cphbusiness

Introduction to Development of Large Systems

Introduction to Software Engineering

cphbusiness

What is this course about?

Large & Complex Software Systems

https://bitwizards.com/Thought-Leadership/Blog/2016/February-2016/Software-Estimates-Aren-t-Free-for-Anyone

What is this course about?

Software Engineers,
i.e., us now

Large & Complex Software Systems

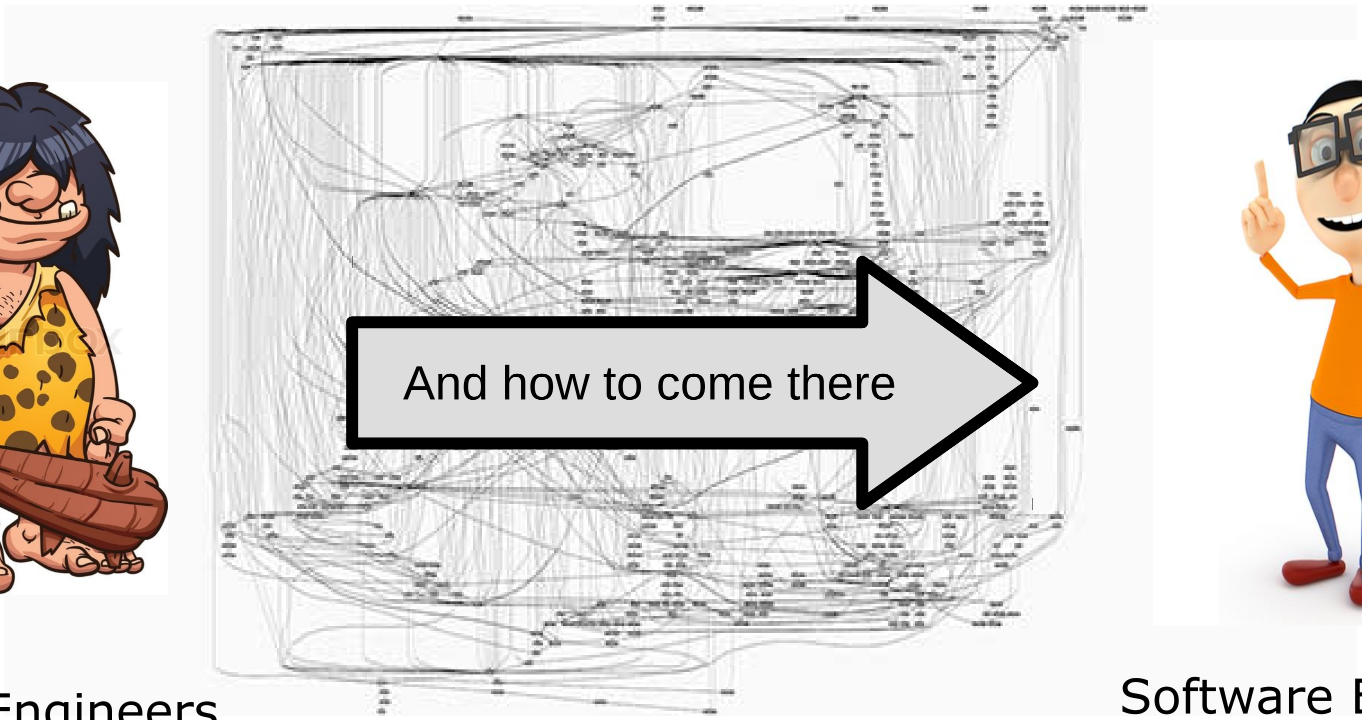
What is this course about?

Software Engineers,
i.e., us now

Large & Complex Software Systems

Software Engineers,
i.e., us at the end
of the course

What is this course about?



