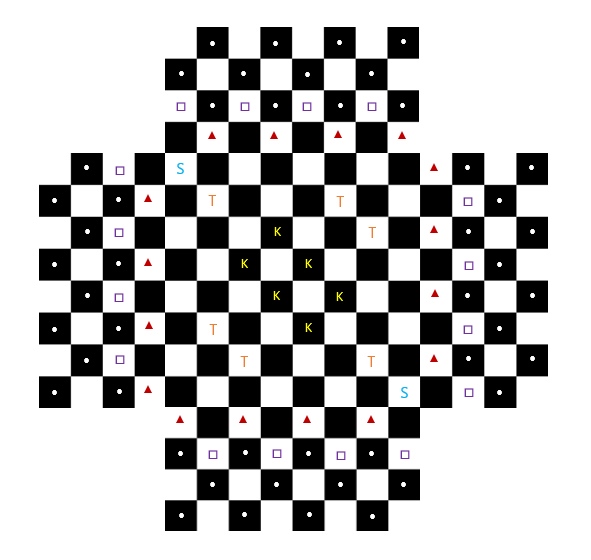
Checker Wars

Project Description Report

**Group 16: Alex Choi, Andrew Macatangay, Sam Alammar, Luke Austin**



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

***Prepared by***

***John T. Bell***

***for use in* CS 440**

**University of Illinois Chicago**

**October 2020**

### *****REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT*****

### ***How to Use This Document***

*This document is intended as a sample template that can be copied and edited to suit a particular software engineering project. It was assembled from a combination of documents [1], [2], and [3].*

***Styles***

*This document was written in Microsoft Word, and makes heavy use of styles. The styles dialog is initially located on the menu bar under the “Home” tab in MS Word. It is recommended that the styles dialog be pulled off into a separate window when working on formatting of the document. If each paragraph is assigned a style, then modifying that particular style will affect all paragraphs in the document having the same style.*

*The table of contents uses the document headings and sub-headings to automatically generate table of contents information.*

***Tracking Changes and Multiple Authors***

*The “Review” tab in MS Word contains several tools that are of particular use when editing large documents, particularly when multiple authors are involved:*

*The “Tracking” section allows you to track the ( proposed ) changes to a document, and to step through each proposed change to either accept or reject the proposed changes.*

*The “Compare” section allows you to merge changes proposed by different authors, ( which will be marked in separate colors for identification ), and then to use the change tracking tools described above to accept or deny each change.*

*The recommended procedure is to start with each author having a copy of a base document, ( possibly this template. ) Then each author changes the section(s) they are responsible for, and submits their changed version to one person who acts as the overall document editor. This author merges the changes, selectively accepts or rejects each change, and then distributes a new base document to all authors for the next round of changes. It is also possible to merge the changes and then distribute the document, so that all authors can review the proposed changes. ( The latter approach may be appropriate for documents such as bylaws, in which the changes must be approved by a committee or a vote before they can be accepted. )*

***Dealing With Material that is Unwanted ( Right Now )***

*Much of this document includes material that is not needed for every project, and/or which may not have been written yet, and so should be removed before printing or distributing the document. There are several ways to do this, however each has their drawbacks:*

1. *Delete the material completely. The drawback is that now it is completely gone, and the only way to get it back is to copy it from some other document, if that is even available.*
2. *Change the font to "Hidden". The drawback is that this does not affect the numbering of sections, either in the text or in the table of contents. However the original style information is retained, so when the text is unhidden, individual paragraphs do not need to be restyled.*
3. *Change the style to "Hidden". This does cause the document to renumber properly, ( because the paragraphs are no longer numbered paragraphs ), but all the original style information is gone, so if it is unhidden later, then all the styles of all the unhidden paragraphs will have to be restored manually.*
4. *Move the text, say to the end of the document, and then do not print or distribute the material at the end of the document. If this is all that is done, then this material will continue to appear in the table of contents. However if the font is also changed to "hidden", then the material will not appear in the TOC, but all the original style information will be retained. The drawback to moving the text is that now you need to keep track of where the information was moved from, in case you ever want to move it back.*

*Hidden text can be made visible by selecting the backwards "P" paragraph mark on the "Home" tab.*

***Table of Contents***

*The table of contents lists the sections of the document and the page upon which each section starts. The table of contents may or may not include subsections, etc.*

*Microsoft Word ( and many similar programs ) has the ability to generate a table of contents automatically from section headings, and to update it when the document changes. The table of contents included in this template is automatically generated by MS Word, and can be updated by selecting the table and clicking on the menu that appears above the table at that time.*

***Note: Remove all instructional materials before finalizing and submitting this document, including this entire page and the one before it.***

**Table of Contents**

[List of Figures 7](#_Toc52556536)

[List of Tables 8](#_Toc52556537)

[I Project Description 9](#_Toc52556538)

[1 Project Overview 9](#_Toc52556539)

[2 The Purpose of the Project 9](#_Toc52556540)

[2a The User Business or Background of the Project Effort 9](#_Toc52556541)

[2b Goals of the Project 9](#_Toc52556542)

[2c Measurement 9](#_Toc52556543)

[3 The Scope of the Work 9](#_Toc52556544)

[3a The Current Situation 9](#_Toc52556545)

[3b The Context of the Work 10](#_Toc52556546)

[3c Work Partitioning 11](#_Toc52556547)

[3d Competing Products 11](#_Toc52556548)

[4 The Scope of the Product 12](#_Toc52556549)

[4a Scenario Diagram(s) 12](#_Toc52556550)

[4b Product Scenario List 13](#_Toc52556551)

[4c Individual Product Scenarios 13](#_Toc52556552)

[5 Stakeholders 14](#_Toc52556553)

[5a The Client 14](#_Toc52556554)

[5b The Customer 14](#_Toc52556555)

[5c Hands-On Users of the Product 14](#_Toc52556556)

[5d Other Stakeholders 15](#_Toc52556557)

[5e User Participation 15](#_Toc52556558)

[6 Mandated Constraints 15](#_Toc52556559)

[6a Solution Constraints 15](#_Toc52556560)

[6b Implementation Environment of the Current System 15](#_Toc52556561)

[6c Partner or Collaborative Applications 16](#_Toc52556562)

[6d Off-the-Shelf Software 16](#_Toc52556563)

[6e Anticipated Workplace Environment 16](#_Toc52556564)

[6f Schedule Constraints 16](#_Toc52556565)

[6g Budget Constraints 16](#_Toc52556566)

[7 Naming Conventions and Definitions 16](#_Toc52556567)

[7a Definitions of Key Terms 16](#_Toc52556568)

[7b UML and Other Notation Used in This Document 17](#_Toc52556569)

[7c Data Dictionary for Any Included Models 17](#_Toc52556570)

[8 Relevant Facts and Assumptions 17](#_Toc52556571)

[8a Facts 17](#_Toc52556572)

[8b Assumptions 17](#_Toc52556573)

[II Requirements 17](#_Toc52556574)

[9 Product Use Cases 17](#_Toc52556575)

[9a Use Case Diagrams 18](#_Toc52556576)

[9b Product Use Case List 19](#_Toc52556577)

[9c Individual Product Use Cases 19](#_Toc52556578)

[10 Functional Requirements 21](#_Toc52556579)

[11 Data Requirements 22](#_Toc52556580)

[12 Performance Requirements 23](#_Toc52556581)

[12a Speed and Latency Requirements 23](#_Toc52556582)

[12b Precision or Accuracy Requirements 24](#_Toc52556583)

[12c Capacity Requirements 25](#_Toc52556584)

[13 Dependability Requirements 26](#_Toc52556585)

[13a Reliability Requirements 26](#_Toc52556586)

[13b Availability Requirements 27](#_Toc52556587)

[13c Robustness or Fault-Tolerance Requirements 28](#_Toc52556588)

[13d Safety-Critical Requirements 29](#_Toc52556589)

[14 Maintainability and Supportability Requirements 30](#_Toc52556590)

[14a Maintenance Requirements 30](#_Toc52556591)

[14b Supportability Requirements 31](#_Toc52556592)

[14c Adaptability Requirements 32](#_Toc52556593)

[14d Scalability or Extensibility Requirements 33](#_Toc52556594)

[14e Longevity Requirements 33](#_Toc52556595)

[15 Security Requirements 34](#_Toc52556596)

[15a Access Requirements 34](#_Toc52556597)

[15b Integrity Requirements 35](#_Toc52556598)

[15c Privacy Requirements 36](#_Toc52556599)

[15d Audit Requirements 37](#_Toc52556600)

[15e Immunity Requirements 38](#_Toc52556601)

[16 Usability and Humanity Requirements 38](#_Toc52556602)

[16a Ease of Use Requirements 38](#_Toc52556603)

[16b Personalization and Internationalization Requirements 40](#_Toc52556604)

[16c Learning Requirements 41](#_Toc52556605)

[16d Understandability and Politeness Requirements 42](#_Toc52556606)

[16e Accessibility Requirements 43](#_Toc52556607)

[16f User Documentation Requirements 44](#_Toc52556608)

[16g Training Requirements 45](#_Toc52556609)

[17 Look and Feel Requirements 46](#_Toc52556610)

[17a Appearance Requirements 46](#_Toc52556611)

[17b Style Requirements 47](#_Toc52556612)

