05506017
Software Engineering

Chapter 3 :
Software Quality

ผศ.ดร.วรางคณา กิ้มปาน

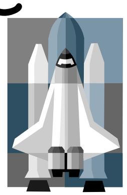
ภาควิชาวิทยาการคอมพิวเตอร์ สจล.

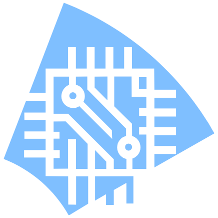
Introductions

In engineering activities is to build???

- Civil engineer => bridge
- Electrical engineer => circuit
- Aerospace engineer => airplane
- Software engineer ???

  It is a “Software system”





External & Internal of Software Qualities

External

- Visible to the users.
- Users care.

Internal

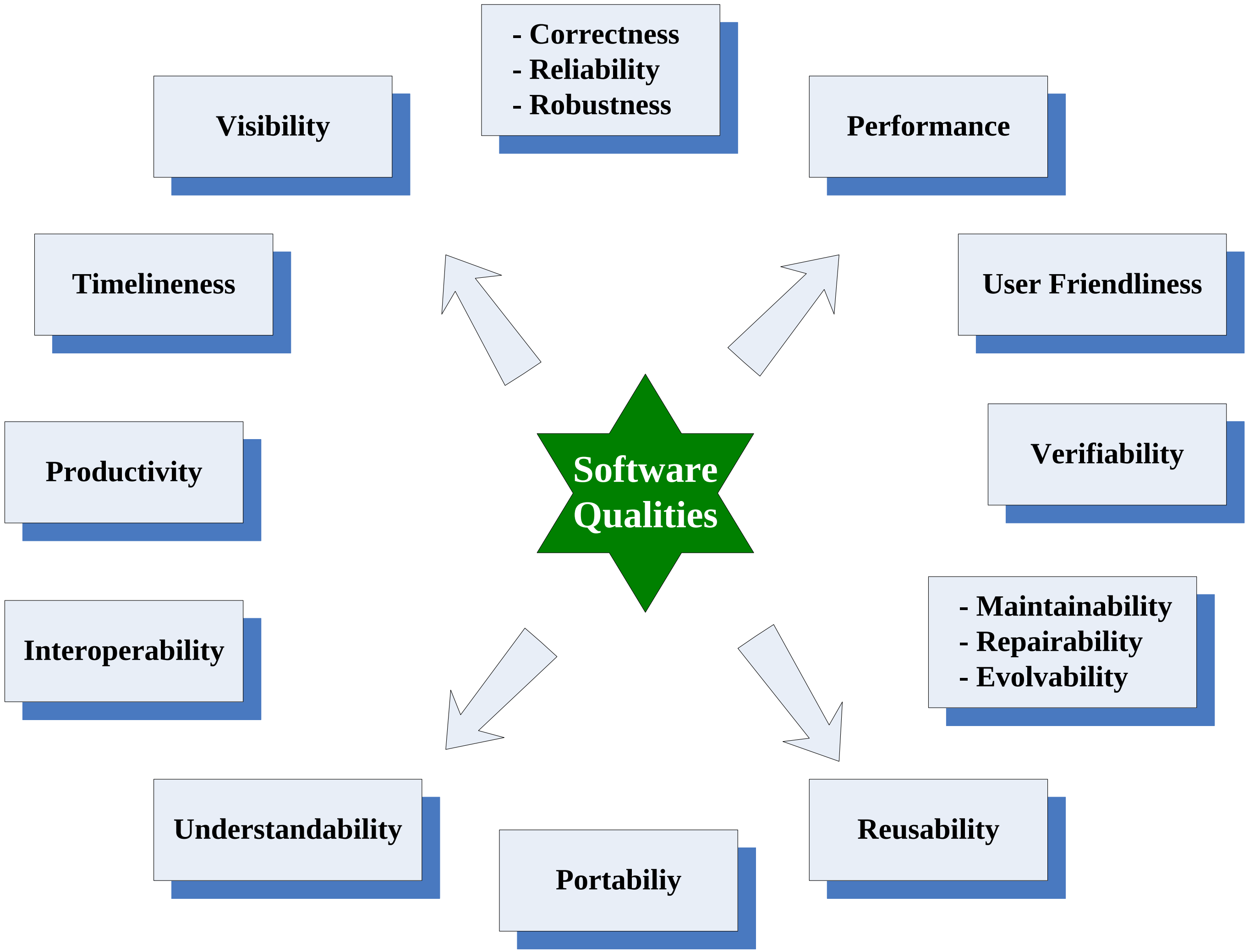
- Concern the developers of the system.
- Largely deal with the structure of SW, helps to achieve the external qualities.
- The internal quality of verifiability is necessary to achieve the external quality of reliability.

