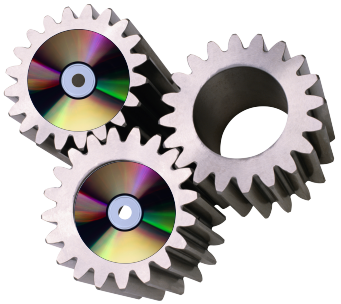
ENSF 613

Software Requirement Analysis and Process Management

What is Software Engineering?



- **Software engineering is the application of engineering to the development of software in a systematic method**.
- Needs hard and soft skills:
  - Hard skills includes: programming, analytical techniques, design techniques
  - Soft skills are the personal attributes and communication abilities needed for success on the job. In other words, skills that characterize the ability of a person to interact with others.

What is engineering?

"Engineering is the development of **cost-effective** **solution**s to **practical problems**, through the **application of scientific knowledge**"

"…Cost-effective solutions…"

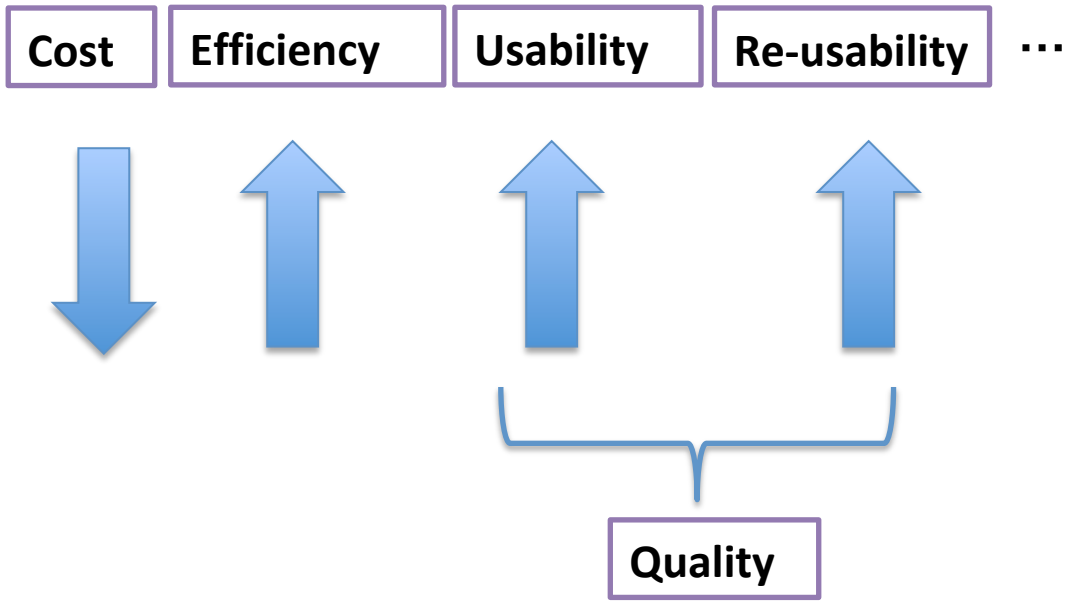
- Minimize cost
- Emphasis on building

"… Practical problems …"

- problems should matter to people

"…Application of scientific knowledge"

- Apply systematic techniques

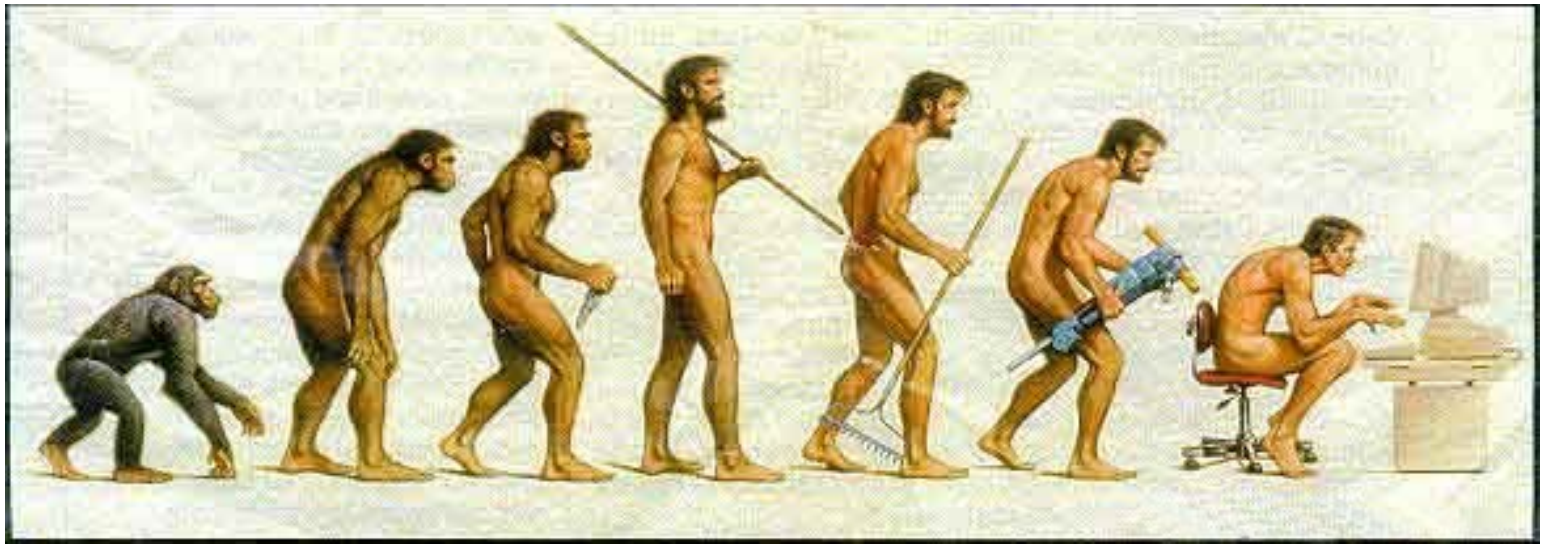


What is and Where is Software?

