Software Conception

台北科技大學
資訊工程系
鄭有進 副教授
http://www.ntut.edu.tw/~yccheng
Sample process in OOMD

- System conception (requirement)
- Analysis
  - Domain analysis
  - Application analysis
- System Design
- Class diagram
- Implementation
- Testing, training, deployment, maintenance
System conception takes place in the initial phase of a software project

- The inception phase in the Unified Process (UP)
  - Focus on requirements; downplay (and defer) decision and implementation decisions.
- Aims to produce the initial requirement artifacts generated, including
  - Problem statement
  - System context
  - Summary of system features
  - Non-functional requirements and constraints
Requirement is a big deal

- 37% of software project failure is attributed to requirement [Larman2002]
- Gaps of understanding between stakeholders, esp., clients + users vs. pm + developers.
- *System conception* looks to close the gaps through disciplined development of requirement document.
Problem statement

- A general description of the proposed software obtained through answering
  - The 6W’s of the software
- Stakeholders (both the client, end-users, and developers) must agree on what is being proposed
  - Reduces the risk that everyone is building a slightly different software (the gap problem)
The 6W’s: the high-level questions

- Who’s the application for?
- What problems will it solve?
- Where will it be needed?
- When is it needed?
- Why is it needed?
- How will it work?

The ATM example.
Who’s the application for?

- Two types of primary stakeholders
  - Financial sponsor (client)
  - End-users
What problems will it solve?

- Bound the size of effort and scope of system
- Determine what feature’s in and what’s out.
Where will it be needed?

- Characterize the environment the system will be used, e.g.,
  - Mission-critical?
  - Experimental?
  - Enhancements to existing system?
- For commercial products, characterize the customer base.
When is it needed?

- Feasible time, $T_f$
  - The time in which the system can be developed within the constraints of cost and available resource

- Required time, $T_r$
  - The time that the system is needed to meet the business goals.

- If ($T_r < T_f$), work with technologists and business experts to trim the system
Why is it needed?

- Prepare a business case
  - Financial justification, including
    - cost,
    - tangible benefits,
    - intangible benefits,
    - risk, and
    - alternatives (why build when you can buy?)
  - For a commercial product, estimate the number of units you can sell and determine a reasonable price.
    - Revenue must cover the cost
How will it work?

- Investigate feasibility of the problem
- Build prototype, if it helps clarifying a concept or removing a technological risk.
System context

- A diagram to accompany the problem statement to show actors (users and/or external services) and the proposed system.
  - Gives you the general context of the system’s use.
  - Gives you the big picture of the system architecture (form aspect): system level concepts and their relationships.
- Example: the ATM network in ch. 11
Functional and non-functional requirements: FURPS+ model

- **Functionality**: Functional requirement: Explored and recorded in system features list and use cases

- **Usability**

- **Reliability**

- **Performance**

- **Supportability**

- (+) Other requirements

Non-functional requirements that strongly influence the architecture
System feature list

- A list of "The system shall do ..."
  - They are observable as the general services the system provide
    - Used extensively in software projects since the old days
  - [Larman2002] A system feature is "an externally observable service provided by the system which directly fulfills a stakeholders need"
- In contrast, use cases captures the operational aspects of each service
Comparison: OOMD and Larman2002 (I)

- In OOMD, use cases are deferred to Analysis – application modeling
  - Encourages investigation of the static conceptual structure of the application before operational concepts.

- [Larman2002] Use cases receive up-front attention
  - They are more concrete.
  - Use cases are heart and center of the UP; it drives the development [UP is use case driven]
Comparison: OOMD and Larman2002 (II)

- But use cases alone are not enough as the requirement document.
  - Vision, supplemental requirement, glossary are needed also.
- Use cases can be difficult to obtain if you are not familiar with the application domain
- Both are good as sources for domain modeling.
Fishing in a lake or streams

Use case 2
Use case 1
Use case k

Problem statement
Artifacts during inception [Larman2002] and mapping

- Use cases
  - Deferred to analysis modeling
- Supplementary requirement
  - URPS+
- Vision
  - Problem statement
  - System context
  - System features
- Glossary
  - Data dictionary
How do these come about?

- Requirement capture requires *innovation* and/or *hard work*
  - Think alone
    - Survey and research
  - Think with a group (requirement workshop)
    - Brain-storming
    - Fish-bone
    - Etc.
  - This is done in a relatively short period of time in each iteration.
Safekeeping your precious

Where to store the ideas? In a *requirement document* that includes the following sections:

- Change history
- Problem statement [OOMD]
- System context [OOMD]
- Summary of system features [Larman2002]
- Use cases [[Larman2002, to be obtained in elaboration-analysis]
- Non-functional requirements and constraints [Larman2002]
- Data dictionary [OOMD-domain modeling, Larman2002]
Completeness of requirement

- The requirement document is not meant to be complete or even consistent during inception.
  - Aiming for complete requirements risks front-loading the software project
- Instead, it evolves in iterations.
  - Version control is necessary
- In RUP, the evolution is disciplined: the requirement document gets [Larman2002]
  - started in inception,
  - stabilized in elaboration, and
  - finalized in construction.
Evolution of the requirement document

Inception: started
Elaboration: stabilized
Construction: finalized

Effort/result vs. time

Transition
Additional reading

- [Larman2002] ch. 4, 5, 7
- [OOMD2005] ch. 11