[18 Operational and Environmental Requirements 48](#_Toc52556613)

[18a Expected Physical Environment 48](#_Toc52556614)

[18b Requirements for Interfacing with Adjacent Systems 49](#_Toc52556615)

[18c Productization Requirements 50](#_Toc52556616)

[18d Release Requirements 51](#_Toc52556617)

[19 Cultural and Political Requirements 52](#_Toc52556618)

[19a Cultural Requirements 52](#_Toc52556619)

[19b Political Requirements 53](#_Toc52556620)

[20 Legal Requirements 54](#_Toc52556621)

[20a Compliance Requirements 54](#_Toc52556622)

[20b Standards Requirements 55](#_Toc52556623)

[21 Requirements Acceptance Tests 56](#_Toc52556624)

[21a Requirements – Test Correspondence Summary 56](#_Toc52556625)

[21b Acceptance Test Descriptions 57](#_Toc52556626)

[III Design 57](#_Toc52556627)

[22 Design Goals 57](#_Toc52556628)

[23 Current System Design 57](#_Toc52556629)

[24 Proposed System Design 57](#_Toc52556630)

[24a Initial System Analysis and Class Identification 58](#_Toc52556631)

[24b Dynamic Modelling of Use-Cases 58](#_Toc52556632)

[24c Proposed System Architecture 58](#_Toc52556633)

[24d Initial Subsystem Decomposition 58](#_Toc52556634)

[25 Additional Design Considerations 58](#_Toc52556635)

[25a Hardware / Software Mapping 58](#_Toc52556636)

[25b Persistent Data Management 59](#_Toc52556637)

[25c Access Control and Security 59](#_Toc52556638)

[25d Global Software Control 59](#_Toc52556639)

[25e Boundary Conditions 59](#_Toc52556640)

[25f User Interface 59](#_Toc52556641)

[25g Application of Design Patterns 60](#_Toc52556642)

[26 Final System Design 60](#_Toc52556643)

[27 Object Design 60](#_Toc52556644)

[27a Packages 60](#_Toc52556645)

[27b Subsystem I 60](#_Toc52556646)

[27c Subsystem II 60](#_Toc52556647)

[27d etc. 60](#_Toc52556648)

[IV Project Issues 61](#_Toc52556649)

[28 Open Issues 61](#_Toc52556650)

[29 Off-the-Shelf Solutions 61](#_Toc52556651)

[29a Ready-Made Products 62](#_Toc52556652)

[29b Reusable Components 62](#_Toc52556653)

[29c Products That Can Be Copied 62](#_Toc52556654)

[30 New Problems 63](#_Toc52556655)

[30a Effects on the Current Environment 63](#_Toc52556656)

[30b Effects on the Installed Systems 64](#_Toc52556657)

[30c Potential User Problems 64](#_Toc52556658)

[30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 65](#_Toc52556659)

[30e Follow-Up Problems 65](#_Toc52556660)

[31 Migration to the New Product 66](#_Toc52556661)

[31a Requirements for Migration to the New Product 66](#_Toc52556662)

[31b Data That Has to Be Modified or Translated for the New System 67](#_Toc52556663)

[32 Risks 67](#_Toc52556664)

[33 Costs 68](#_Toc52556665)

[34 Waiting Room 69](#_Toc52556666)

[35 Ideas for Solutions 70](#_Toc52556667)

[36 Project Retrospective 71](#_Toc52556668)

[V Glossary 71](#_Toc52556669)

[VI References / Bibliography 72](#_Toc52556670)

[VII Index 72](#_Toc52556671)

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

*( The title above is formatted as Heading 3, so that it appears in the table of contents, but was then modified to be centered and include a page break before the paragraph. Likewise for the List of Tables heading on the next page.. )* ***Note: Remove this instructional paragraph.***

*If a document contains a large number of figures, then it is appropriate to include a list of figures at the beginning of the document, following the table of contents. Each figure should include a title, and be numbered in a consistent logical fashion. The following list of figures was automatically generated from figure captions ( see* ***Error! Reference source not found.*** *on page* ***Error! Bookmark not defined.*** *), and can be automatically updated by right-clicking on the table below and selecting “Update Field”. This feature is located in the “Captions” section of the “References” tab in MS Word.* ***Note: Remove this instructional paragraph.***

[*Figure 1: Game Environments Venn Diagram 10*](#_Toc52556439)

[*Figure 2 - Sample Use Case Diagram from Bruegge & DuToit ( modified ) 18*](#_Toc52556440)

[*Figure 3 - Sample Use Case Diagram from Robertson and Robertson 19*](#_Toc52556441)

*On a related note, the references in the paragraph above, “( see* ***Error! Reference source not found.*** *on page* ***Error! Bookmark not defined.*** *)” include cross-references to the Figure and page number that will adjust automatically when other Figures or pages are added or removed. This is done with the “cross-reference” button in the “Captions” section of the “References” tab in MS Word.* ***Note: Remove this instructional paragraph.***

### ****List of Tables****

*If a document contains a large number of tables, then it is appropriate to include a list of tables at the beginning of the document, following the table of contents. Each table should include a title, and be numbered in a consistent logical fashion. The following list of tables was automatically generated from table captions ( see below), and can be automatically updated by right-clicking on the table below and selecting “Update Field”. This feature is located in the “Captions” section of the “References” tab in MS Word.* ***Note: Remove this instructional paragraph.***

[*Table 1: Development Schedule 11*](#_Toc52556480)

[*Table 2 - Requirements - Acceptance Tests Correspondence* 56](#_Toc52556481)

# Project Description

## Project Overview

Checker Wars is a new, virtual take on the game of Checkers. It is being developed by Bohn Jell Entertainment to entertain game players and provide a new game for online competitive play. The game is a multidimensional version of Checkers with extra pieces and challenging elements that provide a more challenging experience with greater replayability.

## The Purpose of the Project

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

The video game industry is enormous. It has soared past the television and film business to become the biggest and fastest growing category of the entertainment business.

Checker Wars hopes to use this continual growth to establish a community of game players who play for fun and compete against each other in Checker Wars.

### Goals of the Project

We wish to provide a new source of entertainment and competition to the growing field of online game playing and, specifically e-sports.

### Measurement

We will measure the success of Checker Wars by the popularity of the game, both in number of downloads, and hopefully, streamers playing the game against each other via services such as twitch.tv.

## The Scope of the Work

### The Current Situation

Checker Wars is the first game developed by Bohn Jell Entertainment. The game will be the first product and therefore only source of revenue for the company.

### The Context of the Work

*SV: Define the boundary between what is included in “the work” and what is not. It also defines what external entities “the work” must interact with and what those interactions entail.* ***The following example diagram should be replaced with one appropriate to this project.***

*Content*

*The work context diagram identifies the work that you need to investigate to be able to build the product. Note that it includes more than the intended product. Unless we understand the work that the product will support, we have little chance of building a product that will fit cleanly into its environment.*

*The adjacent systems on the context diagram (e.g., Weather Forecasting Service) indicate other subject matter domains (systems, people, and organizations) that need to be understood. The interfaces between the adjacent systems and the work context indicate why we are interested in the adjacent system. In the case of Weather Forecasting Service, we can say that we are interested in the details of when, how, where, who, what, and why it produces the District Weather Forecasts information.*

*Motivation*

*To clearly define the boundaries for the study of the work and requirements effort. Without this definition, we have little chance of building a product that will fit seamlessly into its environment.*

*Examples*

*Considerations*

*The names used on the context diagram should be consistent with the naming conventions and data dictionary definitions presented in section 5. Without these definitions, the context model lacks the required rigor, and it may be misunderstood. Relevant stakeholders must agree to the definitions of the interfaces shown on the context model.*

