Nonfunctional Testing

SE 323 Software Construction, Testing and Maintenance

What are the non function Requirements?

College of Arts, Media and Technology
Chiang Mai University

How to test it?

Non Functional Requirements

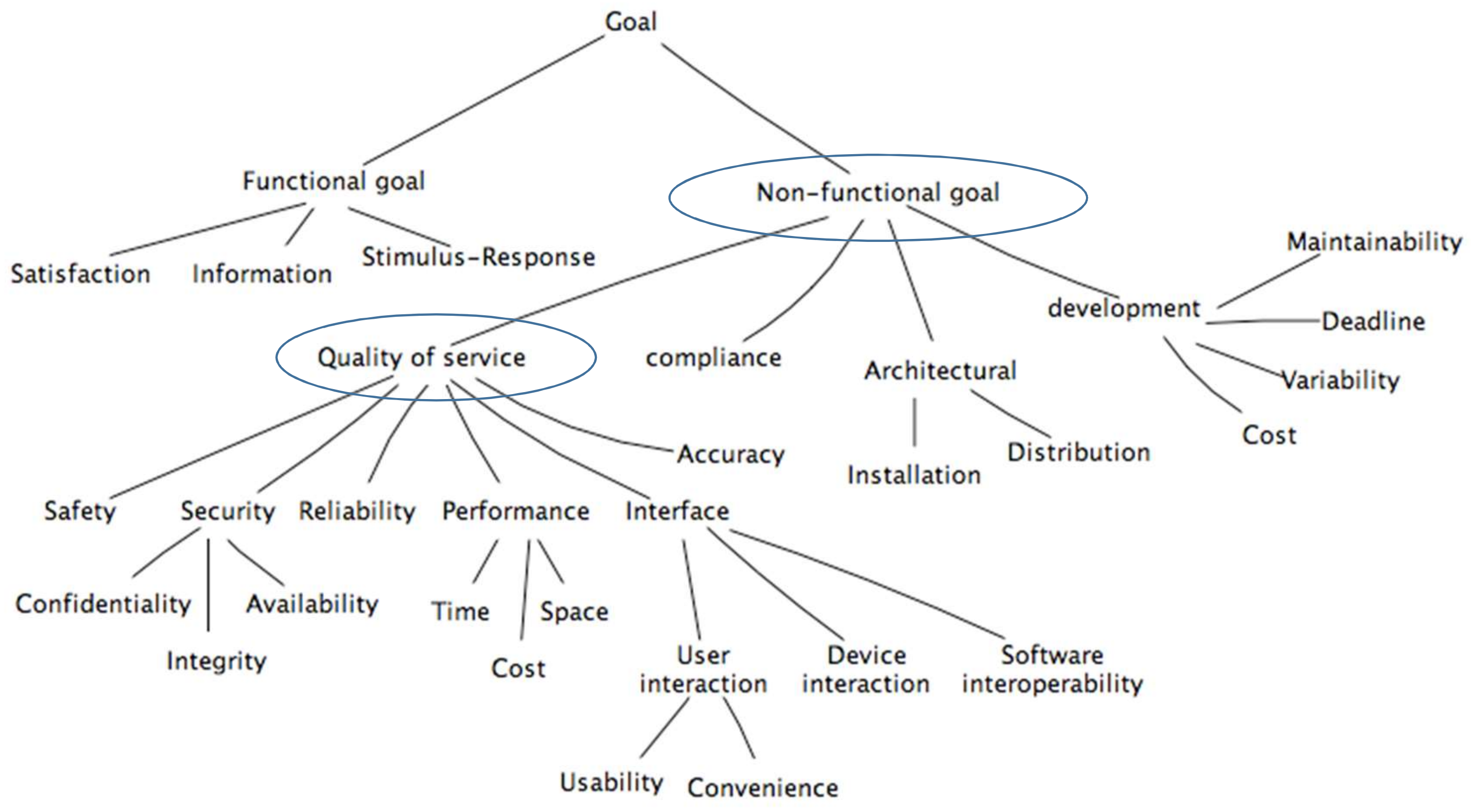
- The overall qualities or attributes of the resulting system
- Place restrictions on the product being developed, the development process, and specify external constraints that the product must meet.
- Examples of NFR
    - safety, security, usability, reliability and performance requirements.

