

Beyond Functionality (Quality Requirements)

Topic # 10
Chapter 14 – Karl Wiegers
Chapter 7 - Reference

Non-functional requirements (NFR)

- Non-functional requirements define the overall qualities or attributes of the resulting system
- Non-functional requirements place restrictions on the product being developed, the development process, and specify external constraints that the product must meet.
- Examples of Non Functional Requirements
 - safety, security, usability, reliability and performance requirements.

Functionality to Quality

- There is so more to success than just delivering the right functionality.
- Users also have expectations, often unstated, about how well the product will work.
- Such characteristics, collectively known as *quality attributes*, *quality factors*, *quality requirements*, *quality of service requirements*, or the “–ilities,” constitute a major portion of the system’s nonfunctional and quality requirements .

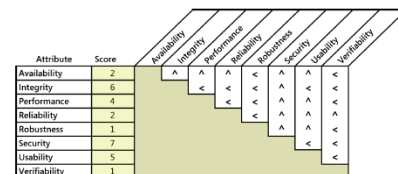
Internal versus External Quality

- One way to classify quality attributes distinguishes those characteristics that are visible through execution of the software (external quality) from those that are not (internal quality).
- External quality factors are primarily important to users, whereas internal qualities are more significant to development and maintenance staff.

Software Quality Attributes

External quality	Brief description
Availability	The extent to which the system's services are available when and where they are needed
Installability	How easy it is to correctly install, uninstall, and reinstall the application
Integrity	The extent to which the system protects against data inaccuracy and loss
Interoperability	How easily the system can interconnect and exchange data with other systems or components
Performance	How quickly and predictably the system responds to user inputs or other events
Reliability	How long the system runs before experiencing a failure
Robustness	How well the system responds to unexpected operating conditions
Safety	How well the system protects against injury or damage
Security	How well the system protects against unauthorized access to the application and its data
Usability	How easy it is for people to learn, remember, and use the system
Internal quality	Brief description
Efficiency	How efficiently the system uses computer resources
Modifiability	How easy it is to maintain, change, enhance, and restructure the system
Portability	How easily the system can be made to work in other operating environments
Reusability	To what extent components can be used in other systems
Scalability	How easily the system can grow to handle more users, transactions, servers, or other extensions
Verifiability	How readily developers and testers can confirm that the software was implemented correctly

Example: Quality Attribute prioritization



Sample quality attribute prioritization for an airport check-in kiosk.

- After prioritizing elicit specific expectations for each attribute.
- Then specify the quality requirements in a well structured way

Specifying Quality Requirements

- Must be well structured.
- SMART – so that one can measure them
- Shouldn't be vague or subjective or unreal
- **SUBJECTIVE & VAGUE:** “the system shall be user friendly”
- **UNREALISTIC:** “The system shall be available 24/7”

EXTERNAL QUALITY REQUIREMENTS

Availability

- Availability is a measure of the planned up time during which the system's services are available for use and fully operational.
- $\text{Availability} = [\text{MTTF} / (\text{MTTF} + \text{MTTR})] * 100\%$
- Scheduled maintenance periods also affect availability.
- Availability requirements are particularly complex and important for websites, cloud-based applications, and applications that have users distributed throughout many time zones.
- **Example:**
 - *AVL-1. The system shall be at least 95 percent available on weekdays between 6:00 A.M. and midnight Eastern Time, and at least 99 percent available on weekdays between 3:00 P.M. and 5:00 P.M. Eastern Time.*
 - *AVL-2. Down time that is excluded from the calculation of availability consists of maintenance scheduled during the hours from 6:00 P.M. Sunday Pacific Time, through 3:00 A.M. Monday Pacific Time.*

Example - Availability

AVL-1: The COS shall be available at least 98% of the time between 5:00 A.M. and midnight local time and at least 90% of the time between midnight and 5:00 A.M. local time, excluding scheduled maintenance windows.

