Software Engineering

Overall Objectives:

After successfully completing this subject, students should be able to:

- **Explain the software engineering principles and techniques in developing quality software products**
- **Apply software engineering principles and techniques appropriately in developing a moderately complex software system**

Main References

1. Ian Sommerville, Software Engineering, 9th Edition, Addison Wesley, 2011.

2. R Pressman, Software Engineering - A Practitioners Approach, 7th Edition, McGraw Hill.

Software Engineering

Introduction

Learning Outcomes

- Appreciate the problems associated with developing software
- Understanding the need for a managed approach to software development
- Be able to define the term ‘Software Engineering’
- Identify software quality attributes and their classification

Types of Software Products

- **Generic Products**

  These are produced by a development organization and sold on the open market to any customer who is able to buy them.

  Eg. Packages : spreadsheet. databases, word processors, library systems etc.

- **Customized (Bespoke) Products**

  These are systems that are commissioned by a particular customer. A software contractor develops the software specially for that customer.

Software

Instructions given to a computer (computer programs) together with relevant documentation

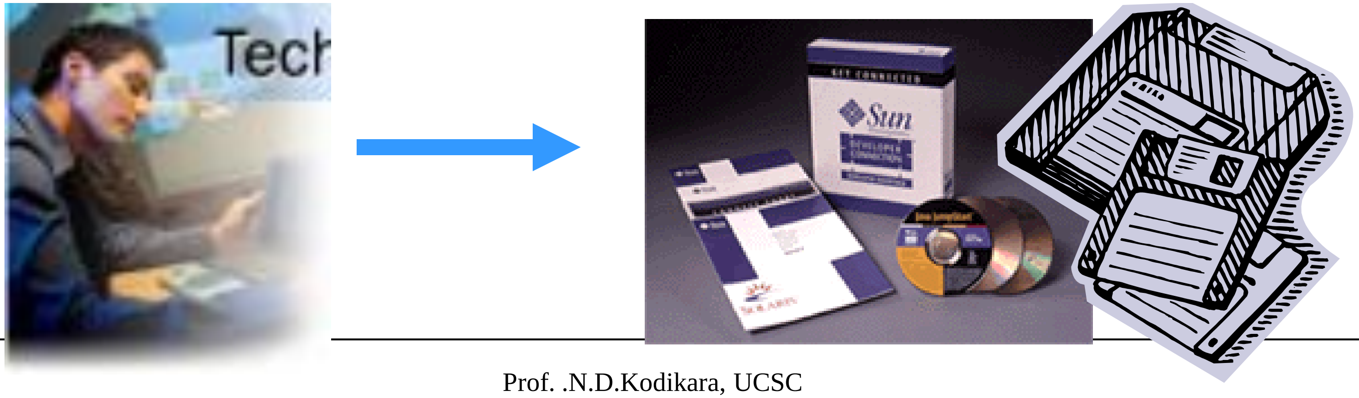
- Systems software
- Application software

Types of software

- System Software:

System software is a collection of programs written to service the other programs.

eg. Operating systems, drivers, compilers communication software.



Types of software…

- Business software:

System have evolved into management information system software that access one or more large database containing business information.

- Embedded software:

Embedded software resides in read-only memory and is used to control product and system for the customer and industrial markets.

Types of software…

Engineering and scientific software:

They have been characterized by number crunching algorithms.

Personal computer software:

Personal computer software market has burgeoned over the past two decades. Word processing, spreadsheets, computer graphic, multimedia and db management

