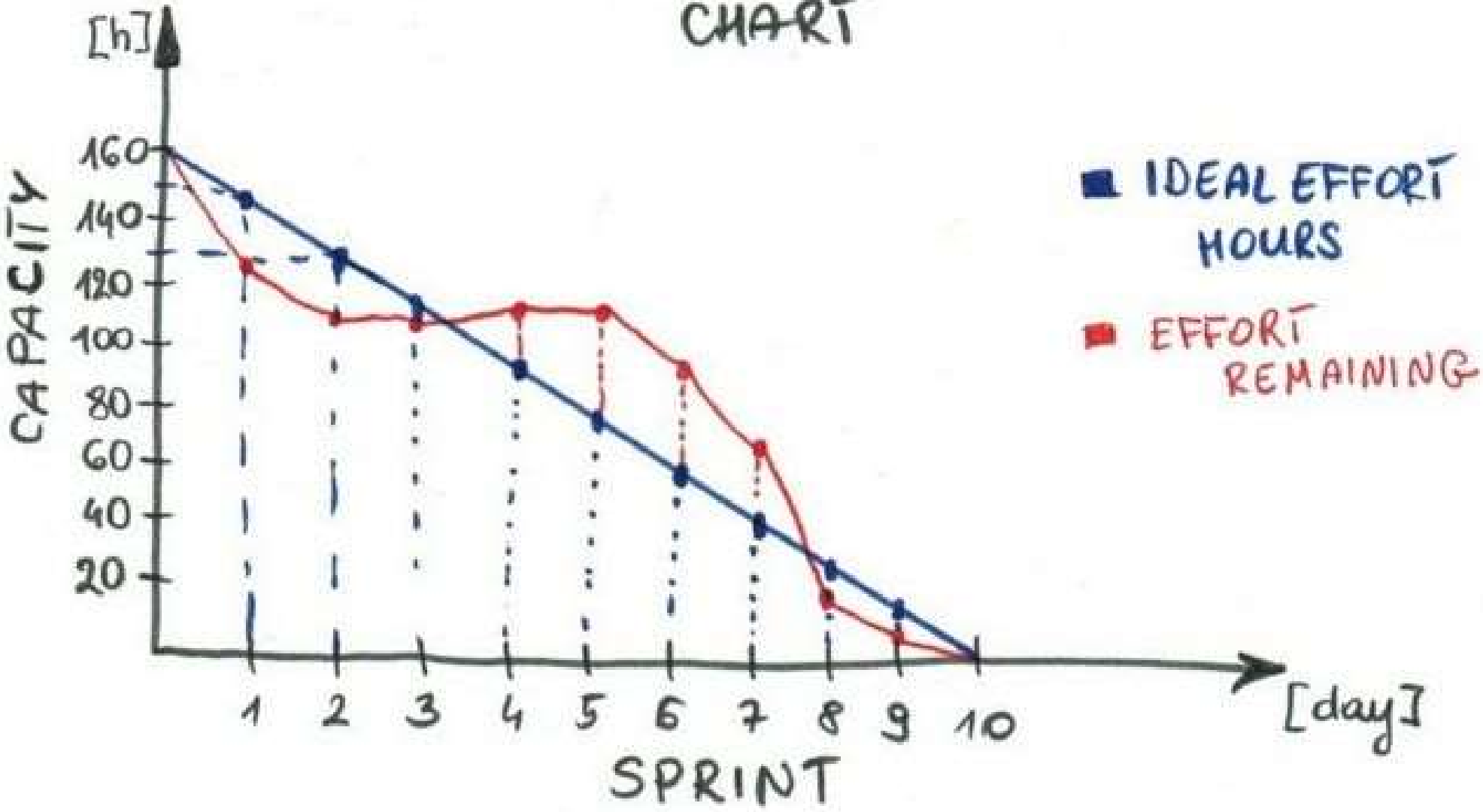
- Are used to represent “work done”.
- Are wonderful Information Radiators
- 3 Types:
  - Sprint Burn down Chart (progress of the Sprint)
  - Release Burn down Chart (progress of release)
  - Product Burn down chart (progress of the Product)
- X-Axis: time (usually in days)
- Y-Axis: remaining effort

