California State University Long Beach
Department of Computer Engineering and Computer Science

Course # CECS 542

Professor: Birgit Penzenstadler

Semester: Spring 2017

Nonfunctional Requirements

Team members:

Chris Cervola

Stephanie Lopez

Date: 04/17/17

Lab Write Up

To complete this assignment, we had to analyze the system as a whole and then compartmentalize the nonfunctional requirements into quality characteristic / system constraint / development process / deployment requirements. This lab was pretty straight forward as we were further decomposing our goal model and finding the related documentation within the OpenMRS system.

The most challenging part of this assignment was determining the Development Process Requirements. Depending on the phrasing that is used the NFR can either fall into the Development Process Requirements section or the System Constraints section. Reshaping the language as more of a guide that needs to be followed versus a requirement that the system had to follow was the main difference.

Type of NFR and quality characteristic	Development process requirement - Consistency
NFR description	Releasing a new module or coding changes shall follow the OpenMRS Platform Release Process to ensure all work products adhere to the same standards set forth by the Release Process Checklist
Rationale	This goal is derived from two business goals: Open Source and Effective/Secure Collaboration. With the product being open source, developer teams must be able to collaborate efficiently when developing the system.
Satisfaction criterion	The stakeholder will be satisfied when every required step of the Release Process Checklist has been completed.
Measurement	The QA engineer will need to review the Release Process Checklist once it has been completed to verify each item on the checklist was completed successfully, not merely checked off to move onto the next step
Risk	If a standard release process is not followed, modules and code changes can become confusing because everyone can potentially release code however they see fit. The Platform Release Process ensures a standard template that needs to be adhered to in order to ensure consistency across OpenMRS
Compliance in OpenMRS	OpenMRS provides a guide for their Platform Release Process

Type of NFR and quality characteristic	Development process requirement - Documentation
NFR description	The code shall follow standard procedure and have proper documentation; classes shall be documented and code shall contain comments. As new classes are created and tested, accompanying documentation shall be created to explain the purpose, assumptions, and business rules that were used in designing the classes.
Rationale	This requirement is derived from Effective Collaboration among developer and users. Documentation will allow future developers to have the tools needed to create updates or enhancements to existing features in the system.
Satisfaction criterion	The stakeholders will be satisfied once every class has accompanying documentation as well as each class containing

	some comments within the code.
Measurement	Each tester will need to verify that each class they are testing contains appropriate documentation before determining a test has passed. In addition, they will need to scan through the source code to determine if there are comments throughout the code.
Risk	If this requirement is not met, new developers might not understand why a class was written a certain way and make a lot of “improvement” which violate existing business rules or assumptions that were never documented, yet vital to the system’s success.
Compliance in OpenMRS	Each module has minimally a description document. https://wiki.openmrs.org/display/docs/Home

Type of NFR and quality characteristic	Development process requirement - Auditability
NFR description	OpenMRS classes should follow the coding templates and procedures for writing to an audit trail. Therefore, when a failure occurs, the failure is logged into an audit trail. More specifically, an audit log (file). This file will be used to help developers understand system failures and modify the classes to prevent further failures
Rationale	This derives from Effective Collaboration and having Cross Clinic Data Sharing. An audit trail will be useful in analysis and make cross clinic collaboration more effective.
Satisfaction criterion	The test engineers need to determine at least one edge case testing requirements to try and generate an error while testing each feature. If an error occurs, they should be able to view the error in the respective log file for each feature.
Measurement	The metric will be assessed based on the number of failures the test engineers are able to generate versus the number of entries written to the log file. There should be a 1 to 1 relationship between errors occurring and a new log being appended to the Log File
Risk	Without a Log File, the developers might not be able to track down and debug errors that are occurring. As the system becomes more complex, identifying the source of errors becomes increasingly difficult. Log files will at least point the developers in the general direction of what is causing the failure.
Compliance in OpenMRS	OpenMRS has a Log Manager Module that enables all modules to implement audit logging,