Product & Process Qualities

- Process is used to produce the SW product.
- If the process requires careful planning of system test data before designing.

- Process qualities are closely related to product qualities
- Product reliability will increase.
- SW product included object code, requirements, design, source code or test data.

Important Quality of Software Products



Software Qualities

- Correctness
- Reliability
- Robustness

Visibility

Performance

Timeliness

User Friendliness

Productivity

Verifiability

Interoperability

- Maintainability
- Repairability
- Evolvability

Understandability

Portabiliy

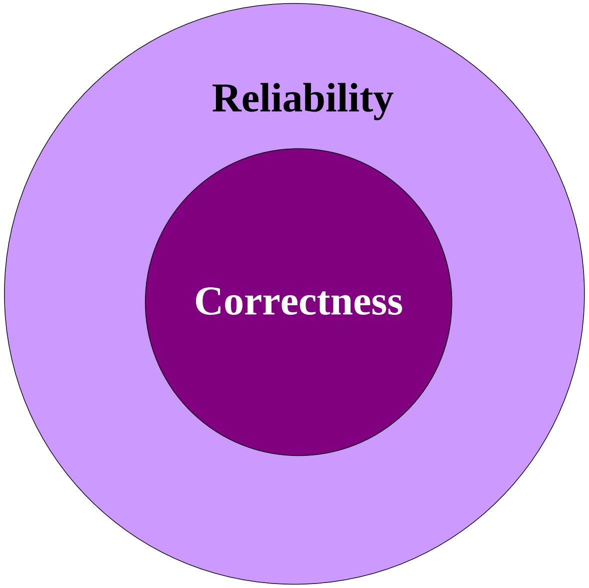
Reusability

Software Qualities

1. Correctness

- Is program functionally correct?
- Program should provide functional requirements specifications.
- Depends on how rigorous in specifying functional requirements.
- The possibility to determine unambiguously whether program meets the specifications or not

Software Qualities (2)



2. Reliability

- The probability that SW can operate as expected over a specified time interval.
- If the consequence of SW error is not serious, the incorrect SW may still be reliable.
- SE will be called as an engineering discipline only when we can achieve SW reliability compare with other products.

Software Qualities (3)

3. Robustness

- Robustness and correctness are strongly related.
- A system written to be used by the beginner computer's users must be more prepared to deal with ill-formatted input than the embedded system (From sensors).
- Can you imagine how Q1 to Q3 are related?

Software Qualities (4)

4. Performance

- Affects the usability of the system.
- System too slow => Reduces the productivity
- System too big => Hard to run
- System uses too much memory => Affects other running applications

To evaluate the performance;

- Analyzing the complexity of algorithms.

