Software Engineering & Project Management

Prof. Yashwant Dongre
Yashwant.dongre@viit.ac.in
Department of Computer Engineering

BRACT'S, Vishwakarma Institute of Information Technology, Pune-48

(An Autonomous Institute affiliated to Savitribai Phule Pune University)
(NBA and NAAC accredited, ISO 9001:2015 certified)

Last Session Recap

(Applicable for subsequent sessions)

1.
2.
3.

Objective/s of this session

1.
2.
3.

Learning Outcome/Course Outcome

1.
2.
3.

Content (for example)

- Nature of Software
- Software Engineering Principles
- Software Process
- Software Myths

Software engineering

- The economies of ALL developed nations are dependent on software
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development.
- Expenditure on software represents a significant fraction of GDP in all developed countries

FAQs about software engineering

- What is software?
- What is software engineering?
- What is the difference between software engineering and computer science?
- What is the difference between software engineering and system engineering?
- What is a software process?
- What is a software process model?

FAQs about software engineering

- What are the costs of software engineering?
- What are software engineering methods?
- What is CASE (Computer-Aided Software Engineering)
- What are the attributes of good software?
- What are the key challenges facing software engineering?

What is software?

- Computer programs and associated documentation such as requirements, design models and user manuals.
- 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 e.g. PC software such as Excel or Word.
  - **Custom** - developed for a single customer according to their specification.
- New software can be created by developing new programs, configuring generic software systems or reusing existing software.

What is Software?

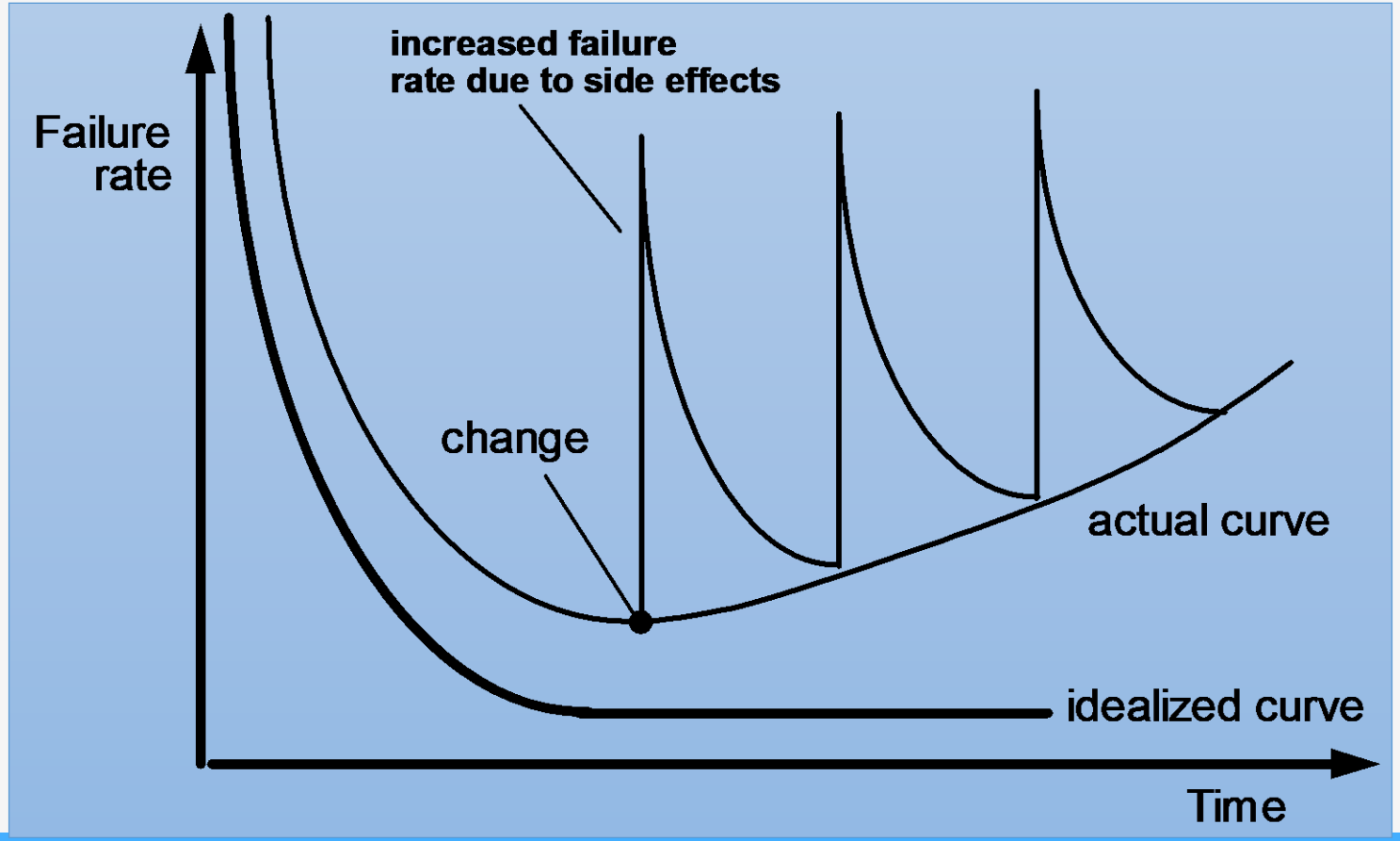
Software is a set of items or objects that form a “configuration” that includes

- programs
- documents
- data

Software's Dual Role

- **Software is a product**
  - ➢Delivers computing potential
  - ➢Produces, manages, acquires, modifies, displays, or transmits information
- **Software is a vehicle for delivering a product**
  - ➢Supports or directly provides system functionality
  - ➢Controls other programs (e.g., an operating system)
  - ➢Effects communications (e.g., networking software)
  - ➢Helps build other software (e.g., software tools)

Wear vs. Deterioration



What is software engineering?

