SENG 471

Software Requirement Engineering

What is Engineering?

Engineering is the development of cost-effective solutions to practical problems, through the application of scientific knowledge.

Cost-effective solutions → ↓cost, emphasis on building

Practical Problems → problems should matter to people

Application of scientific knowledge → Apply systematic techniques

Quality

Cost ↓ Efficiency ↑, Usability ↑, Re-usability ↑

What is Software Engineering?

The application of engineering principles to the development of software in a systematic method.

- Hard Skills: Programming, Analytical Techniques, Design Techniques
- Soft Skills: Personality qualities that positively affect how you work and interact with others, that includes: communication, teamwork, adaptability, creativity

What is Software?

Not just programming. It is an engineering product that is freed from almost all the limitations of the physical world.

- It does not cost almost anything to reproduce once it has been designed & implemented. It also solves difficult problems perfectly. And it does not wear out.
  - Most of the industrial-strength software products have a long-lasting lifecycle. Once they are born, they continue to grow.

What is Software?

- Starts with an idea that needs feasibility studies, analytical studies, followed by design and constructions.
- A successful software product is subject to change, and every new change should go through the same development process

- Software is not just an application sitting on your computer or laptop.

General and Common Forms of Software

Word processors, Spreadsheets, email & chat applications, Transaction Processing Systems (TPS).

Decision Support Tool: This type of software allows for a low cost analysis prior to implementation of costly projects.

And many more.

Software is integrated in almost every industry

Manufacturing, Oil & Gas, Transportation, Biomedical, Robotics, Management & Decision Making, Games & Animations, Music Industry.

Software is deeply embedded in most devices. which means the value of those products is affected by the software quality.

Examples: Cars fuel and brake systems, airplanes, fighter jets, surgical tool, appliances, cell phones, etc.

Other industries → nuclear plants, defense systems

Quality Requirements

Is software quality as Important as quality for other Engineering Products?

- A large portion of engineering budgets are reserved for Product Quality and Quality Assurance.
- Managers easily approve costs associated with quality and safety of an engineering product such as bridges, tunnel, etc.
- Dangers can be easily associated with tangible products, what about dangers of poorly designed software?

There are a lot of examples of fatal situations that have occurred due to software errors.

Software Development Process

- There are different approaches to software development:
  - some use a more structured, and engineering-based approach
  - others use more incremental approach, where software evolves as it is developed piece by piece.

- Most of the commonly used software development methods use a combination of the following development phase:
  - Inception phase
  - Requirement Analysis phase
  - Design phase
  - Construction phase
  - Testing and quality control phase
  - Maintainance phase
- The process of engineering a software product is not much different from engineering of other products.

Inception Phase

- Main Objective:
  - Scope the system adequately as a basis for validating initial costs and budgets (Feasibility and Risk Analysis)
  - Seeking possible alternatives

Requirements Analysis Phase

During the Requirement Analysis phase, you will first collect adequate information about the product that you are going to build, and you will try to find out what are the system requirements. For example, who are the users of the system, what is the scope system, what are the inputs & outputs of the system.

Design Phase

During the Design Phase, based on collected detail information answers to the raised questions, you will try to come up with a solution (design), and the system architecture. In this phase you have to consider the system's constraints and requirements and find the best solution.

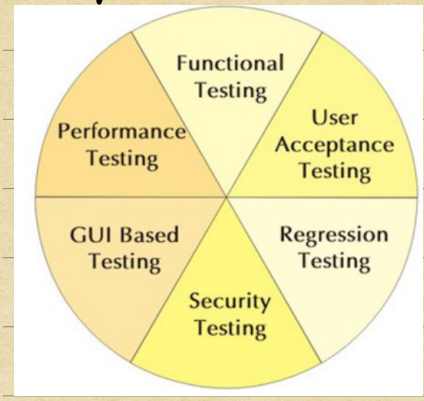
Outcomes:

- System prototype
- System architecture
- System's blueprints

Construction Phase

During this phase based on blueprints provided by design engineers, programmers start gluing the pieces of the puzzle and gluing them together. This stage of the development may take several months or even years, until the full-blown version of the product becomes ready.