Types of software…

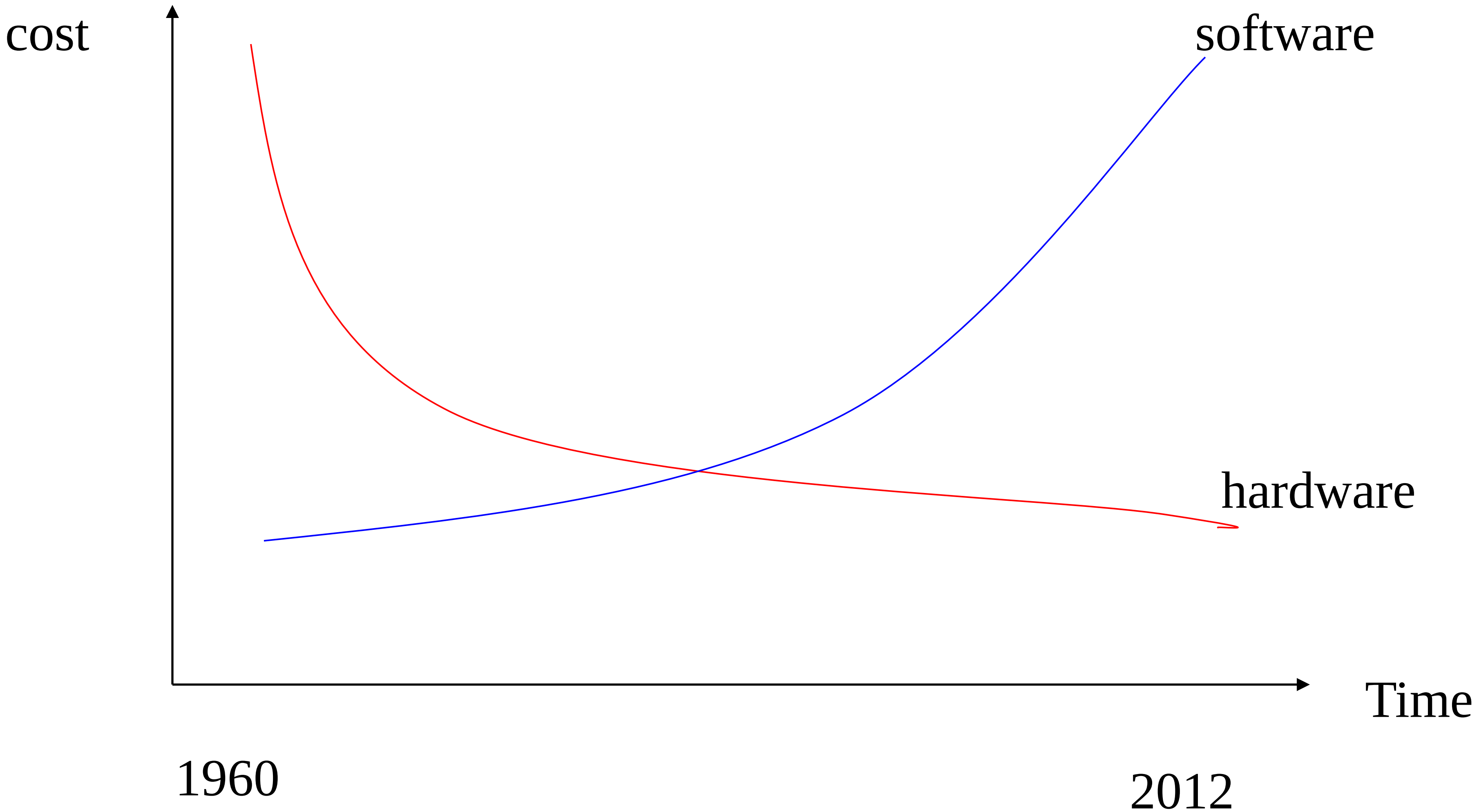
- Web-based software:

The network becomes a massive computer providing an almost unlimited software resources that can be accessed by anyone with a modem.