Software Engineers,
i.e., us now

Large & Complex Software Systems

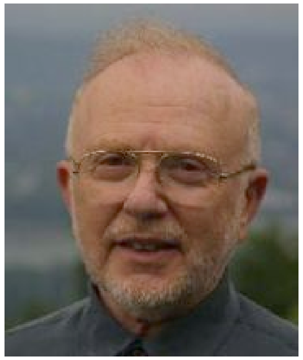
Software Engineers,
i.e., us at the end
of the course

cphbusiness

Software Development is More than Writing Code!

- The problem is usually ambiguous
- The requirements are usually unclear and changing when they become clearer
- The problem domain (called application domain) is complex, and so is the solution domain
- The development process is difficult to manage
- Software offers extreme flexibility
- Software is a discrete system
  - Continuous systems have no hidden surprises
  - Discrete systems can have hidden surprises! (Parnas)

David Lorge Parnas - an early pioneer in software engineering who developed the concepts of modularity and information hiding in systems which are the foundation of object-oriented methodologies.
(Paper: SOFTWARE ASPECTS OF STRATEGIC DEFENSE SYSTEMS)



Why is Software Development Difficult in the Large?

- Many developers
- Geographical distances
- Different time zones
- Different cultures
- Different languages
- Different technical skills
- Organizational challenges
- Sharing source texts
- Frequent deliveries

Why does it Matter to do it Right?

Air-Traffic Control System in LA Airport

Incident Date: 9/14/2004

(IEEE Spectrum) -- It was an air traffic controller's worst nightmare. Without warning, on Tuesday, 14 September, at about 5 p.m. Pacific daylight time, air traffic controllers lost voice contact with 400 airplanes they were tracking over the southwestern United States. Planes started to head toward one another, something that occurs routinely under careful control of the air traffic controllers, who keep airplanes safely apart. But now the controllers had no way to redirect the planes' courses.

...

The controllers lost contact with the planes when the main voice communications system shut down unexpectedly. To make matters worse, a backup system that was supposed to take over in such an event crashed within a minute after it was turned on. The outage disrupted about 800 flights across the country.

Why does it Matter to do it Right?

...

Inside the control system unit is a countdown timer that ticks off time in milliseconds. The VCSU uses the timer as a pulse to send out periodic queries to the VSCS. It starts out at the highest possible number that the system's server and its software can handle—232. It's a number just over 4 billion milliseconds. When the counter reaches zero, the system runs out of ticks and can no longer time itself. So it shuts down.

Counting down from 232 to zero in milliseconds takes just under 50 days. The FAA procedure of having a technician reboot the VSCS every 30 days resets the timer to 232 almost three weeks before it runs out of digits.
...

http://www.cse.psu.edu/~gxt29/bug/softwarebug.html

Why does it Matter to do it Right?

Knight Capital's $440 million loss

Incident Date: 8/1/2012   Price Tag: $440 million

(The Register) Knight Capital, a firm that specialises in executing trades for retail brokers, took $440m in cash losses Wednesday due to a faulty test of new trading software.
...
Unfortunately, the trading algorithm the program was using was a bit eccentric as well. On every stock exchange, there is a "bid" and an "ask" price. The bid price is what you'd like to pay the holder of the stock if you want to buy their shares. The ask price is what they'll pay to buy those same shares from you. There's always a spread between the two prices, with the "ask" being a few cents or more above the "bid". If the stock is thinly traded, then the spread between the ask and the bid is higher than what you.d see for, say, IBM.

Why does it Matter to do it Right?

Knight Capital's software went out and bought at the "market", meaning it paid ask price and then sold at the bid price--instantly. Over and over and over again. One of the stocks the program was trading, electric utility Exelon, had a bid/ask spread of 15 cents. Knight Capital was trading blocks of Exelon common stock at a rate as high as 40 trades per second--and taking a 15 cent per share loss on each round-trip transaction. As one observer put it: "Do that 40 times a second, 2,400 times a minute, and you now have a system that's very efficient at burning money".

