Ocean Inc. Project Report

An Industrial Management Game
Set in the Ocean.

Diego Bravo, Nick Filipov, Jose Bolanos, Alejandro Bravo
Group1
CS 440
University of Illinois Chicago

October 2024

Table of Contents

REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT 2
How to Use This Document 2
List of Figures 9
List of Tables 10

I Project Description 11

1 Project Overview 11

2 The Purpose of the Project 11

2a The User Business or Background of the Project Effort 11
2b Goals of the Project 11
2c Measurement 12

3 The Scope of the Work 13

3a The Current Situation 13
3b The Context of the Work 13
3c Work Partitioning 16
3d Competing Products 18

4 The Scope of the Product 18

4a Scenario Diagram(s) 19
4b Product Scenario List 19
4c Individual Product Scenarios 19

5 Stakeholders 20

5a The Client 20
5b The Customer 20
5c Hands-On Users of the Product 21
5d Maintenance Users and Service Technicians 22
5e Other Stakeholders 22
5f User Participation 23
5g Priorities Assigned to Users 24

6 Mandated Constraints 25

6a Solution Constraints 25
6b Implementation Environment of the Current System 26
6c Partner or Collaborative Applications 27
6d Off-the-Shelf Software 28
6e Anticipated Workplace Environment 29
6f Schedule Constraints 30
6g Budget Constraints 31

7 Naming Conventions and Definitions 31

7a Definitions of Key Terms 31
7b UML and Other Notation Used in This Document 32
7c Data Dictionary for Any Included Models 33

8 Relevant Facts and Assumptions 34

8a Facts 34
8b Assumptions 34

II Requirements 36

9 Product Use Cases 36

9a Use Case Diagrams 36
9b Product Use Case List 38
9c Individual Product Use Cases 38

10 Functional Requirements 40

11 Data Requirements 41

12 Performance Requirements 42

12a Speed and Latency Requirements 42
12b Precision or Accuracy Requirements 43
12c Capacity Requirements 44

13 Dependability Requirements 45

13a Reliability Requirements 45
13b Availability Requirements 46
13c Robustness or Fault-Tolerance Requirements 47
13d Safety-Critical Requirements 48

14 Maintainability and Supportability Requirements 49

14a Maintenance Requirements 49
14b Supportability Requirements 50
14c Adaptability Requirements 51
14d Scalability or Extensibility Requirements 52
14e Longevity Requirements 52

15 Security Requirements 53

15a Access Requirements 53
15b Integrity Requirements 54
15c Privacy Requirements 55
15d Audit Requirements 56
15e Immunity Requirements 57

16 Usability and Humanity Requirements 57

16a Ease of Use Requirements 57
16b Personalization and Internationalization Requirements 59
16c Learning Requirements 60
16d Understandability and Politeness Requirements 61
16e Accessibility Requirements 62
16f User Documentation Requirements 63
16g Training Requirements 64

17 Look and Feel Requirements 65
17a Appearance Requirements 65
17b Style Requirements 66

18 Operational and Environmental Requirements 67
18a Expected Physical Environment 67
18b Requirements for Interfacing with Adjacent Systems 68
18c Productization Requirements 69
18d Release Requirements 70

19 Cultural and Political Requirements 71
19a Cultural Requirements 71
19b Political Requirements 72

20 Legal Requirements 73
20a Compliance Requirements 73
20b Standards Requirements 74

21 Requirements Acceptance Tests 75
21a Requirements – Test Correspondence Summary 75
21b Acceptance Test Descriptions 76

III Design 76

22 Design Goals 76

23 Current System Design 76

24 Proposed System Design 76
24a Initial System Analysis and Class Identification 77
24b Dynamic Modelling of Use-Cases 77
24c Proposed System Architecture 77
24d Initial Subsystem Decomposition 77

25 Additional Design Considerations 77
25a Hardware / Software Mapping 77
25b Persistent Data Management 78

25c Access Control and Security 78
25d Global Software Control 78
25e Boundary Conditions 78
25f User Interface 78
25g Application of Design Patterns 79

26 Final System Design 79

27 Object Design 79

27a Packages 79
27b Subsystem I 79
27c Subsystem II 79
27d etc. 79

IV Project Issues 80

28 Open Issues 80

29 Off-the-Shelf Solutions 80

29a Ready-Made Products 81
29b Reusable Components 81
29c Products That Can Be Copied 81

30 New Problems 82

30a Effects on the Current Environment 82
30b Effects on the Installed Systems 83
30c Potential User Problems 83
30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 84
30e Follow-Up Problems 84

31 Migration to the New Product 85

31a Requirements for Migration to the New Product 85
31b Data That Has to Be Modified or Translated for the New System 86