What is Software?

- Software is arguably the most interesting engineering product ever invented (Dick Hamlet, et al.).
  - "It is freed from almost all the limitations of physical world". Those things that you cannot do in the physical world, you can in the software.
  - It does not cost almost anything to reproduce once it has been designed and implemented.
  - It does not wear out.
  - And, it can in principle solve difficult problems perfectly.

Is it just programming?

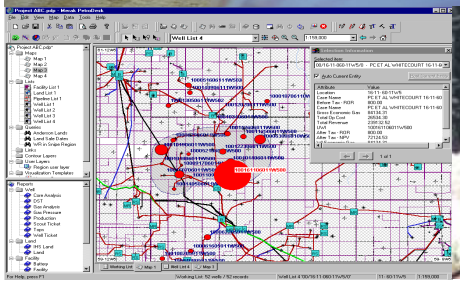


- Software is not just a computer program.
- It is a product that like any other engineering products starts with an idea that needs feasibility studies, analytical studies, design, constructions, testing and maintenance.
- Most of the industrial-strength software products have a long lasting lifecycle. Once they are born, they continue to grow.
- 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. Today, software is integrated into every industry.

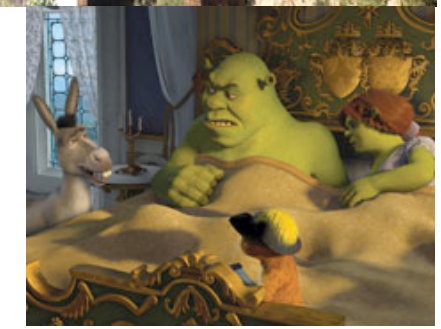
Where is Software?

Software is almost integrated into every industries:

- Manufacturing
- Oil and Gas
- Transportation
- Biomedical
- Robotics
- Management & Decision Making
- Games & Animations
- Music Industry



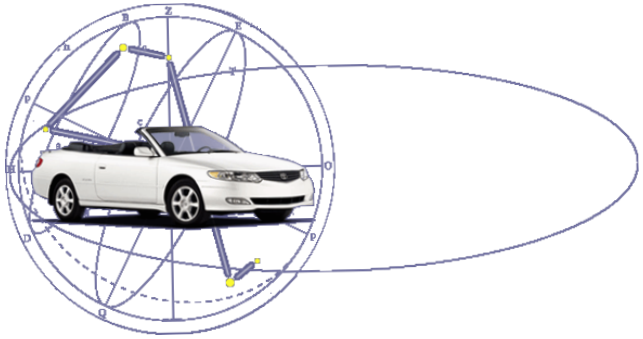
- Our day to day life is strongly depending on quality of software:



Where is Software?

1) Deeply embedded in devices:
   - The value of many state-of-the-art products is determined by the software. Examples:
     - Cars fuel and brake systems
     - Airplanes, fighter jets, …
     - Surgical tool
     - Appliances
     - Cell phones
     - etc.

Several hundreds of parts and systems in a car are controlled by software



How computers took over our cars

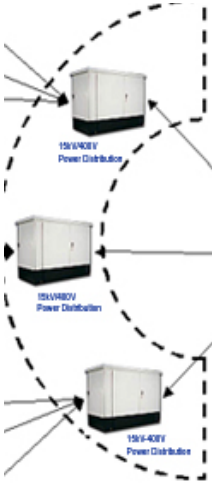
From BBC Website: http://news.bbc.co.uk/2/hi/uk_news/magazine/8510228.stm

Where is Software?

2) Software is also used to run and control many critical Industrial systems:

- Nuclear Power
- Power Plants
- Petroleum and Oil and Gas Refineries
- Defense Systems.

Source: GDC

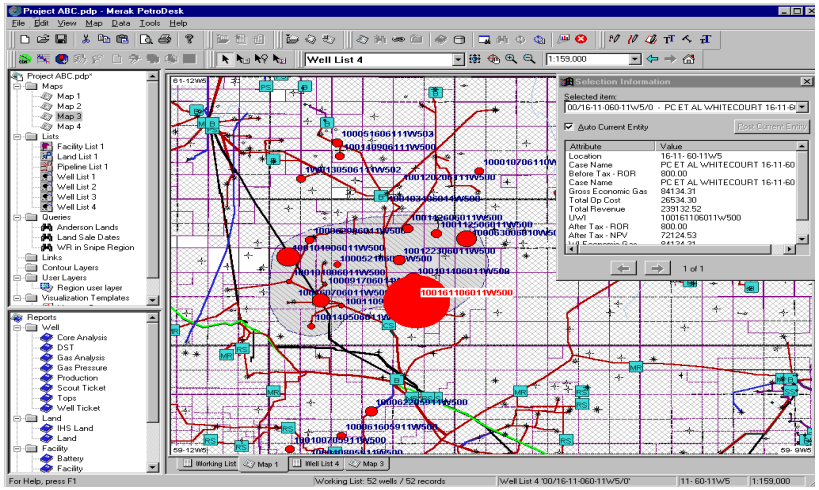


- Quality of products and services is determined dramatically by quality of software.

Where is Software?

3) Software is also used as Decision Support Tool by project managers, and engineers :

- Simulation software for revenue and cost analysis
- Transportation optimization systems
- Astronomical simulations programs
- Etc.



