Chapter 1

Method

Concepets

Object	Entity with identity, state and behaviour
Class	Describes a collection of objects sharing structure, behavioural patterns and attributes
Problem domain	Part that is administrated, monitored or controlled by a system
Application domain	The organization that administrates the problem domain Where the user is
System	A collection components that implements modeling requirements, functions and interfaces
Context	Problem domain and application domain

Describe problem- and appli- cation domain better

Der er eksem- pler bag i bogen

Problem domain

Class structure and behaviour

Application domain

Usage functions and interfaces

Method

Purpose, concepts, principles and results.

1.1 Objects and classes

Objects - *Entity with identity, state and behaviour*

Each object serves as a seperate function. The object could be a customer, where specific people are treated as customers. The object contains that specific customers identity, state and behaviour

Class - *Describes a collection of objects sharing structure*

The class contains multiple objects, meaning a customer class will contain multiple data points. The class also contains multiple different customers and their data points.
When describing a class it's important to choose the right granularity. Gravel pit should not be described by the individual grains of sand, instead by the type, whereas a warehouse the individual packages should be described.

Analysis - outside the system

In analysis the object's behaviour is described by its events it performece and experiences that happens in definite points in time. Eg. customers ordering and shipping goods.

Design - inside the system

In design the object's behaviour is described by the operations it can perform and make available to other objects in the system. Eg. add order etc. This allows the update of eg. the customers object state. The design object encapsulates the internal representation of the object state through its operations.

1.2 Principles

The 4 principles:

Model the context - Useful systems fit the context, so model both application and problem domain during analysis and design.

Emphasize the architecture - Understandable architecture makes collaboration between programmers and designers possible. Flexible architecture makes modifications and improvements affordable

Reuse patterns - Building on well-established ideas and pretested components

Tailor the method to suit specific project - Must be tailored to the specific needs of the analysis and design situation

Model

Model is a representation of the state in the problem domain. How often the model is updated is a design decision
Problem domain -¿ model -¿ application domain.

Chapter 2

System Choice

Define the system in its context problem domain + application domain.
Using the F.A.C.T.O.R system.
F: Functionality - System functions that support the AP tasks.

A: Application domain - The organization that administrates the problem domain. Where the user is

C: Condtion - The conditions under which the system will be developed

T: Technology - Both the technology used to develop the system and the technology which the system will run.

O: Objects

R: Responsibility

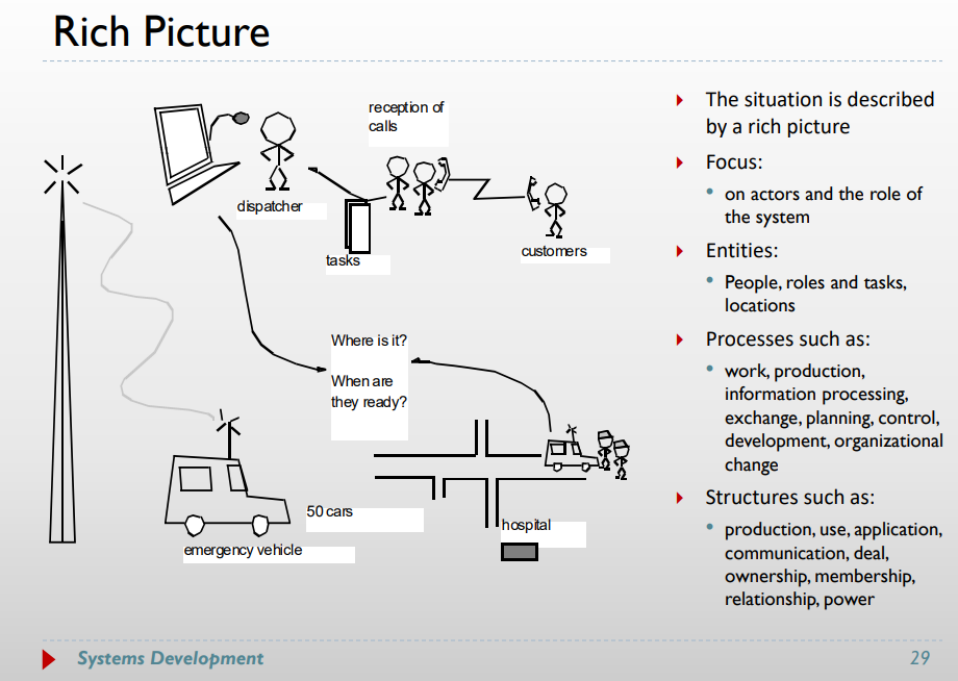
System definition

Situation

Describe the situation based on the context of the system. Described in a rich picture.
Which is the described by its focus, entities, procceses and structure.

Ideas

Examples - eg. Study preexisting systems.



System definition

Not to be confused with *system*. A concise description of a computerrized system expressed in natural language.

Context