A Software Life-cycle Model

Which part will we talk about today?

- Requirements
- System Design (Architecture, High-level Design)
- Module Design (Program Design, Detailed Design)
- Implementation of Units (classes, procedures, functions)
- Unit testing
- System Testing (Integration testing of modules)
- Module Testing (Integration testing of units)
- Acceptance Test (Release testing)
- Validation Requirements, Verify Specification
- Verify System Design
- Verify Module Design
- Verify Implementation
- Maintenance

Part I
Introduction to Project Management

Part II
Time- and Resource Planning

Part III
Risk Management

Part IV
Communication and Documentation

Part I: Introduction to Project Management
Part II: Time- and Resource Planning
Part III: Risk Management
Part IV: Communication and Documentation
Part I
Introduction to Project Management
Typical properties of a project?

- Start and stop
- A budget
- Goal
- A single-time occurrence

Part I
Introduction to Project Management

Part II
Time- and Resource Planning

Part III
Risk Management

Part IV
Communication and Documentation

Who are involved in a software project?

- Customer
  - Pays for the system
  - Provides the system
- User
  - Uses the system
- Development Organization
- Supplier

A person or organization with a major interest in the project outcome.
Dependent project parameters

- Calendar Time
- Resources
- Features
- Quality

SMART Goals

- **Specific**
  Must be straightforward and answer the questions: **What** will you do? **Why** is it important?

- **Measurable**
  If you cannot measure it, how do you then know if the goal is reached or not?

- **Agreed Upon**
  Agreed upon with all stakeholders (e.g. customer, user etc.)

- **Realistic**
  Possible with the current resources, knowledge and time. You must be both willing and able to do it.

- **Timely**
  A clear time frame for the goal.

Note that there exists other similar versions the definition of SMART goals.
Part II
Time- and Resource Planning

A project

Lots of things to do...
A project

Work breakdown

Part I
Introduction to Project Management

Part II
Time- and Resource Planning

Part III
Risk Management

Part IV
Communication and Documentation

A task or an activity

Examples:
- Implement encryption module
- Interview users
- Design user-interface prototype

Task (or activity)

Duration, e.g. 10 days
Dependencies between tasks

Task1 and Task2 are precursors (predecessors) to Task3

Part I
Introduction to Project Management

Part II
Time- and Resource Planning

Part III
Risk Management

Part IV
Communication and Documentation

Tool Support

Microsoft Project

OpenProj

IDA’s MSDN Academic Alliance
(see “Resources” on course page)
### Tasks, Duration, and Dependencies

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Task</td>
<td>30 days</td>
<td>May 01, 2021</td>
<td>May 30, 2021</td>
<td></td>
</tr>
<tr>
<td>Analysis</td>
<td>2 days</td>
<td>May 03, 2021</td>
<td>May 05, 2021</td>
<td></td>
</tr>
<tr>
<td>Market study</td>
<td>2 days</td>
<td>May 07, 2021</td>
<td>May 10, 2021</td>
<td></td>
</tr>
<tr>
<td>User requirements specification</td>
<td>1 day</td>
<td>May 12, 2021</td>
<td>May 13, 2021</td>
<td></td>
</tr>
<tr>
<td>User interfaces prototype</td>
<td>1 day</td>
<td>May 14, 2021</td>
<td>May 15, 2021</td>
<td></td>
</tr>
<tr>
<td>Project plan</td>
<td>2 days</td>
<td>May 17, 2021</td>
<td>May 19, 2021</td>
<td></td>
</tr>
<tr>
<td>User requirements specification ii</td>
<td>2 days</td>
<td>May 20, 2021</td>
<td>May 22, 2021</td>
<td></td>
</tr>
<tr>
<td>Interface requirements</td>
<td>2 days</td>
<td>May 23, 2021</td>
<td>May 24, 2021</td>
<td></td>
</tr>
<tr>
<td>User interfaces prototype ii</td>
<td>1 day</td>
<td>May 25, 2021</td>
<td>May 26, 2021</td>
<td></td>
</tr>
<tr>
<td>User interfaces beta</td>
<td>1 day</td>
<td>May 27, 2021</td>
<td>May 28, 2021</td>
<td></td>
</tr>
<tr>
<td>Deployment</td>
<td>3 days</td>
<td>May 29, 2021</td>
<td>June 01, 2021</td>
<td></td>
</tr>
<tr>
<td>Deployment beta</td>
<td>2 days</td>
<td>June 02, 2021</td>
<td>June 04, 2021</td>
<td></td>
</tr>
<tr>
<td>Deployment release</td>
<td>1 day</td>
<td>June 05, 2021</td>
<td>June 06, 2021</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>5 days</td>
<td>June 07, 2021</td>
<td>June 12, 2021</td>
<td></td>
</tr>
</tbody>
</table>