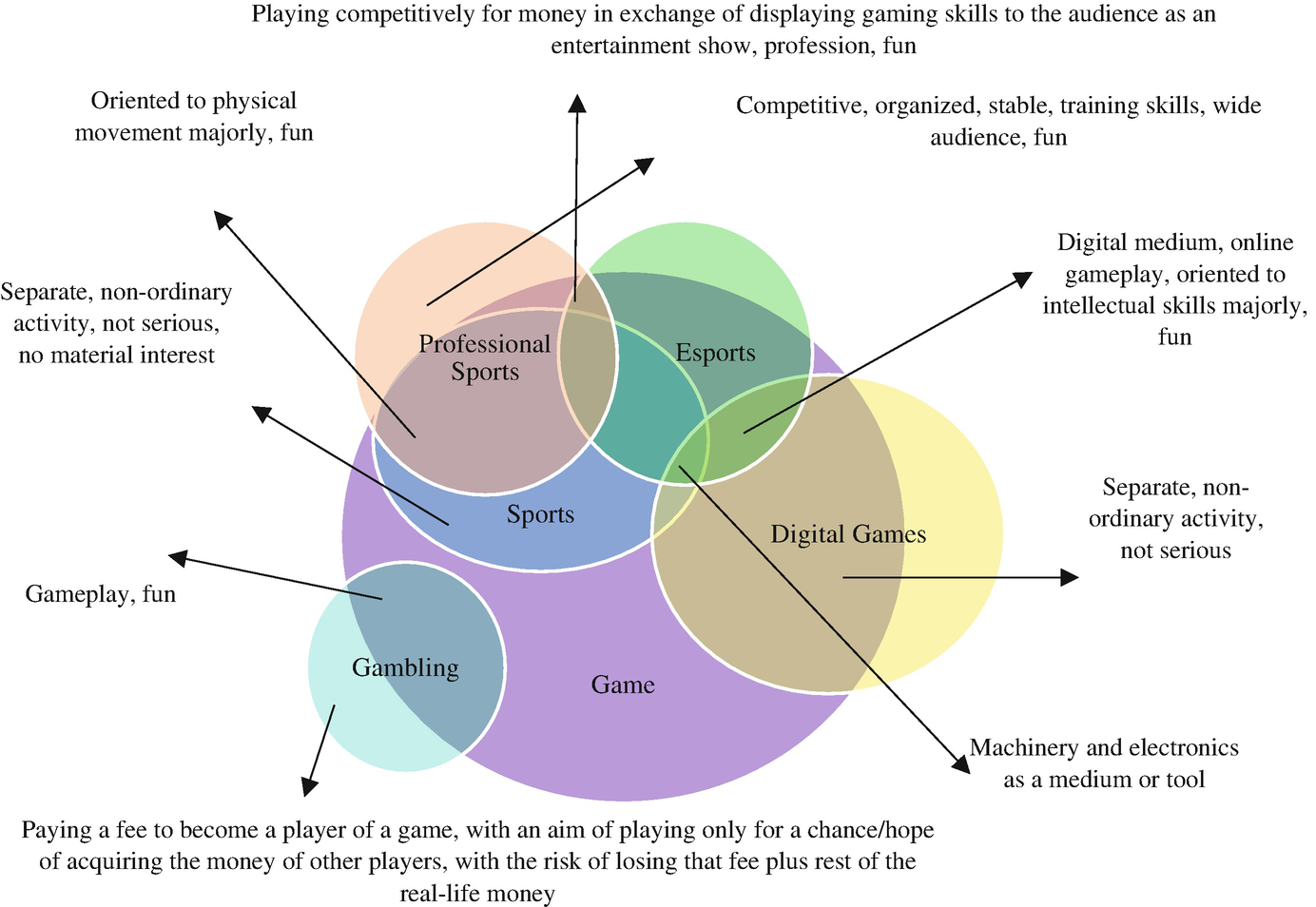


Figure 1: Game Environments Venn Diagram

The competitive gaming market is made up of many different disciplines. Originally physical, gameplay has moved into the virtual world, and e-sports are a large part of what encourages new players and keeps fueling the gaming industry. Without understanding where Checker Wars fits among these different sports and competitive markets, Checker Wars would be doomed to obscurity.

### Work Partitioning

*SV: “The work” is often large and complex, with many different activities and concerns. One good way to break this down and organize it for analysis is to identify the different events to which the business must respond. A “business event” is an external stimulus which causes the business to take a series of actions in response.*

*Content*

*A list showing all business events to which the work responds. Business events are happenings in the real world that affect the work. They also happen because it is time for the work to do something—for example, produce weekly reports, remind nonpaying customers, check the status of a device, and so on. The response to each event is called a business use case; it represents a discrete partition of work that contributes to the total functionality of the work.*

*The event list includes the following elements:*

*● Event name*

*● Input from adjacent systems (identical with name on context diagram)*

*● Output to adjacent systems (identical with name on context diagram)*

*● Brief summary of the business use case (This is optional, but we have found it is a very useful first step in defining the requirements for the business use case—you can think of it as a mini-scenario.)*

*Motivation*

*To identify logical chunks of the system that can be used as the basis for discovering detailed requirements. These business events also provide the subsystems that can be used as the basis for managing detailed analysis and design.*

*Example*

*Business Event List*

*Event Name Input and Output Summary*

|  |  |  |
| --- | --- | --- |
| *1. Weather Station transmits reading* | *Weather Station Readings (in)* | *Record the readings as belonging to the weather station.* |
| *2. Weather Service forecasts weather* | *District Weather Forecast (in)* | *Record the forecast.* |
| *3. Road engineers advise changed roads* | *Changed Road (in)* | *Record the new or changed road. Check that all appropriate weather stations are attached.* |
| *4. Road Engineering installs new Weather Station* | *New Weather Station (in)* | *Record the weather station and attach it to the appropriate roads.* |
| *5. Road Engineering changes Weather Station* | *Changed Weather Station (in)* | *Record the changes to the weather station.* |
| *6. Time to test Weather Stations* | *Failed Weather Station Alert (out)* | *Determine if any weather stations have not transmitted for two hours, and inform Road Engineering of any failures.* |
| *7. Truck Depot changes a truck* | *Truck Change (in)* | *Record the changes to the truck.* |
| *8. Time to detect icy roads* | *Road De-icing Schedule (out)* | *Predict the ice situation for the next two hours. Assign a truck to any roads that will freeze. Issue the schedule.* |
| *9. Truck treats a road* | *Treated Road (in)* | *Record the road as being in a safe condition for the next three hours.* |
| *10 Truck Depot reports problem with truck* | *Truck Breakdown (in)*  *Amended Gritting Schedule (out)* | *Reassign available trucks to the previously assigned roads.* |
| *11. Time to monitor road treatment* | *Untreated Road Reminder (out)* | *Check that all scheduled roads have been treated in the assigned time, and issue reminders for any untreated roads.* |
|  |  |  |

*Considerations*

*Attempting to list the business events is a way of testing the work context. This activity uncovers uncertainties and misunderstandings about the project and facilitates precise communications. When you do an event analysis, it will usually prompt you to make some changes to your work context diagram.*

*We suggest you gather requirements for discrete sections of the work. This requires you to partition the work, and we have found business events to be the most convenient, consistent, and natural way to break the work into manageable units.*

Table 1: Development Schedule

|  |  |  |
| --- | --- | --- |
| **Event Name** | **Input and Output** | **Summary** |
| 1. Game start | User inputs credentials (in) | User logs in with username and password |
| 2. Main Menu | Menu display (out) | Main menu is displayed where user can choose single-player or multiplayer gameplay |
| 3. Single-player game mode | Gameplay board is displayed (out) | User plays against AI opponent to practice in-game skills |
| 4. Multiplayer game mode | Gameplay board is displayed (out) | User connects to server and plays against human opponent |
| 5. Game end | Score displayed (out) | User see’s gameplay results and statistics. Score and rank increase or decrease depending on game result |

### Competing Products

*SV:* ***IF*** *there are other existing products that the client could use instead of the proposed product, then they should be discussed here, along with the reasons why the proposed product is still needed / beneficial.*

*Content*

*Other alternatives that already exist can be described here. Why should we go to all the trouble of creating a new product? What flaws or deficiencies do the existing products have that justify the creation of something new?*

*Motivation*

*Knowing what other choices the customer has to choose from can help us judge whether or not our project is even worth doing, and if so, what we need to do different to be better than the available alternatives.*

*Considerations*

*Note the subtle difference between this item and the “Off the Shelf” solutions documented in sections 0 or 29 below. The latter refers to software that we can buy and incorporate into our solution.*

Competitive single-player (with AI opponents) and multiplayer (both offline and online) games are all competitors of Checker Wars. Online Chess, Go, and similar strategy games are all competitors of Checker Wars. Some of these are standalone games, some of them are web apps, some are sold through services such as Steam or the Epic game store. These products’ successes and failures must be studied and emulated to ensure Checker Wars is a success.

## The Scope of the Product

*SV: This section describes the proposed product as a set of short stories ( scenarios ) providing examples of how the product would be used in practice. This effectively documents what is and what is not included in the product, and who/what would interact with it in what ways. The opening paragraph briefly states what subset of “the work” is to be handled by the proposed product.*

### Scenario Diagram(s)

*SV: The scenario diagram acts as an illustrated list of the scenarios to be presented, showing the boundary of the system and what external “actors” are involved in each of the scenarios.*

*Scenarios are somewhat informal stories describing how the end users would use the product once it is completed. They take the form of narratives and may involve specific individuals and examples. A scenario diagram is a graphical illustration of a number of scenarios, showing their relationships to each other and to external entities. It is similar in nature to use-case diagrams, except for scenarios. ( Note that while many scenarios will eventually evolve into full-blown use-cases, some will not, and additional use-cases will be created by the time the project develops to that stage, so there will not in general be a one-to-one correspondence between the scenario diagram(s) shown here and the use-case diagrams to be developed later. )*