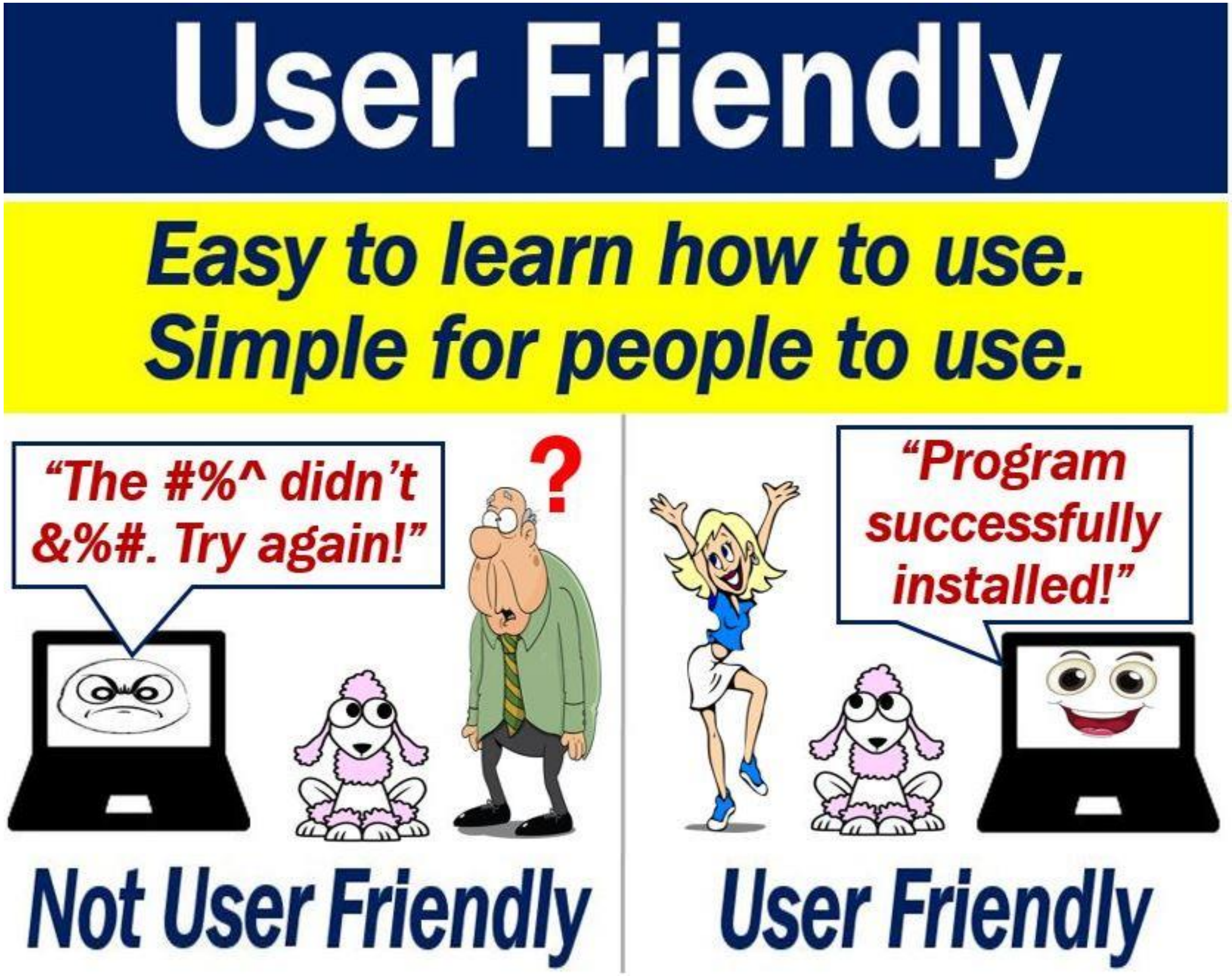
- There are 3 methods to evaluate; Measurement, Analysis, Simulation

Software Qualities (5)

5. User Friendliness

- It is easy to use.
- Programmer => Type command while non-programmer => the use of menus
- The user interface with mouse or window interface it better for the beginner than one-letter commands.
- Correctness and Performance also affect user friendliness.

Software Qualities (6)


User Friendly
Easy to learn how to use.
Simple for people to use.
"The #%^ didn't &%#. Try again!"
?
"Program successfully installed!"
Not User Friendly
User Friendly

Software Qualities (7)

6. Verifiability

- SW system should be verified easily.
- To improve verifiability is the use of SW monitors, code inserted in the SW to monitor the performance or correctness.
- Internal & External quality.

Software Qualities (8)

7. Maintainability

- Is indeed not the proper word for SW. We can use the word SW evolution.
- It can be divided into 3 types of maintenance;
  - Corrective : 20% of cost is to remove error when delivered or errors introduced into the SW during its maintenance.
  - Adaptive : Adjusting the SW to changes in the environment. (New OS, or DB system)
  - Perfective : Change SW to improve qualities. (Add new function, procedure or improve performance)

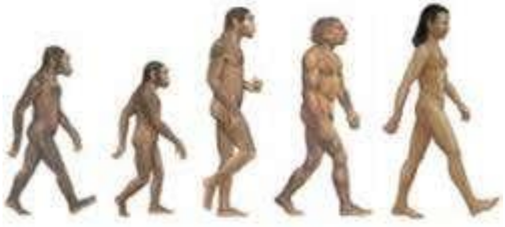
Software Qualities (9)

8. Repairability

- There is RAS (Repairability, Availability, Serviceability) in Computer Hardware Engineering.
- Using standard parts can reduce the cost of SW production.
- It is more easily to replace a single part than the whole body.
- A SW product with well designed modules is much easier to repair.
- It is better to choose right module structure with the right module interfaces to reduce the need for module interconnections.