32 Risks 86

33 Costs 87

34 Waiting Room 88

35 Ideas for Solutions 89

36 Project Retrospective 90

V Glossary 90

VI References / Bibliography.....91

VII Index.....91

List of Figures

Figure 1 - Context of the Work Diagram 11

Figure 2 - Scenario Diagram 14

List of Tables

Table 2 - Requirements - Acceptance Tests Correspondence 75

I Project Description

1 Project Overview

Ocean Inc. is a 3D-isometric industrial building and management game that takes place on an ocean. The player must expand their factories, build oil rigs, dispose of waste, and generate wealth to grow an industrial empire. Meanwhile, the sea level rises due to accelerated global warming from carbon emissions. The Ocean will become heavily polluted as well which could affect resource deposits. At this point, the objective of the game changes once the player has caused enough damage to the Earth's oceans. The player must mitigate or remediate the damage, otherwise the environmental effects will destroy their factories and practically render the game unplayable. This would be considered the "losing" state of the game.

This game will be available exclusively for PC platforms, such as Steam or Epic Games Launcher.

2 The Purpose of the Project

By making an educational but fun game for everyone to use. The game will bring awareness to the sea level rising and rapidly generating industrial-sourced pollution in the ocean. This project is being carried out because we want to bring awareness of the rising sea levels and the rapidly melting glaciers that has been happening for decades. The objective is to bring people together and to try to help fund organizations that are taking on this objective. Nonetheless, it's also being created to be a fun educational game for adults and the youth. Some motivation would be to create a game and be able to educate people about how the industrial revolution has affected the climate. We will know when goals have been reached when users download the game and interact with it because it will show that users are interested in the rising sea levels and are learning about it. The goal from the business perspective is to make as much money as possible which will be achieved by the initial purchase of the digital copy and any DLC content we add. Furthermore, if the game is being played and users purchase items in-game then we will be able to measure that.

3 The Scope of the Work

The "work" addressed by the proposed product, *Ocean Ink*, involves the development of an interactive, data-driven simulation game that educates players on the environmental impacts of industrialization, specifically on ocean ecosystems. This product is focused on creating engaging, educational experiences that merge industrial management gameplay with real-world environmental data, aiming to raise awareness about pollution and climate change.

3a The Current Situation

In analyzing the existing landscape of industrial management and environmental simulation games like *SimCity*, *Cities: Skylines*, and *Anno 2070*, it is clear that while these games address pollution and urban growth, they do so in a broad, often simplified context. Manual processes in these games typically involve the player making decisions

about resource allocation, waste management, and expansion, with limited integration of real-world environmental data. Automated processes in these titles handle pollution mechanics and environmental consequences based on predefined rules and player choices, but without dynamic feedback loops that reflect accurate, real-time environmental data. *Ocean Ink* seeks to replace and enhance these processes by offering a more precise, data-driven simulation, incorporating real-world climate data and ecological models to provide an authentic experience. By transitioning from static pollution mechanics to a dynamic, real-world simulation, *Ocean Ink* will create a more immersive, educational experience, allowing players to see the tangible effects of industrial actions on ocean ecosystems. Business analysts might have already conducted this investigation as part of the business case, identifying gaps in the existing products that *Ocean Ink* aims to fill by offering a more nuanced, scientifically accurate gameplay experience.

The motivation behind *Ocean Ink* is to improve upon the simplified environmental systems in games like *SimCity*, *Cities: Skylines*, and *Anno 2070*. In these games, players manually manage industrial growth, while automated processes simulate pollution based on fixed rules. By studying these systems and how players interact with them, *Ocean Ink* aims to fill gaps, such as the lack of real-world data and dynamic environmental feedback. Integrating actual climate data and more responsive environmental mechanics will offer a deeper, more educational experience, showing the true impact of industrial actions on ocean ecosystems.

