|  |  |
| --- | --- |
|  | **BAHRIA UNIVERSITY,**  **(Karachi Campus)**  *Department of Software Engineering*  **Assignment#01– Fall 2021** |

|  |  |  |  |
| --- | --- | --- | --- |
| COURSE TITLE: | **SRE** | COURSE CODE: | **SEN-211** |
| Class: | **BSE 3B** | Shift: | **Morning** |
| Course Instructor: | **ENGR. BUSHRA FAZAL KHAN** | Assignment Date: | **01-Nov-2021** |
| Max. Marks: | **5 Points: CLO 1** | Assignment Due: | **08-Nov-2021** |

Question 1) For each type of Non-Functional Requirements give two example each and identify Goal, Objective, and Quantitative verifiable metric where possible.

**Reliability**

**Ethical**

**External**

**Implementation**

**Organizational**

**Safety**

**Legislative**

**Privacy**

**Interoperability**

**Delivery**

**Non-Functional Requirement**

**Standards requirements**

**Portability**

**Space**

**Product**

**Efficiency**

**Performance**

**Usability**

**Product**

**Usability**

Usability is the value of a user interface to users. This means that an interface allows users to achieve their goals efficiently in a pleasing environment that doesn’t induce stress or boredom. The following are the examples of usability.

**Examples**

**Error Tolerant**

A system that can correct errors as opposed to giving up and displaying an error message.

**Functionality**

Interfaces may be judged by obvious factors such as look-and-feel. In the long term, they are judged by the power of their functionality.

**Navigability**

It is easy to locate what you are looking for with tools such as navigation functions and search.

**Goals**

* The user must feel satisfied with the system, The user must understand what the system does.
* The goal is to make it easy for you to fill out a form by a user of any age.

**Objectives**

Usability objectives or requirements are typically driven by either safety concerns or business goals. Safety related objectives should originate from the risk management process and are designed to show that risk mitigations are effective. Business related objectives usually come from market research – what the customers are looking for and what the competition currently offers.

The most effective usability objectives contain following elements

* Task – including a quantitative goal so that success can be easily defined, usually in terms of time, accuracy, task completion rate, etc. The highest risk tasks and the most important tasks for using the device effectively should be included when defining tasks.
* User profile – specific user group that will perform the task.
* Use environment – context of device uses for a particular task, including physical and social environmental factors. Use environment is not always specifically called out in the usability objective but is often implied. It should, at a minimum, be defined for the device in the usability documentation and considered when developing usability objectives.

**Quantitative Verifiable**

No of users of an application, reviews.   
This NFR can be measured by how many errors customers are making on that shape, if the errors are not unusual and there's a pattern amongst the users’ statistics to mistake ratio, that means we have didn't attain this non-purposeful requirement

**Efficiency**

Efficiency is the extent to which the software system handles capacity, through out, and response time. Efficiency requirements address the user concern for how fast the system functions, how efficiently the system takes in inputs and processes outputs, and how much can be processed at a time. When eliciting efficiency requirements, consider needs regarding response time, throughput, process capacity, and storage capacity.

**Performance**

Defines how fast a software system responds to certain user’s actions under certain workload. Metric explains how much a user must wait before the target operation happens (the page renders, a transaction is processed, etc.) given the overall number of users now. But it’s not always like that. Performance requirements may describe background processes invisible to users, e.g., backup. But let’s focus on user-centric performance.

**Examples**

* The Website’s load time should not be more than one second for users.
* Google is very sensitive about desktop and mobile speed load times. The search engine considers multiple scenarios, including the type of connection, mobile or desktop load, and type of content that gets displayed. Based on the sum of factors, it suggests different performance scores that you can estimate for your website. This is especially important if you set up the requirements for landing pages, as Google might rank your page lower in consideration of its speed.
* The landing page supporting 5 thousand users per hour must provide 6 seconds or less response time in a Chrome desktop browser, including the rendering of text and images, over an LTE connection.

**Space**

The memory system consumes for its operation. Store the area and costs of your immediate needs, or your forecast needs.