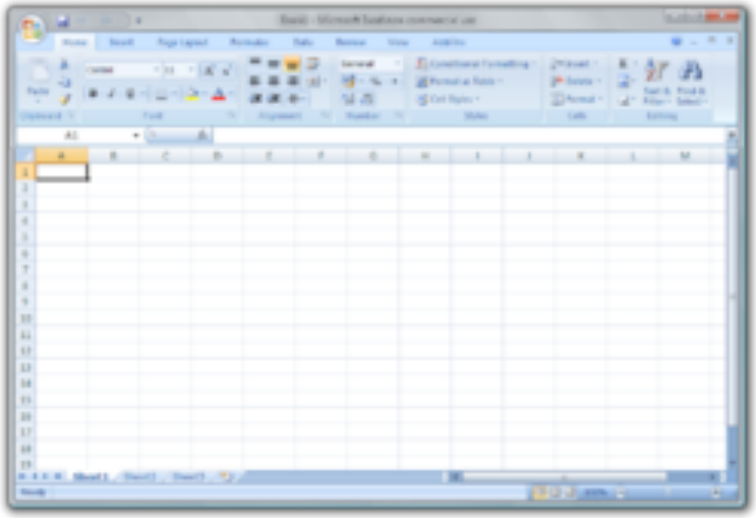
- This type of software allows for a low cost analysis prior to implementation of costly projects.

Where is Software?

4) Another broadly used area for software applications is Transaction Processing Systems, also known as Transaction Monitoring Systems:
   - Banks
   - Stock Markets
   - Ticket Reservation Systems
   - Ecommerce: Online shopping
   - Marketing systems
   - Etc.

Where is Software?

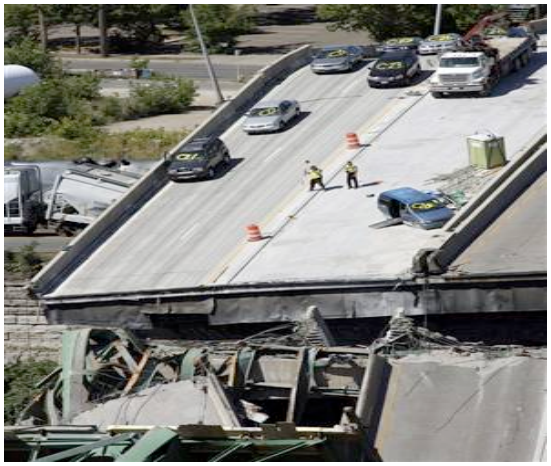
5) Last but not least is the important and common day-to-day use of software applications such as:
   - Word processors: Keynote, MS word, etc.
   - Spreadsheets: Excel, Lotus, Numbers, etc..
   - Email and chat applications.



- Did you know that most of these applications are written in C++?

Software Quality

- Again like any other engineered products, quality is always one of the biggest concerns.
- It is quite tangible when we talk about quality of an engineered product such as a bridge or tunnel. We can easily see the potential danger of passing over a bridge or into a tunnel that is poorly designed or constructed.



- What about potential dangers of poorly designed software?

Software Horror Stories

- On June 3, 1980, the North American Aerospace Defense Command (NORAD) reported that the U.S. was under missile attack. The report was traced to computer and software error.
- [http://www.cs.tau.ac.il/~nachumd/horror.html]

The 1988 shooting down of the Airbus 320 by the USS Vincennes was attributed to the cryptic and misleading output displayed by the tracking software.
[http://www.cs.tau.ac.il/~nachumd/horror.html]

More on Software Horror Stories

- Errors in medical software have caused deaths. Details in B.W. Boehm, "Software and its Impact: A Quantitative Assessment," Datamation, 19(5), 48-59(1973).
- Millions of bank accounts were impacted by errors due to installation of inadequately tested software code in the transaction processing system of a major North American bank, according to mid-2004 news reports.
- Computer blunders were blamed for $650M student loan losses.
- For more stories see:
  - http://www.cse.lehigh.edu/~gtan/bug/softwarebug.htm
  - http://www.sereferences.com/software-failure-list.php

What is Software Development Process?

- There are several different approaches to software development:
  - some use 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 phase
  - Maintenance phase

Closer Look at the Software Development Process

- The process of engineering a software product is not much different from engineering of other products.
- In the following slides we discuss and compare the process of software development with a civil engineering development process.
- As an example, lets consider the following two scenarios:
  1. An engineer who has been asked to develop an engineering simulation software
  2. An engineer who has been asked to design a bridge

Inception Phase

During this phase *the primary objective is to scope the system adequately as a basis for validating initial costing and budgets*

- Target Product: Software
  - Construction of a software may not be feasible, if buying and using an existing software is technically and economically a better alternative

- Target Product: Bridge
  - Construction of a bridge may not be feasible, building tunnel is technically and economically a better alternative.

Software Development - Phase One: Inception

- During this phase you establish the business rationale and the following questions must be answered.
  - Are there other alternatives?
  - Is this product economically feasible?
  - Is this product technically feasible?
- If the project does not pass this phase it can either be cancelled or it can be repeated after being redesigned to better meet the criteria.

Requirements Analysis Phase

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

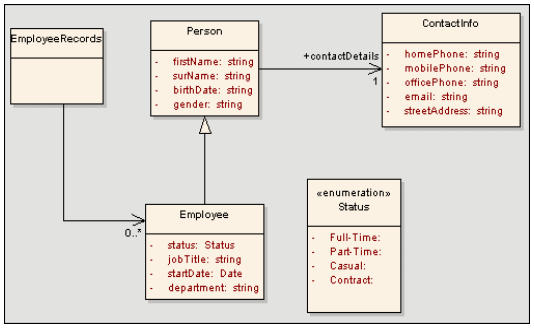
In this phase you should answer the following type of questions:

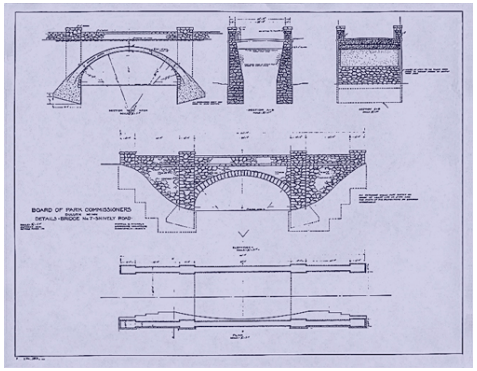
- **Target Product: Software**
  - Who are the users of this product
  - What are the inputs to the system
  - What are the system's output.
  - Etc.

