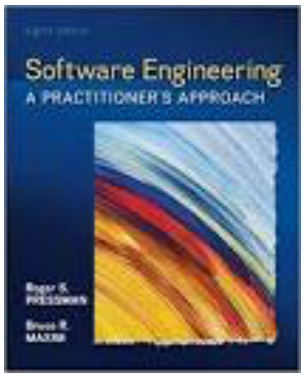
Introduction to Software Engineering

UA.DETI.IES - 2019/20

Resources & Credits

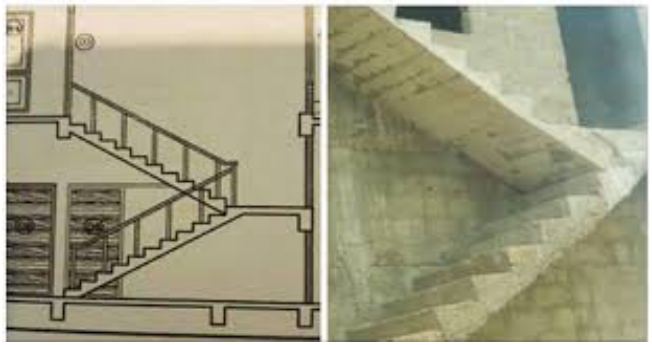


- Roger S. Pressman, Bruce Maxim, Software Engineering: A Practitioner's Approach, 7th Edition, McGraw-Hill Education, 2015

- Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2016

- Grady Booch, "The History of Software Engineering." IEEE Software 35.5, 2018

Importance of Engineering Processes



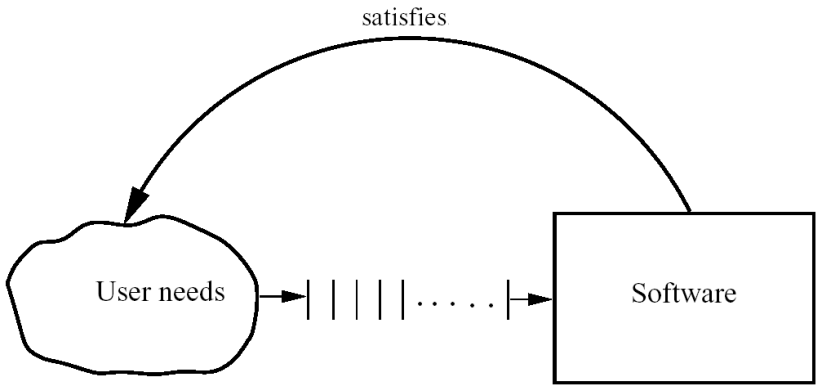
Common Engineers' roles

- Research and development
  - To enhance current processes and stimulate productivity.
- Monitoring & troubleshooting
  - To ensure that systems are functioning correctly.
- Documentation
  - To ensure that all new personnel can understand the best practices and standards of each individual process.
- Risk assessments
  - To determine risk of malfunction, failure, personal, etc.
- Management roles
  - May have responsibilities over operations and personnel.
- Budgeting responsibilities

