IS1204 Fundamentals of Software Engineering

H N D Thilini

hnd@ucsc.cmb.ac.lk

Activity

SEARCH FOR SOFTWARE DISASTERS

SELECT ONE AND PRESENT TO THE CLASS

Software Disasters

The Mariner 1 Spacecraft, 1962

- NASA launched a data-gathering unmanned space mission to fly past Venus. It did not go to plan.
- The Mariner 1 space probe barely made it out of Cape Canaveral before the rocket veered dangerously off course. Worried that the rocket was heading towards a crash-landing on earth, NASA engineers issued a self-destruct command, and the craft was obliterated about 290 seconds after launch.
- An investigation revealed the cause to be a very simple software error. A hyphen was omitted in a line of code, which meant that incorrect guidance signals were sent to the spacecraft. The overall cost of the omission was reported to be more than $18 million at the time (about $169 million in today's world).

Software Disasters

The Morris Worm, 1988

- One of the costliest software bugs ever was caused by a single student. A Cornell University student created a worm as part of an experiment, which ended up spreading like wildfire and crashing tens of thousands of computers due to a coding error.

- The computers were all connected through a very early version of the internet, making the Morris worm essentially the first infectious computer virus. Graduate student Robert Tappan Morris was eventually charged and convicted of criminal hacking and fined $10,000, although the cost of the mess he created was estimated to be as high as $10 million.

Software Disasters

NASA's Mars Climate Orbiter, 1998

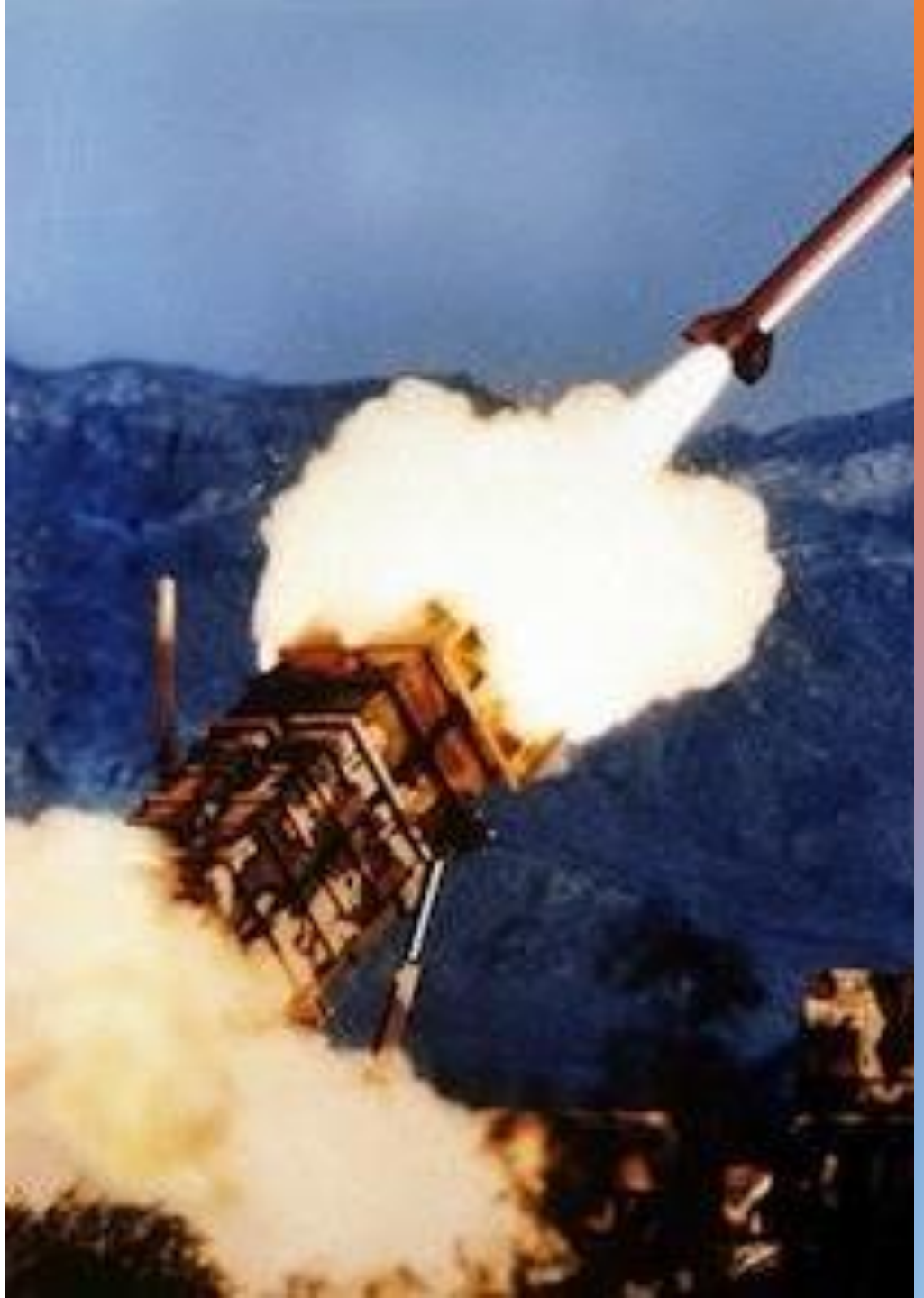
- NASA engineers found out back in 1998 when the Mars Climate Orbiter burned up after getting too close to the surface of Mars.
- It took engineers several months to work out what went wrong. It turned out to be an embarrassingly simple mistake in converting imperial units to metric.
- According to the investigation report, the ground control software produced by Lockheed Martin used imperial measurements, while the software onboard, produced by NASA, was programmed with SI metric units. The overall cost of the failed mission was more than $320 million.

Software Disasters

Patriot Missile System Timing Issue Leads To 28 Dead

- By far the most tragic computer software blunder on our list occurred on February 25, 1991, during the Gulf War. While the Patriot Missile System was largely successful throughout the conflict, it failed to track and intercept a Scud missile that would strike an American barracks.

- The software had a delay and was not tracking the missile launch in real time, thus giving Iraq's missile the opportunity to break through and explode before anything could be done to stop it, according to the US Government Accountability Office. In all, 28 were killed with an additional 100+ injured.



Software Disasters

Apple Maps Goes Nowhere Fast

- With the 2012 Apple iOS 6 update, the company decided to kick the superior Google Maps platform to the curb in favor of its own system. Unfortunately, it did a poor job of mapping out locations resulting in one of the most epic fails of the mobile computing movement.

- TPMIdeaLab noted the software was “missing entries for entire towns, incorrectly placed locations, incorrect locations given for simple queries, satellite imagery obscured by clouds and more” in a September 2012 report. David Pogue of the New York Times added that it was the most embarrassing, “least usable piece of software Apple has ever unleashed.”

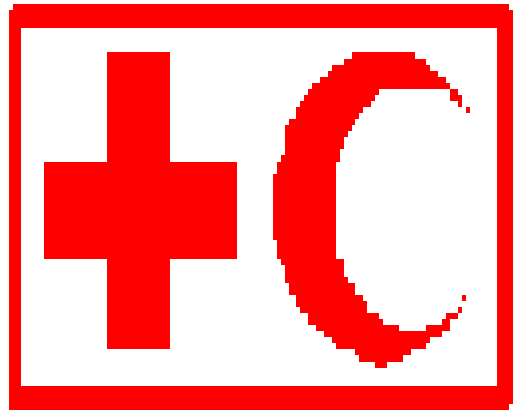
Software Disasters...cont.

