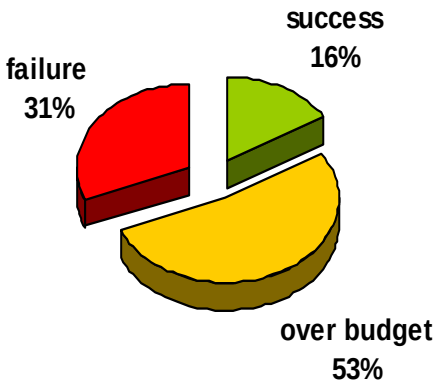
1.1-Why SE?

Without SE, software development or software project will get **31% failure**, **53% over budget**, **only 16% success**. (Extracted from World Statistic) shown as figure below



1.2-Concepts

1.2.1-What is Software (SW)?

-SW is computer programs and associated documentation such as requirements, design models and user manuals.

-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.
- **Bespoke (custom)** - developed for a single customer according to their specification.

-New software can be created by developing new programs, configuring generic software systems or reusing existing software.

1.2.2-What is Software Engineering (SE)?

-Software engineering is an engineering discipline that is concerned with all aspects of software production.

-Software engineers should adopt a systematic and organised approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available.

-Programming-in-the-large vs. programming-in-the-small!

-Managing/Controlling

- Quality
- Complexity
- Resources: Budget, Time, People
- Risks

-Repeatable

1.2.3- Software Quality

Quality, simplistically, means that a product should meet its specification. The attributes of good software (quality) are:

-Usability: Users can learn it fast and get their job done easily

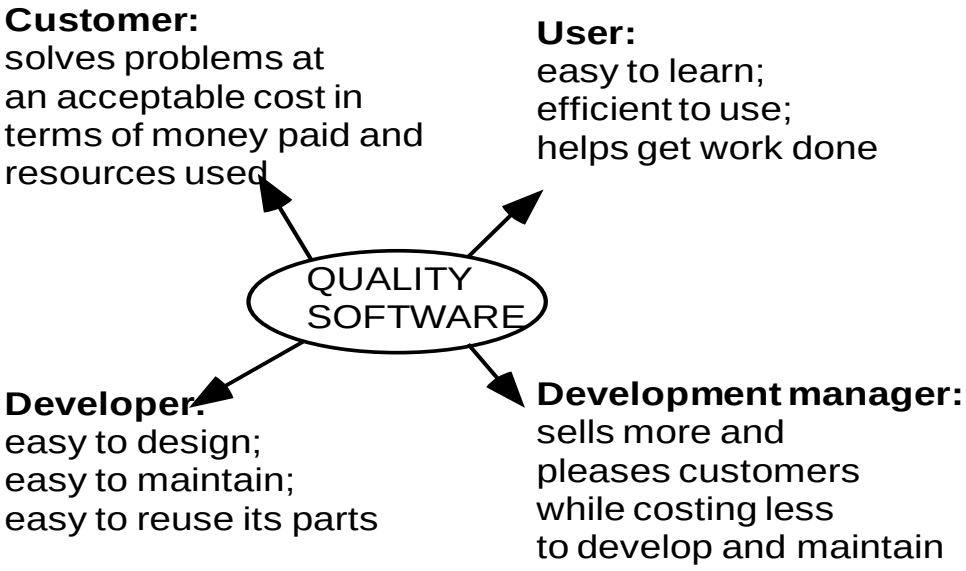
-Efficiency: It doesn't waste resources such as CPU time and memory

-Reliability: It does what it is required to do without failing

-Maintainability: It can be easily changed

-Reusability: Its parts can be used in other projects, so reprogramming is not needed

Good software quality diagram:



The different qualities can conflict

- Increasing efficiency can reduce maintainability or reusability
- Increasing usability can reduce efficiency

Setting objectives for quality is a key engineering activity

- You then design to meet the objectives
- Avoids 'over-engineering' which wastes money

Optimizing is also sometimes necessary

- **E.g. obtain the highest possible quality using a fixed budget**

Short Term Vs. Long Term Quality

-Short term:

- Does the software meet the customer's immediate needs?
- Is it sufficiently efficient for the volume of data we have today?

-Long term:

- Maintainability
- Customer's future needs

1.2.4- What is the difference between software engineering and computer science?

-Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

-Computer science theories are still insufficient to act as a complete underpinning for software engineering (unlike e.g. physics and electrical engineering).

1.2.5- 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 process concerned with developing the software infrastructure, control, applications and databases in the system.

-System engineers are involved in system specification, architectural design, integration and deployment.

1.2.6- What is a software process?

-A set of activities whose goal is the development or evolution of software.

-Generic activities in all software processes are:

- **Specification:** what the system should do and its development constraints
- **Design**: set of decisions on how the software system is going to meet its specification.
- **Development:** production of the software system
- **Validation:** checking that the software is what the customer wants
- **Evolution:** changing the software in response to changing demands.

1.2.7- The Characteristics of Software

-Software is developed or engineered. It is not manufactured.

-Software failure is usually distinct from hardware failure in that software does not wear out, but it does deteriorate.

-When a hardware component wares 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.

-Although the industry is moving toward component-based assembly, most software continues to be custom built.

1.3-Software Applications

It is somewhat difficult to develop meaningful generic categories for software application. The following software areas indicate the breadth of potential applications.

-System software:

A collection written to serve other programs. It is characterized by heavy interaction with computer hardware; heavy usage by multiple users. (e.g. operating system components, drivers, …).

-Real-time software:

Is software that monitors/analyzes/controls the real-world events. It, especially, can be used to collect and format data from the external environment.

-Business software:

Is business information processing that is the largest single software application area (e.g. payroll, accounts receivable/payable, inventory).

-Engineering and scientific software:

Engineering and scientific have been characterized by “number crunching” algorithms. Applications range from astronomy to volcano logy, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing.

-Embedded software:

Embedded software resides in read-only memory and can perform very limited and esoteric functions (e.g. keypad control for a microwave oven) or provides significant function and control capability (e.g. digital functions in an automobile such as fuel control, dashboard display, and braking systems).

-Personal computer software:

e.g. word processing, spreadsheet, computer graphics, multimedia, and so on.

- Web-based software:

e.g. HTML, Perl, Java, and so on. In essence, the network becomes a massive computer providing an almost unlimited software resource that can be accessed by anyone with a modem.

-Artificial intelligence software:

AI software makes use of nonnumeric algorithms to solve complex problems that are not amenable to computation or straightforward analysis. Game playing are representative of application within this category.