- Software engineering is an engineering discipline that 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.

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 foundation for software engineering (unlike e.g. physics and electrical 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.
- System engineers are involved in system specification, architectural design, integration and deployment.

What is a software process?

- A set of activities whose goal is the development or evolution of software.
- Generic activities in all software processes are:
  - ❑**Specification -** what the system should do and its development constraints
  - ❑**Development -** production of the software system
  - ❑**Validation -** checking that the software is what the customer wants
  - ❑**Evolution -** changing the software in response to changing demands.

What is a software process model?

- A simplified representation of a software process, presented from a specific perspective.
- Examples of process perspectives are
  - Workflow perspective - sequence of activities;
  - Data-flow perspective - information flow;
  - Role/action perspective - who does what.
- Generic process models
  - Waterfall;
  - Iterative development;
  - Component-based software engineering.

What are software engineering methods?

- Structured approaches to software development which include system models, notations, rules, design advice and process guidance.
- Model descriptions
  - ➢Descriptions of graphical models which should be produced;
- Rules
  - ➢Constraints applied to system models;
- Recommendations
  - ➢Advice on good design practice;
- Process guidance
  - ➢What activities to follow.

Attributes of good software

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

Key challenges facing software engineering

- Heterogeneity, delivery and trust.
- Heterogeneity
  - ➢Developing techniques for building software that can cope with heterogeneous platforms and execution environments;
- Delivery
  - ➢Developing techniques that lead to faster delivery of software;
- Trust
  - ➢Developing techniques that demonstrate that software can be trusted by its users.

Software Applications

- system software
- application software
- engineering/scientific software
- embedded software
- product-line software
- WebApps (Web applications)
- AI software

Software—New Categories

- **Ubiquitous computing**—wireless networks
- **Netsourcing**—the Web as a computing engine
- **Open source**—”free” source code open to the computing community (a blessing, but also a potential curse!)
- **Data mining**
- **Grid computing**
- **Cognitive machines**
- **Software for nanotechnologies**

Legacy Software

- Software must be **adapted** to meet the needs of new computing environments or technology.
- Software must be **enhanced** to implement new business requirements.
- Software must be **extended to make it interoperable** with other more modern systems or databases.
- Software must be **re-architected** to make it viable within a network environment**.**

Software Myths

- It affects managers, customers (and other non-technical stakeholders) and practitioners
- These are believable because they often have elements of truth, but invariably lead to bad decisions,
- Therefore, it insists on reality as you navigate your way through software engineering

Management Myths

- **Myth :** We already have a book that's full of standards and procedures for building software. Won't that provide my people with everything they need to know ?
- **Reality :** The book of standards may very well exist, but is it used ? Are software practitioners aware of its existence ? Does it reflect modern software development practice ? Is it complete ? In many cases, the answer to all of these questions is "no".

- **Myth :** My people do have state-of-the-art software development tools, after all, we buy them from the newest computers.
- **Reality :** It takes much more than the latest model mainframe, workstation, or PC to do high quality software development.

Management Myths

- **Myth :** If we get behind schedule, we can add more programmers and catch up.
- **Reality :** Software development is not a mechanistic process like manufacturing. However, as new people are added, people who were working must spend time educating the newcomers, thereby reducing the amount of time spent on productive development effort.

Customer Myths

- **Myth :** A general statement of objectives is sufficient to begin writing programs.
- **Reality :** Poor-up-front definition is the major cause of failed software efforts. A formal and detailed description of information domain, function, performance, interfaces, design constraints, and validation criteria is essential.

- **Myth :** Project requirements continuously change, but change can be easily accommodated because software is flexible.
- **Reality :** It is true that software requirements do change, but the impact of change varies with the time at which it is introduced.

Practitioner Myths

- **Myth:** Software engineering will make us create voluminous and unnecessary documentation and will invariably slow us down.
- **Reality:** Software engineering is not about creating documents. It is about creating a quality product. Better quality leads to reduced rework. And reduced rework results in faster delivery times.

- **Myth :** Once the program is written and running, my job is done.
- **Reality :** Someone once said that “the sooner you begin ‘writing code,’ the longer it’ll take you to get done.” Industry data indicate that between 60 and 80 percent of all effort expended on software will be expended after it is delivered to the customer for the first time.

Instructions/Guidelines/References

1.
2.
3.
4.

Wrap up and related outcomes

1.
2.
3.
4.

To be discussed next time

Thank You