

Software engineering



Ohjelmankehityspr., versionhallinta ja testaus – Chapter 4



KAJAANIN
AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Deepak K.C. ; deepak.kc@kamk.fi

Software Engineering

➤ Software:

- Collection of executable programming codes & Associated libraries & documentation
- Serves some specific requirements
- Economies of all developed nations – dependent on software

➤ Software Engineering

- Includes all aspects of software development & maintenance

IEEE Definition of Software Engineering:

1. The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software, that is, the application of engineering to software.
2. The study of the approaches as in (1).

Software engineering

- 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
- Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints
- All aspects of software production: technical development, project management, support, documentation, testing, methods

Software costs



- Costs of software on a PC often greater than the hardware cost
- Software costs more for maintenance than development
- Systems with longer life will have a maintenance costs which is several times higher than the development costs
- Software engineering concerned with cost-effective software development

Software disasters

- Large number of software projects unsuccessful
- Large percentage never completed
- Software project failure
 - Loss of money as well as time
 - Threaten the very existence of the company
- On average, large IT projects run 45% over budget and 7% over time while delivering 56% less value than predicted

Number source: http://calleam.com/WTPF/?page_id=1445

Examples: Failed software projects

Started	Terminated	Name	Cost (Expected)	Status
2001	2011	National programme for IT	£12bn (£2.3bn)	Discontinued
2009	2013	SIREN (police force)	£14.8m	Scrapped
2012	2014	Health care exchange website	\$200m	Cancelled Client & developer both sued each other
2013	---	Welfare payment system	£12.8bn (£2.2bn)	Schedule slipped

Exercise

- Think of some failed (unsuccessful) projects that you have been involved and list reasons for failure.

Undesirable outcomes of software project



- Over budget
- Exceeds schedule and/or misses market window
- Doesn't meet customer requirements
- Performance not meeting expectations
- Too difficult to use
- Lower quality than expected

