**Software Engineering Project Report**



**A Sample Document for   
Generating Consistent Professional Reports**

**Prepared by**

**Mariam Zikaria**

**Priyam Patel**

**Dhruv Patel**

**Mohammad Hasan**

**for use in CS 440**

**at the**

**University of Illinois Chicago**

**Spring 2017**

**Table of Contents**

[How to Use This Document 2](#_Toc461441751)

[List of Figures 9](#_Toc461441752)

[List of Tables 10](#_Toc461441753)

[I Project Description 12](#_Toc461441754)

[1 Project Overview 12](#_Toc461441755)

[2 The Purpose of the Project 12](#_Toc461441756)

[2a The User Business or Background of the Project Effort 12](#_Toc461441757)

[2b Goals of the Project 12](#_Toc461441758)

[2c Measurement 13](#_Toc461441759)

[3 The Scope of the Work 13](#_Toc461441760)

[3a The Current Situation 13](#_Toc461441761)

[3b The Context of the Work 14](#_Toc461441762)

[3c Work Partitioning 16](#_Toc461441763)

[3d Competing Products 18](#_Toc461441764)

[4 The Scope of the Product 18](#_Toc461441765)

[4a Scenario Diagram(s) 18](#_Toc461441766)

[4b Product Scenario List 19](#_Toc461441767)

[4c Individual Product Scenarios 19](#_Toc461441768)

[5 Stakeholders 19](#_Toc461441769)

[5a The Client 19](#_Toc461441770)

[5b The Customer 20](#_Toc461441771)

[5c Hands-On Users of the Product 20](#_Toc461441772)

[5d Priorities Assigned to Users 21](#_Toc461441773)

[5e User Participation 22](#_Toc461441774)

[5f Maintenance Users and Service Technicians 22](#_Toc461441775)

[5g Other Stakeholders 22](#_Toc461441776)

[6 Mandated Constraints 23](#_Toc461441777)

[6a Solution Constraints 24](#_Toc461441778)

[6b Implementation Environment of the Current System 25](#_Toc461441779)

[6c Partner or Collaborative Applications 25](#_Toc461441780)

[6d Off-the-Shelf Software 26](#_Toc461441781)

[6e Anticipated Workplace Environment 27](#_Toc461441782)

[6f Schedule Constraints 28](#_Toc461441783)

[6g Budget Constraints 28](#_Toc461441784)

[7 Naming Conventions and Definitions 29](#_Toc461441785)

[7a Definitions of Key Terms 29](#_Toc461441786)

[7b UML and Other Notation Used in This Document 30](#_Toc461441787)

[7c Data Dictionary for Any Included Models 30](#_Toc461441788)

[8 Relevant Facts and Assumptions 31](#_Toc461441789)

[8a Facts 31](#_Toc461441790)

[8b Assumptions 32](#_Toc461441791)

[II Requirements 33](#_Toc461441792)

[9 Product Use Cases 33](#_Toc461441793)

[9a Use Case Diagrams 33](#_Toc461441794)

[9b Product Use Case List 34](#_Toc461441795)

[9c Individual Product Use Cases 35](#_Toc461441796)

[10 Functional Requirements 35](#_Toc461441797)

[11 Data Requirements 36](#_Toc461441798)

[12 Performance Requirements 38](#_Toc461441799)

[12a Speed and Latency Requirements 38](#_Toc461441800)

[12b Precision or Accuracy Requirements 39](#_Toc461441801)

[12c Capacity Requirements 39](#_Toc461441802)

[13 Dependability Requirements 40](#_Toc461441803)

[13a Reliability Requirements 40](#_Toc461441804)

[13b Availability Requirements 40](#_Toc461441805)

[13c Robustness or Fault-Tolerance Requirements 41](#_Toc461441806)

[13d Safety-Critical Requirements 42](#_Toc461441807)

[14 Maintainability and Supportability Requirements 43](#_Toc461441808)

[14a Maintenance Requirements 43](#_Toc461441809)

[14b Supportability Requirements 43](#_Toc461441810)

[14c Adaptability Requirements 44](#_Toc461441811)

[14d Scalability or Extensibility Requirements 44](#_Toc461441812)

[14e Longevity Requirements 45](#_Toc461441813)

[15 Security Requirements 45](#_Toc461441814)

[15a Access Requirements 45](#_Toc461441815)

[15b Integrity Requirements 46](#_Toc461441816)

[15c Privacy Requirements 47](#_Toc461441817)

[15d Audit Requirements 48](#_Toc461441818)

[15e Immunity Requirements 48](#_Toc461441819)

[16 Usability and Humanity Requirements 48](#_Toc461441820)

[16a Ease of Use Requirements 48](#_Toc461441821)

[16b Personalization and Internationalization Requirements 50](#_Toc461441822)

[16c Learning Requirements 51](#_Toc461441823)

[16d Understandability and Politeness Requirements 52](#_Toc461441824)

[16e Accessibility Requirements 52](#_Toc461441825)

[16f User Documentation Requirements 53](#_Toc461441826)

[16g Training Requirements 54](#_Toc461441827)

[17 Look and Feel Requirements 54](#_Toc461441828)

[17a Appearance Requirements 54](#_Toc461441829)

[17b Style Requirements 55](#_Toc461441830)

[18 Operational and Environmental Requirements 56](#_Toc461441831)

[18a Expected Physical Environment 56](#_Toc461441832)

[18b Requirements for Interfacing with Adjacent Systems 56](#_Toc461441833)

[18c Productization Requirements 57](#_Toc461441834)

[18d Release Requirements 58](#_Toc461441835)

[19 Cultural and Political Requirements 58](#_Toc461441836)

[19a Cultural Requirements 58](#_Toc461441837)

[19b Political Requirements 59](#_Toc461441838)

[20 Legal Requirements 60](#_Toc461441839)

[20a Compliance Requirements 60](#_Toc461441840)

[20b Standards Requirements 61](#_Toc461441841)

[III Design 61](#_Toc461441842)

[21 System Design 61](#_Toc461441843)

[21a Design goals 61](#_Toc461441844)

[22 Current Software Architecture 63](#_Toc461441845)

[23 Proposed Software Architecture 63](#_Toc461441846)

[23a Overview 63](#_Toc461441847)

[23b Class Diagrams 63](#_Toc461441848)

[23c Dynamic Model 63](#_Toc461441849)

[23d Subsystem Decomposition 64](#_Toc461441850)

[23e Hardware / software mapping 64](#_Toc461441851)

[23f Data Dictionary 64](#_Toc461441852)

[23g Persistent Data management 64](#_Toc461441853)

[23h Access control and security 64](#_Toc461441854)

[23i Global software control 65](#_Toc461441855)

[23j Boundary conditions 65](#_Toc461441856)

[24 Subsystem services 65](#_Toc461441857)

[25 User Interface 65](#_Toc461441858)

[26 Object Design 65](#_Toc461441859)

[26a Object Design trade-offs 65](#_Toc461441860)

[26b Interface Documentation guidelines 66](#_Toc461441861)

[26c Packages 66](#_Toc461441862)

[26d Class Interfaces 66](#_Toc461441863)

[IV Test Plans 66](#_Toc461441864)

[27 Features to be tested / not to be tested 66](#_Toc461441865)

[28 Pass/Fail Criteria 66](#_Toc461441866)

[29 Approach 67](#_Toc461441867)

[30 Suspension and resumption 67](#_Toc461441868)

[31 Testing materials ( hardware / software requirements ) 67](#_Toc461441869)

[32 Test cases 67](#_Toc461441870)

[33 Testing schedule 67](#_Toc461441871)

[V Project Issues 68](#_Toc461441872)

[34 Open Issues 68](#_Toc461441873)

[35 Off-the-Shelf Solutions 68](#_Toc461441874)

[35a Ready-Made Products 68](#_Toc461441875)

[35b Reusable Components 69](#_Toc461441876)

[35c Products That Can Be Copied 69](#_Toc461441877)

[36 New Problems 70](#_Toc461441878)

[36a Effects on the Current Environment 70](#_Toc461441879)

[36b Effects on the Installed Systems 70](#_Toc461441880)

[36c Potential User Problems 70](#_Toc461441881)

[36d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 71](#_Toc461441882)

[36e Follow-Up Problems 71](#_Toc461441883)

[37 Tasks 72](#_Toc461441884)

[37a Project Planning 72](#_Toc461441885)

[37b Planning of the Development Phases 72](#_Toc461441886)

[38 Migration to the New Product 73](#_Toc461441887)

[38a Requirements for Migration to the New Product 73](#_Toc461441888)

[38b Data That Has to Be Modified or Translated for the New System 73](#_Toc461441889)

[39 Risks 74](#_Toc461441890)

[40 Costs 75](#_Toc461441891)

[41 Waiting Room 76](#_Toc461441892)

[42 Ideas for Solutions 77](#_Toc461441893)

[43 Project Retrospective 77](#_Toc461441894)

[VI Glossary 78](#_Toc461441895)

[VII References / Bibliography 78](#_Toc461441896)

[VIII Index 78](#_Toc461441897)

### ****List of Figures****

Figure 1- Business Event List ………………………………………………………………… 12

Figure 2- Use Case Diagram ………………………………………………………………….. 13

# 

# Project Description

## Project Overview

A brief description of the product to be produced, before getting into details.