Type of NFR and quality characteristic	Deployment requirement - Software Generation
NFR description	The user shall be able to access the latest code out of a repository and compile into exe to install
Rationale	This derives from usage goal Easy to Use in which the users should have no programming knowledge to install and use the system.
Satisfaction criterion	The stakeholder will be satisfied when the latest release is available to the user in a compiled exe format to install
Measurement	Quality assurance will routinely check once a month to ensure that the necessary format is available to the user.
Risk	The user is not able to download and use OpenMRS.
Compliance in OpenMRS	OpenMRS has a deployment guide that contains a link to the latest stable version of the software in executable form

Type of NFR and quality characteristic	Deployment requirement - Documentation
NFR description	The software needs to be accompanied with Documentation detailing how to install the OpenMRS software. Simply giving the user a download link to the source code will not be enough to ensure the user is able to download and install the software correctly
Rationale	This derives from usage goal Easy to Use in which the users should have no programming knowledge to install and use the system.
Satisfaction criterion	The stakeholder will be satisfied once a manual is created detailing how to install OpenMRS software for multiple environments (single user, server, cross site)
Measurement	A test engineer will need a test computer, server, and simulated cross site location where they will need to follow the installation guide. The test engineer should be able to go through the documentation and setup each location successfully without needing any additional information than what is provided in the guide.
Risk	If documentation does not exist, or the documentation is insufficient, users looking for Medical Software will look for solutions that have better guides/support. This means OpenMRS may never achieve widespread adoption

Compliance in OpenMRS	OpenMRS has a deployment guide that contains the necessary document to ensure each user can configure their computer to install and run OpenMRS

Type of NFR and quality characteristic	Deployment requirement - Software Requirements
NFR description	Certain software (Firefox, Java, Tomcat) must be installed prior to deploying OpenMRS.
Rationale	In order to accomplish usage goal “Access Medical History,” the user must be able to have the correct software tools to access the data.
Satisfaction criterion	OpenMRS can be deployed successfully once all required software tools are installed.
Measurement	The test engineer will install all required software listed in the Deployment guide. From there, they will follow the rest of the guide to install OpenMRS. They should be able to successfully install OpenMRS.
Risk	OpenMRS will not deploy successfully.
Compliance in OpenMRS	OpenMRS has a deployment guide that contains the necessary document to ensure each user can configure their computer to install and run OpenMRS

Type of NFR and quality characteristic	System Requirement - Portability
NFR description	OpenMRS is an open source solution that is hoping to bring a medical system to impoverished areas of the world. To ensure widespread adoption, the system should be able to work with limited hardware resources.
Rationale	The goal this NFR is derived from is that the software should be available in developing countries.
Satisfaction criterion	The stakeholder will be satisfied when the software is able to run on hardware as limited as 1 GHz processor or better, 256 MB of memory or more, 40 GB hard drive or larger.
Measurement	A Test engineer will need to create a virtual machine or find actual hardware with the above hardware components. The tester will then need to install OpenMRS and begin testing the software to ensure all features still work correctly.
Risk	If the software is not able to run on the specified hardware, it may never gain adoption is impoverished areas. Impoverished areas do not have the luxury of cutting edge hardware, therefore, if the software cannot run on limited, older hardware, it will not be available in impoverished areas.
Compliance in OpenMRS	OpenMRS has a system requirements page detailing the minimum hardware requirements to use the software

Type of NFR and quality characteristic	System Requirement - Scalability
NFR description	Based on the hardware being used, the software should be able to function reliably adding x number of users, corresponding to their hardware capabilities.
Rationale	The goal this NFR is derived from is that the software should be available in developing countries.
Satisfaction criterion	Depending on the hardware being tested, the software should be able to perform reliability and without a noticeable negative performance impact as x number of users get added. X being the number referenced at the different hardware levels, 100s, 10,000, 250,000 patients.
Measurement	A test engineer will need to secure a computer, or create a virtual machine with the 3 different levels of minimum requirements. On each computer, the tester will need to install OpenMRS and add tens, hundreds, or thousands of users, based on the requirements

	outlined in the link below. The tester should then test the software and ensure the performance of the tool is not negatively impacted by the new users that were added. This can be measured by ensuring each page loads within 3 seconds or less before and after the creation of the new users.
Risk	If the software cannot support multiple users with limited hardware the software will be of no use to impoverished areas. These areas can have all sizes of populations and the software needs to be able to accommodate all patients that are seeking help. As new users get added, if the software becomes increasingly slow, then the software will become more of a burden to the clinic than its worth.
Compliance in OpenMRS	OpenMRS has a system requirements page detailing the minimum hardware requirements and how many users each configuration can support