Airplane disasters

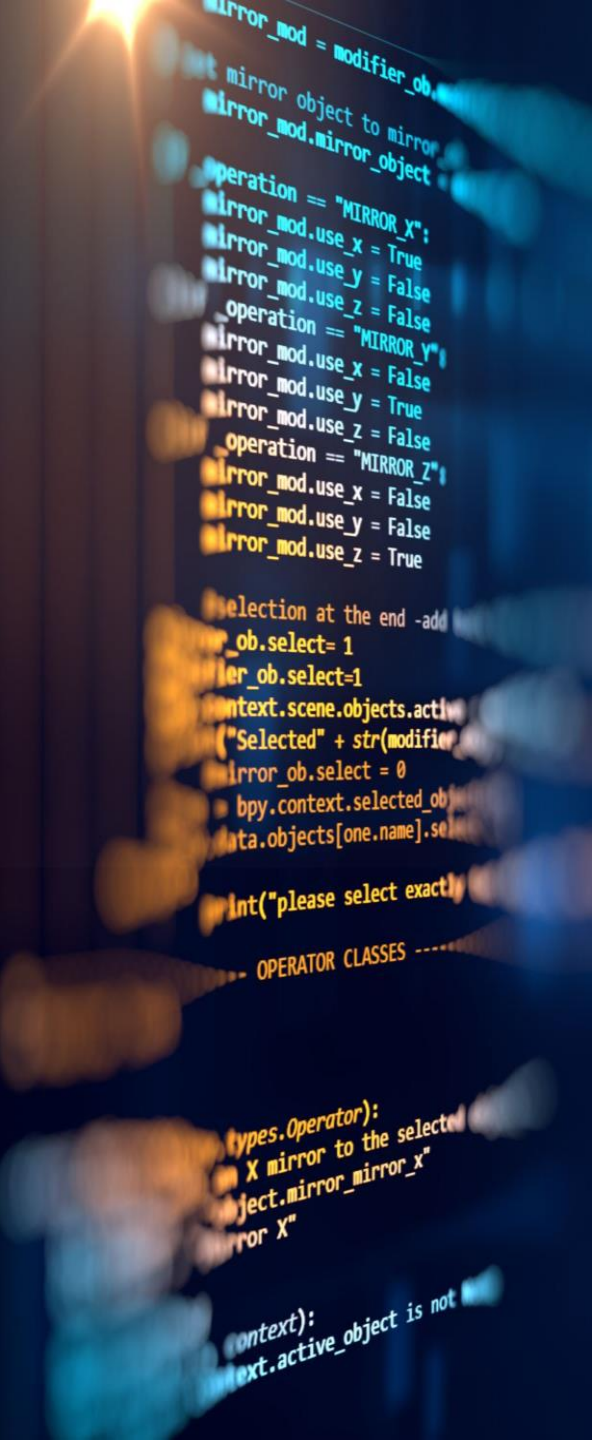
The crash of an Airbus plane, owned by China Airlines on the Nagoya Airport on April 16, 1994 due to the software problems.

Medical disasters

A medical disaster due to software-related accidents in safety-critical systems that used a computerized radiation therapy machine called the Therac-25. (contributed to the death of several cancer patients.)



……..FIND SOME MORE EXAMPLES



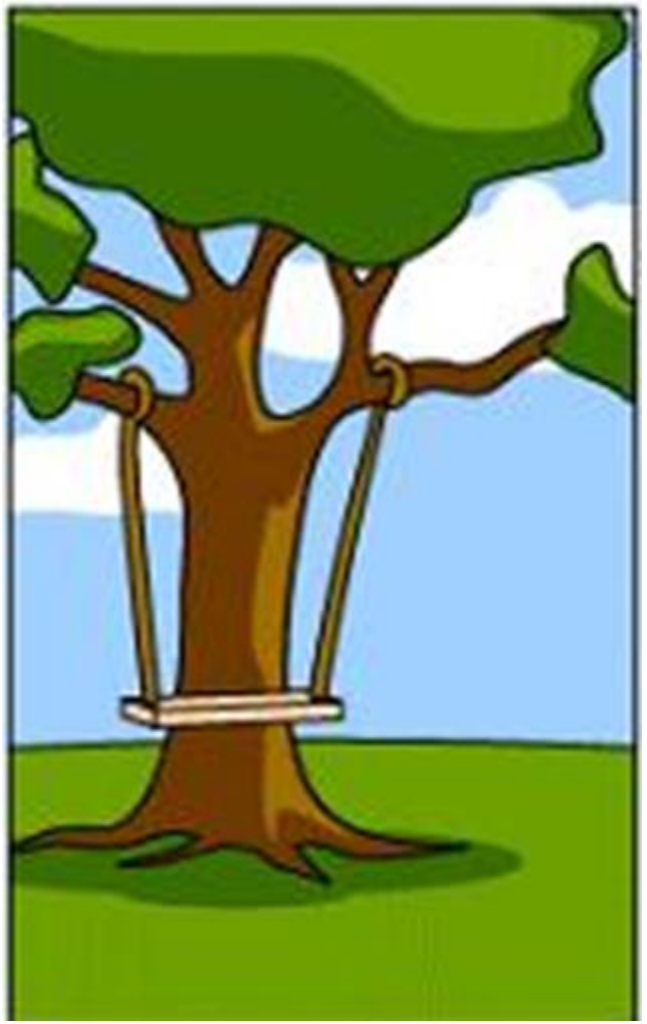
Software Development

- Software development is the computer programming, documenting, testing, and bug fixing resulting in a software product.
- **A software development methodology** (process, model, or life cycle) is a framework that is used to structure, plan, and control the process of developing software systems.
- Software development process is considered to be a life cycle of a software product.