A virtual reality based program that educates children on elementary topics such as basic arithmetic, colors, and shapes.

## The Purpose of the Project

### The User Business or Background of the Project Effort

Our group has decided to create a virtual reality based product to educate young children. This product will be compatible with an HTC Vive. This program will be able to be downloaded onto a computer with a high-definition graphic card. Graphics using Unity 3D will need to be made, in a setting where virtual reality applications can be tested using controllers and screens. Our project was inspired by nostalgic memories of playing with wood blocks as children. It also reminded us of cartoon shows where questions would be asked, and we would have to wait years to answer like Dora the Explorer and Blues Clues. This product will be available for download online, and be marketed towards young parents.

Motivation

The motivation for this project is to provide a program that will teach kids what was once taught to us on educational television shows, but with much more interaction, creating a more realistic experience for the child. This would allow children to learn quicker, and be more independent in their thinking.

Considerations

In certain cases, where children have children disabilities, the problem is very serious, and these children need special attention and time to learn new things. Our application will help kids with learning disabilities as well, because it is making learning fun, and the children will be able to incorporate education with “real-life”.

### Goals of the Project

The real reason for this project is to make education fun for little kids and more interactive. It will help them learn better and be more independent. There are high chances that the development team may forget the main purpose of the project and may end up going in a different direction. To keep the team motivated towards the main goal of the project and to avoid the danger of losing the main purpose of this project, there should be meetings on a regular basis to remind the team what the main goal. To do that it may be necessary to appoint a person to be custodian of the goals, but it is probably sufficient to make the goals public and periodically remind the developers of them. It should be mandatory to acknowledge the goals at every review session. It’s available in a form of a CD or available to download on a computer.

### Measurement

The main goal of this project is the give the users i.e. the children between age 5 – 8 and interactive way of learning. The main target audience for this software would be grade schools and customers with kids between that age who has HTC VIVE or Oculus Rift. This project has a lot of potential since we can replace from having extra teachers in the class to this automation which just requires a person

Initially we will try to demo it in few grade school and get reviews from the kids and teacher and depending on the response we can either modify or make more changes. This is to make sure we have a good understanding of what the main product should look like.

## The Scope of the Work – Dhruv is doing This

### The Current Situation

This product will be used in an educational environment such as a school or tutoring center. It will change the existing processes in schools by requiring kids thought the world getting same level of educational foundation. This product will be used instead of teachers giving students individual attention to learn, which will change the education system in schools and tutoring centers. Since it is meant for grade school students for now, it will have a bigger audience not only in sates but thought out the world. Especially, in the developing or undeveloped countries where kids don’t have the access to good education resources.

### The Context of the Work – Mariam needs help

Some external people that we would need to be involved in this project are graphics designers, HTC, and an A/V team to help us with any technical issues that arise during our set up and testing of our project. The core of this project will



Figure – Use Case Diagram

### Work Partitioning

Business Event List

Event Name Input and Output Summary

|  |  |  |
| --- | --- | --- |
| 1. Graphic Designer | Creates the graphics for the game (in) | Designs the elements that the user will interact with in the game. |
| 2. User Interaction with the shapes | Front end(in) | Makes sure that the user interface is easy to interact with |
| 3. VR Designer | Creates Visuals(in) | Makes sure that everything is in the right place |
| 4. Artist | Draws pieces (in) | Makes the drawings of the game pieces with the designer |
| 5. Score to keep track of activity | Scoring(out) | Records the scoring for teachers and for students’ progress |
| 7. A/V Technician | Setup (in) | Maintaining and making changes and sets up HTC vive |
| 8. HTC Vive System | System Update(out) | Updates the system in real time. |

### Competing Products

There are some educational games out there like Jigsaw Puzzle, letter blocks and more but they’re not in VR form. You can’t interact with those games. They are used for educational purposes but are not used classroom settings. To make sure that the product is being used to its fullest potential and it meets all the requirement for kids to educate themselves. The game is going to be design with the help of child psychologist and a lot of research will go behind the actual game to make sure that the users are benefiting from it and getting the education they need.

## The Scope of the Product

This project will include various sensors that will interact with the movement of the children. These movements will be captured and analyzed to update correctly the program. Some external entities that will interact with our product, aside from the users, will be the camera and sensors. There will be a controller that the users will have. This will directly show the hand movements and interactions by the user with the program. Our hardware will be HTC Vive using the Unity engine for the creation.

### Scenario Diagram(s)

There will be one main scenario that will occur that is playing the game,

Primary Actors:

Student/Child

Teacher

Preconditions:

VR Set up and teacher has access to the lesson plan

Basic Flow:

Set up lesson

Initiate and set up game

Play game

Collect and Review data

Alternative Flow:

Invalid teacher lesson

Child incorrectly interacts with product

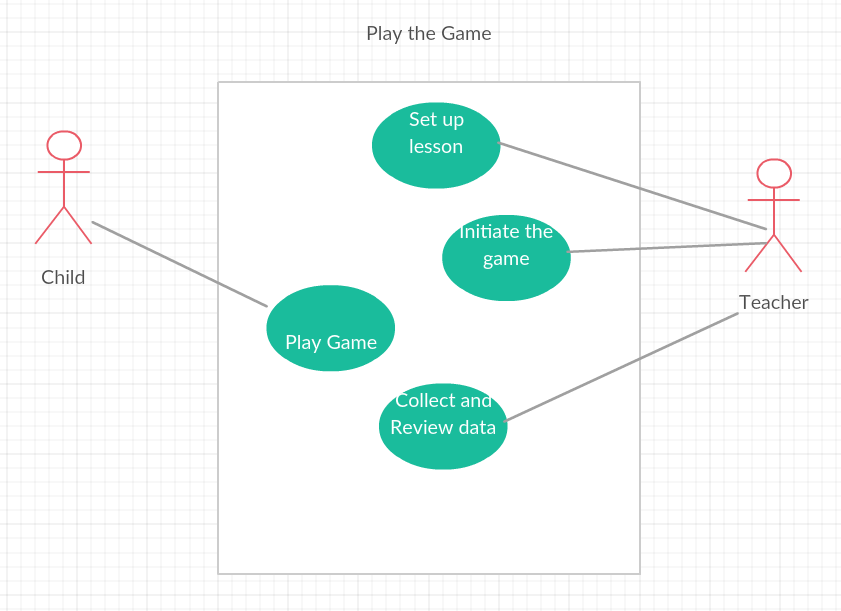
Collect and Review data

Figure 2 – Use case Diagram

### Product Scenario List

Teacher log in and set up lesson

Initiate lesson and send it for using by Children

Users will play the game

Level 1

Different Math Games

Level 2

Different Math Games

Level 2

Different Math Games

Level 3

Different Math Games

Level 4

Different Math Games

Level 5

Different Math Games

Collect data

Analyze data

### Individual Product Scenarios

**VR Gaming**: During the day, Mary has included in her daily lessons a time for VR learning. This entails setting up a classroom lecture, math level 1. These are kindergarten students who are learning how math works. So they need to have their lesson plans set out for them. The kids will come into class and put their VR headsets in their special room so that the sensors and video cameras will record and analyze the work that they do. They will

## Stakeholders

### The Client

A client for our product is an institution for educating children. This product can be used for grade school children but it can be expanded to middle school and high school by adding content to the product. But as of right now since we are focusing on grade school students. Hence, following are the names of our clients around Chicago area.

* Mundelein community grade school
* Stevenson grade school

The client role in this project is going to be specifying the school curriculum and to provide their budget and system description on which they are planning to host the application. This application uses room space; the client is also supposed to have a dedicated space.

### The Customer

A client for our product is an institution for educating children. This product can be used for grade school children but it can be expanded to middle school and high school by adding content to the product. But as of right now since we are focusing on grade school students. Hence, following are the names of our clients around Chicago area.

* Mundelein community grade school
* Stevenson grade school

To meet the client or the customer’s requirement the project will require the help of professionals who understand the need to educate grade level students and their psychology. We shall also provide a month free trail which will include the entire set to test how the students responded and if it is in the client’s interest to buy this product.

### Hands-On Users of the Product

● User name/category: Schoolchildren

● User role: Use the virtual reality game, participate in the activities and answer questions

● Subject matter experience: The children playing this game will be beginner users, new to the game. So, novice.

● Technological experience: Being children, they will have exposure to other children games and technology such as iPad and iPhone games. So, novice.

● Other user characteristics: This game is meant for children who are in preschool through fifth grade. There will be a variety of difficulty levels and topics that range from adding and subtracting, to division and multiplication. Age group and grade level will play into factor in choosing the difficulty level. Children with learning disabilities are the motivation in a virtual reality educational game such as this one, because they have a chance to learn math beyond simply looking at numbers and symbols.

