**Week 4 Software Development**

**What is software development?**

* **Computer application development**: includes a computer software program that executes on a computing device to carry out a specific set of functions
* **Information system development:** includes a set of interrelated components that collects, processes, stores, ad provides as output the information needed to complete business tasks. Includes database and related annual processes.
* **Systems analysis:** the activities that enable a person to understand and specify what an information system should accomplish. (what is required)
* **Systems design:** the activities that enable a person to define and describe the system in detail and the ‘need’ the system solves. (how it will solve the problem)

**Information systems**

An information system is a group of components that interact to produce information.

Five fundamental components of IS:

* Hardware
* Software
* Data
* Procedures
* People

**5 core activities of software development projects**

* Requirements analysis
* Software design
* Implementation
* Testing
* Deployment and maintenance

**Waterfall development model**

Requirements/ analysis -> design-> coding-> testing-> maintenance

In the waterfall development model operation are don’t sequentially, the model has a lot of structure which allows for control over project development. In practice this model is often too rigid; clients may not know exactly what their requirements are before they see working software and so changes must be made, leading to redesign, redevelopment and retesting which in turn increases development costs. Designers may not be aware of future difficulties when designing new software, it is better to revise the design than persist in a design that does not meet requirements.

**Agile development**

Agile philosophy says development should change based on the changing user requirements, no matter what stage of development we are up to.

* Individuals and interactions over processes and tools
* Working software over comprehensive documentation
* Customer collaboration over contract negotiation
* Response to change over following a plan

**Agile methodologies**

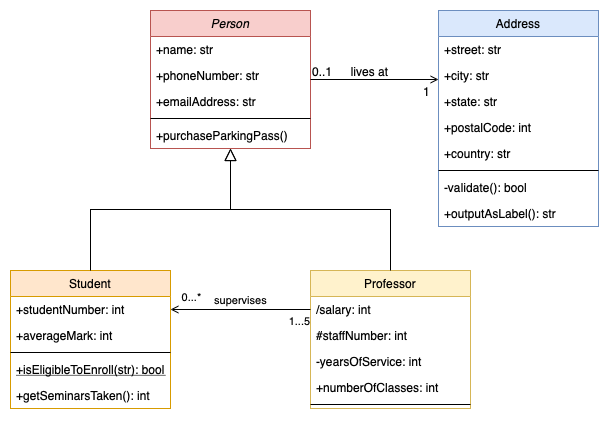
* **Scrum:** break work into goals that can be completes within time-boxed iterations (sprints), Team assesses progress in short daily meetings (scrum meetings)
* **Unified process / rational unified process:** 4 project phases – Inception -> Elaboration -> Construction -> Transition. The key aspects of rational unified process include a risk-driven process, use case focussed development, and architecture-centric design.
* **Extreme programming:** flexibility in timelines. Priority order dictates everything.

Agile lacks structure that large companies or organisations desire, particularly if integrating with other business units or processes that have their own methodologies.

**Iterative development**

* Business modelling
* Requirements
* Analysis & design
* Implementation
* Test
* Deployment

**Other development approach processes**

* **Pre-project**: Identify the problem and document the objectives of the system.
  + Preliminary investigation
  + System vision document
    - Problem description (what is the problem?)
    - System capabilities (what will the system do?)
    - Business benefits (how will the program create benefit?)
    - Who would use the software and what does it need to do?
  + Obtain approval to commence project (meet with stakeholders)
  + Decision reached, approve plan, create budget
* **Requirements analysis:** discover and understand the details of the proposed software.
  + Preliminary fact-finding to understand requirements
  + Preliminary list of use cases and use case diagrams
    - Technique for capturing, modelling and specifying the requirements of a system
  + Preliminary classes and class diagram
    - Main building block of object oriented modelling
    - Describes the system structure by showing classes, their attributes, operations/ methods and the relationships amongst objects
    - Well defined formalized modelling tool
* **Software design document:** includes requirements analysis, resulting use cases and UML class diagrams. Can include other things such as overall system architecture, any pseudocode etc.
* **User interface design:** actively designing and thinking about the UI for the program.
* **Software testing report:** the main document that covers the strategy and results of all software testing performed.