Software Engineering - a brief introduction

Erik Perjons

Questions to answer

- What is software engineering?

Software Engineering

Software Engineering

- "Software Engineering is the branch of computer science that seeks principles to guide the development of large complex software systems" (Brookshear, 2008)

- Examples of sub-disciplines/sub-branches:
  - Requirement engineering
  - Software/System design and development
  - Software/System testing
  - Software/System maintenance
  - ...

System development

System (software) development - is the creation/construction of system/software, and it is usually described as a set of phases/activities

Requirement

Design

Implementation

Test & Deploy

- **Requirement engineering** – define the requirement of the system (this phase/activity is sometimes called analysis)
- **Design** – define the overall architecture and structure of the system
- **Implementation** – code the functions and build the database system
- **Test and deploy** – test that the implemented system fulfill the requirements, and introduce the system in the organisation and integrate it with processes and other systems

Prototype

- There is no agreement on what a prototype is, but it can be seen as an early form of the final system, focusing on certain aspects of it.

- The prototype enables the designers to better understand design challenges before developing the final system, and the prototype is also often constructed to obtain early user-feedback
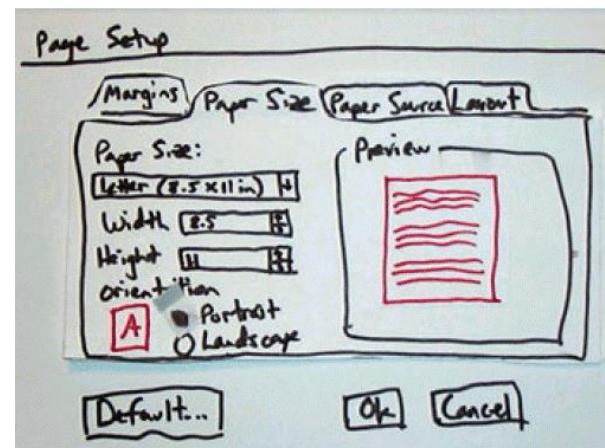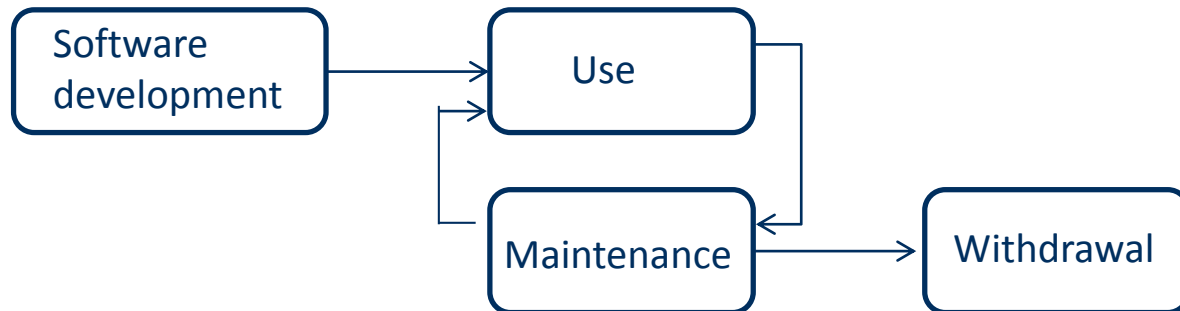
Stakeholders

- Stakeholders in software engineering are people or roles that have an interest in a software system, such as users, business analysts, software developers, IT support, managers, executives, owner, etc

- Stakeholders need to be involved in the system development, especially in the requirement engineering phase, so that the "right" system is developed

Software Life Cycle

- The **software life cycle** describes the major phases of a software system from development to withdrawal of the system

- Since the **maintenance** (i.e., correcting and updating the software) can be very costly, this phase has to be considered during development: is the system developed so that the correcting and updating of the systemare easy to do?
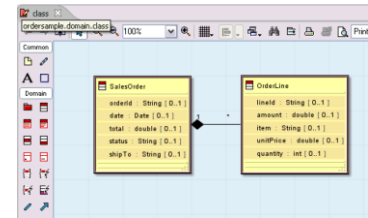
```
┌──────────────┐        ┌──────────────┐
│ Software     │───────▶│     Use      │────┐
│ development  │    ┌──▶ │              │    │
└──────────────┘    │   └──────────────┘    │
                    │   ┌──────────────┐    ▼   ┌──────────────┐
                    └───│ Maintenance  │◀───────│  Withdrawal  │
                        │              │───────▶│              │
                        └──────────────┘        └──────────────┘
```

Tool Support for Software Engineering

- **Integrated development environment (IDE)**

  – is an environment that "combine tools for developing software (that is, editors, compilers, debugging) into a single, integrated package, some of which provide visual programming features..." (Brookshear, 2008)

- An interesting example of an IDE is the **Eclipse platform** that is an open source environment which provide a number of plug in for different purposes
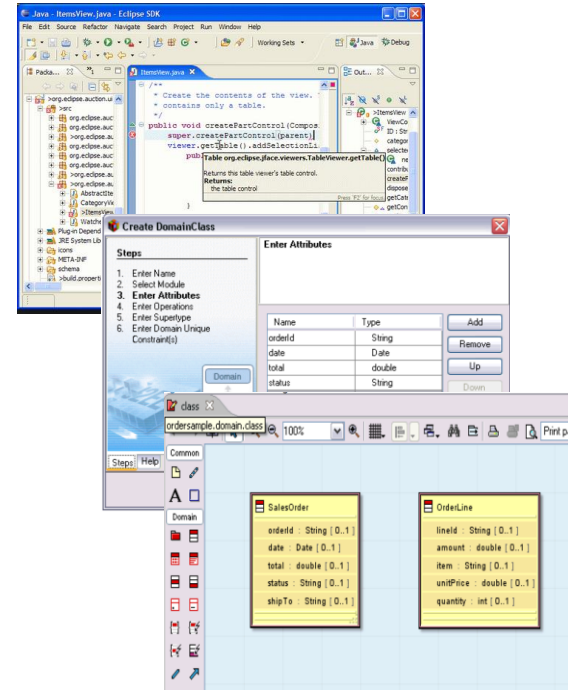
Tool Support for Software Engineering

- **Modelling tool** – is a tool that support graphical modelling of system, including software system, for example UML modelling (like Visio, ArgoUML)

- **Model-driven development tool (MDD tool)** – is a tool that support generation of code and user interface from graphcal models (like Rational Software Architect)

- **Case tool** – "Computer-aided software engineering" – is a tool that support analysis, design och implementation of software – but is sometimes seen as predecessor to IDEs and MDD tools

Questions to answer

- What is software engineering?

Medverkande

Erik Perjons – Lärare

Jonas Collin – Mediepedagog

Inspelat 2015-08-30
Institutionen för data- och systemvetenskap, DSV