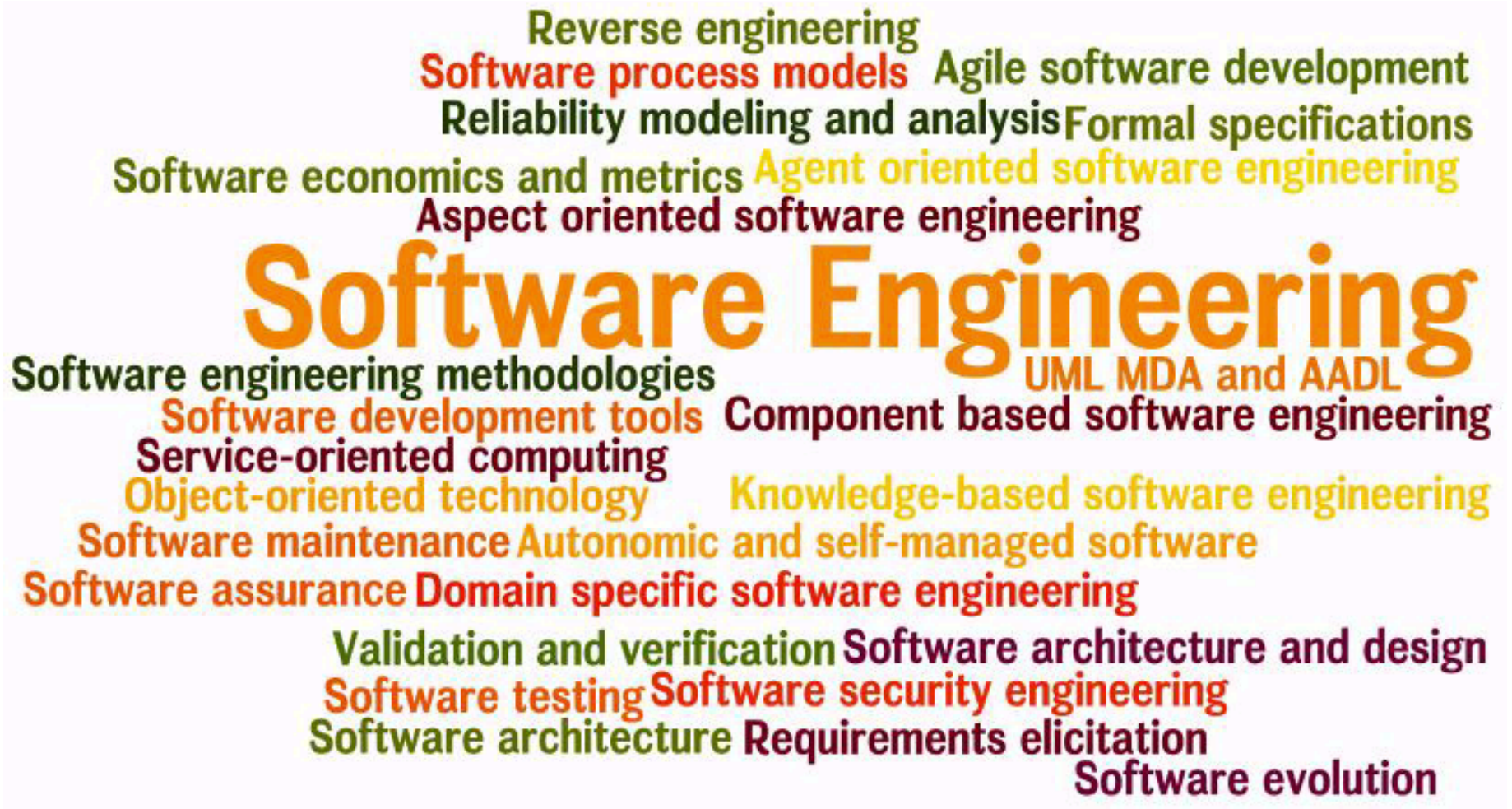
What is Software Engineering?

- IEEE definition:
  - The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software;
  - that is, **the application of engineering to software**.
- Roger S. Pressman's definition:
  - Software engineering is the technology that encompasses a **process**, a set of **methods** and an array of **tools**.

(Software Engineering: A Practitioner's Approach)

