DTP@Work
Dedicated Top Professionals

<Enter Department Name>

<Project Name> Non Functional Requirements

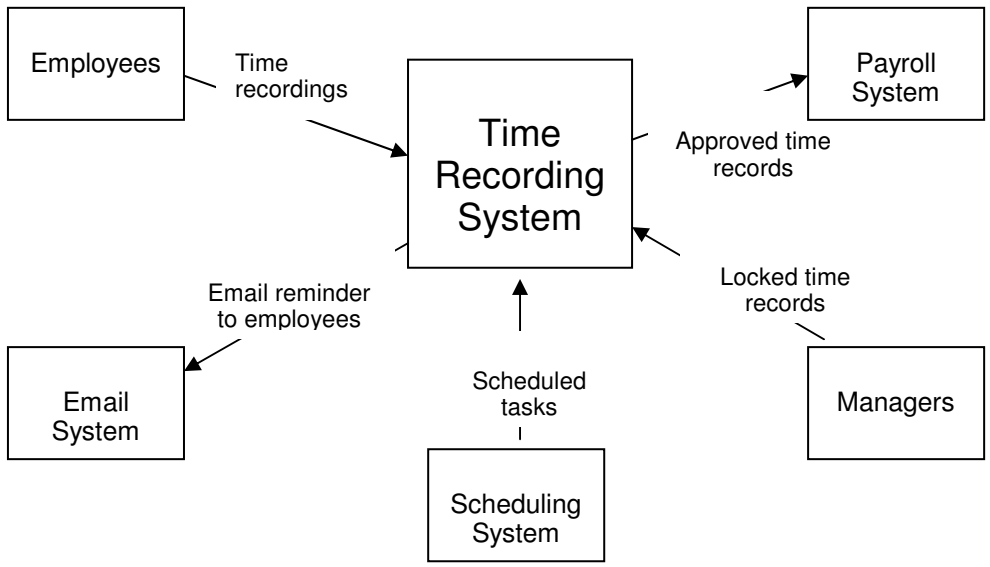
From: < >
To: <>
File Name: *<file name>*
Date First Created: *<mm/dd/yyyy>*
Date Last Updated: *<mm/dd/yyyy this date is very important because there may be multiple versions as new data is analyzed>*

1 SYSTEM CONTEXT & SCOPE

1.1 SCOPE OF THE SYSTEM

Provide a simple context diagram identifying the broad system that may be developed. Note that this includes more than the intended product. Designate the system to be developed as the center of the diagram and then indicate other subject matter domains (persons, system, organizations). To be understood.

Example



1.2 WORKFLOW PARTITIONING

Identify the major business events to which the work responds. The business events are user-defined. The response to each event represents a portion of work that contributes to the total functionality of the work. This will help identify logical chunks of the system that can be used as the basis for discovering detailed requirements.

The event list includes:

Event Name, and Input and output from others (identical with name on system context diagram)

Example Event List

Event Name	Input & Output
1. Scheduling system transmits assigned work tasks	Scheduled tasks (in)
2. Time to record time spent on tasks	Email reminder to employees (out)
3. Employees Record Time against tasks	Time Recordings (in)
4. Managers lock time	Locked Time Records (in)
5. Time to update Payroll records	Approved Time Records (out)

1.3 PRODUCT BOUNDARY

A use case diagram or other diagram can be used to identify boundaries between the users and product. You derive the product use cases by deciding where the product boundary should be for each one of the business events. These decisions are based on your knowledge of the work and the requirements constraints.

1.4 USE CASE LIST

The use case diagram is a graphical way of summarizing all the use cases relevant to the product. If you have a large number of use cases, we find 15-20, is around the limit, then it is better to list the use cases. For each use case on the list you should have: use case number, user/actor name, use case description and use case fit criterion. Also if you have built a use case description and/or any scenario models for this use case then this list can point to them.

2 FUNCTIONAL AND DATA REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS.

Specify the hi-level functional requirements that must be supported by the system. You may also refer to the detailed requirements recorded in the Requirements Traceability Matrix.