**Examples**

* A function that performs the execution and processing of data to the space vehicle control system must strictly comply with their execution time.
* The system should be able to operate more faster than the existing system.

**Goals**

* While improving efficiency means doing things faster, using fewer resources, in fewer steps.
* As a business, increasing the efficiency of your systems is key in ensuring you get the maximum return on your efforts. Every system you employ should focus on having the best efficiency and productivity standards while enhancing profitability. Truthfully, your businesses processes should work to your benefit in saving time and money – not the opposite.

**Objectives**

* The key to having good all-round performance is five performance objectives: quality, speed, dependability, flexibility, and cost.
* Efficiency goals are targets to achieve more output for each unit of input. This has many variations including goals for productivity, time management, economic efficiency, and resource efficiency.
* Specific – Clearly identify and define what the objective is.
* Measurable – Is the objective quantifiable? (Response time is an example of a measurement that can be used).
* Achievable – Make sure the objective can be met.
* Realistic – Ensure the objective is realistic. (a practical scenario which is true to life).
* Time-related – Specify the time for the objective i.e., “peak hour” of system load.

**Reliability** Technology that is highly reliable functions with the same or similar efficiency after extensive use. Here are three ways you can assess a device's reliability:

**Percentage of the probability of failure** You can check the percentage of the probability of failure, or failure rate, to determine the reliability of a system. If the percentage is higher, the system is likely to function normally after substantial use.

**Number of critical failures**  Consider recording the number of critical failures a system experiences during testing to check its reliability. If the number of failures is low, it means that the system operates properly.

**Time between critical failures** Tracking the time between critical failures can help you understand the reliability of a system. When critical failures occur rarely, it means that a system functions normally most of the time.

**Examples**

* The Automated Teller Machine (ATM) probability of failure on demand shall be 0.001 (1 out of 1000) when reading the magnetic stripe data on an undamaged card.
* The rate of failure occurrence per ATM shall be 1/1000 (1 occurrence in 1000 days). Failure means the ATM fails to operate with any card inserted, and the software must be restarted to correct the failure.
* The mean time to failure of the ATM timing out due to user inactivity shall be 1/1000 (1 occurrence in 1000 transactions). Failure means the ATM must cancel the transaction and must allow the user to start over.
* The account update process shall roll back all related updates when any update fails to commit. e. The authorization transaction match process shall require a 100‐percent match to post a transaction.
* The data transmission process shall confirm the receiving terminal is in a ready state prior to the start of transmission.
* The point‐of‐sale terminal shall have an MTTF of 1/10,000 transactions in an ROCOF of 1/30 days. Failure is defined as an invalid transaction presented for processing.

**Quantitative Verifiable**

As you may have guessed, it’s tricky to define critical failure, time, and normal usage conditions. Another, somewhat simpler approach to that metric is to count the number of critical bugs found in production for some period or calculate a mean time to failure. Three ways to measure it are:

* Probability percentage, time.
* The number of critical failures, time; and
* Mean time between failures.

**Objectives**

* Establish clearly stated, measurable, and meaningful reliability goals that can be converted into reliability activities in a reliability program and drive the appropriate behavior across the product team.
* To apply engineering knowledge and specialist techniques to prevent or to reduce the likelihood or frequency of failures. To identify and correct the causes of failures that occur despite the efforts to prevent them.

**Goals**

Before we can start choosing the individual reliability elements, we must set reliability goals. After the goals are set, we determine our level of capability. Reliability goals can be derived from customer-specified or implied requirements, internally specified or self-imposed requirements.

**Customer-Specified or Implied Requirements** For Customer-Specified or Implied Requirements, we review customer specifications and determine the different reliability metrics. Often, the metrics will not be called out specifically, but rather they will have to be inferred.

**Internally Specified or Self-Imposed Requirements** Internally Specified or Self-Imposed Requirements are usually based on trying to be better than previous products. The process involves interviewing key members of various departments and at contract manufacturing partners to find out what they have set as internal goals. These goals may need to be adjusted as information is gathered, but this represents a good starting point.

**Portability**

Portability means how effectively a system performs in one environment compared to another.