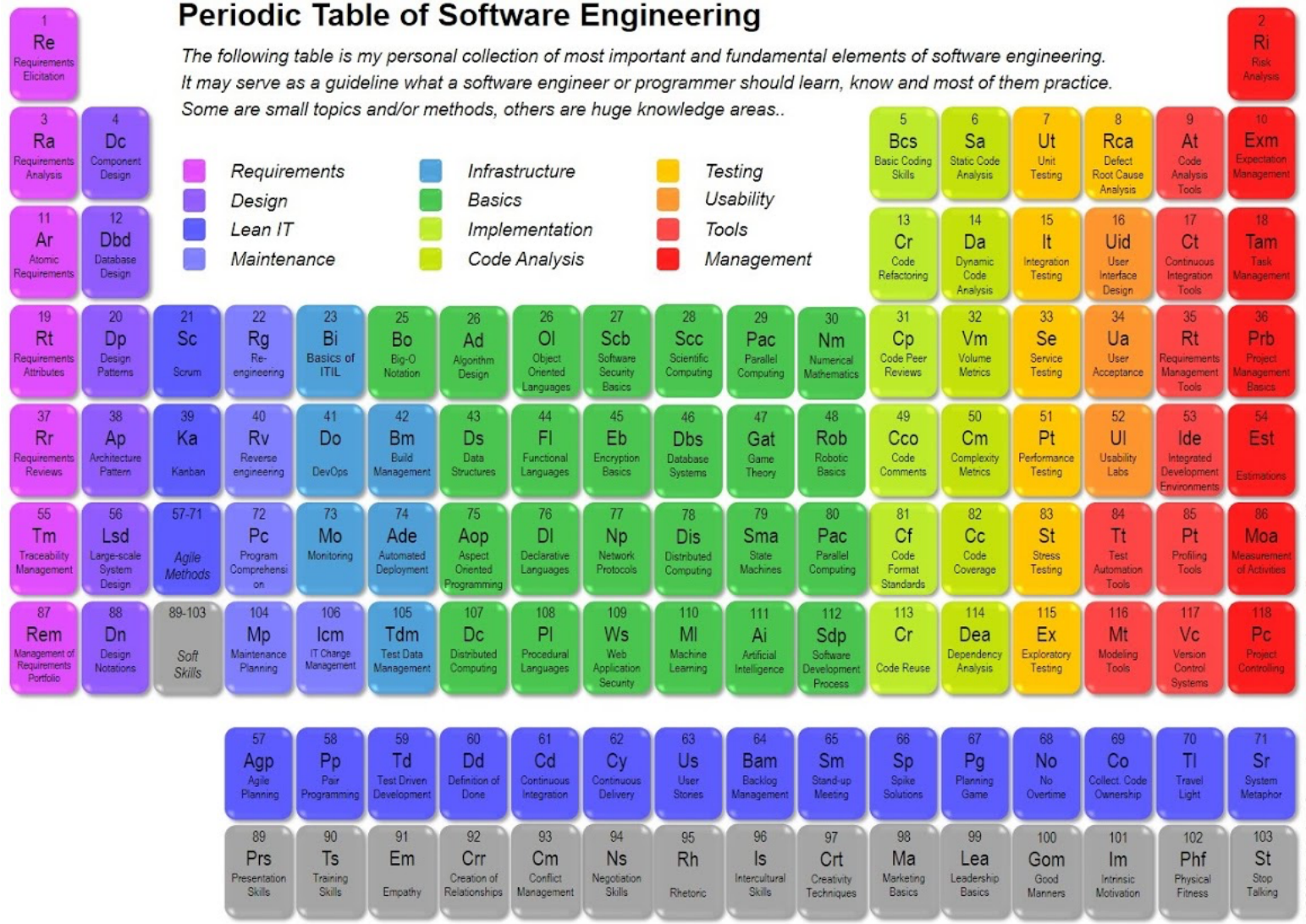


What is Software Engineering?



https://wonderfulengineering.com/what-is-software-engineering/

What is Software Engineering?