Some users we have incorporated into our project are the HTC Company, because we are using their HTC Vive device and have to keep up with the HTC Vive System Updates and be in communication regarding the device when troubleshooting. We also have a Graphic Designer as a user because they are in charge of the visuals and animations required for this project. Another user, the Audio/Visual (A/V) Team, will be in contact with us to help set up the device, and help us in times of technical issues.

### Priorities Assigned to Users

Key User: Graphic Designer

Secondary User: A/V Technician, HTC

Unimportant user: N/A

The focus group for these products are education institution. More specifically it will be the grade school and middle schools. Since our product currently focuses on content for small kids, the plan is to get the feedback from the users i.e. the kids and see how they respond to it. If this turns out to be successful and helps the kids, we will try to incorporate more content. Since grade school students are good at picking things up when they play and also their content being the addition of content or update.

### User Participation

As from the client side it will be necessary for institution or academies to give the descriptions of their course work. The client would need to have a dedicated space and have an instructor to learn the technology and can supervise students while they are using the product. They also need a server with GPU of Nvidia GeFOrce GTX970, or AMD Radeon R9 290 equivalent or greater and HTV Vive which will be either provided with the software or can also be order and a client technician can set up for them.

### Maintenance Users and Service Technicians

HTC Vive is a product which will need some technical knowledge to set-up and installation for the program. Since the targets user are grade school students, we want to make sure it is a same environment for them. This device which is HTC Vive will require maintain ace and some technical some technical training to set up the device.

### Other Stakeholders

● Educational Experts (Teachers)

We require a team that will be decimated for helping the teacher in the institute to set up the product and learn the set-up of the application and how to monitor a user’s i.e. the student’s progress based on the score.

● Testers

The Testers are a group of grade school students who will be examining the software and giving us feedback.

● Business analysts

Come up with different or better business solutions. Also, responsible for getting the client requirement in term of technical requirement to the Project Manager.

● Technology experts

Technology are the developers who take the business technical requirement from the Project Manager. They must also be in constant contact with the Graphic Designer because of the usability and functional requirement that they will have to consider while designing.

● System designers

The systems designers are the electrical engineer that will help set HTC Vive.

● Marketing experts

We must have a marketing team that will be responsible foe the Online and Television advertisement.

● Legal experts

The legal experts are there to assist the company when in time of a contract signing or dealing with any legal matters.

● Graphic Designer

The Graphics designer will be responsible for the animation and the shapes, text and objects that will be involved in the game. They will also be in constant contact with the developers and projects managers to get the right specifics of the system before they start designing.

* Child Psychologist

Child Psychologist are a vital part of this project since we must understand how the kids respond in the application and what will be the right amount of challenge question should be added in the content and what changes we can make to the application.

(1) Easy to use functionality: Make sure that the AI system and the visuals are easy on the eyes and not too confusing to navigate through

* + - 1. Teacher controlled: The teacher has autonomy on the lessons administered and graded
      2. Friendly colors: this is strictly in the visuals. Needs to be child friendly and not intense
      3. Common sense understanding: the UI needs to be in a common sense method. No complex “clicks” to get to the end result.
      4. Understanding the logic behind the code: When coding or writing out the design of the project to use. There are certain logical designs that need to be implemented. This is why the Unity engine is used to make the proper systematic approach in the construction. This will pair perfectly with the HTC Vive.
      5. Stabilization: With the use of HTC Vive, we need to account for the camera set up. Due to many different technologies out there, we need to limit our spending and cost. This requires us to invest in a stabilizing camera, so a stable environment can be rendered for the children. Stabilization is important for the spotting and interaction notice from the users.
      6. A safe environment: Since our product requires a room to be built and set up in a manner that is geared towards the product, we would need to make sure that those accessing the room are not at risk. This would require us to invest in gold plated wiring, wall mounting rotating cameras. The walls would need to be fitted with foam and soft material.

Examples

Description: Users will use the head piece and the controllers to interact with the product in a stable room fitted with body recognizing cameras.

Rationale: The room is necessary for the data to be read and registered in an accurate manner

Fit criterion: All movements will be analyzed and registered to give the users the best experience

Description: The cameras will need to be fitted to be stable and view the student from all angles. This will mean that the room needs to have multiple cameras to see all angles.

Rationale: This is because any shaking will cause the body of the user to be distorted and the program will not register the movements properly.

Fit criterion: The Product will be HTC Vive exclusive.

Description: The product shall be a hand-held device. HTC Vive will be the product our software will work with efficiently

Rationale: This product is being marketed to children and schools. A software technology that is child friendly and school safe is something we need to gear towards.

Fit criterion: The product shall be made to fit the HTC Vive requirements, such as using high graphic cards and having the proper technology.

### Implementation Environment of the Current System

The product will be installed on hand held devices, owned or lent to schools, that will have the software and the capability to interact with the HTC Vive.

### Partner or Collaborative Applications

Computers, green screens, cameras to monitor activity, etc. A computer is required for the HTC Vive application, but because it is a high definition virtual reality device, a special graphic card will be needed for the computer to ensure proper visuals We will also need to use Unity 3D, Microsoft Visual Studio, and blender which will be used by the graphic designer and the programmers.

### Off-the-Shelf Software

This software can also be distributed off-the-shelf so that it can be used as any commercial software and be ready to us. But to use this software the machine or system on which this is application will be hosted should meet the minimum requirement of being able to run certain graphic. This program will be able to be downloaded onto a computer with a high-definition graphic card. Graphics using Unity 3D will need to be made, in a setting where virtual reality applications can be tested using controllers and screens.

### Anticipated Workplace Environment

The work place for this project must be at least 10 ft x 10 ft room so that the HTC Vive can be set-up

Also there should always be a trainer teacher monitoring the user and keep track of the progress.

### Schedule Constraints

Content

Have a working prototype by the beginning of a school year, and then have a product released by mid-year. So there is so level of hype that is created

Motivation

Having something to show by the beginning of the year gives us working data and tweaks we can make before releasing something substantial

Considerations

The deadline for a product release would be the end of a calendar year. But the testing product can be released for secure testing by users at the beginning of a school year

● What happens if we don’t build the product by the end of the calendar year?

● What is the financial impact of not having the product by the beginning of the Christmas buying season?

* Is there a leeway if the product cannot be released for testing by the beginning of the school year?

We do plan to have the product demoed initially to the client and depending on the

### 

### Budget Constraints

If the product is too expensive for the users to get into their schools, then that will defeat the purpose of supplying it to them.

## Naming Conventions and Definitions

### Definitions of Key Terms

HTC Vive: This is a VR headset that will provide the kids an interact way of learning.

Controllers: These controller will track the users gesture and perform the necessary task.

Unity 3D: This is a 3D modeling software that will be used by the Graphic Designer to make the shapes more appealing and content more interesting.

### UML and Other Notation Used in This Document

This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in [4]. Any exceptions are noted where used.

## Relevant Facts and Assumptions

### Facts

The HTC Vive game uses Blender, Unity 3D, and HTV Vive software for application development and its interactive feather helps student learn faster.

### Assumptions

The main assumption is that the combination of the software’s will work.

To run the appropriate graphics for this application the customers system must have the appropriate situation.

# Requirements

## Product Use Cases \_ Dhruv

This section begins to describe in more specific and precise detail exactly what steps the system takes in the course of its performance. Use cases serve not only to more specifically define the system ( and its boundaries ), but also to identify functional requirements, to identify initial objects / classes, and to organize the work.

### Use Case Diagrams

Use Case diagrams serve two purposes: As a form of graphical table of contents listing the individual use-cases, and also to define the boundary of what is included as part of the proposed system and what is not included.

A use case diagram identifies the boundaries between the users (actors) and the product. You arrive at the product boundary by inspecting each business use case and determining, in conjunction with the appropriate stakeholders, which part of the business use case should be automated (or satisfied by some sort of product) and what part should be done by the user. This task must take into account the abilities of the actors (section 3), the constraints (section 4), the goals of the project (section 1), and your knowledge of both the work and the technology that can make the best contribution to the work.

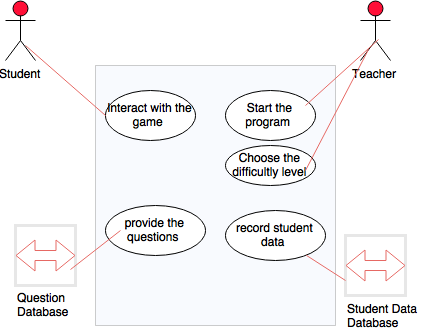
The use case diagram shows the actors outside the product boundary (the rectangle). The product use cases are the ellipses inside the boundary. The lines denote usage. Note that actors can be either automated or human.

Depending on the complexity of the product it may be necessary to use more than one diagram to list all of the use cases. When more than one diagram is required the use-cases can be divided up several ways: Normal operations versus exceptional cases, or daily tasks versus monthly tasks, or user tasks versus administration tasks, etc.

Example



Derive the product use cases by deciding where the product boundary should be for each business use case. These decisions are based on your knowledge of the work and the requirements constraints.