Swing Project

How the Customer Described it

How the Business Analyst Understood it



Swing Project

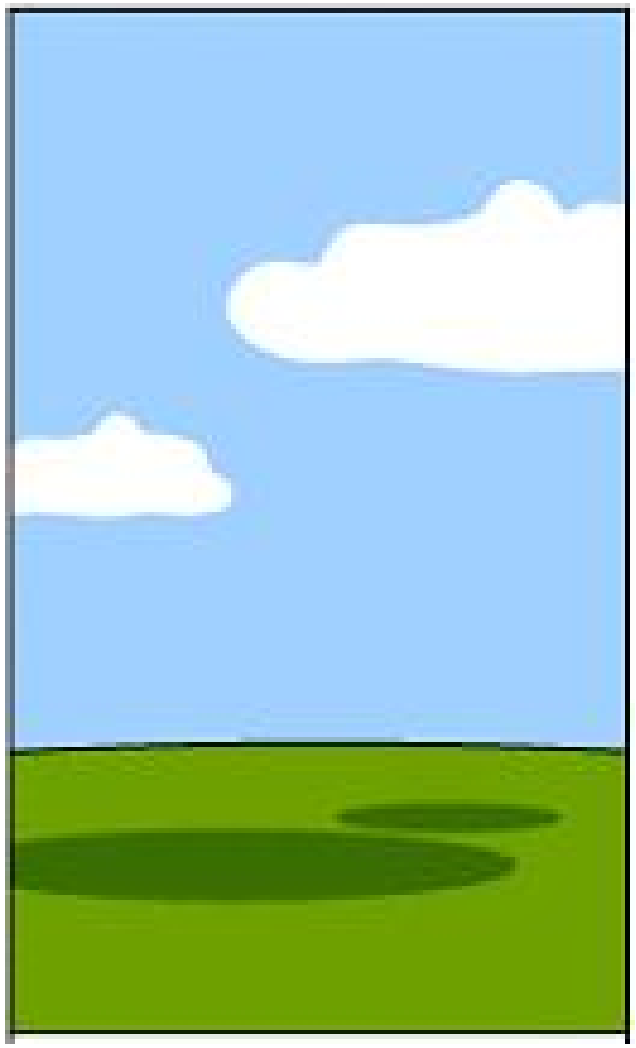
How the Architect Designed it

How the Developers coded it

Swing Project

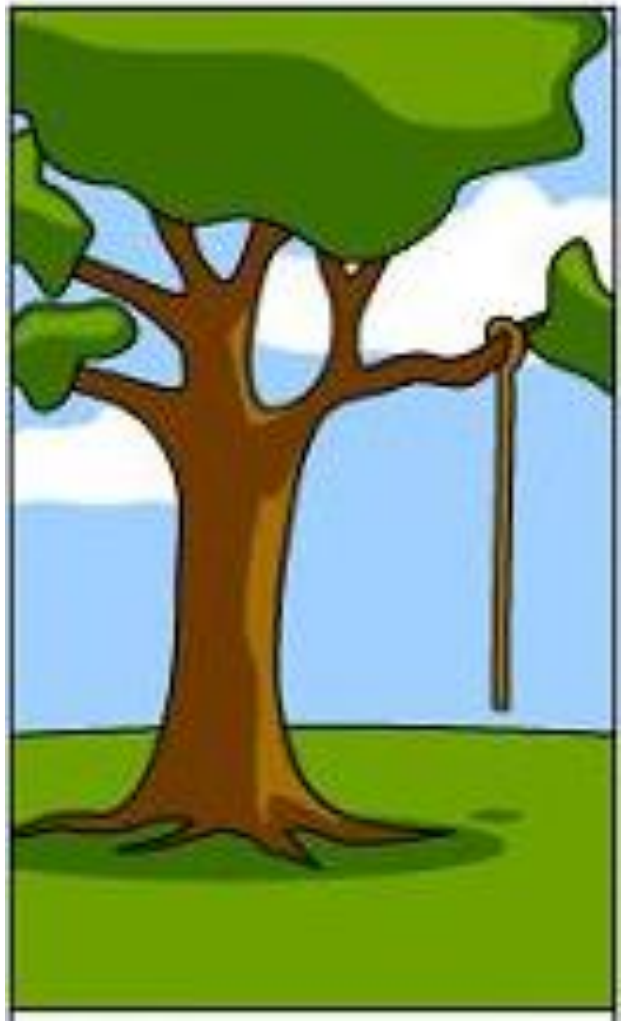
How the Marketing team described it to the customer

