Software Engineering and Management Overview

Buyang Cao
School of Software Engineering
Tongji University
Cell: 15000215319
E-mail: caobuyang@tongji.edu.cn

What is the software engineering?

the study and an application of **engineering** to the **design**, **development** and **maintenance** of software

-- Wikipedia

Software engineering

- The economies of ALL 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 GNP (Gross National Product) in all developed countries.

Software costs

- Software costs often dominate computer system costs. The costs of software are at times greater than the hardware cost.
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs could be several times development costs.
- Software engineering is concerned with cost-effective software development.

Software Products

- Generic products
  - Stand-alone systems that are marketed and sold to any customer who needs them.
  - Examples – Desktop software such as graphics programs, project management tools; CAD software; software for specific markets such as financial assistant products (tax, accounting, etc.).
- Customized products
  - Software that is commissioned by a specific customer to meet their own needs.
  - Examples – embedded control systems, air traffic control software, traffic monitoring systems, ERP systems.

Software Specifications

- Generic products
  - The specification of what the software should do is owned by the software developer
  - The decisions on software change are made by the developer.
- Customized products
  - The specification of what the software should do is owned by the customer for the software
  - The customer makes decisions on software changes upon needs.

Frequently asked questions about software engineering

Question	Answer
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.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and 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 more general process.

Frequently asked questions about software engineering

Question	Answer
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.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

Essential attributes of good software

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

Software Engineering

- ***Software engineering***: an engineering discipline concerning all aspects of software production from the early stages of system specification to maintaining the system after deployment.
- Engineering discipline
  - Using appropriate theories and methods to solve problems considering organizational and financial constraints.
- All aspects of software production
  - Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.

Importance of Software Engineering

- Individuals and society rely more on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.
- It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as it were a personal programming project.
- For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

Software process activities

- Software specification, where customers and engineers define the software that is to be produced and the constraints on its operation.
- Software development, where the software is designed and programmed.
- Software validation, where the software is checked to ensure that it is what the customer requires.
- Software evolution/upgrade, where the software is modified to reflect changing customer and market requirements.

General issues affecting most software

- Heterogeneity
  - Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.
- 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.
- Security and trust
  - As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

Software engineering diversity

- There are many different types of software system and there is no universal set of software techniques that is applicable to all of these.
- 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.

Application types

- Stand-alone applications
  - Application systems running 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 are accessed by users from their own PCs or terminals. These include web or cloud-based applications such as e-commerce applications.
- Embedded control systems
  - Software control systems control and manage hardware devices. Numerically, there are probably more embedded systems than any other type of system (on-board system of a car, etc.).

Application types

- Batch processing systems
  - These are business systems that are designed to process data in large batches. They process large numbers of individual inputs to create corresponding outputs.
- Entertainment systems
  - These are systems that are primarily for personal use (game, for instance) and which are intended to entertain the user.
- Systems for modelling and simulation
  - These are systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.

Application types

- Data collection systems
  - These are systems that collect data from their environment using a set of sensors 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.

Software engineering fundamentals

- Some applicable fundamental principles irrespective of the development techniques used:
  - Systems should be developed using a managed and understood development process. Different processes are used for different types of software.
  - Dependability and performance are important for all types of system.
  - Understanding and managing the software specification and requirements (what the software should do) are important.
  - Whenever possible, software that has already been developed should be reused rather than writing new software.

Software engineering and the web

- The Web is a platform for running applications and more organizations are developing web-based systems rather than local systems.
- Web services (RESTFul API etc.) allow application functionality to be accessed over the web.
- Cloud computing is an approach to the provision of computer services where applications run remotely on the ‘cloud’.
  - Users do not buy software buy pay according to use.

Web software engineering

- Software reuse is the dominant approach for constructing web-based systems.
  - When building these systems, thinking about how to assemble them from pre-existing software components and systems.
- Web-based systems should be developed and delivered incrementally.
  - It is impractical to specify all the requirements for such systems in advance.
- User interfaces are constrained by the capabilities of web browsers.

Web-based software engineering

- Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.
- The fundamental ideas of software engineering apply to web-based software in the same way that they apply to other types of software system.

Key points

- Software engineering is an engineering discipline that is concerned with all aspects of software production.
- Essential software product attributes are *maintainability*, *dependability* and *security*, *efficiency* and *acceptability*.
- The high-level activities of specification, development, validation and evolution are part of all software processes.
- The fundamental notions of software engineering are universally applicable to all types of system development.

Key points

- There are many different types of system and each requires appropriate software engineering tools and techniques for their development.
- The fundamental ideas of software engineering are applicable to all types of software system.


HELP
WANTED

Software Engineering Project Management

the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled.