#### Milestone and Tollgate

- **Milestone**: Verify internal sub-goal fulfillment
  - Properties of a SMART goal

- **Tollgate**: External decision point
  - E.g., after a pre-study phase, the customer decides if the project should continue or not.
Critical Path, Slack and Real time

Available time = Slacktime + Real time

Part I
Introduction to Project Management

Part II
Time- and Resource Planning

Part III
Risk Management

Part IV
Communication and Documentation

Critical Path
Slack (float) time
Real time (estimated)

Resource Planning

Who is going to "do" the task and with what?

Resource planning
A key to success - buffer time

Buffer Time

Time

Internal Deadline

External Deadline

To who should you communicate the deadlines?

Part I
Introduction to Project Management

Part II
Time- and Resource Planning

Part III
Risk Management

Part IV
Communication and Documentation

Effort Estimation in reality

How long time does it take for you to implement the encryption layer?

Sam
the seller

Harry
the hacker

No idea. I have never done this before... I wonder if it is even possible.

8 months +/- 2 months
Prioritization of requirements

Customer Value

Sweet Spot

Avoid

High

Low

Development Effort

High

Low

Effort Estimation

Expert Judgment - the Delphi technique

- Experts make individual predictions secretly
- Calculate Mean
- Mean is presented to expert group
- [No change]
- [An expert changes its estimate]

Algorithmic Methods - COCOMO and COCOMO II

COCOMO (Boehm, 1981)
- An formula where parameters are estimated using real projects.
- Input: No of code lines
- Output: Effort (time)

COCOMO II
- Takes into account changes in SE, such as component reuse, prototyping
- Other inputs than number of code lines, E.g. function-points (e.g. external in/out, user interactions, files)
Illustrating example, COCOMO

- **Effort = \( C_1 \) \( EAF \) (Size)\(^{P_1}\)**
  - Effort = number of staff months
  - \( C_1 \) = scaling constant
  - \( EAF \) = Effort Adjustment Factor
  - Size = number of delivered, human produced source code instructions (KDSI)
  - \( P_1 \) = exponent describing the scaling inherent of the process (0.91-1.23)

Illustrating example, COCOMO II

**Predict maintenance size:**

- **Size = ASLOC \( \times 0.01 \)**
  - Assessment and Assimilation (0-8) (effort to test other S/W)
  - Software Understanding (10-50) (low:good structure)
  - 0.4 \( \times \) percentage of changed design
  - 0.3 \( \times \) percentage of changed code
  - 0.3 \( \times \) percentage of integrated external code
Algorithmic or parametric methods

Pros:
- Based on empirical data
- Potential up to +/- 20% accuracy
- No human bias

Cons:
- Data collection planned and performed
- Expensive consultants
- Rapid change in technology

Relative importance – Analytical Hierarchy Process

1. Expert pairwise comparison

F1: On line group-booking  7  F3: Last minute tickets
  3

F2: Round-trip tickets

2. Comparison matrix

\[
\begin{array}{ccc}
F1 & F2 & F3 \\
F1 & 1 & 1/3 & 1/7 \\
F2 & 3 & 1 & 1/5 \\
F3 & 7 & 5 & 1 \\
\end{array}
\]

3. Calculate normalized eigenvector = relative importance

Approximation:

\[
\begin{align*}
F1 & = 0.083 \\
F2 & = 0.193 \\
F3 & = 0.724 \\
\end{align*}
\]

For enthusiasts: http://www.boku.ac.at/mi/ahp/ahptutorial.pdf
AHP usage

Pros:
- Multiple criteria
- Simple comparison
- Fast
- Gives values on rational scale

Cons:
- Grows quadratically
- Relative values only
- Though chains of inconsistency
- Hard to add new alternatives
- Needs a tool
Expert judgment – Planning Poker

- Each developer has a set of cards, usually with values: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, “don’t know”
- Values translate into days or “points”
- Requirement and user story is described
- All developers picks a card
- All disclose their cards
- Discussion, time-boxed, lowest and highest estimator start.
- New estimation round

Enthusiasts: http://www.planningpoker.com/

Planning poker

Pros:
- Each participant make own decision
- All participants are active
- Participants get deeper knowledge
- Iterative work
- Consensus based final estimate
- Fun

Cons:
- Need of tools for distributed projects
- The orderer of the project needs to take consequences
- Risk of over-rating confidence when too little information is present
Part III
Risk Management

What is a risk?