OR

The ease with which a software system can be transferred from its current hardware or software environment to another.

**Examples**

* A user might purchase a new cell phone model and download a mobile application they had on their last device. If the application runs as efficiently on the new phone as it did on the old phone, then it's highly portable. As a developer, you can design your applications to function properly on multiple devices to improve portability.
* The Home Accounting software may be ported to any personal computer or workstation environment supporting at least 16‐bit color on a 15-inch display monitor, achieving a SPECfp95 benchmark rating of at least 5.0, and having a data storage capacity of at least 8 MB.

**Objectives**

* Partially validate the system (i.e., to determine if it fulfills its portability requirements.
* Determine if the system can be ported to each of its required environments:
  + - Hardware ram and disk space.
    - Hardware processor and processor speed.
    - Monitor resolution.
    - Operating system make and version.
    - Browser make and version.
* Determine if the look and feel of the webpages is similar and functional in the various browser types and their versions.
* Cause failures concerning the portability requirements that help identify [defects](http://tryqa.com/what-is-defect-or-bugs-or-faults-in-software-testing/) that are not efficiently found during unit and integration testing.

**Goals**

* The goal of portability may be limited by conflicting goals such as high performance or exploiting the features of a specific environment.
* Development costs and timetables will be affected by this choice. There is no commonly accepted method for quantifying portability, but the Degree of Portability discussed above offers one possible approach

**Quantitative Verifiable**

We can affirm whether we have executed this NFR simply through strolling the app on previous variations of iOS and iPad and notice if it plays the favored functionalities.

**Organizational**

**Standard Requirements**

Quality standards are defined as documents that provide requirements, specifications, guidelines, or characteristics that can be used consistently to ensure that materials, products, processes, and services are fit for their purpose. The software should be developed using standards specified by the ISO and IEEE standards.

**Examples**

* A bank is understood for their confidentiality, in order that they need the data in their customers to be surely exclusive.
* The device shall not use any of the databases said inside the record CFDS-982.

**Objectives**

* Our goal must be to choose comfy methods to application wherein encapsulation and facts hiding is used, there ought to be a couple of authorization checkpoints via-out our software. We must additionally choose a server that has the least records leaks in line with year so that the information can reside on our servers appropriately.
* Our objective should be to select the next best database that works with our system requirements and abide by the client’s standards of not using the provided databases.

**Goals**

* The goal is to make software surely comfortable with utmost priority given to the facts.
* The aim is to no longer use any of the databases which might be cited within the said record.

**Quantitative Verifiable**

We can measure this NFR by way of simply checking the record and verifying whether we've used the perfect databases.

**Implementation**

It is a process that concerns both technical implementation of new technology and a slower process of organizational members adopting the technology in use and aligning it to their work tasks, probably transforming routine work practices to fit the new situation.

**Examples**

* Arena should run on any Unix OS e.g., MacOS Linux and Solaris.
* All users should be able to access with a web browser supporting cookies, Administration function used by operator are not available through the web.
* The system should seamlessly fit into company’s previous suite of apps and take advantage of their already existing Oracle database.
* The gadget must be written in. internet language due to the fact the patron already owns a license to. internet languages.

**Goals**

* The goal is to apply the license that the consumer already has, and code in a .net language.
* The aim is to make a gadget that suits proper in with employer’s present packages and makes use of their present Oracle databases.

**Objectives**

* To achieve the intention, we set our goal that the appearance and experience of our application should be much like that of customer’s previous apps.
* Select the best .NET language that suites our needs and fulfills our requirements.

**Quantitative Verifiable**

To verify this requirement, we simply look at that the language we used is classified as a. internet language or now not.

**Delivery**

Software delivery is the entire process of getting a software product to customers, from conceptualization, through development, and ending in the actual purchase and installation of the product's license.

**Examples**

* A game should be deployed on the iOS App Store as well as the Google Play Store on the date mentioned in the document ASW-6R5.
* The Application or Custom Software deliver on time as mentioned in the documentations.
* An online shopping app should be delivered product on limited time.

**Goals**

* The goal is to deliver the app on iOS and google stores on the day of release.
* The goal is to deliver a program before Eid with a special Eid theme that disappears after Eid.

