CSE Department of Computer Science & Engineering

Introduction to Software Engineering

CSC445/CSE451 Software Engineering
CSC/SEN545: Software Engineering Concepts

Lecture 01

Objectives

- To introduce software engineering and to explain its importance
- To set out the answers to key questions about software engineering
- To introduce ethical and professional issues and to explain why they are of concern to software engineers

Topics covered

- Over view of software engineering and Software
- Objective and Basic steps of Software Engineering
- Basic Principles of Software Engineering and Management

What is software engineering?

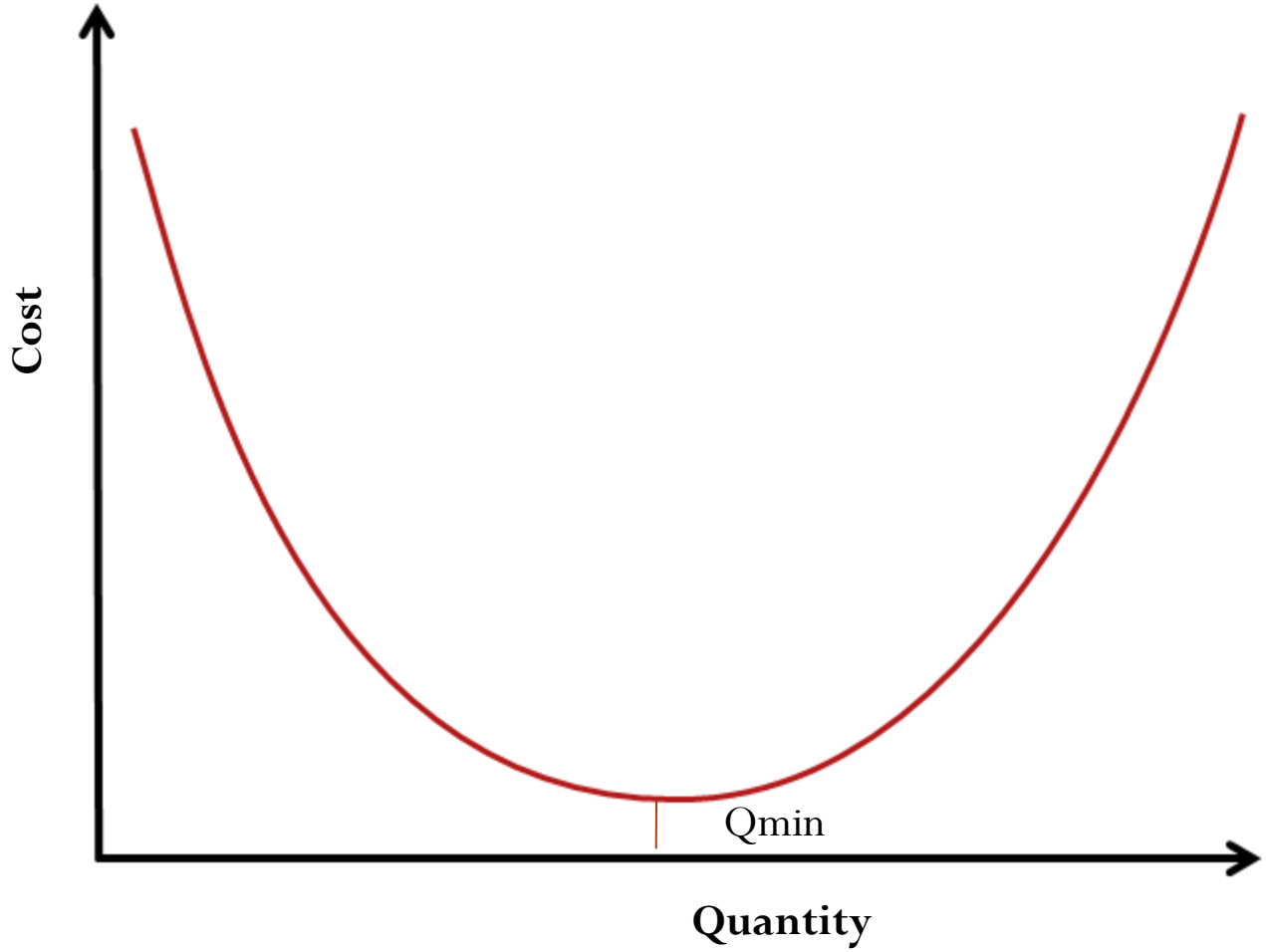
- Software engineering is an engineering discipline which is concerned with all aspects of **software production**

- Software engineers should adopt a **systematic** and **organised** approach to their work and use **appropriate tools and techniques** depending on the **problem** to be solved, the **development constraints** and the **resources** available

Why Software Engineering?