Something that can eliminate full success of the project

Examples:

Staff turnover - Experienced team members will leave the project

Requirement change - Significant requirements will change late in the process.

Size underestimated - The size project was larger then expected
Kinds of risks

General

"A team member gets sick"
"There is a risk that the project gets delayed"

Project Specific

"The delivery of the development hardware environment is delayed."
"Anders needs to visit his family, since his father is dying."

Direct

The project has great control
"The Windows platform will not scale"

Indirect

Risk where the project has little control
"The servers will stop running due to an earthquake"

What is risk management?

Risk identification
List of potential risks

Risk analysis
Prioritized list

Risk planning
Risk plan

Risk monitoring
Risk assessment

"What can go wrong" "How bad is it" "What shall we do with it" "Has the probability changed?"
1. Risk Identification

Brainstorming with the whole team for 10 minutes.

**What can go bad?!?**

**Types of risks**
- **Technology risks** - Hardware/software technology used for development, e.g. using Java
- **People risks** - People in the development team
- **Organizational risks**
- **Tools risks** - Risks with the current tool used
- **Requirements risks** - Changes in customer requirements
- **Estimation risks** - Wrong project estimations

2. Risk Analysis

![Probability vs. Impact Diagram]

- **Probability**
  - low 1
  - moderate 2
  - high 3
  - very high 4

- **Impact**
  - insignificant 1
  - tolerable 2
  - serious 3
  - catastrophic 4

**Probability x Impact = Risk Magnitude Indicator**

Sort list after risk magnitude
3. Risk Planning

1. Risk Avoidance
Reorganize so that the risk disappears.

"Communication problem between develop sites in Stockholm and India - localize all development in India?"

"the web-server fails often - low accessibility outsource the operation?"

"Changes of requirements late in project - a prototype?"

2. Risk Transfer
Reorganize so that someone else takes the risk, insurance, customer, bank.

Mitigate the risk
Lower the probability.

"The key architect starts to work for another company - 2 architects?"

Define Contingency plan
A plan B...

3. Risk Acceptance
Live with it

Mitigate
Define Contingency plan
A plan B...

Example

Analyze

<table>
<thead>
<tr>
<th>No Risk Description</th>
<th>Probability</th>
<th>Impact</th>
<th>Risk Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. During implementation it is discovered that the new web-platform cannot talk to the legacy database system</td>
<td>Moderate (2)</td>
<td>Serious (3)</td>
<td>6</td>
</tr>
</tbody>
</table>

Plan

Avoid risk Do not introduce a new web-platform. Use the existing platform.

Transfer risk Sign a contract with a contractor, who guarantees access to the system.

Accept risk

Mitigate Create a prototype early in the process. Solve issues before implementation phase

Contingency plan Transfer the whole old legacy database system to a modern DBMS.
Make the risks useful

- Few (3-10)
- Project Specific
- Regular meetings

Part IV
Communication and Documentation
The Project Plan

Why a project plan?
- Tool for the project manager
- Communication medium between project members and other stakeholders
- What should be done, when and by who

When is the plan finished?

The Project Plan - Content

- Project Description
  - Background to the project
  - Relevant constraints (budget etc.)
  - Project Goal
  - Start and expected end date.

- Project Organization
  - Roles
  - Knowledge / skill
  - Communication and reports

- Risk Management
  - Risks, Probability and Impact
  - Mitigation and Contingency plan

- Time and Resource Plan
  - Milestones
  - Tollgates
  - Deliverables
  - Activities
  - Resources

- Training Plan
  - Needed knowledge and skills.
  - Who needs what? Budget?

- Change and configuration management
  (In larger projects, this part is a document of its own.)
Project Status Reports

Content of a status report?
- Summary - current status
- What has happened since last report
- What happens next (both in long and short term)
- Problems and risks

Part I
Introduction to Project Management

Part II
Time- and Resource Planning

Part III
Risk Management

Part IV
Communication and Documentation

More information...

Finally, never underestimate...

... a project Kick-off