*See Section 9a for a discussion of and examples of use-case diagrams.*

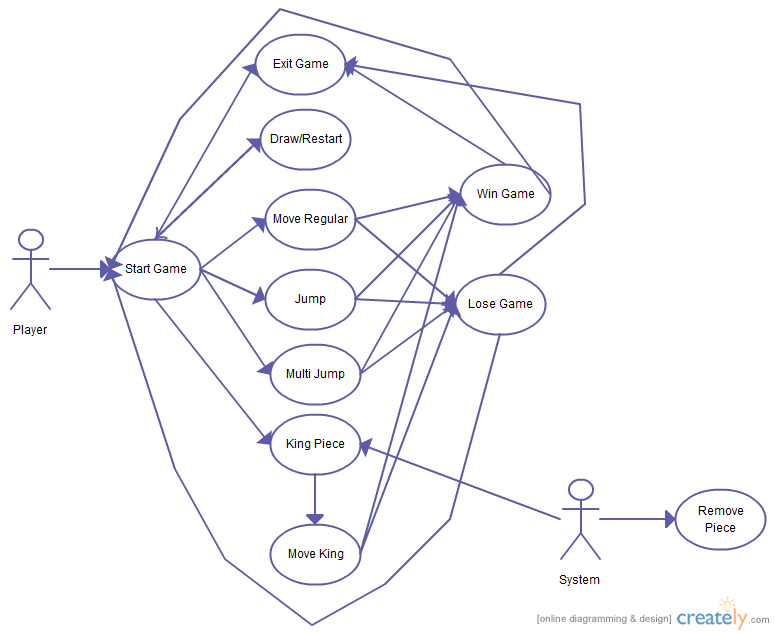


Figure 2: Checker Wars Scenario Diagram

### Product Scenario List

*SV: A table listing the scenarios by name, external actors involved, and possibly other information if relevant.*

*The product scenario list is quite simply a list of the product scenarios that will appear in the next section. It is a good idea to either number or name each scenario for later reference, and it can also be a good idea to organize the list so that related scenarios appear together. ( Depending on the naming / numbering scheme, they can be grouped into sections and subsections, etc. )*

Game Mode Select Scenario

Move Scenario

Move Pawn Scenario

Move Guardsman Scenario

Move Noble Scenario

Upgrade to King Scenario

Move King Scenario

Summon Space Scenario

Tower Space Scenario

Victory Scenario

### Individual Product Scenarios

*SV: This section contains the actual scenarios, the stories of the product being used.*

*Product scenarios are written in a natural narrative fashion, easily understood by clients and other non-technical stakeholders. Each one tells a story of how the end users are expected to eventually use the finished product. For example:*

***Monthly Reports:*** *At the end of every month Mary has to generate the monthly reports, and distribute copies to all the managers and sub-managers. The first thing she has to do is to make sure that all the end-of-the-month tests have been run, and that everyone else is logged off of the system. Then she selects the date range and the specific information she wants included in her reports, selects either the long or short format, and selects a printer. Depending on how busy the month has been, it may take as long as fifteen minutes, during which time no one else can use the system. She only prints one copy on the computer, and then makes all the rest of the copies she needs on the copy machine.*

**Game Mode Select Scenario**: On the menu select screen, the user may select to play single player mode against an AI opponent or multiplayer mode against a human opponent via the internet.

**Move Scenario**: Each player’s turn is determined by the roll of two die. The player with the highest roll goes first. This player gets to decide whether the order of turns goes clockwise or counterclockwise from them. Each player receives 5 turns per round. Each round ends when every player has exhausted their turns. Though, similar to checkers, each attack takes up one turn, and if an attack is chained, each chained attack will not take up another turn.

**Move Pawn Scenario**: The user opts to move a pawn piece on the game board. Pawns may only move to black spaces on the board.

**Move Guardsman Scenario**: The user opts to move a guardsman piece on the game board. Guardsman pieces may only move on white spaces on the board.

**Move Noble Scenario**: The user opts to move a noble piece on the game board. noble pieces may only move on white spaces on the board and cannot attack other pieces.

**Move King Scenario**: The user opts to move a King piece on the game board. The king pieces can move through either white or black spaces on the board. When a king transitions from one space color to another, that uses a turn and that king may not be able to move until the next round. When a king transitions from one color space, white or black, to another, it will do so on a space adjacent to it.

**King Space Scenario**: A pawn that lands on a King space is upgraded to a King piece for that player.

**Summon Space Scenario**: If a king and a noble lands on a summon space and is able to summon a new pawn piece on an adjacent black square. A summon space used by a noble or a king can only be used once per turn.

**Tower Space Scenario**: If a king or a noble lands on a tower space, they occupy that tower. If three towers are occupied by a single player, that player is victorious.

**Attack Scenario**: The user moves to attack an opponent’s piece. Guardsman can attack all other pieces on white spaces. Nobles cannot attack other pieces. King’s can attack any other piece. All pieces attack in a manner similar to the game of checkers, in that they “jump” over the piece to “take” them, and land on the other side of the opponents piece. A piece can only successfully attack another piece if it’s movement is unobstructed (can successfully land on the other side of the enemy piece.

**Victory Scenario**: A win is determined when either three towers are captured by one army or if the other players have no more pieces. When one player is completely defeated, they are out for the rest of the game and the game will continue with the other players. If for some reason, no moves exist for all players, the game ends in a draw.

## Stakeholders

*SV: Stakeholders include all persons or entities that have an interest in the proposed product or its development, either directly or indirectly.*

### The Client

*SV: The client pays up front for the product to be developed, and provides guidance or other input for its development. Some projects do not have an external client, in which case the developing organization acts as the client.*

*Content*

*This item gives the name of the client. It is permissible to have several names, but having more than three negates the point.*

*Motivation*

*The client has the final say on acceptance of the product, and thus must be satisfied with the product as delivered. You can think of the client as the person who makes the investment in the product. Where the product is being developed for in-house consumption, the roles of the client and the customer are often filled by the same person. If you cannot find a name for your client, then perhaps you should not be building the product.*

*Considerations*

*Sometimes, when building a package or a product for external users, the client is the marketing department. In this case, a person from the marketing department must be named as the client.*

This product is being designed, produced, and also commissioned by Bohn Jell Entertainment. Bohn Jell Entertainment has designed the specifications for the game, but is outsourcing development and graphics.

### The Customer

*SV: The customer is the person or entity who will buy the product after it has been completed. Some projects do not have an external customer, if they are to be used in-house or for the client’s use only.*

*Content*

*The person intended to buy the product. In the case of in-house development, the client and the customer are often the same person. In the case of development of a mass-market product, this section contains a description of the kind of person who is likely to buy the product.*

*Motivation*

*The customer is ultimately responsible for deciding whether to buy the product from the client. The correct requirements can be gathered only if you understand the customer and his aspirations when it comes to using your product.*

Checker Wars is aimed at teens and young adult players, however the game appeals to players of all ages due to its similarities with the game of checkers. The main focus will be on players who use streaming services such as twitch.tv.

### Hands-On Users of the Product

*SV: These are the people who will actually use the product in practice, and who may be separate from the customer or client. For example, educational software may be purchased by the school system ( customer ) and used by students ( hands-on users. )*

*Content*

*A list of a special type of stakeholder—the potential users of the product. For each category of user, provide the following information:*

*● User name/category: Most likely the name of a user group, such as schoolchildren, road engineers, or project managers.*

*● User role: Summarizes the users’ responsibilities.*

*● Subject matter experience: Summarizes the users’ knowledge of the business. Rate as novice, journeyman, or master.*

*● Technological experience: Describes the users’ experience with relevant technology. Rate as novice, journeyman, or master.*

*● Other user characteristics: Describe any characteristics of the users that have an effect on the requirements and eventual design of the product. For example:*

*Physical abilities/disabilities*

*Intellectual abilities/disabilities*

*Attitude toward job*

*Attitude toward technology*

*Education*

*Linguistic skills*

*Age group*

*Gender*

*Motivation*

*Users are human beings who interface with the product in some way. Use the characteristics of the users to define the usability requirements for the product. Users are also known as actors.*

*Examples*

*Users can come from wide variety of (sometimes unexpected) sources. Consider the possibility of your users being clerical staff, shop workers, managers, highly trained operators, the general public, casual users, passers-by, illiterate people, tradesmen, students, test engineers, foreigners, children, lawyers, remote users, people using the system over the telephone or an Internet connection, emergency workers, and so on.*

**User**: Game Players

**a.** **Role**: both serious, competitive gamers and casual gamers will enjoy playing Checker Wars. Games usually take 5 to 10 minutes to start and finish, allowing students or workers who want a quick mental break from their study or work to fit in a quick game.

**b.** **Subject Matter Experience**: All players will begin as novice, and if they continue to play and learn the ins and outs and meta of the game, over time they will progress to journeyman and then master.

**c.** **Technological Experience**: Users technological experience is only required at a novice level. The game is very easy to install, and the gameplay is very intuitive and easy to pick up. The game is easy to start playing, and hard to master.

**d.** **Other User Characteristics**:

**i.** **Age**: All ages

**ii. Gender**: All Genders

**iii.** **Motivation**: Entertainment or competition.

*SV: Describe users that will install, maintain, update, and otherwise service the product as needed. May not apply to all projects.*

*Content*

*Maintenance users are a special type of hands-on users who have requirements that are specific to maintaining and changing the product.*

*Motivation*

*Many of these requirements will be discovered by considering the various types of maintenance requirements detailed in section 14. However, if we define the characteristics of the people who maintain the product, it will help to trigger requirements that might otherwise be missed.*

Your text goes here . . .