As the program continued its ill-fated test run, Knight's fast buys and sells moved prices up and attracted more action from other trading programs. This only increased the amount of losses resulting from their trades to the point where, at the end of the debacle 45 minutes later, Knight Capital had lost $440m and was teetering on the brink of insolvency.
...

Computer Science vs. Engineering

Computer Scientist:
- Assumes techniques and tools have to be developed.
- Proves theorems about algorithms, designs languages, defines knowledge representation schemes
- Has quite much time…

Engineer:
- Develops a solution for a problem formulated by a client
- Uses computers & languages, techniques and tools

Software Engineer:
- Works in multiple application domains
- Has only 3 months...
- …while changes occurs in the problem formulation (requirements) and also in the available technology

What is Software Engineering About??

- It is **problem solving**
  - Understanding a problem
  - Proposing a solution and plan
  - Engineering a system based on the proposed solution using a good design
- It is about dealing with **complexity**
  - Creating abstractions and models
  - Notations for abstractions
- It is **knowledge management**

Techniques, Methodologies, Tools

Techniques:

- Formal procedures for producing results using some well-defined notation

Methodologies:

- Collection of techniques applied across software development and unified by a philosophical approach

Tools:

- Instruments or automated systems to accomplish a technique
- Interactive Development Environment (IDE)
- Computer Aided Software Engineering (CASE)

Software Engineering: A Working Definition

Software Engineering is a collection of **techniques**, **methodologies** and **tools** that help with the production of

A **high quality** *software system* developed with a given **budget** before a given **deadline** while **change** occurs.

Two major challanges:

- Complexity
- Change

Software Engineering: A Problem Solving Activity

Analysis

- Understand the nature of the problem and break the problem into pieces

Synthesis

- Put the pieces together into a large structure

For problem solving we use techniques, methodologies and tools.

