OOA&D in practice: Waterfall and Iterative approach.

Unclear requirements

- There are many examples of problems with requirements
- Unclear requirements have been denoted as the main source of problems in system development projects.

Requirement defect: the product works as intended by the programmers, but doesn't match the surroundings (Context: App. Domain + Prop. Domain).

Examples:

- Users and customers are not satisfied with the product. They find it too difficult to use, unable to support certain user tasks, etc.
- The program doesn’t cooperate properly with existing, surrounding software.

Examples of things coursing errors:

- Missing requirements: a requirement that is indirectly required but not written down.
- Mistaken tacit requirements: mistakes related to something not being formulated. (tacit = not spoken)
- Mistaken specs
- Defects relating to external software

Defects and Potential Techniques

Defects in existing product:

- 60% of the defects related to instated demands (tacit requirements)
- Almost 70% of the defects had to do with ease of understanding or ease of use (usability)
- Most related to the user interface and misunderstood interfaces to third-party software.

Potential techniques:

- Identified 44 techniques from literature or practice.
- For each defect they identified the techniques that might find or prevent the defect - and with what probability
- About 10 techniques were worth considering in a project of this kind.

Measured effect in a New Project

A new project:

- The user tasks were studied directly and the user interface was designed and usability tested before any part of it was programmed.

Obstacles:

1. Some top techniques were useful in one kind of project, but much less important in other projects.
2. The organisational surroundings may block the use of some techniques.

3. Developers have difficulties using many new techniques at the same time.
4. Unforeseen events, such as a new project manager, can overturn earlier decisions to use a certain technique.

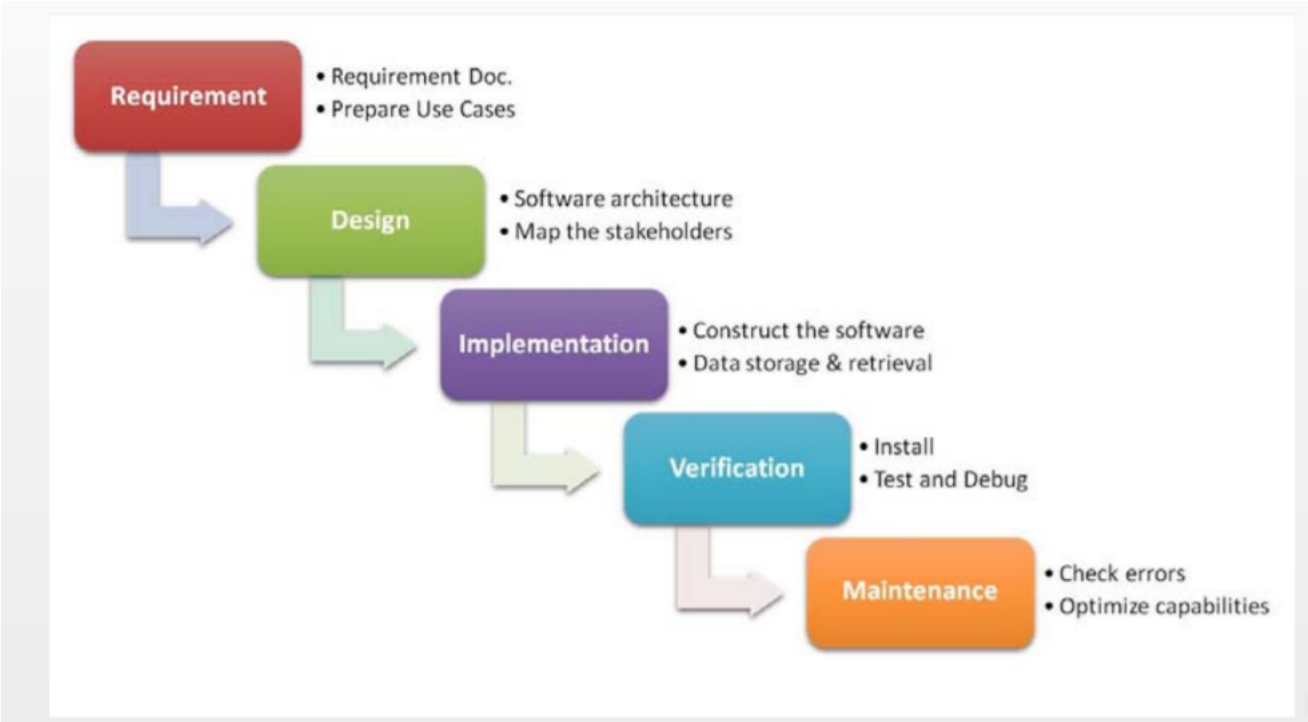
In the new project (experiences)