CAMT
College of Arts, Media and Technology
Chiang Mai University

Concrete requirements from high level goals

Goal categorization:

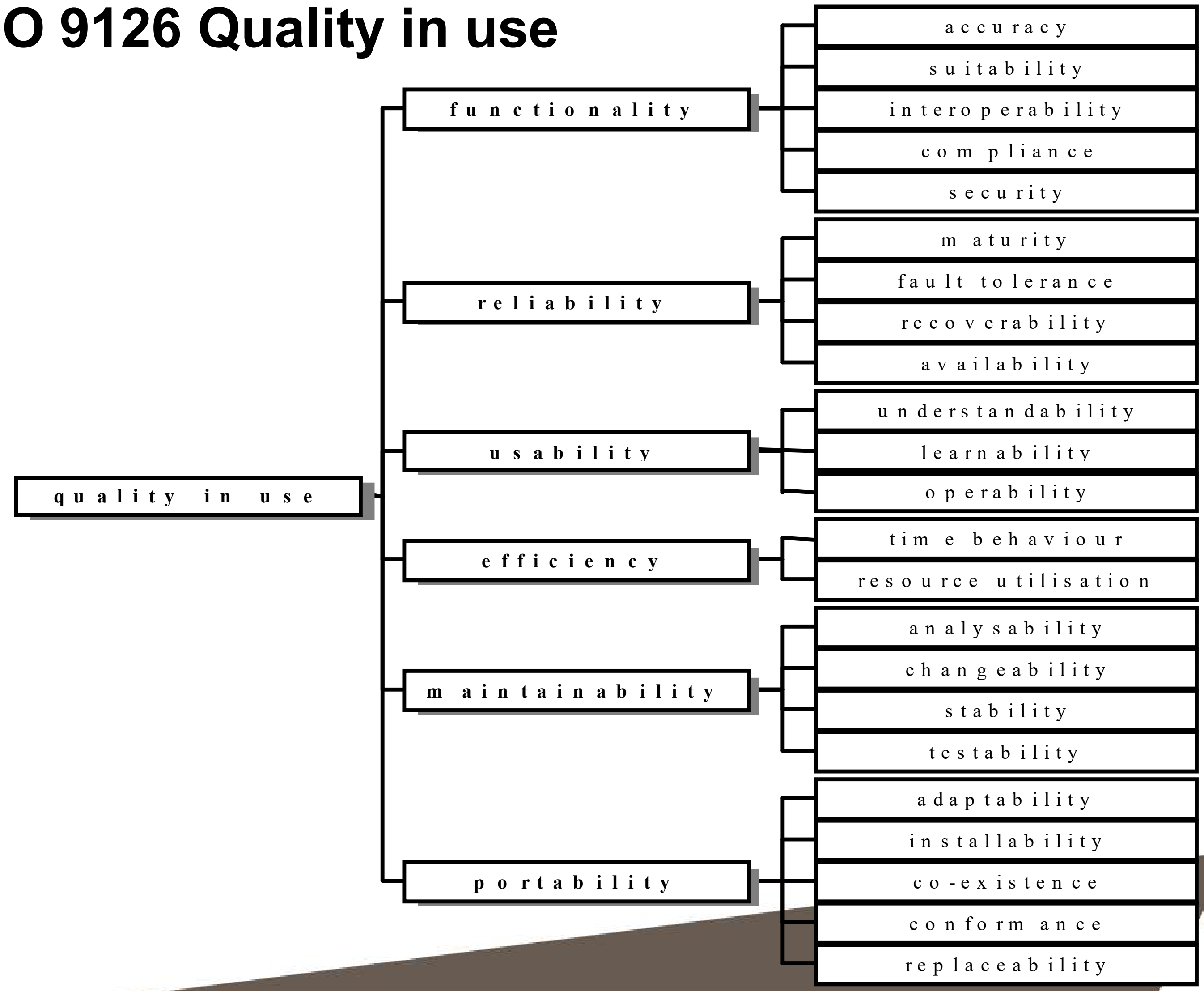
Similar to requirements categorizations:



Non-Functional Requirement - ISO 9126

- ISO 9126 - non-functional requirements linked to “quality in use”.
- Quality in use - users experience when using the system.
  - Since the users’ experience is subjective, many of the quality factors will also be subjective.
- Not an ideal situation, but a situation that we must live with.

ISO 9126 Quality in use



CAMT
College of Arts, Media and Technology
Chiang Mai University

Software Qualities

- Think of everyday objects
  - e.g. a chair
  - How would you measure the “quality”?
    - Construction quality? (e.g. strength of the joint?,…)
    - Aesthetic values? (e.g. elegance,…)
    - Fit for purpose? (e.g. comfortable)
- All quality measures are relative
  - There is no absolute scale
  - We can sometimes say A is better than B
    - … but it is usually hard to say how much better?

Software Qualities

- Construction quality?
  - Software is not manufactured
- Aesthetic value?
  - But most of the software is invisible
  - Aesthetic value matters for the user interface, but is only marginal concern
- Fit for purpose?
  - What’s the purpose? What’s fit?

Factors vs. Criteria

- Quality Factor
  - Customer-related concerns
    - e.g.: efficiency, integrity, reliability, correctness, survivability, usability,…
- Criteria
  - Technical concerns
    - Completeness, consistency, visibility,…
  - Use to analyze the factor
    - Validate by the criteria

Reliability – Factor

- The capability of the software to maintain the level of performance of the system when used under specified conditions
- Wear or aging does not occur in software.
- Limitations in reliability are due to faults in requirements, design, and implementation.
- Failures due to these faults depend on the way the software product is used and the program options selected rather than on elapsed time.

Reliability – Criteria

- Maturity: Capability of the software to avoid failure as a result of faults in the software.
- Fault tolerance: Capability of the software to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.

Reliability – Criteria

- Recoverability: Capability of the software to re-establish its level of performance and recover the data directly affected in the case of a failure.
- Availability: Capability of the software to be in a state to perform a required function at a given point in time, under stated conditions of use.

Usability – Factor

- Capability of the software to be understood, learned, used and liked by the user, when used under specified conditions.
- Some aspects of functionality, reliability and efficiency will also affect usability, but for the purposes of this International Standard are not classified as usability.

Usability – Criteria

- Understandability: Capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use.
- Learnability: Capability of the software product to enable the user to learn its application

Usability – Criteria

- Operability: Capability of the software product to enable the user to operate and control it.
- Likeability: Capability of the software product to be liked by the user.

Efficiency – Factor

- The capability of the software to provide the required performance relative to the amount of resources used, under stated conditions
- Resources may include other software products, hardware facilities, materials, (e.g. print paper, diskettes).

Efficiency – Criteria

- Time behavior: Capability of the software to provide appropriate response and processing times and throughput rates when performing its function, under stated conditions.
- Resource utilisation: Capability of the software to use appropriate resources in an appropriate time when the software performs its function under stated conditions.

Maintainability – Factor

- The capability of the software to be modified.
- Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements, and functional specifications.

Maintainability – Criteria

- Changeability: Capability of the software product to enable a specified modification to be implemented.
- Stability: Capability of the software to minimise unexpected effects from modifications of the software
- Testability: Capability of the software product to enable modified software to be validated.

Portability – Factor

- The capability of software to be transferred from one environment to another.
- The environment may include organizational, hardware or software environment.

Portability – Criteria

- Adaptability: Capability of the software to be modified for different specified environments without applying actions or means other than those provided for this purpose for the software considered.
- Installability: Capability of the software to be installed in a specified environment.