**Objectives**

* The goal needs to be to supply the software program earlier than Eid days with a unique Eid themed interface that reverts to the unique theme after Eid days
* The objective must be to finish all the pre-requisites of uploading an app to any of these structures.

**Quantitative Verifiable**

To affirm this requirement, we just test whether we will upload the app to the platforms.

**Interoperability**

In the sea of non-functional requirements, interoperability is defined as how easily a system can share information and exchange data with other systems and external hardware.

**Examples**

* A taxi service software that makes use of google maps to retrieve site visitors’ facts and trip routing.
* A photo enhancing software which can import documents from every other photograph composing app from the same client with all its authentic shade facts as well as the log of all the changes implemented.

**Goals**

* The aim is we want to import a photo from one program to another with all of the unique color statistics and all the backlogs of the changes made to that picture.
* The goal of our consumer is to use google maps for site visitors’ data and top-rated ride routing.

**Objectives**

* Our objective is to implement google maps’ API in our source code in a way that it doesn’t appreciably change our system view, or our machine shape interferes with the API.
* To attain the goal, we set our objective to apply the identical technique for logging coloration statistics and backlogs in both the apps, so the apps can communicate with each other.

**Quantitative Verifiable**

To confirm this NFR we test whether our import has all the unique color statistics from our unique photo and whether the backlogs are legitimate or now not.

**Ethical**

Developer Ethics describes the field of ethics when applied to the behavior of software developers. According to The Internet Encyclopedia of Philosophy, the field of ethics (or moral philosophy) involves **“systematizing, defending, and recommending concepts of right and wrong.”**

**Examples**

* An aggressive recreation that has guidelines that a participant must acquire a pressure ban if a malicious dishonest software is detected jogging at the same time as the game is strolling.
* A user has no proper to mood with the supply code of a financial institution application and if observed tempering his/her account and device need to be flagged for in addition research.

**Goals**

* The purpose of this requirement is to punish cheaters in a recreation.
* To defend the account holder from malicious activities.

**Objectives**

* To obtain this objective, we check the source code at regular durations, if the code is changed, we flag the device as suspicious and additionally flag the account.
* To attain this aim we can make our goal to detect what software program are jogging within the heritage, and if any software program is altering the source code of our sport.

**Quantitative Verifiable**

* This NFR may be quantitatively established with the aid of tempering with the source code in a couple of methods and observing the system’s conduct.
* This NFR can be quantitatively validated by way of testing out more than one cheating software with our recreation and amassing the records on what actions our gadget took toward the one’s malicious software.

**Legislative**

Legislative Requirements means any applicable law, statute, byelaw, regulation, order, consent, permit, approval, regulatory policy, guidance or industry code, rule of court or directives or requirements of any Regulatory Body, delegated or subordinate legislation or notice of any Regulatory Body.

**Privacy**

Privacy requirements are statements that reference key privacy objectives (e.g., Fair Information Practice Principles, or FIPPs) **and specify capabilities and functions that a system must be able to perform.**

**Safety**

The safety requirements are those requirements that are **defined for the purpose of risk reduction**. When a safety function is implemented via software, there also needs to be a hardware platform, in which case a computer system is necessary.

**Examples**

* A digital camera filter app to now not save any form of pictures and area metadata at the cloud servers, most effective filters carried out must be logged for future usage.
* A social media application should handily permit users of age 14 and older.

**Goals**

* The intention of this NFR is to preserve minors secure from the publicity of social media.
* The purpose of this NFR is to maintain privacy as a pinnacle precedence.

**Objectives**

* To acquire this aim, we set our objective that we ought to ask consumer for his/her google identity and retrieve age facts from there, we'd additionally supply the consumer a way to manually enter his/her age then we compare the two.
* To obtain this intention, we don’t log any personal records, complete pictures, or vicinity facts at the severs these statistics receives stored locally on tool.

**Quantitative Verifiable**

* Verify by way of how lots of youngsters get admission to our social media platform.
* Confirm this with the aid of looking at how a great deal of this information are logging to perform functionalities.