
# Glossary of terms for Non-Functional Requirements and Project Requirements used in software project performance measurement, benchmarking and estimating

**VERSION 1.0**

**September 2015**

# Version control

| Date | Reviewer(s) | Modifications / Additions |
|---|---|---|
| September 2015 | See below | Version 1.0 |
| | | |

# Acknowledgements


| Version 1.0 authors and reviewers 2015 (alphabetical order) | | |
|---|---|---|
| Alain Abran - COSMIC<br>ETS Montreal,<br>Canada | Mauricio Aguiar- IFPUG<br>TI Metricas,<br>Brazil | Talmon Ben-Cnaan- IFPUG *<br>Amdocs,<br>Israel |
| Dácil Castelo - IFPUG<br>Leda MC,<br>Spain | Peter Fagg- COSMIC<br>Pentad Ltd<br>United Kingdom | Arlan Lesterhuis - COSMIC<br>The Netherlands |
| George M. Mitwasi- IFPUG<br>Optum Inc,<br>USA | Pablo Soneira- IFPUG<br>Sopra Steria,<br>Spain | Charles Symons- COSMIC *<br>United Kingdom |
| Frank Vogelezang- COSMIC<br>Ordina<br>The Netherlands | Chris Woodward- COSMIC<br>CW Associates<br>United Kingdom | |

* Editors of this Glossary

This document has also been reviewed by members of the COSMIC Measurement Practices Committee and of the IFPUG Non-Functional Sizing Standards Committee, and by the IFPUG Board of Directors.

# Foreword

An organization that wants to understand the performance of its software system projects needs to gather a lot of data about the requirements for and attributes of the delivered system, and the requirements and constraints for the projects. Exploiting these data to develop performance benchmarks and for estimating new projects needs consistent terminology and data definitions across all these activities.

The most important characteristic of the software that must be captured in support of these activities is a measure of its size. The COSMIC and IFPUG organizations[1] offer ISO standard methods of measuring a size of the Functional User Requirements (FUR) for software, but have differing ideas on what to do about so-called 'Non-Functional Requirements' (NFR). Examples of software system requirements that are often cited as NFR are: response time, security, availability, portability, programming language, etc.

The topic of NFR is difficult because there are so many possible types of NFR (over 100 according to some authors) and only rather fuzzy definitions of what are NFR. The boundaries between 'functional' and 'non-functional' requirements are often not clear, with no agreed standard distinction. In addition, there are many types of project requirements and constraints, but again no agreement on the most important of these parameters that should be captured across the activities of performance measurement, benchmarking and estimating and their definitions.

As a first step to try to introduce some common understanding, the COSMIC and IFPUG organizations decided to collaborate to produce this standard Glossary of terms for NFR and for project requirements and constraints (PRC). The exercise has also helped the two organizations towards a better understanding of what exactly are NFR.

In addition to this Glossary, each organization also publishes its own standard which goes into further detail on these concepts and how to consider or how to measure non-functional requirements (COSMIC [1] or IFPUG [2], respectively).

We hope that the Glossary will be valuable in providing a common language to anyone involved in developing requirements for a software system project and/or who participates in the processes of contracting, developing or managing software projects, even if not specifically concerned with software measurement and its uses.

Chapter 1 is an introduction that presents the purpose of this document

Chapter 2 gives the COSMIC and IFPUG definitions for FUR, for NFR and for PRC.

Chapter 3 describes a classification scheme for NFR and for PRC to help the users of this Glossary to understand their scope, and to explain the reasons for the choice of which terms to include in the Glossary.

Chapter 4 contains the Glossary of NFR terms and their definitions, (ISO wherever possible), the Glossary of PRC terms and some terms that were excluded from the Glossaries for particular reasons

Frank Vogelezang

President, COSMIC

Kriste Lawrence

President, IFPUG

[1] 'COSMIC' = the Common Software Measurement International Consortium (www.cosmic-sizing.org). 'IFPUG' = the International Function Point Users Group (www.ifpug.org )

# Table of Contents

# Acronyms

The following acronyms are used in this Glossary

| | |
|---|---|
| **CMMI®** | Capability Maturity Model Integration |
| **COBIT** | Control Objectives for Information and Related Technology, http://www.isaca.org/cobit/pages/default.aspx |
| **COSMIC** | Common Software Measurement International Consortium |
| **FSM** | Functional Size Measurement |
| **FUR** | Functional User Requirements |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IEC** | International Electrotechnical Commission |
| **IFPUG** | International Function Point User Group |
| **ISBSG** | International Software Benchmarking Standards Group |
| **ISO** | International Organization for Standardization |
| **NFR** | Non Functional Requirements |
| **PMI®** | Project Management Institute |
| **PRC** | Project Requirements and Constraints |
| **PRINCE2** | PRojects IN a Controlled Environment, Version 2 |
| **ROI** | Return on Investment |
| **SPICE** | Software Process Improvement and Capability Determination |
| **SQuaRE** | System and software product Quality Requirements and Evaluation |

# 1

## INTRODUCTION

To ensure coherence and consistency across the activities of measuring the performance of software system projects, benchmarking and software system project estimating, it is vital to establish common understanding and terminology.

The purpose of this document is to define and classify the terms for non-functional requirements and project requirements, in order to create a common Glossary and a common understanding to be used for these activities.

Functional Size Measurement (FSM) methods are well defined and are used consistently across these activities, but they only measure part of the User Requirements. There are other types of User Requirements for the 'system' or for the 'software product', usually referred to as 'Non-Functional Requirements (NFR), which are not measured by current FSM methods and yet affect project effort. In addition, there are requirements for the project itself which do not affect the software size but may also affect effort.

The entire set of requirements and constraints for a project to deliver a software system can be divided into three components [2]:

- Functional User Requirements (FUR)
- Non-Functional Requirements (NFR)
- Project Requirements and Constraints. (PRC)

This document can be used to provide a common vocabulary and as a checklist for NFR and PRC for software development companies, integrators of software systems, customers of software projects and regulators in the three activities of software system project performance measurement, benchmarking and estimating.

The selection of NFR and PRC terms and their definitions that are proposed in this document are common to IFPUG and COSMIC. However, when different approaches are taken by the two organizations, this document presents the COSMIC approach and the IFPUG approach side by side.

[2] A software project may have requirements for 'other deliverables' besides the software, e.g. documentation, training, etc. These are beyond the scope of this Glossary.

# 2

## DEFINITIONS OF REQUIREMENTS

All definitions in this chapter have been developed and agreed by COSMIC and IFPUG except those shown as originated from ISO/IEC/IEEE.

### 2.1 The relationship between 'requirements' and 'constraints'

The terms 'requirements' and 'constraints' are often used inter-changeably, which can be confusing.

In ordinary English, a requirement is a necessary condition whilst a constraint is a limiting condition. It follows that all requirements are constraints, but not all constraints are requirements. Figure 1 illustrates this difference with examples for NFR and PRC by means of a Venn diagram.