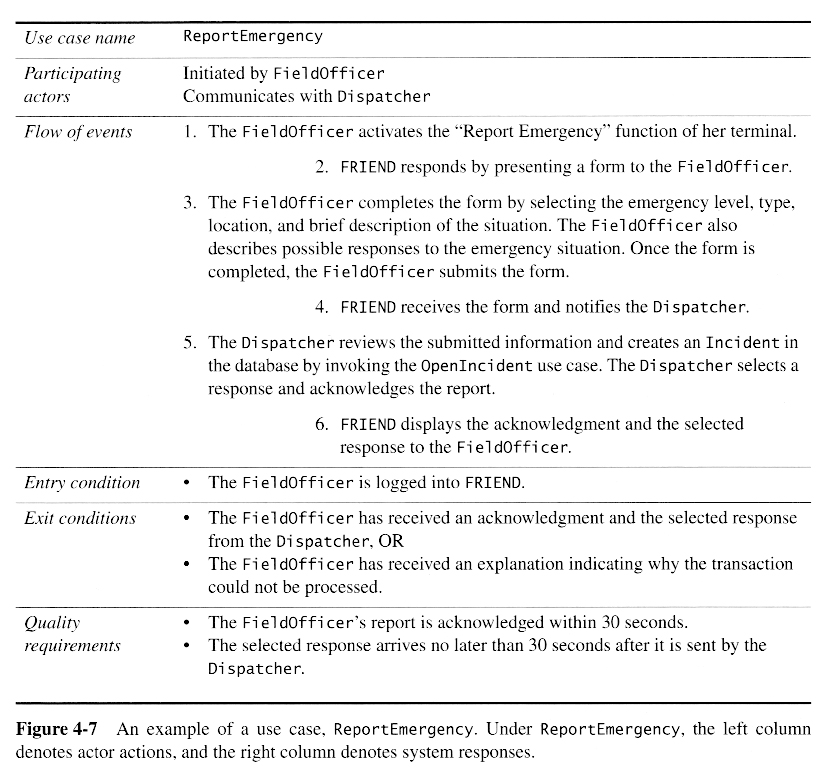
### Product Use Case List

The use case diagram is a graphical way of summarizing the product use cases relevant to the product. If you have a large number of product use cases (we find 15–20 is a good limit), then it is better to make a list of the product use cases and model or describe each one individually.

### Individual Product Use Cases

Use cases are similar to scenarios, in that both tell the story of how the system interacts with the user(s) in response to some business event or while conducting some business task. The difference is that use-cases are much more formal, with certain pre-determined sections for each use-case, and that use-cases indicate clearly what action the system takes in response to what actions taken by the user.

For example, here is Figure 4.7 from "Object Oriented Software Engineering" by Bruegge and DuToit:



## Functional Requirements

Content

A specification for each functional requirement. As with all types of requirements, use the requirements shell*.* A full explanation is included in this template’s introductory material.

Motivation

To specify the detailed functional requirements for the activity of the product.

Examples



Fit Criterion

Each functional requirement should have a fit criterion or a test case. In any event, the fit criterion is the benchmark to allow the tester to determine whether the implemented product has met the requirement.

Considerations

If you have produced an event/use case list (see sections 7b and 8a), then you can use it 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, partition these requirements into event/use case–related groups later in the development process.

## Data Requirements

Content

A specification of the essential subject matter, business objects, entities, and classes that are germane to the product. It might take the form of a first-cut class model, an object model, or a domain model. Alternatively, these requirements might be described by defining the terms in the dictionary described in section 5.

Motivation

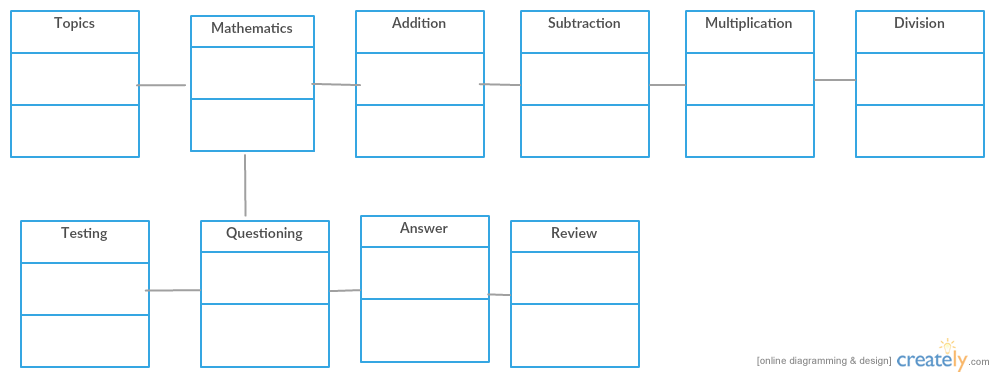
To clarify the system’s subject matter, thereby triggering recognition of requirements not yet considered.

Example

This is a model of the system’s business subject matter using the Unified Modeling Language (UML) class model notation.



You can use any type of data or object model to capture this knowledge. The issue is to capture the meaning of the business subject matter and the connections between the individual parts, and to show that you are consistent within your project. If you have an established company standard notation, use that, as it will help you to reuse knowledge between projects.



Considerations

Are there any data or object models for similar or overlapping systems that might be a useful starting point? Is there a domain model for the subject matter dealt with by this system?

## Performance Requirements

### Speed and Latency Requirements

Content

Specifies the amount of time available to complete specified tasks. These requirements often refer to response times. They can also refer to the product’s ability to operate at a speed suitable for the intended environment.

We will have time constraints on answering questions based on the mathematical questions given to the children. The time the child takes to answer questions will be recorded, and there will be a timer to implement this.

Motivation

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 (e.g., a ground-sensing radar in an airplane fails to detect an upcoming mountain) or the product will not cope with the required volume of use (e.g., an automated ticket-selling machine).

The motivation behind the timed questions is to get children to answer questions, and children can see what questions took them longer, and what they did quickly during the review of the answers they have given. Failure to answer questions in the given time will cause the child to have to repeat similar questions until they answer in the given amount of time, or else they are asked to answer questions with less difficulty. This scaling of difficulty will be implemented with each question, where each question will have its own “weight”.

Examples

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

The response shall be fast enough to avoid interrupting the user’s flow of thought.

The product shall poll the sensor every 10 seconds.

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

-----------------------------------------

The movement of the blocks will have a maximum response time of 5 seconds.

The gestures given by the children to move these blocks will have a maximum response time of 5 seconds (<5 seconds for the blocks to be placed in the desired location on the screen)

The question interface will be displayed within 30 seconds after all blocks have been displayed, and the number of blocks moved will be displayed for basic counting learning.

For the mathematical operation questions, the questions will be displayed within 10 seconds of being chosen.

The response time to answer these questions will be based on the difficulty of the questions, but no longer than 1 minute per question. This time can be adjusted to be shorter for a “times table” implementation, similar to multiplication and division tables that we were given as children, where questions are displayed rapidly, where questions can be skipped, and the review section will display the score.

Fit Criterion

Fit criteria are needed when the description of the requirement is not quantified. However, we find that most performance requirements are stated in quantified terms. The exception is the second requirement shown above, for which the suggested fit criterion is

The product shall respond in less than 1 second for 90 percent of the interrogations. No response shall take longer than 2.5 seconds.

Considerations

There is a wide variation in the importance of different types of speed requirements. If you are working on a missile guidance system, then speed is extremely important. By contrast, an inventory control report that is run once every six months has very little need for a lightning-fast response time.

Customize this section of the template to give examples of the speed requirements that are important within your environment.

### Precision or Accuracy Requirements

Content

Quantification of the desired accuracy of the results produced by the product.

We hope to achieve progress being made throughout the activities, increasing difficulty level the more the child plays.

Motivation

To set the client’s and users’ expectations for the precision of the product.

Examples

All monetary amounts shall be accurate to two decimal places.

Accuracy of road temperature readings shall be within ±2°C.

All answers shall be accurate and every wrong answer shall be marked as incorrect.

Considerations

If you have done any detailed work on definitions, then some precision requirements might be adequately defined by definitions in section 5.

You might consider which units the product is intended to use. Readers will recall the spacecraft that crashed on Mars when coordinates were sent as metric data rather than imperial data.

**The product might also need to keep accurate time, be synchronized with a time server, or work in UTC.**

Also, be aware that some currencies have no decimal places, such as the Japanese yen.

**A timer will need to be implemented.**

### Capacity Requirements

Content

This section specifies the volumes that the product must be able to deal with and the amount of data stored by the product.

Our product will have to deal with very large files, because there will be high graphics used. Therefore, we will be using computers solely for the purpose of making this game, nothing extra on them. As well as having large amounts of external hard drives, that can include terabytes of data.

Motivation

To ensure that the product is capable of processing the expected volumes.

Examples

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

During a launch period, the product shall cater for a maximum of 20 people to be in the inner chamber.

Our product can allow for no more than one user at a time, because we want the child to be focused on learning solo.

Fit Criterion

In this case, the requirement description is quantified, and thus can be tested.

## Dependability Requirements

### Reliability Requirements

Content

This section quantifies the necessary reliability of the product. The reliability is usually expressed as the allowable time between failures, or the total allowable failure rate.

