**Khulna University**
Computer Science & Engineering Discipline


Copyright © 2015, SK Alamgir Hossain


SK Alamgir Hossain

Assistant Professor
Computer Science & Engineering Discipline
Khulna University, Khulna, Bangladesh
<http://alamgirhossain.com>

CSE3203 - Software Engineering and Information System Design

S SOFTWARE ENGINEERING

**Khulna University**
Computer Science & Engineering Discipline



"I choose a lazy person to do a hard job. Because a lazy person will find an easy way to do it."

Bill Gates

Chapter 1 - Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 2

Chapter 1

Introduction to Software Engineering

Lecture 1

Introduction to Software Engineering

- What is Software?
- Hardware vs. Software
- Software characteristics
- Types of software
- Software Myths

What is Software?



Software is a product

- Transforms information - produces, manages, acquires, modifies, displays, or transmits information
- Delivers computing potential of hardware and networks

Software is a vehicle for delivering a product

- Controls other programs (*operating system*)
- Effects communications (*networking software*)
- Helps build other software (*software tools & environments*)

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 5

What is Software ?



Software can define as:

- Instruction – executed provide desire features, function & performance.
- Data structure – to adequately manipulate operation.
- Documents – operation and use of the program.

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.

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 6

Hardware vs. Software



Hardware

- Manufactured
- Wear out
- Built using components
- Relatively simple

Software

- Developed/ engineered
- Deteriorate
- Custom built
- Complex

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 7

Manufacturing vs. Development



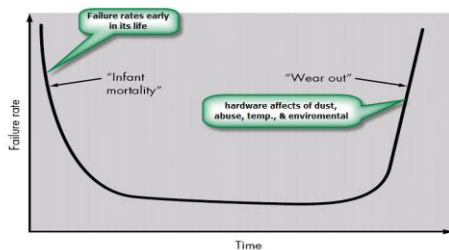
- Once a hardware product has been manufactured, it is difficult or impossible to modify. In contrast, software products are routinely modified and upgraded.
- In hardware, hiring more people allows you to accomplish more work, but the same does not necessarily hold true in software engineering.
- Unlike hardware, software costs are concentrated in design rather than production.

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 8

Failure curve for Hardware

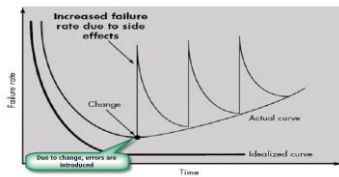


Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 9

Failure curve for Software



- When a hardware component wears out, it is replaced by a spare part.
- There are no software spare parts. Every software failure indicates an error in design or in the process through which design was translated into machine executable code.
- Therefore, software maintenance involves considerably more complexity

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 10

Component Based vs. Custom Built



- Hardware products typically employ many standardized design components.
- Most software continues to be custom built.
- The software industry does seem to be moving (slowly) toward component-based construction.

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 11

So What are the Software characteristics?



- Software is developed or engineered; it is not manufactured.
- Software does not "wear out" but it does deteriorate.
- Software continues to be custom built, as industry is moving toward component based construction.

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 12

Types of Software



- System software
- Application software
- Engineering/scientific software
- Embedded software
- Product line software
- Web applications
- Artificial intelligence software

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 13

Legacy Software



- Older programs
- The quality of Legacy software is poor
- Sometimes has inextensible design, convoluted code, poor or non-existent documentation, test cases and results that were never achieved, a poorly managed change history...
 - ✓ The software must be adapted
 - ✓ must be enhanced
 - ✓ must be extended
 - ✓ must be re-architected

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 14

Software Myths



Definition: Beliefs about software and the process used to build it.

- Management myths
- Customer Myths
- Practitioner's myths

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 15

Management myths



Managers in most disciplines, are often under pressure to maintain budgets, keep schedules on time, and improve quality.

Myth1: 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:

- Are software practitioners aware of existence standards?
- Does it reflect modern software engineering practices?
- Is it complete? Is it streamlined to improve time to delivery while still maintaining a focus on quality?

Myth2: If we get behind schedule, we can add more programmers and catch up

Reality: Software development is not a mechanistic process like manufacturing. Adding people to a late software project makes it later. People can be added but only in a planned and well-coordinated manner

Myth3: If I decide to outsource the software project to a third party, I can just relax and let that firm build it.

Reality: If an organization does not understand how to manage and control software projects internally, it will invariably struggle when it outsource software projects

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 16

Customer Myths



Customer may be a person from inside or outside the company that has requested software under contract.

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

Reality: A poor up-front definition is the major cause of failed software efforts. A formal and detailed description of the information domain, function, behavior, performance, interfaces, design constraints, and validation criteria is essential. These characteristics can be determined only after thorough communication between customer and developer.

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

Reality: Customer can review requirements and recommend modifications with relatively little impact on cost. When changes are requested during software design, the cost impact grows rapidly.

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 17

Practitioner's myths



Myth1: Once we write the program and get it to work, our job is done.

Reality: 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.

Myth2: 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 formal technical review.

Myth3: 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. Documentation provides a foundation for successful engineering and, more important, guidance for software support.

Myth4: 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 quality. Better quality leads to reduced rework. And reduced rework results in faster delivery times.

Chapter 1- Introduction to Software Engineering






<http://alamgirhossain.com>

Slide: 18

Just Fun



Evolution of Software Engineer

UNIXC, C++JAVAJAVA .NetMS EXCEL :-(

1st Yr. 2nd Yr. 3rd Yr. Last Yr. Rest Of Life

Chapter 1- Introduction to Software Engineering

<http://alamgirhossain.com>

Slide: 19

THANK YOU

Chapter 1- Introduction to Software Engineering

20