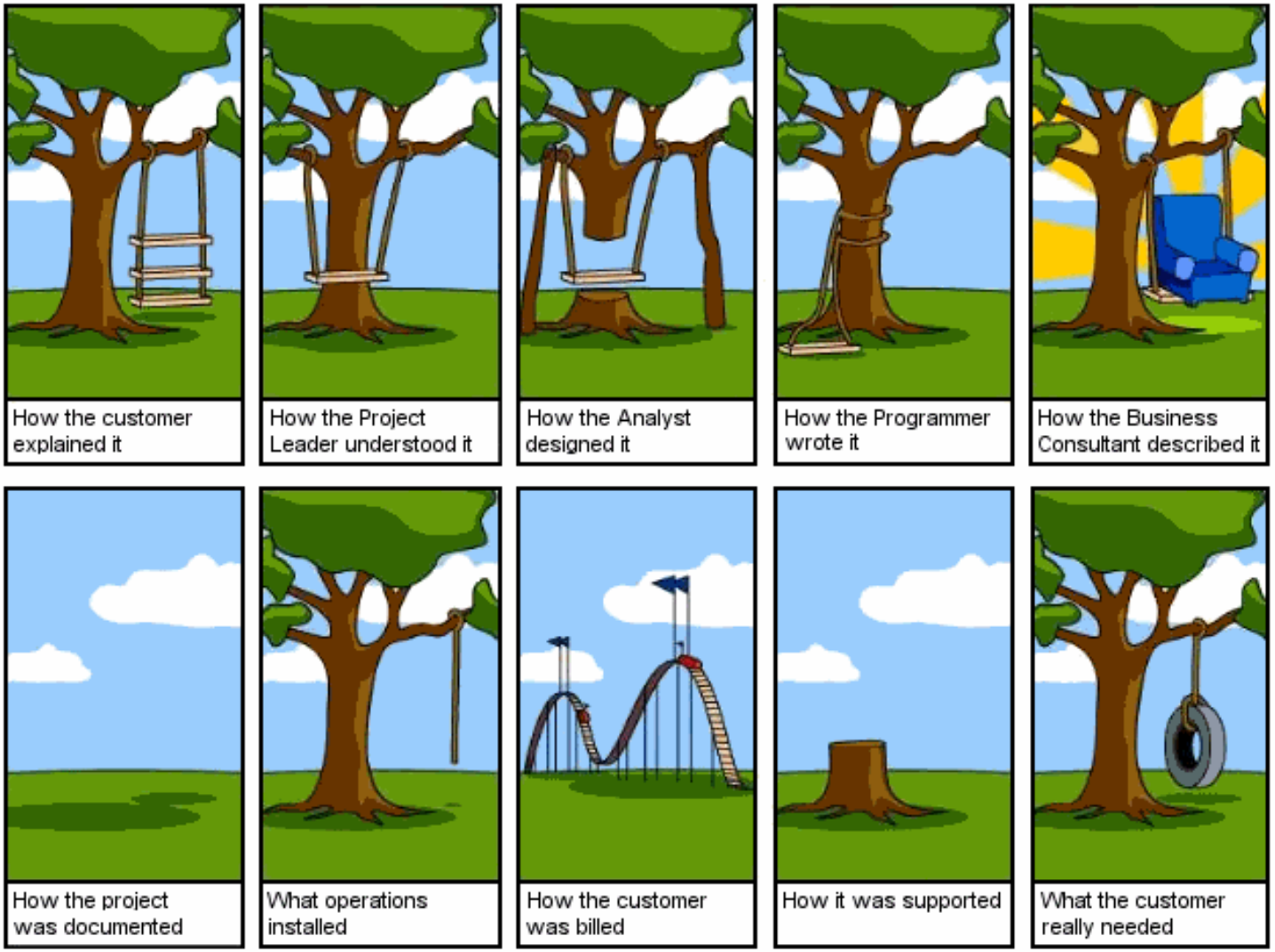


https://www.sw-engineering-candies.com/blog-1/periodic-table-of-software-engineering-know-how

Importance of Software Engineering

- **Reduces complexity**
  - SE divides big problems into several small problems, which are solved independently to each other.
- **Minimizes risks, software cost, and delivery time**
  - Highly paid professionals are needed to develop software with millions lines of code. Good planning and good practice reduce efforts, costs and delivery time.
- **Allow managing large projects**
  - To invest months/years in a product requires lots of planning, direction, testing and maintenance.
- **Enforces development of reliable software**
  - Testing and maintenance are key aspects addressed in SE
- **Standardize processes**
  - Software standards are critical to produce more effective software.

Importance of Software Engineering



http://projectcartoon.com

Importance of Software Engineering

Academic

- Developer is the user
  - Works most of the time
  - Bugs are tolerable
  - UI not important
  - No documentation
- SW not in critical use
- Reliability not important
- No investment
- Don't care about portability
- Most of the cases: "present and forget"

Industrial

- Others are the users
  - Works robustly
  - Bugs not tolerated
  - UI very important issue
  - Documents needed
- Supports business
- Reliability is key
- Heavy investment
- Portability is a key issue
- Evolve the product through years

Software is Expensive?