- Generate Wealth through software.

Production Curve



Difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software

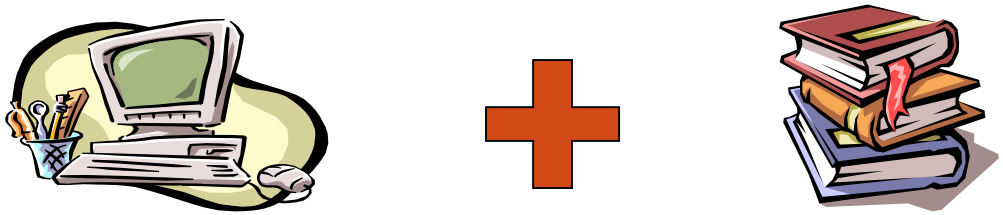
- Computer science theories are currently insufficient to act as a complete underpinning for software engineering

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this process

- System engineers are involved in system specification, architectural design, integration and deployment

What is software?

- **Computer programs** and **associated documentation**



- **Software products** may be developed for a particular customer or may be developed for a general market
- **Software products** may be
  - **Generic** - developed to be sold to a range of different customers
  - **Bespoke** (custom) - developed for a single customer according to their specification

Software?

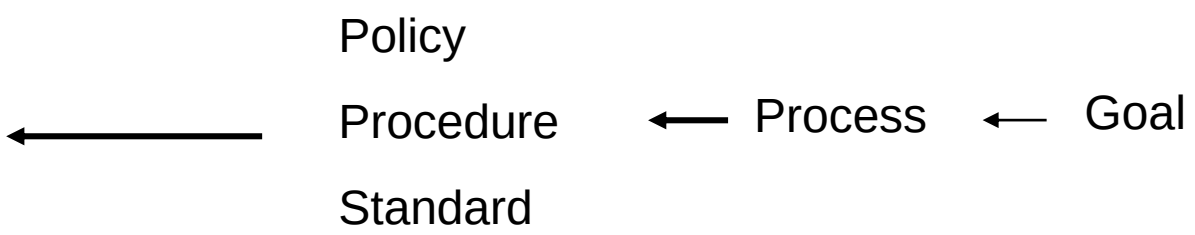
- A textbook description of software might take the following form:

- *Software is (1) instructions (computer programs) that when executed provide desired function and performance, (2) data structures that enable the programs to adequately manipulate information, and (3) documents that describe the operation and use of the programs.*

What is software?

- Software is a tool to improve work processes and/or engineering artifacts.

- Basic structure of work processes
  - Man/Woman
  - Machine/Tool
    - Software
    - Database
    - Hardware
      - Computing
      - Non-Computing
    - Communication network



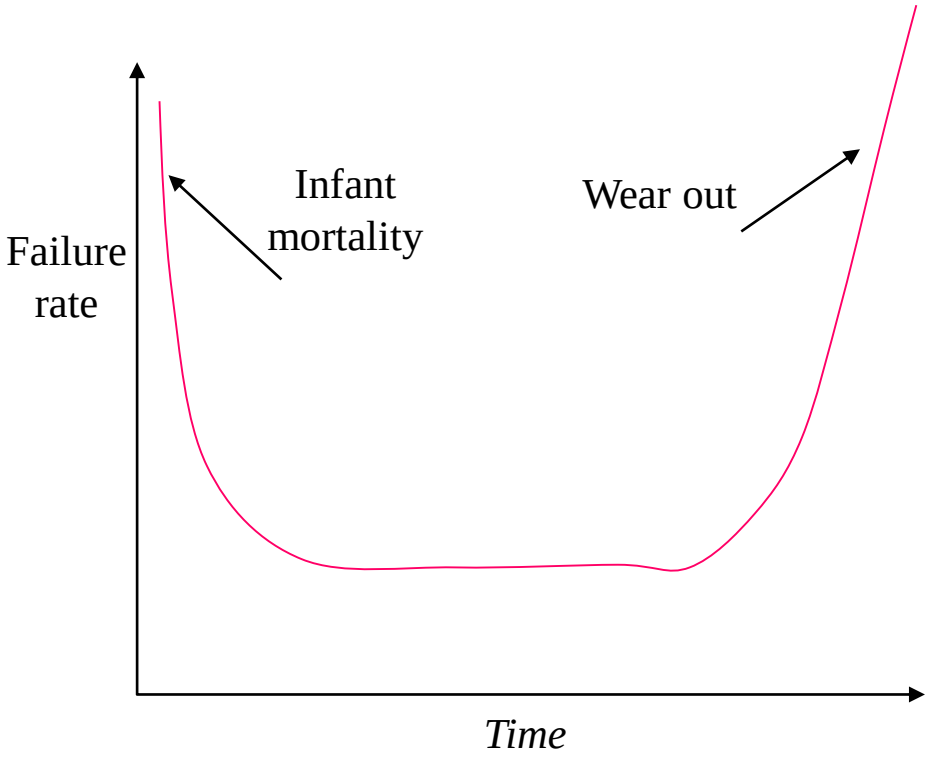
Software characteristics:

- Software is developed or engineered, it is not manufactured in the classical sense.
- Software does not "wear out."
- Most software is custom built, rather than being assembled from existing components.

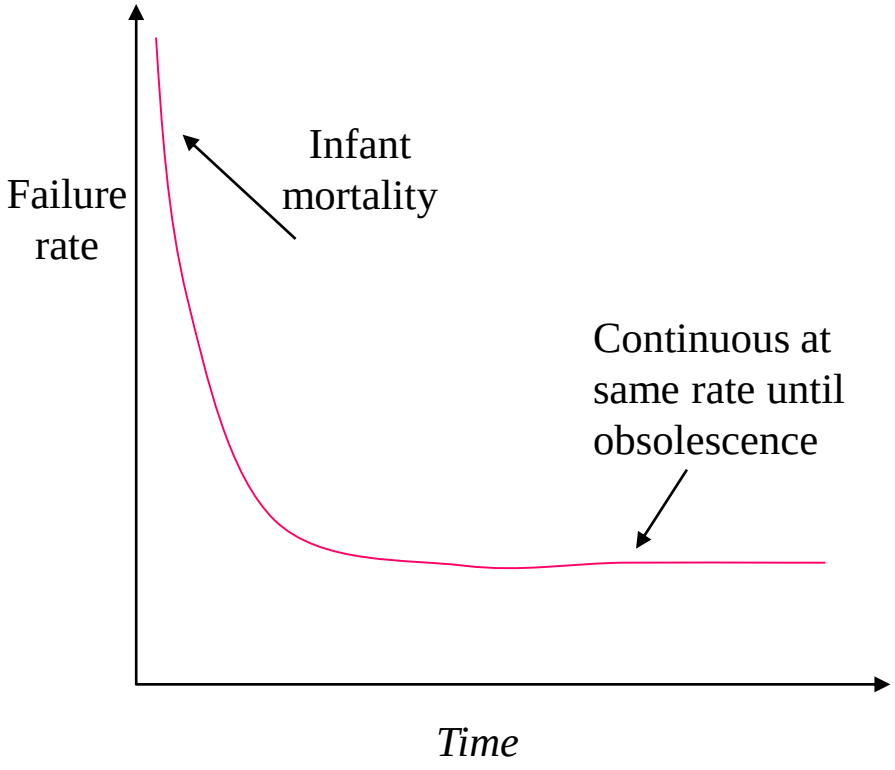
Software applications:

- System Software
- Real-time Software
- Business Software
- Engineering and Scientific Software
- Embedded Software
- Personal Computer Software
- Artificial Intelligence Software

Failure characteristics of software products:

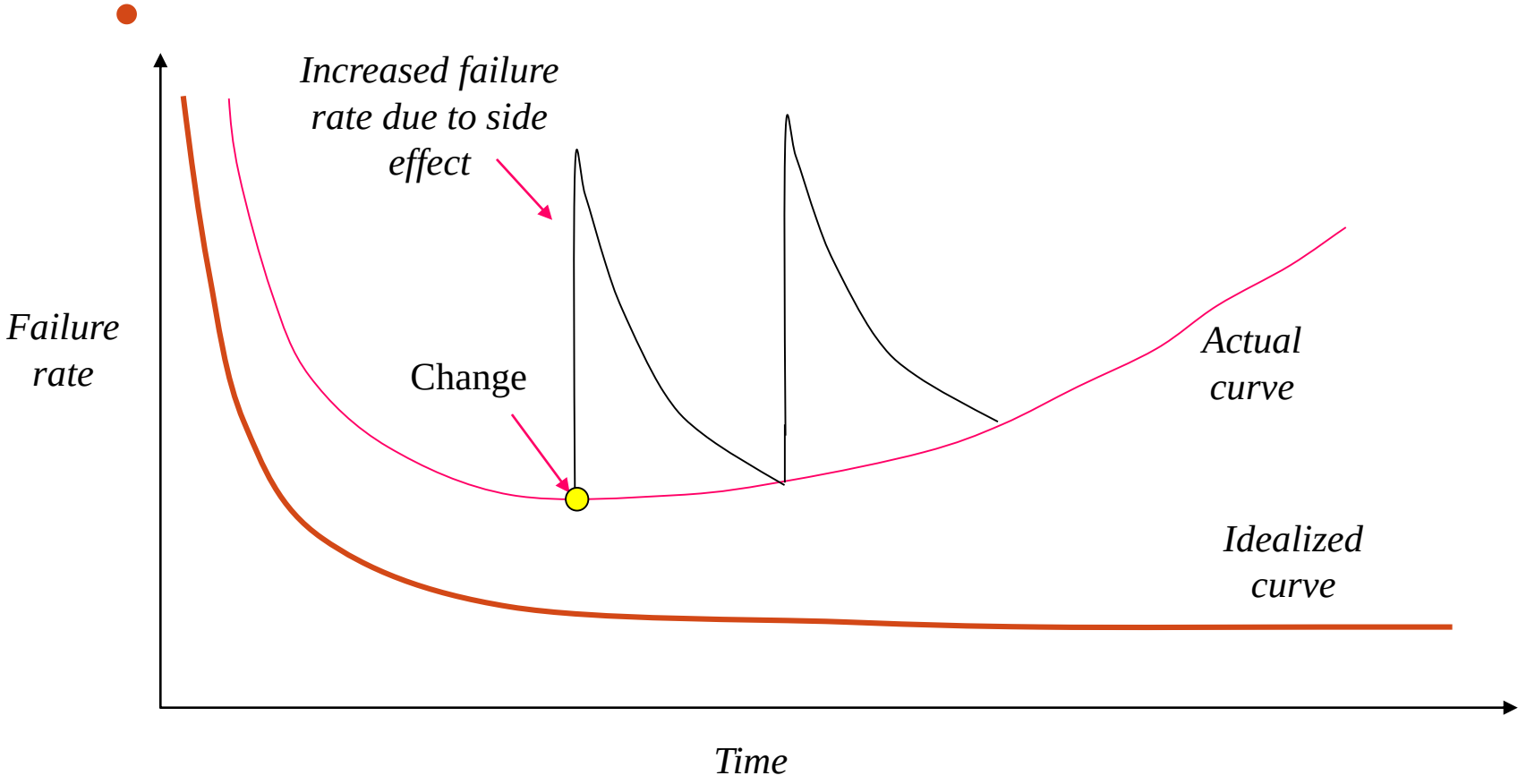


Failure curve for hardware

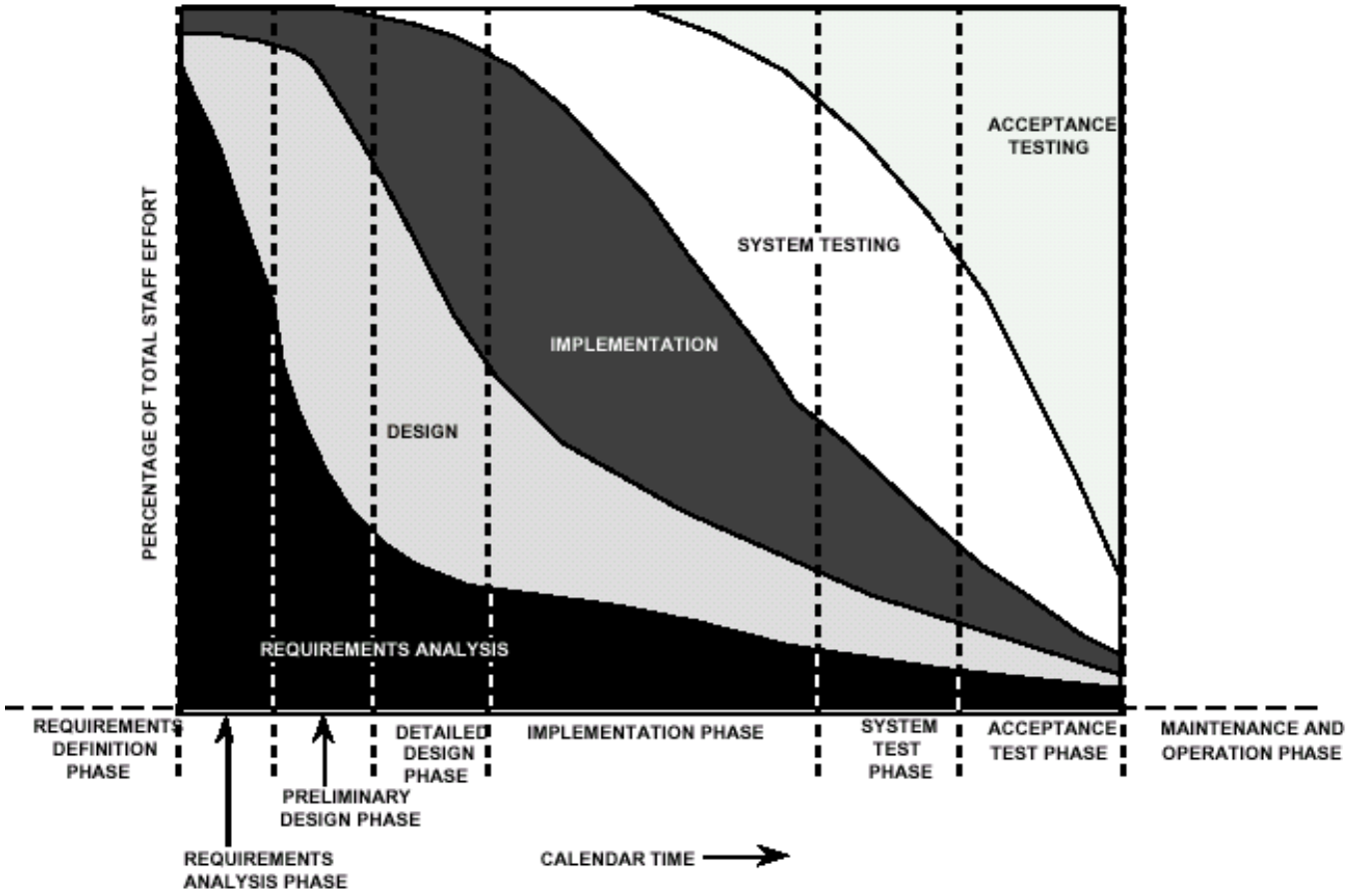


Failure curve for software

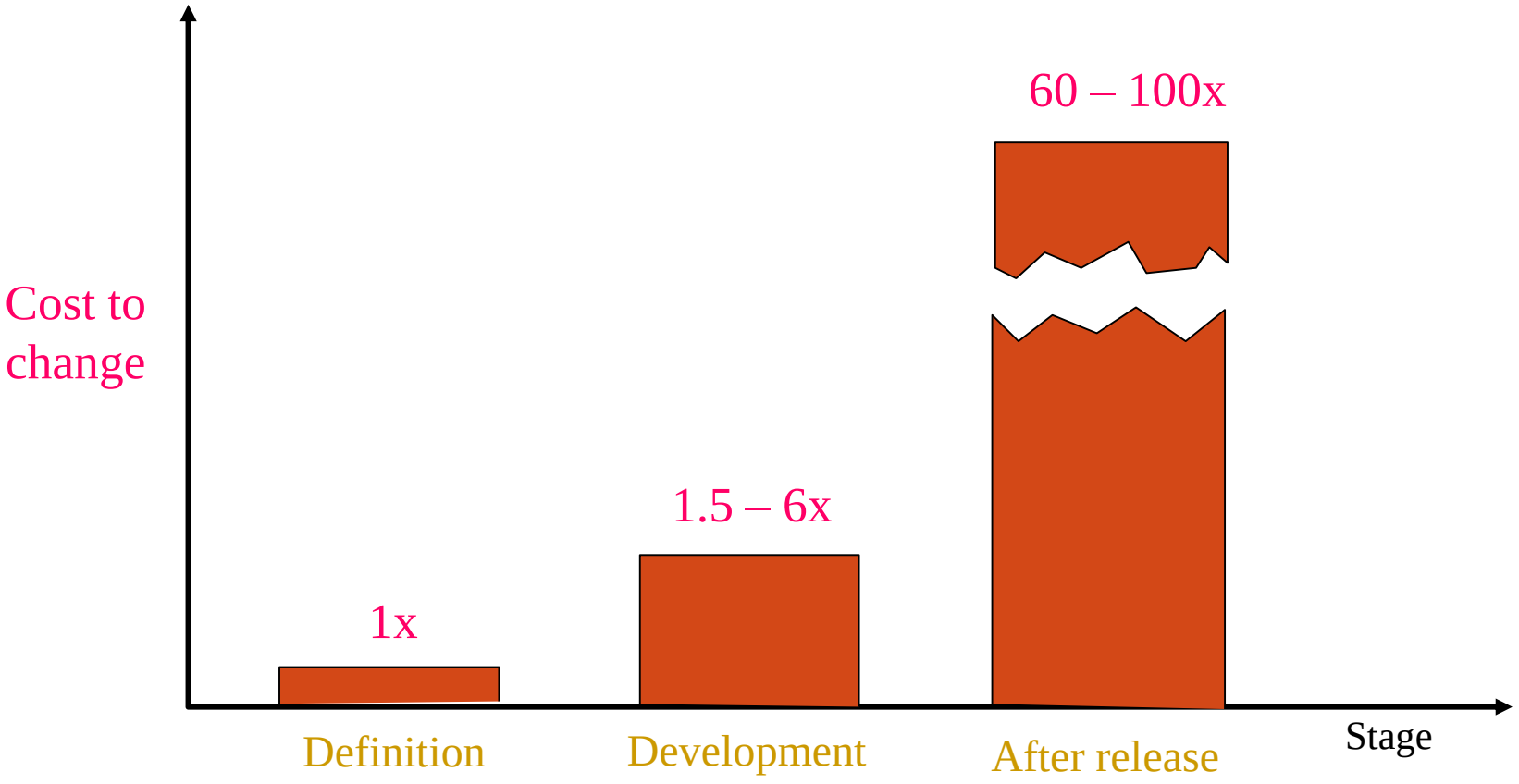
Failure characteristics of software products



Software Development Process...



The impact of change of software:



Why should people pay for software?

- It's about the improvement what it makes to work processes/engineering artifacts.
- How much should they pay?

Objective of software engineering:

- To capture right requirements with the optimum usage of other components of the system.

- To implement requirement by ensuring predictability and continuous improvement of
  - Quality
  - Time
  - Cost

- Quality
  - Conformance to requirement

Basic steps of software engineering:

- Technical steps:
  - Work process/engineering artifacts understanding.
  - Determine the role of software with the optimum roles played by other components of the system.
  - Design of software requirements
  - Coding and testing
  - Integration, system testing
  - Implementation/deployment/commissioning