Note - If you have produced an event/use case list you can use them to help you trigger the functional requirements for each event/use case. If you have not produced an event/use case list, give each functional requirement a unique number and, to help with traceability, they can be partitioned into event/use case-related groups later in the development process.

2.2 DATA REQUIREMENTS.

A specification of the essential subject matter/business objects/entities/classes that is germane to the system. This might take the form of a first-cut data model, an object model or a domain model. Or it might be adequately dealt with by defining the terms in a data dictionary. If you use a diagram it should clarify the thought of.

3 LOOK, FEEL AND USE REQUIREMENTS

3.1 USER INTERFACE REQUIREMENTS

Provide a description of the spirit of the interface. This can be in the form of text descriptions or preliminary sketches of a possible interface. The intention of this section is to state requirements relating to the interface. Your client may have given you particular demands such as style, colors to be used, degree of interaction and so on. This section captures the requirements for the interface rather than the design for the interface. You should capture the expectations, the constraints and the client's demands for the interface before designing it.

Examples

The product shall have the same layout as the district maps that the engineering department uses now.

The product shall use the company colors.

The product shall be attractive to a teenage audience.

The product shall appear authoritative.

Interface design may overlap the requirements gathering process. This particularly true if you are using prototyping as part of your requirements process. As prototypes develop it is important to capture the requirements that relate to the look and feel. Record these as requirements instead of merely having a prototype to which the client has nodded his approval.

3.2 STYLE OF THE PRODUCT REQUIREMENTS

A description of salient features of the product that are related to the way a potential customer will see the product. For example, if your client wants the product to appeal to the business executive, then a look and feel requirement is that the product has a conservative and

professional appearance. Similarly if the product is for sale to children, then the look and feel requirement is that it be colorful and look like it's intended for children.

You would also consider here the design of the package if this were to be a manufactured product. The package may have some requirements as to its size, style, and consistency with other packages put out by your organization, etc.

The requirements that you record here will guide the designers to produce a product as envisioned by your client.

Note - The look and feel requirements specify the your client's vision of the product's appearance. The requirements may at first seem to be rather vague – "conservative and professional appearance" – but these will be quantified by their fit criterion. The fit criterion in this case give you the opportunity to extract from your client precisely what is meant, and gives the designer precise instructions on what he is to accomplish.

3.3 EASE OF USE REQUIREMENTS

This section describes your client's aspirations for how easy it will be for the intended users of the product to operate it. The product's usability is derived from the abilities of the expected users of the product and the complexity of its functionality.

Examples

The product shall be easy for 11 year-old children to use.

The product shall help the user to avoid making mistakes.

The product shall make the users want to use it.

The product shall be used by people with no training, and possibly no understanding of English.

Note - These examples may seem simplistic, but they do express the intention of the client. To completely specify what is meant by the requirement it is necessary to add a measurement of acceptance. For example a modification to add measurement to the above examples would be:

[An agreed percentage, say 90%] of a test panel of 11 year olds shall be able to successfully complete [list of tasks] within [specified time]

One month's use of the product shall result in a total error rate of less than [an agreed percentage, say 2%]

An anonymous survey shall show that [an agreed percentage, say 75%] of the users are regularly using the product after [an agreed time] familiarization period.

3.4 EASE OF LEARNING REQUIREMENTS

Provide a statement of how easy it should be to learn to use the product. This will range from zero time to a considerable time for complex, highly technical products. This should quantify the amount of time that your client feels is allowable before a user can successfully use the product. This requirement will guide designers in how users will learn the product. For example, the designers may build elaborate interactive help facilities into the product, or the product may be packaged with a tutorial. Alternatively the product may have to be constructed so that all of its functionality is apparent upon first encountering it.

Examples

An engineer shall produce a [specified result] within [specified time] of beginning to use the product, without needing to use the manual.

After receiving [number of hours] training a clerk shall be able to produce [quantity of specified outputs] per [unit of time].

