RESOLVING THE HISTORICAL CONFUSIONS ABOUT THE MEANING OF SOFTWARE SIZE AND ITS USE FOR PROJECT EFFORT ESTIMATION

Charles Symons

cr.symons@btinternet.com

IWSM MENSURA
September 15, 2023
Rome, Italy

These slides have been slightly edited from the version presented at the IWSM Mensura Conference in Rome on September 15th 2023, so that they can be read and understood without the need to listen to the oral presentation. Some Notes have also been added.

Due to shortage of time allowed for presentation, only the last two of the five 'confusion topics' listed on the next slide are elaborated in this slide-set. The five topics are actually all inter-related; for a full discussion of all the topics, please see my paper '*Resolving the historical confusions about the meaning of software size and its use for project effort estimation'.* The paper will eventually be made available for open-access as part of the IWSM-Mensura Conference proceedings on the CEUR-WS.org site.

Charles Symons, Reigate, UK, September 19th 2023

Some confusions for which I am at least partly guilty

1. Not being clear about the different *meanings* of software ‘size’
2. Studying convertibility between different *measures* of software size, where no good correlation can be expected.
3. Not making a clear distinction between ‘size-drivers’ and ‘effort-drivers’ in software project effort estimation
4. Contributing to a poor definition of ‘Functional User Requirements’ in the ISO/IEC 14143/1 standard on the principles of Functional Size Measurement
5. Drafting a misleading definition of ‘Non-Functional Requirements’ in the COSMIC/IFPUG ‘Glossary of terms for NFR and Project Requirements…’

ISO definition of Functional User Requirement (‘FUR’)

Functional user requirement

requirement that describes what the software does, in terms of tasks and services

NOTE: Subset of the user requirements. FUR relate to, but are not limited to

a) Data transfer (for example: input customer data; send control signal);
b) Data transformation (for example: calculate bank interest; derive average temperature);
c) Data storage (for example: store customer order; record ambient temperature over time);
d) Data retrieval (for example: list current employees; retrieve latest aircraft position).

User requirements that are not FUR include, but are not limited to:

a) Quality constraints (for example: usability, reliability, efficiency and portability);
b) Organizational constraints (for example: locations for operation, target hardware and compliance to standards);
c) Environmental constraints (for example: interoperability, security, privacy, and safety);
d) Implementation constraints (for example: development language, delivery schedule).

A starting ‘definition’ for Non-Functional Requirements (NFR)?

In practice, current understanding of NFR varies widely. (Some ‘definitions’ on the web)

“A software requirement that describes ***not what the software will do but how the software will do it.”***
(ISO/IEC/IEEE 24765:2010, S&SE Vocabulary**)**

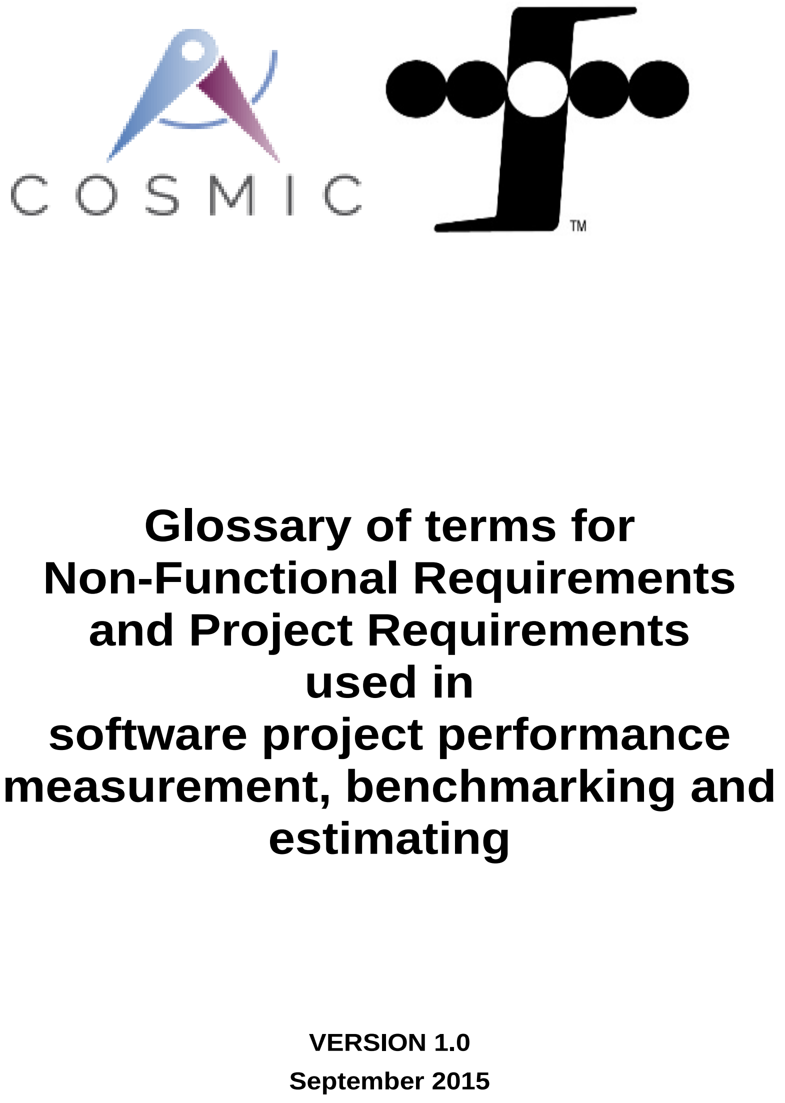
“NFRs are the ***system quality attributes ….. as distinct from the functional requirements***’
IBM’s *‘NFR Checklist’* includes *‘Availability, Performance, Security, …..* ***Globalization, Testing, Staffing***”
(IBM Cloud web-site)

