

Introduction to Software Engineering

Hao Wu

haowu@nuim.ie

Software is Everywhere



Business

Education

Military

Scientific

and more

“Software engineering is essential for the functioning of national and international societies” (Sommerville)

Software In the past

A program is a sequence of instructions (machine codes) to get the computer to do something useful. Program was written by mathematicians (e.g. solve differential equation)

Single user operating system, high level programming languages and compiler: isolating the programmer /user from the machine internals

Rise of Programmer profession (User had to specify the requirements, the programmer develops programs)

Beginning of Software Industry

- Expanding automated information-processing tasks in companies
- Importance of programming to companies activities is increased
- Tools support to increase software programmer's productivity
- The introduction of the personal computer and its widespread adoption after 1980 accelerated the demand for software and programming

What is Software ?

- Not just about computer programs !
- It is also associated with documentation and configuration data that is needed to make these programs operate correctly
- Software products may be developed for a particular customer or may be developed for a general market

Diversity of Software

- System software (operating system: Ms Windows, etc)
- Application software (MS.Office, ERP, etc.)
- Engineering/scientific software (AUTOCAD, DNA Analysis)
- Embedded software (software in mobile phones, washing machine, etc.)

Nature of Software/System

- Abstract and intangible
- Lack of physical constraints
 - No natural limits to the potential of software
- Easily becomes extremely complex
- Effort intensive (organize carefully)

Nature of Software/Software System

- Changing needs/requirements for users
- High risk of failure
 - user acceptance – may reject it
 - performance
 - maintainability

Adhoc Approach

- No planning of development work (no milestones defined)
- No control or review
 - Committing a lot of cost and time
- Deliverables to user not clearly defined
- Poor understanding of cost and effort by both developer and user

Engineering Approach

- Common large projects and successfully done
 - Building bridges, dams
 - Aircraft, missile
- Need a reliable, robust and maintainable product
- Engineering is a solution
 - To design, develop an artifact that meet specification efficiently, cost effectively and ensuring quality

Engineering Approach (contd.)

- Using scientific principles
 - Require well-defined approach: repeatable and predictable
-
- Thus, the intent of software engineering is to provide a framework for building software with higher quality

What is Software Engineering ?

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

Software Engineering

- *An Engineering discipline:*
 - Engineers make things work. They apply theories, methods and tools where they are appropriate – but they use them selectively
 - Also engineers recognize that they must work to organizational and financial constraints, so they look for solutions within these constraints.

