

Software Engineering 1

Project Management

Arkadiusz Chrobot

Department of Computer Science, Kielce University of Technology

Kielce, November 3, 2025

Outline

- 1 Motto
- 2 Introduction
- 3 Scrum
- 4 Planning
- 5 Risk Management

Motto

”Question: How does a large software project get to be one year late?

Answer: One day at a time!”

Fred Brooks Jr. ”*The Mythical Man-Month: Essays on Software Engineering*”

Introduction

Software project management is a difficult task. There are many reasons for that. One of them is the nature of its product. The software has no physical form, it is pure information. That makes it challenging to follow the progress of the work. The conclusion is that software projects need special management methods.

In this lecture the widely used Scrum approach to project management is described and the planning and risk management methods in traditional and agile project management are discussed.

Scrum

Scrum was developed in the early 1990s by Ken Schwaber and Jeff Sutherland and was inspired by Toyota Production System, which in turn is based upon the works of Dr. William Edwards Deming. It is a *framework* for managing projects and solving complex problems. Its usage is not restricted to only software engineering. Detailed description of Scrum is given in the [► Scrum Guide](#) which is available on-line, free of charges. Please note, that Scrum **is not** a complete recipe, just a collection of guidelines that are intended to help organize the work on product. It has to be adapted by the project members to their needs.

Scrum Principles

Scrum is build around three principles:

Transparency The process of developing the product must be visible for the people who perform the work and the people who are waiting for the outcome. Important decisions are made according to the perceived state of three formal artefacts, that are described later. Transparency allows the project team to inspect their work.

Inspection The Scrum artefacts and progress toward goals of the project must be inspected frequently and thoroughly to detect any issues. Inspections allows for adaptation of the existing processes to the new problems.

Adaptation If an aspect of a process or a product doesn't have the required, assumed quality, then the entire process has to be adjusted as soon as possible, to prevent other losses.

Scrum Values

It is assumed that people who successfully practice Scrum gain:

Commitment to achieve the project goals and to support each other.

Focus to make the best possible effort toward these goals.

Openness to discuss their work and challenges.

Respect for the skills and independence of each others.

Courage to tackle complex problems.

Scrum Team

The Scrum Team should have no more than 10 members. It consists of:

Scrum Master is responsible for introducing and overseeing the Scrum in the team and removing any obstacles that may hamper the productivity of the team.

Product Owner is responsible for maximizing the value of product resulting from the work of the Scrum Team, manages the *Product Backlog*, communicates the *Product Goal*, represents the interests of stakeholders.

Developers are self-managed, self-organizing professionals with different set of skills needed to create a value to each *Sprint*.

Work Organization in Scrum

The work in Scrum is done in an iterative and incremental way. The iteration is called a *Sprint*. The maximal duration of a single Sprint is one month, but usually it is two weeks. It may be even shorter. No changes can be made when the Sprint lasts, that would endanger the *Sprint Goal* or decrease the product's quality. The *Product Backlog* may be refined and the scope of the Sprint may be renegotiated with the Product Owner, if the Developers gain some new knowledge of the increment in progress. The Sprint may be cancelled by the Product Owner, if its goal becomes obsolete.

Each sprint consists of four events: *Sprint Planning*, *Daily Scrum*, *Sprint Review* and *Sprint Retrospective*.

Sprint Events

During *Sprint Planning* the Scrum Team defines the *Sprint Goal* describing what value the Sprint should bring to stakeholders. They also decide on what items from the Product Backlog may be completed in the planned Sprint. Then the Developers determine what needs to be done to implement the selected items in order to turn them into valuable increment. The *Daily Scrum* is an event that allows the Developers to inspect the progress and adapt the *Sprint Backlog* if necessary. As its name suggests it is held daily and should not take longer than 15 minutes. The *Sprint Review* takes place at the end of the Sprint. Its goal is to inspect the outcome of the Sprint and decide on future adaptations. The results are presented to the most important stakeholders and the progress towards the Product Goal is discussed. For a month-long Sprint the Sprint Review should last no longer than 4 hours. For shorter Sprints the Review is also shorter. The *Sprint Retrospective* is also held at the end of the Sprint and its objective is to find ways for increasing the quality of the product and the effectiveness of the work. In this event the last Sprint is examined, all significant issues are identified, and remedies are deployed — they may be even added to the Sprint Backlog of the next Sprint.

Scrum Artefacts

There are three Scrum Artefacts:

Product Backlog is an ordered list of items that are needed to improve the product. An item describes the work that has to be done. The items may be added, removed and modified within this list. The modification is done by breaking down large items into smaller, more precise. Items are moved from the Product Backlog to the Sprint Backlog if they can be implemented/performed/finished within one Sprint. Developers decide that an item is fit for a Sprint as a result of sizing the items. The *Product Goal* defines the intended state of the final product and guides the work of the entire Scrum Team. The items in the Product Backlog adhere to the Product Goal.

Sprint Backlog contains the *Sprint Goal*, a plan for delivering the *Increment* and items chosen from the Product Backlog. The Sprint Backlog is created and managed by Developers, but in cooperation with the Product Owner. The *Sprint Goal* defines what should be achieved as the result of the Sprint.