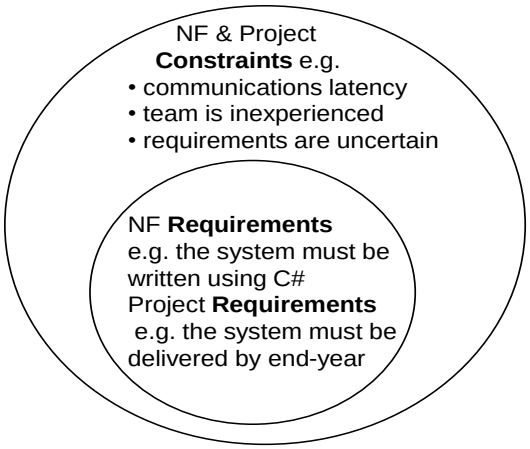


***Figure 1 - The relationship between constraints and requirements (Examples)***

*EXAMPLE: a requirement that the software shall be written in C# is also a constraint. But a situation where it happens that the functional requirements are uncertain and very difficult to establish, is a constraint, not a requirement.*

*EXAMPLE: Some terms can be either a constraint or a requirement depending on the context. Achieving a certain 'latency' target could be a requirement for real-time processing of audio signals, or latency could be a 'design constraint' for a space communications system.*

Constraints that are not requirements may only be recognized after a project is completed, e.g. in a post-project review. The importance of this point is that if we are to understand project performance properly, then we must take into account project constraints that were not requirements.

*EXAMPLE: A project might be measured as poorly-performing and a post-project review determined that this was due to the constraint that the team was inexperienced with the technology used. (However, if the team was very experienced with the technology, this should have been a positive factor for the project, not a 'constraint'.)*

**In this Glossary we will mainly use 'requirements' for convenience. We only use 'constraints' when it is helpful to distinguish them from requirements. This is particularly necessary for 'project requirements and constraints'**.

## 2.2 The 'things' to which requirements apply

It is important for understanding the definitions in this chapter that we distinguish the 'things' to which various types of requirements and constraints apply or that determine them.

First we must distinguish PRC, i.e. requirements and constraints for the *project*, from the FUR and NFR for the '*product'* delivered by the project. In the context of this Glossary, we define a 'project' (more specifically a 'software system project') as "**a temporary endeavor to achieve defined objectives of delivering a product by defined dates**", where the 'product' is "**a hardware/software system or an item of software such as a software package"**

ISO/IEC standards sometimes refer to 'Systems and software', e.g. [4, 6], sometimes to 'software product' e.g. [9] and sometimes to 'the software' [3]. For simplicity, we will refer to the product of a 'hardware/software system project' as a 'software system' and to the software part of the product as a 'software product'.

FUR and NFR (as defined in sections 2.3 and 2.4) for a software system or software product, can apply to five 'things'. These are:

- The **software**;
- The **data** maintained or used by the software;
- The **technology** to be used, e.g. 'the system must execute on a Unix platform', the requirements must be captured using CASE tool XYZ';
- **Other deliverables**, e.g. documentation or training;
- The combined **hardware/software system**[3], e.g. a response time or an availability requirement will apply to the hardware/software system as a whole (not just to the software);

Further, the business or organizational **environment** may impose some requirements or constraints on the software system, e.g. that the organization sponsoring the system is in a particular industry, that the system is subject to specific regulations, e.g. for safety, that it must support a specific number of concurrent users, etc.

The relationship between these various 'things' to which requirements can apply or that determine requirements is shown in Figure 2. (The one-to-many 'crows-foot' symbol indicates that a project may develop and/or enhance one or more hardware/software systems.)

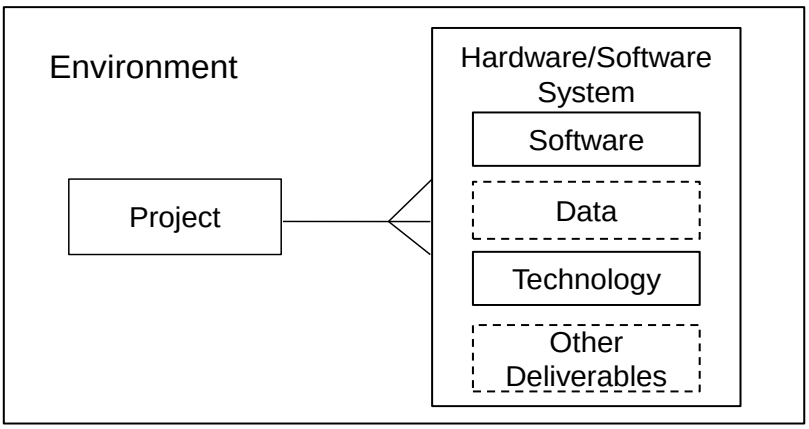


***Figure 2: The 'things' to which requirements and constraints may apply[4]***

[3] A 'system' may well be considered to include human business processes and support activities, or the totality of a machine that includes software (e.g. a vehicle 'power train' system), but in this Glossary we limit 'system' to mean a 'computer hardware-plus-software' system.

[4] The two 'things' in boxes with dotted outlines are not dealt with in this first version of the Glossary. 'Other deliverables' (e.g. documentation, training, etc.,) are beyond the scope of the Glossary. Non-functional requirements for 'Data' will be considered for inclusion in a future version of the Glossary

Two important observations in relation to Figure 2:

- From their definition (see 2.3), Functional User Requirements apply only to software.
- Other types of requirements that are commonly thought of as 'Non-Functional' can apply to the software system, to the software product, to the data, the technology and to other deliverables, or may be are determined by the environment.

## 2.3 Functional User Requirements

The ISO/IEC 14143-1:2007 definition [3] for the term 'Functional User Requirements' (or 'FUR') is used by both COSMIC and IFPUG.

| **Functional User Requirements (FUR) - ISO/IEC 14143-1 Definition** |
|---|
| A sub-set of the user requirements. Requirements that describe what the software shall do, in terms of tasks and services.<br>NOTE: Functional User Requirements relate to but are not limited to:<br>• data transfer (for example Input customer data; Send control signal)<br>• data transformation (for example Calculate bank interest; Derive average temperature)<br>• data storage (for example Store customer order; Record ambient temperature over time)<br>• data retrieval (for example List current employees; Retrieve latest aircraft position)<br>Examples of user requirements that are not Functional User Requirements include but are not limited to:<br>• quality constraints (for example usability, reliability, efficiency and portability)<br>• organizational constraints (for example locations for operation, target hardware and compliance to standards)<br>• environmental constraints (for example interoperability, security, privacy and safety)<br>• implementation constraints (for example development language, delivery schedule) |

## 2.4 Non-Functional Requirements

### *2.4.1 Definition*

There is no universally-accepted definition of a non-functional requirement.