Software Qualities (10)

9. Evolvability



- SW products will also be modified over time to provide new functions or change existing function.
- If the modification is not documented, it will be more difficult to make a future changes.
- The modifications should be applied with careful study of the original design with precise description of changes in both design and requirements specification.

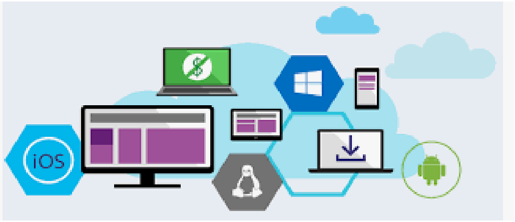
Software Qualities (11)

10. Reusability

- Modify product to build a new version or to build another product with minor changes.
- Reduce SW production costs.
- We can reuse parts of previous requirements specification instead of developing a new one.

11. Portability

○ Ability to run a system in the different environments or platform (OS).



Software Qualities (12)

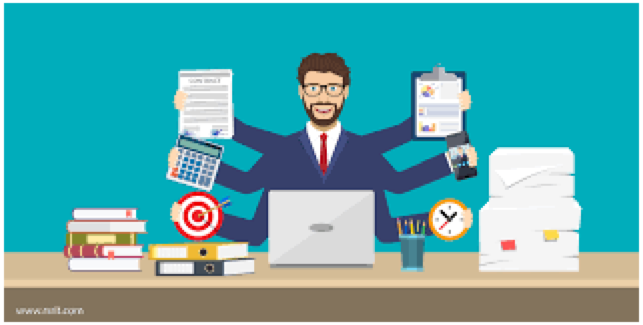
12. Understandability

- The design that is easy to understand.
- Internal product quality.
- It helps to achieve evolvability and verifiability.

13. Interoperability

- The ability of a system to coexist and cooperate with other systems.
- Many vendors produce integrated products that include several different functions.

Software Qualities (13)



14. Productivity

- Productivity is a performance quality applied to the process.
- SW reuse is a technique leads to overall productivity of organization involved with many developed products, but developing reusable modules is harder than for one's own use.

15. Timeliness

- A process that related to the quality refers to the ability to deliver a product on time.

Software Qualities (14)

16. Visibility

- SW development process is visible if…all steps and current status are documented clearly.
- It is both internal quality and external quality.
- Allows the SEng to weigh the impact of their actions and guides them in making decision.
- Product is visible if it is clearly structured with clearly understandable functions and easily accessible documentation.

Quality Requirements

Quality requirements in 4 different application areas are as the following;

- **Information Systems** (Information-Based Systems) (To manage information)
  - Banking systems, library-cataloguing systems, personnel systems.
  - The heart is database, apply transactions to create, retrieve, update or delete items.

Qualities;

□ Data integrity : will data be corrupted?

□ Security : does the system protect data from unauthorized access?

□ Data availability : what conditions will data become unavailable?

□ Transaction performance : the goal of information systems is to support transactions against the information.

Quality Requirements (2)

Quality Requirements (3)

- **Real-Time Systems**
  - □ The systems must respond to events within a predefined and strict time period. (Factory-monitoring system for increasing temp.)
  - □ Example, mouse-handling SW that needs to respond to mouse click interrupts.
  - □ The interface for real-time requirement on the SW.
  - □ Response that comes too early maybe as incorrect as response that comes too late. Such as if the first mouse click is processed too fast, the double-click may not be detected correctly.
  - □ Are characterized by how well they satisfy the response time requirements which is one of correctness criteria.

