Software Engineering

Lydie du Bousquet

Lydie.du-bousquet@imag.fr

In collaboration with J.-M. Favre, I. Parissis, Ph. Lalanda, Y. Ledru

Short introduction

- Lydie du Bousquet
  - Professor at UGA
  - *Software engineering, validation, test*

- Frédéric Lang
  - Researcher at INRIA
  - *Compositional Verification, New Generation Formal Description Techniques*

This class

- Introduction to software engineering as a tool box

Drill…


Mitre Saw


Sheet Sander


Orbital Sander with dust collection bag


Rechargeable Drill


Power Screwdriver


Circular Saw


Power Nailer/Stapler


Power Plane


Jigsaw


Power Drill


Palm Sander


Router


Oscillating Multi-Tool With Blade


Multi Purpose Saw


Circular Saw Blade

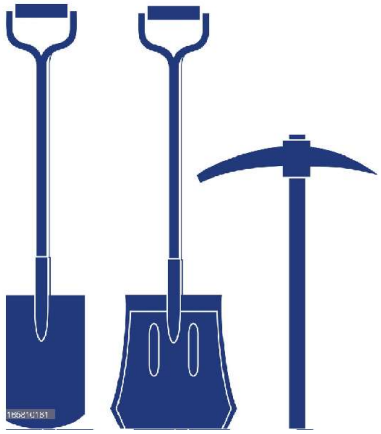

Lathe


Table saw


Drill Stand


Drill Press

Drill what ?



Before using a tool, you should :

- Know the **tools'characteristics**
- Know how to **use** them
- Be able to **choose** the appropriate one(s)

Class programm

- Software engineering and process (Week 1)
- Requirement engineering (Week 1)

Schedule

- What is Software Engineering?
- What are the activities during the development?
- How are they organized?
- Which process should you choose?

What is software Engineering?

Exercise 0, Q1

What is software Engineering?

- Engineering?
  - derived from the Latin
  - ***ingenium***, meaning "cleverness"
  - ***ingeniare***, meaning "to contrive, devise" (find a solution, build)
- Software Engineering
  - discipline that is concerned with all aspects of software production

Why Software Engineering?

- Provide systematic methods, tools
- To achieve
  - predictability
  - precision
  - mitigated risk
  - professionalism

Schedule

- What is Software Engineering?
- What are the activities during the development?
- How are they organized?
- Which process should you choose?

What are the activities during a software project?

Exercise 0, Q2

Classical activities

- Requirements Analysis
- Specification
- Software architecture
- Design
- Implementation
- Testing
- Documentation
- Installation, deployment
- Training and Support
- Maintenance

Development process

- Also known as
  - development methodology
  - software development life cycle,
  - software process
- Structure imposed on the development of a software product
  - organization of the tasks or activities that take place during the process
  - several models for such processes

What development process do you know ?

How the previous activities are organized?