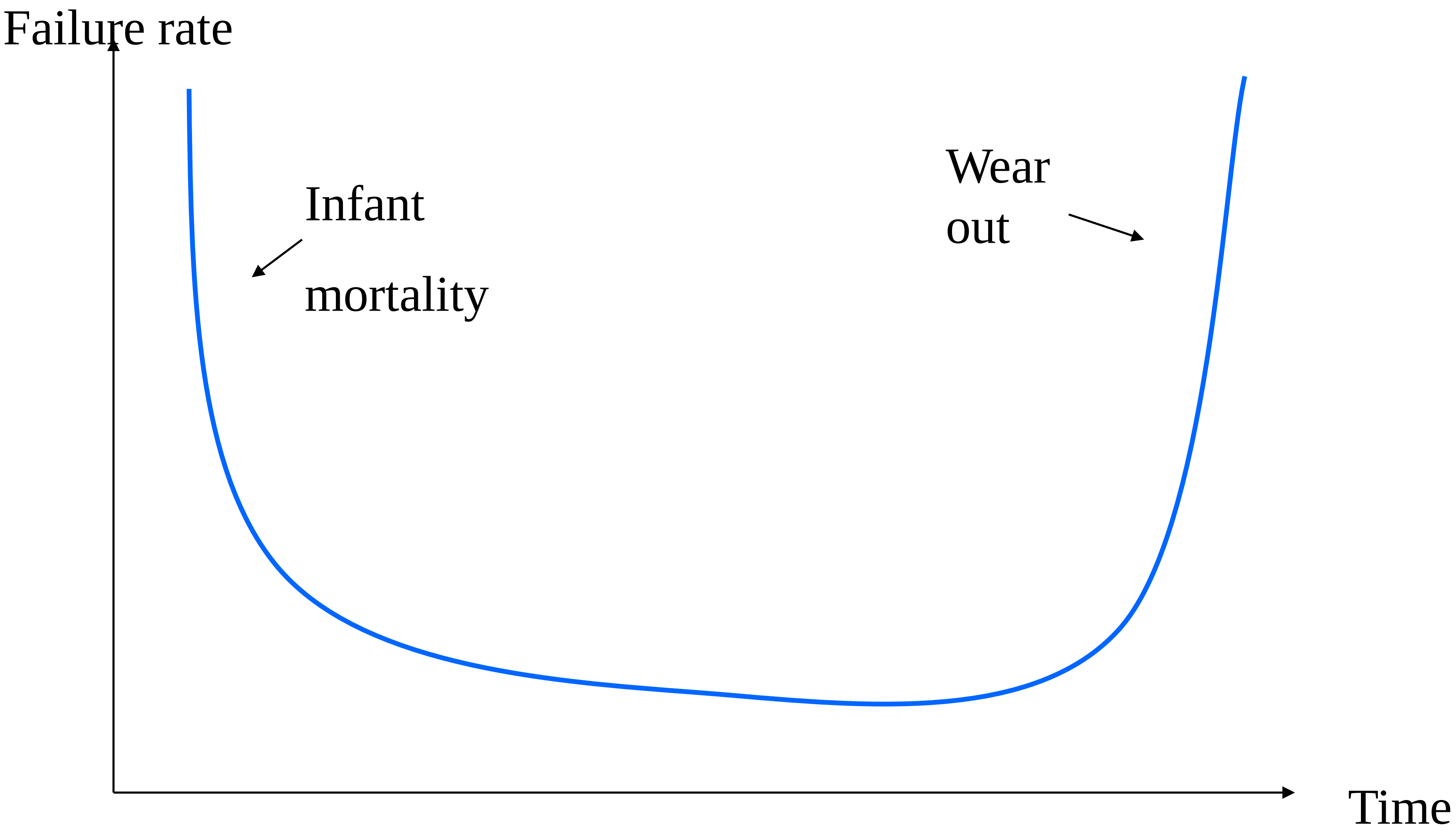
- Artificial intelligence software:

AI software makes use of nonnumerical algorithms to solve the complex problems that are not amenable to computing or straightforward analysis.

