

Dr. Michael Eichberg

Software Engineering

Department of Computer Science

Technische Universität Darmstadt

Software Engineering

Software Project Management



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- **The product is intangible**

(intangible =dt. nicht greifbar)

Project managers cannot (easily) see progress and have to rely on others to produce the documentation needed to review progress. (To alleviate this software, deliver working software frequently.)

- **Agile software development processes often clash with the requirements from other domain (e.g., business)**

- **Large projects are often "one-off" projects**

These makes estimations and anticipations of problems (very) hard. Rapid technological changes also render previous experience obsolete.

- **Proposal writing**

=dt. Angebotserstellung / Antragsstellung

I.e., an important skill that software project managers have to have is to communicate effectively both orally and in writing.

- **Project planning and scheduling**

planning =dt. Planung, Planungsvorbereitung

scheduling =dt. Terminierung, Anberaumung eines Termins

- **Project cost calculation**

=dt. Projektkostenkalkulation

- **Project monitoring and reviews**

- **Personnel selection and evaluation**

- **Report writing and presentations**

Project Planning is an iterative process.

- A plan drawn up at the start should be used as the driver for the project; this should be the best possible plan given the available information
- The plan evolves as the project progresses.

- **Project Plan** (dt. Projektplan) (following slides...)
- **Quality plan** (dt. Qualitätssicherungsplan)
Describes the quality procedures and standards that will be used.
- **Staff development plan** (dt. Personalentwicklungsplan)
Describes how the skills and experience of the project team members will be developed.
- **Configuration management plan**
Describes the configuration management procedures and structures to be used.
- ...

1. **Introduction**

Objectives of the project and its constraints (time, budget,...).

2. **Project organization**

Organization of the development team, the involved people and their roles.

3. **Risk analysis**

Possible project risks, their likelihood and risk reduction strategies.

4. **Hardware and software resource requirements**

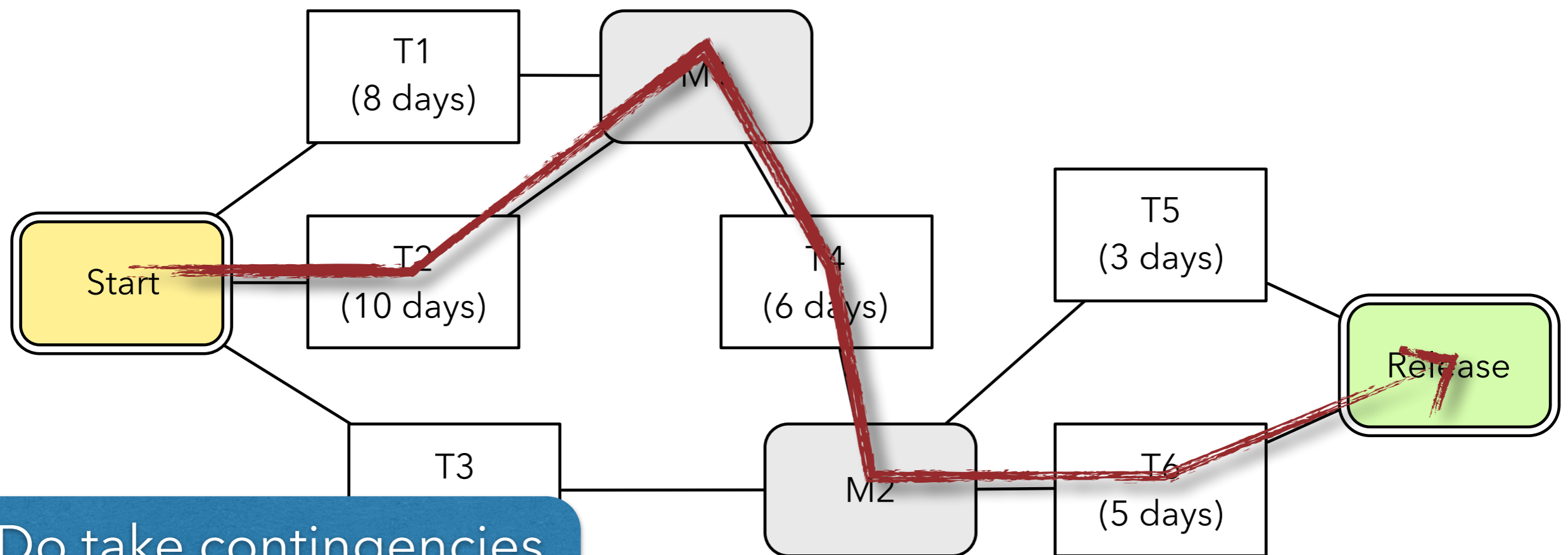
The hardware and support software required to carry out the project.