cphbusiness

Software Engineering Concepts

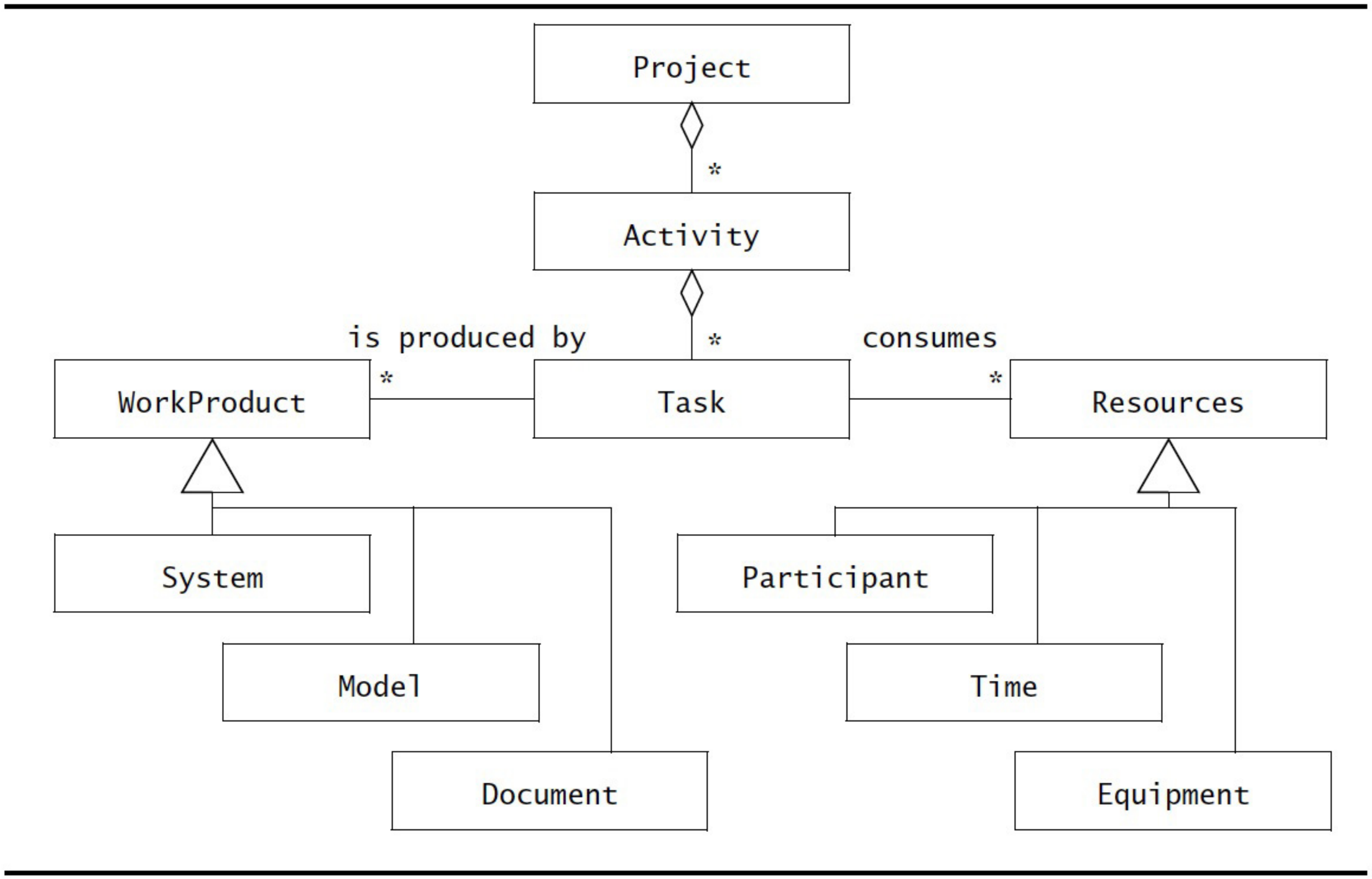
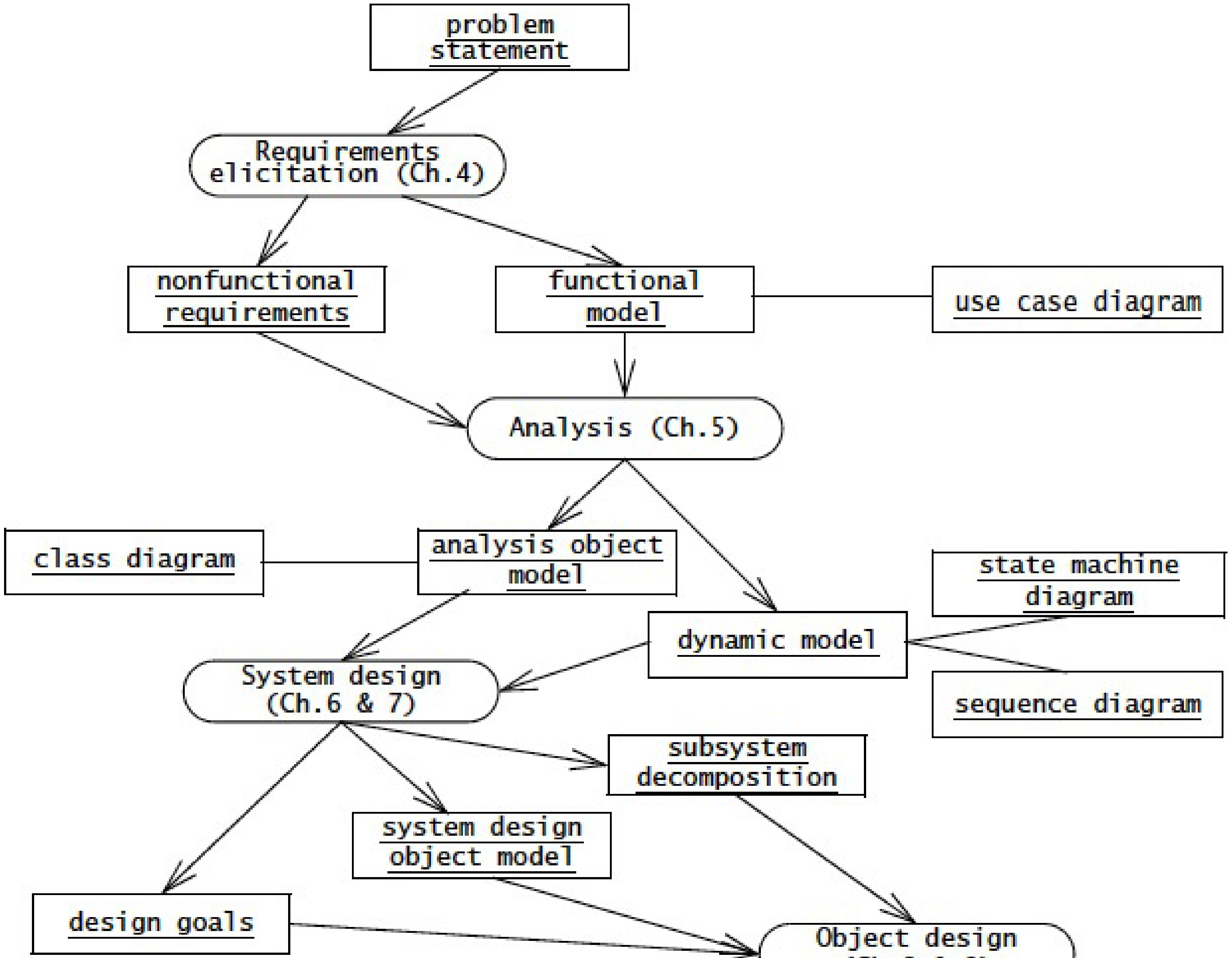