Testing and Quality Control



- The building blocks of the software must be tested during the entire build process, at different levels (system level, subsystem level, component level, object or function levels)

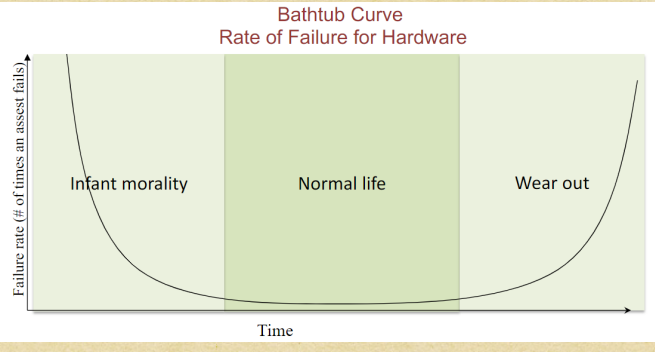
Maintenance Phase

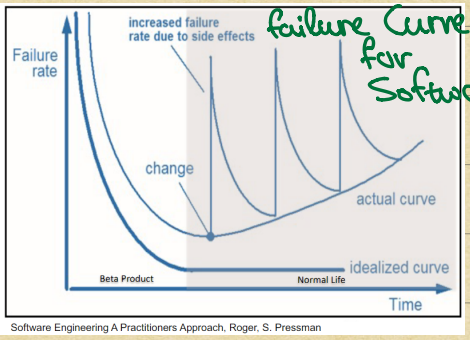
- Once software is created, constructed and deployed, is always subject to modification and changes.
- Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.
- Categories of software maintenance:
  - Corrective maintenance: to fix the errors of a software product discovered after delivery.
  - Adaptive maintenance: to keep a software product usable in a changed or changing environment.
  - Perfective maintenance: to improve performance or maintenability.
  - Preventative maintenance: To detect and correct latent faults in the software product before they become effective faults.

Software vs Hardware

Failure Curve for Hardware

- Bathtub cure showing the rate for failure.
- This rate represents failure rate for a population of hardware.





failure Curve for Software

Software Engineering A Practitioners Approach, Roger, S. Pressman

Software Systems Classifications

- **Information Systems:**
  - Deal with large amounts of data.
  - Have complex storage or file structures
  - Data sources and destinations tend to be people and other systems
  - Have fairly simple procedural details (i.e. create, report, update, delete).
  - Used in businesses and similar organizations
- **Technical Systems**
  - Deal with small amounts of stored data.
  - Have complex procedural details.
  - Data sources & destinations tend to be people & devices
  - Often used for device control, or in science & engg applications
- **Embedded systems:**
  - Mostly signal processing with very small amount of flow of data.
  - Real-time computing constraints.
  - Data sources and destinations tend to be mostly devices or machines.
  - Used for device control & automations

Software Project Management

1) Involves planning, monitoring, and control of the people, process, and events that occur as software evolves from a preliminary concept to an operational implementation.
2) Effective SPM focuses on 4 P's: People, Product, Process & Project

People

- In an enterprise application or industrial-strength software, there are normally many people involved. Examples:
  - Product Owner: Customers/Clients/Users
  - Management: Product Manager, Project Manager
  - Business Analyst
  - Development Team
    - System Analysts
    - System Designers/Architects
    - Programmers
    - Test Engineers
    - Quality Assurance
  - Operation Staff
  - External Standards Personnel

Note: Business analysis is a wider concept that covers the entire organization, where system analysis focuses on the aspects of a particular product.

Myth: Adding more people if we fall behind schedule will help us catch up (Mongolian Horde Concept). Usually, adding more people to a late software project delays the project even more.