We will have 2 weeks between testing each portion of the game, and daily meetings to talk about and try to fix issues with the product.

Motivation

It is critical for some products not to fail too often. This section allows you to explore the possibility of failure and to specify realistic levels of service. It also gives you the opportunity to set the client’s and users’ expectations about the expected frequency and significance of potential failures.

Examples

The product shall not fail more than once per day.

No data shall be lost or damaged in the event of a failure. ( This is an example of a ***fail-safe*** requirement, which states that the product is allowed to fail, but it must do so safely. )

All changes being made will be noted, big or small, so if failure occurs, the makers of the product can backtrack and see what they did wrong.

Considerations

Consider carefully whether the real requirement for your product is that it is available for use or that it does not fail at any time.

Consider also the cost of reliability and availability, and whether it is justified for your product.

### Availability Requirements

Content

This section quantifies the necessary availability of the product. The availability is usually expressed as the fraction of total time that the system is up and available for use.

Availability is a function of the mean time between failures, the mean time required to bring the system back up after a failure, and the mean time the system is expected to be down for routine maintenance.

Motivation

There is a subtle distinction between how often a system goes down ( reliability )3and how much total time it spends being down ( availability ). This section allows you to specify realistic expectations about the amount of time that the product will be available for use.

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 8:00 a.m. and 5:30 p.m.

The escalator shall run from 6 a.m. until 10 p.m. or the last flight arrives.

The product shall achieve 99 percent uptime.

Considerations

Consider carefully whether the real requirement for your product is that it is available for use or that it does not fail at any time.

Consider also the cost of reliability and availability, and whether it is justified for your product.

The sections on reliability and availability can sometimes be combined.

### Robustness or Fault-Tolerance Requirements

Content

Robustness specifies the ability of the product to continue to function under abnormal circumstances.

Motivation

To ensure that the product is able to provide some or all of its services after or during some abnormal happening in its environment.

Examples

The product shall continue to operate in local mode whenever it loses its link to the central server.

The product shall provide 10 minutes of emergency operation should it become disconnected from the electricity source.

Considerations

Abnormal happenings can almost be considered normal. Today’s products are so large and complex that there is a good chance that at any given time, one component will not be functioning correctly. Robustness requirements are intended to prevent total failure of the product.

You could also consider disaster recovery in this section. This plan describes the ability of the product to reestablish acceptable performance after faults or abnormal happenings.

### Safety-Critical Requirements

Content

Quantification of the perceived risk of damage to people, property, and environment. Different countries have different standards, so the fit criteria must specify precisely which standards the product must meet.

Motivation

To understand and highlight the damage that could potentially occur when using the product within the expected operational environment.

Examples

The product shall not emit noxious gases that damage people’s health.

The heat exchanger shall be shielded from human contact.

Fit Criterion

The product shall be certified to comply with the Health Department’s standard E110-98. It 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].

Considerations

The example requirements given here apply to some, but not all, products. It is not possible to give examples of every variation of safety-critical requirement. To make the template work in your environment, you should customize it by adding examples that are specific to your products.

Also, be aware that different countries have different safety standards and laws relating to safety. If you plan to sell your product internationally, you must be aware of these laws. A colleague has suggested that for electrical products, if you follow the German standards, the largest number of countries will be supported.

If you are building safety-critical systems, then the relevant safety-critical standards are already well specified. You will likely have safety experts on your staff. These experts are the best source of the relevant safety-critical requirements for your type of product. They will almost certainly have copious information that you can use.

Consult your legal department. Members of this department will be aware of the kinds of lawsuits that have resulted from product safety failure. This is probably the best starting place for generating relevant safety requirements.

## Maintainability and Supportability Requirements

### Maintenance Requirements

Content

A quantification of the time necessary to make specified changes to the product.

Motivation

To make everyone aware of the maintenance needs of the product.

Examples

New MIS reports must be available within one working week of the date when the requirements are agreed upon.

A new weather station must be able to be added to the system overnight.

Considerations

There may be special requirements for maintainability, such as that the product must be able to be maintained by its end users or by developers who are not the original developers. These requirements have an effect on the way that the product is developed. In addition, there may be requirements for documentation or training.

You might also consider writing testability requirements in this section.

### Supportability Requirements

Content

This specifies the level of support that the product requires. Support is often provided via a help desk. If people will provide support for the product, that service is considered part of the product: Are there any requirements for that support? You might also build support into the product itself, in which case this section is the place to write those requirements.

Motivation

To ensure that the support aspect of the product is adequately specified.

Considerations

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

### Adaptability Requirements

Content

Description of other platforms or environments to which the product must be ported.

Motivation

To quantify the client’s and users’ expectations about the platforms on which the product will be able to run.

Examples

The product is expected to run under Windows XP and Linux.

The product might eventually be sold in the Japanese market.

The product is designed to run in offices, but we intend to have a version running in restaurant kitchens.

Fit Criterion

Specification of system software on which the product must operate.

Specification of future environments in which the product is expected to operate.

Time allowed to make the transition.

Considerations

Question your marketing department to discover unstated assumptions that have been made about the portability of the product.

### Scalability or Extensibility Requirements

Content

This specifies the expected increases in size that the product must be able to handle. As a business grows (or is expected to grow), our software products must increase their capacities to cope with the new volumes.

Motivation

To ensure that the designers allow for future capacities.

Examples

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

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

### Longevity Requirements

Content

This specifies the expected lifetime of the product.

Motivation

To ensure that the product is built based on an understanding of expected return on investment.

Examples

The product shall be expected to operate within the maximum maintenance budget for a minimum of five years.

## Security Requirements

### Access Requirements

Content

Specification of who has authorized access to the product (both functionality and data), under what circumstances that access is granted, and to which parts of the product access is allowed.

Motivation

To understand the expectations for confidentiality aspects of the system.

Examples

Only direct managers can see the personnel records of their staff.

Only holders of current security clearance can enter the building.

Fit Criterion

System function name or system data name.

User roles and/or names of people who have clearance.

Considerations

Is there any data that management considers to be sensitive? 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 stating how you will design a solution to the security requirements. For instance, don’t “design a password system.” Your aim here is to identify the security requirement; the design will then come from this description.

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

### Integrity Requirements

Content

Specification of the required integrity of databases and other files, and of the product itself.

Motivation

To understand the expectations for the integrity of the product’s data. To specify what the product will do to ensure its integrity in the case of an unwanted happening such as attack from the outside or unintentional misuse by an authorized user.

Examples

The product shall prevent incorrect data from being introduced.

The product shall protect itself from intentional abuse.

Considerations

Organizations are relying more and more on their stored data. If this data should be come corrupt or incorrect—or disappear—then it could be a fatal blow to the organization. For example, almost half of small businesses go bankrupt after a fire destroys their computer systems. Integrity requirements are aimed at preventing complete loss, as well as corruption, of data and processes.

### Privacy Requirements

Content

Specification of what the product has to do to ensure the privacy of individuals about whom it stores information. The product must also ensure that all laws related to privacy of an individual’s data are observed.

Motivation

To ensure that the product complies with the law, and to protect the individual privacy of your customers. Few people today look kindly on organizations that do not observe their privacy.

Examples

The product shall make its users aware of its information practices before collecting data from them.

The product shall notify customers of changes to its information policy.

The product shall reveal private information only in compliance with the organization’s information policy.

The product shall protect private information in accordance with the relevant privacy laws and the organization’s information policy.

Considerations

Privacy issues may well have legal implications, and you are advised to consult with your organization’s legal department about the requirements to be written in this section.

Consider what notices you must issue to your customers before collecting their personal information. A notice might go so far as to warn customers that you intend to put a cookie in their computer. Also, do you have to do anything to keep customers aware that you hold their personal information?

Customers must always be in a position to give or withhold consent when their private data is collected or stored. Similarly, customers should be able to view any private data and, where appropriate, ask for correction of the data.

Also consider the integrity and security of private data—for example, when you are storing credit card information.

### Audit Requirements

Content

Specification of what the product has to do (usually retain records) to permit the required audit checks.

Motivation

To build a system that complies with the appropriate audit rules.

Considerations

This section may have legal implications. You are advised to seek the approval of your organization’s auditors regarding what you write here.

You should also consider whether the product should retain information on who has used it. The intention is to provide security such that a user may not later deny having used the product or participated in some form of transaction using the product.

### Immunity Requirements

Content

The requirements for what the product has to do to protect itself from infection by unauthorized or undesirable software programs, such as viruses, worms, and Trojan horses, among others.

Motivation

To build a product that is as secure as possible from malicious interference.

Considerations

Each day brings more malevolence from the unknown, outside world. People buying software, or any other kind of product, expect that it can protect itself from outside interference.

## Usability and Humanity Requirements

This section is concerned with requirements that make the product usable and ergonomically acceptable to its hands-on users.

### Ease of Use Requirements

Content

This section describes your client’s aspirations for how easy it is 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.

The usability requirements should cover properties such as these:

● Efficiency of use: How quickly or accurately the user can use the product.

● Ease of remembering: How much the casual user is expected to remember about using the product.