How the Project was Documented



Swing Project

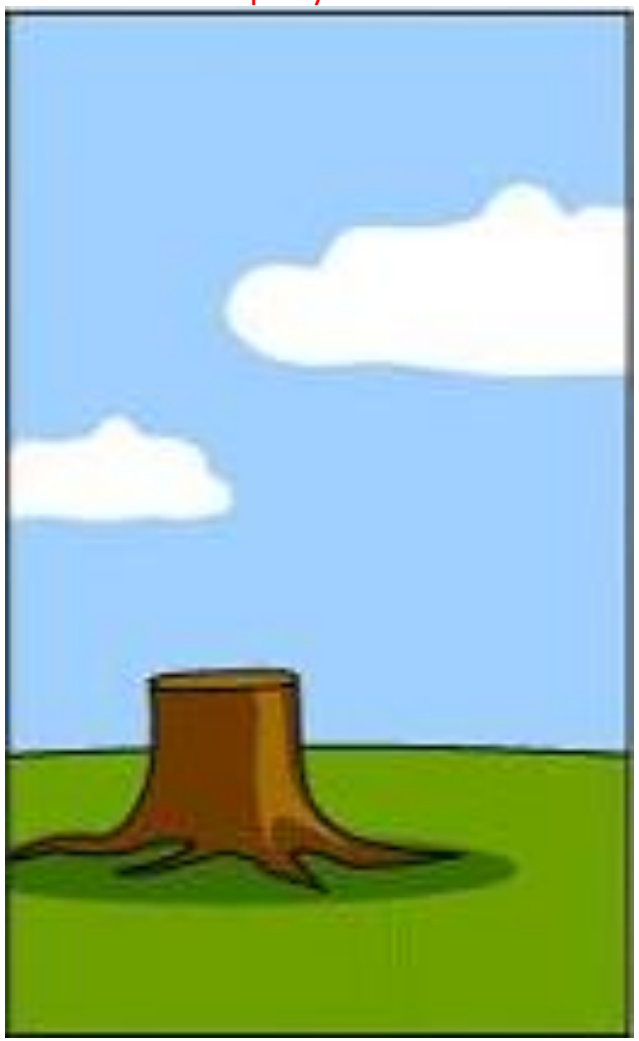
How it was Deployed at Customer Site



How the Customer was Billed

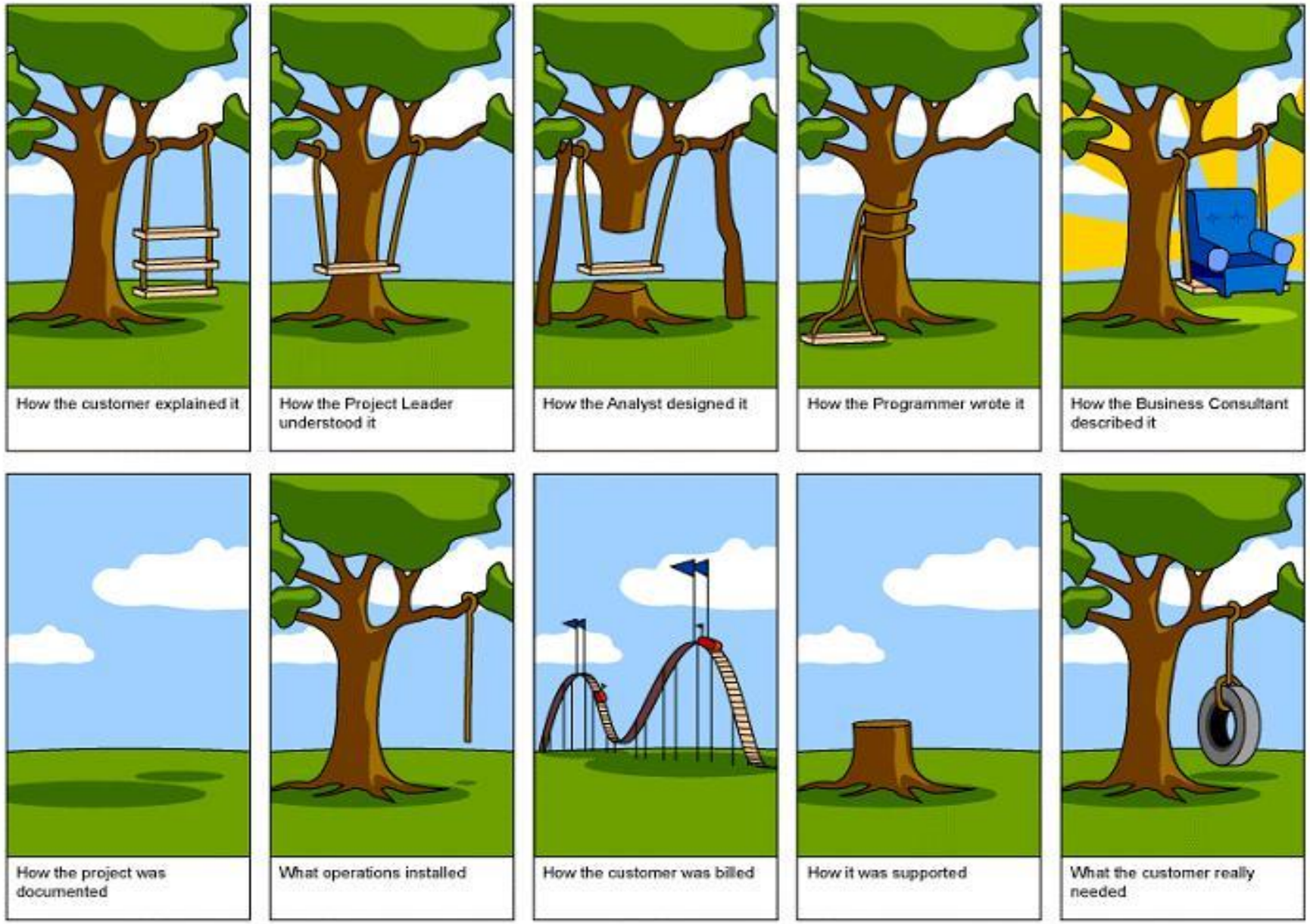
Swing Project

How it was Supported after Deployment



What Customers Really wanted!

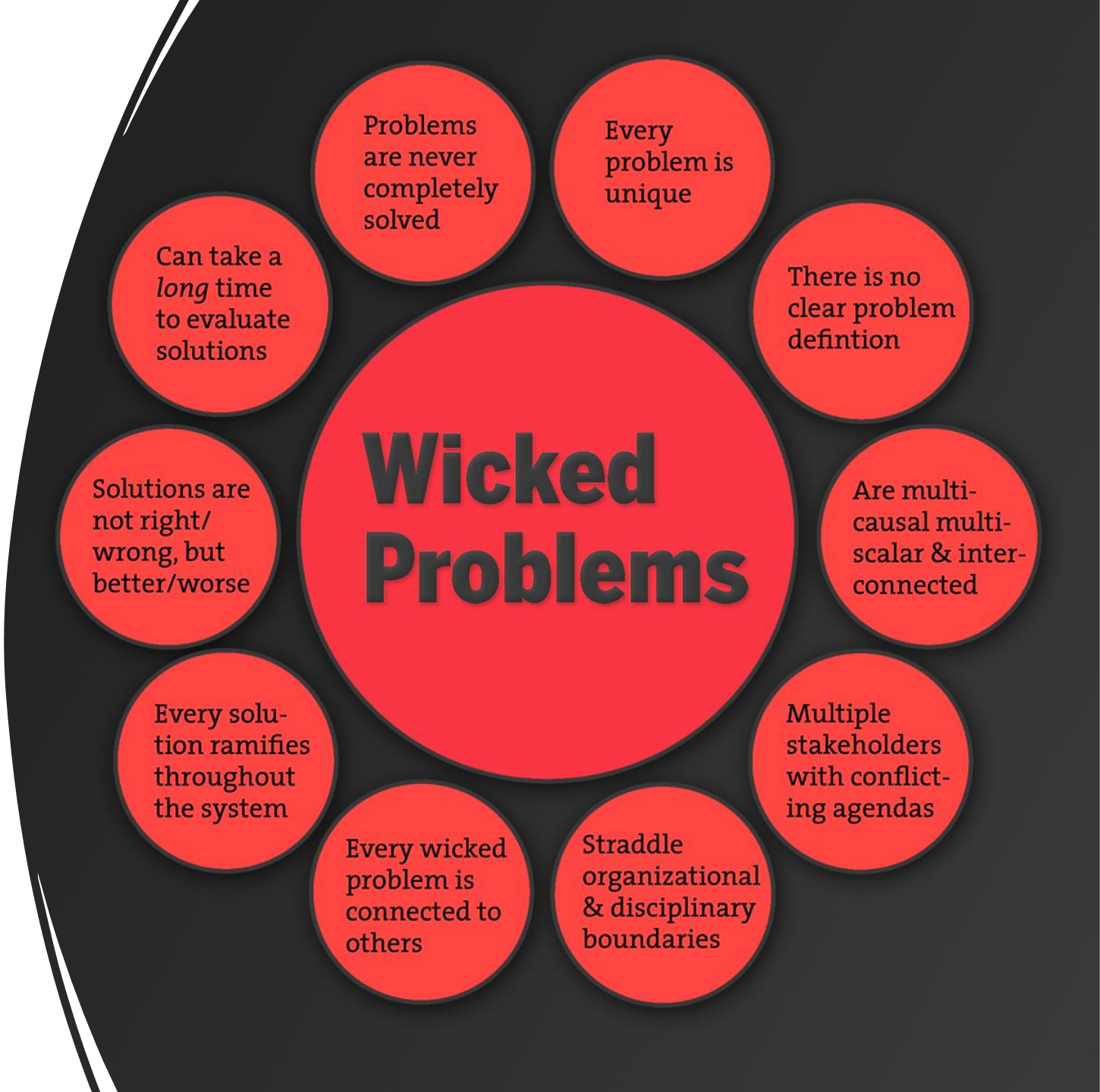
How the Swing Project happened!!!



Problems of Software Development

Problems of Software Development

- Large software is usually designed to solve 'wicked' problems

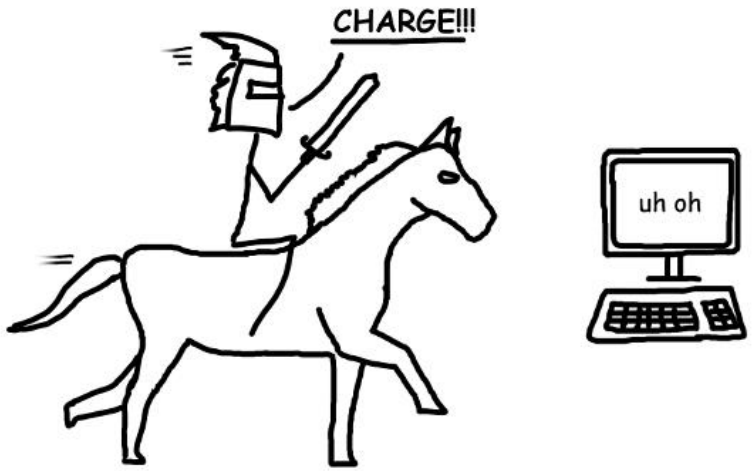


Problems of Software Development

- Software engineering requires a great deal of co-ordination across disciplines
  - Almost infinite possibilities for design trade-offs across components
  - Mutual distrust and lack of understanding across engineering disciplines

Problems of Software Development

- Systems must be designed to last many years in a changing environment



Problems of Software Development

- The process of efficiently and effectively developing requirements.