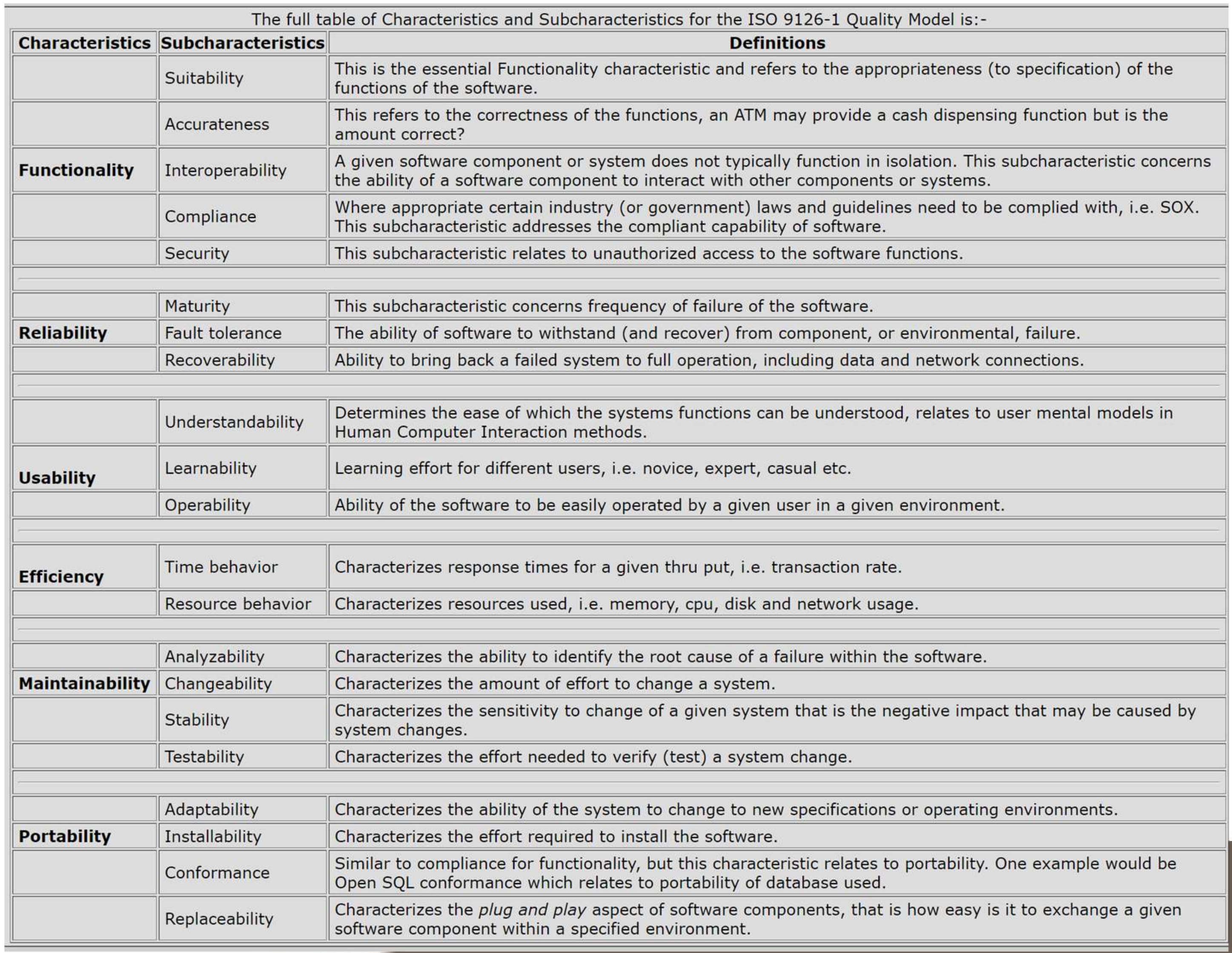
Portability – Criteria

- Co-existence: Capability of the software to co-exist with other independent software in a common environment sharing common resources
- Conformance: Capability of the software to adhere to standards or conventions relating to portability.
- Replaceability: Capability of the software to be used in place of other specified software in the environment of that software.