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				

Sprint Burn down Chart

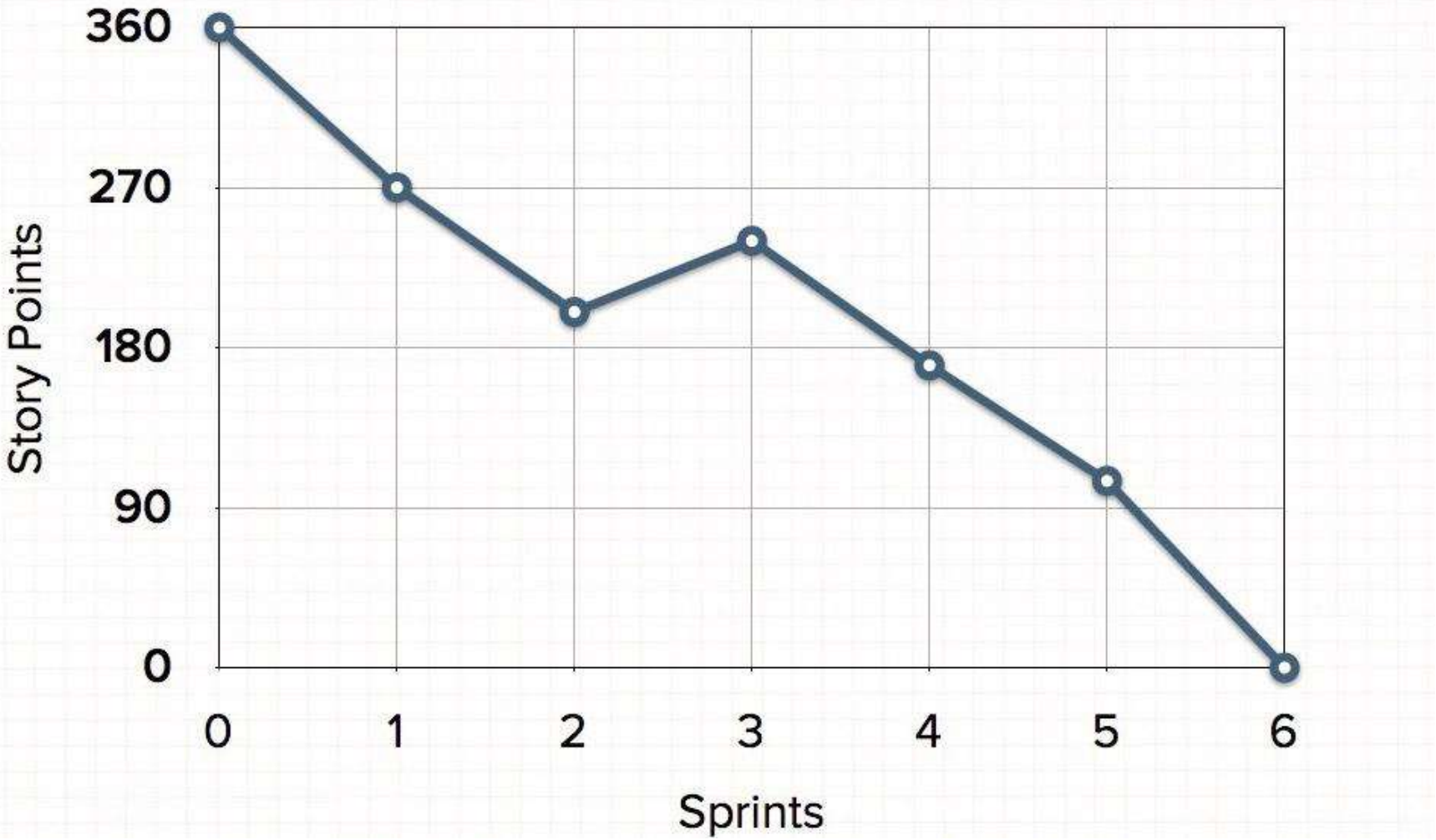
- Depicts the total Sprint Backlog hours remaining per day
- Shows the estimated amount of time to release
- Ideally should burn down to zero to the end of the Sprint
- Actually is not a straight line
- Can bump UP

