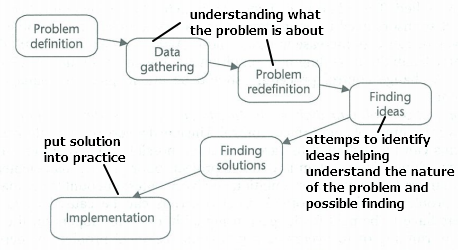
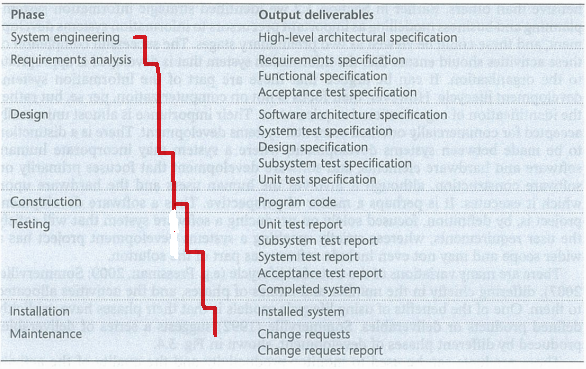
Project Lifecycles

Subdividing problems into more tasks: easier to manage, there can be many ways but the core activities (understanding the problem, choosing and designing, and building the solution).

Software development project = focused solely on producing a software system that satisfies the user requirements

System development project = wider cope, may not even include software.



**Waterfall lifecycle** = systematic approach,

good when requirements are not likely to change during the development process and nothing has to be delivered during the dev. proc.:

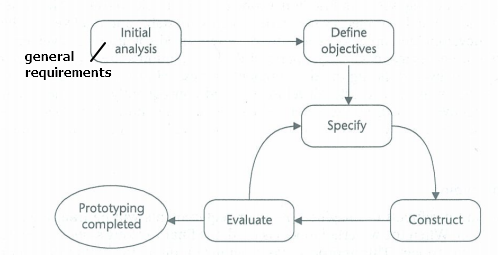
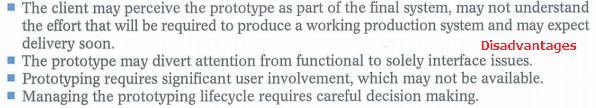
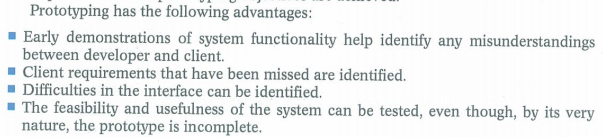
+ the phases have explicitly defined products or deliverables which can be manages individually.

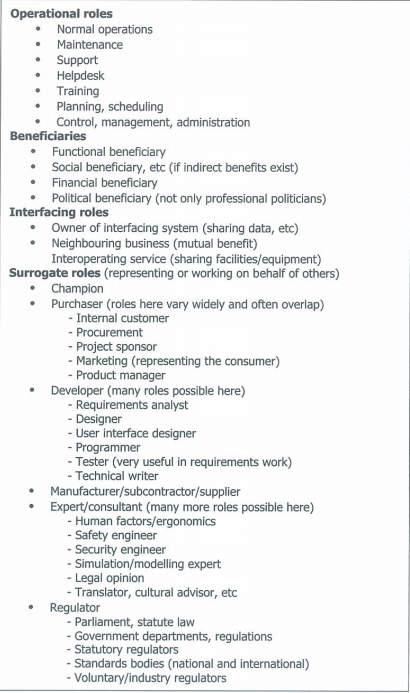
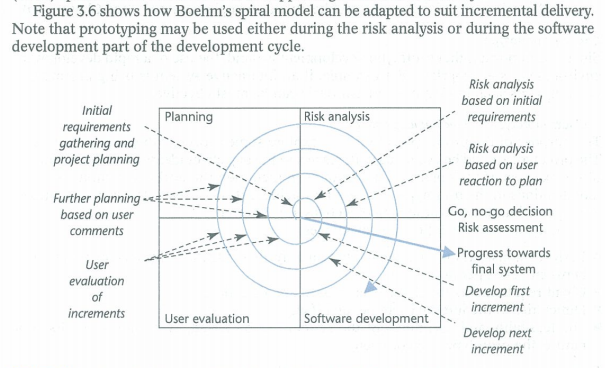
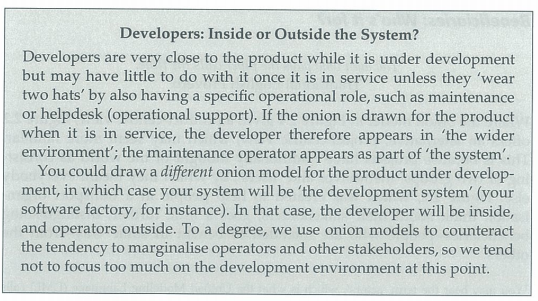
- Real project rarely are that simply structured

- Task have to be repeated

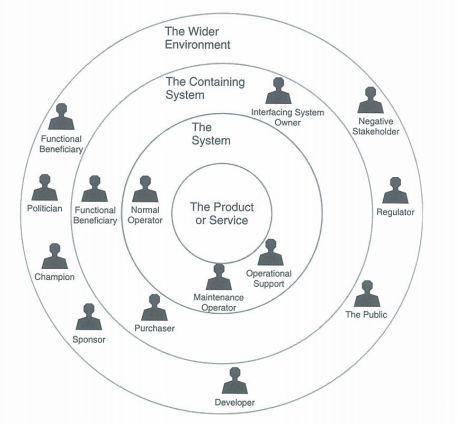
- requirements may change (when lot of time between start and end)

- difficult to change, inflexible (f.ex when new technologies comes up)

Prototype:



Prototype



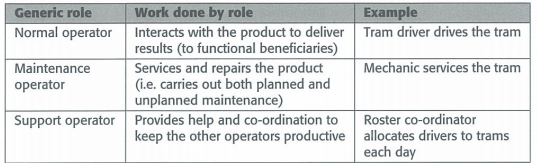
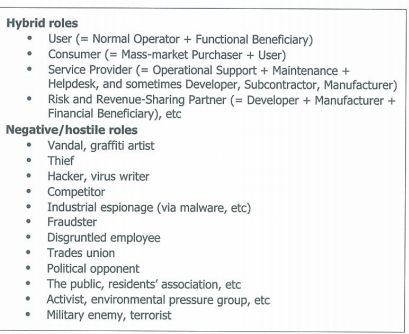
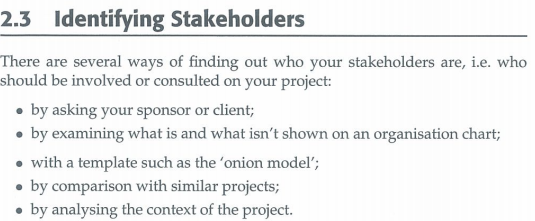
**System:** people that deliver results to the world outside.

**Beneficiary:** who intend to benefit in some way (financial, political, functional, social)

**Regulator:** according to law, standards

**Interfacing Roles:** Responsible project / operation of systems that our service or project has to cooperate with

**Negative Stakeholders:** peaceful opposition to active hostilitiy, can threaten or harm project; f.ex. Competitors, security threats

**Champion:** really wants the project to work out, provides “political” support (Can be the same as sponsor but normally not)