“NFRs are intended to specify ‘***system qualities,’ ….. that are not directly related to their functionality***.” (© SAFe)

“(NFRs) describe ***system behaviors, attributes and constraints*** *……. There are hundreds of examples.”*
(Scrum.org)

“(*NFR) is a requirement that specifies criteria that can be used to judge* ***the operation of a system, rather than specific behaviours”***
(Wikipedia)

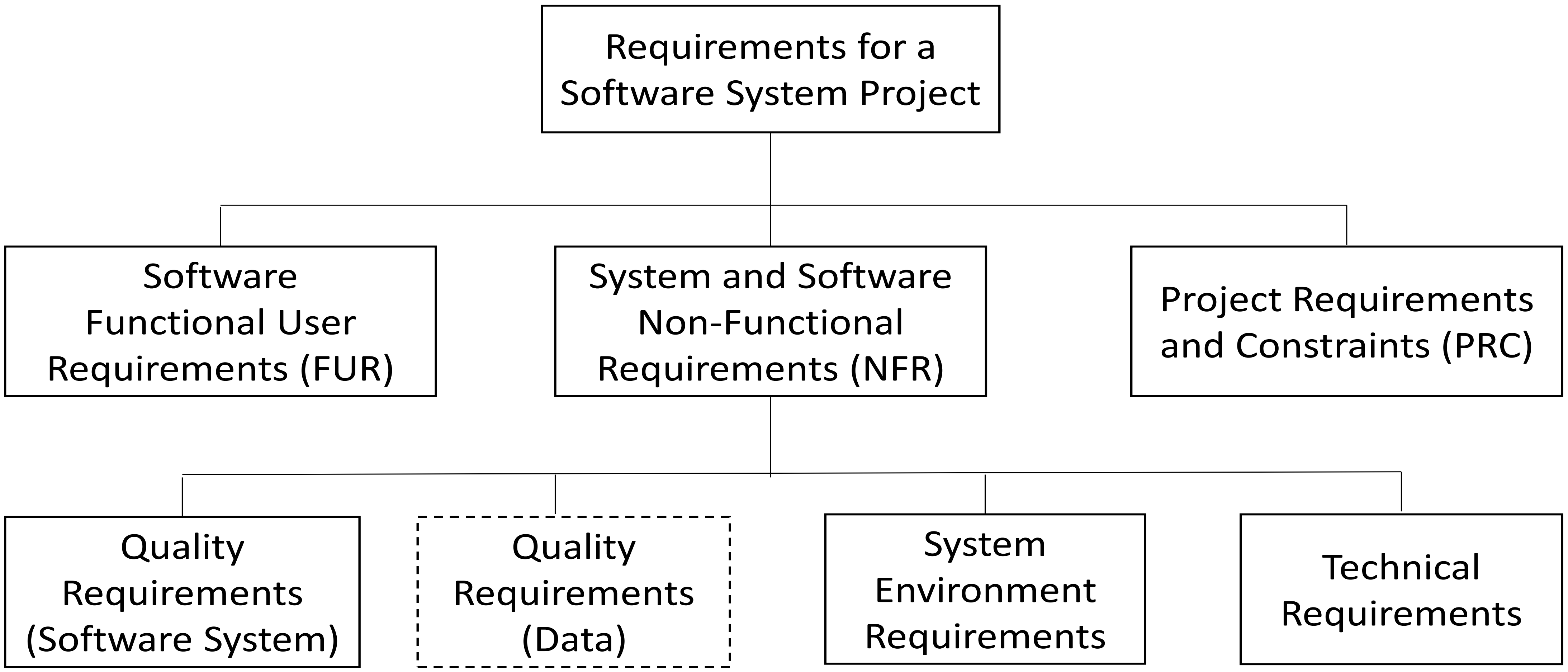
In 2015, COSMIC and IFPUG collaborated to produce a Glossary ….