- **Target Product: Bridge**
  - What type vehicle is going to pass over the bridge
  - What is the peak flood level in the river
  - What is the pick time of traffic over the bridge.
  - Etc.

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 possible solution.





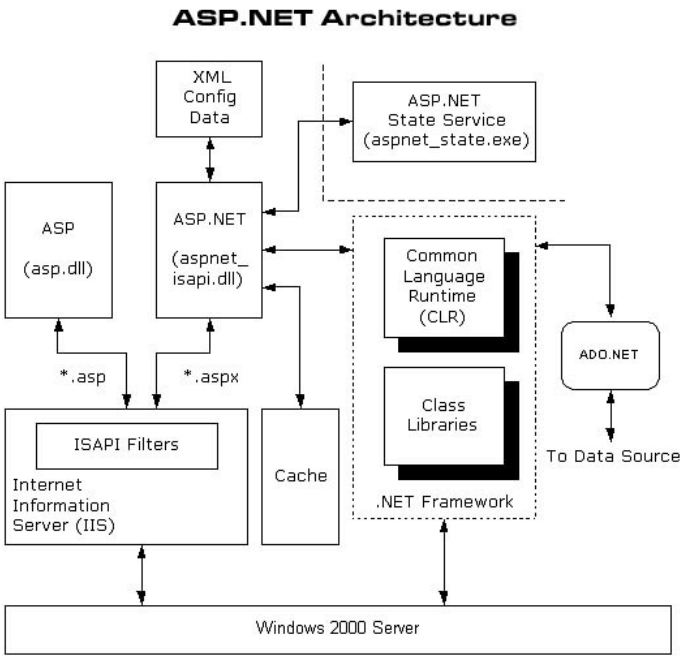
The outcomes of this phase include:

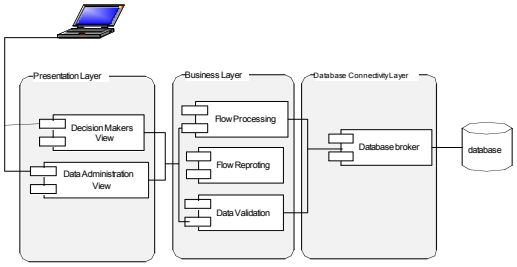
- **Target Product: Software**
  - System's prototype
  - System architecture
  - System's blueprints
  - Etc.

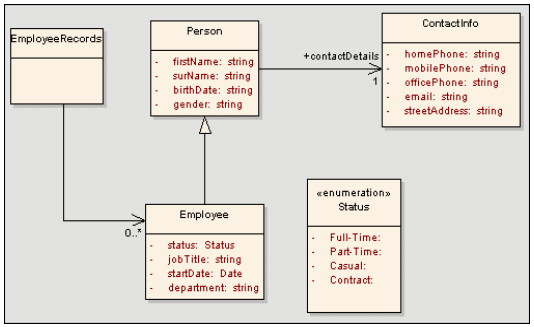
- **Target Product: Bridge**
  - physical model (prototype)
  - Architecture
  - Blueprints
  - Etc

Design Phase

- Software design engineers like other engineering disciplines (e.g. civil, mechanical, electrical), use tools and special notations to express their design decisions. Unified Modeling Language, also known as UML is one of the commonly accepted notation among software developers.