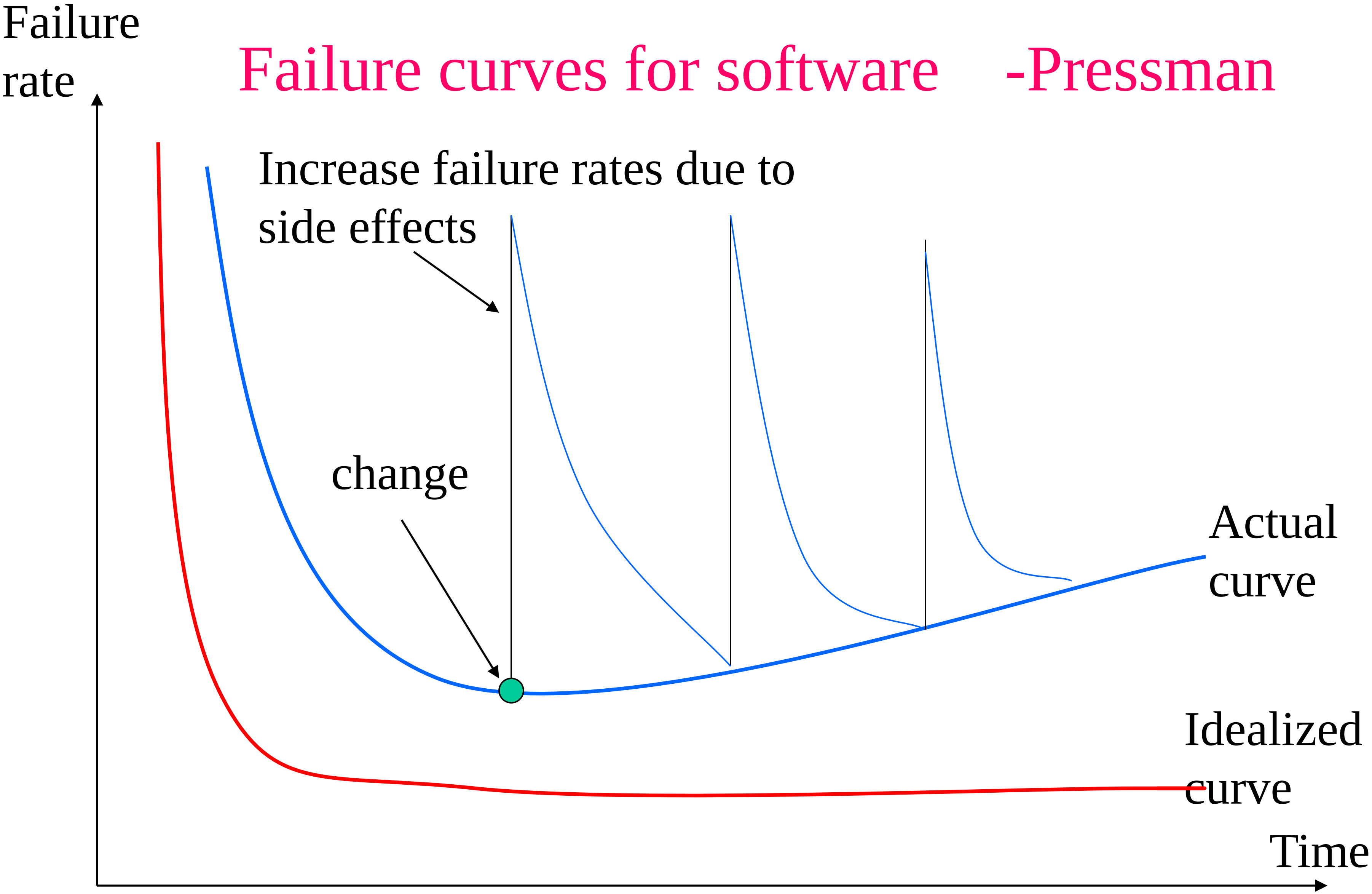
Hardware vs Software



Failure curve for hardware [Pressman]



Failure curves for software -Pressman



Development Failures

IBM Survey, 2000

- 55% of systems cost more than expected
- 68% overran the schedules
- 88% had to be substantially redesigned

Bureau of Labour Statistics (2001)

- for every 6 new systems put into operation, 2 cancelled
- probability of cancellation is about 50% for large systems
- average project overshoots schedule by 50%

Over Budget

Home Office IT project millions over budget

Home Office (UK) IT project run by Bull Information Systems is expected to blow its budget by millions of pounds and is hampered by a restrictive contract, according to a leaked report. The National Audit Office Report is expected to reveal damning evidence that the project to implement two systems – the National Probation Service Information System, and the Case Record and Management System will cost 118m pounds by the end of the year, 70% over its original budget.

www.computing.co.uk/News/111627

Over Schedule

New air traffic system is already obsolete

National Air Traffic Services (Nats) is already looking at replacing the systems at its new control center at Swanwick in Hampshire, even though the system doesn’t become operational until next week. This project is six years late and 180m pounds over budget.

Swanwick was originally meant to be operational by 1997, but problems with the development of software by Lockheed Martin caused delays, according to Nats.

www.vnunet.com/News/1128597

Safety – London Ambulance Dispatching System

The full introduction of the computer system effectively did away with the radio and telephone calls to stations, with the computer dispatching crews to answer calls. But within hours, during the morning rush, it became obvious to crews and control room staff that calls were going missing in the system; ambulances were arriving late or doubling up on calls. Distraught emergency callers were also held in a queuing system which failed to put them through for up to 30 minutes. Chris Humphreys, Nupe's divisional officer, said that it was hard to verify how many people might have died because of the delays but it could be as many as 20.

