4.3 Non-Functional Requirements

Return to Requirements

About

Non-functional requirements are often incorrectly assumed rather than being 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[1)]. 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 SMS message. [2)].

- 4.3.1 Portability
  - 4.3.1.1 Adaptability
  - 4.3.1.2 Installability
  - 4.3.1.3 Replaceability
2. 4.3.2 Reliability
  - 4.3.2.1 Maturity
  - 4.3.2.2 Availability
  - 4.3.2.3 Fault Tolerance
  - 4.3.2.4 Recoverability
3. 4.3.3 Maintainability
  - 4.3.3.1 Modularity
  - 4.3.3.2 Reusability
  - 4.3.3.3 Analysability
  - 4.3.3.4 Modifiability
  - 4.3.3.5 Testability
4. 4.3.4 Securability
  - 4.3.4.1 Confidentiality
  - 4.3.4.2 Data Integrity
  - 4.3.4.3 Non-Repudiation
  - 4.3.4.4 Authenticity

    - 4.3.4.5 Accountability
5. 4.3.5 Manageability
    - 4.3.5.1 Types of Manageability Functions
    - 4.3.5.2 Manageability Costs
    - 4.3.5.3 System Manageability Issues
    - 4.3.5.4 Software Manageability Issues
6. 4.3.6 Usability
    - 4.3.6.1 Effectiveness Metrics
    - 4.3.6.2 Efficiency Metrics
    - 4.3.6.3 Attitude / Satisfaction Metrics
7. 4.3.7 Performance
    - 4.3.7.1 Platform Performance
    - 4.3.7.2 Application Performance
    - 4.3.7.3 Network Performance
8. 4.3.8 Interoperability
9. 4.3.9 Elasticity
10. 4.3.10 Scalability

Creating a Trade Study

Return to Top

A trade study or trade-off study helps consumers compare products on an equal footing, in other words, to help a consumer compare apples-to-apples so to speak. For example, when comparing cameras it is important to know the resolution metric of the camera. Simplistically, the higher the resolution, the better the camera. Speakers on the other hand might use the highest or lowest speaker frequency responses as a metric. Each metric is unique to the product being evaluated. However, if two cameras each have the same camera resolution, then the additional metric of frequency response might be used as a differentiator.

In the examples above, the camera resolution and frequency responses are specific to the product being evaluated and are referred to as functional requirements. However, there are also a set of requirements with corresponding metrics that capture products non-functional requirements (i.e., sometimes referred to as the "**ilities**" and include things like maintainab**ility**, portab**ility**, reliab**ility**, etc.)

Often a trade study is based on a collection of Figure of Merits with each FoM representing a single evaluation score for a single function. In the examples above, the camera resolution, or the high frequency response. An entire camera may have a FoM, but it represents the cumulative FoM of each function. A Camera may have other FoMs such as weight, battery life, size, or exchangeable lenses.

- How to Use the Boiler Plate

[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 EnergyConsumption in MobileApplications with respect toQuality of Experience, School of Computing, Blekinge Institute of Technology, 37179 Karlskrona, Sweden, March 2012, Acessed on 10 November 2020, https://www.diva-portal.org/smash/get/diva2:829733/FULLTEXT01.pdf

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:2_nonfunc&rev=1622506086

Last update: **2021/05/31 20:08**