Classical organizations

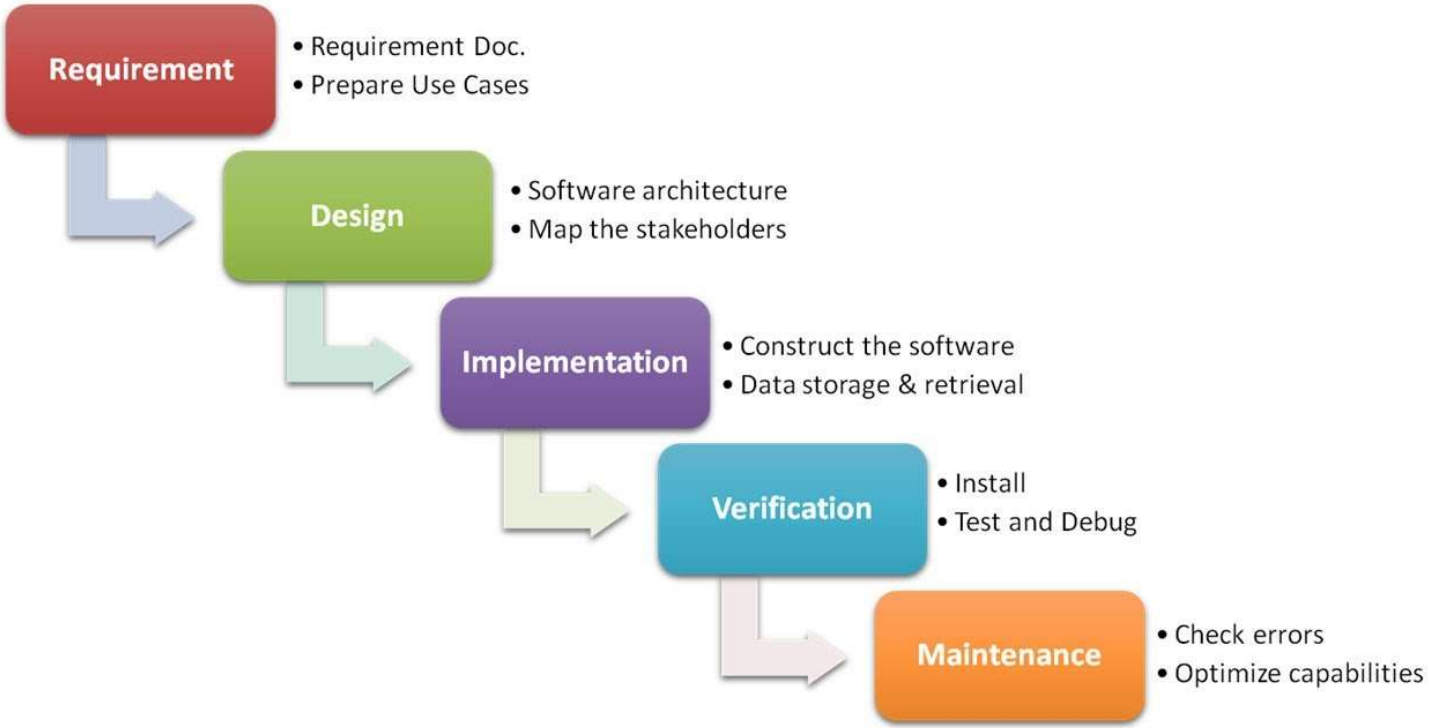
- Code and fix
- Waterfall development
- V-shaped Model
- Prototyping
- Incremental/iterative development
- Spiral
- Agile development

Code and fix

- Without much of a design in the way, programmers immediately begin producing code.
- At some point, testing begins (often late), unavoidable bugs must then be fixed before the product can be shipped.

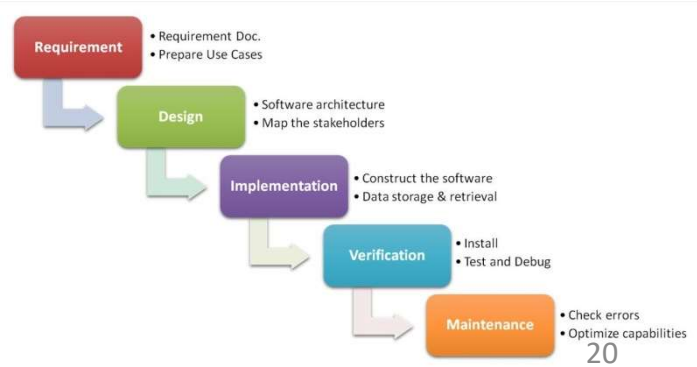
Waterfall development

- sequential development approach,
  in which development is seen as flowing steadily downwards through several phases