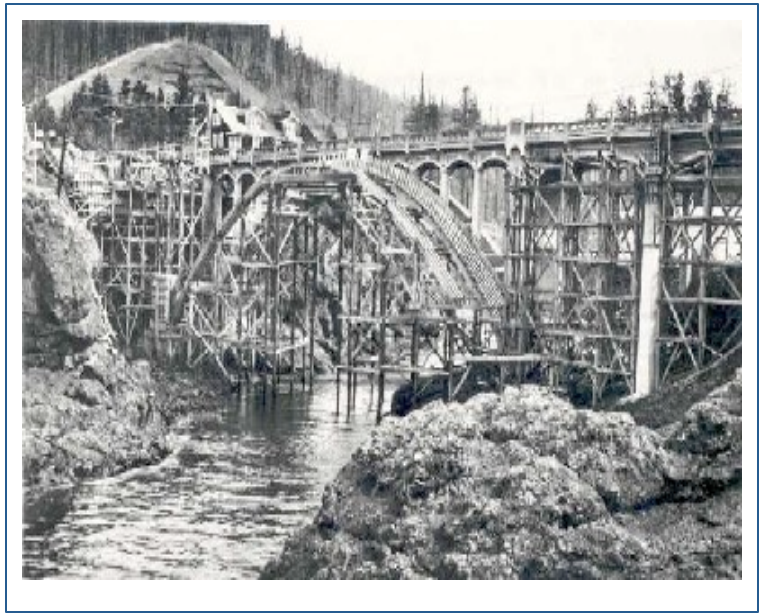






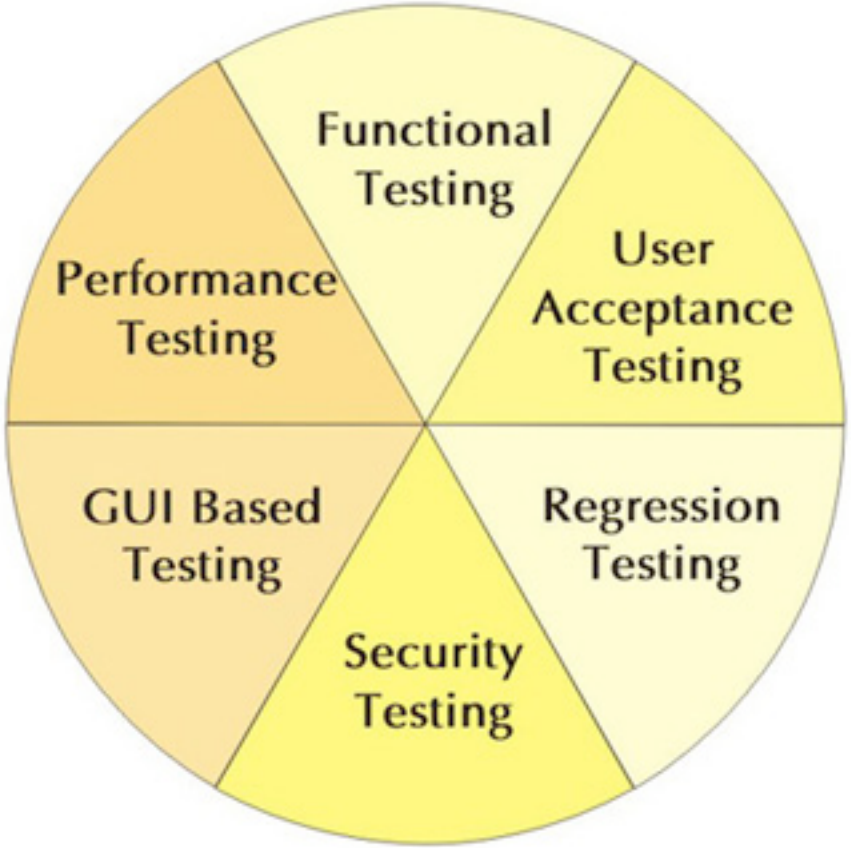
Construction - Phase

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 - Phase

- Like any other engineering processes, the building blocks of the software must be tested during the entire build process, at different levels, (system level, subsystem level, component level, and object or function levels).
  - The importance of software testing becomes even more vital when software controls important systems such nuclear plants, security and defense systems.



Source: http://www.visionetsystems.com/images/software_quality_testing_qc.jpg

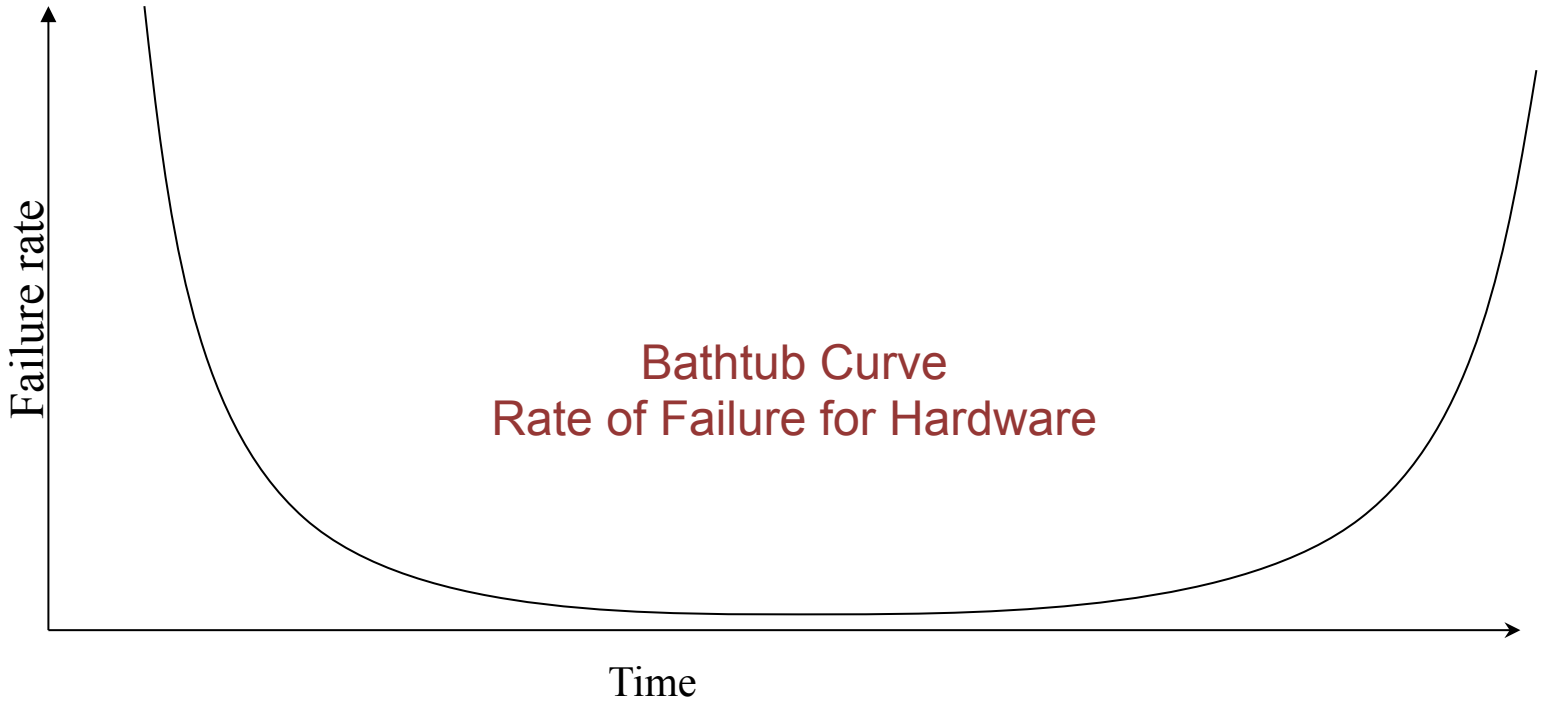
Maintenance Phase

- Once software is created, constructed, and deployed, is always subject to modification and changes.
- In other words, 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.
- There are several categories of software maintenance:
  - Corrective maintenance: to fix the errors 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 maintainability.
  - Preventive maintenance: to detect and correct latent faults in the software product before they become effective faults.