- Rough cost estimate…
  - Productivity = 500 LOC/PM (Lines Of Code / Person Month)
  - Cost to the company = 5K€/PM
  - So each line of delivered code costs about 10€
- A simple application for a business may have 10KLOC to 50KLOC
  - Cost = 100K€ to 500K€
  - Can easily run on less then 10K€ hardware
  - So HW costs <<< SW costs

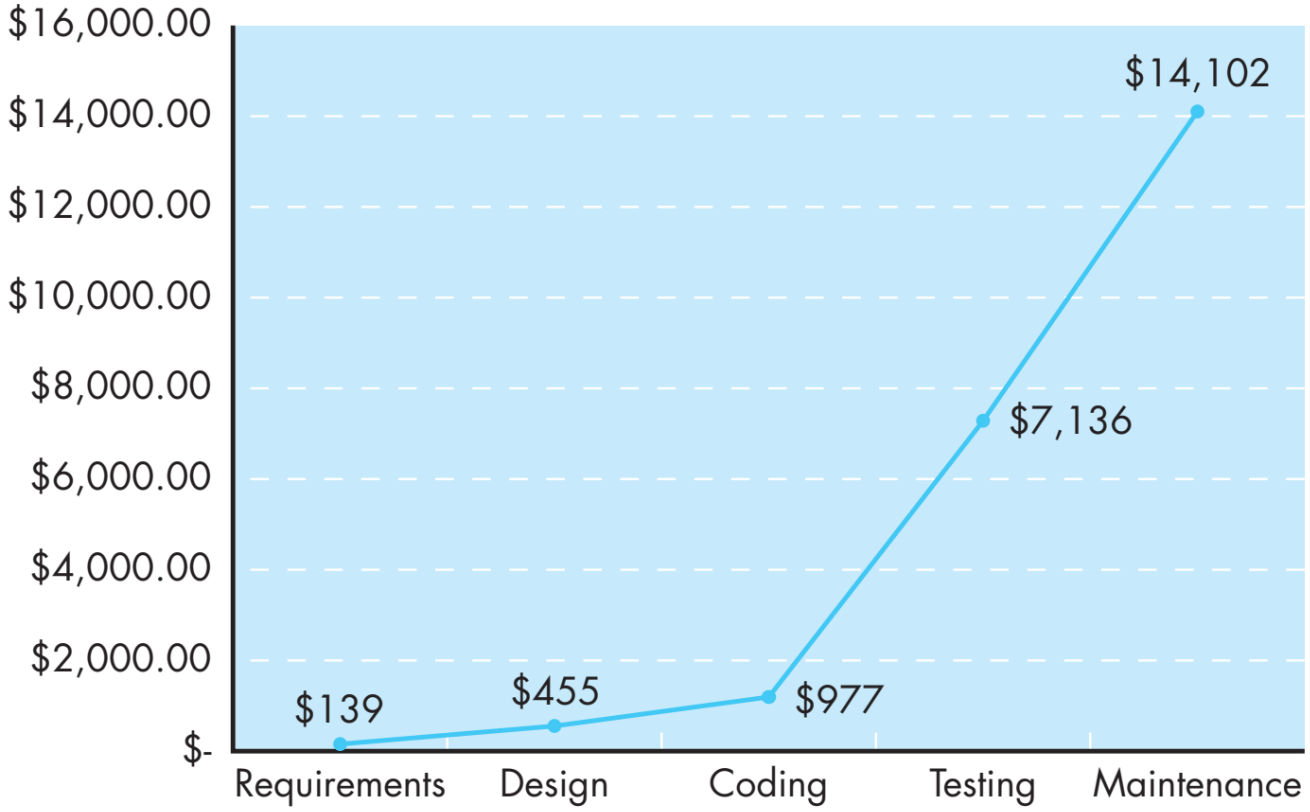
Software is Expensive?