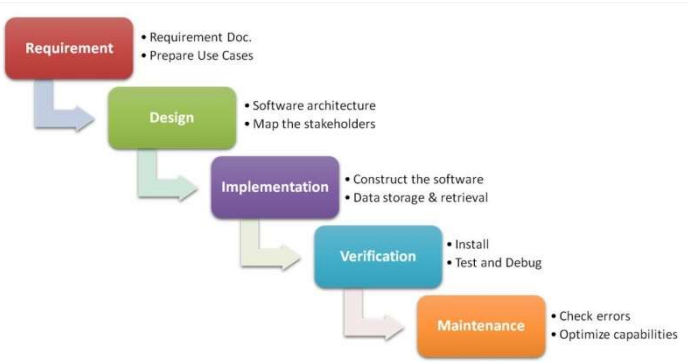


Waterfall development

- The idea behind:
  - **structured approach**: the current phase should be finished before starting a new one
  - identifiable **milestones**: each phase is documented and validated
- Criticisms
  - Clients may not know exactly what their requirements
  - Requirements may change
- Can be used
  - Projects with well-known requirements



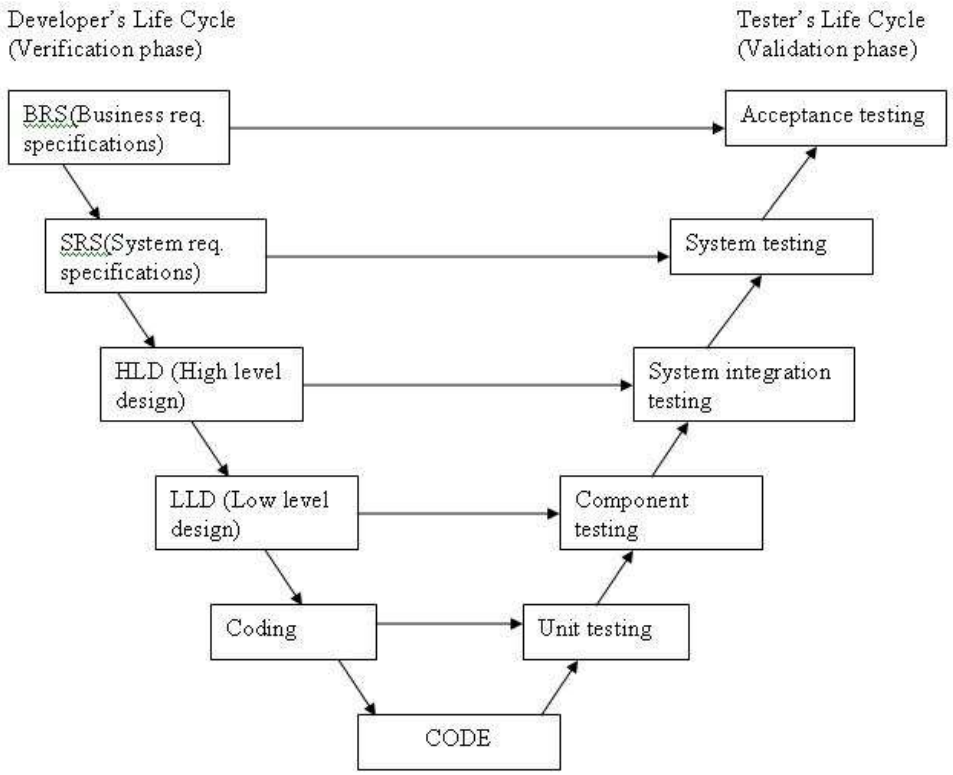
Waterfall development



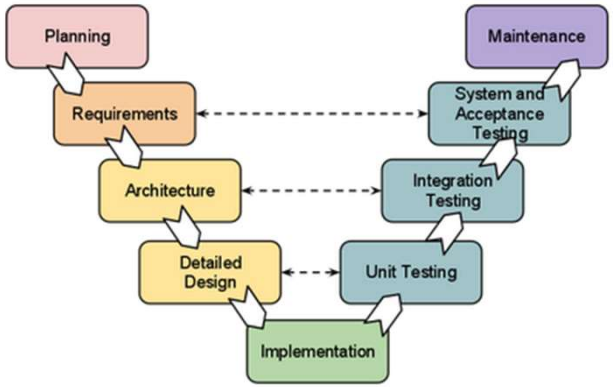
- **Advantages**
  - Each phase has specific **deliverables**
  - **Verification** at each stage for early detection of errors / misunderstanding
- **Disadvantages**
  - Assumes that the requirements are **frozen**
  - Very **difficult to go back** to any stage after it finished
  - Little flexibility and adjusting scope
  - Executable is only available at the end