Reasons for failure

Unrealistic project goals

Poor project management

Inaccurate estimates of needed resources

Poor reporting of the project's status

Unmanaged risks

Poor communication among customers, developers and users

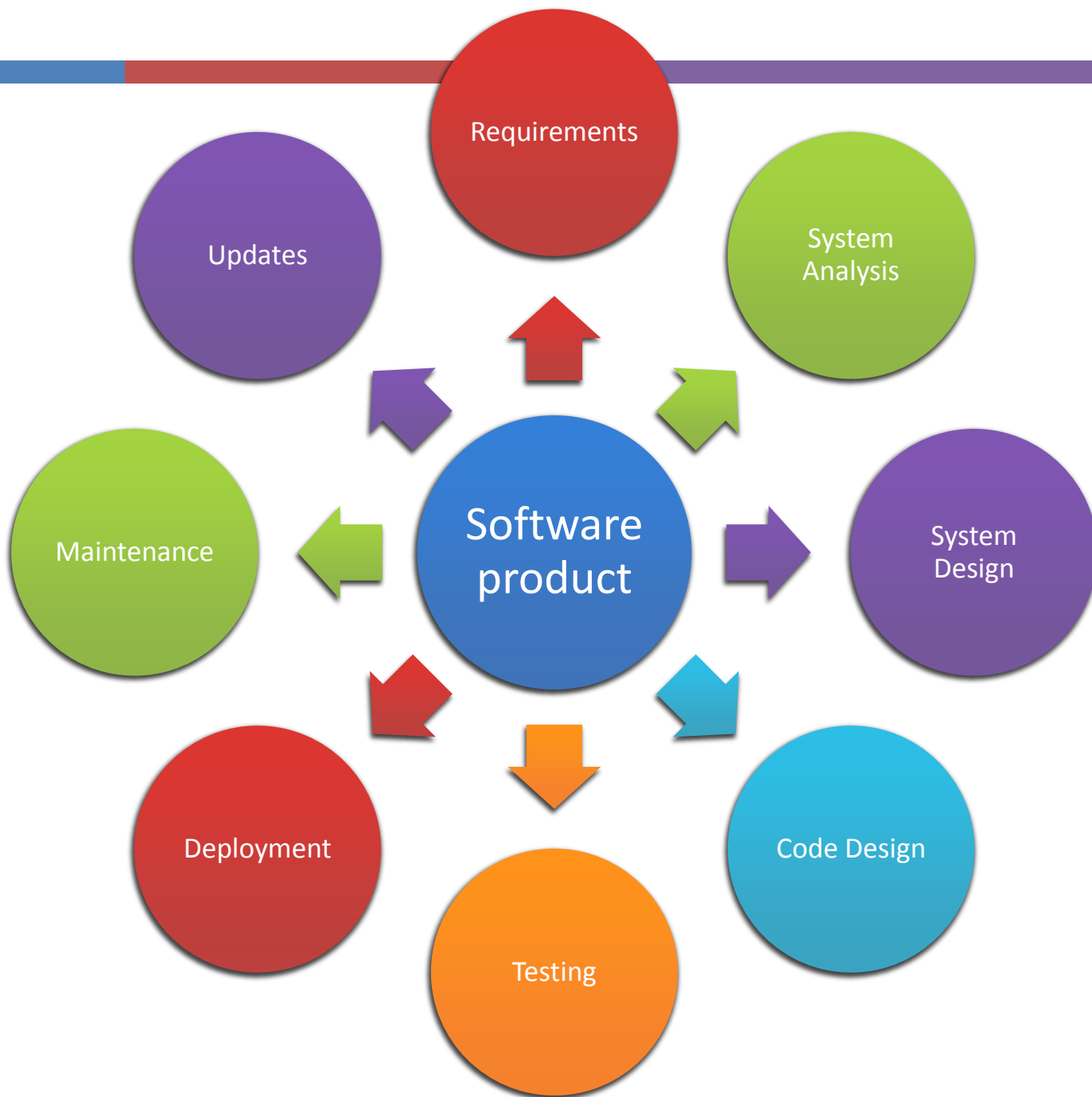
Project's complexity

Poor software design methodology

Poor testing methodology

Inadequate test coverage

Inappropriate software process



Software product

Generic products

- Stand-alone systems
 - Marketed & sold to any customer
- Examples
 - Graphics programs
 - Project management tools

Customized products

- Commissioned by a specific customer to meet their own needs
- Examples
 - Embedded control systems
 - Air traffic control
 - Traffic monitoring systems

Product specification

Generic products

- Owned by the software developer
- Decisions on software changes made by the developer

Customized products

- Customer owns the product
- Customer make decisions on required software changes

FAQs – 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.

FAQs – 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.

Importance of software engineering

- To produce reliable and trustworthy systems economically and quickly
- Better to use software engineering methods and techniques rather than just writing programs
- Software development is not a personal programming project
- Majority of the costs – changing the software once it has gone into use

Software process activities

```
graph TD; A[Software specification  
Software definition & operation] --> B[Software development  
Designed & programmed]; B --> C[Software validation  
Checked to ensure customer requirements are met]; C --> D[Software evolution  
Software modification to reflect changing customer & market requirements];
```

Software specification

Software definition & operation

Software development

Designed & programmed

Software validation

Checked to ensure customer requirements are met

Software evolution

Software modification to reflect changing customer & market requirements

General issues affecting most software

➤ Heterogeneity

- To operate as distributed systems across networks that include different types of computer & mobile devices

➤ Business & social change

- Society & business change quickly
- New economies develop
- To be able to cope with the changes and rapidly develop new software

➤ Security & trust

- Software intertwined with all aspects of our lives
- Essential that we can trust software

Exercise

You have planned to develop a game or any application. List general issues such as heterogeneity, security and trust and rapid technological changes that can affect your game development process.

Diversity

- Many different types of software system
- No universal set of software techniques that is applicable to all of these
- Methods and tools used depend
 - on the type of application being developed
 - Requirements of the customer &
 - The background of the development team

Application types

➤ Stand alone applications

- Systems running on a local computer
- Include all necessary functionality
- Do not need to be connected to a network

➤ Interactive transaction-based applications

- Execute on a remote computer
- Accessed by users from their own PCs or terminals
- Eg: Web application such as e-commerce applications

➤ Embedded control systems

- Systems that control and manage hardware devices
- Numerically more embedded systems than any other type of system

➤ Batch processing systems

- Business systems designed to process data in large batches
- Process large number of inputs to create corresponding outputs

Application types

- Entertainment systems
 - Primarily for personal use
 - Intended to entertain the user
- Systems for modeling & simulation
 - To model physical processes or situations
 - Include many, separate, interacting objects
- Data collection systems
 - Systems that collect data from the environment using a set of sensors
 - Send data to other systems for processing
- Systems of systems
 - Systems that are composed of a number of other software systems

Case Study

A mental health case patient management system

- A system used to maintain records of people receiving care for mental health problems.