Scrum Artefacts — Cont.

Increment is a concrete result that brings the project closer to the Product Goal. It must integrate with other Increments, be usable and thoroughly verified. Within a Sprint multiple Increments may be created and delivered to stakeholders. The Increment must meet the *Definition of Done*, that formally describes the quality measures for the product. If it isn't fulfilling the *DoD* then it is returned to the Product Backlog. The Scrum Team defines the *DoD*.

Scrum — Pros and Cons

Advantages of the Scrum:

- + allows the team to quickly react to requirement changes,
- + makes the progress more transparent than traditional methods,
- + incorporates the stakeholders' point of view into the process.

Disadvantages of the Scrum:

- it is quite abstract and may be difficult to introduce,
- it is "cultural biased".

Planning

In the Scrum and other agile methods there is almost no long-term planning. However, the progress of the work is followed and in each iteration (Sprint) the amount of the work that can be done is estimated.

On the other hand in the traditional management methods the *Project Manager* has to prepare the project schedule in advance and update it with the progress of the work.

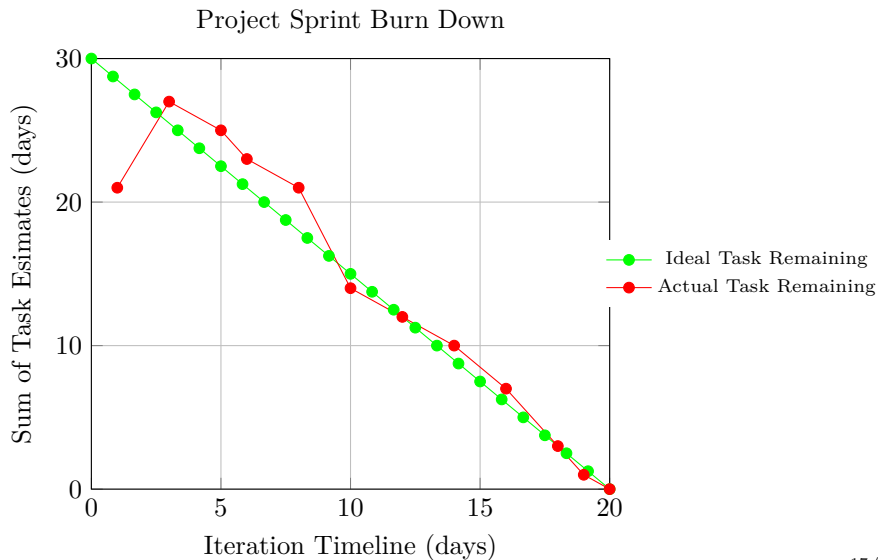
Planning Poker

The *Planning Poker* is an interesting approach to estimating work that needs to be done for each item selected from the Product Backlog to the Sprint Backlog. It takes a form of a card game, where each of the cards has assigned a number from the *Modified Fibonacci Sequence*: 0, $\frac{1}{2}$, 1, 2, 3, 5, 8, 13, 20, 40, 100. Those are *Story Points* that describe the effort needed to complete the task. The deck may also include cards with a cup of coffee (the player needs to make a break), ∞ symbol (the task is too large to estimate) and the ? symbol (the player doesn't know how to estimate the task). Each person that estimates the tasks has its own deck of cards. The *Moderator* takes one of the items from Product Backlog, describes it and then the rest of the players discussed its cost. After the discussion each player chooses one card, without showing it to the others. On a mark given by the Moderator the players show their cards. If all of them are the same then this is the estimate. If they are different, then the players with the extreme (lowest and/or highest estimates) are allowed to explain their decisions and the process repeats.

Burn Down Chart

To follow the progress of a single Sprint (iteration) or the entire project in the Scrum, the *Burn Down* and *Burn Up Charts* may be used. An example of a Burnt Down Chart is shown in the Figure 1. The green plot is the ideal burn down line that shows the amount of the work that should be done in each day of the Sprint to complete all tasks. The red plot shows the actual work done. Its shape indicates that sometimes the team was ahead of estimates, but usually struggled to keep the peace, however it finally managed to complete all required tasks. The Burn Up Chart is a mirror reflection of the Burn Down Chart, and it describes the work already completed.

Burn Down Chart



Planning in Traditional Management

In traditional management methods (like Waterfall), the Project Manager breaks down the work into a list of *tasks* and *milestones*. Each milestone ends a single task or a collection of tasks and brings some result. The time needed for completing a task is estimated. It shouldn't be shorter than one week and longer than eight weeks. In the first case there would be too many tasks in the project, and in the second the progress of work would be difficult to follow. The initial schedule is only an estimation, and it needs to be updated as the work progresses.