V-shaped Model

- Like the waterfall model, it is a **sequential** path of execution of processes
- **Testing** of the product is planned in parallel with a corresponding phase of development



V-shaped Model



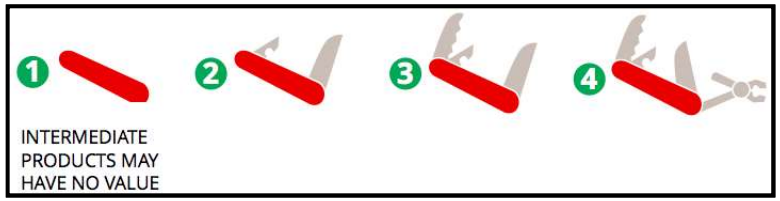
- The **idea** behind
  - Focusing on validation (testing, most of the time)
- **Advantages** of V-model
  - Simple and easy to use.
  - Testing happens well before/in parallel of coding.
- **Disadvantages** of V-model
  - Same as water-fall model
  - If any changes happen in midway, then the test documents along with requirement documents have to be updated.
- When to use the V-model
  - For projects where requirements are easily understood
  - When validation is a key point

Incremental/iterative development

- Objective: reduce risks by breaking a project into smaller parts and providing more ease-of-change during the development
- Different possibilities:
  - A series of mini-Waterfalls are performed. All phases of the Waterfall are completed for a small part, before proceeding to the next increment,
  - Overall requirements are defined before proceeding to evolutionary, mini-Waterfall development of individual increments of a system, or
  - Requirements analysis, and design of architecture and system core are defined via Waterfall, followed by iterative prototyping for different parts.

Incremental development

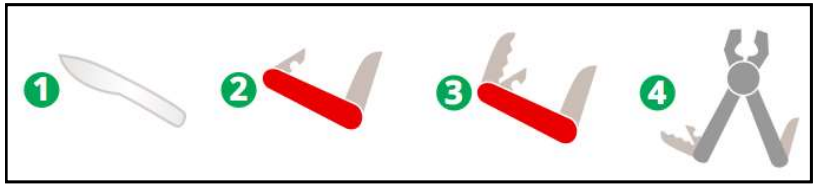
- **Principles**:
  - Requirements are globally collected,
  - Development is sliced into parts
  - Parts are developed at various times and integrated to obtain the releases
  - Each increment adds more software value – like Adding package



- **Advantages**
  - First release can be delivered sooner than in a waterfall process
  - Major parts can be developed first, the order of others can be modify
  - Users can provide feedbacks to adjust requirements of the release
  - Easier to test and easier to manage risks than waterfall
- **Disadvantages**
  - Architecture generally chosen at the beginning => needs a complete definition of the whole system before it can be broken down into parts
  - Total cost is higher than waterfall

Illustration from https://productcoalition.com/product-development-using-agile-methodology-446c01ecd510

Iterative development



- **Principles**
  - Build something,
  - Get some feedback and refine it to make better,
  - Keep doing that until the product is good
- **Advantages**
  - Reduce rarely used features, maximize frequently used features
  - Usable product at any time
  - Suited for Agile Organizations
- **Disadvantages**
  - Increased Pressure on User Engagement
  - Each phase of an iteration is rigid with no overlaps
  - Requires higher level of technical excellence (than other process)
  - Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

Illustration from https://productcoalition.com/product-development-using-agile-methodology-446c01ecd510