[Agreed percentage] of a test panel shall successfully complete [specified task] within [specified time limit].

The engineers shall achieve [agreed percentage] pass rate from the final examination of the training.

4 PERFORMANCE REQUIREMENTS

4.1 SPEED REQUIREMENTS

Specifies the amount of time available for response or response times.

Some products, usually real-time products, must be able to perform some of their functionality within a given time slot. Failure to do so may mean catastrophic failure or the product will not cope with the required volume of use.

Examples

"Any interface between a user and the automated system shall have a maximum response time of 2 seconds"

"The product shall poll the sensor every 10 seconds"

"The product shall download the new status parameters within 5 minutes of a change"

4.2 SAFETY CRITICAL REQUIREMENTS

Quantify the perceived risk of possible damage to people, property and environment. To understand and highlight the potential damage that could occur when using the product within the expected operational environment.

Provide:

Description of the perceived risk

Factors that could cause the damage

Unit for measuring the factors that could cause the damage

"The product shall be certified to comply with the Health Department's standard E110-98. This is to be certified by qualified testing engineers."

"No member of a test panel of [specified size] shall be able to touch the heat exchanger. The heat exchanger must also comply with safety standard [specify which one]."

4.3 PRECISION REQUIREMENTS

Quantify the desired accuracy of the results produced by the product. Collect from the client or user the expectations for the precision of the product.

Examples

All monetary amounts shall be accurate to 2 decimal places.

Accuracy of road temperature readings shall be within + or - 2 degrees centigrade.

4.4 RELIABILITY AND AVAILABILITY REQUIREMENTS

This section quantifies the necessary reliability of the product. This is usually expressed as the allowable time between failures, or the total allowable failure rate. It should also quantifies the expected availability of the product.

Examples

The product shall be available for use 24 hours per day, 365 days per year.

The product shall be available for use between the hours of 8:00am and 5:30pm.

The escalator shall run from 6am until the last flight arrives at 10pm.

The product shall achieve 99% up time.

4.5 Capacity and Scalability Requirements

This section specifies the volumes that the product must be able to deal with and the numbers of data stored by the product. And, the expected increases in size that the product must be able to handle.

Examples

The product shall cater for 300 simultaneous users within the period from 9:00am to 11:am. Maximum loading at other periods will be 150.

The product shall be capable of processing the existing 100,000 customers. This number is expected to grow to 500,000 within three years.

The product shall be able to process 50,000 transactions an hour within two years of its launch.

5 Operational Requirements

5.1 Expected Technological Environment Requirements

Specify the hardware and other devices that make up the operating environment for the new system. Identify all the components of the new system so that the acquisition, installation and testing can be effectively managed.

Describe the hardware and other devices that make up the operating environment for the new system. This may not be known at the time of the requirements process, as these devices may be decided at design time.

5.2 Partner Applications & Interfaces Requirements

Describe the other applications that the product must interface with.

For each inter-application interface specify:

- The data content
- The physical material content
- The medium that carries the interface
- The frequency
- The volume

5.3 SUPPORTABILITY REQUIREMENTS

Specify the level of support that the product requires. This is often done using a help desk. If there are to be people who provide support for the product, is that considered part of the product and are there any requirements for that support. You might also build support into the product itself, in which case this is the place to write those requirements.

Consider the anticipated level of support, and what forms it might take. For example, there may be a constraint that there is to be no printed manual. Or you might consider that the product is to be entirely self-supporting.

5.4 MAINTAINABILITY AND PORTABILITY REQUIREMENTS

Specify how easy it must be to maintain this product. Quantify the time necessary to make specified changes to the product. Describe other platforms or environments to which the product must be ported.

6 SECURITY REQUIREMENTS

6.1 SYSTEM CONFIDENTIALITY REQUIREMENTS

Specification of who has authorized access to the system, and under what circumstances that access is granted.

User role/s and/or names of people who have clearance

Is there any data that is sensitive to the management? Is there any data that low level users do not want management to have access to? Are there any processes that might cause damage or might be used for personal gain? Are there any people who should not have access to the system?