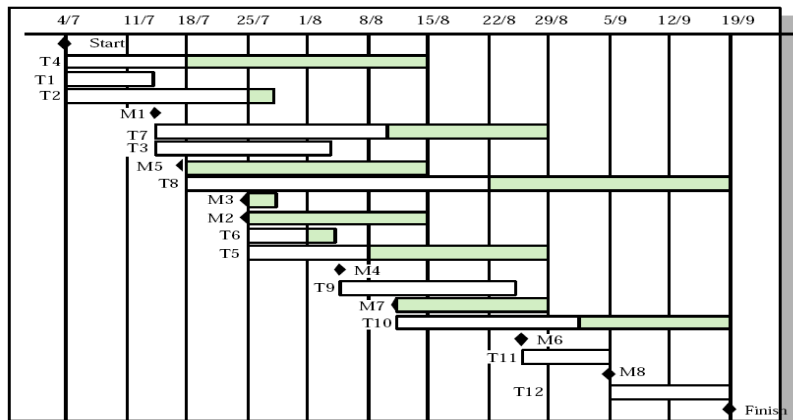
Tasks And Milestones

Task	Time (days)	Dependency
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2,T4 (M2)
T6	5	T1,T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3,T6 (M4)
T10	15	T5,T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

Gantt Chart

The *Gantt Chart* is a graphical notation that helps to visualize the schedule of the project. An example of such a chart, that corresponds to the table from the previous slide is in the next slide. The "diamonds" represent the milestones and the "bars" the tasks. Please note, that some tasks and milestones have a green "tail". It is a *time buffer* or *time spare* and means that the task completion or milestone achieving can be delayed by such amount (but not more) of time without consequences for the project deadline. Tasks and milestones without such a "tail" cannot be delayed at all if the delivery date should be kept.

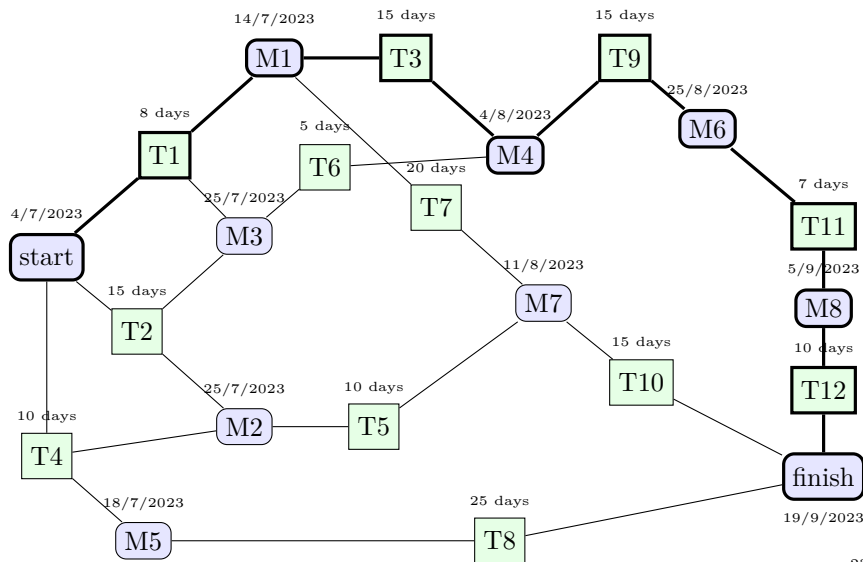
Gantt Chart



PERT Network Chart

The *PERT Network Chart* was developed for the US Navy Polaris Rocket Program. Originally PERT stood for *Program Evaluation and Review Technique* but latter it was renamed to *Project Evaluation and Review Technique*. It represents the project schedule as a directed graph in which vertexes are milestones and tasks. That way it is easier for the *Project Team* to spot dependencies between tasks. It also allows them to find the *Critical Path* — a sequence of tasks and milestones that determine the total time needed to complete the project and thus cannot be delayed.

PERT Network Chart



Risk Management

Risk is the probability that some undesired event will occur combined with the estimated result of that event. In software projects there are many sources of threats, like team frustration, staff rotation, bad management, wrong tools and estimates, etc. Threats may have negative impact on the project, the product, and even the company.

In agile methods, like Scrum, the feedback is provided very often, so the changes in the risk level and the threats are almost immediately visible, and the team can react quickly. In traditional methods the Project Manager has to assess the threats and risks first and prepare some plans to counter them.

Risk Assessment

Initially the Project Manager has to discover all likely threats to the projects, assess the probability of their occurrence and estimate their impact on the project. The probability can be descriptive, like "very small", "small", "average", "big", "huge". The impact can be described as "negligible", "tolerable", "serious" and "catastrophic". These two variables allow the Project Manager to find the most dangerous threats and prepare plans for countering them.

Risk Countering Plans

Three types of risk countering plans may be prepared by the Project Manager:

Avoidance Strategies their objective is to decrease the probability of threat occurrence,

Minimization Strategies their goal is to decrease the impact of the threat,

Emergency Plan such a plan is deployed when the worst threats materialize.

Threat Monitoring

As the project progresses some threats become unlikely and the probability of some others increases. That's why the Project Manager has to constantly monitor the sources of potential threats and adjust the threat countering plans to the current situation.

Questions

?

THE END

Thank You for Your attention!