Incremental vs iterative development

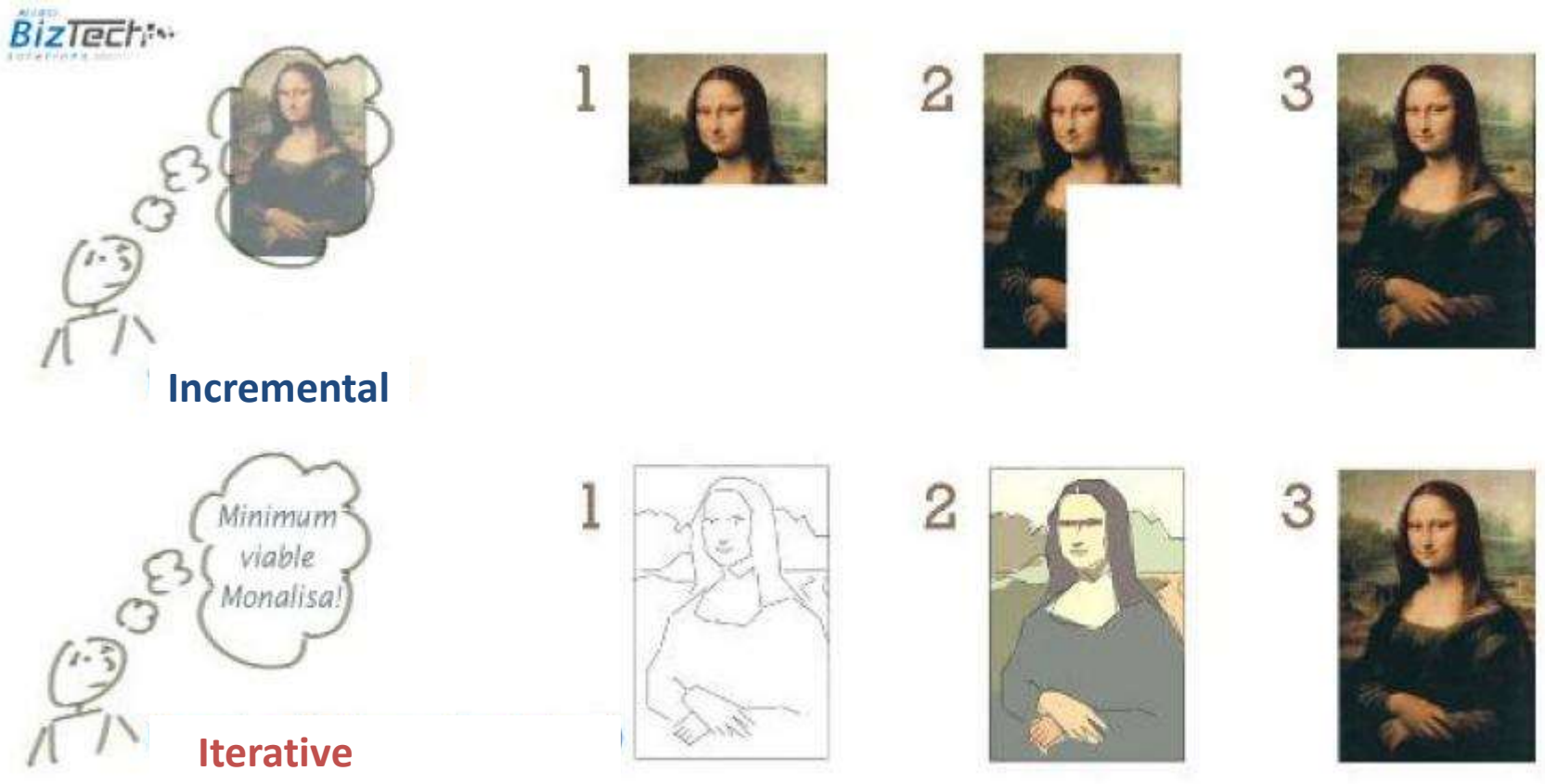


Illustration from http://www.golali.co/waterfall-vs-iterative-model/

Incremental vs **iterative** development

- The word **increment** fundamentally means **add onto**
- The word **iterate** fundamentally means **re-do**