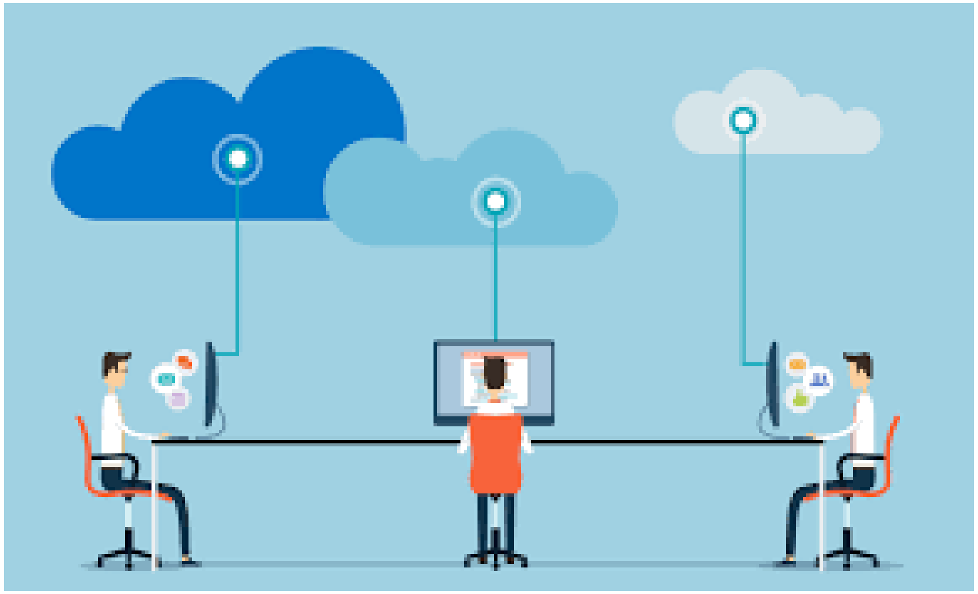
Quality Requirements (4)

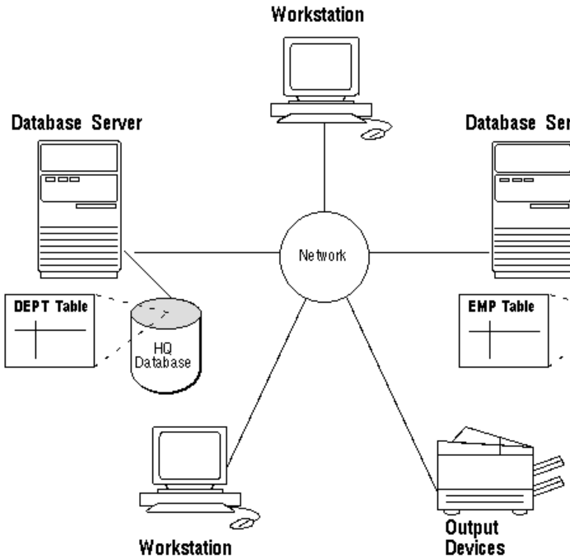
Quality Requirements (5)

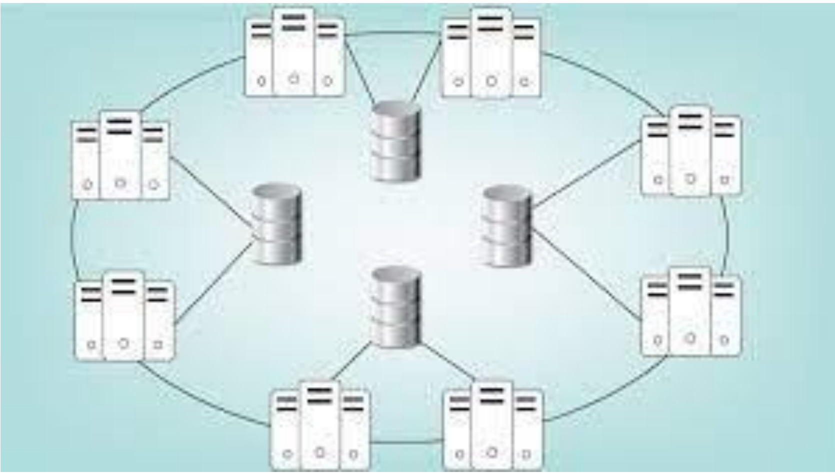
- **Distributed Systems**
  - □ The systems with components that run on different computers.
  - □ The characteristics of distributed systems are;
    1. The amount of distribution supported; data distributed?   Or processing? Or both?
    2. The system can tolerate the partitioning of the network;

       when it impossible for two subsets of the computers to communicate.
    3. The system tolerates the failure of individual computers.
  - □ By replicating the same data on more than one computer, we can increase system reliability.

Quality Requirements (6)







Quality Requirements (7)

- **Embedded Systems**
  - The systems with the SW is one of many components and often has no interface to the end user but rather the SW has interfaces with the other system components and probably controls them.
  - The SW is used in airplanes, robots, microwave ovens, ... And so on.
  - What make this systems different from others? The interface with other devices rather than humans.
  - Ex. Coin-operated vending machine.
  - Ex. A hospital patient-monitoring system that maintain database from patient histories.

Quality Requirements (8)

Examples of Embedded Systems

Many Different Products Depend on Embedded Systems