### Other Stakeholders

*SV: This section is a catch-all for all other stakeholders not previously mentioned. Note that some stakeholders may be negatively impacted by the proposed project, for example if their work duties change or are eliminated.*

*Content*

*The roles and (if possible) names of other people and organizations who are affected by the product, or whose input is needed to build the product.*

*Examples of stakeholders:*

*● Sponsor*

*● Testers*

*● Business analysts*

*● Technology experts*

*● System designers*

*● Marketing experts*

*● Legal experts*

*● Domain experts*

*● Usability experts*

*● Representatives of external associations*

*For a complete checklist, download the stakeholder analysis template at* [*www.volere.co.uk*](http://www.volere.co.uk)*.*

*For each type of stakeholder, provide the following information:*

*● Stakeholder identification (some combination of role/job title, person name, and organization name)*

*● Knowledge needed by the project*

*● The degree of involvement necessary for that stakeholder/knowledge combination*

*● The degree of influence for that stakeholder/knowledge combination*

*● Agreement on how to address conflicts between stakeholders who have an interest in the same knowledge*

*Motivation*

*Failure to recognize stakeholders results in missing requirements.*

Because the game is focused around the competitive gaming and the e-sports scene, other potential stakeholders will be computer gear companies such as Razer or Logitech for sponsorship of the game or competitive players of the game.

### User Participation

*SV: To what extent can we expect users to participate during the development of the product?*

*Content*

*Where appropriate, attach to the category of user a statement of the participation that you think will be necessary for those users to provide the requirements. Describe the contribution that you expect these users to provide—for example, business knowledge, interface prototyping, or usability requirements. If possible, assess the minimum amount of time that these users must spend for you to be able to determine the complete requirements.*

*Motivation*

*Many projects fail through lack of user participation, sometimes because the required degree of participation was not made clear. When people have to make a choice between getting their everyday work done and working on a new project, the everyday work usually takes priority. This requirement makes it clear, from the outset, that specified user resources must be allocated to the project.*

There will be an open beta that players can opt in to join (for free) in order to help the developers test the gameplay mechanics.

## Mandated Constraints

*SV: Mandated constraints are requirements that are set in stone by the client before the project is really even started, and before the full set of requirements are determined. Note that not all of these sections will apply to every project, and that some constraints could be placed equally well in more than one section ( but should not be duplicated. ).*

*This section describes constraints on the eventual design of the product. They are the same as other requirements except that constraints are mandated, usually at the beginning of the project. Constraints have a description, rationale, and fit criterion, and generally are written in the same format as functional and nonfunctional requirements.*

### Solution Constraints

*SV: These are general constraints on the product to be developed or the manner in which it is to be developed that are not covered elsewhere.*

*Content*

*This specifies constraints on the way that the problem must be solved. Describe the mandated technology or solution. Include any appropriate version numbers. You should also explain the reason for using the technology.*

*Motivation*

*To identify constraints that guide the final product. Your client, customer, or user may have design preferences, or only certain solutions may be acceptable. If these constraints are not met, your solution is not acceptable.*

*Examples*

*Constraints are written using the same form as other atomic requirements (refer to the requirements shell for the attributes). It is important for each constraint to have a rationale and a fit criterion, as they help to expose false constraints (solutions masquerading as constraints). Also, you will usually find that a constraint affects the entire product rather than one or more product use cases.*

*Description: The product shall use the current two-way radio system to communicate with the drivers in their trucks.*

*Rationale: The client will not pay for a new radio system, nor are any other means of communication available to the drivers.*

*Fit criterion: All signals generated by the product shall be audible and understandable by all drivers via their two-way radio system.*

*Description: The product shall operate using Windows XP.*

*Rationale: The client uses XP and does not wish to change.*

*Fit criterion: The product shall be approved as XP compliant by the MS testing group.*

*Description: The product shall be a hand-held device.*

*Rationale: The product is to be marketed to hikers and mountain climbers.*

*Fit criterion: The product shall weigh no more than 300 grams, no dimension shall be more than 15 centimeters, and there shall be no external power source.*

*Considerations*

*We want to define the boundaries within which we can solve the problem. Be careful, because anyone who has experience with or exposure to a piece of technology tends to see requirements in terms of that technology. This tendency leads people to impose solution constraints for the wrong reason, making it very easy for false constraints to creep into a specification. The solution constraints should only be those that are absolutely non-negotiable. In other words, however you solve this problem, you must use this particular technology. Any other solution would be unacceptable.*

**Description**: The game will be on a computing device such as a PC, Mac, or iPad.

**Rationale**: The game will need Internet connection to play against opponents via the internet.

**Fit Criterion:** The product will be connected to Internet using the user’s connection to the Bohn Jell Entertainment’s servers.

**Description**: The game will most likely be developed in an object-oriented language.

**Rationale:** The game will be played on PC, Mac, and Android operating systems, and will have objects that need to be manipulated.

**Fit Criterion:** All of these platforms support development in object-oriented programming languages and coding style.

**Description**: The game will be played by players all around the world.

**Rationale:** The server of the game will be located in United States, but users from other countries will be able to use these game servers.

**Fit Criterion:** Any potential player with an internet connection will be able to download and play the game wherever they are located.

**Description:** The game will require the user to create an account.

**Rationale:** Users will need to create an account in order to track their stats and update their rank on both the local and global leaderboards.

**Fit Criterion:** The user will complete the registration form to sign up.

### Implementation Environment of the Current System

*SV: This section deals with the physical and technical environment in which the proposed product will operate, such as hardware, operating system, and communications issues.*

*Content*

*This describes the technological and physical environment in which the product is to be installed. It includes automated, mechanical, organizational, and other devices, along with the nonhuman adjacent systems.*

*Motivation*

*To describe the technological environment into which the product must fit. The environment places design constraints on the product. This part of the specification provides enough information about the environment for the designers to make the product successfully interact with its surrounding technology.*

*The operational requirements are derived from this description.*

*Examples*

*Examples can be shown as a diagram, with some kind of icon to represent each separate device or person (processor). Draw arrows to identify the interfaces between the processors, and annotate them with their form and content.*

*Considerations*

*All component parts of the current system, regardless of their type, should be included in the description of the implementation environment.*

*If the product is to affect, or be important to, the current organization, then include an organization chart.*

The game is to be played on Windows, Mac, iOS, and Android devices. The easiest way to accommodate this is to have the app developed as a web application, wherein a user is required to log in through a website to play the game. This will be the first step in development, and local clients for each platform will be developed once the web application is completed.

### Partner or Collaborative Applications

*SV: This section documents external applications with which this product must be compatible, such as the ability to read and write Microsoft Excel format data files.*

*Content*

*This describes applications that are not part of the product but with which the product will collaborate. They can be external applications, commercial packages, or preexisting in-house applications.*

*Motivation*

*To provide information about design constraints caused by using partner applications. By describing or modeling these partner applications, you discover and highlight potential problems of integration.*

*Examples*

*This section can be completed by including written descriptions, models, or references to other specifications. The descriptions must include a full specification of all interfaces that have an effect on the product.*

*Considerations*

*Examine the work context model to determine whether any of the adjacent systems should be treated as partner applications. It might also be necessary to examine some of the details of the work to discover relevant partner applications.*

Checker Wars currently has no partner or collaborative applications.

### Off-the-Shelf Software

*SV: This section describes commercial off-the-shelf ( COTS ) software that* ***MUST*** *be included in the final product.*

*Content*

*This describes commercial, open source, or any other off-the-shelf software (OTS) that must be used to implement some of the requirements for the product. It could also apply to nonsoftware OTS components such as hardware or any other commercial product that is intended as part of the solution.*

*Motivation*

*To identify and describe existing commercial, free, open source, or other products to be incorporated into the eventual product. The characteristics, behavior, and interfaces of the package are design constraints.*

*Examples*

*This section can be completed by including written descriptions, models, or references to supplier’s specifications.*

*Considerations*

*When gathering requirements, you may discover requirements that conflict with the behavior and characteristics of the OTS software. Keep in mind that the use of OTS software was mandated before the full extent of the requirements became known. In light of your discoveries, you must consider whether the OTS product is a viable choice. If the use of the OTS software is not negotiable, then the conflicting requirements must be discarded.*

*Note that your strategy for discovering requirements is affected by the decision to use OTS software. In this situation you investigate the work context in parallel with making comparisons with the capabilities of the OTS product. Depending on the comprehensibility of the OTS software, you might be able to discover the matches or mismatches without having to write each of the business requirements in atomic detail. The mismatches are the requirements that you will need to specify so that you can decide whether to satisfy them by either modifying the OTS software or modifying the business requirements.*

*Given the spate of lawsuits in the software arena, you should consider whether any legal implications might arise from your use of OTS. You can cover this in the section on Legal Requirements.*

*Note the subtle difference between this section and section 29 below. This section documents OTS solutions that* ***must be*** *included in the final solution, and the latter offers suggestions for OTS that could be included.*

Checker Wars is being programmed from the ground up, and so does not contain any off-the-shelf software.

### Anticipated Workplace Environment

*SV: This section deals with human factors regarding the environment in which the product will be used, such as noisy environments or mobile applications.*

*Content*

*This describes the workplace in which the users are to work and use the product. It should describe any features of the workplace that could have an effect on the design of the product, and the social and culture of the workplace.*

*Motivation*

*To identify characteristics of the workplace so that the product is designed to compensate for any difficulties.*

*Examples*

*The printer is a considerable distance from the user’s desk. This constraint suggests that printed output should be deemphasized.*

*The workplace is noisy, so audible signals might not work.*

*The workplace is outside, so the product must be weather resistant, have displays that are visible in sunlight, and allow for the effect of wind on any paper output.*

*The product is to be used in a library; it must be extra quiet.*

*The product is a photocopier to be used by an environmentally conscious organization; it must work with recycled paper.*

*The user will be standing up or working in positions where he must hold the product. This suggests a hand-held product, but only a careful study of the users’ work and workplace will provide the necessary input to identifying the operational requirements.*

*Considerations*

*The physical work environment constrains the way that work is done. The product should overcome whatever difficulties exist; however, you might consider a redesign of the workplace as an alternative to having the product compensate for it.*

Both single player and multiplayer gameplay will require an internet connection initially while the web application is the only source of gameplay. Further down the line when the standalone applications for the game are developed for each system, users will be able to play single player games against an AI opponent offline.