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

VERSION 1.0
September 2015

We defined a model for all requirements for a software *system* (= software, hardware, etc.) project



Requirements for a Software System Project

Software Functional User Requirements (FUR)

System and Software Non-Functional Requirements (NFR)

Project Requirements and Constraints (PRC)

Quality Requirements (Software System)

Quality Requirements (Data)

System Environment Requirements

Technical Requirements

We produced a new definition for NFR

Non-Functional Requirement (NFR)
any requirement for a software-intensive 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

NOTE: NFRs concern the software system or software product:

- quality,
- the environment in which the software system or software product is implemented and which it serves,
- the processes and technology used to develop and maintain the software system or software product, and the technology used for its execution.

Conclusions from these definitions of FUR and NFR

Initial conclusions.

a) the existing ISO definition of FUR and our new definition of NFR account for *all* requirements for a software system

b) A requirement for a software system must be *either* a FUR *or* an NFR

With hindsight:

Our initial conclusion b) is wrong and the two definitions are **highly misleading!**

Why? Because most *quality requirements* result in some software functionality

Some quality reqts. are usually allocated *wholly* to ***software***:

e.g. *portability, maintainability, auditability, reusability, learnability, ease of use, inter-operability*

Quality reqts. are usually applicable to the ***system,*** but then allocated to ***software***, or to ***hardware***, or to a mixture of ***software*** and ***hardware***:

e.g. *security, response-time, availability, safety*

For performance measurement and project estimation purposes, what matters is how requirements are allocated (not how they are initially expressed, as FUR or as NFR)

Requirements for a software system can be allocated only to:

- **Software** → must add to **Software size**

or to

- **‘Non-software’** (i.e. hardware, business processes, documentation, etc.) → must add to **Project effort and/or costs**

Whether a quality requirement is expressed as a FUR or NFR, if the requirement is allocated wholly or partly to software, **the size of the software must increase**

(These size increases will be visible by size measures such as counts of SLOC or bytes.)

Function Point methods may not be able to account for all requirements allocated to software. This does NOT mean such requirements are ‘non-functional’.

Options for improving the definitions of FUR and of NFR

Option 1: Amend definitions of FUR and NFR

- Remove the references to ‘quality’ from the ISO definition of FUR and from the COSMIC/IFPUG definition of NFR
- Replace by: ‘*Quality requirements or constraints may be expressed as either functional or as non-functional requirements*.’

Option 2. In addition, redefine NFR:

‘Any requirement for a software-intensive system or for a software product that does not add to software functionality.

NOTE: Non-functional requirements concern organizational constraints (for example, numbers of implementations), the processes and technology used to develop and maintain the software system or software product, and the technical environment in which it is executed.’

Conclusion: focus on whether quality requirements are allocated to software, or to ‘non-software’ (NOT on whether they are expressed as FUR or NFR)

Process

- Determine the allocation of quality requirements to software functionality, or to ‘non-software’ activities and costs, **early** in a project’s life.

Measurement

- Use a method for measuring functional size that accounts for as many types of requirements as possible.
- Do not try to measure a separate size of NFR. (This is unwise for many reasons– see my paper*.)

* The SNAP method described in ISO/IEC CD 32430 does NOT measure a size of the NFR for software. It measures the so-called ‘non-functional size’ of software requirements – a quite different concept.

THANK YOU FOR LISTENING

Remember: you learn most from acknowledging your mistakes – read my paper for more confessions

Charles Symons

IWSM MENSURA
September 15, 2023
Rome, Italy

✉ cr.symons@btinternet.com