Figure 1-1 Software engineering concepts depicted as a UML class diagram [OMG, 2009].

SE Activities & Products



SE Activities & Products

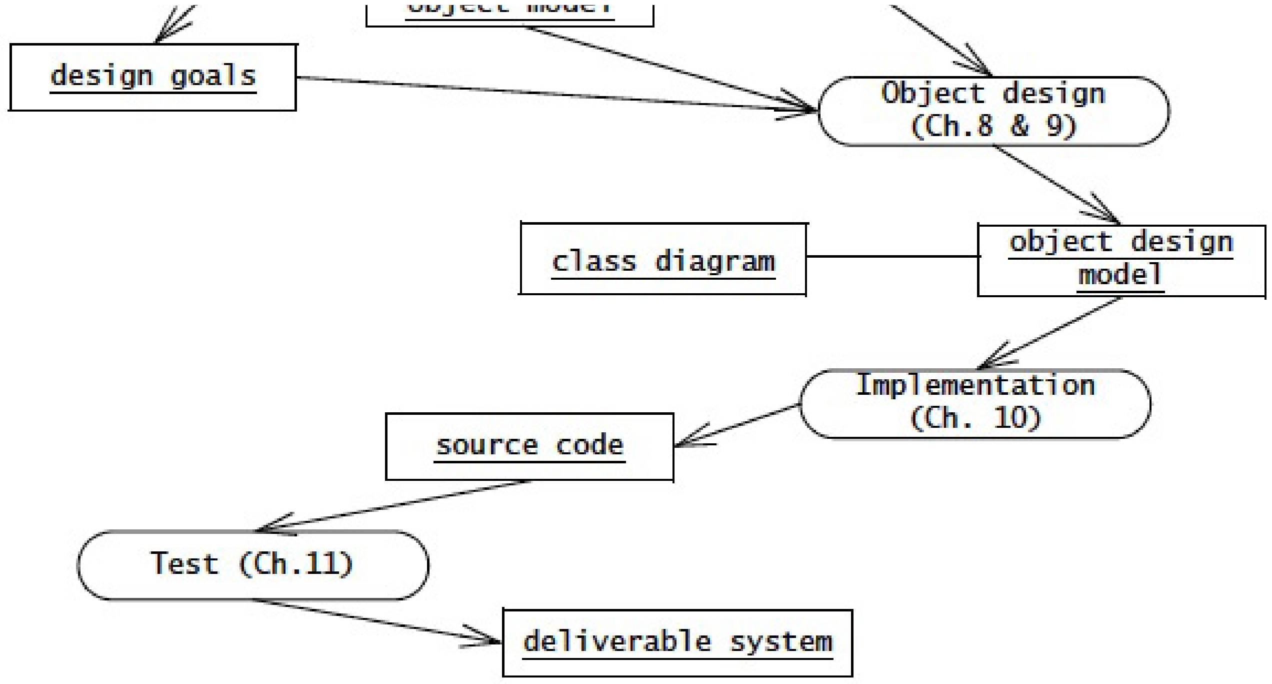


Figure 1-2 An overview of object-oriented software engineering development activities and their products. This diagram depicts only logical dependencies among work products. Object-oriented software engineering is iterative; that is, activities can occur in parallel and more than once.

Software Engineering: A Problem Solving Activity

Dealing with Complexity

- Notations (UML, OCL)
- Requirements Engineering, Analysis and Design
  - OOSE, SA/SD, scenario-based design, formal specifications
- Testing
  - Vertical and horizontal testing

Dealing with Change

- Rationale Management
  - Knowledge Management
  - Patterns
- Release Management
  - Configuration Management, Continuous Integration
- Software Life Cycle
  - Linear models
  - Iterative models
  - Activity-vs Entity-based views

Application of these Concepts in the Exercises

Until Next Time

- Read B&D ch. 4 Requirements Elicitation
- You and your group prepares to present your answers to Assignment 1 UP & Agile next Monday

Introduction to the Course

cphbusiness

Curriculum

- Bernd Bruegge & Allan H. **Dutoit: *Object-oriented Software Engineering Using UML, Patterns and Java***; 3rd edition - Chapter 1, 3, 4, 5, 6, 7, 11, 14, and 15 ISBN: 9781292024011
- ***Introduction to Project Management*** http://www.upedu.org/process/discplns/manageme/int_pm.htm
- Project Management Institute, Inc. **A Guide to the Project Management Body of Knowledge** (PMBOK® Guide)—Fifth Edition Chapter 10 & Annex 1 ISBN 978-1-935589-67-9
- And other papers and publications...

Assignments

- **Assignment 1: UP and Agile methods** –presented to you on Wednesday 7th Sept.– and your answers should be presented in plenum Wednesday 14th Sept. (in your groups)
- Large systems case description and pre-study report: a system for ferry ticket reservations – presented to you on Monday 3rd Sept.
- ***Compulsory* Assignment 2: Pre-study report** incl. work effort estimates –to be uploaded not later than 24o'clock on Wednesday 21th Sept
- **Assignment 3: Technical and architectural considerations** -present Wednesday 5th Oct.– presented to you today briefly
- ***Compulsory* Assignment 4: Development of Large Systems** – hand in start of January

Assignment 1 UP and Agile

1) Briefly describe the main characteristics in UP & Agile Development
2) How do they fit to ”Development of Large Systems”?
3) Make a list of advantages and disadvantages for each of the methods
4) What are the charateristics of the process and the deliverables?
5) When will you recommend UP versus Agile Development? Explain why
6) Is there any difference if you are looking at ”The System Lifecycle” and the requirements for maintainability and Non Functional Requirements (e.g. FURPS+ see next page)?