5...(next slide)

- ...(previous slide)
- **Work breakdown** (=dt. Arbeitsaufteilung)
Sets out the breakdown of the project into activities and identifies the milestones and deliverables associated with each activity.
- **Project schedule**
(Sometimes mistakenly called "Project Plan.")
Dependencies between activities, estimated time required to reach each milestone and the allocation of people to activities.
- **Monitoring and reporting mechanisms**

Project Schedules can be illustrated using **activity networks** and / or gantt charts.

- A milestone has to be a concrete, verifiable goal

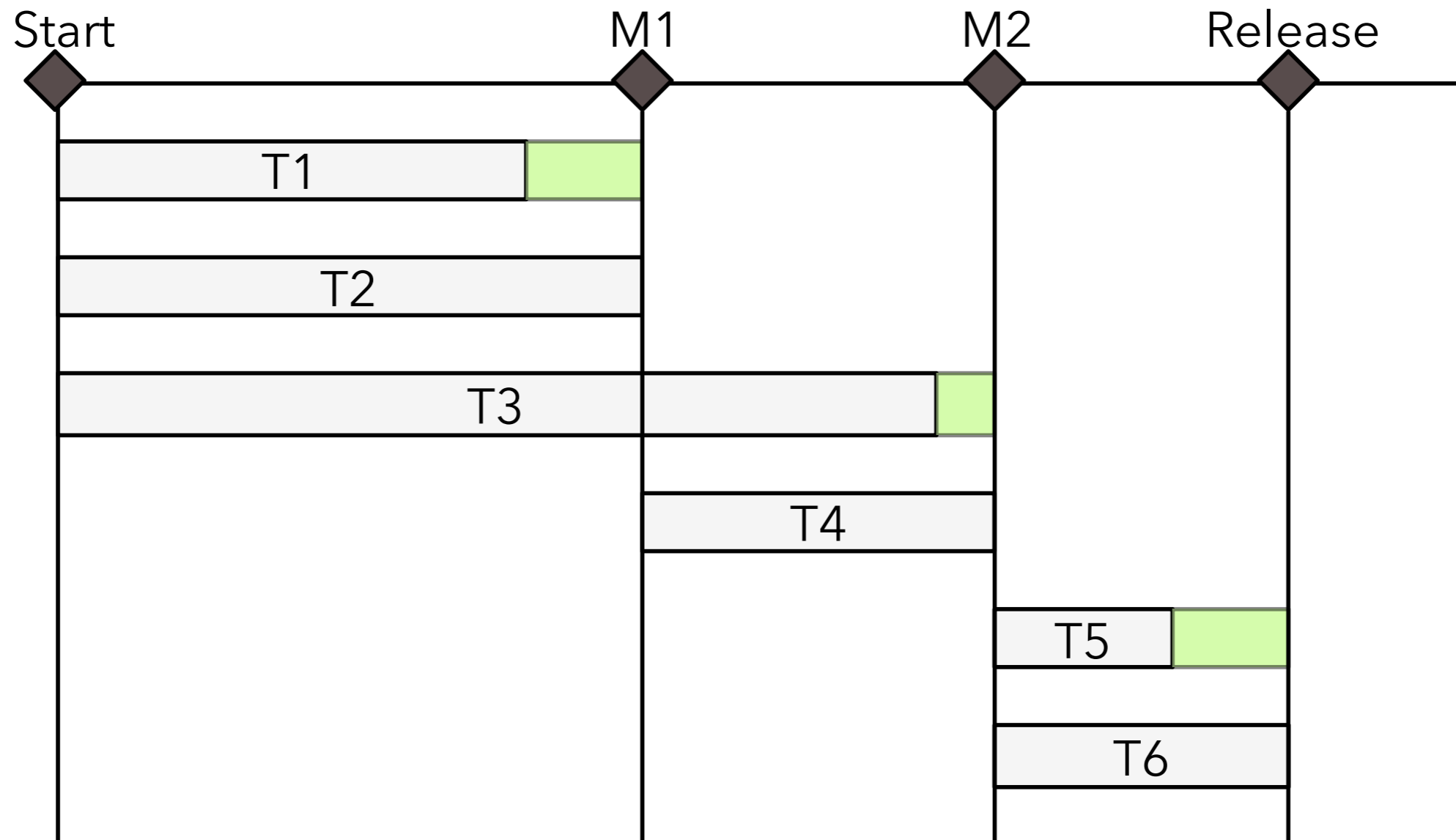


Do take contingencies
(dt.= Eventualitäten) into
account.