### Schedule Constraints

*SV:* ***When*** *things must be done, or when they may be most/least beneficial.*

*Content*

*Any known deadlines, or windows of opportunity, should be stated here.*

*Motivation*

*To identify critical times and dates that have an effect on product requirements. If the deadline is short, then the requirements must be kept to whatever can be built within the time allowed.*

*Examples*

*To meet scheduled software releases.*

*There may be other parts of the business or other software products that are dependent on this product.*

*Windows of marketing opportunity.*

*Scheduled changes to the business that will use your product. For example, the organization may be starting up a new factory and your product is needed before production can commence.*

*Considerations*

*State deadline limitations by giving the date and describing why it is critical. Also, identify prior dates where parts of your product need to be available for testing.*

*You should also ask questions about the impact of not meeting the deadline:*

*● 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?*

As previously discussed, the web application of the game will be developed first. Depending on reception and popularity of the game, development further into standalone Mac, Windows, iOS, and Android clients will continue, as will general development to the underlying codebase of the game. We aim to have the game developed and ready for launch by the Christmas holiday season.

### Budget Constraints

*SV: Limitations on the funds and other resources available for this project.*

*Content*

*The budget for the project, expressed in money or available resources.*

*Motivation*

*The requirements must not exceed the budget. This limitation may constrain the number of requirements that can be included in the product.*

*The intention of this question is to determine whether the product is really wanted.*

*Considerations*

*Is it realistic to build a product within this budget? If the answer to this question is no, then either the client is not really committed to building the product or the client does not place enough value on the product. In either case you should consider whether it is worthwhile continuing.*

Our first goal is to find an investor or sponsor to join us in funding the project for development. Sponsors can include computer and peripheral companies such as Razer, who can market their products beside the game, and when the game is complete, sponsorship of professional players of the game.

In-app purchases will take the form of cosmetic additions to the gameplay, as we want to create a fair competitive environment for al players of the game regardless of funds available to player. Thus, it is important that we find investors and sponsors who share this vision with us. The scope of the game is not massive, so we hope that we will not need copious funds to complete development of the game.

## Naming Conventions and Definitions

### Definitions of Key Terms

*SV: Define* ***words*** *that may have special or multiple meanings.*

*All Terms, Including Acronyms and Abbreviations, Used in the Project must be defined at some point. List the most important ones here, and refer the reader to the glossary on page 90 for a complete list. ( Note: that page number is a cross-reference, and will automatically be updated whenever the glossary moves. )*

*Content*

*A glossary containing the meanings of all names, acronyms, and abbreviations used within the requirements specification. Select names carefully to avoid giving a different, unintended meaning.*

*This glossary reflects the terminology in current use within the work area. You might also build on the standard names used within your industry.*

*For each term, write a succinct definition. The appropriate stakeholders must agree on this definition.*

*Avoid abbreviations, as they introduce ambiguity, require additional translations, and could potentially lead to misinterpretation in the mind of anyone who is trying to understand your requirements. Ask your requirements analysts to replace all abbreviations with the correct term. This is easily done with word processors.*

*Acronyms are acceptable if they are completely explained by a definition.*

*Motivation*

*Names are very important. They invoke meanings that, if carefully defined, can save hours of explanations. Attention to names at this stage of the project helps to highlight misunderstandings.*

*The glossary produced during requirements is used and extended throughout the project.*

*Examples*

*Truck: A vehicle used for spreading de-icing material on roads. “Truck” is not used to refer to goods-carrying vehicles.*

*BIS: Business Intelligence Service. The department run by Steven Peters to supply business intelligence for the rest of the organization.*

*Considerations*

*Make use of existing references and data dictionaries. Obviously, it is best to avoid renaming existing items unless they are so ambiguous that they cause confusion.*

*From the beginning of the project, emphasize the need to avoid homonyms and synonyms. Explain how they increase the cost of the project.*

All terms planned for the game are of general use and knowledge or are common to all other games played. These include, for example:

**RANK**: A player’s rank among other players of the game, both locally and globally.

**SCORE**: A player’s individual score at the end of every game

### UML and Other Notation Used in This Document

*SV: Define* ***symbols****,* ***diagrams****, and other* ***notations*** *used. May refer to a standard reference, such as “UML Distilled” by Fowler. ( Include in bibliography. )*

*Content*

*This section should describe the specific meaning of any symbols, punctuation, subscripts, superscripts, etc. used commonly throughout the document. If following published or common standards, then it is acceptable to reference those standards, and list any exceptions.*

*Motivation*

*If the distinction between a hollow arrow and a solid arrow is significant, for example, then everyone must know exactly what the distinctions and meanings are.*

*Considerations*

*If a particular notation is only used in one place, say on a single diagram or in a single section, then it may be more appropriate to document it in that specific location.*

*Example*

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

This document generally follows the Version 3.0 OMG UML standard. Any exceptions are noted.

### Data Dictionary for Any Included Models

*SV: Define* ***data structures*** *and* ***data properties*** *relative to this project, such as the contents of an employee record or the fact that student GPA ranges from 0.0 to 4.0 corresponding to letter grades of F to A. Data file formats may be referenced to documented standards, such as jpg or pdf.*

*Content*

*Dictionary definitions of all information flows and stores used in models. Particular consideration should be given to defining the data attributes of all flows shown the context models (see sections 7 and 8).*

*This section should also contain any technical specifications for interfaces shown on the context models.*

*Motivation*

*The context diagram provides an accurate definition of the scope of the work being studied or the scope of the product to be built. This definition can be completely accurate only if the information flows bordering the scope have their attributes defined.*

*Examples*

*Road de-icing schedule = issue number + {road section identifier + treatment start time + critical start time + truck identifier} + depot identifier*

*As you progress through the requirements specification, define each of the elementary terms in detail.*

*Considerations*

*The dictionary provides a link between the requirements analysts and the implementers. The implementers add implementation details to the terms in the dictionary, defining how the data will be implemented. Also, implementers add terms that are present because of the chosen technology and that are independent of the business requirements.*

Players may export and share their configuration (.conf) files for other players to load and experience what other players’ personal gameplay UI looks like.

## Relevant Facts and Assumptions

### Facts

*SV: Factual information relevant to the project, such as census data.*

*Content*

*Factors that have an effect on the product, but are not mandated requirements constraints. They could be business rules, organizational systems, or any other activities that have an effect on this product. Facts are things you want the reader of the specification to know.*

*Motivation*

*Relevant facts provide background information to the specification readers, and might contribute to requirements. They will have an effect on the eventual design of the product.*

*Examples*

*One ton of de-icing material will treat three miles of single-lane roadway.*

*The existing application is 10,000 lines of C code.*

The game will take up a small amount of space when developed for individual platforms, but will initially all be hosted on Bohn Jell Entertainment’s servers, and loaded (from Bohn Jell Entertainment’s servers or local user cache) whenever a user logs on through the website. The game will be under 20MB in size.

### Assumptions

It is assumed that the end user is using a sufficiently powerful computing device to play the game (2GB+ of RAM, modern ARM, Intel, or AMD processor).

# Requirements

*SV: Sections 9 and 10 deal with functional requirements. Sections 11 to 20 are a very thorough list of possible non-functional requirements, not all of which apply to every project. You should think carefully about each of these, form requirements if applicable, or write “Not Applicable” otherwise. See section 10 for the format of individual requirements. Section 21 documents the acceptance tests planned to verify the requirements – See that section for further details, and be aware that every requirement needs at least one verifying acceptance test ( though some tests may verify more than one requirement. )*

## Product Use Cases

*SV: Product Use Cases are very similar to Product Scenarios, but in more formal detail. They serve as a first step towards developing functional requirements, and can aid in organizing requirements according to the use case(s) from which they were developed. See the CS 440 web site for a sample use-case form, with instructions.*

*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

*SV: Use case diagrams list the use cases developed for a system, mark the boundary of what is internal or external to the system to be developed, and indicate which external entities ( actors ) are associated with each use case.*