Basic steps of software engineering..

- Engineering management steps
  - Project management
  - Quality assurance
  - Requirements/change management
  - Configuration management

Every business is software business

- Why?
- Dilemma in software business:
  - If you extend schedules to make realistic customer commitment, you loose business.
  - If you make competitive commitments, your software is late and customers are unhappy.
  - If you do this too often, you will be known as an unreliable supplier.
  - If this condition continues for long, you will lose so many customers that you could well go out of business.

Principles of Software Management

- Principle number one: Recognize that you are in software business
  - Customers
  - Product managers
  - Developers
- Principle number Two: The quality must be the top priority
- Principle number Three: Quality software is developed by disciplined and motivated people

Why project fails

- Unrealistic schedules
- Inappropriate staffing
- Changing requirements
- Poor quality work
- Believing in magic

Key lessons to deal with project failure

- The two key components of software management are a dedication to quality and consistently disciplined engineering work
- When software projects fails, it is usually because a manager did not insist that the work be done in the right way
- All of those causes could have been avoided had management insisted on planned and disciplined work.

Rational management

- Four elements to rational management:
  - Set aggressive long-term goals, but break these goals into realistic and measurable short-term goals.
  - Require detailed and complete plans, review these plans, and then negotiate the commitments with the people who will do the work.
  - Insist on facts and data, and use these facts and data to run these businesses
  - Monitor the work, and use current data to anticipate and resolve future problems.

Why should quality be managed?

- There are three reasons to insist that software quality be measured and managed:
  - Poor-quality software can cause major property damage and even kill people
  - Quality works saves time and money
  - If management do not manage software quality, nobody else will do.