3b The Context of the Work

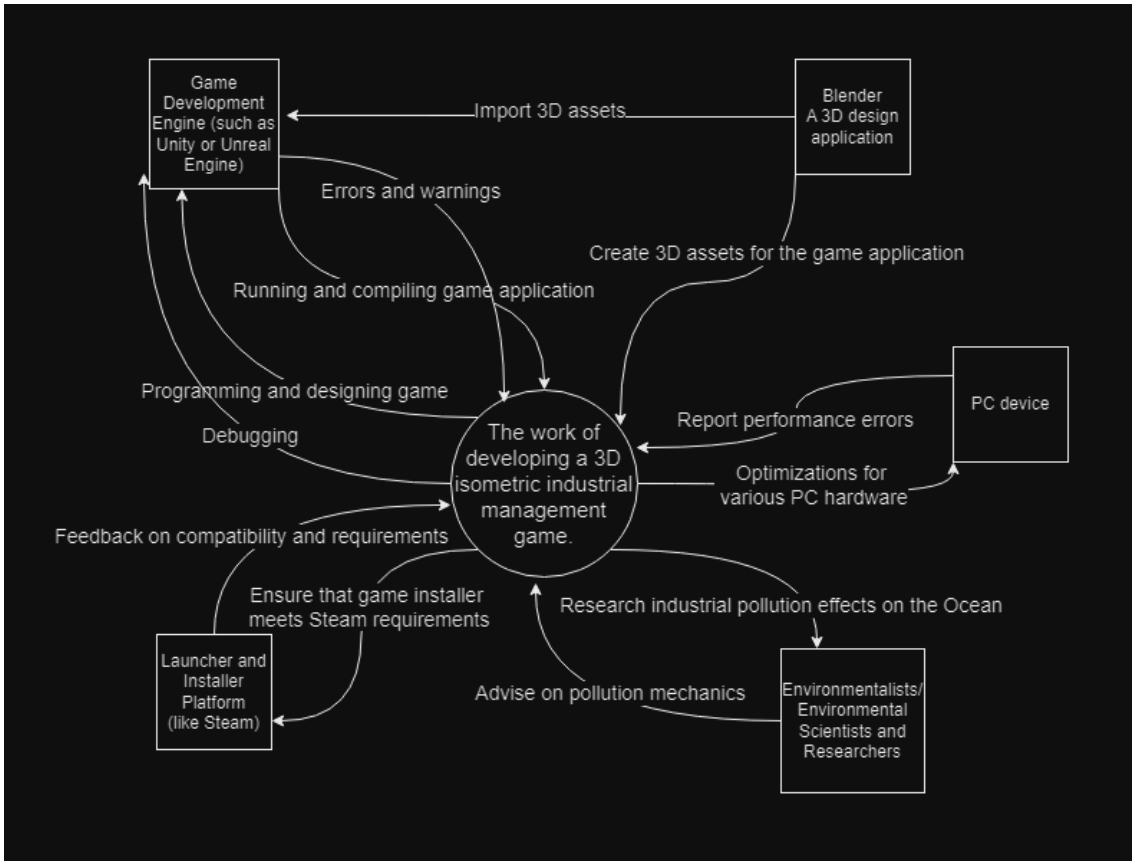


Figure 1

3c Work Partitioning

Business Event List

Event Name	*Input and Output*	*Summary*
1. Ocean Level Station transmits reading	*Ocean Level Sensor Data (in). Updated Ocean Level Reports (out)*	*Collects and analyzes real-time data on sea levels, providing critical information for assessing the impact of industrial activities and informing future business decisions.*
2. Ocean Pollution Reading Station	*Pollution Sensor Data (in). Pollution Level Reports (out).*	*Monitors various pollutants in the water, helping players understand the consequences of their industrial actions and prioritize remediation efforts.*

3. Building More Oil Rigs	*Resource Availability Data, Economic Forecasts (in). Oil Production Reports, Environmental Impact Assessments(out)*	*Players can expand their operations by building additional oil rigs. This decision will affect resource management and environmental health, necessitating careful analysis of potential impacts.*
4. Disposition of waste of business.	*Waste Generation Reports, Local Regulations (In). Waste Management Solutions, Compliance Reports (out).*	*Involves determining how to dispose of generated waste responsibly. Players must navigate regulations and potential fines while aiming to minimize environmental harm.*
5. Cleaning Up Ocean Waset	*Ocean Pollution Reports (in). Cleanup Success Reports, Environmental Health Improvements (out)*	*Initiate cleanup operations to reduce ocean waste, which not only mitigates pollution but also enhances their reputation and fosters marine biodiversity.*
6. Adding more operational buildings	*Infrastructure Capacity Data (in). Operational Efficiency Reports, Resource Allocation Updates (out).*	*Expanding operational buildings improves productivity. Players need to balance expansion with environmental considerations to avoid negative impacts on the ecosystem.*
7. Creating partnerships with other businesses.	*Market Analysis Reports, Business Collaboration Proposals (in). Partnership Agreements, Resource Sharing Plans(out)*	*Forming partnerships can enhance resource efficiency and reduce environmental impacts. Players must assess potential benefits and risks associated with each partnership.*
8. Hiring more personal	*Workforce Demand Analysis (in). Staff Allocation Reports,*	*As the business grows, hiring more personnel is crucial for maintaining efficiency. This decision*