- For **incremental**,
  - you need to have well defined requirements
  - Between each increment, you can adjust part of the requirements
  - Several increments can be produced in parallel

- For **iterative**,
  - You start with a set of requirements in order to produce a first delivery
  - At each iteration, you collect new requirements and you make a new version

Agile development

- Mainly an iterative development processes
  - Rapid cycles (1 to 3 weeks)
  - Small release
- Example of agile methods
  - XP (Extreme Programming)
  - Scrum
  - …

Agile development: 12 principles

- **Customer satisfaction** by early and continuous delivery of useful software
- Welcome **changing requirements**, even in late development
- Working software is delivered frequently (weeks rather than months)
- Close, **daily cooperation** between business people and developers
- Projects are built around motivated individuals, who should be trusted
- **Face-to-face conversation** is the best form of communication (co-location)
- **Working software** is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Continuous attention to **technical excellence** and good design
- **Simplicity** (maximizing the amount of work not done) is essential
- Self-organizing teams
- Regular **adaptation** to changing circumstance

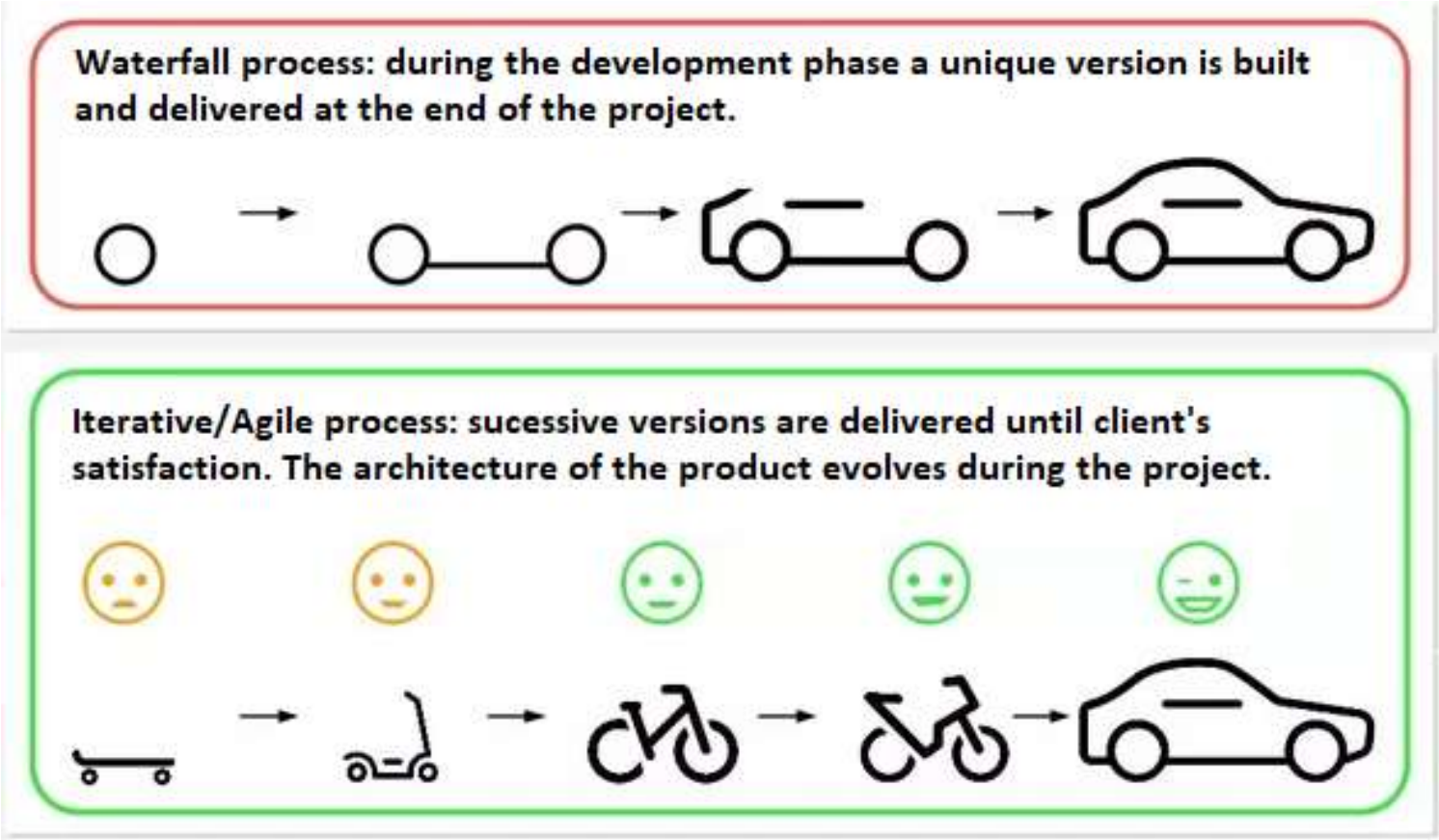
Agile processes:

- **Advantages**
  - **Customer satisfaction** by rapid, continuous delivery of useful software
  - **People and interactions** are emphasized rather than process and tools
  - Customers, developers and testers **constantly interact** with each other
  - Working software is **delivered frequently** (weeks rather than months)
  - Face-to-face conversation is the best form of communication
  - Close, **daily cooperation** between business people and developers
  - Continuous attention to **technical excellence** and good design
  - Regular adaptation to changing circumstances
  - Even late changes in requirements are welcomed

Agile processes

- **Disadvantages**
  - Difficult to **assess the effort** required at the beginning of the dev. life cycle.
  - There is **lack of emphasis** on necessary **designing** and **documentation**.
  - The project can easily get taken off track
- When to use Agile processes:
  - When new changes are needed to be implemented.
  - End users' needs are ever changing in a dynamic business and IT world

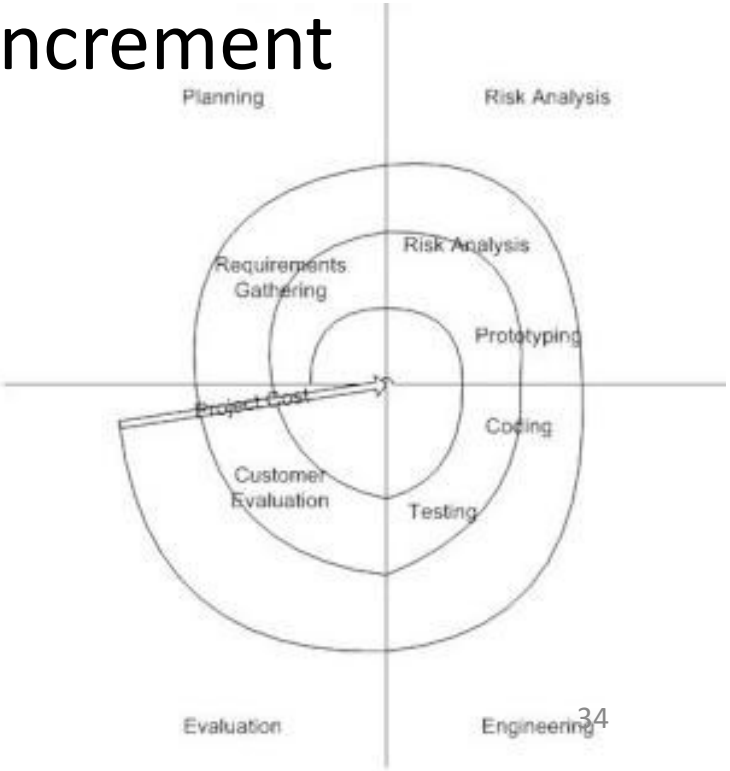
Waterfall vs Iterative/Agile



Adapted from an image of http://www.golali.co/waterfall-vs-iterative-model/

Spiral development model

- Risk-driven process model
  - Similar to the incremental model,
  - with more emphasis placed on risk analysis.
- Each spiral corresponds to a increment
- Each spiral has four phases:
  - Planning,
  - Risk Analysis,
  - Engineering and
  - Evaluation