Why quality pays

- Defective software is expensive and can even be life threatening.
- Software engineers inject defects because they are human.
- The cost to find and fix a defect increases exponentially as development progresses.
- The engineer who developed a program is best able to find and fix defects.
- Effective defect-removal methods require special training.
- If you do not manage software quality, nobody else will.

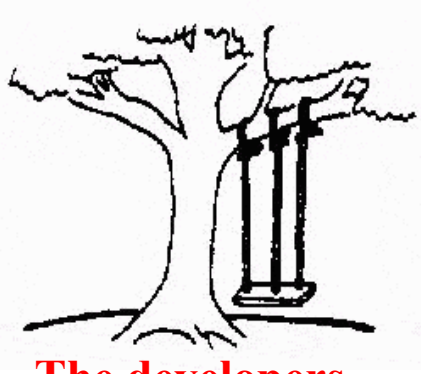
Leadership goals:

- In setting goals, make them clear and measurable, assign them to specific people, and track performance against them.
- Goals that are measured get managed, and goals that are managed are often met.
- Goals that are not measured and managed won't be met, and performance may actually deteriorate.
- To improve engineering performance, you must change what people do.
- To change their behavior, the engineers must know how to make detailed plans, manage quality, and use efficient working methods.
- To actually improve the organization, the engineers must consistently use the methods they know and the managers must guide, support, and coach them in doing so.

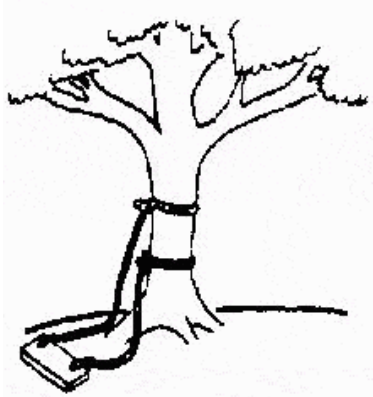
Changing engineering behavior:

- For engineers to use disciplined methods, their managers must agree with the methods and support their use.
- To actually change engineering and management behavior, the managers must be responsible for implementing the changes.
- If software engineers are to change their behavior, they must believe that the new methods will work for them.

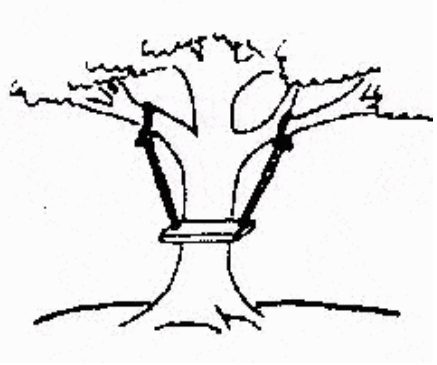
The requirements specification was defined like this



The developers understood it in that way

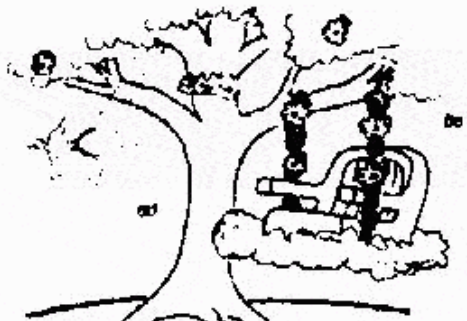


This is how the problem was solved before.

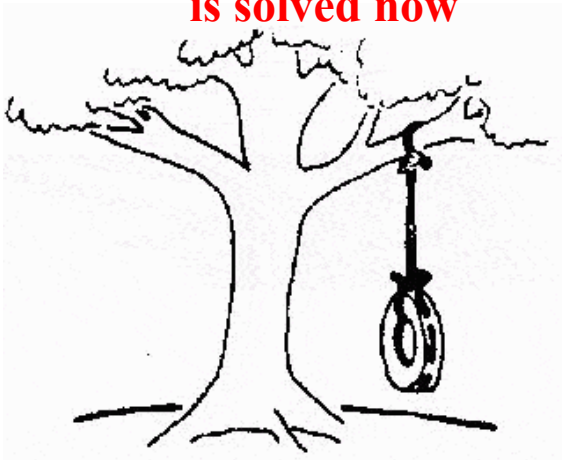


This is how the problem is solved now

That is the program after debugging



This is how the program is described by marketing department



This, in fact, is what the customer wanted … ;-)

Thank You