CAMT
College of Arts, Media and Technology
Chiang Mai University

Homework Activities

- Select one criteria in any factor
- Design how will you test them



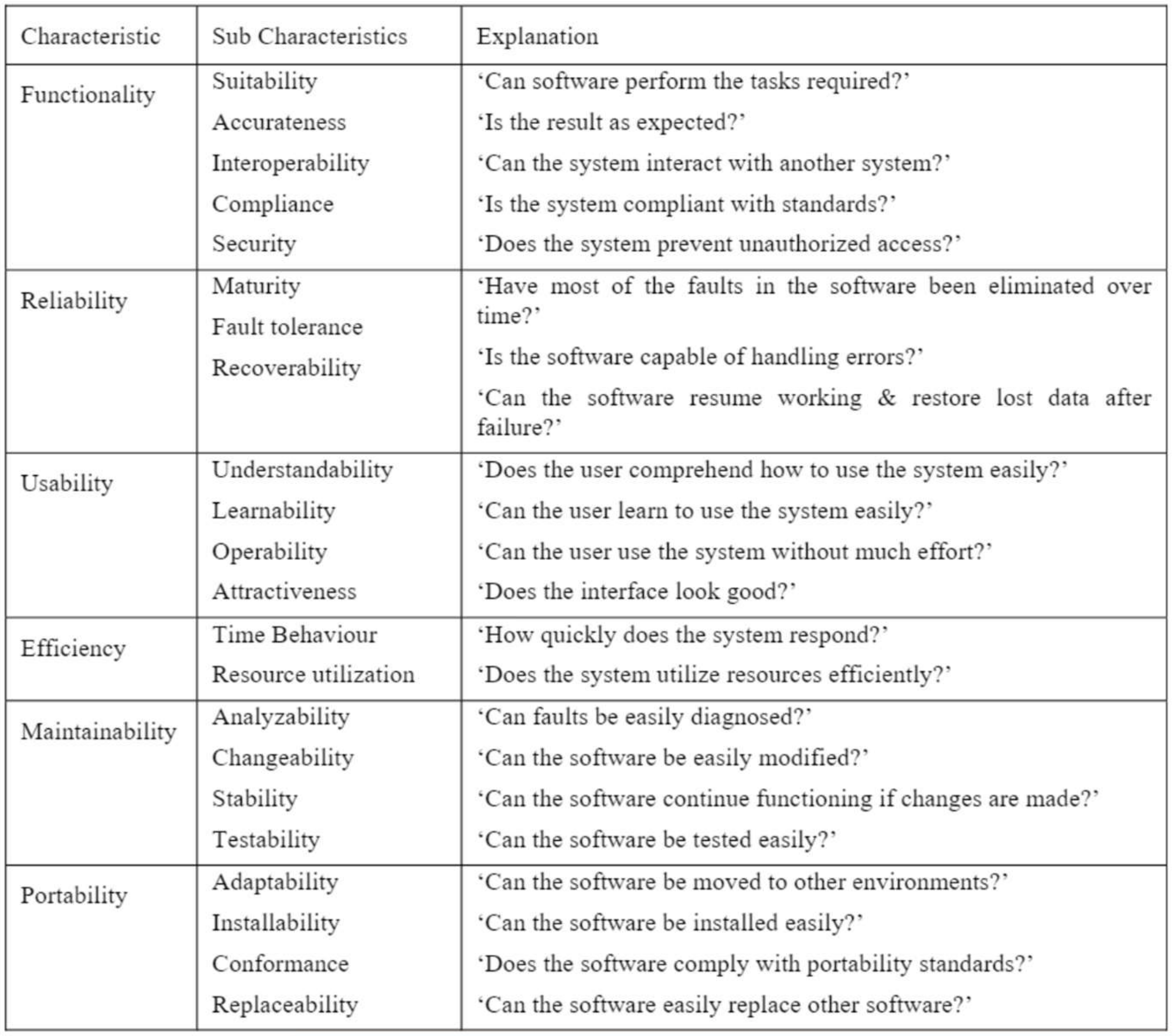
The full table of Characteristics and Subcharacteristics for the ISO 9126-1 Quality Model is:-