Critical Path

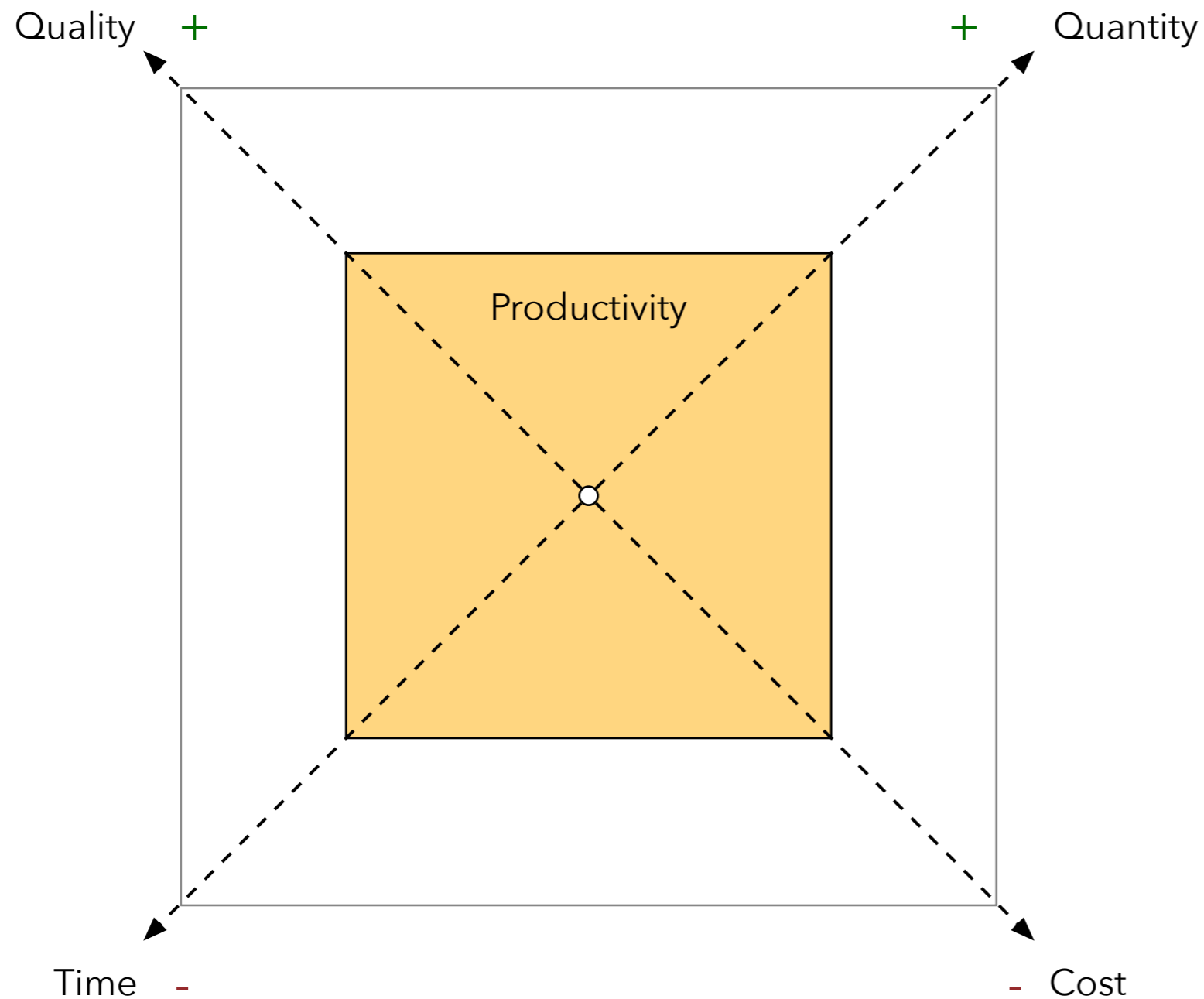


Project Schedules can be illustrated using activity networks and / or **gant charts**.