Type of NFR and quality characteristic	System Requirement - Flexibility
NFR description	Each clinic will can potentially have different specialties. This means, the software needed for one clinic can be drastically different than the software needed for another clinic. To ensure the software is useful for all specialties, custom modules can be installed/uninstalled based on each sites need.
Rationale	This NFR is derived from the goal stating that the system should be customizable.
Satisfaction criterion	The stakeholders will be satisfied once OpenMRS has a core set of features that is incorporated into every OpenMRS build as well as a list of additional modules that can be installed to depending on each clinic's specialty.
Measurement	A test engineer will need to secure a computer and install OpenMRS. From there, they will need to identify at least two modules to uninstall. They will test the features by clicking on the respective icons and ensuring the features work. They will then uninstall the modules and try to retest the same features. The features and functionality should no longer exist. Then they will need to identify two modules they wish to enable. Try and use the features described by each module before installing the module, which will fail. Then install the modules and test their features, at this point they should be successful.
Risk	If OpenMRS forces their vision of the software upon all users, users may find the software is not flexible enough for their needs and will look to another solution. This means OpenMRS will not gain adoption and may eventually disappear.

Compliance in OpenMRS	OpenMRS contains a list of modules that can be installed to customize each clinics version of OpenMRS to suit their needs

Type of NFR and quality characteristic	Quality characteristic - Performance
NFR description	Users of OpenMRS should be able to enter data into any form within OpenMRS and be able to save their inputs in less than 1 second.
Rationale	This NFR is derived from the system goal, Standardized Data Entry
Satisfaction criterion	Pages will need to be able to save data in less than 1 second
Measurement	A test engineer will need to identify all websites that contain data entry forms. They will need to populate all required form fields and click the Save button. They can use the built in timing functions in the browser to record the time it takes to save the data after clicking Save.
Risk	If this requirement is not met, the risk to OpenMRS will depend on how long the save time actually is. The higher the save time, the more likely users will look to additional solutions. The tool should make its user’s life easier, if the user has to wait for a long period of time, it can be seen as more negative than positive.
Compliance in OpenMRS	When using the demo, each click loads the page in less than a second (even when inputting data).

Type of NFR and quality characteristic	Quality characteristic - Availability
NFR description	OpenMRS server should be up 90% of the time
Rationale	This NFR is derived from the usage goal, Access to Medical History
Satisfaction criterion	The measurement for this goal should result in 90% or higher in order to satisfy the stakeholders.
Measurement	At the end of each quarter, the quality engineer will need to calculate the number of business hours that OpenMRS should have been available for. They will then determine the amount of down time OpenMRS experienced by viewing the Logs that track when OpenMRS goes down and for how long. The final calculation will be the time OpenMRS was actually available for

	divided by the time OpenMRS was supposed to be available for.
Risk	If this requirement is not met, users will definitely be looking for a new software solution. If the software is down more than 10% of the time during normal operating hours it either means the clinic can not help any patients or the notes about the visits during that time will be lost; both options are unacceptable.
Compliance in OpenMRS	OpenMRS contains a module that helps Expose System Metrics for Monitoring which is to ensure maximum uptime and performance of the system

Type of NFR and quality characteristic	Quality characteristic - Usability
NFR description	OpenMRS target audience is users who do not have any programming knowledge. This means the User Interface should be very straightforward and easy to use.
Rationale	This goal is derived from the Easy to Use and No Programming Necessary usage goals
Satisfaction criterion	User should be able to retrieve patient record in less than 5 clicks
Measurement	The test user will need to launch the OpenMRS software and login. From there they will need to search for a patient and click on them to open their record. The tester will need to record each click while they navigate from the home page to the search screen, input the user's name, click search, and then open the user's profile.
Risk	If the software is not intuitive and easy to use, users will have a very unsatisfying time using the software. It will prevent them from exploring other features, both core and add on, that they could use to improve their clinic. If basic functionality is burdensome they may try to create their own shortcuts by entering data in the wrong fields just because it is easier.
Compliance in OpenMRS	When using the demo, we were able to retrieve a patient's record in less than 5 clicks. In addition, OpenMRS also has some limited User Training to ensure the system in usable.

