

# 6.5 Formally Define Non-Functional Requirements

## [Return to Recommendations](#)

[Non-functional requirements](#) are often incorrectly assumed rather than explicitly defined by users. This can lead to problems towards the end of a project as the user expectations for non-functional requirements are not met. Many times, the developers dismiss non-functional requirements as non-testable and therefore not enforceable.

This lack of specificity in non-functional requirements sets the stage for conflicts between the users, system architects, systems engineers, and developers. For example, users expect software to start and run every time it is used however, the non-functional requirement of reliability may never have been explicitly specified.

Users expect new features to be added to a system and tested before they use them. Users assume the software is maintainable without an explicit declaration for “[maintainability](#)”. In many ways, they expect it to be an unwritten requirement and or [goal](#). In other words, users expect the system to be analyzable, changeable, stable and testable<sup>1)</sup>. For example, smartphone users will switch apps to other apps if the energy consumed by the app is not efficient. Efficiency is therefore a non-functional requirement. Energy consumption may also be a [functional requirements](#) (i.e., An [application](#) can not use more than 1040 mW (milli-Watt) per

[https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.a\\_glossary:s:sms](https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.a_glossary:s:sms) message.<sup>2)</sup>.

It is recommended the Federal Reserve consider reviewing and specifying values for each of these Non-FUnctional Requirements.

### 1. [Portability](#)

- [Adaptability](#)
- [Installability](#)
- [Replaceability](#)

### 2. [Reliability](#)

- [Maturity](#)
- [Availability](#)
- [Fault Tolerance](#)
- [Recoverability](#)

### 3. [Maintainability](#)

- [Modularity](#)
- [Reusability](#)
- [Analysability](#)
- [Modifiability](#)

- Testability

#### 4. Security

- Confidentiality
- Data Integrity
- Non-Repudiation
- Authenticity
- Accountability

#### 5. Manageability

- Types of Manageability Functions
- Manageability Costs
- System Manageability Issues
- Software Manageability Issues

#### 6. Usability

- Effectiveness Metrics
- Efficiency Metrics
- Satisfaction Metrics

#### 7. Performance

- Platform Performance
- Application Performance
- Network Performance

#### 8. Interoperability

#### 9. Elasticity

#### 10. Scalability

1)

Prolifics Testiing, Achieving Requirements Testability, 10 October 2018, Accessed 10 November 2020, <https://www.prolifics-testing.com/news/achieving-requirements-testability>

2)

Sai Suren Kumar Kasireddy and Vishnuvardhan Reddy Bojja, Measurements of Energy Consumption in Mobile Applications with respect to the quality of Experience, School of Computing, Blekinge Institute of Technology, 37179 Karlskrona, Sweden, March 2012, Accessed: 10 November 2020, <https://www.diva-portal.org/smash/get/diva2:829733/FULLTEXT01.pdf>

From:

<https://www.omgwiki.org/CBDC/> - **OMG Central Bank Digital Currency (OMG-CBDC) Working Group (WG) Wiki**

Permanent link:

[https://www.omgwiki.org/CBDC/doku.php?id=cdbc:public:cdbc\\_omg:04\\_doc:90\\_recommend:50\\_recomend:start&rev=1652211624](https://www.omgwiki.org/CBDC/doku.php?id=cdbc:public:cdbc_omg:04_doc:90_recommend:50_recomend:start&rev=1652211624)

Last update: **2022/05/10 15:40**