- **Purpose: to produce software product within budget, as specified, on time, and with good quality**

Project characteristics

- **Temporary**
  - Having definite beginning and definite end
  - Project team seldom outliving the project
- **Unique products, service, or results**
  - Product: an end item or a component item
  - Service: business functions supporting production or distribution
  - Result: outcomes or documents
- **Progressive elaboration**

Project management knowledge area

- Project Scope Management
  - Collect requirements
  - Define scope
  - Create WBS
  - Verify scope
  - Control scope

Project management knowledge area

- Project Time Management and Critical Path Method
  - Define activities
  - Sequence activities
  - Critical Path Method
  - Topological Sorting
  - Estimate activity resources
  - Estimate activity durations
  - Develop schedule
  - Control schedule

Project management knowledge area

- Project Cost Management
  - Estimate costs
  - Determine budget
  - control costs

Project management knowledge area

- Project Human Resource Management
  - Develop human resource plan
  - Acquire project team
  - Develop project team
  - Manage project team

Project management knowledge area

- Project Communications Management
  - Identify Stakeholders
  - Plan communications
  - Distribute Information
  - Manage Stakeholder expectations
  - Report performance

Project management knowledge area

- Project Risk Management
  - Plan risk management
  - Identify risks
  - Risk analysis
  - Plan risk responses
  - Monitor and control risks

Project management knowledge area

- Project Quality Management
  - Plan quality
  - Perform quality assurance
  - Perform quality control

Project management knowledge area

- Project Procurement Management
  - Plan procurements
  - Conduct procurements
  - Administer procurements
  - Close procurements

Course Case Studies

Case study 1

- Logistics application system
  - Project description

Case study 2

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 centres.

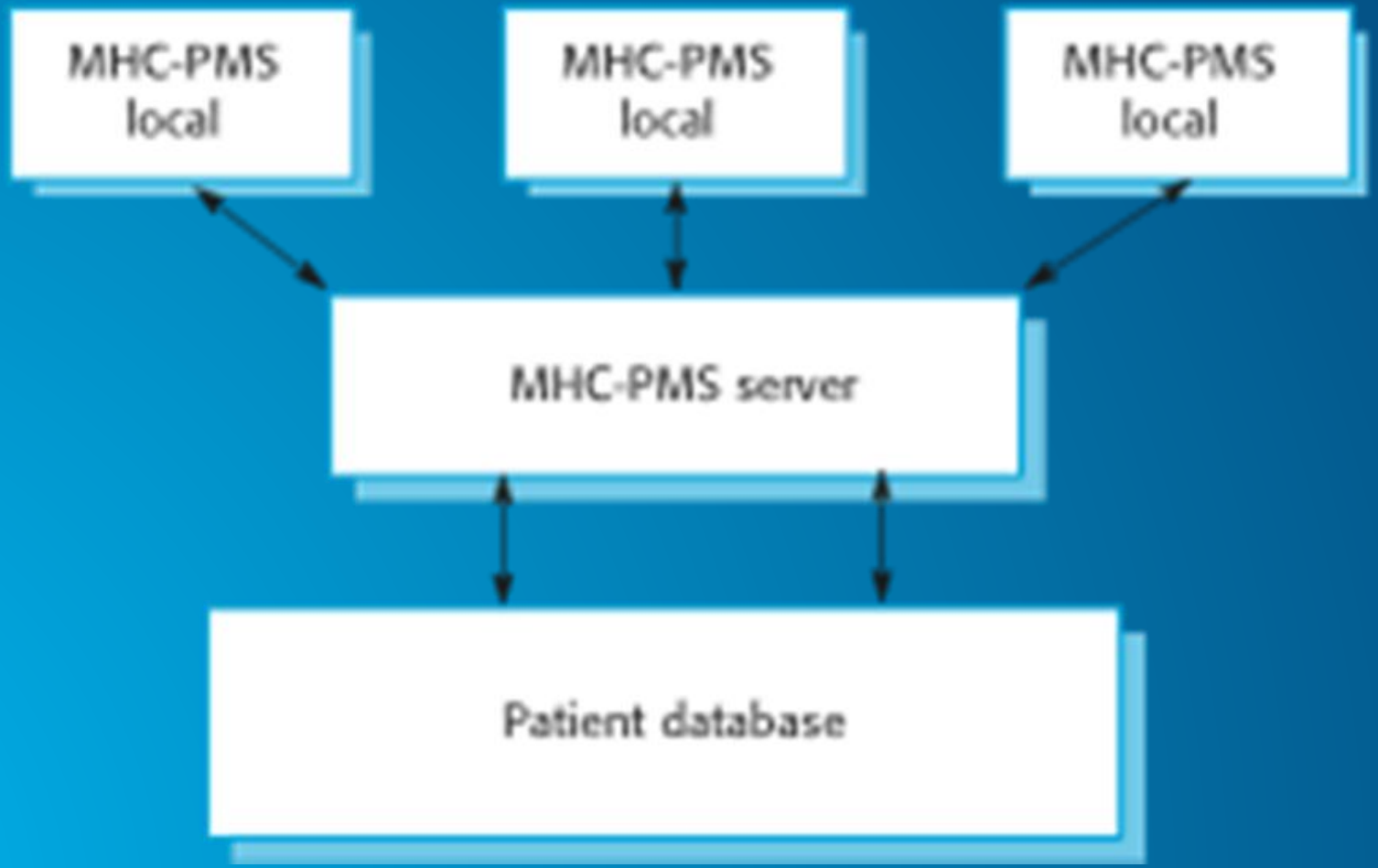
MHC-PMS

- The MHC-PMS (Mental Health Care-Patient Management System) 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.

MHC-PMS 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 MHC-PMS



MHC-PMS key features

- 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.

MHC-PMS concerns

- Privacy
  - It is essential that patient information is confidential and is never disclosed to anyone apart from authorised 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 whenever needed otherwise safety may be compromised and it may be impossible to prescribe the correct medication to patients.