ISO/IEC/IEEE 24765:2010 [4] defines non-functional requirements as:

| **Non-functional requirement - ISO/IEC/IEEE 24765:2010 Definition** |
|---|
| A software requirement that describes not what the software will do but how the software will do it. |

COSMIC and IFPUG have sought to develop a definition that may be easily compared and contrasted with the ISO/IEC/IEEE definition of FUR, to make it easier to distinguish FUR and NFR, as below:

| **Non-Functional requirements – Definition** |
|---|
| Any requirement for a software system or for a software product, including how it should be developed and maintained, and how it should perform in operation, except any functional user requirement for the software. Non-functional requirements concern:<br>• the software system or software product quality; |

- the environment in which the software system or software product must be implemented and which it must serve;
- the processes and technology to be used to develop and maintain the software system or software product and the technology to be used for their execution.

### *2.4.2 The main classes of NFR*

The list of NFR terms was developed starting from the list in [5], expanded by including a wider range of ISO or IEEE terms, then followed by some rationalization as explained in section 3.1. The resulting list comprises 60 terms.

The 60 terms are divided into three Main Classes (corresponding to the three bullet points of the NFR definition) to make them more manageable and easier to find, as shown in the tables in Chapter 3. A very important factor in deciding on this structure was to reconcile the NFR classification with the structure of the 'product quality model' of the ISO/IEC 25010:2011 'System and software product Quality Requirements and Evaluation' ('SQuaRE') standard [6].

| **Quality Requirements - Definition** |
|---|
| Requirements for the quality or for the architecture or design of a delivered software system or software product. |

| **System Environment Requirements - Definition** |
|---|
| Characteristics of the environment in which a software system or software product is developed and maintained and which it must support in operation, e.g., its user base, etc. |

| **Technical Requirements - Definition** |
|---|
| Requirements for how a software system or software product will be built, such as the programming language to be used and the technology (hardware and communications) that the software system or software product will need in operation. |

Notes:

1. From the COSMIC perspective, Quality NFR may evolve wholly or partly as a project progresses into functionality that increases the functional size of the software [1]. From the IFPUG perspective, Quality NFR contribute to the non-functional size [2].

   System Environment Requirements and Technical Requirements do not affect the functional size of the software product but may affect the project effort.

2. In addition to quality NFR for the software systems and software product, ISO/IEC has published a 'Data Quality Model' [9] which includes 15 terms that describe data quality from two perspectives, 'inherent' and 'system dependent', which partly overlap.

   'Inherent' terms concern data quality attributes which describe whether the data itself meets needs, e.g. Accuracy, Completeness, Consistency, and Credibility

   'System dependent' terms concern data quality attributes that depend on the system environment in which the data are used (hardware, devices, software etc.), e.g. Accessibility, Availability, Portability, and Precision.

Several of the terms in this Glossary with definitions relevant to NFR of the software system or software product also occur in the ISO/IEC 'Data Quality Model' with definitions relevant to the quality of data used or maintained by the software product [9].

This first version of the NFR Glossary does not include the definitions that are relevant to data quality. They will be considered for inclusion in a future version of the Glossary.

## 2.5 Project Requirements and Constraints

| **Project Requirements and Constraints - Definition** |
|---|
| Requirements that define how a software system project should be managed and resourced, or constraints that affect its performance.<br>Requirements may include:<br>• the targets the project should achieve (e.g. budget, delivery date, product quality);<br>• the project management processes that should be used;<br>• how the project should be governed and resourced.<br>Constraints may include:<br>• limitations on the project resources planned or needed;<br>• dependencies on other projects outside the control of the project concerned. |

## 2.6 Summary model of requirements for a software system project

Figure 3 shows the overall classification scheme for the various types of requirements that may arise in a software project, as used in this Glossary.

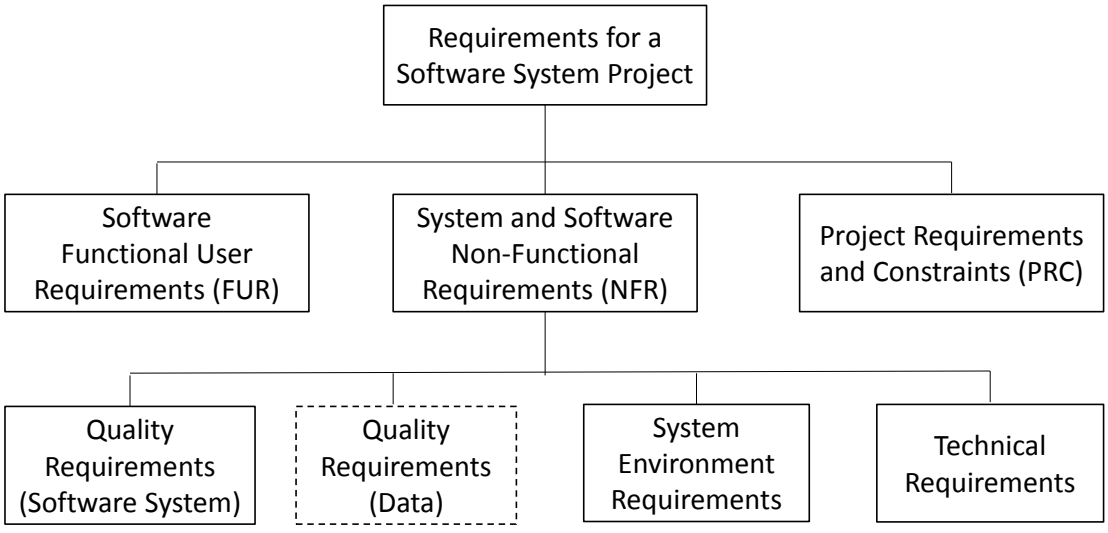


***Figure 3: Summary model of requirements for a software system project***

# 3

# SELECTION AND CLASSIFICATION OF NFR AND PROJECT REQUIREMENT AND CONSTRAINT TERMS

## 3.1 Selection of NFR terms

Because there are so many possible NFR terms and because of the overlaps, subtleties and variations of definition, there will probably never be a complete universally-accepted list. (One study reportedly listed 186 NFR terms. Another study [7] found 122 terms and structured them into a hierarchy.) The list of 60 NFR terms given in the Glossary aims to be reasonably comprehensive and a useful practical starting point.

Readers should feel free to modify this list for their own purposes and, if they feel strongly, suggest changes to the list in the Glossary, using the procedure of Appendix A.

Selection of terms involved many pragmatic judgments. First we included only terms that are 'elementary' NFR, i.e. terms that are not composites of, or derived from, or classes of other NFR terms. This is because the number of terms derived from two or more elementary NFR is huge. If needed, these can be defined locally.

*EXAMPLE 1: The term 'performance' is one of the parameters of the SQuaRE Quality Model [6], defined as 'the degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage'. The term 'performance' is therefore a composite NFR covering several possible parameters. To state a non-functional requirement of 'performance' would be meaningless without further information; you must specify a specific performance parameter, such as 'response time ', or 'transaction rate'. So 'performance' is not included.*

*EXAMPLE 2: All terms concerned with costs have been excluded, e.g. 'cost of ownership', 'ROI', etc., as they can all be derived from other data. Cost comparisons are, of course, extremely important but to understand them properly may require knowledge of cost accounting conventions (e.g. whether staff-rates are fully-loaded with all overheads, or only partially loaded), cost inflation (if comparing historic data) currency exchange rates, etc. These factors can be considered locally.*

*EXAMPLE 3: The ISBSG requirement to record Programming language 'level', (i.e. 2GL. 3 GL, 4 GL, etc.) [8], was omitted since these classes of 'programming language' are not well defined. Instead, 'programming language' and 'programming paradigm' are included.*

Second, we excluded terms that are 'sub-sorts' of other terms included in the Glossary.