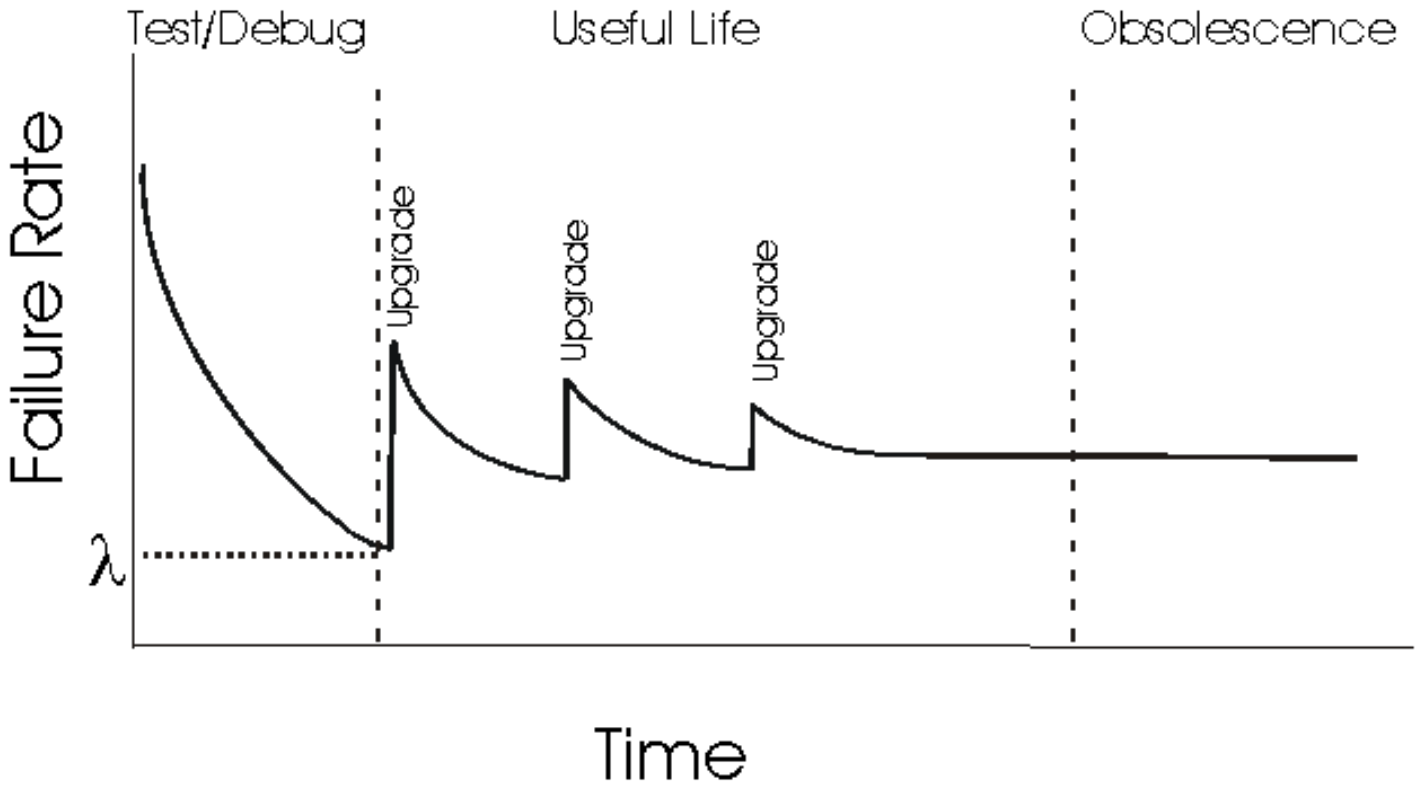
Software vs. Hardware

Failure Curve for Hardware

- Bathtub cure showing the rate of failure for hardware.
- Notice that this rate represents failure rate for a population of hardware.



Failure Curve for Software Failure



https://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/

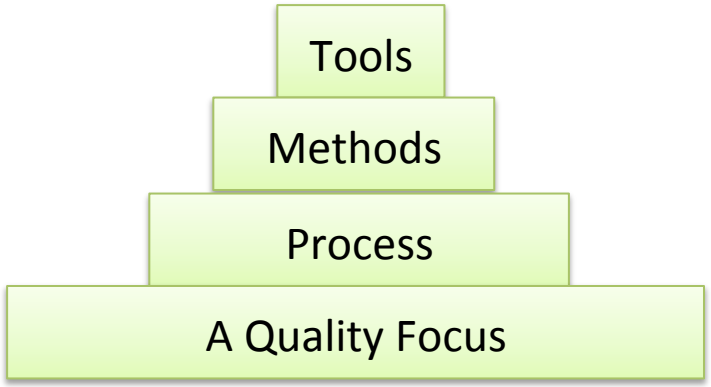
Software Classification

Software Systems Classifications

- Many classifications of software systems are possible:
  - 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 and destinations tend to be people and devices.
    - Often used for device control, or in science and engineering 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 and automations

Software Engineering as a Layered Technology

- Started in early 60s, as profession concerned about how to create software with the objective of maximizing its quality.
- Quality may refer to: maintainability, stability, performance, reusability, usability, testability, readability, security, reliability, size, cost, and many more.
- It's a layered Technology



Software Project Management

- Software project management involves planning, monitoring, and control of the people, process, and events that occur as software evolves from a preliminary concept to an operational implementation.
- Effective software project management focuses on 4 P's: People, product, process, and project.