Type of NFR and quality characteristic	Quality characteristic - Efficiency
NFR description	OpenMRS should provide efficiency for the user, especially if they are currently keeping paper records. The advantage of using the

	software is faster data entry, being able to link all of a patient's records together, and being able to share data with other clinics that treat the client.
Rationale	This NFR is derived from the Cross Clinic Sharing goal as well as the Accept All Types of Data and Attachments goal.
Satisfaction criterion	The stakeholder will be satisfied when they can confirm all types of patient data (x-rays, vitals, dr notes, etc) can be linked to a patient account as well as being able to share this data with other clinics using OpenMRS.
Measurement	The test engineer will need to create a fake user account. They will then proceed to fill out the user's profile, attempt to add attachments to the record, attempt to add vitals, attempt to add dr notes about the visit. From there they will need to share the patient's record with another installation of OpenMRS. They will record whether or not they are able to successfully modify and share the record with another clinic.
Risk	If OpenMRS does not all the clinic staff to attach all relevant visit information it means the software is not suitable for their needs and will not be used. Sharing the patient data can be treated as more of a convenience, if they are able to attach all data successfully. A failure is this category alone would not be detrimental to the adoption of the software.
Compliance in OpenMRS	OpenMRS has an Operational Theatre Module that is used for managing and scheduling critical tasks in the clinic. The module was written to help clinics manage their time and resources efficiently

Type of NFR and quality characteristic	Quality characteristic - Reliability
NFR description	The user shall receive the same response every time a new record is created regardless of how many user records the system contains and during periods when there is a lot of traffic on the system.
Rationale	This derives from usage goal Easy to Use in which the users should have no programming knowledge and all features shall be standardized to allow for easy understandability.
Satisfaction criterion	The stakeholder is satisfied when the user receives consistent responses while there are at least 10,000 patient records and a minimum of 50 users are logged in.
Measurement	The test engineer will create 10,000 records via a script, then will

	proceed to simulate at least 50 active users. They will go into the patient record screen and attempt to add vitals to a user's profile. Each time they edit a record, the system should save the data successfully.
Risk	Inconsistent responses as the number of users increases will cause confusion and frustration among the user. Ultimately it will make users less likely to trust and use the system
Compliance in OpenMRS	OpenMRS has a System Performance and Utilization Module that is used to monitor reliability and the impact of the electronic medical records system

Type of NFR and quality characteristic	Quality characteristic - Security
NFR description	All patient and clinic data shall be able to prevent brute force attacks and also store data in an encrypted format to prevent unauthorized users from accessing patient records
Rationale	This requirements is derived from the usage goal, Secure Data
Satisfaction criterion	The stakeholder will be satisfied when the system can withstand brute force attacks for a minimum of 8 hours. If the system is breached, the data that is retrievable should be stored in an encrypted format that is unreadable without being decrypted.
Measurement	The test engineers will run software that is typically used in Brute Force attacks to try and gain access to the system. The test engineer will let the software run for 8 hours before determining if the security measures prevented brute force attacks or not. The second part of the test would be to run the software until it is able to gain access to the system, the user will then download patient records from OpenMRS. The downloaded data should be encrypted and not be human readable.
Risk	Hackers could easily gain crucial patient and clinic information including but not limited to patient medical records, address, and clinic procedures. If this happens, the user base will lose confidence in the software and look for a new solution, even one that is less robust, if it means their data is secure.
Compliance in OpenMRS	OpenMRS has a Security and Encryption page that details the different encryption methods needed to secure OpenMRS.