Characteristics	**Subcharacteristics**	**Definitions**
	Suitability	This is the essential Functionality characteristic and refers to the appropriateness (to specification) of the functions of the software.
	Accurateness	This refers to the correctness of the functions, an ATM may provide a cash dispensing function but is the amount correct?
Functionality	Interoperability	A given software component or system does not typically function in isolation. This subcharacteristic concerns the ability of a software component to interact with other components or systems.
	Compliance	Where appropriate certain industry (or government) laws and guidelines need to be complied with, i.e. SOX. This subcharacteristic addresses the compliant capability of software.
	Security	This subcharacteristic relates to unauthorized access to the software functions.
	Maturity	This subcharacteristic concerns frequency of failure of the software.
Reliability	Fault tolerance	The ability of software to withstand (and recover) from component, or environmental, failure.
	Recoverability	Ability to bring back a failed system to full operation, including data and network connections.
	Understandability	Determines the ease of which the systems functions can be understood, relates to user mental models in Human Computer Interaction methods.
Usability	Learnability	Learning effort for different users, i.e. novice, expert, casual etc.
	Operability	Ability of the software to be easily operated by a given user in a given environment.
Efficiency	Time behavior	Characterizes response times for a given thru put, i.e. transaction rate.
	Resource behavior	Characterizes resources used, i.e. memory, cpu, disk and network usage.
	Analyzability	Characterizes the ability to identify the root cause of a failure within the software.
Maintainability	Changeability	Characterizes the amount of effort to change a system.
	Stability	Characterizes the sensitivity to change of a given system that is the negative impact that may be caused by system changes.
	Testability	Characterizes the effort needed to verify (test) a system change.
	Adaptability	Characterizes the ability of the system to change to new specifications or operating environments.
Portability	Installability	Characterizes the effort required to install the software.
	Conformance	Similar to compliance for functionality, but this characteristic relates to portability. One example would be Open SQL conformance which relates to portability of database used.
	Replaceability	Characterizes the *plug and play* aspect of software components, that is how easy is it to exchange a given software component within a specified environment.

http://www.sqa.net/iso9126.html

CAMT
College of Arts, Media and Technology
Chiang Mai University

Table 2: ISO 9126Characteristic and sub-characteristics [7]



Characteristic	Sub Characteristics	Explanation
Functionality	Suitability	'Can software perform the tasks required?'
	Accurateness	'Is the result as expected?'
	Interoperability	'Can the system interact with another system?'
	Compliance	'Is the system compliant with standards?'
	Security	'Does the system prevent unauthorized access?'
Reliability	Maturity	'Have most of the faults in the software been eliminated over time?'
	Fault tolerance	'Is the software capable of handling errors?'
	Recoverability	'Can the software resume working & restore lost data after failure?'
Usability	Understandability	'Does the user comprehend how to use the system easily?'
	Learnability	'Can the user learn to use the system easily?'
	Operability	'Can the user use the system without much effort?'
	Attractiveness	'Does the interface look good?'
Efficiency	Time Behaviour	'How quickly does the system respond?'
	Resource utilization	'Does the system utilize resources efficiently?'
Maintainability	Analyzability	'Can faults be easily diagnosed?'
	Changeability	'Can the software be easily modified?'
	Stability	'Can the software continue functioning if changes are made?'
	Testability	'Can the software be tested easily?'
Portability	Adaptability	'Can the software be moved to other environments?'
	Installability	'Can the software be installed easily?'
	Conformance	'Does the software comply with portability standards?'
	Replaceability	'Can the software easily replace other software?'