Process

Introduction to Software Development Process

Simplified Linear Model of Software Development Process

Inception → Knowledge Elicitation → Requirement Analysis → Design → Implementation → Testing

Management Myths

We shouldn't change the way we do things since we do the same things as years ago. The demand and importance of software have increased greatly. The presence of modern software development process models such as Agile Development, renders activities such as Requirements Analysis & Software Design obselete (myth).

Project

Some Sources of Project Failure

- Projects are often undertaken with only a vague indication of requirements.
- Schedule and costs estimates are often grossly inaccurate
- The productivity of software developers does not keep up with the demand for their services.

Project Management Myths

- Analysis Paralysis: Why should we waste our budget for tedious and expensive activities of analysis and design.
- Once the program is written and it works we are done.

→ Definition, Development, maintenance → 50% → 70% of effort after the product is delivered to the client for the first time.

Software Requirements

- The definition of what the product should do. Includes graphical models and textual description of the product.
- Hardest part is deciding precisely what to build.
- What system should we build?
  - Understand, analyze, and build the systems requirements models, and focus on 4-Ws:
    - What problem to solve
    - Why such a problem should be solved
    - Who should be involved.
    - And how-good should be built.

General Challenges

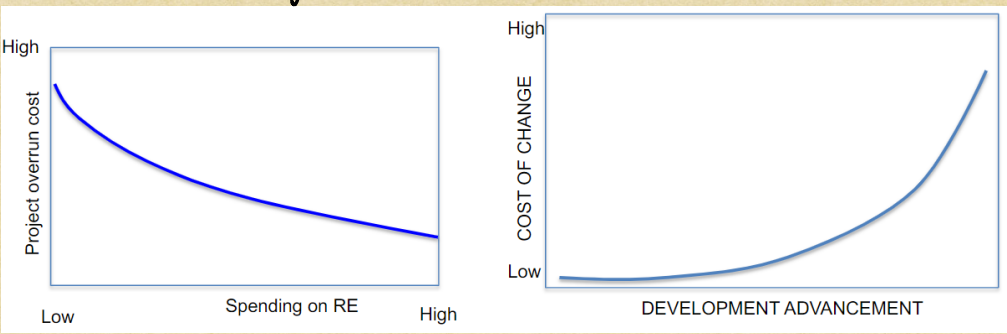
- Understanding of each domain.
- Variety of stakeholders in each domain and understanding their needs.
- Requirements conflicts between and within each domain.
- Knowledge transfer and updating team members and managers.
- Legals and political constraints.
- Continuous change of requirements.
- High expectations from stakeholders and customers.
- Technical Challenges.

RE Specific Challenges

- Process Management, diverse perspectives among users and developers, coping with rapid tech changes, requirements change, requirements conflicts, high-user expectations, Problem-domain & its complexity.

Why is Requirement Analysis is important

- Alternative ways to handle the same problem
- Possibility to break the traditions
- Volume of the requirement
- Requirements conflicts
- Must be measurable and testable
- Must be documented carefully to be sufficient for system design.
- Time and budget limitations.