Installability

- Software is not useful until it is installed on the appropriate device or platform.
- Installability describes how easy is it to perform installation operations correctly.
- Installability addresses the following activities:
 - Initial installation
 - Recovery from an incomplete, incorrect, or user-aborted installation
 - Reinstallation of the same version
 - Installation of a new version
 - Reverting to a previous version
 - Installation of additional components or updates
 - Uninstallation

Installability

- A measure of a system's installability is the mean time to install the system.
- **Examples:**
 - *INS-1. An untrained user shall be able to successfully perform an initial installation of the application in an average of 10 minutes.*
 - *INS-2. When installing an upgraded version of the application, all customizations in the user's profile shall be retained and converted to the new version's data format if needed.*
 - *INS-3. The installation program shall verify the correctness of the download before beginning the installation process.*
 - *INS-4. Installing this software on a server requires administrator privileges.*
 - *INS-5. Following successful installation, the installation program shall delete all temporary, backup, obsolete, and unneeded files associated with the application.*

Integrity

- Integrity deals with preventing information loss and preserving the correctness of data entered into the system.
- Data needs to be protected against threats such as accidental loss or corruption, apparently identical data sets that do not match, physical damage to storage media, accidental file erasure, or data overwriting by users.
- Intentional attacks that attempt to deliberately corrupt or steal data are also a risk.
- Data integrity also addresses the accuracy and proper formatting of the data. This includes concerns such as formatting of fields for dates, restricting fields to the correct data type or length, ensuring that data elements have valid values, checking for an appropriate entry in one field when another field has a certain value, and so on.

Integrity

- *INT-1. After performing a file backup, the system shall verify the backup copy against the original and report any discrepancies.*
- *INT-2. The system shall protect against the unauthorized addition, deletion, or modification of data.*
- *INT-3. The Chemical Tracking system shall confirm that an encoded chemical structure from third party structure-drawing tool represents a valid chemical structure*
- *INT-4. The system shall confirm daily that the application executables have not been modified by the unauthorized code.*
- **Backups (frequency)**

Data Integrity Examples

DI-1: The COS shall retain individual Patron meal orders for 6 months following the meal's delivery date.

DI-2: The COS shall retain menus for 1 year following the menu date.

Interoperability

- Interoperability indicates how readily the system can exchange data and services with other software systems and how easily it can integrate with external hardware devices
- **Example:**
 - *IOP-1. The Chemical Tracking System shall be able to import any valid chemical structure from the ChemDraw (version 13.0 or earlier) and MarvinSketch (version 5.0 or earlier) tools.*
- Interoperability requirements might dictate that standard data interchange formats be used to facilitate exchanging information with other software systems.
- Ask questions like: what other systems must this one interface? What services or data must they exchange? Standard data formats? Communication protocol b/w systems?

Performance

- Performance is one of the quality attributes that users often will bring up spontaneously. Performance represents the responsiveness of the system to various user inquiries and actions, but it encompasses much more than that.
- Satisfying performance requirements can be tricky because they depend so much upon external factors such as the speed of the computer being used, network connections, and other hardware components.

Some aspects of performance

Performance dimension	Example
Response time	Number of seconds to display a webpage
Throughput	Credit card transactions processed per second
Data capacity	Maximum number of records stored in a database
Dynamic capacity	Maximum number of concurrent users of a social media website
Predictability in real-time systems	Hard timing requirements for an airplane's flight-control system
Latency	Time delays in music recording and production software
Behavior in degraded modes or overloaded conditions	A natural disaster leads to a massive number of emergency telephone system calls

Performance

- Performance is external quality attribute which is closely related to internal quality attribute "efficiency".
- Providing rationales to guide developers with appropriate design choices when documenting performance requirements is recommended.
- For instance, stringent database response time demands might lead the designers to mirror the database in multiple geographical locations.
- Specify the number of transactions per second to be performed, response times, and task scheduling relationships for real-time systems.
- Specify memory and disk space requirements, concurrent user loads, or the maximum number of rows stored in database tables.

Performance

- **Examples:**

- *PER-1. Authorization of an ATM withdrawal request shall take no more than 2.0 seconds.*
- *PER-2. Webpages shall fully download in an average of 3 seconds or less over a 30 megabits/second Internet connection.*
- *PER-3. At least 98 percent of the time, the trading system shall update the transaction status display within 1 second after the completion of each trade.*

Performance – More examples

PER-1: The system shall accommodate a total of 400 users and a maximum of 100 concurrent users during the peak usage time window of 9:00 A.M. to 10:00 A.M. local time, with an estimated average session duration of 8 minutes.

PER-2: 95% of webpages generated by the COS shall download completely within 4 seconds from the time the user requests the page over a 20 Mbps or faster Internet connection.

PER-3: The system shall display confirmation messages to users within an average of 3 seconds and a maximum of 6 seconds after the user submits information to the system.

