



Non Functional Product Requirements (ilieties)



MANAGEMENT SUMMARY

This whitepaper list several Non functional, Illeties or Quality Requirements



TABLE OF CONTENTS

1 INTRODUCTION.....	2
2 STUCTURE.....	2
3 THE LIST OF ILLETIES.....	3
4 REFERENCES.....	4

1 Introduction

Non-functional requirements or quality requirements are typically requirements in the area of e.g. performance, reliability, maintainability, etc.

The so called: illeties.

- Have often a system wide character and cannot be attached to a specific functional requirement.
- Are often difficult to quantify (measure!)
Compliance to requirements that can't be measured is difficult to proof.
e.g. "maintainability is important" is not a good (measurable) requirement

2 Stucture

The ISO standard: ISO 9126: Software Quality Model
defines 6 Main Quality Characteristics

The table below has its origin in: ISO 9126 standard and lists a number of Quality Characteristics, their properties and property descriptions.

There are some modifications to the original ISO list:

- Main Quality Characteristic: Capability is from another source.
- Properties may have been shuffled among the Characteristics!
- Additional properties are added from other sources
- Some properties may appear twice as they may not be unique for a single characteristic.
- Some properties have the same name but have different descriptions for different characteristics
- Other sources use different property names for seemingly the same description.





3 The list of illeties

C	PROPERTY	DESCRIPTION
Functionality	Compliance	Adherence to application related or industry standards or regulations by law
	Security / Integrity / Privacy	Detection and prevention of unauthorised access (accidental or deliberate) to data managed by the system. (Incl data privacy.)
	Configurability	The degree to which the system can be adapted to different needs (users)
	Compatibility	The degree to which the system should be compatible to earlier or future versions or systems
	Correctness / Accuracy / Repeatability	The degree to which the system conforms to the stated requirements (including accuracy and repeatability of results)
	Suitability	The appropriateness (to specification) of the functions of the software
Reliability	Correctness / Accuracy	The degree to which the system conforms to the stated requirements (including accuracy of results)
	Maturity	Prevention of system failures by software errors.
	Availability	The degree to which (un attention) unauthorised access to the system prevented/protected.
	Fault tolerance / Survivability / Resilience / Recoverability / Autonomy / Robustness	The degree to which the system is able to maintain or re-establish a specified level of performance in case of failures with e.g. Data, resources, interfaces, service, devices, etc.
	Safety	The degree to which the system impacts the absence of unsafe system conditions
	Distributivity	The (redundant) software is located on different processing or storage devices.
Usability (for all type of users)	Understandability	The degree to which the user interface enables the user to understand the system and its operation. (incl. Error management)
	Attractiveness	The ability of the system to be attractive for the user. (e.g. Userinterface, packaging, price, etc.)
	Learnability	Learning effort for different users, e.i.. Novice, expert, casual, etc.
	Operability	The degree of effort and ease to operate the user interface (efficiency of the UI)
	Explicitness	The degree of clarity of the user interface with regard to the status of the system (or a function)
	Responsiveness	The degree to which the user interface responds to user expectations (processing feedback)
	Customisability	The degree to which the user interface can be customised by the user
Performance	Helpfulness	Availability of information for the user which support effective and efficient use of the system. E.g. documentation, help text, etc.
	Time behaviour	The degree to which the system provides appropriate response/processing times and throughput rates
	Resource utilisation / Efficiency / Effectiveness	The degree to which the system uses appropriate resources (ROM, RAM, communication channels)
Maintainability / Serviceability	Conciseness	No excessive information is present. (Reduce lines of code.)
	Correctability / Understandability / Analyzability / Traceability	The degree to which defects (bugs) can be identified (located) and repaired
	Manageability	The administrative aspects of modifications to the software. (Tools, media handling, backup, Configuration management, etc.)
	Consistency	Code/documentation uniformity in notation, symbology and terminology.
	Structure	Code/Documentation possesses a definite pattern of organization.



IMproved QUality SOLUTIONS

C	PROPERTY	DESCRIPTION
	Changeability	The amount of effort to change the system.
	Expandability / Modifiability / Augementability / Scalability	The degree to which the system can be extended with new functions to meet new needs
	Stability	The degree to which the system minimises unexpected effect from modifications
	Testability / Verifiability	The degree to which the developed or modified system can be tested
	Virtuality / Replaceability	The ability to replace functions, modules, devices, etc. without changing the interfaces.
	Future proofness	The degree to which the system supports evolutions in applied technologies and development infrastructure. (This includes escrow agreements.)
	Portability	Interoperability / Co-existence
Replaceability / Virtuality		The possibility to change a software component within an specified environment
Adaptability / Flexibility		The degree to which the system can be modified for different HW/SW environments
Installability		The degree to which the system can be easily installed and set up
Upgradability		The degree to which the system can be easily upgraded (new versions, releases)
Conformance / Commonality		The usage of (open) (portable) standards
Reusability		The degree to which part of the system can be completely or partly reused in another system
Creatability	Independence	The degree to which the system supports development by different teams
	Outsourceability	The degree to which the implementation of parts of the system can be outsourced
	Buy-in	The degree to which existing system components can be applied in the system (incl. standard technology)
	Integratability	The degree of effort for integrating parts of the system to a working product
	Manufacturability	The degree of effort (time) to produce the system

This list is to be used as a checklist to derive the non-functional requirements. Preferably, do not use the complete list of attributes as a table of contents. Do not extensively discuss or incorporate those attributes that are regarded less relevant for the application. Note that some product quality attributes might result in functional requirements (e.g. security can be realised with a password mechanism, usability is often implicitly included, time-behaviour requirements can be addressed specifically per functional requirement). These are typically described in chapters of a requirement document. In that case, a reference from this chapter to the source can be useful, to understand how the quality attribute is achieved with functional requirements.

4 References

Sources used:

- ISO 9126: Software Quality Model
- <http://www.sqa.net/iso9126.html>
- <http://www.qpit.ltd.uk/Software%20Quality%20Definition.html>
- <http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=COL&ObjectId=2909>
- <http://www.swqual.com/newsletter/vol1/no4/vol1no4.html>
- http://en.wikipedia.org/wiki/Software_quality
- <http://www.tmap.net/>
- Software Reliability Course by Vincent Ronteltap



IMproved QUality SOLUTIONS

*

ImQuSo Improved Quality Solutions.

Po Box 169

5540 AD REUSEL, The Netherlands

All mentioned names are used for identification purposes only and are trademarks or registered trademarks of their respective companies.

© Copyright 2010 ImQuSo.

ALL RIGHTS RESERVED.

Doc. ImQuSo-WP-2010-001 Rev. 1.2 2010-03-10