● Error rates: For some products it is crucial that the user commits very few, or no, errors.

● Overall satisfaction in using the product: This is especially important for commercial, interactive products that face a lot of competition. Web sites are a good example.

● Feedback: How much feedback the user needs to feel confident that the product is actually accurately doing what the user expects. The necessary degree of feedback will be higher for some products (e.g., safety-critical products) than for others.

Motivation

To guide the product’s designers toward building a product that meets the expectations of its eventual users.

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.

Fit Criterion

These examples may seem simplistic, but they do express the intention of the client. To completely specify what is meant by the requirement, you must add a measurement against which it can be tested—that is, a fit criterion. Here are the fit criteria for the preceding examples:

Eighty percent of a test panel of 11-year-old children 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 1 percent.

An anonymous survey shall show that 75 percent of the intended users are regularly using the product after a three-week familiarization period.

Considerations

Refer to section 3, Users of the Product, to ensure that you have considered the usability requirements from the perspective of all the different types of users.

It may be necessary to have special consulting sessions with your users and your client to determine whether any special usability considerations must be built into the product.

You could also consider consulting a usability laboratory experienced in testing the usability of products that have a project situation (sections 1–7 of this template) similar to yours.

### Personalization and Internationalization Requirements

Content

This section describes the way in which the product can be altered or configured to take into account the user’s personal preferences or choice of language.

The personalization requirements should cover issues such as the following:

● Languages, spelling preferences, and language idioms

● Currencies, including the symbols and decimal conventions

● Personal configuration options

Motivation

To ensure that the product’s users do not have to struggle with, or meekly accept, the builder’s cultural conventions.

Examples

The product shall retain the buyer’s buying preferences.

The product shall allow the user to select a chosen language.

Considerations

Consider the country and culture of the potential customers and users of your product. Any out-of-country users will welcome the opportunity to convert to their home spelling and expressions.

By allowing users to customize the way in which they use the product, you give them the opportunity to participate more closely with your organization as well as enjoy their own personal user experience.

You might also consider the configurability of the product. Configurability allows different users to have different functional variations of the product.

### Learning Requirements

Content

Requirements specifying how easy it should be to learn to use the product. This learning curve ranges from zero time for products intended for placement in the public domain (e.g., a parking meter or a web site) to a considerable amount of time for complex, highly technical products. (We know of one product where it was necessary for graduate engineers to spend 18 months in a training program before being qualified to use the product.)

Motivation

To quantify the amount of time that your client feels is allowable before a user can successfully use the product. This requirement guides designers to understand how users will learn the product. For example, 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

The product shall be easy for an engineer to learn.

A clerk shall be able to be productive within a short time.

The product shall be able to be used by members of the public who will receive no training before using it.

The product shall be used by engineers who will attend five weeks of training before using the product.

Fit Criterion

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.

Considerations

Refer to section 3, Users of the Product, to ensure that you have considered the ease of learning requirements from the perspective of all the different types of users.

### Understandability and Politeness Requirements

This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end users.

Content

This specifies the requirement for the product to be understood by its users. While “usability” refers to ease of use, efficiency, and similar characteristics, “understandability” determines whether the users instinctively know what the product will do for them and how it fits into their view of the world. You can think of understandability as the product being polite to its users and not expecting them to know or learn things that have nothing to do with their business problem.

Motivation

To avoid forcing users to learn terms and concepts that are part of the product’s internal construction and are not relevant to the users’ world. To make the product more comprehensible and thus more likely to be adopted by its intended users.

Examples

The product shall use symbols and words that are naturally understandable by the user community.

The product shall hide the details of its construction from the user.

Considerations

Refer to section 3, Users of the Product, and consider the world from the point of view of each of the different types of users.

### Accessibility Requirements

Content

The requirements for how easy it should be for people with common disabilities to access the product. These disabilities might be related to physical disability or visual, hearing, cognitive, or other abilities.

Motivation

In many countries it is required that some products be made available to the disabled. In any event, it is self-defeating to exclude this sizable community of potential customers.

Examples

The product shall be usable by partially sighted users.

The product shall conform to the Americans with Disabilities Act.

Considerations

Some users have disabilities other than the commonly described ones. In addition, some partial disabilities are fairly common. A simple, and not very consequential, example is that approximately 20 percent of males are red-green colorblind.

### User Documentation Requirements

Content

List of the user documentation to be supplied as part of the product.

Motivation

To set expectations for the documentation and to identify who will be responsible for creating it.

Examples

Technical specifications to accompany the product.

User manuals.

Service manuals (if not covered by the technical specification).

Emergency procedure manuals (e.g., the card found in airplanes).

Installation manuals.

Considerations

Which documents do you need to deliver, and to whom? Bear in mind that the answer to this questions depends on your organizational procedures and roles.

For each document, consider these issues:

● The purpose of the document

● The people who will use the document

● Maintenance of the document

What level of documentation is expected? Will the users be involved in the production of the documentation? Who will be responsible for keeping the documentation up-to-date? What form will the documentation take?

### Training Requirements

Content

A description of the training needed by users of the product.

Motivation

To set expectations for the training. To identify who is responsible for creating and providing that training.

Considerations

What training will be necessary? Who will design the training? Who will provide the training?

## Look and Feel Requirements

### Appearance Requirements

Content

The section contains requirements relating to the spirit of the product. Your client may have made particular demands for the product, such as corporate branding, colors to be used, and so on. This section captures the requirements for the appearance. Do not attempt to design it until the appearance requirements are known.

Motivation

To ensure that the appearance of the product conforms to the organization’s expectations.

Examples

The product shall be attractive to a teenage audience.

The product shall comply with corporate branding standards.

Fit Criterion

A sampling of representative teenagers shall, without prompting or enticement, start using the product within four minutes of their first encounter with it.

The office of branding shall certify the product complies with the current standards.

Considerations

Even if you are using prototypes, it is important to understand the requirements for the appearance. The prototype is used to help elicit requirements; it should not be thought of as a substitute for the requirements.

### Style Requirements

Content

Requirements that specify the mood, style, or feeling of the product, which influences the way a potential customer will see the product. Also, the stakeholders’ intentions for the amount of interaction the user is to have with the product.

In this section, you would also describe the appearance of the package if this is 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. Keep in mind the European laws on packaging, which require that the package not be significantly larger than the product it encloses.

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

Motivation

Given the state of today’s market and people’s expectations, we cannot afford to build products that have the wrong style. Once the functional requirements are satisfied, it is often the appearance and style of products that determine whether they are successful. Your task in this section is to determine precisely how the product shall appear to its intended consumer.

Example

The product shall appear authoritative.

Fit Criterion

After their first encounter with the product, 70 percent of representative potential customers shall agree they feel they can trust the product.

Considerations

The look and feel requirements specify your client’s vision of the product’s appearance. The requirements may at first seem to be rather vague (e.g., “conservative and professional appearance”), but these will be quantified by their fit criteria. The fit criteria give you the opportunity to extract from your client precisely what is meant, and give the designer precise instructions on what he is to accomplish.

## Operational and Environmental Requirements

### Expected Physical Environment

Content

This section specifies the physical environment in which the product will operate.

Motivation

To highlight conditions that might need special requirements, preparations, or training. These requirements ensure that the product is fit to be used in its intended environment.

Examples

The product shall be used by a worker, standing up, outside in cold, rainy conditions.

The product shall be used in noisy conditions with a lot of dust.

The product shall be able to fit in a pocket or purse.

The product shall be usable in dim light.

The product shall not be louder than the existing noise level in the environment.

Considerations

The work environment: Is the product to operate in some unusual environment? Does this lead to special requirements? Also see section 11, Usability and Humanity Requirements.

### Requirements for Interfacing with Adjacent Systems

Content

This section describes the requirements to interface with partner applications and/or devices that the product needs to successfully operate.

Motivation

Requirements for the interfaces to other applications often remain undiscovered until implementation time. Avoid a high degree of rework by discovering these requirements early.

Examples

The products shall work on the last four releases of the five most popular browsers.

The new version of the spreadsheet must be able to access data from the previous two versions.

Our product must interface with the applications that run on the remote weather stations.

Fit Criterion

For each inter-application interface, specify the following elements:

● The data content

● The physical material content

● The medium that carries the interface

● The frequency

● The volume

### Productization Requirements

Content

Any requirements that are necessary to make the product into a distributable or salable item. It is also appropriate to describe here the operations needed to install a software product successfully.

Motivation

To ensure that if work must be done to get the product out the door, then that work becomes part of the requirements. Also, to quantify the client’s and users’ expectations about the amount of time, money, and resources they will need to allocate to install the product.

Examples

The product shall be distributed as a ZIP file.

The product shall be able to be installed by an untrained user without recourse to separately printed instructions.

The product shall be of a size such that it can fit on one CD.

Considerations

Some products have special needs to turn them into a salable or usable product. You might consider that the product has to be protected such that only paid-up customers can access it.

Ask questions of your marketing department to discover unstated assumptions that have been made about the specified environment and the customers’ expectations of how long installation will take and how much it will cost.

Most commercial products have some needs in this area.