Robustness

- Robustness is the degree to which a system continues to function properly when confronted with invalid inputs, defects in connected software or hardware components, external attack, or unexpected operating conditions.
- Robust s/w recovers from internal failures without adversely affecting the end-user experience.
- Other attribute terms associated with robustness are *fault tolerance* (are user input errors caught and corrected?), *survivability* (can the camera experience a drop from a certain height without damage?), and *recoverability* (can the PC resume proper operation if it loses power in the middle of an operating system update?).

Robustness

- *ROB-1. If the text editor fails before the user saves the file, it shall recover the contents of the file being edited as of, at most, one minute prior to the failure the next time the same user launches the application.*
- *ROB-2. All plot description parameters shall have default values specified, which the Graphics Engine will use if a parameter's input data is missing or invalid.*

Robustness - Example

ROB-1: If the connection between the user and the COS is broken prior to a new order being either confirmed or terminated, the COS shall enable the user to recover an incomplete order and continue working on it.

Reliability

- The probability of the software executing without failure for a specific period of time is known as reliability.
- Reliability problems can occur because of improper inputs, errors in the software code itself, components that are not available when needed, and hardware failures.
- Robustness and availability are closely related to reliability.
- Often measured by MTBF
- **Examples:**
 - *REL-1. No more than 5 experimental runs out of 1,000 can be lost because of software failures.*
 - *REL-2. The mean time between failures of the card reader component shall be at least 90 days.*

Safety

- Safety requirements deal with the need to prevent a system from doing any injury to people or damage to property.
- Might be dictated by government regulations, business rules, legal or certification issues.
- An application to let people order meals from a cafeteria might include a safety requirement like the following:
 - *SAF-1. The user shall be able to see a list of all ingredients in any menu items, with ingredients highlighted that are known to cause allergic reactions in more than 0.5% of North American Population*

Safety

- **Examples:**
 - *SAF-2. If the reactor vessel's temperature is rising faster than 5°C per minute, the Chemical Reactor Control System shall turn off the heat source and signal a warning to the operator.*
 - *SAF-3. The system shall terminate any operation within 1 second if the measured tank pressure exceeds 90 percent of the specified maximum pressure.*

Security

- Security deals with blocking unauthorized access to system functions or data, ensuring that the software is protected from malware attacks, and so on.
- Protection against viruses, worms, Trojan horses, spyware, rootkits, and other malware
- Encryption of security data

Security

- **Examples:**
 - *SEC-1. The system shall lock a user's account after four consecutive unsuccessful logon attempts within a period of five minutes.*
 - *SEC-2. The system shall log all attempts to access secure data by users having insufficient privilege levels.*
 - *SEC-3. A user shall have to change the temporary password assigned by the security officer to a previously unused password immediately following the first successful logon with the temporary password.*
 - *SEC-4. A door unlock that results from a successful security badge read shall keep the door unlocked for 8.0 seconds, with a tolerance of 0.5 second.*
 - *SEC-5. The resident antimalware software shall quarantine any incoming Internet traffic that exhibits characteristics of known or suspected virus signatures.*
 - *SEC-6. Only users who have Auditor access privileges shall be able to view customer transaction histories.*

Security – More examples

- SEC-1. All network transactions that involve financial information or personally identifiable information shall be encrypted per BR-33.*
- SEC-2: Users shall be required to log on to the COS for all operations except viewing a menu.*
- SEC-3: Only authorized Menu Managers shall be permitted to work with menus, per BR-24.*
- SEC-4. The system shall permit Patrons to view only orders that they placed.*

Usability

- User-friendliness, ease of use and human engineering.
- Usability measures the effort required to prepare input for a system, operate it, and interpret its outputs.

Possible design approaches for ease of learning and ease of use

Ease of learning	Ease of use
Verbose prompts	Keyboard shortcuts
Wizards	Rich, customizable menus and toolbars
Visible menu options	Multiple ways to access the same function
Meaningful, plain-language messages	Autocompletion of entries
Help screens and tooltips	Autocorrection of errors
Similarity to other familiar systems	Macro recording and scripting capabilities
Limited number of options and widgets displayed	Ability to carry over information from a previous transaction
	Automatically fill in form fields
	Command-line interface

Usability

- **Examples:**

- *USE-1. A trained user shall be able to submit a request for a chemical from a vendor catalog in an average of three minutes, and in a maximum of five minutes, 95 percent of the time.*
- *USE-2. All functions on the File menu shall have shortcut keys defined that use the Control key pressed simultaneously with one other key. Menu commands that also appear in Microsoft Word shall use the same default shortcut keys that Word uses.*
- *USE-3. 95 percent of chemists who have never used the Chemical Tracking System before shall be able to place a request for a chemical correctly with no more than 15 minutes of orientation.*

Usability Examples

USE-1: The COS shall allow a Patron to retrieve the previous meal ordered with a single interaction

USE-2: 95% of new users shall be able to successfully order a meal without errors on their first try.