- SW failures are different from failures of mechanical or electrical systems
  - In software, failures are not due to aging related problems. They occur due to errors introduced during development
- Once SW delivered, it enters maintenance phase
  - Residual errors requiring corrective maintenance
  - Upgrades and environment changes – adaptive maintenance
- Over SW lifetime, maintenance can cost more than the development cost of SW

Cost of correction along the lifecycle

FIGURE 19.2

Relative cost of correcting errors and defects
Source: Adapted from [Boe01b].



Software Engineering Methodology

- SE focuses mostly on processes for achieving goals
  - Process must be systematic
  - Design of proper processes and their control is a key challenge SE faces
- SE separates process for developing sw from the developed product (i.e the sw)
  - SW process is the equivalent of manufacturing process
- The development process is typically phased
  - Phases separate concerns with each phase focusing on some aspect
    - Requirements, architecture, design, coding, testing are key phases
  - This phased process has to be properly managed to achieve the objectives
    - Metrics and measurement important for this

Software Architect role

- Makes high-level design choices and dictates technical standards, including software coding standards, tools, and platforms.

Valentina Sosa, The Architect Manifest: Principles of design in agile environments, O'Reilly Software Architecture

Software Architect skills

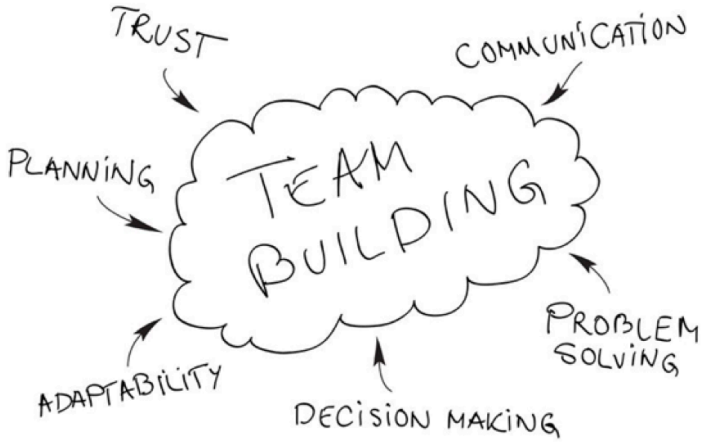
- Leader
- Problem solver and analyst
- Communicator
- Technologist
- Researcher
- Software Designer
- Promoter of Good Practices

Software Engineering phases

Matching the organization culture

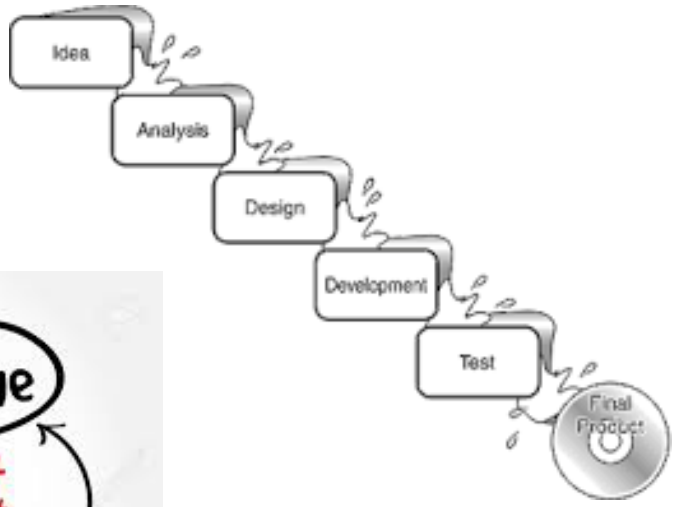
Working together with the team

- Discuss your ideas with the team
- Be open minded towards new ideas
- Propose your ideas in a storytelling fashion
- Your objective is to have a successful project
- Be prepared to give up your ideas



Other key issues

- Methods
- Time management
- Efficiency
- Quality
- Ethics



Summary

- The problem domain for software engineering is industrial strength software
- Software comprises programs, documentation, and data
- Software engineering aims to provide methods for systematically developing high quality products with the high productivity and low costs