Problems of Software Development

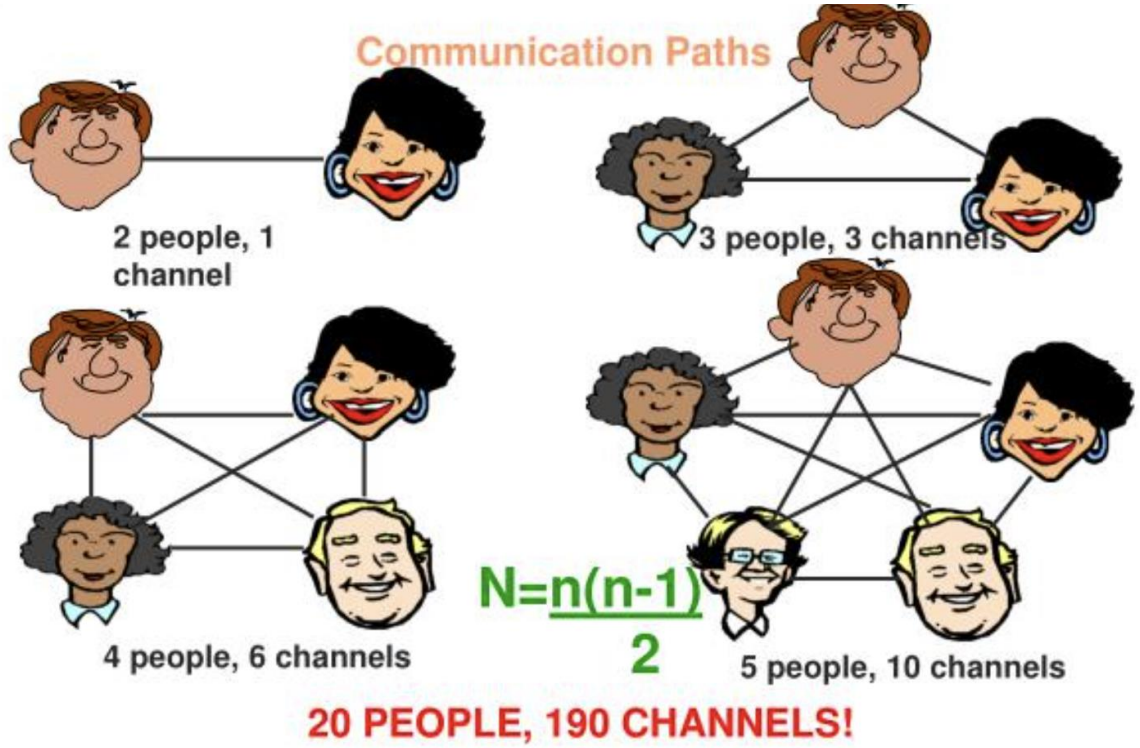
- Tooling required to create the solutions, may change as quick as the client's mind.

Problems of Software Development

- **User expectations:**
  - User expectations increase as the technology becomes more and more sophisticated.

Problems of Software Development

- **The mythical man-month factor:**
  - Adding personnel to a project may not increase productivity.
  - Adding personnel to a late project will just make it later.



Problems of Software Development

- **Communications:**
  - Communications among the various constituencies is a difficult problem. Sometimes different constituencies speak completely different languages.
  - For example, developers may not have the domain knowledge of clients and / or users.
  - The larger the project, the more difficult the communications problems become.

Problems of Software Development

- **Project characteristics:**
  - size / complexity
  - novelty of the application
  - response-time characteristics
  - security requirements
  - user interface requirements
  - reliability / criticality requirements

How to overcome these problems?

SOLUTION:
SOFTWARE ENGINEERING

How to overcome such Problems?

- Using a systematic process to produce high quality software products

What is Software Engineering?

Composed of two words ‘software’ and ‘engineering’.

Engineering forces us to focus on systematic, scientific and well defined processes to produce a good quality product.

What is Software Engineering?

- Software engineering is the establishment and use of sound engineering principles in order to obtain economical software that is reliable and works efficiently on real machines.
- Software Engineering is a profession dedicated to designing, implementing, and modifying,
  - High quality
  - More affordable
  - Maintainable software

Software Engineering - Evolution

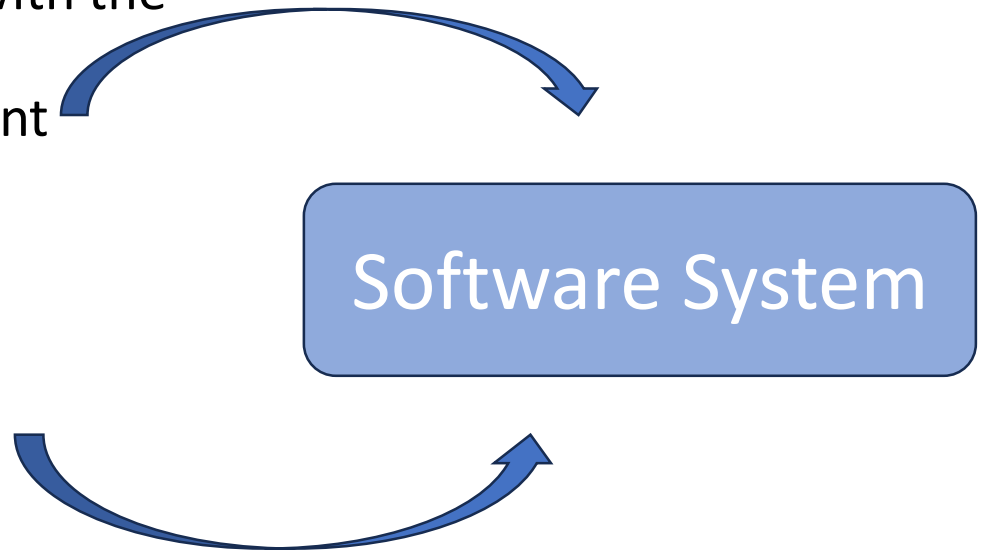
1940s and 1950s	1960s and 1970s	1980s and 1990s
Software development began as a single person activity in 1940s and 1950s.	Software engineering was considered a new scientific discipline in 1960s and 1970s.	In 1980s and 1990s engineering ideas dominated software development

Importance of Software Engineering

- The economies of ALL developed nations are dependent on software. More and more systems are software controlled
  - Transportation
  - Medical (Channeling, Pharmacy)
  - Telecommunications
  - Military
  - Industrial (Car Manufacturing)
  - Entertainment (Video and Audio Players)
- Software is found in products and situations where very high reliability is expected
  - E.g. Monitoring and controlling Nuclear power plants
- Contain millions of lines of code
- Comparably more complex

Importance of Software Engineering

- Concerned with the
  - Conception
  - Development
  - Verification
- Deals with
  - Identifying
  - Defining
  - Realizing
  - Verifying



Importance of Software Engineering

- Software engineering is concerned with theories, methods and tools for professional software development, which helps to
  - reduce complexity
  - minimize software cost
  - decrease project time
  - handle big projects
  - develop reliable software
  - develop more effective software

Software Engineering: Definitions

- ‘Software engineering is concerned with the theories, methods and tools for developing, managing and evolving software products’

- I Sommerville

- ‘The practical application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate and maintain them’

- B.W.Boehm

Software engineering - Definitions

- ‘ The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines’

-F.L. Bauer

- ‘The application of systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software’

- IEEE Standard 610.12

What is Software Engineering?