	Operational Efficiency Metrics (out)	*requires evaluating the potential environmental impact of increased industrial activity.*
9. Upgrading current tools.	*Equipment Performance Data (in). Upgraded Tool Efficiency Reports, Environmental Impact Assessments (out).*	*Upgrading tools can enhance productivity and reduce environmental impact. Players must weigh the costs against the benefits of improved operations.*
10. Marine Wildlife Population Report	*Environmental Impact Assessments(in). Wildlife Health Reports, Conservation Recommendations (out)*	*This report provides players with insights into marine life populations, prompting them to consider conservation strategies to protect affected species amid industrial expansion.*
11. Implementing Green Technology Upgrades	*Technology Assessment Reports (in). Energy Efficiency Metrics, Reduced Pollution Levels (out).*	*Players can invest in green technology to improve sustainability. This event requires assessing current technologies and the potential long-term benefits for both the business and the environment.*

3d Competing Products

At the moment, there are no games on the market that are trying to achieve what we are doing. Games that are similar, though, are the ones I mentioned earlier in the section, such as *SimCity*, *Cities: Skylines*, and *Anno 2070*. Our game is different, though, because it specifically focuses on growing an industrial business and the direct negative effects it has on the ocean. While those games touch on pollution and urban development, *Ocean Ink* delves deeper into environmental consequences, using real-world climate data to simulate the impact of industrial actions on marine ecosystems and highlighting the critical need for environmental remediation.

4 The Scope of the Product

The main objective of this game is to balance industrial growth with environmental responsibility. Players will have the opportunity to monitor pollution levels while

making strategic decisions to grow their business. If pollution levels become too high, players can take action to mitigate environmental damage. Conversely, if their business needs expansion, players will have opportunities to invest in growth. The key challenge is to find the right balance between growing an industrial empire and minimizing harm to the planet.

4a Scenario Diagram(s)