*EXAMPLE 4: 'Maintainability' is included but it has many sub-sorts, e.g. modularity, modifiability, extendibility, flexibility, testability, etc. These sub-sort terms are not included. Maintainability might also be made easier by using re-usable software, hence 're-usability' may also be considered as a sub-sort of maintainability. However a requirement for 'reusability' is different from a requirement for 'maintainability', with different consequences for project activities, so both these terms are included.*

Third, we avoided terms with strongly overlapping definitions.

*EXAMPLE 5: 'Modifiability', 'Evolvability' and 'Extensibility' overlap with 'Adaptability'. Only the latter was included.*

Some decisions on what to include or exclude were marginal.

*EXAMPLE 6: The* ISO/IEC 25010:2011 *definition of 'adaptability' (defined as 'degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments') overlaps strongly with that*

*of 'portability' (defined as '(1) ease with which a system or component can be transferred from one hardware or software environment to another (*ISO/IEC/IEEE *24765:2010) (2) capability of a program to be executed on various types of data processing systems without converting the program to a different language and with little or no modification (ISO 2382-1:1993)). However, given the ISO/IEC 25010:2011 definition, 'adaptability' could also be applied to evolving business requirements. We therefore included both portability and adaptability.*

A few terms that have ISO definitions were excluded for reasons that might not be clear. See section 4.4 of the Glossary on 'Miscellaneous terms', where the reasons for their exclusion are given.

### *3.1.1 Quality Requirements*

The table below shows the 42 NFR terms in this class quite closely mapped to the eight groups of the ISO/IEC 25010:2011 'Product Quality' model [6]. A ninth group ('Related to system or software architecture or design') of terms that are not mentioned in ISO/IEC 25010:2011 also seems to fit naturally into this main class.

| | **Quality Group** | **NFR terms** |
|---|---|---|
| 1 | Related to the quality of the data maintained by the software | Accountability<br>Accuracy<br>Auditability<br>Precision<br>Validation (of data) |
| 2 | Related to system performance | Response time<br>Transaction rate |
| 3 | Related to compatibility | Co-existence<br>Compatibility<br>Interoperability |
| 4 | Related to the ease of use by the intended user | Accessibility<br>Aesthetics (of the UI)<br>Customer satisfaction (software)<br>Learnability<br>Multi-lingual support<br>Operability<br>Usability |
| 5 | Related to system reliability | Availability<br>Back-up<br>Dependability<br>Diversity<br>Failure management<br>Fault tolerance<br>Recoverability<br>Reliability<br>Safety |
| 6 | Related to control of access | Authenticity<br>Confidentiality<br>Non-repudiation<br>Privacy<br>Security<br>Usage mode (live vs training/testing) |

| | Quality Group | NFR terms |
|---|---|---|
| 7 | Related to maintainability | Adaptability<br>Maintainability<br>Reusability<br>Re-use type |
| 8 | Related to ease of deployment | Installability<br>Portability |
| 9 | Related to system or software architecture or design | Architecture/Design<br>Interfaces<br>Open source<br>Operational processing mode |

### *3.1.2 System Environment Requirements*

The six System Environment Constraints that characterize the environment that a software system must support are taken mainly from the ISBSG Data Collection forms [8].

| | System Environment Group | NFR terms |
|---|---|---|
| 1 | Context | Industry |
| 2 | Application Domain | Application type (or software type)<br>Application sub-type |
| 3 | Implementations | Implementations (no. of) |
| 4 | User Base | Distinct users - maximum number<br>Concurrent users - maximum number |

### *3.1.3 Technical Requirements*

The 12 Technical Requirements are taken mainly from the ISBSG Data Collection forms [8].

| | Technical Group | NFR terms |
|---|---|---|
| 1 | Operational Platform | Operational platform type<br>Operational platform physical distribution<br>Operational platform volatility |
| 2 | Database | Database management system<br>Database size |
| 3 | Operational Platform constraints | Communications network<br>Operational processor memory<br>Operational processor speed<br>Operational storage capacity |
| 4 | Development requirements | Methods and tools<br>Programming language<br>Programming paradigm |

## 3.2 Selection of Project Requirements and Constraints Terms

The 19 terms are mostly taken from ISBSG and PMI® Terminology [11].

| | Project Requirement and Constraint Group | PRC terms |
|---|---|---|
| 1 | Project Type | Project type (e.g. new vs enhancement) |
| 2 | Project Resources | Effort<br>Skills and experience level<br>Staffing level<br>Team relationships<br>Work breakdown structure |
| 3 | Project Quality | Customer satisfaction (project)<br>Defect count |
| 4 | Project Risk | Dependencies on other parties<br>Post-project review findings<br>Risk<br>Scope change |
| 5 | Project Processes | Development environment<br>Governance<br>Location<br>Process maturity<br>Project management method |
| 6 | Project Duration (Schedule) | Duration<br>Schedule compression / expansion |

# 4

## GLOSSARIES OF TERMS

This Glossary lists the terms in alphabetic order in their respective sections, selected according to the criteria described in section 3.1. Their definitions are taken from the sources listed below.

Readers who wish to add to the list or amend this list are asked to use the Change Request and Comment Procedure in the Appendix to this Guideline.

### 4.1 Sources of ISO standard and other definitions

| Doc. Reference No. | Document Title |
|---|---|
| Chambers | The Chambers Dictionary, 13th addition, 2014 |
| COSMIC | Definition proposed by the Measurement Practices Committee of the Common Software Measurement International Consortium, usually based on The Chambers Dictionary |
| IEC 60050-191 | International Electrotechnical Vocabulary, Chapter 191, Dependability and quality of service, 1990. |
| IEEE 982.1-2005 | IEEE Standard Dictionary of Measures of the Software Aspects of Dependability |
| IEEE 1012-2004 | IEEE Standard for Software Verification and Validation |
| ISBSG | Glossary of terms for software project development and enhancement v5.16a, 22/08/12 |
| ISO 5725-1:1994 | Accuracy (trueness and precision) of measurement methods and results — Part 1: General principles and definitions |
| ISO 9241-110: 2006 | Ergonomics of human-system interaction – Part 110: Dialogue principles |
| ISO 9241-171:2008 | Ergonomics of human-system interaction – Part 171: Guidance on software accessibility |
| ISO/IEC 2382-1: 1993 | Information technology–Vocabulary–Part 1: Fundamental terms |
| ISO/IEC 2382-20:1990 | Information technology–Vocabulary–Part 20: System development |
| ISO/IEC 10746-2:2009 | Information technology – Open Distributed Processing – Reference Model: Foundations |
| ISO/IEC 12207:2008 | Systems and software engineering–Software life cycle processes |
| ISO/IEC 15026-1:2013 | Systems and software engineering — Systems and software assurance — Part 1: Concepts and vocabulary |
| ISO/IEC 15288:2008 | Systems and software engineering–Software life cycle processes |
| ISO/IEC 20000-1:2011 | Information technology–Service management–Part 1: Service management system requirements |

| | |
|---|---|
| ISO/IEC/IEEE 24765:2010 | Systems and software engineering–Vocabulary |
| ISO/IEC 25010:2011 | System and software engineering–Systems and software product Quality Requirements and Evaluation (SquaRE)–System and software quality models |
| ISO/IEC 25062:2006 | Software engineering – Software product Quality Requirements and Evaluation (SquaRE) |
| ISO/IEC 42010:2011 | Systems and software engineering – Architecture description |
| ISBN-13: 893-7485908328 | Project Management Body of Knowledge, PMI® |
| Wikipedia | www.wikipedia.org |