INTERNAL QUALITY REQUIREMENTS

Efficiency

- Efficiency is closely related to the external quality attribute of performance.
- Efficiency is a measure of how well the system utilizes processor capacity, disk space, memory, or communication bandwidth.
- If a system consumes too much of the available resources, users will encounter degraded performance.
- **Examples:**
 - *EFF-1. At least 30 percent of the processor capacity and memory available to the application shall be unused at the planned peak load conditions.*
 - *EFF-2. The system shall provide the operator with a warning message when the usage load exceeds 80 percent of the maximum planned capacity.*

Modifiability

- Modifiability addresses how easily the software designs and code can be understood, changed, and extended.
- High modifiability is critical for systems that will undergo frequent revision, such as those being developed by using an incremental or iterative life cycle.
- Ways to measure modifiability include the average time required to add a capability or fix a problem, and the percentage of fixes that are made correctly.
- **Example:**
 - *MOD-1. Function calls shall not be nested more than two levels deep.*

Supportability (Modifiability contd)

- Hardware devices containing embedded software often have requirements for supportability in the field.
- Some of these lead to software design choices, whereas others influence the hardware design.
 - *SUP-1. A certified repair technician shall be able to replace the scanner module in no more than 10 minutes.*
- Supportability requirements might also help make the user's life easier.
 - *SUP-2. The printer shall display an error message if replacement ink cartridges were not inserted in the proper slots.*

Portability

- The effort needed to migrate software from one operating environment to another is a measure of portability.
- Portability has become increasingly important as applications must run in multiple environments, such as Windows, Mac, and Linux; iOS and Android; and PCs, tablets, and phones. Data portability requirements are also important.
- **Examples:**
 - *POR-1. Modifying the iOS version of the application to run on Android devices shall require changing no more than 10 percent of the source code.*
 - *POR-2. The user shall be able to port browser bookmarks to and from Firefox, Internet Explorer, Opera, Chrome, and Safari.*

Reusability

- Reusability indicates the relative effort required to convert a software component for use in other applications.
- Reusable software must be modular, well documented, independent of a specific application and operating environment, and somewhat generic in capability.
- Reusability goals are difficult to quantify.
- **Examples:**
 - *REU-1. The chemical structure input functions shall be reusable at the object code level in other applications.*
 - *REU-2. At least 30 percent of the application architecture shall be reused from the approved reference architectures.*
 - *REU-3. The pricing algorithms shall be reusable by future store-management applications.*

Scalability

- Scalability requirements address the ability of the application to grow to accommodate more users, data, servers, geographic locations, transactions, network traffic, searches, and other services without compromising performance or correctness.
- Scalability has both hardware and software implications.
- **Examples:**
 - *SCA-1. The capacity of the emergency telephone system must be able to be increased from 500 calls per day to 2,500 calls per day within 12 hours.*
 - *SCA-2. The website shall be able to handle a page-view growth rate of 30 percent per quarter for at least two years without user-perceptible performance degradation.*
 - *SCA-3. The distribution system shall be able to accommodate up to 20 new warehouse centers.*

CONSTRAINTS

Constraints

- A constraint places restrictions on the design or implementation choices available to the developer.
- Constraints can be imposed by external stakeholders, by other systems that interact with the one you're building or maintaining, or by other life cycle activities for your system, such as transition and maintenance.
- Other constraints result from existing agreements, management decisions, and technical decisions
- **Examples:**
 - *CON-1. The user clicks at the top of the project list to change the sort sequence. [specific user interface control imposed as a design constraint on a functional requirement]*
 - *CON-2. Only open source software available under the GNU General Public License may be used to implement the product. [implementation constraint]*

Constraints

- *CON-3. The application must use Microsoft .NET framework 4.5. [architecture constraint]*
- *CON-4. ATMs contain only \$20 bills. [physical constraint]*
- *CON-5. Online payments may be made only through PayPal. [design constraint]*
- *CON-6. All textual data used by the application shall be stored in the form of XML files. [data constraint]*

END OF LECTURE # 10

- COMING UP!!!!!!
- Stakeholder Analysis
- Requirements Management
- Semester Exam

43