CSE 423: Software Engineering

Introduction to Software Engineering

Tajkia Nuri Ananna

January 31, 2023

Lecturer, Metropolitan University

Table of contents

1. Some Questions

2. Software Myths

3. General Issues

4. Software engineering ethics

Some Questions

Some Questions i

What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times, and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs; 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.

Software Myths

Software Myths i

Software Myth

Software Myths: Wrong beliefs about software and the process that is used to build it

Management Myths

Myth

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 engineering practice? Is it complete? Is it adaptable? Is it streamlined to improve time-to-delivery while still maintaining a focus on quality?

Management Myths

Myth

Myth: If we get behind schedule, we can add more programmers and catch up (sometimes called the “Mongolian horde” concept).

Reality

Software development is not a mechanistic process like manufacturing. People can be added but only in a planned and well coordinated manner.

Customer Myths

Myth

Myth: A general statement of objectives is sufficient to begin writing programs—we can fill in the details later.

Reality

Although a comprehensive and stable statement of requirements is not always possible, an ambiguous “statement of objectives” is a recipe for disaster. Unambiguous requirements (usually derived iteratively) are developed only through effective and continuous communication between customer and developer

Customer Myths

Myth

Myth: Software requirements continually change, but change can be easily accommodated because software is flexible.

Reality

- It is true that software requirements change, but the impact of change varies with the time at which it is introduced.
- Requirements changes are requested early (before design or code has been started), the cost impact is relatively small.

Practitioner's Myths

Myth

Myth: Once we write the program and get it to work, our job is done.

Reality

- Someone once said that "the sooner you begin 'writing code,' the longer it'll take you to get done."

Practitioner's Myths

Myth

Myth: Until I get the program "running" I have no way of assessing its quality.

Reality

- One of the most effective software quality assurance mechanisms can be applied from the inception of a project—the technical review.
- Technical review: Guessing the defects at an early stage and find it's solution. COde review is also done in this phase.

Practitioner's Myths

Myth

Myth: The only deliverable work product for a successful project is the working program.

Reality

A working program is only one part of a software configuration that includes many elements. A variety of work products (e.g., models, documents, plans) provide a foundation for successful engineering and, more important, guidance for software support.

General Issues

General Issues

There are three general issues that affect a software.

Heterogeneity

- Quality of being diverse in character or content.
- Systems are required to operate as distributed systems across networks that include types of computer and mobile devices.
- We often have to integrate new software with older legacy systems written in different programming languages.
- **Legacy System:** A legacy system is any outdated computing system, hardware or software that is still in use.
- **Challenge**: Develop techniques for building dependable software that is flexible enough to cope with this heterogeneity.

General Issues

Business and social change

- Business and society are changing incredibly quickly as emerging economies develop and new technologies become available.
- They need to replace their old software systems with updated ones
- Problem with traditional software engineering techniques: time consuming and delivery of new systems often takes longer than planned.
- They need to evolve so that the time required for software to deliver value to its customers is reduced.

General Issues

Security and trust

- As software is intertwined with all aspects of our lives, it is essential that we can trust that software.
- We have to make sure that malicious users cannot attack our software and that information security is maintained.

Software engineering ethics

Software engineering ethics

Confidentiality

Respect the confidentiality of your employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.

Competence

- You should not misrepresent your level of competence.
- You should not knowingly accept work that is outside your competence.

Software engineering ethics

Intellectual property rights

- You should be aware of local laws governing the use of intellectual property such as patents and copyright.
- You should be careful to ensure that the intellectual property of employers and clients is protected.
- **NB: Intellectual property rights are the rights given to persons over the creations of their minds**

Computer misuse

- One should not use your 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 or other malware).


Thank
YOU!