https://www.researchgate.net/publication/228987388_ISO_9126_external_systems_quality_characteristics_sub-characteristics_and_domain_specific_criteria_for_evaluating_e-Learning_systems

Non Functional Requirement Testing

- Quantitative Technique
  - The data that can be measured
  - Strictly on numerical measurement
- Qualitative Technique
  - The data that depending on the observer
  - An observation or interpretation

Maintainability Requirements - 1

- Changeability:
  No error shall need more than one person-days to identify and fix
- Stability:
  Not more than 10% of the corrections shall have side-effects
- Testability:
  The correction shall need no more than one person-day of testing.
  This includes all necessary regression testing

Reliability Requirements

- Maturity:
  MTTF (Mean Time to Failure)
  = TTT(Time to Test) / n.
  MTTF > 500 hrs.
- Fault tolerance:
  Under no circumstances shall the system crash.

Reliability Requirements

- Recoverability:
  In case of an error, the time needed to get the system up and running again shall not exceed one hour (MTTR-Mean time to repair).
- Availability:
  MTTF /(MTTF + MTTR) > 0.998

Reliability Tests – 1

- MTTF = TTT / n. MTTF > 500 hrs. Use 10 PCs. Test for two weeks => TTT = 800. Not more than one error.
- Under no circumstances shall the system crash. Generate random input sets. No check for result is necessary – only crash / no crash,
- In case of an error, the time needed to get the system up and running again shall not exceed one hour (MTTR).

Reliability Tests – 2

- We need to consider three data:
- The total testing time – TTT.
  - For how long have we tested the system?
- The usage frequency – UF.
  - How often is the system used at the users’ site?
- Number of users each time the system is used
- We need to distinguish between test time – TTT – and usage time.

College of Arts, Media and Technology
Chiang Mai University

Reliability Tests – 3

- Simple Example:
- Have TTT = 400 hrs.
- The system will be used one hour once a week
  - e.g. for accounting purposes – at 10 sites.
- We then have 10 hrs. of use per week.
  - Under the assumption that all 10 sites use the system the way it is tested, our test is equivalent to 40 weeks of real use.

More Non-functional Requirements

- Relatively simple to make requirement and tests for reliability, maintainability and other “objective” non-functional factors.
- Subjective factors (e.g. usability) are more difficult.
  - Will use usability as an example to show the couplings between

Usability requirements – 1

- Understandability: The capability of the software product to enable the user to understand whether the software is suitable, and how it can be used for particular tasks and conditions of use.
- Learnability: The capability of the software product to enable the user to learn its application

Usability requirements – 1

- Operability: The capability of the software product to enable the user to operate and control it.
- Likeability: The capability of the software product to be liked by the user.

Usability Requirements – 2

- All the usability criteria are subjective.
- As a result, tests will also have a strong component of subjectivism.
- Understand
  - Whether the software is suitable for a particular task
  - How it can be used for particular tasks
  - Under which conditions it can be used

Usability Requirements – 2

- Learn its application
- Operate and control it (the software)
- Is the software product liked by the user?

How to measure

- Investigation
  - Look how understand the users are
- Questionnaire
  - Likert Scale
    - Rating scale

    1. Strongly disagree
    2. Disagree
    3. Neither agree nor disagree
    4. Agree
    5. Strongly agree

How to measure

- Define the criteria to measure
- Check whether the users satisfy or not

Criteria	1	2	3	4	5	Remarks
The theme of the user interface is appropriated						
The size of the button is appropriated						

Acceptance Criteria

- Specify when to finish the test
- Not all tests may be executed before the deployment
- Small errors are accepted

Acceptance criteria

- Quantitative methods
  - The functional requirements are all tests
  - The MTTR should not more than 2 mins
- Qualitative methods
  - The UI is satisfied more than 4.00
  - The new user can use the major functions with 3.5 satisfaction

Homework

- Define your Nonfunctional testing for your
- Also define the exit criteria

College of Arts, Media and Technology
Chiang Mai University

Q & A

College of Arts, Media and Technology
Chiang Mai University