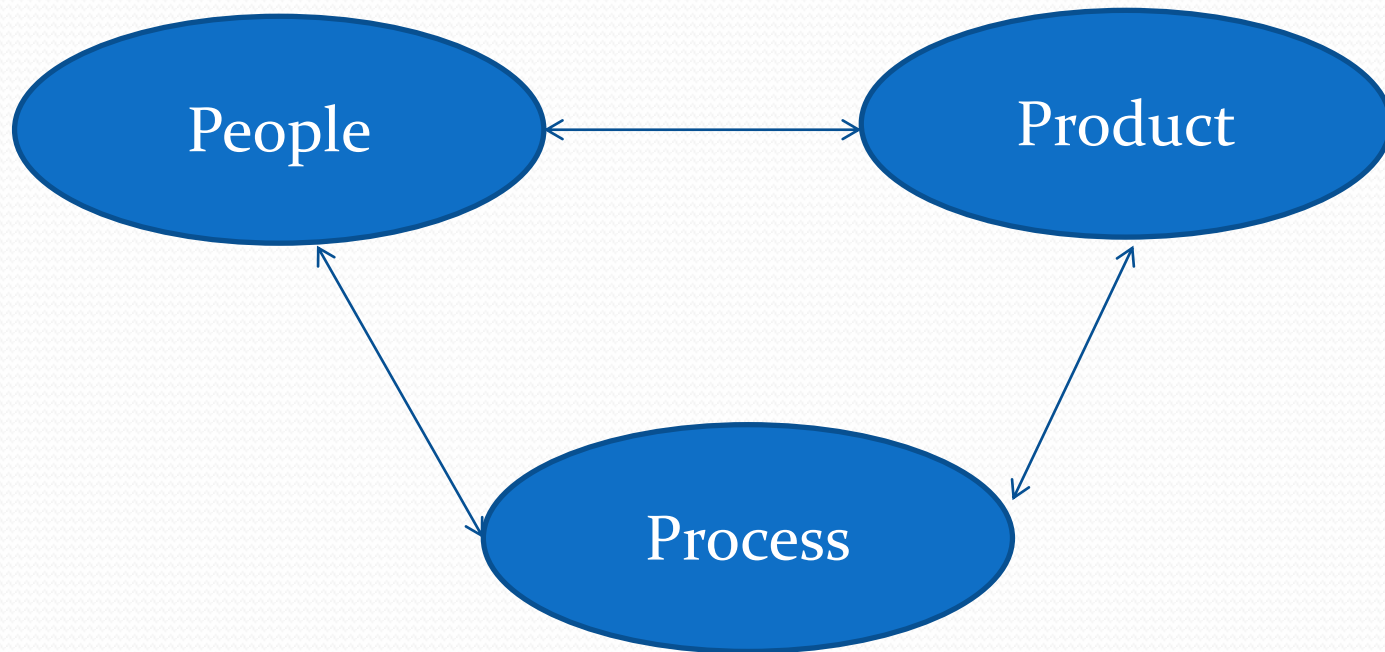
Software Engineering

- *It Covers all aspects of software production:*
 - Software Engineering is not just concerned with the technical processes of software development
 - It is also concerned with activities such as software project management and with the development of tools, methods and theories to support software production

What difference has the web made to Software Engineering ?

- Web services come into existence. Web services are software components that deliver specific, useful functionality/service over the web (e.g. meteor web service, route navigation, etc.).
- The Web has led to the availability of software services and the possibility of developing highly distributed service-based system (software reuse).

3Ps View on Software Engineering



People View

- People use the system
- People design the system
- People build the system
- People maintain the system
- People pay for the system!

Product/Software product

- Generic products/ Commercial Off-The-Shelf (COTS)
 - Stand-alone systems which are produced by a development organization and sold on the open market to any customer
- Bespoke (customized) products
 - Systems which are commissioned by a specific customer and developed specially by some contractors

Product/software product

- Most software expenditure is on generic products but most development effort is on bespoke systems
- The trend is towards the development of bespoke systems by integrating generic components (which must themselves be interoperable)

Process/Software process

- Structured set of activities required to production of a software product.
- Common fundamental activities to all software processes:
 - Software Specification
 - Software Development
 - Software Validation
 - Software Evolution

Process/Software process (contd.)

- Software specification
 - Customers and engineers define the software to be produced and its constraints
- Software development
 - Software is designed and programmed
- Software Validation
 - The software is tested and validated if it is what the customer requires
- Software Evolution
 - Software is updated to customer needs and market

Software Design Document examples

- Example 1:

<https://www.oasis-open.org/committees/download.php/24846/Example-SoftwareDesignDocument-LegalXMLUtility.pdf>

- Example 2:

https://cise.ufl.edu/class/cen3031fa11/documents/examples/SDD_Example_2_2011.pdf

- Example 3:

https://cise.ufl.edu/class/cen3031fa11/documents/examples/SDD_Example_1_2011.pdf

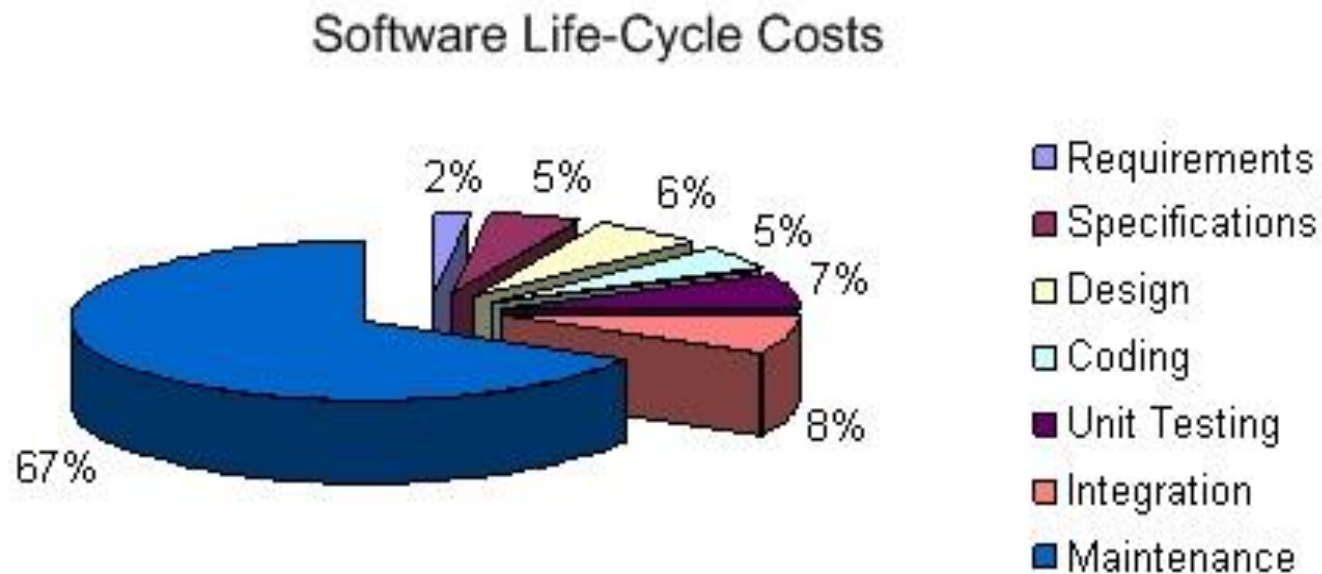
Process/Software process (contd.)