SPRINT BURN-DOWN CHART



Release Burn down Chart

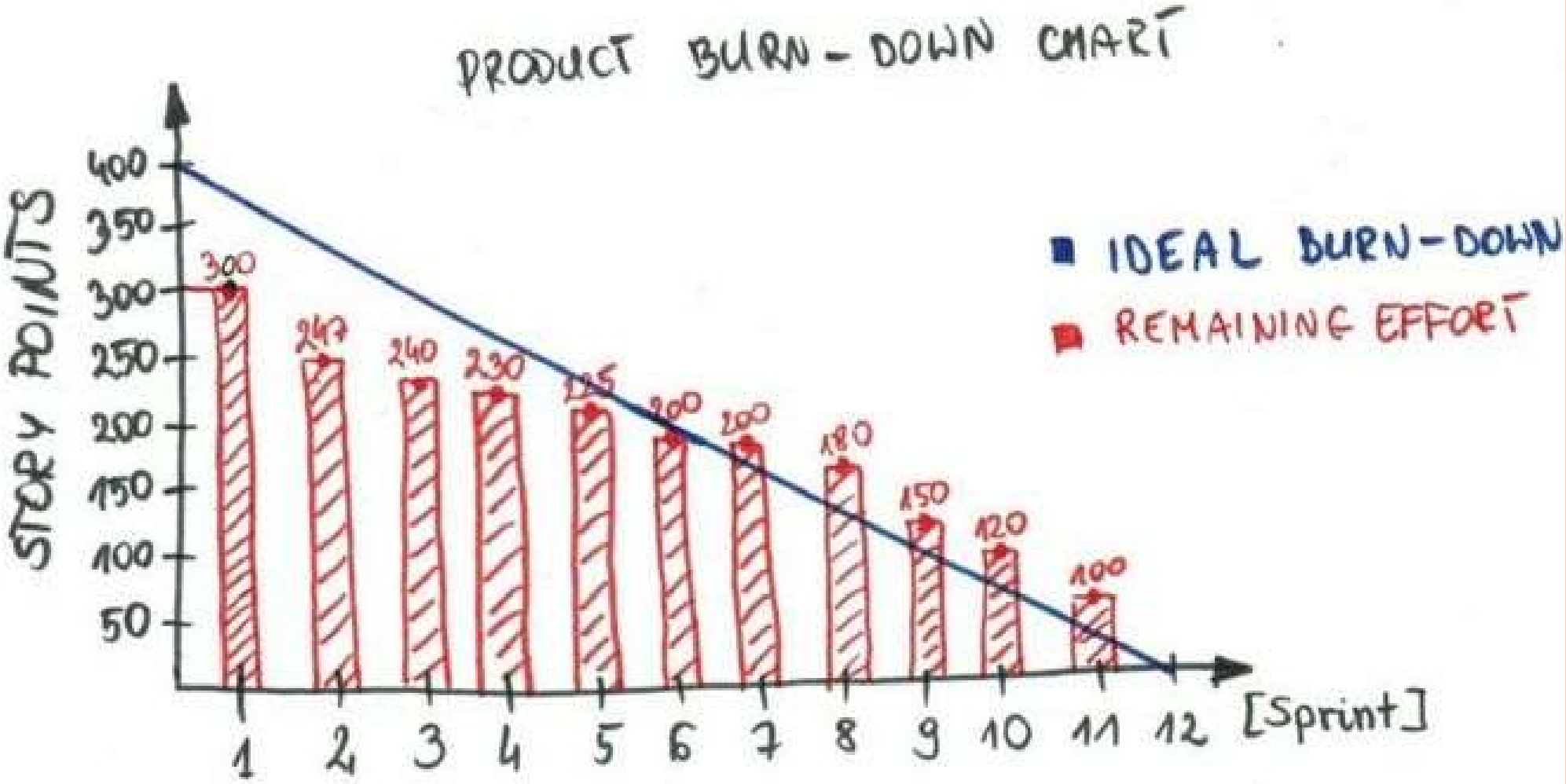
- Will the release be done on right time?
  - **X-axis**: sprints
  - **Y-axis**: amount of hours remaining
- The estimated work remaining can also burn up
- Progress on a Scrum project can be tracked by means of a release **burndown chart**.
- The Scrum Master should update the release **burndown chart** at the end of each **sprint**.
- The horizontal axis of the **sprint burndown chart** shows the sprints; the vertical axis shows the amount of work remaining at the start of each **sprint**.


