Non-functional Requirements

Introduction

Non-functional requirements are best described by Shakespeare's definition of requirements: "necessity's sharp pinch." The non-functional requirements defined below are just as important as all the functionality you provide in your system. Why you ask? Because IT systems that process data in this interconnected global world where we work, play and consume must be maintainable, portable, scalable, and secure.

Description & Definitions

Maintainability. This sounds like fixing holes in technology is just like filling pot holes on the roadways. This is **not** what is meant regarding the non-functional requirement of maintainability. Instead, IT maintainability means creating systems that are easy to change or modify without great expense allowing the technology to preserve its value as it changes over time.

Changes in the global situation, as well as technology advances lead to frequent changes in strategies and tactics by upper management. Government regulations are another source of demands for frequent change. This means that the software systems running the business must be able to be changed or modified without great expense to adapt to the user's ever-changing needs.

Additionally, an entrepreneur's goal is to get a basic system to market as soon as possible. This means you will deploy a phased approach by getting a Minimum Viable Product (MVP) into the marketplace and continue adding functionality or features that distinguish the system from all the others.

Your maintainability goal is to plan for these strategic shifts and future phases and create an infrastructure and software architecture that can handle these dynamic demands. The clean coding with clear comments that you learned in your programming classes is relevant here, but you must think more broadly: choose tools that allow module changes, and ensure that the system is designed with enhancements in mind.

Here is an elementary example: once a system is in production, permission levels should be set by a user interacting with a data entry screen with advanced administrative privileges. No one should need to go into the code and set permission levels. Likewise web content updates for client situations in MIS 374 should not require knowledge of html or even visual studio—web content should be part of the system update process, just like such routine tasks as adding an employee.

Portability. How many electronic gadgets do you use daily? Are you able to retrieve the same information across the multiple platforms? If yes, then you know the developer planned for the non-functional portability requirement.

The most common cross-platform concern is browser compatibility. What looks good in Internet Explorer doesn't always look good in Firefox or Chrome. Access by mobile phone is an increasingly common portability requirement for systems that depend on high user volume for profitability. Global companies sometimes require a special process for systems that are critical to operations in remote locations with undependable internet access—a problem you may encounter professionally, but not for MIS 374 clients.

Consider these portability issues. Someone in South Korea is trying to access a North American website but there is no translation software. How much business is the North American company losing by failing to plan for the non-functional portability requirement?

Someone in Saudi Arabia is using an Arabic keyboard to access a North American site but the IT specialist did not plan for translation of non-English keys. The North American business will not be making any sales to potential customers who happen to be using non-English keyboards. Worse, the strange characters might cause a system crash if the system does not handle non-standard characters.

Our globalized economy makes it imperative for the IT specialist to take into account the non-functional portability requirements of the system.

Scalability. Imagine you launched your web site yesterday morning and this afternoon you go to Google analytics and read that you had 5 views and two sales. The following day you see that you had 50 views and 25 sales. Wow! And the next day your fantasy comes true and your web site gets one million hits and then it crashes hard. If only you had considered the non-functional requirement of scalability.

Also, as the organization grows, internal systems must be ready and able to handle the increased usage. Instead of implementing a recovery plan for a crash, implement a non-functional scalability requirement plan early, enabling growth and prosperity.

Security. Non-functional security requirements are dependent on production requirement concerns regarding fraud and theft of money, identity, and ideas. The tiered permission levels required for your MIS 333k project are access level security measures. For production systems you need to consider company policies on maintaining permissions, as well as checks and balances for any functionality involving money.

Purpose

How your system performs is just as important as how it functions. Creating a Non-functional Requirements Summary will ensure a maintainable, portable, scalable, and secure system providing a high-quality user experience and a low total cost of ownership (TCO).