### Release Requirements

Content

Specification of the intended release cycle for the product and the form that the release shall take.

Motivation

To make everyone aware of how often you intend to produce new releases of the product.

Examples

The maintenance releases will be offered to end users once a year.

Each release shall not cause previous features to fail.

Fit Criterion

Description of the type of maintenance plus the amount of effort budgeted for it.

Considerations

Do you have any existing contractual commitments or maintenance agreements that might be affected by the new product?

## Cultural and Political Requirements

### Cultural Requirements

Content

This section contains requirements that are specific to the sociological factors that affect the acceptability of the product. If you are developing a product for foreign markets, then these requirements are particularly relevant.

Motivation

To bring out in the open requirements that are difficult to discover because they are outside the cultural experience of the developers.

Examples

The product shall not be offensive to religious or ethnic groups.

The product shall be able to distinguish between French, Italian, and British road-numbering systems.

The product shall keep a record of public holidays for all countries in the European Union and for all states in the United States.

Considerations

Question whether the product is intended for a culture other than the one with which you are familiar. Ask whether people in other countries or in other types of organizations will use the product. Do these people have different habits, holidays, superstitions, or cultural norms that do not apply to your own culture? Are there colors, icons, or words that have different meanings in another cultural environment?

### Political Requirements

Content

This section contains requirements that are specific to the political factors that affect the acceptability of the product.

Motivation

To understand requirements that sometimes appear irrational.

Examples

The product shall be installed using only American-made components.

The product shall make all functionality available to the CEO.

Considerations

Did you intend to develop the product on a Macintosh, when the office manager has laid down an edict that only Windows machines are permitted?

Is a director also on the board of a company that manufactures products similar to the one that you intend to build?

Whether you agree with these political requirements has little bearing on the outcome. The reality is that the system has to comply with political requirements even if you can find a better, more efficient, or more economical solution. A few probing questions here may save some heartache later.

The political requirements might be purely concerned with the politics inside your organization. However, in other situations you may need to consider the politics inside your customers’ organizations or the national politics of the country.

## Legal Requirements

### Compliance Requirements

Content

A statement specifying the legal requirements for this system.

Motivation

To comply with the law so as to avoid later delays, lawsuits, and legal fees.

Examples

Personal information shall be implemented so as to comply with the Data Protection Act.

Fit Criterion

Lawyers’ opinion that the product does not break any laws.

Considerations

Consider consulting lawyers to help identify the legal requirements.

Are there any copyrights or other intellectual property that must be protected? Conversely, do any competitors have copyrights on which you might be in danger of infringing?

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

The Sarbanes-Oxley (SOX) Act, the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act may have implications for you. Check with your company lawyer.

Might any pending legislation affect the development of this system?

Are there any aspects of criminal law you should consider?

Have you considered the tax laws that affect your product?

Are there any labor laws (e.g., working hours) relevant to your product?

### Standards Requirements

Content

A statement specifying applicable standards and referencing detailed standards descriptions. This does not refer to the law of the land—think of it as an internal law imposed by your company.

Motivation

To comply with standards so as to avoid later delays.

Example

The product shall comply with MilSpec standards.

The product shall comply with insurance industry standards.

The product shall be developed according to SSADM standard development steps.

Fit Criterion

The appropriate standard-keeper certifies that the standard has been adhered to.

Considerations

It is not always apparent that there are applicable standards because their existence is often taken for granted. Consider the following:

● Do any industry bodies have applicable standards?

● Does the industry have a code of practice, watchdog, or ombudsman?

● Are there any special development steps for this type of product?

# Design

## System Design

### Design goals

Content

Design goals are important properties of the system to be optimized, and which may affect the overall design of the system. For example computer games place a higher priority on speed than accuracy, and so the physics engine for a computer game may make some rough approximations and assumptions that allow it to run as fast as possible while sacrificing accuracy, whereas the physics calculations performed by NASA must be much more rigorously correct, even at the expense of speed.

Note an important difference between design goals and requirements: Requirements include specific values that must be met in order for the product to be acceptable to the client, whereas design goals are properties that the designers strive to make "as good as possible", without specific criteria for acceptability. ( Note also that the same property may appear in both a requirement and a design goal, so a design goal may be to make the system run as fast as possible, with a requirement that says any speed below a certain specified threshold is unacceptable. )

Motivation

Considerations

Example

## Current Software Architecture

Content

Motivation

Considerations

Example

## Proposed Software Architecture

### Overview

Content

Motivation

Considerations

Example

### Class Diagrams

Content

Motivation

Considerations

Example

### Dynamic Model

Content

Include sequence diagrams of each use-case here. This is a first step towards identifying preliminary objects. ( If the sequence diagram would be too big to fit, then it can either be broken down into pieces or a communication diagram can be used in its place. )

Depending on the particular design, this section may also include finite state diagrams.

Motivation

Considerations

Example

### Subsystem Decomposition

Content

Motivation

Considerations

Example

### Hardware / software mapping

Content

Motivation

Considerations

Example

### Data Dictionary

Content

Motivation

Considerations

Example

### Persistent Data management

Content

Motivation

Considerations

Example

### Access control and security

Content

Motivation

Considerations

Example

### Global software control

Content

Motivation

Considerations

Example

### Boundary conditions

Content

Motivation

Considerations

Example

## Subsystem services

Content

Motivation

Considerations

Example

## User Interface

Content

Motivation

Considerations

Example

## Object Design

### Object Design trade-offs

Content

Motivation

Considerations

Example

### Interface Documentation guidelines

Content

Motivation

Considerations

Example

### Packages

Content

Motivation

Considerations

Example

### Class Interfaces

Content

Motivation

Considerations

Example

# Test Plans

## Features to be tested / not to be tested

Content

Motivation

Considerations

Example

## Pass/Fail Criteria

Content

Motivation

Considerations

Example

## Approach

Content

Motivation

Considerations

Example

## Suspension and resumption

Content

Motivation

Considerations

Example

## Testing materials ( hardware / software requirements )

Content

Motivation

Considerations

Example

## Test cases

Content

Motivation

Considerations

Example

## Testing schedule

Content

Motivation

Considerations

Example

# Project Issues

## Open Issues

Issues that have been raised and do not yet have a conclusion.

Content

A statement of factors that are uncertain and might make significant difference to the product.

Motivation

To bring uncertainty out in the open and provide objective input to risk analysis.

Examples

Our investigation into whether the new version of the processor will be suitable for our application is not yet complete.

The government is planning to change the rules about who is responsible for gritting the motorways, but we do not know what those changes might be.

Considerations

Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations or systems on your context diagram? Are there any legislative changes that might affect your system? Are there any rumors about your hardware or software suppliers that might have an impact?

## Off-the-Shelf Solutions

### Ready-Made Products

Content

List of existing products that should be investigated as potential solutions. Reference any surveys that have been done on these products.

Motivation

To give consideration to whether a solution can be bought.

Considerations

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

Also consider whether some products must not be used.

### Reusable Components

Content

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

Motivation

Reuse rather than reinvention.

### Products That Can Be Copied

Content

List of other similar products or parts of products that you can legally copy or easily modify.

Motivation

Reuse rather than reinvention.

Examples

Another electricity company has built a customer service system. Its hardware is different from ours, but we could buy its specification and cut our analysis effort by approximately 60 percent.

Considerations

While a ready-made solution may not exist, perhaps something, in its essence, is similar enough that you could copy, and possibly modify, it to better effect than starting from scratch. This approach is potentially dangerous because it relies on the base system being of good quality.

This question should always be answered. The act of answering it will force you to look at other existing solutions to similar problems.

## New Problems

### Effects on the Current Environment

Content

A description of how the new product will affect the current implementation environment. This section should also cover things that the new product should *not* do.

Motivation

The intention is to discover early any potential conflicts that might otherwise not be realized until implementation time.

Examples

Any change to the scheduling system will affect the work of the engineers in the divisions and the truck drivers.

Considerations

Is it possible that the new system might damage some existing system? Can people be displaced or otherwise affected by the new system?

These issues require a study of the current environment. A model highlighting the effects of the change is a good way to make this information widely understandable.

### Effects on the Installed Systems

Content

Specification of the interfaces between new and existing systems.

Motivation

Very rarely is a new development intended to stand completely alone. Usually the new system must coexist with some older system. This question forces you to look carefully at the existing system, examining it for potential conflicts with the new development.

### Potential User Problems

Content

Details of any adverse reaction that might be suffered by existing users.

Motivation

Sometimes existing users are using a product in such a way that they will suffer ill effects from the new system or feature. Identify any likely adverse user reactions, and determine whether we care about those reactions and what precautions we will take.

### Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Content

Statement of any potential problems with the new automated technology or new ways of structuring the organization.

Motivation

The intention is to make early discovery of any potential conflicts that might otherwise not be realized until implementation time.

Examples

The planned new server is not powerful enough to cope with our projected growth pattern.

The size and weight of the new product do not fit into the physical environment.

The power capabilities will not satisfy the new product’s projected consumption.

Considerations

This requires a study of the intended implementation environment.