An **engineering discipline** that is concerned with **all aspects of software production**

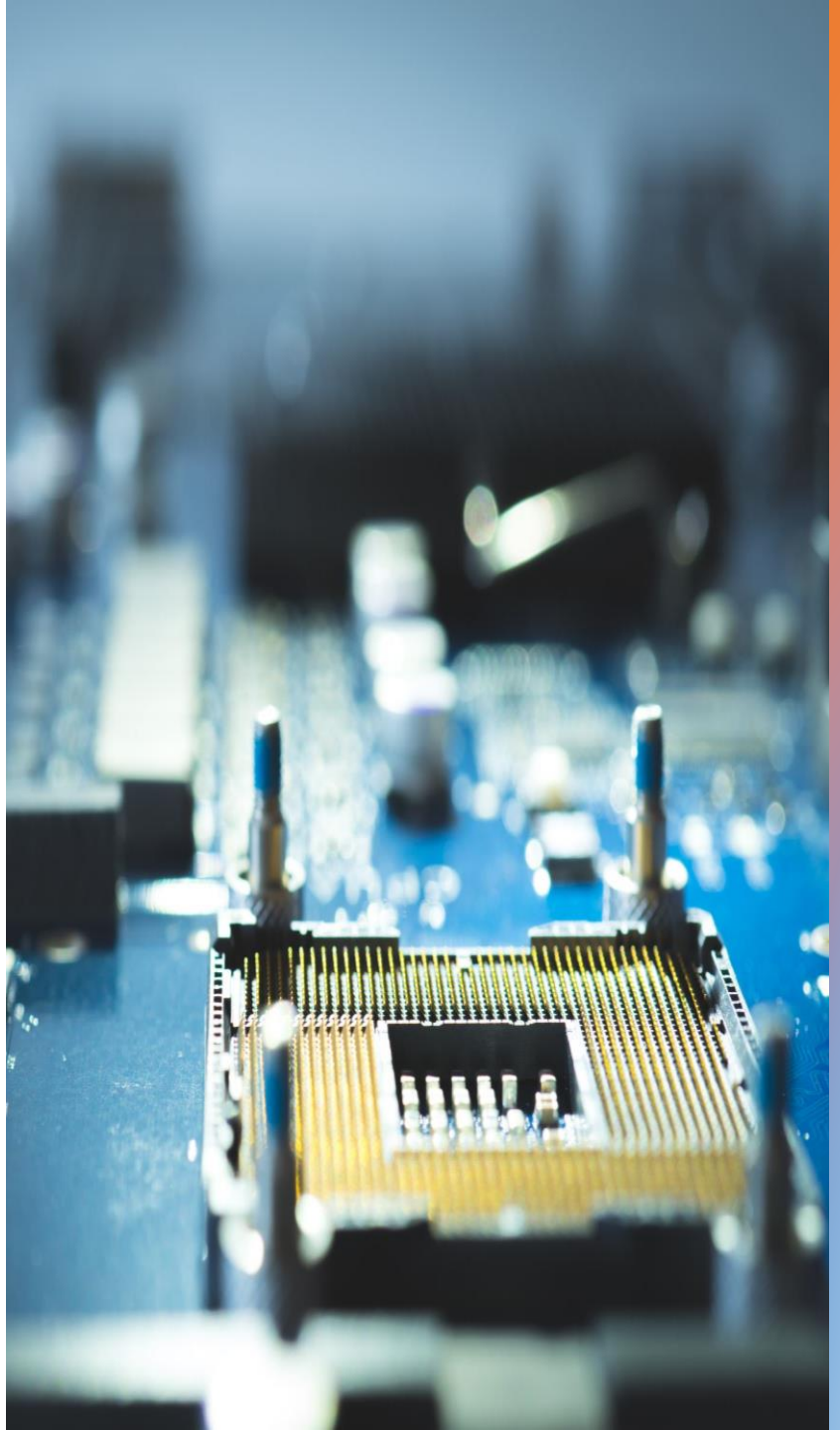
Software engineers should adopt a systematic and organized approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available.

Software engineering

In the given definition, there are two key phrases:

- **Engineering discipline** - Engineers make things work. They apply concepts, theories, methods and tools where theses are appropriate.

- **All Aspects of Software Production** – Software Engineering is not just concerned with the technical processes of software development. It also includes activities such as software project management, quality management



Software Engineering vs 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 still insufficient to act as a complete underpinning for software engineering

Software Engineering vs System 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 concerned with developing the software infrastructure, control, applications and databases in the system.

Software Quality Attributes

- *Software quality*
  - The degree in which software possesses a desired combination quality attributes.

Software Quality Attributes

The software should deliver the required functionality and performance to the user and should be maintainable, dependable and acceptable.

Maintainability

- Software must evolve to meet changing needs;

Dependability

- Software must be trustworthy;

Efficiency

- Software should not make wasteful use of system resources;

Acceptability

- Software must be accepted by the users for which it was designed. This means it must be understandable, usable and compatible with other systems.

Maintainability

The ease with which changes can be made to satisfy new requirements or to correct deficiencies.

Correctness

The degree with which software adheres to its specified requirements.

Reusability

The ease with which software can be reused in developing other software.

Reliability

The frequency and criticality of software failure, where failure is an unacceptable effect or behavior occurring under permissible operating conditions.

Portability

The ease with which software can be used on computer configurations other than its current one.

Efficiency

The degree with which software fulfills its purpose without waste of resources.

General Issues that affect Software

Heterogeneity

- Developing techniques for building software that can cope with heterogeneous platforms and execution environments

Business and social change

- Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

Delivery

- Developing techniques that lead to faster delivery of software

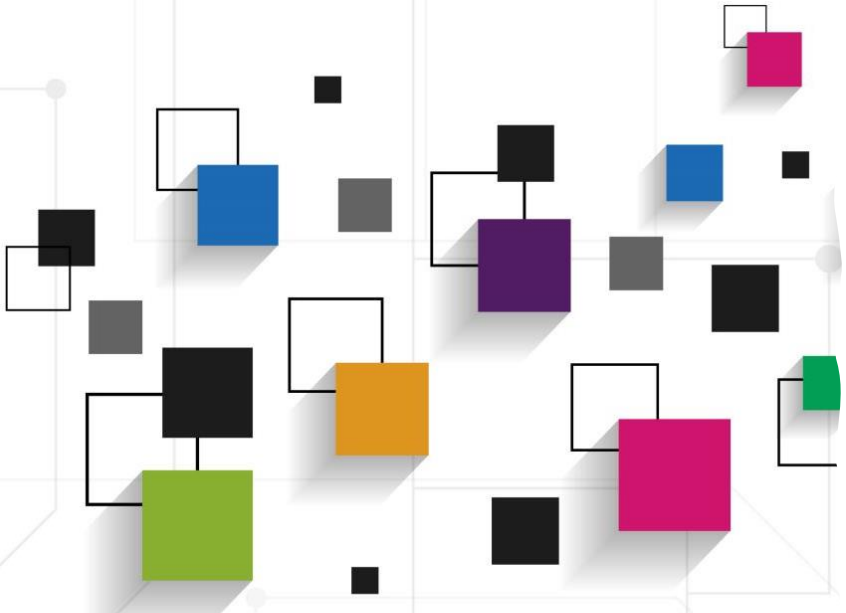
General Issues that affect Software

Scale

- Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet scale, cloud-based systems that serve a global community.