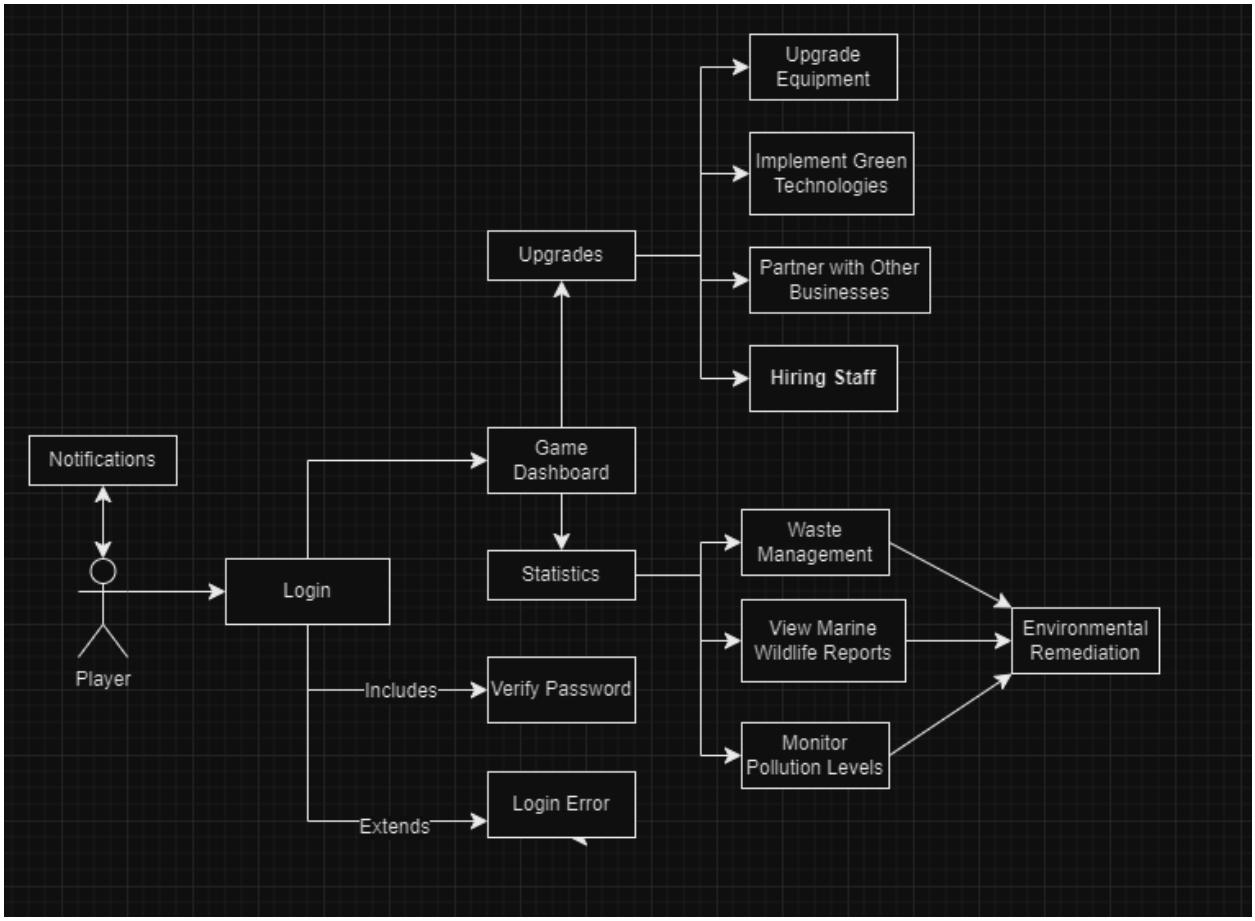


Figure 2

4b Product Scenario List

Scenario Name	Actors
1. Login	User
a. Verify password	Developer
b. Login error	User/Developer
2. Notifications	Developer
3. Upgrades	User
4. Upgrade Equipment	User
5. Implement Green Technologies	User
6. Partner with Other Businesses	User
7. Hiring Staff	User

8. Waste Management User
9. View Wildlife Reports User
10. Monitor Pollution Levels User
11. Environmental Remediation User

4c Individual Product Scenarios

- Login: This feature allows users to securely access their game accounts. Players will enter their credentials, including a username and password, to verify their identity. A successful login will grant access to their saved game progress, achievements, and customization options. If users encounter any issues, such as incorrect passwords or account lockouts, they will receive appropriate error messages and guidance for recovery.
- Notifications: This feature informs players about important updates, events, and alerts related to their industrial activities and environmental conditions. Notifications can include changes in pollution levels, wildlife population reports, successful upgrades, or urgent environmental issues requiring attention. Additionally, players receive reminders for tasks, such as waste management or partnership opportunities. By delivering timely information, this feature helps players make informed decisions and stay engaged with the game's dynamic challenges.
- Upgrades: This tab allows users to manage and improve various aspects of their industrial empire. When players open the Upgrades tab, they will see a list of potential upgrades available for their factories, equipment, and other resources. These upgrades might focus on enhancing productivity, reducing environmental impact, or implementing new technologies.
- Upgrade Equipment: This feature allows users to enhance their existing equipment, provided they have accumulated sufficient funds. Players can browse through available upgrades for their machinery and tools, which can improve operational efficiency, increase production output, or reduce environmental impact.
- Implement Green Technologies: This feature enables users to invest in sustainable technologies aimed at reducing their business's environmental footprint. Once players have the necessary resources, they can adopt eco-friendly solutions such as renewable energy sources, waste reduction systems, or pollution control measures.
- Partner With Other Businesses: This feature allows players to collaborate with other virtual businesses in the game. By forming partnerships, users can share resources, expand their market reach, and reduce operational costs. Collaborations may also open up opportunities for joint ventures in environmentally friendly technologies, waste management solutions, or other mutually beneficial projects.
- Hiring Staff: This option enables players to recruit additional personnel to improve the efficiency and productivity of their business. As the industrial empire grows, expanding the workforce becomes crucial to manage new operations, maintain equipment, and oversee environmental remediation efforts.
- Waste Management: This feature allows players to manage the disposal of waste generated by their industrial activities. Players must decide how to handle different types of waste—whether to invest in environmentally friendly disposal

methods, such as recycling or waste treatment, or opt for cheaper but more harmful options.

- View Wildlife Reports: This feature allows players to monitor the health and population levels of marine wildlife affected by their industrial activities. Players will receive detailed reports on various species, highlighting the impact of pollution, habitat destruction, and other environmental factors.
- Monitor Pollution Levels: This feature gives players real-time data on the pollution their industrial activities are generating in the ocean. Players can track various types of pollutants, such as chemical waste, oil spills, or plastic, through a pollution meter. The pollution levels will fluctuate based on their decisions related to production, waste management, and environmental policies.
- Environmental Remediation: This feature enables players to take action to repair the environmental damage caused by their industrial activities. Players can invest in various remediation strategies, such as cleaning up ocean pollution and restoring marine habitats. The effectiveness of these efforts will depend on the severity of the damage and the chosen method of remediation.

5 Stakeholders

5a The Client

The clients for the game would be game development companies or environmental organizations that are aiming and thinking that making an educational and engaging game like this would be a great way to educate people about the industrial impact on the oceans. The client needs to be interested in making a great product that raises awareness about climate change.

5b The Customer

The customers of the game would be multiple types of people. The first type of people would be schools that want to educate their students in a fun and interactive way about this specific topic, the second would be just ordinary people who like playing games and either want to play a fun management/simulation game for fun or educational purposes.