Sneed's "Devil's Square"

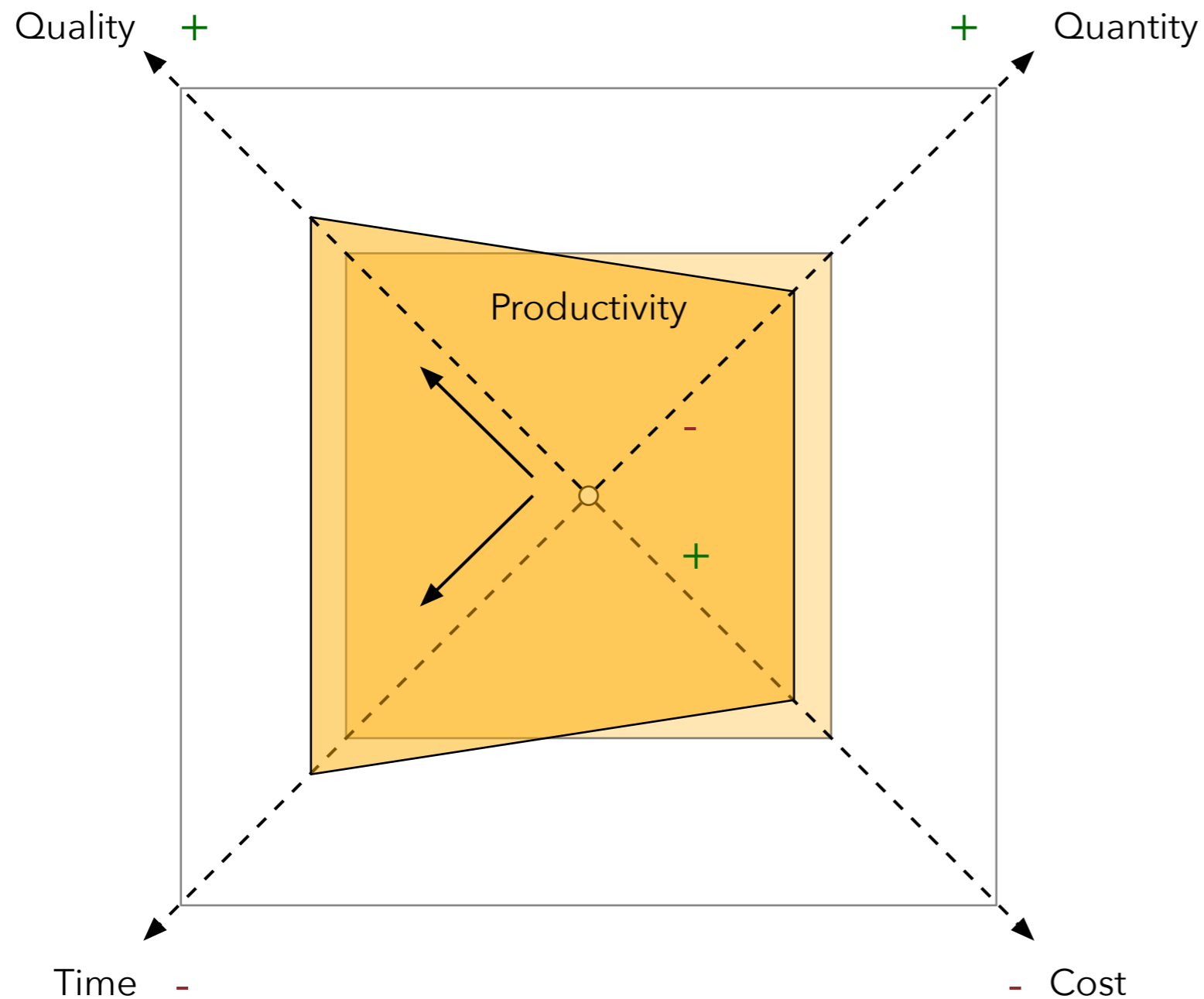
The quality and the quantity of the software is measured as well as the time and costs it takes to complete the project.



In the **short term**, the productivity (the "orange" area) is fix.



Sneed's "Devil's Square"

If the quality of the software should be increased and the time should be reduced, the costs will rise and the number of features has to be reduced.



In the **short term**, the productivity (the "orange" area) is fix.

The goal of this lecture is to enable you to systematically carry out small(er) commercial or open-source projects.

-  You should now have a good understanding of the specifics of software projects.
-  You should have a very rough idea of some tasks that need to be carried out when managing software projects.