1. The number of usability problems per screen picture was reduced by bout 70% (they expected18%).
2. The project was the first one ever in the company that had been completed on time and without stress.
3. The product sold twice as many units as comparable products and at twice the unit price.

Construction and Iteration

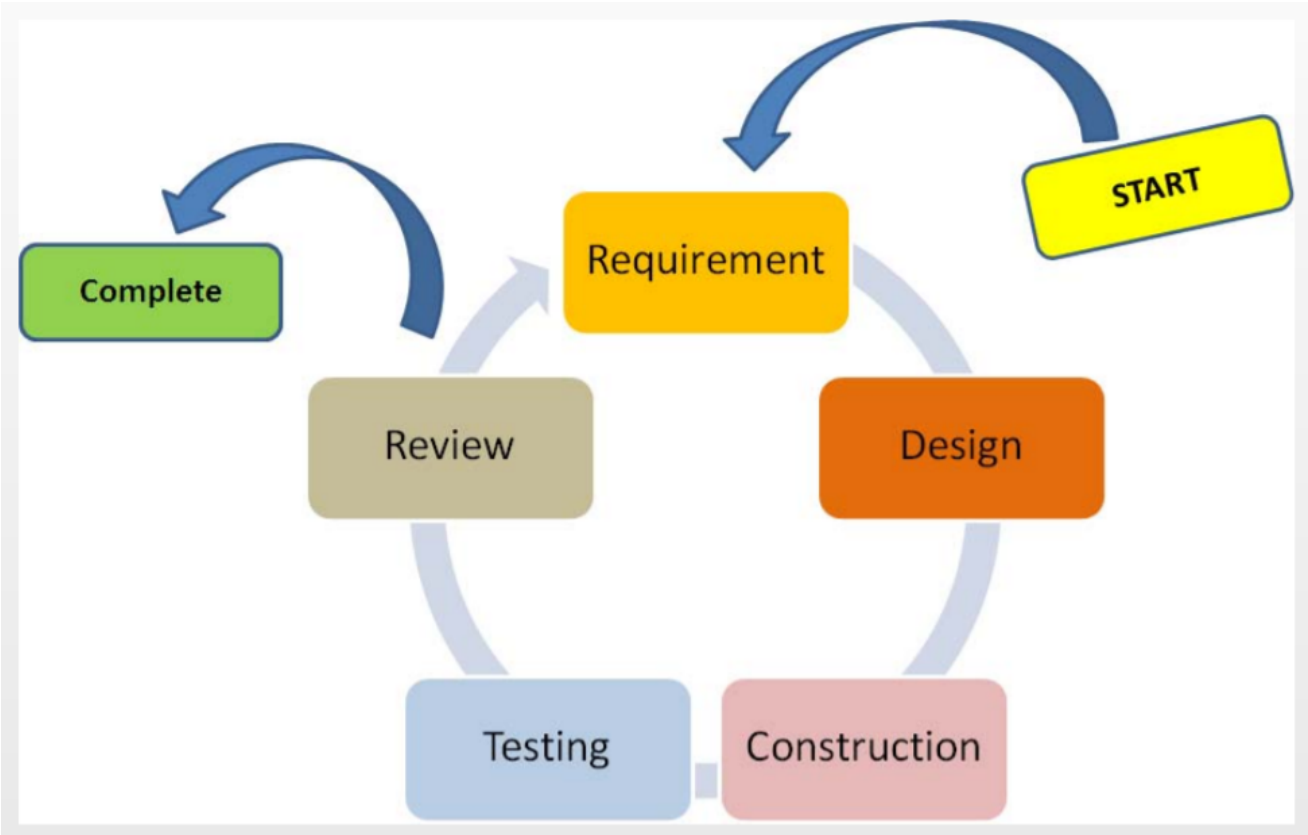
Waterfall

- Construction.
- Example: 8 queen problem.
- The problem can be specified precisely and in detail.
- Development through a number of phases (refinement).
- Each phase has a clear purpose.
- Challenges:
    - Based only on specifications but they are difficult to produce and understand.
    - Difficult to get the users to describe their work
    - Non-technical aspects are difficult to specify
    - Requirements are changing over time
    - Works only when we know exactly what we want and we are able to describe it precisely and unambiguously
    - Feedback loops become necessary
    - Many negative effects of the system developed



Iterative

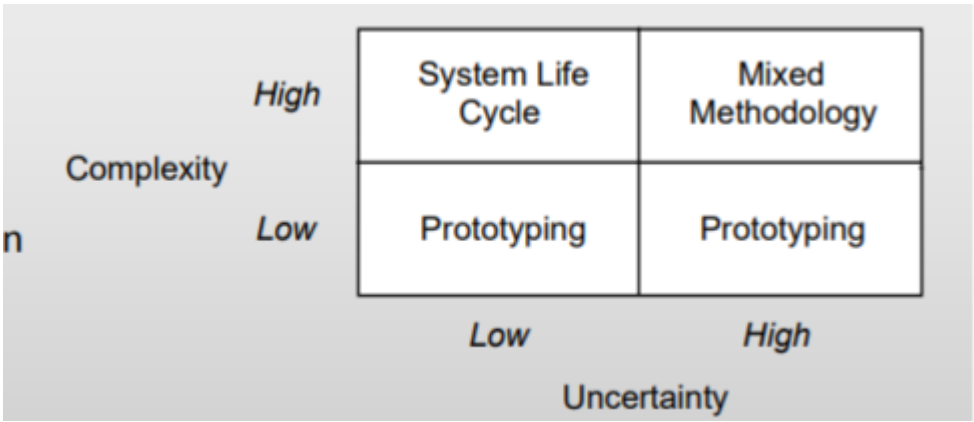
- Evolution with prototypes
- Example: support to nurses in patient care.
- No set of clear requirements to depart from.
- The understanding of the problem is changing during development
- Development through a series of cycles.
- The requirements and the system are improved in each iteration.



Contingency Theory

- How do you choose between waterfall/life cycle and prototyping?
- The relevance of these categories of methods can be determined from contingency factors.
- Analyze uncertainty in terms of the following four factors:
  - the organizational and technical context of the system
  - the future computer system
  - the experience and skills of the users
  - the experience and skills of the system developers
- Selection of approach:
  - If uncertainty is low: base requirements determination on an informal approach or on analysis of existing systems.
  - If uncertainty is high: use specifications or prototypes.
- Uncertainty:
  - the degree of structuredness that characterizes the users' work
  - the degree of understanding the users have about their work

    - the degree of experience and training of the system developers
- Complexity:
    - project size
    - number of users
    - volume of new information
    - complexity of new information



Too much quantity + too difficult quality = complexity.

Too little quantity + too unreliable quality = uncertainty.

- Principle of limited reduction:
    - Relying on an analytical mode of operation to reduce *complexity* introduces new sources of uncertainty requiring experimental countermeasures. = waterfall approach
    - Relying on an experimental mode of operation to reduce *uncertainty* introduces new sources of complexity requiring analytical countermeasures. = Iterative approach