*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.*

*Examples*

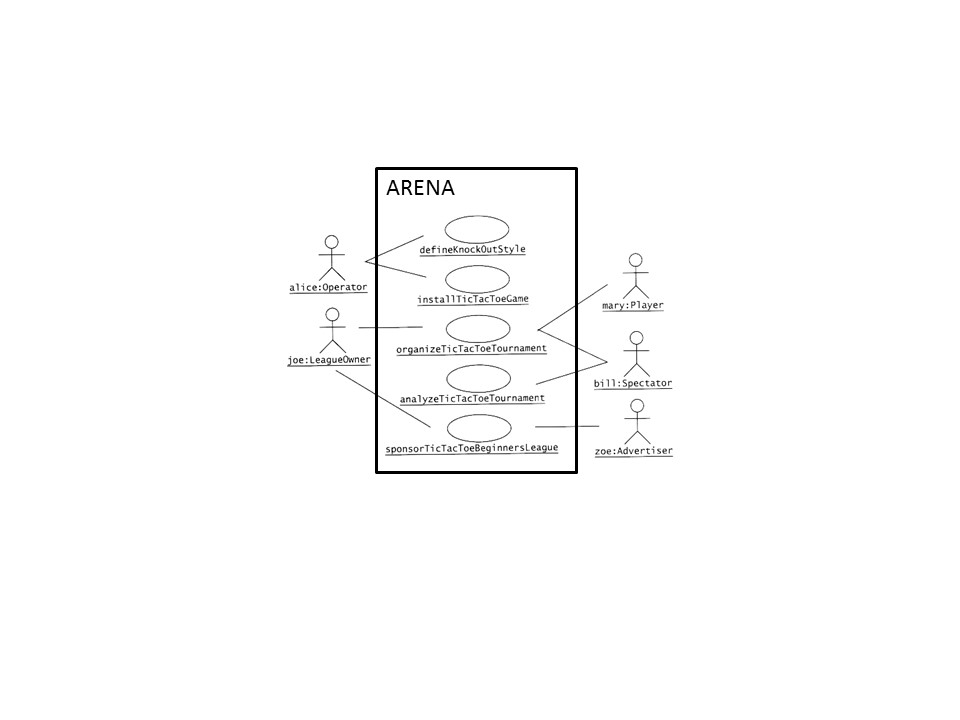


Figure - Sample Use Case Diagram from Bruegge & DuToit ( modified )

**

Figure - Sample Use Case Diagram from Robertson and Robertson

*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

*SV: A list ( table ) of use cases is an alternative to the use case diagram, particularly when there are many use cases. There may be additional information in the table not found in the diagram, such as cross referencing to other sections or materials.*

*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.*

*SV: The following example was copied from “useCaseFormWithInstructions.docx”, available on the CS 440 web site. ( There is also a blank version available. )*

|  |
| --- |
| Use case ID: Name:  pre-conditions:  post-conditions:  Initiated by:  Triggering Event:  Additional Actors: |
| Sequence of Events:  Initiating event or action should be step 1, taken by initiating actor.   1. System response follows, indented right.   All external action steps are aligned with step 1. ( "stimulus" style )   1. All system responses are indented right, aligned with step 2. ( "response" style )   All steps should be expressed in the active voice, clearly indicating **who** performs each action   1. The sequence of events should show a back-and-forth stimulus-response relationship. |
| Alternatives: These would be normal and expected variations from the base case.  Exceptions: These would be unusual variations from the base case, often caused by problems. |

* *For all of the above, list as NA if not applicable.*
* *The following may be added if relevant, or omitted otherwise:*
  + *related use cases or scenarios*
  + *associated tests, systems, classes, etc.*
  + *revision history*
  + *references to other documents*
  + *author(s) / originator( s )*
  + *notes*
* *Alternatives and Exceptions may be listed either as separate use cases or as notes to a base case, depending on their significance and similarity.*
* *For regularly occurring periodic events, "time" can be listed as the initiating actor.*

*You may also want to view Figure 4.7 from "Object Oriented Software Engineering" by Bruegge and DuToit*

## Functional Requirements

*SV: Each requirement listed needs to have a unique identifier, a short name, a one- or two-sentence description, a rationale, a fit criteria, and reference to one or more acceptance tests to be used to confirm the completion of this particular requirement. The acceptance tests themselves are documented in section 0- See that section for further details. It is recommended to number the requirements according to their type, such as F-4 for the fourth functional requirement or U-2 for the second usability requirement. Functional requirements specifically deal with the functionality the system must have, and are generally derived directly from the steps the system takes during use cases.*

*Content*

*A specification for each functional requirement. A full explanation is included in this template’s introductory material.*

*Motivation*

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

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Data Requirements

*SV: Data requirements deal with requirements that are somehow related to data, such as the definition of what is included in a “student record” or the acceptable form of an e-mail address or allowable range of certain data items.*

*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?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Performance Requirements

### Speed and Latency Requirements

*SV: Requirements specifying how fast ( or slow ) the product must operate or how much lag is allowable between stimulus and either initial response or task completion. Other timing-related requirements could go in this section.*

*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.*

*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).*

*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.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Precision or Accuracy Requirements

*SV: Self-explanatory. How accurate or precise must the system be.*

*Content*

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

*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.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Capacity Requirements

*SV: Requirements regarding the largest “thing” the system must be able to handle, or perhaps how many things it can handle ( at once. ) Note: Requirements regarding how many things it can handle in a given time period would be a speed requirement, covered in section 12a above.*

*Content*

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

*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.*

*Fit Criterion*

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

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Dependability Requirements

### Reliability Requirements

*SV: Reliability relates to how frequently the system fails, ( either by shutting down or by delivering erroneous results ), and the consequences of those failures. These requirements may also address the conditions under which it is allowed to fail ( or not. ), See also availability and robustness in the following sections.*

*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.*

*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. )*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Availability Requirements

*SV: Availability addresses the amount of time the system is running and available for use. It is affected by how often the system goes down ( reliability ), but also by the time required to bring the system back up again, the availability lost due to regularly scheduled maintenance down times, and the ability of the system to offer at least partial functionality in the face of failures or resource shortages. See also reliability and robustness.*

*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 ) and 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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Robustness or Fault-Tolerance Requirements

*SV: This section deals with the system’s ability to provide at least partial functionality in the face of failures or resource shortages, such as operating in offline mode when network connectivity is unavailable. See also reliability and availability.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Safety-Critical Requirements

*SV: These requirements address potential harm to health, safety, or property, and may refer to relevant standards such as OSHA compliance.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Maintainability and Supportability Requirements

### Maintenance Requirements

*SV: This section deals with the ease with which the system can be maintained, and possibly who will perform system maintenance and under what conditions. The ease of evolving the system into future versions may also be addressed here, or in a separate section ( not included in this template ) if that is a major concern.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Supportability Requirements

*SV: What ongoing support is to be provided, e.g. through a help desk. See also training requirements in section 16g below.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Adaptability Requirements

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

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Scalability or Extensibility Requirements

*SV: The ease of expanding the system to a larger capacity as the business grows.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Longevity Requirements

*SV: This specifies the expected lifetime of the product.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Security Requirements

*SV: Security requirements address who is allowed what type of access to the system, and what areas require special protection or diligence. In practice security requirements must often be written by security experts, and may refer to standards.*

### Access Requirements

*SV: These requirements address who has access to what ( data or functionality ) and under what conditions or restrictions.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Integrity Requirements

*SV: These requirements address the protection of data(bases) from intentional or accidental corruption, loss, or theft.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Privacy Requirements

*SV: These requirements address data that must remain confidential, such as medical records or other personally identifiable data. Laws often apply. (See also section 20.)*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Audit Requirements

*SV: This section applies when a system must provide support for transaction auditing, such as some financial or medical systems.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Immunity Requirements

*SV: This section addresses the system’s ability to resist viruses, worms, Trojan Horses, etc.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Usability and Humanity Requirements

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

### Ease of Use Requirements

*SV: This section addresses the ease with which the intended audience can use the system properly, and conversely the difficulty with which they can use it improperly.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Personalization and Internationalization Requirements

*SV: This section addresses the ease with which the system can be configured for personal preferences, and for things such as language, currency, units, symbols, etc.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Learning Requirements

*SV: Requirements related to how easy it is for the intended audience to learn to use the product.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Understandability and Politeness Requirements

*SV: These requirements relate to how intuitively the intended audience understands what the program does, what its messages mean, and how to use it. Definitely related to ease of use, ( section 16a ), but more specifically addressing comprehension of the program output, instructions, and other messages.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Accessibility Requirements

*SV: Requirements related to use of the product by individuals with disabilities.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### User Documentation Requirements

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

*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?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Training Requirements

*SV: A description of the training needed by users of the product.*

*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?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Look and Feel Requirements

### Appearance Requirements

*SV: These requirements address things such as the colors, fonts, and logos used, often to reflect corporate branding or similarity to related products. See also style in the next section.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Style Requirements

*SV: Style requirements address the impression the product makes upon users, such as professionalism for a tax accounting package, friendliness for a children’s game, or how “cool” it is for a teenage audience. Product packaging may also be addressed here, and/or appearance in the previous section.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Operational and Environmental Requirements

### Expected Physical Environment

*SV: These requirements relate to the physical environment in which the product will operate.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Requirements for Interfacing with Adjacent Systems

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

*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*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Productization Requirements

*SV: Requirements related to the distribution and/or installation of the product.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Release Requirements

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

*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?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Cultural and Political Requirements

### Cultural Requirements

*SV: 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. Bear in mind that “cultural groups” may also apply to population subgroups such as teenagers, the elderly, or ironworkers.*

*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?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Political Requirements

*SV: Requirements included strictly to make “the boss” happy, either internally to the development company, or internally to the client company, or possibly an external third party.*

*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.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Legal Requirements

### Compliance Requirements

*SV: A statement specifying the legal requirements for this system, often referring to relevant laws and/or requiring approval by the legal department.*

*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?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Standards Requirements

*SV: These requirements specify documented standards to which the product must conform, as opposed to legal regulations.*

*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?*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Requirements Acceptance Tests

*SV: Every requirement must have one or more acceptance tests associated with it, to confirm that the requirement has been met. At this point these tests are not yet completely specified – A one- or two-sentence description of each test will suffice. Note that some tests may verify more than one requirement, and that some requirements may require multiple tests for their confirmation.*

### Requirements – Test Correspondence Summary

*SV: The following sample table is available from the CS 440 web site as “Sample Requirement Test Correspondence Table.xlsx” It is recommended that you work with the table in Excel, and then drag it into the document when it is completed. Depending on the number of requirements and/or tests included, it may be necessary to use multiple tables, and/or use landscape mode. Every row and every column of the table should include at least one X. Below the table list the ID #, name, and short description of each individual acceptance test.*

**

*Table 2 - Requirements - Acceptance Tests Correspondence*

### Acceptance Test Descriptions

*SV: Provide a brief description of each acceptance test. Detailed test specifications will appear in a separate document, which may be referenced here when available.*

**ID # - Name**

**Description:** Your description here . . .

# Design

## Design Goals

*SV: Identify the important design goals that are to be optimized in the proposed design.*

*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. )*

Your text goes here . . .