Creating a Non-functional Requirements Summary

To create your Non-functional Requirements Summary start with the Non-functional Requirements template found on the Resources page. Replace the definition of each category of non-functional requirements with a sentence or two summarizing the relevant facts about the context for your project for that category. Fill in the non-functional requirements for each area. Then fill in the corresponding priority levels. Add more rows for additional requirements as necessary. For an example see the Non-functional Requirements Summary in Figure 1 for the Austin Waldorf School (AWS).

Benefits

Your end goal is to develop a system that has all the functionality agreed upon and includes planning for non-functional requirements including maintainability, portability, scalability and security. Ultimately, you take into account your client's current situation while simultaneously planning for their future technology needs. By looking at the non-functional requirement summary examples posted on the Resources page you can see that these requirements are dependent on the firm's situation. The non-functional requirements for creating a system for the non-profit, Latinitas, with only a volunteer "staff" are similar to the Austin Waldorf School with full time employees, who still want as little time spent maintaining a web site as possible, but are different from Hillel that has hired tech support. The goal is always to match your non-functional requirements with your client's situation.

Compare the Austin Waldorf School summary (AWS) in Figure 1 and Texas Hillel Foundation NFR summary on the Resources page to notice the differences between a public website seeking maximum visibility to increase enrollment and donations (ASW) with an internal system designed to be accessed by a limited number of users all co-located in a building near where their systems are stored on a computer service company's well-protected servers (Hillel).

Later in the semester we will cover strategies for improving the maintainability of systems. For now, focus on MIS 374 examples, your background in MIS classes, internships, and your own experience of using systems to complete the non-functional requirements analyses that are part of your Group Project 2 and your Project Charter for your client project.

Figure 1. Non-functional Requirements Summary for the Austin Waldorf School (AWS)

Non-functional Requirements for Austin Waldorf School (AWS): Maintainability, Portability, Scalability, Security

Maintainability – AWS staff has little technical knowledge and needs to have a website that is easy to maintain and requires minimal updates outside content.

Requirement	Priority
System must be easy to maintain and update after project team leaves	High
Training should be conducted to ensure staff understands how to use the CMS (Content Management System) and is able to maintain the system internally	High
Provide full documentation that AWS staff can use if internal maintenance or updating is required; create a quick reference guide to troubleshoot common issues.	High
Customize CMS (Content Management System) to meet AWS needs so that little internal customization is needed after the project team leaves	High

Portability – The AWS website will be viewed from several different platforms and browsers, so compatibility is crucial. While mobile use may be considered in the future with the portals, it is not currently a priority.

Requirement	Priority
Check cross-browser compatibility (IE 7, IE 8, Firefox, Google Chrome, Safari)	High
Downloadable content must be compatible with different platforms (Windows, Mac, Linux)	High

Scalability – AWS has relatively constant staff numbers, so internally the system does not need to handle significant growth in the foreseeable future. AWS hopes to expand its student population and its alumni base grows each year, so the Content Management System (CMS) can handle growing data storage and servers can handle increasing traffic.

Requirement	Priority
Ensure that CMS provides database that has the capacity for the potential growth	High
Check capacity requirements with host to ensure server scalability	Medium

Security – A multi-level access system needs to be in place so only admins have full back-end access. Also, AWS has several portals and a donation system, so password logins and a secure credit card or online payment system is necessary.

Requirement	Priority
Multi-level access for AWS staff (admin, teacher, etc.)	High
Password login for each portal (parent, alumni, etc.)	High
Secure online payment and credit card systems for donations. (May post pone this until another semester—AWS would then continue to handle donations like they do now).	Medium

High Quality Delivery Tips

- Clearly outline maintainability, portability, scalability, and security requirements.
- Requirements under each category describe the environment that must be in place for the system to perform the non-functional requirements optimally.
- Ensure that the context issues for each of the sections are consistent with the details in the case or your client project situation.
- Make sure that these requirements are consistent with the focus of your strategy discussion and planning documents.

Template and Examples on Resources Page:

- Non-functional Requirements Template
- Non-functional Requirements – Waldorf
- Non-functional Requirements – Hillel