## 4.2 Glossary of terms for a software system or software product

The following abbreviations are used in the Glossary:

'Env.' = System Environment Requirements; 'Tech.' = Technical Requirements;

N.B. The classification given for each term is not absolute. Some NFR terms could be classified under more than one heading.

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Accessibility | Quality | Ease of use | (1) Usability of a product, service, environment or facility by people with the widest range of capabilities (ISO/IEC 25062:2006)<br>(2) Degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use (ISO/IEC 25010:2011).<br>Note: Although "accessibility" typically addresses users who have disabilities, the concept is not limited to disability issues. The range of capabilities includes disabilities associated with age. (ISO 9241-171:2008) |
| Accountability | Quality | Data quality | (1) Degree to which the actions of an entity can be traced uniquely to the entity (ISO/IEC 25010:2011).<br>(2) State of being answerable for decisions and activities to the organizational governing bodies, legal authorities and, more broadly, its stakeholders (ISO 26000:2010) |
| Accuracy | Quality | Data quality | (1) A qualitative assessment of correctness, or freedom from error (ISO/IEC/IEEE 24765:2010).<br>(2) A quantitative measure of the magnitude of error (ISO/IEC/IEEE 24765:2010). The proximity of a result or a measure to the true value (ISO 5725-1:1994)<br>See also: 'Precision'. |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Adaptability | Quality | Maintainability | Degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments (ISO/IEC 25010:2011). Note: Adaptability includes the scalability of internal capacity, such as screen fields, tables, transaction volumes, and report formats. Adaptations include those carried out by specialized support staff, business or operational staff, or end users. If the system is to be adapted by the end user, adaptability corresponds to suitability for individualization as defined in ISO 9241-110:2006.<br>See also: 'Portability'<br>Related concepts: modifiability, evolvability, extendability, flexibility |
| Aesthetics (of the user interface) | Quality | Ease of use | Degree to which a user interface enables pleasing and satisfying interaction for the user (ISO/IEC 25010:2011).<br>Note: refers to properties of the product or system that increase the pleasure and satisfaction of the user, such as the use of color and the nature of the graphical design.'<br>Related concept: 'Customer experience (software)' |
| Application Domain | Env. | Application domain | See 'Application Type (or 'Software type')'. |
| Application Sub-Type | Env. | Application domain | A type of software within each of the four ISBSG 'Application Types'. (The ISBSG Glossary lists 20 sub-types of Business Applications, 8 sub-types of Real-time Applications, 7 sub-types of Mathematically-intensive software and 6 sub-types of Infrastructure software.) |
| Application Type (or 'Software Type') | Env. | Application domain | A classification of software into four groups: business applications, real-time applications, mathematically-intensive software, infrastructure software. (ISBSG) |
| Architecture / design | Quality. | System or software architecture or design | Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution. (ISO/IEC/IEEE 42010:2011). Examples: requirement to conform to the ISO 7-layer model, or to the AUTOSAR architecture [12]. |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Auditability | Quality | Data quality | Facility of a software system or software product to enable an auditor to examine whether data is processed correctly so as to meet requirements and internal or external audit standards (COSMIC/IFPUG).<br>Related concepts: assurance, compliance to regulations |
| Authenticity | Quality | Access control | The degree to which the identity of a subject or resource can be proved to be the one claimed (ISO/IEC 25020:2011) |
| Availability | Quality | System reliability | (1) Ability of a service or service component to perform its required function at an agreed instant or over an agreed period of time (ISO/IEC 20000-1:2011)<br>(2) The degree to which a system or component is operational and accessible when required for use (ISO/IEC 25010:2011)<br>Note: Availability is normally expressed as a ratio or percentage of the time that the service or service component is actually available for use by the customer to the agreed time that the service should be available. Availability is a combination of maturity (which reflects the frequency of failure), fault tolerance and recoverability (which reflect the length of downtime following each failure).<br>See also: 'Fault tolerance', 'Reliability', 'Recoverability' |
| Back-up | Quality | System reliability | (1) A system, component, file, procedure, or person available to replace or help restore a primary item in the event of a failure or externally caused disaster (ISO/IEC/IEEE 24765:2010).<br>(2) to create or designate a system, component, file, procedure, or person as a replacement (ISO/IEC/IEEE 24765:2010) |
| Co-existence | Quality | Compatibility | Degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product (ISO/IEC 25010:2011) |
| Communications network | Tech. | Operational Platform constraints | The type of data communication protocols that a software system or software product must observe, e.g. none, standard LAN/WAN protocols, special open protocols, proprietary or classified protocols, etc.' (COSMIC/IFPUG) |
| Compatibility | Quality | Compatibility | (1) Degree to which a product, system or component can exchange information with other products, systems or components, or perform its required functions, while sharing the same hardware or software environment (ISO/IEC 25020:2011).<br>(2) The ability of two or more systems or components to exchange information (ISO/IEC/IEEE 24765:2010)<br>(3) The capability of a functional unit to meet the requirements of a specified interface without appreciable modification (2382-1:1993) |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Concurrent users - maximum number | Env. | User base | The maximum number of users that the system can support concurrently under specified conditions (COSMIC/IFPUG) |
| Confidentiality | Quality | Access control | Degree to which a product ensures that data is accessible only by those authorized (ISO/IEC 25010:2011) Example degrees: 'internal use only', 'secret', 'top secret'. See also 'Privacy' |
| Customer Satisfaction (software) | Quality | Ease of use | The degree to which the customer of a software system or software product is satisfied with the system/product (COSMIC/IFPUG) |
| Database management system (software) | Tech. | Database | Software system that is used by an application to efficiently manage the access control, storage and retrieval of persistent data used by the application. Sometimes regarded as part of the infrastructure software (COSMIC/IFPUG) |
| Database size | Tech. | Database | (1) A measure of the physical storage space needed for a database, usually measured in units such as 'megabytes'.<br>(2) A measure of the size of a database in units relevant to the business application software that will use the database, e.g. no. of customers, no. of employees (COSMIC/IFPUG) |
| Dependability | Quality | System Reliability | (1) Trustworthiness of a computer system such that reliance can be justifiably placed on the service it delivers (IEEE 982.1-2005).<br>(2) Availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance (ISO/IEC 15026-1:2013)<br>(3) The ability to perform when required (IEC 60050-191:1990)<br>Note: Dependability characteristics include availability and its inherent or external influencing factors, such as availability, reliability (including fault tolerance and recoverability), security (including confidentiality and integrity), maintainability, durability, and maintenance support. (taken from the definition of 'Reliability' in ISO/IEC 25010:2011) |
| Disaster recovery | Quality | System Reliability | See 'Recoverability'' |
| Distinct users - maximum number | Env. | User base | The maximum number of distinctly identifiable users that the system can support (COSMIC/IFPUG) |
| Distributed Processing | Tech. | Operational Platform | See 'Operational Platform type' |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Diversity | Quality | System Reliability | In fault tolerance, realization of the same function by different means (ISO/IEC 24765:2010) Example: use of different processors, storage media, programming languages, algorithms, or development teams. |
| Ease of use | Quality | Ease of use | See 'Usability' |
| Emotional Factors | Quality | Ease of use | See 'Aesthetics' (of the user interface) |
| Failure Management | Quality | System reliability | The management of failures from their occurrence until full resolution (COSMIC/IFPUG), where 'failure' is defined as (1) termination of the ability of a product to perform a required function or its inability to perform within previously specified limits (ISO/IEC 25000:2005) (2) an event in which a system or system component does not perform a required function within specified limits (ISO/IEC/IEEE 24765:2010) |