### Follow-Up Problems

Content

Identification of situations that we might not be able to cope with.

Motivation

To guard against situations where the product might fail.

Considerations

Will we create a demand for our product that we are not able to service? Will the new system cause us to run afoul of laws that do not currently apply? Will the existing hardware cope?

There are potentially hundreds of unwanted effects. It pays to answer this question very carefully.

## Tasks

### Project Planning

Content

Details of the life cycle and approach that will be used to deliver the product. A high-level process diagram showing the tasks and the interfaces between them is a good way to communicate this information.

Motivation

To specify the approach that will be taken to deliver the product so that everyone has the same expectations.

Considerations

Depending on the maturity level of your process, the new product will be developed using your standard approach. However, some circumstances are unique to a particular product and will necessitate changes to your life cycle. While these considerations are not product requirements, they are needed if the product is to be successfully developed.

If possible, attach an estimate of the time and resources needed for each task based on the requirements that you have specified. Attach your estimates to the events, use cases, and/or functions that you specified in sections 8 and 9.

Do not forget issues related to data conversion, user training, and cutover. These needs are usually ignored when projects set implementation dates.

### Planning of the Development Phases

Content

Specification of each phase of development and the components in the operating environment.

Motivation

To identify the phases necessary to implement the operating environment for the new system so that the implementation can be managed.

Fit Criterion

Name of the phase.

Required operational date.

Operating environment components included.

Functional requirements included.

Nonfunctional requirements included.

Considerations

Identify which hardware and other devices are necessary for each phase of the new system. This list may not be known at the time of the requirements process, as these devices may be decided at design time.

## Migration to the New Product

### Requirements for Migration to the New Product

Content

A list of the conversion activities. Timetable for implementation.

Motivation

To identify conversion tasks as input to the project planning process.

Considerations

Will you use a phased implementation to install the new system? If so, describe which requirements will be implemented by each of the major phases.

What kind of data conversion is necessary? Must special programs be written to transport data from an existing system to the new one? If so, describe the requirements for these programs here.

What kind of manual backup is needed while the new system is installed?

When are each of the major components to be put in place? When are the phases of the implementation to be released?

Is there a need to run the new product in parallel with the existing product?

Will we need additional or different staff?

Is any special effort needed to decommission the old product?

This section is the timetable for implementation of the new system.

### Data That Has to Be Modified or Translated for the New System

Content

List of data translation tasks.

Motivation

To discover missing tasks that will affect the size and boundaries of the project.

Fit Criterion

Description of the current technology that holds the data.

Description of the new technology that will hold the data.

Description of the data translation tasks.

Foreseeable problems.

Considerations

Every time you make an addition to your dictionary (see section 5), ask this question: Where is this data currently held, and will the new system affect that implementation?

## Risks

All projects involve risk—namely, the risk that something will go wrong. Risk is not necessarily a bad thing, as no progress is made without taking some risk. However, there is a difference between unmanaged risk—say, shooting dice at a craps table—and managed risk, where the probabilities are well understood and contingency plans are made. Risk is only a bad thing if the risks are ignored and they become problems. Risk management entails assessing which risks are most likely to apply to the project, deciding a course of action if they become problems, and monitoring projects to give early warnings of risks becoming problems.

This section of your specification should contain a list of the most likely risks and the most serious risks for your project. For each risk, include the probability of that risk becoming a problem. Capers Jones’s *Assessment and Control of Software Risks* (Prentice-Hall, Englewood Cliffs, N.J., 1994) gives comprehensive lists of risks and their probabilities; you can use these lists as a starting point. For example, Jones cites the following risks as being the most serious:

• Inaccurate metrics

• Inadequate measurement

• Excessive schedule pressure

• Management malpractice

• Inaccurate cost estimating

• Silver bullet syndrome

• Creeping user requirements

• Low quality

• Low productivity

• Cancelled projects

Use your knowledge of the requirements as input to discover which risks are most relevant to your project.

It is also useful input to project management if you include the impact on the schedule, or the cost, if the risk does become a problem.

## Costs

For details on how to estimate requirements effort and costs, refer to Appendix C Function Point Counting: A Simplified Introduction

The other cost of requirements is the amount of money or effort that you have to spend building them into a product. Once the requirements specification is complete, you can use one of the estimating methods to assess the cost, expressing the result as a monetary amount or time to build.

There is no best method to use when estimating. Keep in mind, however, that your estimates should be based on some tangible, countable artifact. If you are using this template, then, as a result of doing the work of requirements specification, you are producing many measurable deliverables. For example:

● Number of input and output flows on the work context

● Number of business events

● Number of product use cases

● Number of functional requirements

● Number of nonfunctional requirements

● Number of requirements constraints

● Number of function points

The more detailed the work you do on your requirements, the more accurate your deliverables will be. Your cost estimate is the amount of resources you estimate each type of deliverable will take to produce within your environment. You can create some very early cost estimates based on the work context. At that stage, your knowledge of the work will be general, and you should reflect this vagueness by making the cost estimate a range rather than a single figure.

As you increase your knowledge of the requirements, we suggest you try using function point counting—not because it is an inherently superior method, but because it is so widely accepted. So much is known about function point counting that it is possible to make easy comparisons with other products and other installations’ productivity.

It is important that your client be told at this stage what the product is likely to cost. You usually express this amount as the total cost to complete the product, but you may also find it advantageous to point out the cost of the requirements effort, or the costs of individual requirements.

Whatever you do, do not leave the costs in the lap of hysterical optimism. Make sure that this section includes meaningful numbers based on tangible deliverables.

## Waiting Room

Requirements that will not be part of the next release. These requirements might be included in future releases of the product.

Content

Any type of requirement.

Motivation

To allow requirements to be gathered, even though they cannot be part of the current development. To ensure that good ideas are not lost.

Considerations

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 holds these requirements in waiting. The intention is to avoid stifling the creativity of your users and clients, by using a repository to retain future requirements. You are also managing expectations by making it clear that you take these requirements seriously, although they will not be part of the agreed-upon product.

Many people use the waiting room as a way of planning future versions of the product. Each requirement in the waiting room is tagged with its intended version number. As a requirement progresses closer to implementation, then you can spend more time on it and add details such as the cost and benefit attached to that requirement.

You might also prioritize the contents of your waiting room. “Low-hanging fruit”—requirements that provide a high benefit at a low cost of implementation—are the highest-ranking candidates for the next release. You would also give a high waiting room rank to requirements for which there is a pent-up demand.

## Ideas for Solutions

When you gather requirements, you focus on finding out what the real requirements are and try to avoid coming up with solutions. However, when creative people start to think about a problem, they always generate ideas about potential solutions. This section of the template is a place to put those ideas so that you do not forget them and so that you can separate them from the real business requirements.

Content

Any idea for a solution that you think is worth keeping for future consideration. This can take the form of rough notes, sketches, pointers to other documents, pointers to people, pointers to existing products, and so on. The aim is to capture, with the least amount of effort, an idea that you can return to later.

Motivation

To make sure that good ideas are not lost. To help you separate requirements from solutions.

Considerations

While you are gathering requirements, you will inevitably have solution ideas; this section offers a way to capture them. Bear in mind that this section will not necessarily be included in every document that you publish.

## Project Retrospective

Content

At the end of every project you should reflect upon what methods were used that worked out well and should be repeated in the future, and also what methods did not work out well and should be avoided. Any recommendations, suggestions, or ideas for how to do things better in the future should also be documented

Motivation

To learn from experience, and to continually strive for process improvement.

Considerations

When things don't go well, it is important to distinguish whether the methods themselves were poor, or simply poorly implemented in this particular case, or whether they just weren't right for this particular project / group of engineers.

# Glossary

The glossary defines terms that may not be familiar to all readers. This is especially important if the document is expected to reach a wide and varied audience, such as school children. The glossary may be placed at either the beginning or the end of the document.

**Flotsam:** Any part of a ship or its cargo found floating on the water, whether it was deliberately or accidentally lost by its original owners.

**Jetsam:** Any part of a ship or its cargo that is deliberately cast off ( jettisoned ) by its original owners, generally in order to lighten the ship, whether it floats or sinks.

# References / Bibliography

This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the “Insert Citation” and “Bibliography” buttons in the “Citations & Bibliography” section under the “References” tab of MS Word. Creating new citations will not update this list unless you click on it and select “Update Field”. You may need to reset the style for this paragraph to “normal” after updating.

|  |  |
| --- | --- |
| [1] | Robertson and Robertson, Mastering the Requirements Process. |
| [2] | A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013. |
| [3] | J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012. |
| [4] | M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004. |

# Index

This section provides an index to the report. The sample below was generated using the “Mark Entry” and “Insert Index” items from the “Index” section on the “References” tab, and can be automatically updated by right clicking on the table below and selecting “Update Field”. To remove marked entries from the document, toggle the display of hidden paragraph marks ( the paragraph button on the “Home” tab ), and remove the tags shown with XE in { curly braces. }

Design 61, 65

Requirements 36, 53, 60

Test 66, 67

Refrences:

<http://www.vrroom.buzz/sites/default/files/chinese_students_2.jpg>