Spiral development model

- **Advantages** of Spiral model
  - High amount of risk analysis hence, avoidance of Risk is enhanced
  - Strong approval and documentation control
  - Additional Functionality can be added at a later date
- **Disadvantages** of Spiral model
  - Can be a costly model to use.
  - Risk analysis requires highly specific expertise.
  - Project's success is highly dependent on the risk analysis
  - Doesn't work well for smaller projects.

Spiral development model

- **When** to use Spiral model?
  - When costs and risk evaluation is important
  - For medium to high-risk projects
  - Users are unsure of their needs
  - Requirements are complex
  - Significant changes are expected (research and exploration)

Prototyping (activity)

- Activity of creating **prototypes** of the application, i.e., **incomplete** versions of the product
  - simulates only a few aspects of the final product
- **Outline**
  - Identify basic requirements
  - Develop an Initial Prototype
  - Review of the customers, including end-users, to examine the prototype and to provide feedback on additions or changes.
  - Revise and Enhance the specification / prototype using the feedback

Prototyping (activity)

- The **idea** behind
  - **Not** a standalone, complete development methodology
  - Attempts to reduce inherent project risk
  - User is involved throughout the development process
- Example of usages
  - Understanding of user requirements
  - Checking some technology

Prototyping (activity)

- **Advantages** of prototyping
  - Reduced time and costs
  - Improved and increased user involvement
- **Risks** using of prototyping
  - Insufficient analysis
  - Prototype vs finished system
  - Excessive development time of the prototype

Choice of a development process

- Learn the about the development processes
- Assess the needs of Stakeholders
- Use the criteria

Factors	Waterfall	V-shaped	Proto-type	Iter & increm	Agile
Unclear User Requirement	**Poor**	**Poor**	Good	Good	**Excellent**
Complex System	Good	Good	**Excellent**	Good	**Poor**
Reliable system	Good	Good	-	Good	Good
Documentations	**Excellent**	**Excellent**	-	**Excellent**	**Poor**
Component reusability	**Excellent**	**Excellent**	**Poor**	**Excellent**	**Poor**

Another comparison

Feature	Waterfall Model	Prototype Model	Spiral Model	Iterative Model	Agile Model
Requirement Specifications	Beginning	Frequently Changed	Beginning	Beginning	Frequently Changed
Understanding Requirements	Well Understood	Not Well understood	Well Understood	Not Well understood	Well understood
Cost	Low	High	Intermediate	Low	Very high
Guarantee of Success	Low	Good	High	High	Very high
Resource Control	Yes	No	Yes	Yes	No
Cost Control	Yes	No	Yes	No	Yes
Simplicity	Simple	Simple	Intermediate	Intermediate	Complex
Risk Involvement	High	High	Low	Intermediate	Moderate
Expertise Required	High	Medium	High	High	Very high
Changes Incorporated	Difficult	Easy	Easy	Easy	difficult
Risk Analysis	Only at beginning	No Risk Analysis	Yes	No	yes

https://www.semanticscholar.org/paper/Prototype-centric-(-PC-)-software-development-model-Munassar/12a4cc1c0e369162affe647296c830155d9aa416/figure/5

Exercise 3 for
Which development process?

- System for student management in a university (this system replace an existing system without any functional evolutions)
- An new interactive system for travelers to have schedules on their smartphones
- A system to control subway without drivers
- A very large system for Air traffic management
- A very new 3D-system for software maintenance
- Infrastructure and services for city that wants to become a “smart-city”