| Fault tolerance | Quality | System reliability | (1) The ability of a system or component to continue normal operation despite the presence of hardware or software fault (ISO/IEC 25010:2011).<br>(2) Pertaining to the study of errors, faults, and failures, and of methods for enabling systems to continue normal operation in the presence of faults. cf. error tolerance, fail safe, fail soft, fault secure, robustness. (ISO/IEC/IEEE 24765:2010)<br>See also 'Operational Platform type' |
| Implementations (number of) | Env. | Implementations | The number of times that a software system or software product must be installed. See also 'Installability', as 'installation' is virtually a synonym of 'implementation'. (COSMIC/IFPUG)<br>Note: Normally, the effort for a development project includes only the first implementation. |
| Industry | Env. | Context | The type of business that a software system or software product must support, as identified by a Standard Industry Code. SIC codes 'are assigned based on common characteristics shared in the products, services, production and delivery system of a business'. (Wikipedia) |
| Installability | Quality | Ease of deployment | Degree of effectiveness and efficiency with which a product or system can be successfully installed or uninstalled in a specified environment (ISO/IEC 25010:2011)<br>'Installation' is defined as 'system development phase at the end of which the hardware, software and procedures of the system become operational (ISO/IEC 2382:2015)<br>See also 'Implementations (number of)'.<br>Related concept: configurability |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Interfaces | Quality | System or software architecture or design | Shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical signal exchanges, and other characteristics (ISO/IEC 25010:2011)<br>Related concepts: autonomy, inter-process communication<br>For project interfaces, see 'Dependencies on other parties' |
| Interoperability | Quality | Compatibility | Degree to which two or more systems, products or components can exchange information and use the information that has been exchanged (ISO/IEC 25010:2011) |
| Learnability | Quality | Ease of use | Degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use (ISO/IEC 25010:2011) Note: Can be specified or measured either as the extent to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use, or by product properties corresponding to suitability for learning as defined in ISO 9241-110:2006.<br>Related concept: teachability |
| Maintainability | Quality | Maintainability | Ease with which a software system or component can be modified to change or add capabilities, correct faults or defects, improve performance or other attributes, or adapt to a changed environment (ISO/IEC/IEEE 24765:2010).<br>Note: Maintainability includes installation of updates and upgrades. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications. Modifications include those carried out by specialized support staff, and those carried out by business or operational staff, or end users.<br>See also: 'Adaptability'.<br>Related concepts: comprehensibility, modularity, supportability |
| Methods and Tools | Tech. | Development requirements | Procedures for carrying out tasks, and supporting software aids used by the project team. (COSMIC/IFPUG)<br>See also 'effort'.<br>Note: methods and tools used should be recorded by the principal software activities, i.e. requirements determination, analysis, design, programming, testing, implementation, maintenance and support.<br>See also 'Project Management methods'. |
| Multilingual support | Quality | Ease of use | Requirement for a system to be usable in two or more natural languages (COSMIC/IFPUG) |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Non-repudiation | Quality | Access control | Degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later. (ISO/IEC 250101:2011) |
| Open source | Quality | System or software architecture or design | Requirement to use open source software. (COSMIC/IFPUG) |
| Operability | Quality | Ease of use | Degree to which a product or system has attributes that make it easy to operate and control (ISO/IEC 250101:2011)<br>Note: Operability corresponds to controllability, (operator) error tolerance, and conformity with user expectations as defined in ISO 9241-110:2006. |
| Operational Platform physical distribution | Tech. | Operational platform | An indicator of whether the platform on which a software system or software product is required to execute is located at a single site or is distributed over multiple sites. (COSMIC/IFPUG)<br>Note: not to be confused with a requirement to implement the software system or software product on a single platform at multiple sites. |
| Operational Platform type | Tech. | Operational platform | The hardware/software environment on which a software system or software product executes. Examples: shared utility (e.g. 'cloud'); main-frame; mid-range; PC; embedded; mobile; multi-platform (for a distributed system); parallel (or 'array') processor, communications network processor (e.g. a router). (ISBSG, COSMIC/IFPUG) |
| Operational Platform volatility | Tech. | Operational platform | An indicator of whether an operational platform (hardware or software) is stable or changes often. (COSMIC/IFPUG) |
| Operational processing mode | Quality | System or software architecture or design | An indicator of whether a software system or software product is required to execute transactions on-demand ('i.e. 'on-line'); in batches; mixed on-line and in batches; or subject to real-time constraints. (COSMIC/IFPUG) |
| Operational processor memory | Tech. | Operational Platform constraints | The memory capacity of the processor on which a software system or software product executes. (Used to indicate whether the processor memory is limited, thus requiring special effort when developing the software.) (COSMIC/IFPUG) |
| Operational processor speed | Tech. | Operational Platform constraints | The speed of the processor on which a software system or software product executes. (Used to indicate whether the processor speed is limited, thus requiring special effort when developing the software system.) (COSMIC/IFPUG) |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Operational storage capacity | Tech. | Operational Platform constraints | The on-line storage capacity available to an executing software system or software product. (Used to indicate whether the storage capacity is limited, thus requiring special effort when developing the software system.) (COSMIC/IFPUG) |
| Portability | Quality | Ease of deployment | (1) Ease with which a system or component can be transferred from one hardware or software environment to another (ISO/IEC/IEEE 24765:2010)<br>(2) capability of a program to be executed on various types of data processing systems without converting the program to a different language and with little or no modification (2382-1:1993) |
| Precision | Quality | Data quality | The degree of exactness or discrimination with which a quantity is stated (24765:2010) Example: a precision of 2 decimal places versus a precision of 5 decimal places |
| Privacy | Quality | Access control | Ability of a software system or software product to protect personal data from unauthorized or unwarranted disclosure (COSMIC/IFPUG). See also 'Confidentiality' |
| Programming language | Tech. | Development requirements | The computer languages in which a software system or software product is required to be programmed e.g. C, C#, Java (COSMIC/IFPUG) |
| Programming paradigm | Tech | Development requirements | A fundamental style of computer programming, a way of building the structure and elements of computer programs (Wikipedia), e.g. procedural, object-oriented, imperative, literate, declarative, functional, logic, symbolic, synchronous, etc. |
| Recoverability | Quality | System reliability | Degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system (ISO/IEC 25010:2011) |
| Reliability | Quality | System reliability | (1) The ability of a system or component to perform its required functions under stated conditions for a specified period of time (ISO/IEC/IEEE 24765:2010)<br>(2) Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time (ISO/IEC 25010:2011). |
| Response time | Quality | System performance | The elapsed time between the end of an inquiry or command to an interactive computer system and the beginning of the system's response (ISO/IEC/IEEE 24765:2010).<br>Related concept: 'latency' |
| Reusability | Quality | Maintainability | Degree to which an asset can be used in more than one system, or in building other assets, (ISO/IEC 25010:2011)<br>See also: 'Re-use type'. |
| Re-use type | Quality | Maintainability | Types of re-usable assets, e.g. re-usable requirements, designs, code (modules, object classes), test suites, and documentation (COSMIC/IFPUG) |