Prepare in your groups and a short presentation (approx. 10 min) until next time.

cphbusiness

Assignment 1 UP and Agile

FURPS+

A requirement is defined as ”a condition or capability to which a system must conform”
There are many different kinds of requirements. One way of categorizing them is described as the FURPS+ model [GRA92], using the acronym FURPS to describe the major categories of requirements with subcategories as shown below
Functionality, **U**sability, **R**eliability, **P**erformance, **S**upportability
The ”+” in FURPS+ reminds you to include such requirements as: Design constraints, Implementation requirements, Interface requirements, physical requirements

See also:
http://www.upedu.org/process/gcncpt/co_req.htm
http://epf.eclipse.org/wikis/openup/core.tech.common.extend_supp/guidances/checklists/system_wide_rqmts_furps_3158BF2F.html
http://www.agilemodeling.com

Large System Case 1/2

A medium sized Danish municipality runs two ferry lines which are serviced by three permanent ferries.

The ferry lines service two small islands, one close by (approx. 20 minutes) and one about an hour away. The biggest ferry sails mainly on the long route, but can on rare occasions be used on the short one. A medium sized ferry is used on the short route in summer, and on both rou-tes in winter. The smallest ferry is used only on the short route. When one or more ferries are in dock, one of two substitute ferries might be borrowed from a neighbour municipality.

The municipality wants a system for reservations. The largest ferry has a movable deck, which when lowered supports more private cars but fewer lorries. The medium sized ferry can be rearranged to support big machinery as harvesters and big tractors. This rearrangement is not simple, and has to be planned at least 24 hours ahead. The smallest ferry can only transport people, no vehicles. On the short route only residents can bring cars, because the island is quite small and is subject to nature conservation.

cphbusiness

Large System Case 2/2

The municipality already has a reservation system, but it is not user friendly, and has shown to be prone to changes; due to the purchase of a new medium sized ferry; the system has not been working for the short line for the last couple of months.

The municipality is negotiating with the neighbour municipality about a merger between their shipping activities. Ticket prices differ a lot between the two municipalities, and even between the different ferry lines. At some lines to the minor islands, citizens with permanent addresses on the islands are entitled free rides, rules for how many changes over time and from line to line. Offering online tickets has been discussed.

For inspiration concerning content of reservations for ferry lines you can see links such as:
http://www.scandlines.com/
http://www.aferry.co.uk/
http://www.kalundborg.dk/Forside-2.aspx

Assignment 2 Large System Case

Pre study report 1/3

You – as a software developer – are now responsible for analysis and definition of requirements for the best solution. Therefore you have to document your considerations in a minor pre-study report.

The content of the pre study report must be:

- Problem analysis – with a short description of Causes and Effects as you see in the project
- Project Goals – define the goals you set up for the project. Look at the process and of course the products
- Stakeholders and their position relative to your project. Position is things like: Is the stakeholder positive or negative? How can they contribute (Skills; Knowledge; Money; etc.)? Are there any potential conflicts between some of the stakeholders? Are there any coalitions between some stakeholders?
- Risk analysis. It's useful to look at the attached example template and the link http://www.upedu.org/process/artifact/ar_rskls.htm
- Overall description of (you have to define them)
    - Functional & Non Functional Requirements
    - Description of the domain (Model)
    - Description of the required hardware incl. basic software (if any)