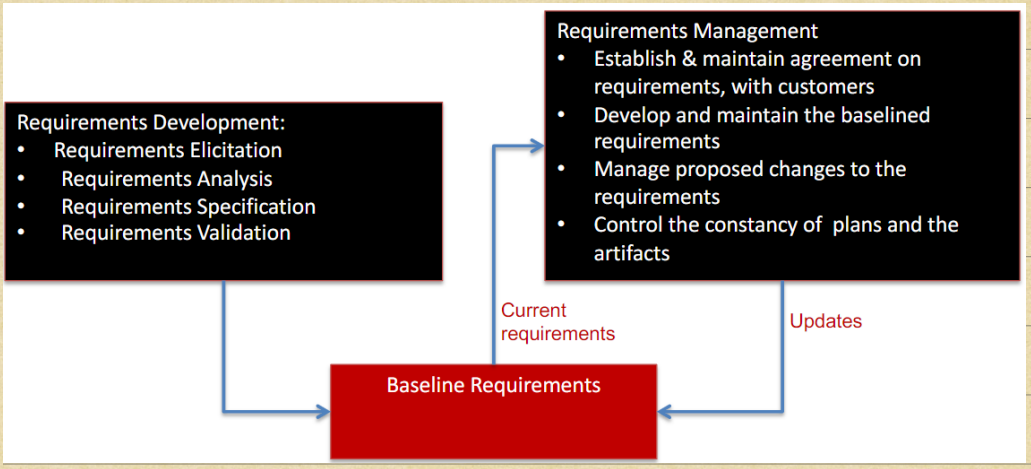
The RE Process

- The requirement engineering process includes two stages and each stage include several steps:
  - Requirement Development
  - Requirement Management

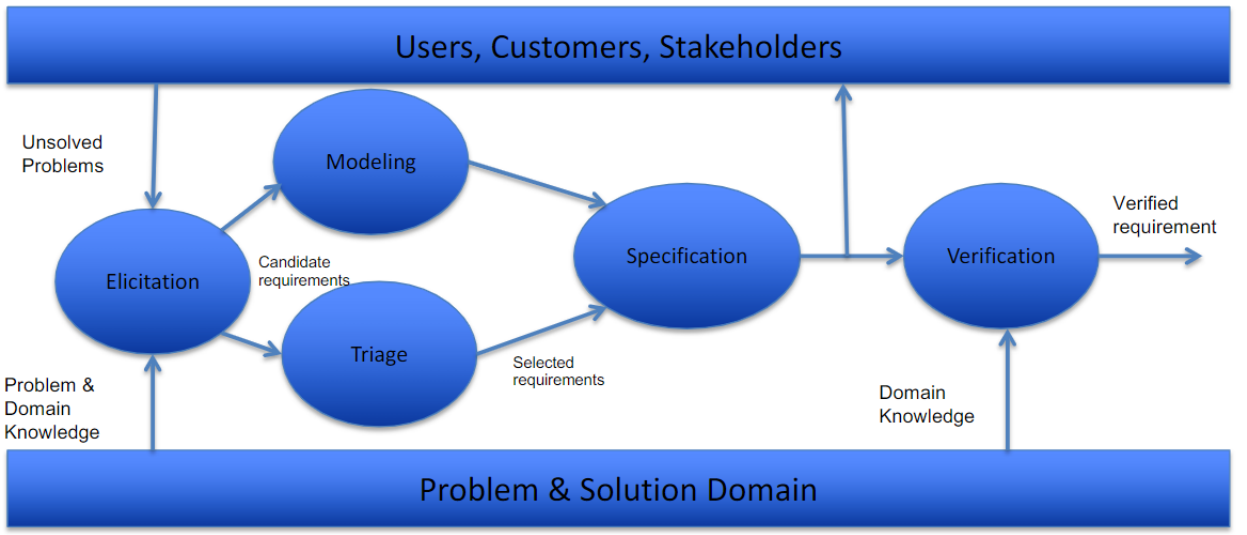
- Requirement Engineering encompasses the activities involved in eliciting, analyzing, specifying, and validating the requirements.

  Two areas: Requirements Development, and Requirements Management. Both sets of activities target to establish or evolve the Requirements Baseline.

- Requirements Baseline: A snapshot in time that represents an agreed-upon, reviewed, and approved set of requirements that have been committed to a specific product release. That release could be a complete delivered or any interim development increment of the product.



Flow of Activities in Requirements Engineering



- Elicitation: Learning, uncovering, extracting, surfacing, and/or discovering needs of customers, users, and other potential stakeholders.
- Modeling: Creating and analyzing models of requirements, with the goals of increasing understanding and searching for incompleteness and inconsistency. (Different from design phase)
- Triage (Assigning degree of urgency): Determining which subset of the requirements learned by elicitation are appropriate to be addressed in specific releases of a system.
- Specification: The documentation of the desired external behavior of a system.
- Verification: Determing the reasonableness, consistency, completeness, suitability, and lack of defects in a set of requirements.