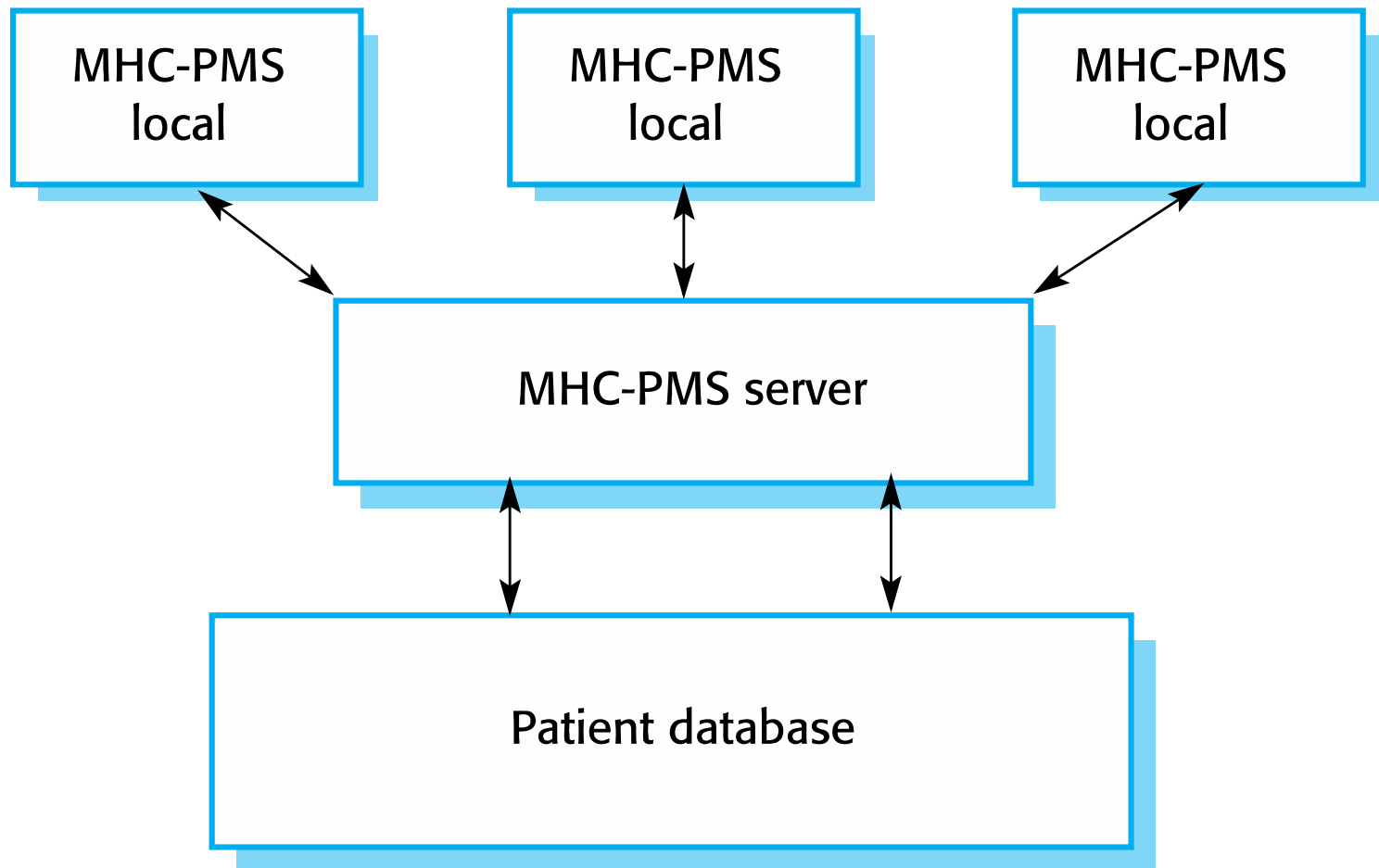
A patient information system for mental health care

- A medical information system that maintains information about patients suffering from mental health problems and the treatments that they have received.
- An information system that is intended for use in clinics
- It makes use of a centralized database of patient information
- 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

Organization of the MHC-PMS



Key features

- Individual care management
 - create records for patients, edit the information in the system, view patient history
- Patient monitoring
 - Monitors the record of patients 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

MHC-PMS concerns

➤ Privacy

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

➤ Safety

- system must be available when needed otherwise safety may be compromised
- Warning about potential dangerous concerning patients

Software engineering process



- Define the problem
- Describe the solution
- Develop the solution
- Maintenance



How the customer explained it



How the Project Leader understood it



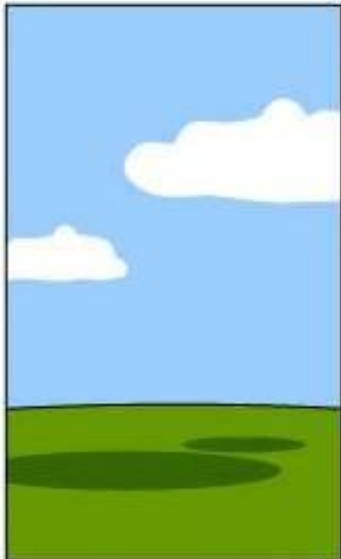
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



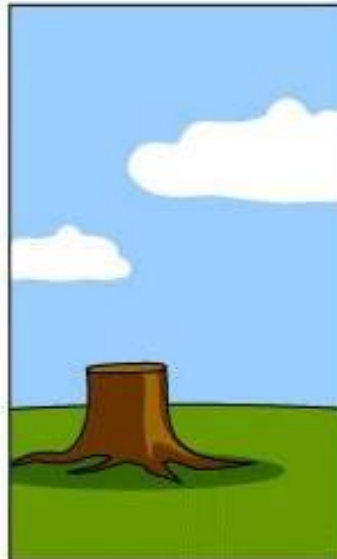
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

References

➤ Software Engineering, 9th Edition by Ian Sommerville