Avoid solving how you will design a solution to the security requirements. For instance, don't design a password system. Your

aim here is to identify what the security requirement is. The design will come from this description.

Consider asking for help. Computer security is a highly specialized field, and one where improperly qualified people have no business being. If your product has need of more than average security, we advise that you make use of a security consultant. They are not cheap, but the results of inadequate security can be even more expensive.

6.2 DATA INTEGRITY REQUIREMENTS

Specification of the required integrity of databases and other files.

Examples

“The clients shall receive updated customer files once every 24 hours.”

“Identical up-to-date booking information shall be available to all users of the system.”

How will the information be used? What is the impact on the customer’s business if the information is out of date? Will there be a ripple effect if two different users have different versions of the system?

6.3 AUDIT REQUIREMENTS

Specification of the required audit checks.

7 LEGAL REQUIREMENTS

Provide a statement specifying the legal requirements for this system.

Examples

“Personal information shall be implemented so as to comply with the data protection act.”

Are there any copyrights that must be protected? Alternatively, do any competitors have copyrights that you might be in danger of infringing?

Is it a requirement that developers have not seen competitors’ code or even have worked for competitors?

Is there any pending legislation that might affect the development of this system?

8 STANDARDS REQUIREMENTS

Specifying applicable standards and reference detailed standards descriptions.

Example

“The product shall comply with MilSpec standards.”

“The product shall comply with insurance industry standards”.

“The product shall be developed according to System Lifecycle Standard development steps.”

9 OFF-THE-SHELF SOLUTIONS REQUIREMENTS

9.1 COTS SYSTEM SOLUTIONS REQUIREMENTS

List of existing products that should be investigated as potential solutions.

Is it possible to buy something that already exists or is about to become available? It may not be possible at this stage to say with a lot of confidence, but any likely products should be listed here.

Also consider whether there are products that must not be used.

9.2 READY-MADE COMPONENTS REQUIREMENTS

Description of the candidate components, either bought-in or built by your company, that could be used by this project. List libraries that could be a source of components.

10 CONSTRAINTS AND ASSUMPTIONS

This section describes constraints on the requirements and the eventual design of the product.

10.1 SOLUTION CONSTRAINTS

This specifies constraints on the way that the problem must be technically solved. Carefully describe any standard technology and the reason for using the standard technology. Be careful because anyone who has experience/exposure to a piece of technology tends

to see requirements in terms of that technology. This tendency leads people to impose solution constraints for the wrong reason and it's very easy for untrue constraints to creep into a specification. If you impose untrue constraints the danger is that you do not have the creative freedom to come up with the best solution to the problem. The solution constraints should only be those that are absolutely non-negotiable. In other words, however you solve this problem you must use this particular technology. Any other solution would be unacceptable.

10.2 EXTERNAL FACTORS

Describe other forces, systems, and activities in the world that have an effect on this system. External facts might contribute to requirements. They will have an effect on the eventual design of the product.

10.3 ASSUMPTIONS

Provide a list of the assumptions that the project team is making. These might be about the intended operational environment, but can be about anything that has an effect on the product. This helps make everyone on the project aware of assumptions that have been made.

Examples

Assumptions about new laws or political decisions.

Assumptions about the technological environment in which the product will operate. These assumptions should highlight areas of expected compatibility.

The software components that will be available to the developers.

Other products being developed at the same time as this one.

Availability and capability of bought-in components.

Dependencies on computer systems or people external to this project.

11 FUTURE REQUIREMENTS

Define requirements that will not be part of the agreed product. These requirements might be included in future versions of the product. This allows requirements to be gathered, even though they cannot be

part of the current development. To ensure that good ideas are not lost.

The requirements gathering process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section is a hold-all for requirements in waiting. The intention is to avoid stifling your users and clients by having a repository of future requirements. You are also managing expectations by making it clear that you take these requirements seriously but they will not be part of the agreed product.