## Project

**Project**

- **Temporary activity** that is characterized by
  - having a start date,
  - specific objectives and constraints,
  - established responsibilities,
  - a budget and schedule, and
  - a completion date.

If the objective of the project is to develop a software system, then it is sometimes called a software development project or software engineering project. (R. H. Thayer, 1997)

We could refine our earlier definition as follows: a project is **successful** if and only if
- started at start date,
- achieved objectives,
- respected constraints,
- adheres to budget and schedule,
- stops at completion date.

Whether, e.g., objectives have been achieved can still be subjective (→ customer/user happy).

## Software Project

### Characteristics:

- **Duration** is limited.
- Has an **originator** (person or institution which initiated the project).
- The **project owner** is the originator or its representative.
- The **project leader** reports to the project owner.
- Has a **purpose**, i.e. pursues a bunch of goals.
  - The most important goal is usually to create or modify software; this software is thus the result of the project, the **product**.
  - Other important goals are extension of know-how, preparation of building blocks for later projects, or utilisation of employees.

The project is called **successful** if the goals are reached to a high degree.

- Has a **recipient** (or will have one).
  - This recipient is the **customer**.
  - Later users (conceptionally) belong to the customer.

- **Links** people, results (intermediate/final products), and resources.

The **organisation** determines roles of and relations between people/results/resources, and the **external interfaces** of the project. (Ludewig & Lichter, 2013)
Unforeseen difficulties and problems in projects are not exceptional but usual. Therefore, project management needs to constantly "screen the horizon for icebergs," and, when spotting one, react timely and effectively. In other words:

- **Risk Management**: is not a matter of "problems with systematic or 'Extremely Improbable'" (AC 25.1309-1). The (infamous) Waterfall Model for developers requires: "Average Probability per Flight Hour for Catastrophic Failure Conditions of Avionics: Code and Fix". Costs and Deadlines may differ from plans and compared to the plans may prove unacceptable. Risk assessment is based on market, project goals or results. Whether it will occur, cannot be surely predicted.

\[ K = p \times C \]

Project management activities are not risks, but environment conditions to be dealt with.

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Processes</td>
<td>Common Activities of Project Management</td>
</tr>
<tr>
<td>Costs and Deadlines</td>
<td>Leading and Motivation of Employees</td>
</tr>
<tr>
<td>Risk Management</td>
<td>Communication</td>
</tr>
</tbody>
</table>

Common Activities of Project Management

- **Recognition and Fighting Difficulties as Early as Possible**
- **Assessment and Control**
- **Planning**
- **Common Activities**
- **Leading and Motivation of Employees**

**Note**: mistakes in planning can be hard to resolve. Without plans, a project cannot be managed. The result of a project will have to be attractive to project participants (project owner, customer, developers, administration). Work results and project progress have to be assessed, which is useful for later projects, knowledge acquisition and re-usable components, e.g., on market, software delivery. Have a main and general goal.

- **Assessment and Control**
- **Planning**
- **Communication**
- **Leading and Motivation of Employees**

**Excursion: Risk and Risk Value**

Quick Excursion: Risk and Risk Value

- **Risk Value**: a problem, which did not occur yet, but on occurrence threatens important conditions, i.e., the project, i.e. the project may prove unacceptable. Risk value is composed of probability of problem occurrence, \( p \), and cost incidence probability, \( C \). Risk value expresses a problem, which has not yet occurred, but on occurrence threatens important conditions.

- **Risk Value**: a problem, which did not occur yet, but on occurrence threatens important conditions. **Risk Value** is composed of probability of problem occurrence, \( p \), and cost incidence probability, \( C \). Risk value expresses a problem, which has not yet occurred, but on occurrence threatens important conditions.
The concept of roles cont'd

A role has, and necessary skills and capabilities. Each role is assigned to one person, which often represent many legal persons, e.g. a company. The same legal person may act as “Customer” and “Developer” in the same project. For example, there may be different assignments for different phases.

In a software project, at each point in time, there is a set of roles, e.g. R = \{ ana, tst, prg, mgr \}. Given a set P of people, e.g. P = \{ user, tester, programmer, developer \}, each with \( r(p) = R \) is defined such that each person has (at least) the skills and capabilities required by role. \( r(p) \) assigned to role \( r \) has (at least) the skills and capabilities required by role.

Recall that a role is a set of rights and responsibilities. Each role is defined by the set of rights and responsibilities, which often represent many legal persons. For example, a role may be defined by the set of rights and responsibilities of a company. The same legal person may act as “Customer” and “Developer” in the same project. For example, there may be different assignments for different phases.