5c Hands-On Users of the Product

The Hands-On users of this application would be the students in the school who will be using the educational app, and other users would be ordinary people of any age who want to play the experience to learn or have fun.

The users role would be to build and grow their industries and while that is happening their factories will be producing pollution which they would then learn about and learn about how to deal with it to prevent the sea level from rising and hurting the wildlife and prevent too much pollution from occurring to be able to keep playing the game and learning.

It will be used by any age group of people, for people who are 18+ people who want to learn about pollution of the ocean from different industries, and for younger kids still in school who will use it as an educational tool.

5d Maintenance Users and Service Technicians

Different people who provide maintenance would include game developers, technical support teams, ocean pollution researchers, and possibly modders who want to update the game or extend its features.

These users will ensure that the game works, receive new content, and make sure that all the information of the game stays up to date and accurate to make sure that the different effects in the game match real life for each industry.

5e Other Stakeholders

Sponsor: Environmental organizations funding the project to raise awareness about ocean pollution.

Testers: Game testers who will ensure the product is both fun and educational, including different people who already play and enjoy educational games as well as students from different schools who will give suggestions on how to improve the game to make it more enjoyable to learn.

System designers: Responsible for creating an intuitive and well-designed user interface.

Marketing experts: Will ensure the game reaches the intended audience, such as people who enjoy simulation and education games, environmentally conscious people, and schools.

Domain experts: Climate change scientists will provide expertise on the game's environmental effects, messages, and gameplay.

These stakeholders will ensure the game is accurate, educational, and fun so that all people who play the game will enjoy their time while also learning.

5f User Participation

Before the game fully releases there will be people who play educational games, normal games, students from schools, and environmental experts who will play the game and give feedback on things that can be improved. This feedback will be used to make the game balanced between enjoyability and learning.

Beta testing would likely last for a few months until a high amount of feedback is received and the changes to be made.

5g Priorities Assigned to Users

Key users would be kids who are in school from middle to high school, as well as people who enjoy playing games and care about the environment.

Secondary users would be people who are not playing the game to learn but want to play a game that involves strategy and simulation. This will at the same time still be educational because the user would need to learn about the game and the pollution from industries to be able to progress through the game.

6 Mandated Constraints

6a Solution Constraints

Description: The final product will be a game developed to be played on a personal computer/laptop.

Rationale: Computers/Laptops and computer labs are located in mostly all schools and homes so it would be playable by people anywhere they are.

Fit Criteria: The game must not need a lot of power to run, meaning that it will be playable on almost all computers or laptops no matter how powerful it is.

Description: The game should be educational and entertaining.

Rationale: The game should be able to be a fun game for the users while also educating them on the importance of pollution in the ocean.

Fit Criteria: The user would need to play the game and pay attention to all of the important information that affects the gameplay and pollution.

6b Implementation Environment of the Current System

The product will be a game that will be accessible on any computer. It will be available for download on “Steam” which is a digital distribution platform and storefront for video games. It will be accessible for both Windows and Apple computers.

6c Partner or Collaborative Applications

The product will need collaboration with ocean pollution and industrial pollution experts to be able to collect accurate information to be used in the game as an educational tool. The product would also want to collaborate with many schools to be able to get the schools to use the game as an educational tool in schools for certain classes.

6d Off-the-Shelf Software

Content: To be able to create the game it would need Unity or Unreal Engine to be able to create the 3D graphics, physics simulation, and gameplay mechanics. A

database will be needed to manage the player's game state, player data, and progress. Other tools like Blender and Adobe will be used for 3D modeling for the game.

Motivation: Existed tools like these would allow well-designed and rapid prototyping of different mechanics of the game to test for accuracy and gameplay of all of the features.

Examples: Unity offers extensive documentation and a large asset store, making it easy to find resources. Databases such as MongoDB allow for flexibility and easy integration with the game's backend.

Considerations: Game design requirements and commercial off-the-shelf software may conflict in different ways such as the software not allowing a certain level of needed customization for the game. Identifying the needs of the game and software as soon as possible will help decide whether the game needs to have a specific feature or use a different solution or software.

6e Anticipated Workplace Environment

The game will be accessible using wifi through the Steam game launcher. The user would be located in a place that has wifi connectivity and has access to a computer or laptop to be able to access and play the game.

6f Schedule Constraints

There is no specific schedule but the sooner the game is ready for beta testing the better. As soon as the game is in a playable state it will be released to schools and people who want to beta test the game and give feedback on what can be improved both for educational and entertainment purposes. This will let the game be able to be fully created as soon as possible and released to the public in a working state.