Security and Trust

- As software is intertwined with all aspects of our lives, it is essential that the developing techniques that demonstrate that software can be trusted by its users



Software Engineering Diversity

- There are many different types of software systems.
- There is no universal set of software techniques that are suitable for all systems and all companies.
- The software engineering methods and tools used depend on the type of application being developed, the requirements of the customer and the background of the development team.

Types of Applications

Stand-alone applications

- Application systems that run on a local computer, such as a PC. They include all necessary functionality and do not need to be connected to a network.

Interactive transaction-based applications

- Applications that execute on a remote computer and that are accessed by users from their own computers, phones, or tablets. These include web applications such as e-commerce applications.

Embedded control systems

- Software control systems that control and manage hardware devices. software in a mobile (cell) phone, software that controls antilock braking in a car, and software in a microwave oven to control the cooking process

Types of Applications

Batch processing systems

- Business systems that are designed to process data in large batches. E.g.: periodic billing systems, such as phone billing systems, and salary payment systems.

Entertainment systems

- Systems that are primarily for personal use and which are intended to entertain the user.

Systems for modeling and simulation

- Systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.

Types of Applications

Data collection systems

- Systems that collect data from their environment and send that data to other systems for processing.

Systems of systems

- These are systems that are composed of a number of other software systems. E.g. ERP system

Software Engineering Ethics

- Software engineering is carried out within a social and legal framework that limits the freedom of people working in that area.
- As a software engineer, your job involves wider responsibilities than simply the application of technical skills
- You should not use your skills and abilities to behave in a dishonest way

Definition: – Ethics is a set of moral principles that govern the behavior of a group or individual

Issues of Professional Responsibility

Confidentiality

Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.

Competence

Engineers should not misrepresent their level of competence. They should not knowingly accept work which is out with their competence.

Intellectual property rights

Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.

Computer misuse

Software engineers should not use their technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses).

ACM/IEEE Code of Ethics

- The professional societies in the US have cooperated to produce a code of ethical practice.
- Members of these organizations sign up to the code of practice when they join.
- The Code contains eight Principles related to the behavior of and decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.

Software Engineering Code of Ethics and Professional Practice

ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices

PREAMBLE

The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

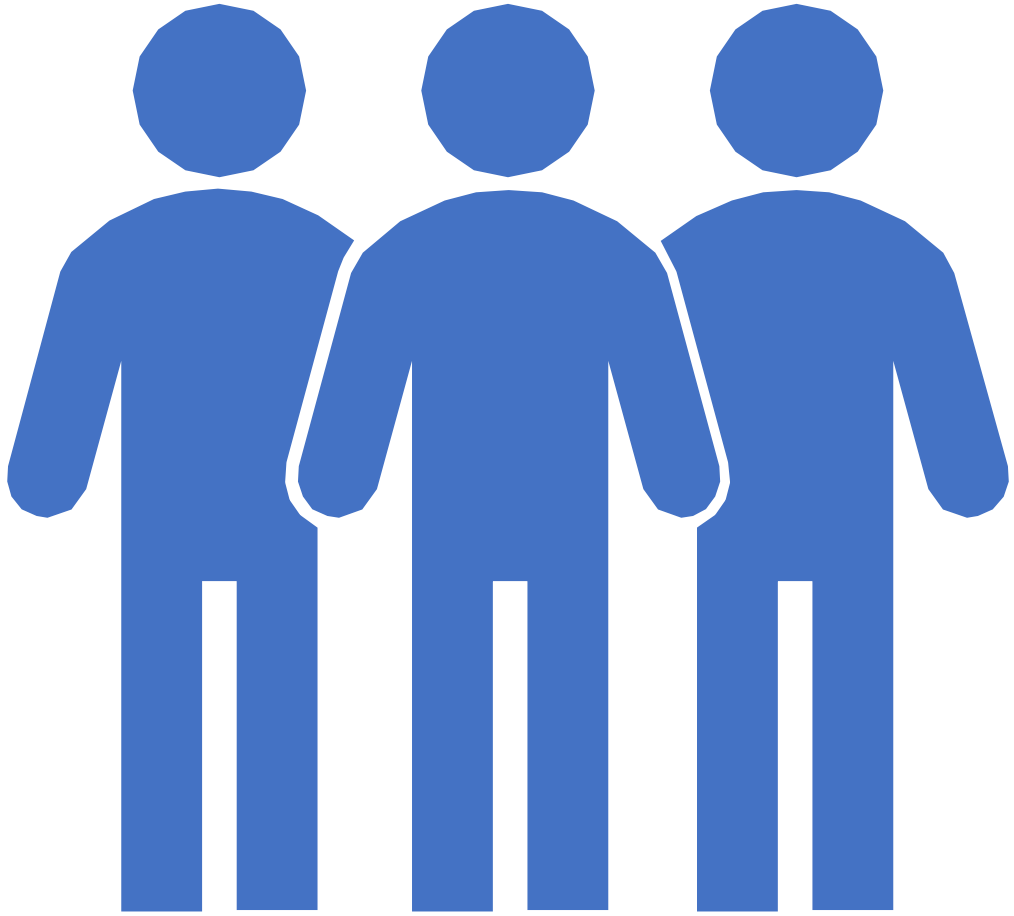
Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

Ethical Principles

1. PUBLIC - Software engineers shall act consistently with the public interest.
2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Ethical Problems

- Disagreement in principle with the policies of senior management.
- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system.
- Participation in the development of military weapons systems or nuclear systems.

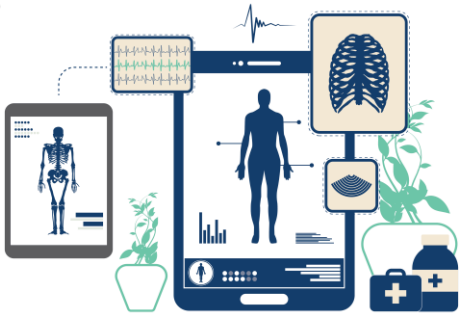


Safety-Critical System

- Safety critical system is a system whose failure or malfunction may result in one (or more) of the following outcomes:
  - death or serious injury to people
  - loss or severe damage to equipment/property
  - environmental harm

Examples

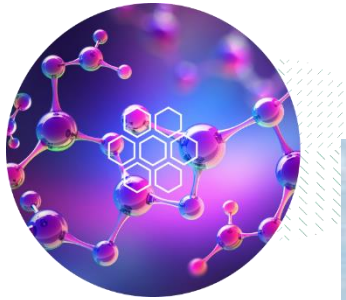
Medical Devices.