- Different types of systems need different development process
- Activities vary depending on the organization and the type of system being developed
- Example: E-commerce system and embedded system for medical device control need different types of software process

Why is Software engineering important ?

- We need reliable , robust and maintainable software systems but saving cost
- For most types of system the cost of changing the software after it has gone into use is the majority of costs.

Software Life-Cycle Cost



Ref. Schach 2002 p.12

© 2003-2005 by Digital Aggregates Corp. All rights reserved.

Successful Software System

- Software development projects have not always been successful
- Success rate a lot lower than other engineering disciplines
- When do we consider it successful?
 - Development completed within time and budget constraint
 - (sounds obvious but many abandoned)
 - It is useful
 - It is usable, and
 - It is used
 - (the users would find it attractive and keen to use it)
 - Cost-effective
 - Maintainable

Essential characteristic of good software

- Maintainability
 - Software can evolve to meet the changing needs of customers
- Dependability and security
 - Dependable software should not cause physical or economic damage in the event of system failures. Malicious users should not be able to access or damage the system.

Essential characteristic of good software

- Efficiency
 - Software should not make wasteful use of system resources: processing time, memory utilisation, responsiveness, etc.
- Acceptability
 - Software must be acceptable to the type of users for which it is designed: usable, understandable and compatible with other systems that they use.

Software Failures

- Does not solve user's problem
- Schedule slippage
 - Not ready in time
- Cost over-runs
 - becoming too costly to complete
- Poor quality
- Poor maintainability

Software Failures Example

- **Mars Climate Orbiter doesn't orbit (1998)**
- **Cost:** \$327.6 million



- **Disaster:** After a 286-day journey from Earth, the Mars Climate Orbiter fired its engines to push into orbit around Mars. The engines fired, but the spacecraft fell too far into the planet's atmosphere, causing it to crash on Mars.
- **Cause:** Approached Mars at the wrong angle. The software that controlled the Orbiter thrusters used imperial units for force (pound-seconds), rather than metric units (Newton-seconds, defined in the Software Interface Specification (SIS)) for thrust instructions as specified by NASA and used in the software generating the instructions on the ground – *a metric mixup*

Software failures

- Recent Gartner survey (June 2012):
 - Runaway budget costs are behind one-quarter of project failures for projects with budgets greater than \$350,000.
 - Small projects are easier to manage and execute. The failure rate of large IT projects with budgets exceeding \$1 million was found to be almost 50% higher than for projects with budgets below \$350,000.
- KPMG survey of Project Management practices of 100 business in New Zealand, Dec 2010
 - An incredible 70% of organizations have suffered at least one project failure in the prior 12 months, and 50% of respondent indicated that their project failed to consistently achieve what they set out to achieve.
- IBM study
 - Only 40% of projects meet schedule, budget and quality goals. Biggest barriers to success are people factors.

Chaos Report -2015

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

- Study 50,000 IT projects around the world
- Success factors: deliver on time, on budget with a satisfactory result.
- Challenged projects: delivered but overrun projects

Chaos Report by size

CHAOS RESOLUTION BY PROJECT SIZE

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

Software failures thoughts

- The causes (Sommerville):
 - Complexity:
 - complex application domain
 - domain expert vs. software developer: different background and vocabularies
 - Difficulty of mentally grasping the details of large and complex projects
 - Low expectation: not really use software engineering methods in their everyday work in companies

Summary points

- Software development needs an engineering approach to get software with higher quality
- Software engineering is an engineering discipline that is concerned with all aspects of software production
- Software process includes all activities in software development
- Different types of software need different software process.