| NFR Term | Class | Group | Definition |
|---|---|---|---|
| Safety | Quality | System reliability | Expectation that a system does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered (ISO/IEC/IEEE 24765:2011) |
| Scalability | Quality | Maintainability | See 'Adaptability' |
| Security | Quality | Access control | (1) Protection of information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them (ISO/IEC 12207:2008)<br>(2) The protection of computer hardware or software from accidental or malicious access, use, modification, destruction, or disclosure. Security also pertains to personnel, data, communications, and the physical protection of computer installations. (ISO/IEC 1012-2012)<br>See also: 'Accountability', 'Authenticity', 'Confidentiality', 'Non-repudiation', 'Privacy'.<br>Related concept: integrity. |
| Transaction rate | Quality | System performance | The rate at which a defined mix of transactions is processed by a software system or software product on a defined operational platform; it may be a target rate or an actual rate and it may be the average rate over a defined time-period, a maximum rate or a percentile rate (e.g. 90% of the transactions shall complete faster than the target rate). Synonym: 'throughput rate'. (COSMIC/IFPUG)<br>Note: A transaction is the implementation of a software system or software product requirement that may correspond to a part of, or a whole, or more than one COSMIC functional process, or similarly correspond to an IFPUG elementary process. |
| Usability | Quality | Ease of use | (1) Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (ISO/IEC 25010:2011)<br>(2) Ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component (ISO/IEC/IEEE 24765:2010).<br>See also 'Accessibility', 'Aesthetics (of the user interface)', 'Learnability', and 'Operability'.<br>Related concepts: appropriateness, recognizability, user error protection. |
| Usage modes | Quality | Access control | Requirement for a software system or software product to be able to be used in different modes, i.e. live, test, training, or combinations thereof (COSMIC/IFPUG) |
| User numbers | Env. | User Base | (See 'Distinct users – maximum number' and 'Concurrent users – maximum number') |
| Validation (of data) | Quality | Data quality | Process of controlling that the data entered into a software system or software product satisfies requirements allocated to software in terms of format, range and type of permitted data values. |

| | | | |
|---|---|---|---|
| | | | The process should not allow invalid data to enter a data store and should inform the user of the nature of any defects. (COSMIC/IFPUG, partly based on IEEE 1012-2004) |

## 4.3 The Glossary of project requirement and constraint terms

In this Chapter, the word 'project' in any of the terms means a project of any 'Project type' as defined below.

Where 'locally' appears in a definition, this could mean 'within your organization' or 'for a given benchmarking exercise', i.e. whatever is appropriate depending on the context

All term definitions are proposed by COSMIC and IFPUG, unless another source is given explicitly.

| Term | Project Group | Definition |
|---|---|---|
| Customer Satisfaction (project) | Project Quality | The degree to which the customer of a software system or software product is satisfied with the project that developed or enhanced it. |
| Defect count | Project Quality | The number of defects, within a defined period starting from the date of first implementation of a software system or software product. (Defect counts should be classified by severity and may be target or actual.)<br>Note: 'Defect count' is an attribute of the delivered software product. However it is not a Quality Requirement of the product. It is classified as a Project Requirement and Constraint term, i.e. as an attribute of a project, together with other project performance-related characteristics, such as effort and duration. |
| Dependencies on other parties | Risk | Dependencies of the project activities on activities that are performed by other parties, e.g. other projects or decision-making bodies, which may affect the progress of the project. |
| Development environment | Processes | The hardware/software platforms used by the development project. To be recorded if different from the Operational Platform.<br>See also the classification of 'Operational Platform type' |
| Duration (Schedule) | Duration | The elapsed time for a project from Project Start Date (when a project is given resources and starts work) until Project Finish Date (the end of first site implementation).<br>NOTE: Both the estimated and actual duration should be recorded, the latter excluding periods when the project was inactive.<br>See also 'Schedule compression/expansion' |

| Term | Project Group | Definition |
|---|---|---|
| Effort | Resources | The amount of work (in labor units such as staff-months) required to complete a project.<br>Notes:<br>1) Effort must be further clarified locally, e.g. it may:<br>• be estimated, planned or actual;<br>• be for a whole project or broken down by activity (see 'work breakdown');<br>• define whether users, customers, or support staff (e.g. database specialists, project management office staff, etc.) are included in or excluded from the project team<br>2) Project Team: The people who report either directly or indirectly to the project manager. (PMI) |
| Governance | Processes | (1) The management framework within which project decisions are made, e.g. COBIT, PRINCE, etc.<br>(2) The organization that is accountable for the project, e.g. a Steering Committee, Change Control Board. |
| Location | Processes | The country(ies) or site(s) where the project takes place.<br>Note: Project location may be classified as e.g.: On-site, Multi-site, Near-shore, Multi-country, Off-shore, etc.<br>Off-shore: The practice of hiring external organizations to perform work in a country other than the one where the products or services are required or will be used; Near-shore: The practice of hiring external organizations to perform work in neighboring countries. |
| Post-project review findings | Risk | (1) Measures of actual project performance, e.g. actual productivity, customer satisfaction (project), defect count, etc.<br>(2) Factors (positive and negative) identified in a post-project review that affected the project outcome, such as unanticipated staff turnover, scope or technology changes, experienced team, etc.<br>Note 1: Ideally the impact on the planned effort and/or schedule should be estimated for each factor.<br>Note 2: see also 'Scope change (Scope creep)' |
| Process maturity | Processes | The level of adherence of the project processes to a quality standard, e.g. as per CMMI®, SPICE [10] or similar assessment. |
| Project management method(s) | Processes | A method for dividing project activities into distinct phases (or stages, or iterations) for the purposes of planning and control.<br>Note: Common project management methods include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and agile methods.<br>(Also known as software development methodology, or software development life cycle.) |