Useful and common roles

- Project Manager (pm): leads the project and is responsible for reporting unforeseen problems to the project manager. For example, the project manager may be responsible for reporting unforeseen problems to the project manager. The project manager has the right to change the code right.
- Designer (d): designs the software and is responsible for respecting coding conventions. For example, the designer may be responsible for respecting coding conventions.
- Tester (t): tests the software and is responsible for raising issue reports. For example, the tester may be responsible for raising issue reports.
- Programmer (p): writes the software and is responsible for closing issue reports. For example, the programmer may be responsible for closing issue reports.
- Architect (a): designs the architecture and is responsible for quality control. For example, the architect may be responsible for quality control.

For each role = \( r \), \( r(p) \) must hold the rights and responsibilities of role. For example, the project manager has the right to change the code right. The designer and tester have the right to test the software.

The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, creating a design, and implementing the code. The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, creating a design, and implementing the code.

The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, creating a design, and implementing the code. The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, creating a design, and implementing the code.
legal claims not reaching a defined milestone as planned can lead to development contract milestones can be part of the
• define (customer involved) milestones.

ends when it is no longer available for use

IEEE 610.12 (1990): If necessary and conceived— The period of time that begins when a system is available for use and ends when the software is no longer

— The period of time that begins when a system is

— The period of time that begins when a software product is

— The period of time that begins with the

The

A phase — (1) A period of time during which a set of events is completed. [...]

Cycle and Life Cycle

Milestones, Deadlines

Formal methods: verify code implements design.

Implementation: the needed modules using the chosen programming language(s). Done if tested as needed, and ready for integration.

For 1.3.2 Integrating modules into the module sets: The set of all modules is a superset of the module sets of the lower-level modules.

Component interfaces as precise as possible to enable concurrent development and seamless integration.

Depends on maintenance project — any processing of artefacts, manually or automatic; solves tasks.

Component interfaces as precise as possible to enable concurrent development and seamless integration.

Common activities in order to develop or adapt software.

• Requirements specification
• Analysis: understand the problem, assess whether/in how far software can be used to solve it.
• Design, specifica- tion of modules
• Architecture artefacts: specify the component interfaces, composition, and hierarchy of the system.
• Code implements design
• Code refines to spec: verify properties of these results, which need to be achieved.
• Software is actually implemented: code implements design.
• Deployment, operation, and maintenance
• Integration, test, approval

In particular: has responsibility for all products emerging during a development process. Is processed by artefact → has responsibilities and rights, needs skills and capabilities.

Maintenance activities: all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.

— has responsibilities and rights, needs skills and capabilities.

— all documents, evaluation protocols, software modules, etc., are kept in their initial state.
A decision point corresponds to a milestone, delimits phases, and sometimes creates artefacts (in a certain state), artefacts — special case of activity: a decision is made based on decision point artefacts, creates/modifies activities, depends on ...
Building Blocks Can Be Arbitrarily Complex

How the Plan for Creating is More Easily Created

May Have Been Created...

...and May Participate in the Plan...
There is not universally agreed distinction.

In the literature, • (2010) propose to distinguish:

1. Process descriptions can be found at the process model — a model comprising definitions of processes in a life cycle.
3. Procedure and process models

• (2013) propose to distinguish:

1. Process model
2. Process vs. Procedure Model

Process vs. Procedure Models

In (Software) Project Management and Responsibilities, Organisational structure (i) and (ii) propose to distinguish:

- Process vs. Process Model
- Procedure vs. Procedure Model

Process Description and Reference Model

Process description — a documented expression of a set of activities performed to achieve a given purpose. NOTE: A process description provides an operational definition of the major components of a process.

V-Modell, RUP, XP (90s/00s).


Process vs. Procedure Model

Roles, Artefacts, Activities

- Costs and Deadlines
- Excursion: Risk
- Common Activities
- By the Way: Process Model of Tutorials

By the Way: Process Model of Tutorials

Common Activities

- Excursion: Risk
- Costs and Deadlines
- By the Way: Process Model of Tutorials
Anticipated Benefits of Process Models

- "economy of thought" — don’t re-invent principles.
- Quantification, reproducibility — one can assess the quality of how products are created (→ CMMI).
- Identify weaknesses, learn from (bad) experience, improve the process.
- Fewer errors — e.g., testing a module cannot be forgotten because the “ready” decision point depends on module with “test passed” flagged.
- Clear responsibilities — fewer “I thought you’d fix the module!”
- Process model-ing is easily overdone — the best process model is worthless if your software people don’t “live” it.
- Before introducing a process model, understand what you have, understand what you need.
- Process-model as much as needed, not more (→ tailoring).
- Assess whether the new/changed process model makes matters better or worse (→ metrics).
- Note: customer may require a certain process model.

Tell Them What You’ve Told Them...