People

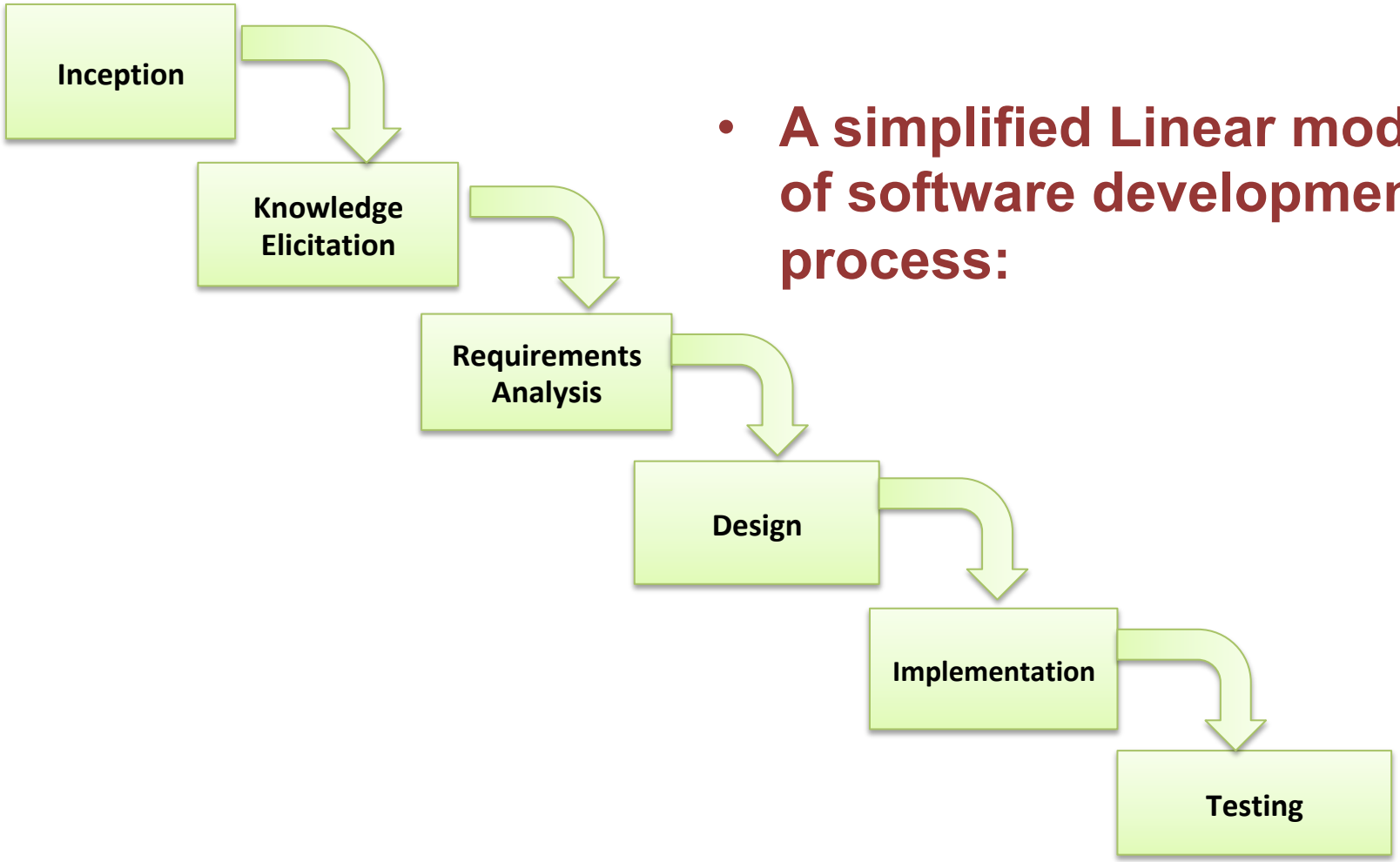
- Players:
  - Customers/Clients/Users
  - Management
  - Practitioners (software engineers)
    - Systems Analysts
    - Requirements Engineer
    - Systems Designers/Architects
    - Programmers
    - Test Engineers
    - Quality Assurance
  - Operation Staff
  - External Standards Personnel

Skills Required for a Software Engineer

- To succeed as a Software Engineer you need four sets of skill:
  - Analytical skills, including systems thinking
    - You should be able to define the system's boundary's, to analyze the system.
    - You should be able to create an abstract description of a system.
    - You should be able to understand the required interfaces between systems and other actors.
    - To focus on the essential characteristics of the system, without being distracted by implementation details.
  - Technical skills
  - Management skills
  - Interpersonal skills
  - Communication skills