## Current System Design

*SV:* ***IF*** *the proposed new system is to replace an existing system, then the current system should be described here. Otherwise insert a brief statement that there is no pre-existing system.*

Your text goes here . . .

## Proposed System Design

*This section will make heavy use of class diagrams, and also sequence and deployment diagrams where noted. However don’t overlook finite state, activity, communication, or other diagram types as needed for effective communication.*

### Initial System Analysis and Class Identification

*SV: Perform grammatical and similar analyses to identify the most import and obviously needed classes, and to organize them into an initial class structure. An initial class diagram is appropriate, containing few if any internal details.*

Your text goes here . . .

### Dynamic Modelling of Use-Cases

*SV: Insert sequence diagrams of ( at least the most important ) use-cases, as a means of identifying other needed classes.*

*Content*

*Include sequence diagrams of each important 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. )*

Your text goes here . . .

### Proposed System Architecture

*SV: Identify the Software Architecture to be applied to this project, such as Client-Server, Repository, MVC, etc., along with justification for the choice.*

Your text goes here . . .

### Initial Subsystem Decomposition

*SV: A slightly more detailed class diagram, showing the classes identified in sections 24a, 24b, and 0 above, partitioned into subsystems. For each subsystem provide a brief description of the subsystem, including its key responsibilities. There should still be few if any internal details.*

Your text goes here . . .

## Additional Design Considerations

*SV: The sections listed here do not need to be presented in the order given, and may not all be relevant for any particular project. Those that are relevant can help identify additional classes that are needed as a result.*

### Hardware / Software Mapping

*SV: This is particularly important for distributed systems, such as those employing a client-server architecture. Use a deployment diagram to indicate which subsystems are mapped onto which piece(s) of hardware, and what communication subsystems need to be added to the system as a result.*

Your text goes here . . .

### Persistent Data Management

*SV: Document the classes and perhaps subsystems necessary to store persistent data when the system shuts down, and to restore that data when the system starts back up again.*

*Reiterate key data structures and information as necessary for the understanding of this design phase. Refer the reader back to the data dictionary in section I7c above to avoid undue repetition, while reviewing only the most relevant items here.*

Your text goes here . . .

### Access Control and Security

*SV: Identify the access control and security concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### Global Software Control

*SV: Identify the global software control concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### Boundary Conditions

*SV: Identify the boundary condition concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns. In particular consider startup, shutdown ( normal or abnormal ), and the creation and/or maintenance of any configuration files, databases, or similar supporting data files.*

Your text goes here . . .

### User Interface

*SV: Include a preliminary user interface design here, possibly as a rough sketch or other mockup, in order to identify additional classes needed to implement the interface.*

*The final user interface design will normally be developed by appropriate experts in that area. However it is appropriate to include an initial design here, including possibly a low- or high- fidelity sketch/mockup, in order to identify key classes necessary to implement the user interface, such as forms and dialog windows. It may also go towards addressing usability and/or look-and-feel requirements, and/or identifying other overlooked components.*

Your text goes here . . .

### Application of Design Patterns

*SV: Any design patterns applied as a result of previous sections should have been addressed there, and identified as such at the time. Use this section to document only the additional design patterns that were not previously covered elsewhere. ( If any. )*

Your text goes here . . .

## Final System Design

*SV: Include here the final version of the overall system design, incorporating all the subsystems and classes added as a result of additional design considerations. Multiple diagrams may be needed, possibly starting with an overall package diagram showing all the different subsystems and the ( important ) classes contained within each one. Still not a lot of internal details.*

Your text goes here . . .

## Object Design

*This section documents the internal details of each class, to the extent that they can be designed at this time. Included should be the class interfaces ( public method signatures and responsibilities ) and constraints. It is probably best to break this section up into subsections corresponding to subsystems as documented above, and/or by ( Java ) packages if those are designed. It may also be appropriate to address additional design pattern considerations here, but not to the point of being redundant of previous documentation.*

*Certain methods, such as simple getters, setters, and constructors are not always documented, unless there is something special about them such as in the Singleton or Factory Method design patterns.*

### Packages

*SV: If the design involves assigning classes to packages ( .e.g Java packages ), then the packages to be created should be documented here.*

Your text goes here . . .

### Subsystem I

Your text goes here . . .

### Subsystem II

Your text goes here . . .

### etc.

Your text goes here . . .

# Project Issues

## Open Issues

*SV: 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?*

Your text goes here . . .

## Off-the-Shelf Solutions

*SV: Discussion of products or components currently available that could either be incorporated into the new solution or simply used instead of developing ( parts of ) the new solution.  The distinction between sections 35 a, b, and c is subtle, and not very important.*

Your text goes here . . .

### Ready-Made Products

*SV: Products available for purchase that could be used either as part of a solution or instead of ( a part of ) a solution.*

*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.*

Your text goes here . . .

### Reusable Components

*SV: Similar to 35a, but for components such as libraries or toolkits instead of fully blown products.*

*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.*

Your text goes here . . .

### Products That Can Be Copied

*SV: Products that could legally be copied would typically be past projects developed by the same development group, provided there were no restrictions that would prevent their reuse.*

*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.*

Your text goes here . . .

## New Problems

*SV: The proposed new system certainly has its benefits, but it could also raise new problems.  It is a good idea to identify any such potential problems early on, rather than being surprised by them later.*

### Effects on the Current Environment

*SV: Could the new system have any adverse effects on the working environment, e.g. the way people do their jobs?*

*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.*

Your text goes here . . .

### Effects on the Installed Systems

*SV: Could the new system have any adverse effects on other hardware or software 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.*

Your text goes here . . .

### Potential User Problems

*SV: Could the new system have any adverse effects on the users of the software? Could users possibly have a negative response to the new system?*

*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.*

Your text goes here . . .

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

*SV: Are there any ( physical ) limitations in the expected environment that could inhibit the proposed product?  ( e.g. weather, electrical interference, radiation, lack of reliable power, etc. )*

*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.*

Your text goes here . . .

### Follow-Up Problems

*SV: Basically any other possible problems that could occur.*

*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.*

Your text goes here . . .

## Migration to the New Product

*SV: This section only applies when there is an existing system that is being replaced by a new system, particularly when data must be preserved and possibly translated / reformatted.  Otherwise just write "Not Applicable" under section 38 and remove sections 38a and 38b.*

### Requirements for Migration to the New Product

*SV: These are a list of requirements relevant to the migration procedures.  For example a requirement that the two systems be run in parallel for a time until the client is satisfied with the new system and the users know how to use it.*

*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.*

Your text goes here . . .

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

*SV: This section specifically addresses****data****that must be preserved and/or translated / reformatted during the migration process.*

*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?*

Your text goes here . . .

## Risks

*SV: Consideration of the potential risks that could cause the project to fail / underperform.*

*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.*

Your text goes here . . .

## Costs

*SV: An estimate of what it will cost to complete this project.  Think not only in terms of dollars, but also time, resources, lost opportunities, etc.*

*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.*

Your text goes here . . .

## Waiting Room

*SV: This is a place to record ideas or wishes that will not be included in the current release of the product, but which might be worth reconsidering at a later date.*

*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.*

Your text goes here . . .

## Ideas for Solutions

*SV: When developing requirements only, it is not the role of the business analyst to dictate the implementation of the solution.  However they can pass along any ideas they have here as suggestions to the developers.  For CS 440 this report includes system and object design, so this section would make suggestions for implementation and testing that would come after design, such as the use of a particular language, IDE, library, or other tools.*

*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.*

Your text goes here . . .

## Project Retrospective

*SV: At the conclusion of the ( CS 440 ) project, reflect back on what worked well and what didn't, and how the process could be improved in the future.*

*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.*

Your text goes here . . .

# Glossary

*SV: The glossary is a more complete and inclusive dictionary of defined terms than that found in section I.7.a, the latter of which only covered the most important key terms needed to understand the report.*

*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.*

Your text goes here . . .

# 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, 63

Requirements 35, 51, 58

Test 64, 65