6g Budget Constraints

Content: The project budget would be around $500 thousand dollars, this budget would include development, marketing, and research costs needed for the project. The development would include game design, programming, and asset design and creation. The research will include research on the different effects of pollution on oceans from industrial buildings.

Motivation: These are essential to developing the game to make it look good, fun, and interesting to play, as well as having real facts and data from real life to make it an educational experience.

The budget should be properly allocated to different parts of the project to make sure that the right amount of development, marketing, and research is done to complete the project.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Factories: Buildings that the player can build which will either make them money, gather resources, or create products.

Playable Area: The space where the player can build factories.

Build Menu: A menu that contains the different factories available to build.

Pollution Meter: A meter that displays how much pollution the factory is producing.

Upgrades: Factories can be upgraded in different ways to increase productivity or reduce pollution.

Pollution Rate: Rate of pollution produced by a factory. {Pollution unit per day}

7b UML and Other Notation Used in This Document

This document typically follows the Version 2.0 OMG UML standard, as described by Fowler in UML Distilled, Third Edition, Boston: Pearson Education, 2004. Exceptions may be noted in specific cases.

7c Data Dictionary for Any Included Models

pollution meter amount = {factory1 pollution rate + factory2 pollution rate + factoryN pollution rate…} x {days passed}

Determining Pollution Rate: Depends on the factory type and the amount of waste it produces.

8 Relevant Facts and Assumptions

8a Facts

- Census data from websites like NASA will be pulled in order to try to make the game as realistic as possible.
- Climate data will be collected in order to make industry factories show real effects in the game.
- Sea-level and glacier images will be used in order to show real-life effects in the game.
- Census data will be collected from Government organizations to see how the climate has been affected by industry factories.

8b Assumptions

- Players will be able to assume their industrial actions have consequences.

- Assuming that the development team has all the tools it needs and will be managing the game for bug fixes, feedback, and updates.
- Users will assume it will work on their PC.
- The development team will assume the user has no knowledge of climate change and factories' effects on climate change.
- It will be assumed that there will be a partnership with environmental organizations in order to try to make the game as realistic and educational as possible.
- It will be assumed that micro-transactions will be made in-game for cosmetics.
- There will be no changes due to environmental effects that should impact the development.
- Software tools that are expected to be available will be Unity or Unreal engine for the game development and somewhere to store the code like Github. Furthermore, in order to create tasks and manage activity JIRA.
- We are expecting to use the product in order to educate and make a profitable game to bring awareness to climate change's effect on Oceans.
- Any business changes we will be able to deal with are any copyright issues because this is very different from products out in the market.

II 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.)

1 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.

1a 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

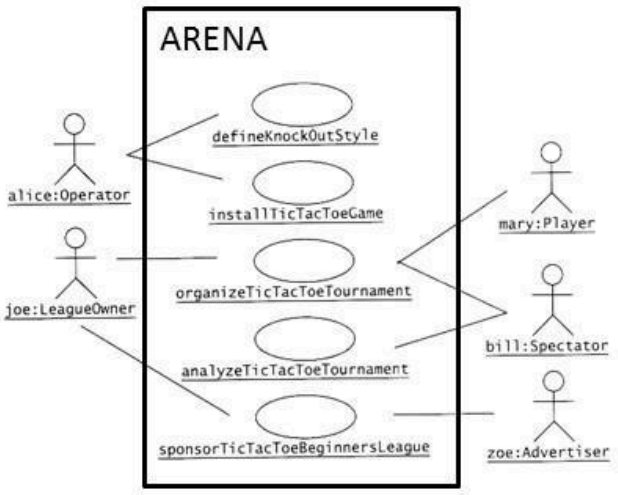


Figure 1 - Sample Use Case Diagram from Bruegge & DuToit (modified)

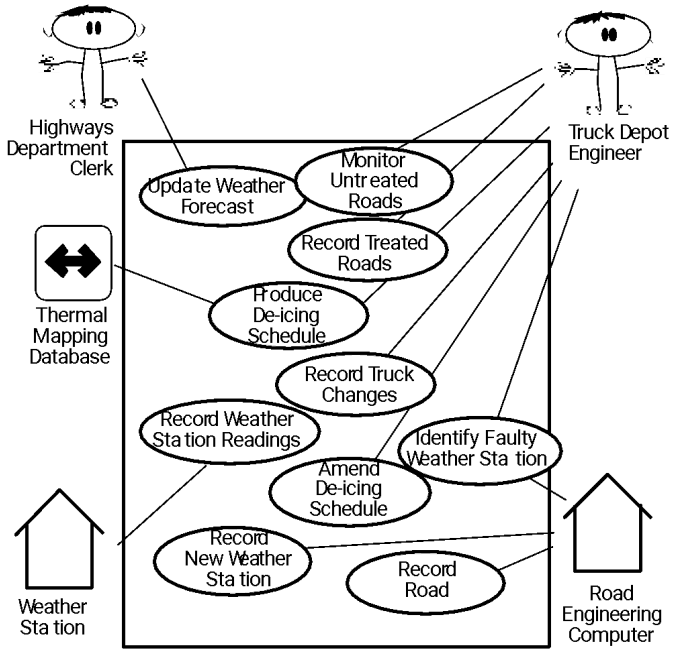


Figure 2 - Sample Use Case Diagram from Robertson and Robertson

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.