Causes : The managers who produced the software were naïve. They made a terrible mistake of trying to go on-line abruptly, without running the new and old systems together for a while.

http:/128.240.150.127/Risks/13.88.html#subj1.1

Programming/testing Error – Ariane 5 (June 1996)

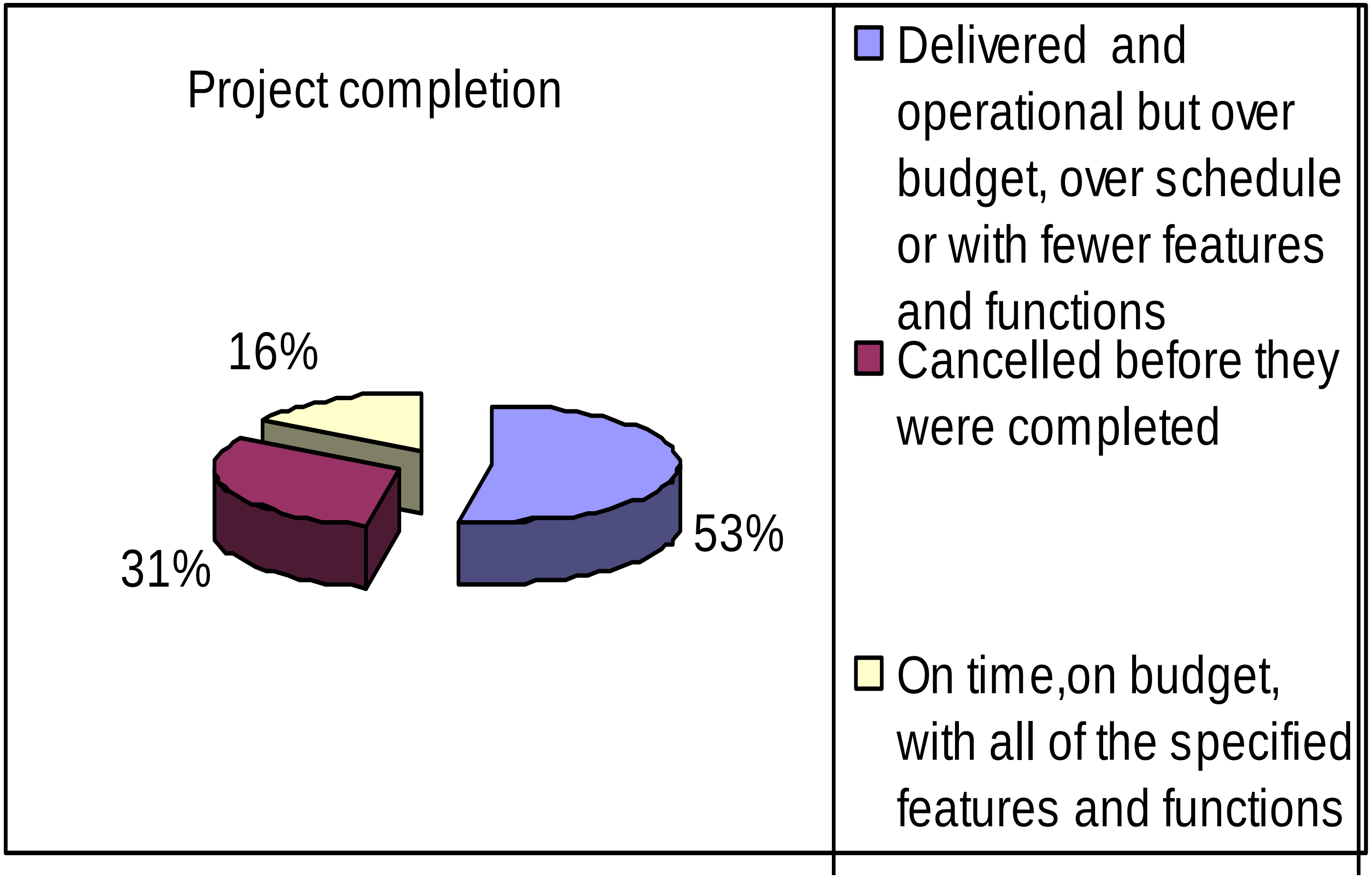
It took the European Space Agency 10 years and $7 billion to produce Ariane 5, a giant rocket capable of hurling a pair of three ton satellites into orbit.