Process

Brief Introduction to Software Development Process



- **A simplified Linear model of software development process:**

Software Crisis

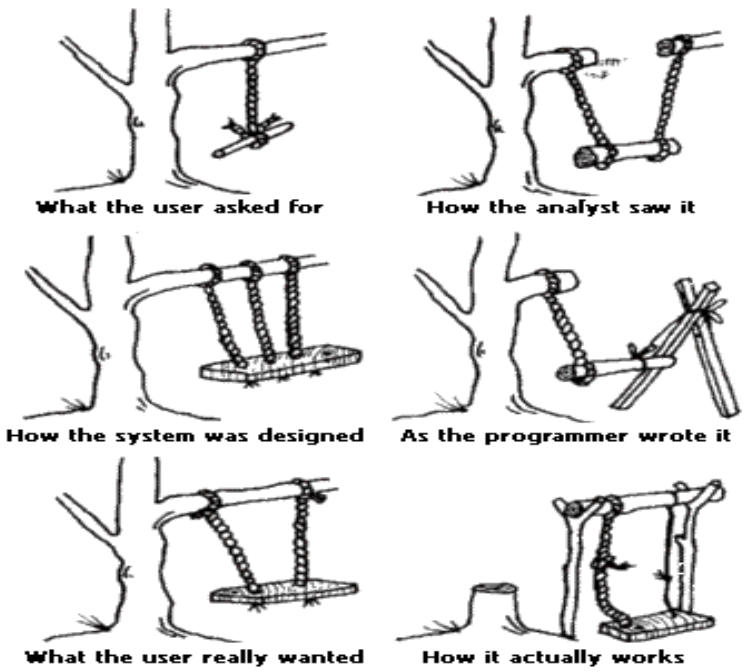
- Software crisis:
  - Schedule and cost estimates are often grossly inaccurate
  - The productivity of software developers has not kept pace with the demand for their services
  - The quality of software is sometimes “less than adequate”.
- Source of problem?
  - Projects are often undertaken with only a vague indication of requirements:

Myths

- All you need to start is a general statement of objectives.
  - Poor initial definition is the major cause of failed software efforts
- It doesn't matter that our requirements always change. Software is very flexible.
  - Very true, but the impact of change varies with the time it is introduced
- Why should we change the way we develop software? We're doing the same kinds of things as ten years ago.
  - Even if applications are the same, demand and *importance* of software have increased greatly
- If we fall behind schedule, we can add more programmers and catch up (the "Mongolian horde concept")
  - "Adding people to a late software project makes it later" --- Brooks
- Once the program is written and it works, we're done
  - Of the three phases of software development (definition, development, maintenance), 50% to 70% of effort is expended after it is delivered to the client for the first time

What is Requirement?

- The Definition of **what** the product should do. Includes graphical models and textual description of
- Brooks, 1995: The hardest part of building a software system is deciding precisely **what** to build.



- Also known as "Requirements Engineering":
  - concerns how to decide what system we are going to build…
    - Understand, analyze, and build the systems requirements models, and focus on 4-Ws:
    - What problem to be solve
    - Why such a problem should be solved
    - Who should be involved
    - And How-good should be built

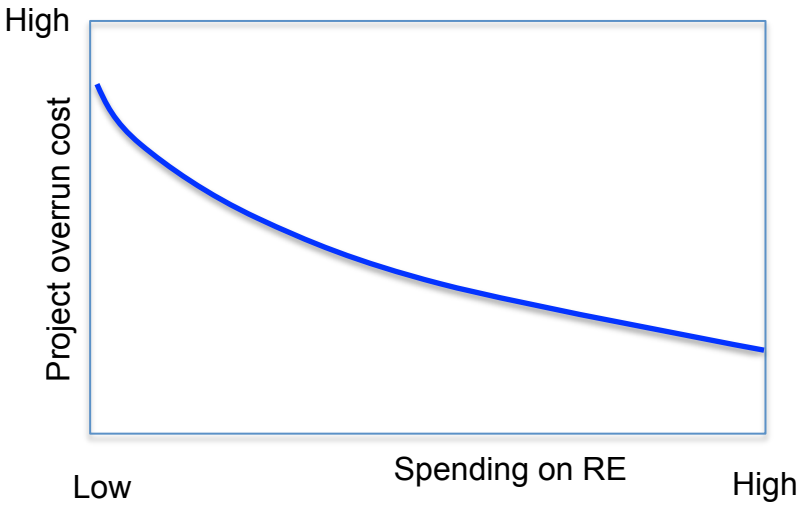
General Challenges

- Understanding of each domain.
- Variety of stakeholder in each domain and understanding their needs.
- Requirements conflicts between and within each domain
- Knowledge transfer, and updating team members and mangers.
- Legal and political constraints
- Continuous change of requirement
- High expectations from stakeholders and our customer
- Technical challenges

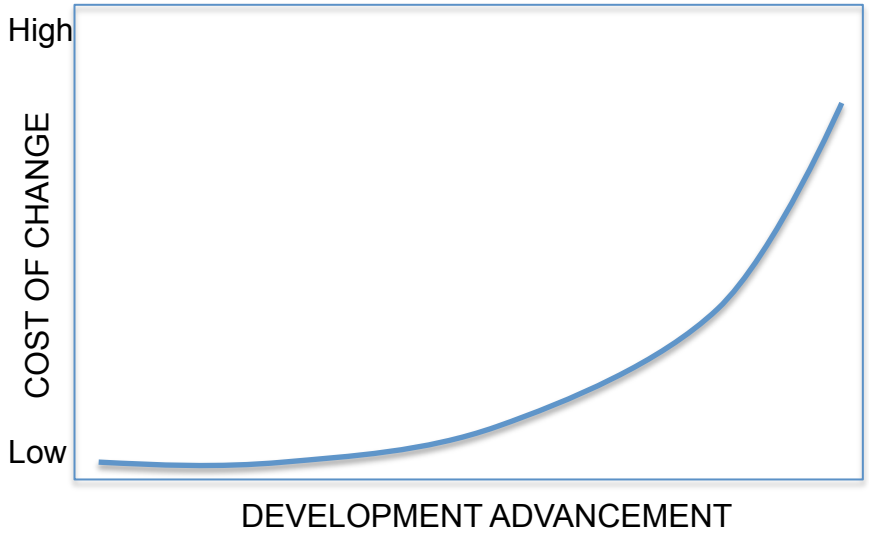
RE Specific Challenges

Why Requirements 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



[Forsberg 1996]



The RE Process

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

RE Process

- Requirements development encompasses all of the activities involved in eliciting, analyzing, specifying, and validating the requirements.