360
270
180
90
0
Story Points
0
1
2
3
4
5
6
Sprints

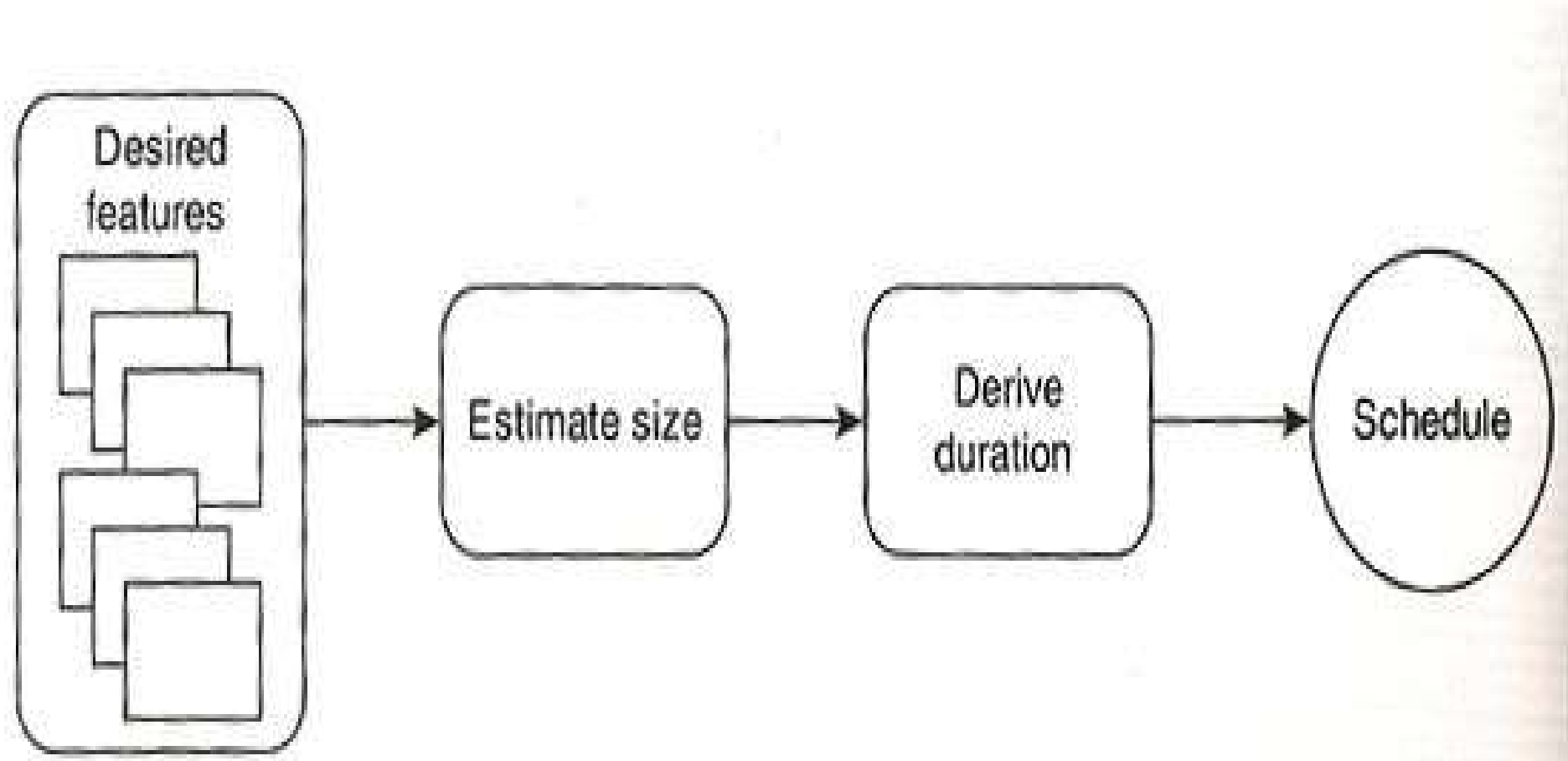
- **Story point** is a arbitrary measure used by **Scrum** teams.
- This is used to measure the effort required to implement a **story**.
- In simple terms its a number that tells the team how hard the **story** is.

Product Burn down Chart

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



Agile Projects Estimation



Estimating the duration of a project begins with estimating its size.