| Term | Project Group | Definition |
|---|---|---|
| Project type | Type | A class of software project dependent on its purpose in relation to the software. A project type may be New development, Enhancement, Maintenance, Re-development (ISBSG), where 'Maintenance' includes Adaptive, Corrective, Perfective and Preventive maintenance.<br>Notes:<br>1. The criterion for when an activity is considered as a maintenance activity and when it is an enhancement project should be defined locally.<br>2. Maintenance may also be defined as a continuing activity to evolve the system and not as a project |
| Risk | Risk | (1) The aggregate probability of a project not succeeding in meeting its goals. (COSMIC/IFPUG)<br>(2) An uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives (PMI®)<br>Aggregate risk is usually derived from other data that encompasses the size of the software to be delivered, anticipated requirements stability & validity, staff skills and experience in the problem area, stakeholders cohesion etc. Risk analysis may take into account the impact of failing to meet project goals and the uncertainty in the risk assessment. |
| Schedule compression / expansion | Duration | The degree to which the target project duration ('schedule') is compressed or expanded compared with the estimated duration that is ideally or optimally estimated as needed.<br>Note: The PMI defines 'schedule compression' as 'taking actions to decrease the total project duration after analyzing number of alternatives to determine how to get the maximum duration compression for the least cost.' |
| Scope change ("Scope creep") | Risk | Any change to the project's scope. A scope change almost always requires an adjustment to the project cost or schedule (PMI®) |
| Skills and experience level | Resources | The degree to which the human resources who perform the project as defined by the plan have the necessary skills and expertise to perform or support the processes they are assigned to (after CMMI®) |
| Staffing level | Resources | The number of staff employed on the project.<br>Note: Need to distinguish the average number over the life of the project from the peak number of staff, and planned versus actual. |
| Team relationships | Resources | Any factor that affects the team's ability to work effectively, e.g. team stability, culture (single/multi-culture) and cohesion, physical working conditions, relationships with non-project staff, e.g. other development teams, users, customers, specialist staff, etc. |
| Work-breakdown structure | Resources | A deliverable-oriented grouping of project work elements that organizes and defines the total work scope of the project. (PMI) |

## 4.4 The Glossary of miscellaneous terms

This Chapter contains terms that:

- were considered for inclusion but were NOT included in the Glossary. The reasons for their exclusion are given;
- or are used in the analysis of project performance data, but are not NFR nor project requirements

| Term | Definition |
|---|---|
| Complexity | Composed of more than one or many parts; not simple or straightforward; intricate, difficult. (Chambers). Note that there can be several types of complexity of software: algorithmic, architectural, data, process, operational, semantic, etc.<br>Excluded because the term is very general, with many possible types. |
| Control | (1) In engineering, the monitoring of system output to compare with expected output and taking corrective action when the actual output does not match the expected output (ISO/IEC/IEEE 24765:2010).<br>(2) A requirement that a software system or software product must operate, regulate or direct some other device or process, probably in real-time (COSMIC/IFPUG)<br>Excluded because the requirement to control is really a functional user requirement |
| Criticality | A requirement that is decisively important for some imperative goal such as the organization's mission, or for human safety (COSMIC/IFPUG)<br>Excluded because it is a very high-level NFR that in practice would be elaborated in more detail |
| Programming language maturity | A classification of programming language maturity levels of historical development used by the ISBSG (2GL, 3GL, 4GL)<br>Excluded because the distinctions between the three classes are not well defined. |
| Quality | The degree to which a set of inherent characteristics fulfills a set of requirements.<br>(ISO 9000)<br>(See 'customer satisfaction', 'defect level')<br>Excluded because it is too general to be a non-functional requirement. |

## REFERENCES


[1] 'Guideline on Non-Functional & Project Requirements: How to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating', www.cosmic-sizing.org, to be published 2015.

[2] Software Non-functional Assessment Process (SNAP) - Assessment Practices Manual, Release 2.3, May 2015 www.ifpug.org .

[3] ISO/IEC 14143/1:2011, 'Information Technology - software measurement – functional size measurement'.

[4] ISO/IEC 24765:2010, 'Systems and Software Engineering Vocabulary'.

[5] Symons, C.R., 'Accounting for non-functional requirements in productivity measurement, benchmarking and estimating', UKSMA/COSMIC International conference on software metrics and estimating', 27/28 October 2011.

[6] ISO/IEC 25010:2011, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SquaRE) – System and software quality models.

[7] Saito, Y., Monden A., Matsumoto K., 'Evaluation of non-functional requirements in a request for proposal (RFP)', Nara Institute of Science & Technology, Japan, at IWSM-Mensura 2012.
http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6472573

[8] The International Software Benchmarking Standards Group, 'Data Collection Questionnaire(s)', www.isbsg.org.

[9] ISO/IEC 25012:2008, 'Software Engineering – Software Product Quality Requirements and Evaluation (SQuaRE) – Data Quality Model'.

[10] ISO/IEC 15504-1:2004, 'Information technology -- Process assessment -- Part 1: Concepts and vocabulary'

[11] A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition, 2013. http://www.pmi.org/PMBOK-Guide-and-Standards/pmbok-guide.aspx

[12] AUTOSAR (AUTomotive Open System ARchitecture), www.autosar.org

## APPENDIX A: CHANGE REQUEST AND COMMENT PROCEDURE

The COSMIC Measurement Practices Committee (MPC) and the IFPUG Non-Functional Standards Sizing Committee (NFSCC) are very eager to receive feedback, comments and, if needed, Change Requests for this Glossary.

All communications to the COSMIC MPC and to the IFPUG NFSSC should be sent by e-mail to **both** the following addresses:

mpc-chair@cosmic-sizing.org

nfssc@ifpug.org

**Informal general feedback and comments**

Informal comments and/or feedback concerning the Glossary, such as suggestions for general improvement, etc., should be sent by e-mail to the above addresses. COSMIC and IFPUG cannot guarantee to action such general comments.

**Formal change requests**

Where the reader of the guideline believes there is a defect in the text, a need for clarification, or that some text needs enhancing, a formal Change Request ('CR') may be submitted to both COSMIC and IFPUG. Formal CR's will be logged and acknowledged within two weeks of receipt. The outcome of the review may be that the CR will be accepted, or rejected, or 'held pending further discussion' (in the latter case, for example if there is a dependency on another CR), and the outcome will be communicated back to the Submitter as soon as practicable.

A formal CR will be accepted only if it is documented with all the following information.

- Name, position and organization of the person submitting the CR.
- Contact details for the person submitting the CR.
- Date of submission.
- General statement of the purpose of the CR (e.g. 'need to improve text…').
- Actual text that needs changing, replacing or deleting (or clear reference thereto).
- Proposed additional or replacement text.
- Full explanation of why the change is necessary.

The decision of COSMIC and IFPUG on the outcome of a CR review and, if accepted, on which future version of the Glossary the CR will be applied to, is final.