Aerospace

- Civil aviation.
- Military aviation.
- Manned space travel

Chemical Industry.



Nuclear Power Stations.

Automotive control systems

Traffic control.

- Railway control system.
- Air traffic control.
- Road traffic control (esp. traffic lights).

CASE STUDIES

REF: SOFTWARE ENGINEERING BY IAN SOMERVILLE

Case Studies

- A personal insulin pump
  - An embedded system in an insulin pump used by diabetics to maintain blood glucose control.
- A mental health case patient management system
  - Mentcare. A system used to maintain records of people receiving care for mental health problems.
- A wilderness weather station
  - A data collection system that collects data about weather conditions in remote areas.
- iLearn: a digital learning environment
  - A system to support learning in schools

Mentcare: A patient information system for mental health care

- A patient information system to support mental health care is a medical information system that maintains information about patients suffering from mental health problems and the treatments that they have received.
- Most mental health patients do not require dedicated hospital treatment but need to attend specialist clinics regularly where they can meet a doctor who has detailed knowledge of their problems.
- To make it easier for patients to attend, these clinics are not just run in hospitals. They may also be held in local medical practices or community centers.

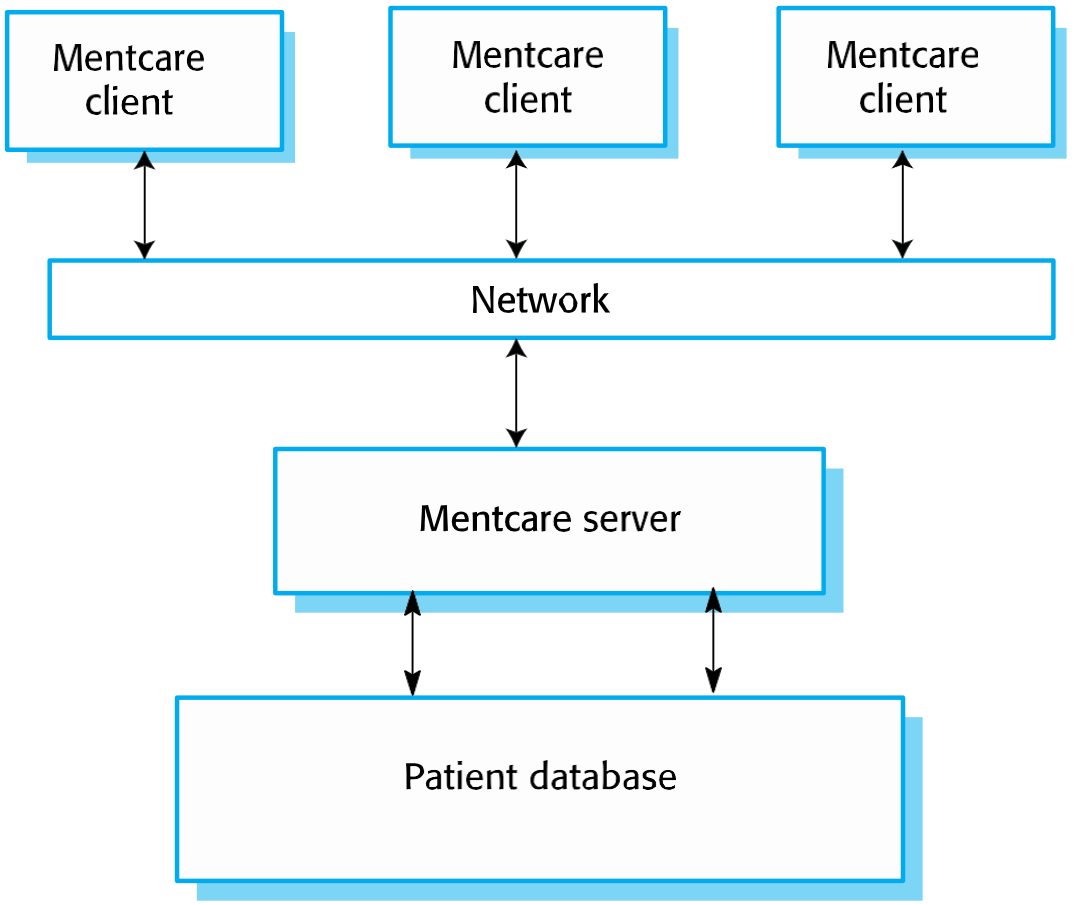
Mentcare

- Mentcare is an information system that is intended for use in clinics.
- It makes use of a centralized database of patient information but has also been designed to run on a PC, so that it may be accessed and used from sites that do not have secure network connectivity.
- When the local systems have secure network access, they use patient information in the database but they can download and use local copies of patient records when they are disconnected.

Mentcare goals

- To generate management information that allows health service managers to assess performance against local and government targets.
- To provide medical staff with timely information to support the treatment of patients.

The organization of the Mentcare system



Key features of the Mentcare system

- Individual care management
  - Clinicians can create records for patients, edit the information in the system, view patient history, etc. The system supports data summaries so that doctors can quickly learn about the key problems and treatments that have been prescribed.
- Patient monitoring
  - The system monitors the records of patients that are involved in treatment and issues warnings, if possible, problems are detected.
- Administrative reporting
  - The system generates monthly management reports showing the number of patients treated at each clinic, the number of patients who have entered and left the care system, number of patients sectioned, the drugs prescribed and their costs, etc.

Mentcare system concerns

- Privacy
  - It is essential that patient information is confidential and is never disclosed to anyone apart from authorized medical staff and the patient themselves.

- Safety
  - Some mental illnesses cause patients to become suicidal or a danger to other people. Wherever possible, the system should warn medical staff about potentially suicidal or dangerous patients.
  - The system must be available when needed otherwise safety may be compromised and it may be impossible to prescribe the correct medication to patients.

MCQs

Which of the following is / are true regarding software engineering concepts?

1. Packaged software is another name given for generic software.
2. Packaged software is another name given for customized software.
3. Software engineering is concerned with theories, methods and tools for professional software development.
4. Elegant theories of computer science cannot always apply to real, complex problems that require a software solution.
5. Software engineering is an older discipline when compared to system engineering.

MCQs -Answer

Which of the following is / are true regarding software engineering concepts?

1. **Packaged software is another name given for generic software.**
2. Packaged software is another name given for customized software.
3. **Software engineering is concerned with theories, methods and tools for professional software development.**
4. **Elegant theories of computer science cannot always apply to real, complex problems that require a software solution.**
5. Software engineering is an older discipline when compared to system engineering.

Home work

- Search for more “software disasters”
- Write short notes on main software quality attributes.
- Try to Find some more software quality attributes and write short notes
- ‘All Microsoft software is bug free ‘ – Bill Gates
  - Is he correct? Explain your answer.

Summary

Software engineering is an engineering discipline that is concerned with all aspects of software production.

Software products consist of developed programs and associated documentation.

Essential product attributes are maintainability, dependability, efficiency and usability

Software engineers have responsibilities to the engineering profession and society. They should not simply be concerned with technical issues.