Assignment 2 Large System Case

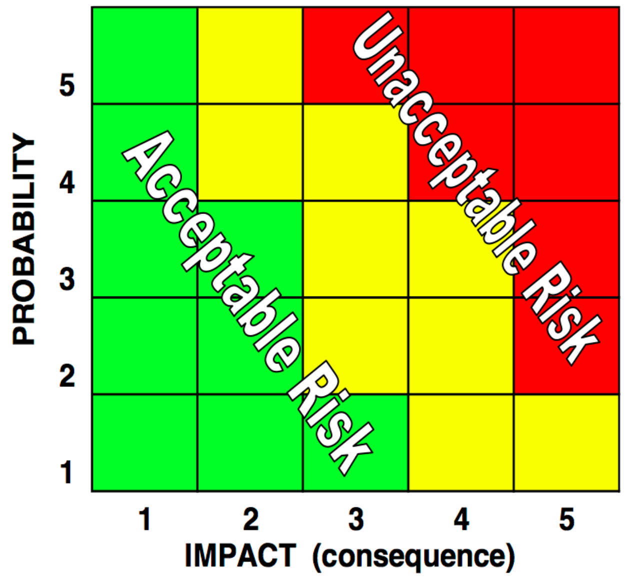
Pre study report 2/3

- Project definition and recommendations:
  - Development path – which development methodology will you recommend? (UP; Agile; Prototyping; etc.)
  - Methods, Techniques and Tools (MTT) – Is there any specific set of MTT you will recommend?
  - Quality Assurance activities – which kind, with whom, and where in the project life?
  - Further development after this project? – Is there a phase 2?
  - Is it an incremental development approach that you have chosen?
  - Overall plan for the whole project life (Gantt-Chart or Burndown Chart etc.)
  - Detailed plan for the next step/phase
  - Cost/benefit

OBS: It’s a compulsory assignment. It must be approved and you must upload the pre-study report not later than than 24o'clock on Wednesday 21th Sept. You will get feedback just during autumn holiday.

Assignment 2 Large System Case

Pre study report 3/3



Value consecuences

1	A minor issue in the project that creates irritation
2	
3	Issue that cause delays, necessitating a revision of the plan and budget
4	
5	Serious issue that prevents the project implementation, and consistency can be closure of the project

Prevent when High Probability an Minor Consequence
Mitigate when High Consequence and Minor Probaliility (Plan B)

				Define action taken	
Item	**Probability**	**Consequence**	**P*C (Risk value)**	**Prevent**	**Mitigate**
Do information technology projects align with and support business directions and priorities?					
Will clients formally sign off on the reviews at the project's milestones and be involved in decisions on the project's future?	1	1	1		
Has the project leader ensured that representatives of client groups have formally committed to the level of effort required to meet their defined responsibilities?	1	1	1		
Have client names and explicit responsibilities been included in the project charter?	1	1	1		
Are clients seen as full team members who take part in all requirements definition, design and implementation decisions of the project, rather than just as sounding boards for bouncing ideas off of or to lend ceremonial approval to project team decisions?	1	1	1		
Are clients involved in the decision to release funds and continue the project at milestone reviews?	1	1	1		
Are clear accountabilities established?					
Are overall departmental accountabilities for the project defined in a project charter?	1	1	1		

Assignment 3 Large System Case

Preparation 1/3

For the ferry company, two user interfaces are needed, one for administration and one for use by regular customers. The system should be created in a strongly typed language as Java or C#.
This leaves us with at least five subsystems. Depending whether we choose to use the same technology for both the administrative and the customer part.

cphbusiness

Assignment 3 Large System Case

Preparation 2/3

Contracts between frontends and backends will be defined in the course ”Contract based Development”
Before 5th Oct. you should consider the following:

- What language will you use in your group, C# or Java
- Which subsystem would you prefer to develop
- Consider which models you need, and which you already have
- Consider how to apply Git as version control system in your project
- Consider how to apply Jenkins as CI system in your project
- Consider how to apply Docker containers in your project

Assignment 4 Large System Case

Development of Large Systems 1/2

Refer to first and second slide in Assignment 3

The goal of the project is to set up a continuous delivery framework.

- The code should be placed on Git (e.g. GitHub or BitBucket), and Jenkins should be set up to build and test the code.
- A full implementation of the system is not expected.
- There shall be separate repositories for at least frontend, contract, contract test, and backend in each trail (Administrator – Customer, in Java and C#)

cphbusiness

Assignment 4 Large System Case

Development of Large Systems 2/2

You should in groups create a report that:
- Includes the final toolbox report made on the contract
- Explains the setup including urls for the repositories
- Includes a short instruction on setting up Jenkins in your particular setup
- Implements one or two use cases, as a proof of concept for the setup
- Implements tests for the implemented use cases. The coverage does not have to be 100% for the tests
- Include UML and code in the report, where it is relevant

The report must be handed in on Fronter no later than start of January at 24 o'clock. Feedback will be given, also on Fronter before the exam.