Agile Estimation

- **'Story Point'** is the measure of size (or complexity) of user stories in agile projects
- **'Ideal Days'** is the measure of effort in agile projects, which is the number of days a task will take if one person does that task without any interruptions
- **'Velocity'** is the sum of story points delivered by a team per cycle of iteration (sprint)
- Two commonly used estimation techniques for agile projects are:
  - Delphi Wideband
  - Planning Poker

Story Points

- A story point is an abstract measure of effort required to implement a user story.
- In simple terms, it is a number that tells the team about the difficulty level of the story.
- Difficulty could be related to complexities, risks, and efforts involved.
- Do not have any relevance to actual hours
- Complexity assessed & story points estimated in comparison with a baseline story
- Baseline story need not be the smallest one; it should be the one which all team members can relate to.

Criteria for estimating story points

1. Complexity

Effort needed to develop a particular feature

2. Risk

Unclear demand, uncertainty, dependence on third party

3. Amount of work

Actual work to complete the system

(consider existing system, expert, etc)

- For example, consider following user stories to estimate:
  1. A login screen to be developed
  2. A screen for entering customer data is required
  3. A mailing module to send periodic mails to the customers

- User story 2 – customer data entry screen may be taken as the baseline story, since all team members can clearly understand this functionality
- Assign a story point of 6 (any arbitrary number) for the user story 2 - customer data entry screen
- Complexity of all other user stories will be compared with the customer data entry screen & assigned a relative story point
- For every team, story size could mean different things depending on what baseline they chose
- Story points of different projects / teams are not comparable

- **Story points are a relative measure to represent complexity of user stories in comparison with a baseline story.** So, story points could be:
  - 1, 2, 3, 4, …, 10
  - 10, 20, 30, …, 100
  - 1, 2, 4, 8 and 16
  - 0, 1, 2, 3, 5, 8 and 13 (Fibonacci series)
  - S, M, L, XL, XXL
  - Cat, Dog, Lion and Elephant
- Team members should have clear understanding of the scale they use for story points
- Story points are converted to ideal days using the average velocity of previous sprints

Planning Poker(planning game)

- Most widely used estimation technique of Agile projects
- Planning Poker combines three estimation techniques – Wideband Delphi Technique, Analogous Estimation, and Estimation using WBS.
- used to **estimate** effort or relative size of user stories in Scrum.
- Simple, fun-filled and results in reliable estimates
- In Scrum, planning poker (also called Scrum poker) is a game you can **play to decide how much work a particular task might be.**

- **Steps:**
  1. Identify a team to do the estimation (4-6 senior team members)
  2. Each estimator is given a deck of cards.
  3. Each card has a valid estimate written on it
  4. Product owner / team member acts as moderator. He identifies a baseline story & explain it to the team and assigns a story point to the baseline story in consultation with the team
  5. Then, moderator reads the user story to be estimated.

6. Each estimator selects a card that's his or her estimate for the user story and puts down the card (with number down)
7. Cards are turned over only after everybody puts their chosen card down
8. Discuss the differences and re-estimate until estimates converge
9. Moderator intervenes when there is no convergence and clarify doubts if any for the estimators

10. The person who selected the lowest number and the person who selected the highest number are given a chance to share their view
11. People select the cards once again until there is some convergence
12. It rarely takes more than three rounds, but continue the process as long as estimates are moving closer together
13. Repeat steps from 5 to 12 for all user stories one after other

TOOLS FOR AGILE PROJECT MANAGEMENT

- **Zephyr**
  - Test management tool
  - It providing end-to-end solutions for agile teams of all sizes.
  - provide flexibility, visibility, and better release
- **JIRA**
  - Defect tracking tool
  - used for Agile testing as well as project management.
  - This tool is not only used for recording, reporting but also integrated with code development environment.

SoapUI

- **SoapUI**
  - It is an agile testing tool for service-oriented architectures (SOA) and REST.
  - Its functionality includes web service inspection, invoking, development, functional testing, and load testing.

- **The Apache Jmeter**
  - It is an open source agile performance testing tool.
  - It is used to load functional test behavior and measure performance of the website

- **Selenium WebDriver**
  - It is an automation agile testing tool. It aims to mimic the behavior of a real user, and as such interacts with the HTML of the application

- **Appium**
  - It is free to use open-source Agile testing tool. It is helpful for automating mobile web, iOS, and Android and hybrid applications. Native apps are those written using Android, iOS, or Windows SDKs


Bug Shooting


USERSNAP


PivotalTracker


qTest


TestRail


QMETRY
Total Test Management