At 39 seconds after launch, as the rocket reached an altitude of two and a half miles, a self-destruct mechanism finished off Ariane 5, along with its payload of two expensive and uninsured scientific satellites. The rocket was making an abrupt course correction that was not needed, compensating for a wrong turn that had not taken place.

The cause (Ariane –5)

Steering was controlled by the on-board computer, which mistakenly thought the rocket needed a course change because of the numbers, which in fact was an error, coming from the inertial guidance system. The guidance system had in fact shut down 36.7 seconds after launch, when the guidance system’s own computer tried to convert one piece of data – the sideways velocity of the rocket – from a 64 bit format to a 16 bit format = overflow error.

The Statistics



Distribution of software cost over life cycle

1. Requirements capture
2. Requirement specification 14%
3. Design
4. Implementation
5. Testing 6%

40%

6. Maintenance 60%

Software Problems

1. Time Schedules and cost estimates of many software projects are grossly inaccurate.

2. Software is costly.

3. The quality of software is not satisfactory.

4. Software is difficult to maintain.

5. The productivity of software people is not satisfactory to meet the demand.

What makes software special?

The main difference in software engineering compared to other engineering disciplines are listed below.

1. It is difficult for a customer to specify requirements completely.
2. It is difficult for the developer to understand fully the customer needs.
3. Software requirements change regularly.
4. Software is primarily intangible; much of the process of creating software is also intangible, involving experience, thought and imagination.
5. It is difficult to test software exhaustively.

A Solution – Software Engineering

- Greater emphasis on systematic , scientific development.
- Computer assistance in software development (CASE)
- A concentration on finding out the user’s requirements
- Formal/Semi Formal specification of the requirements of a system.
- Demonstration of early version of a system (prototyping)
- Grater emphasis on development of error free easy to understand code.

Evolution of software engineering

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

Software Engineering

Definitions

- Simple definition : Designing, building and maintening large software systems.
- Others

=> ‘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

=> ‘ 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

Software Engineering

Software Engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.

Software Engineering

In the given definition, there are two key phrases:

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

2. **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 Ethics

Like other engineering disciplines software engineering is carried out within a social and legal framework. Software engineers must accept that their job involves wider responsibilities than simply the application of technical skills. Some of these are:

- Confidentiality
- Competence
- Intellectual property rights
- Computer misuse

Software Quality Attributes

Bohem’s Classification

- **Current usefulness**

  - The qualities expected from a software system in user’s point of view.

- **Potential Usefulness**

  - The qualities expected from a software system in developer’s point of view.

Current usefulness

- Efficiency
- Reliability
- Usability
- Correctness
- User friendliness
- Robustness

Potential usefulness

- Maintainability
- Modularity
- Reusability
- Portability

McCall’s Classification

- Product operation
- Product revision
- Product transition

Product Operation

- Efficiency
- Correctness
- User friendliness
- Usability
- Reliability
- Robustness

Product Revision

- Maintainability
- Flexibility
- Testability

Product Transition

- Interoperability
- Reusability
- Portability

Software Maintenance

Any software system needs to be changed. Maintenance is the costliest operation in the software development process (about 60%). Software need to be changed due to various reasons.

- Errors in the system.
- Changes in the user requirements
- Availability of new technology
- Changes in the enterprise or Govt. policy.

Review Questions

1. Look at the following list of goals of software

   a) Meeting the user’s needs

   b) Low cost of production

   c) High Performance

   d) Portability

   e) Low cost of maintenance

   f) High Reliability

   e) Delivery on time

For each of the following systems choose the two most important goals.

(i) A system to manage student fees – to track whether they have paid, calculate the correct amount, issue invoices, send out reminders etc.

(ii) A system to manage the temperature of a nuclear reactor core.

(iii) An experimental program to test the efficiency of an algorithm for some mathematical theory.

(iv) A program that allows viewing of cricket scores on mobile phones.

2. Which stage of software development is most expensive?
3. Where are the most errors introduced in software projects?
4. ‘All Microsoft software is bug free ‘ – Bill Gates

   Is he correct?
5. Describe all software quality attributes given in this lesson in detail.