1b 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.

1c 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:

1. Initiating event or action should be step 1, taken by initiating actor.
2. System response follows, indented right.
3. All external action steps are aligned with step 1. ("stimulus" style)
4. All system responses are indented right, aligned with step 2. ("response" style)
5. All steps should be expressed in the active voice, clearly indicating **who** performs each action
6. 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

2 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 . . .

3 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

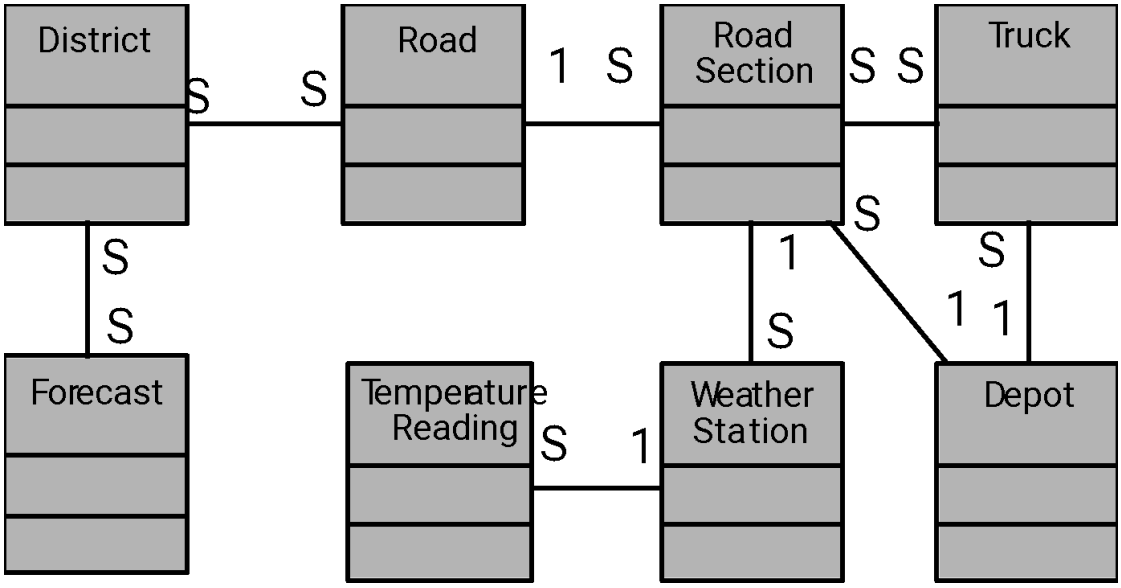
A specification of the essential subject matter, business objects, entities, and classes that are germane to the product. It might take the form of a first-cut class model, an object model, or a domain model. Alternatively, these requirements might be described by defining the terms in the dictionary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

4 Performance Requirements

4a 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 . . .

4b 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 . . .

4c 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 . . .

5 Dependability Requirements

5a 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 . . .

5b 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 . . .

5c 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 . . .

5d 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 . . .

6 Maintainability and Supportability Requirements

6a 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 . . .

6b 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 . . .

6c 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 . . .

6d 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 . . .

6e 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 . . .

7 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.

7a 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 . . .

7b 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 . . .

7c 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 . . .

7d 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 . . .

7e 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 . . .

8 Usability and Humanity Requirements

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

8a 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 . . .

8b 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 . . .

8c 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 . . .

8d 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 . . .

8e 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 . . .

8f 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 . . .

8g 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 . . .

9 Look and Feel Requirements

9a 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 . . .

9b 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 . . .

10 Operational and Environmental Requirements

10a 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 . . .

10b 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 . . .

10c 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 . . .

10d 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 . . .

11 Cultural and Political Requirements

11a 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 . . .

11b 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 . . .

12 Legal Requirements

12a 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 . . .

12b 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 . . .

13 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.

13a 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.

	Requirements																			
Test	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9	Req 10	Req 11	Req 12	Req 13	Req 14	Req 15	Req 16	Req 17	Req 18	Req 19	Req 20
Test 1	X																			
Test 2		X				X														
Test 3			X	X																
Test 4					X	X														
Test 5																				
Test 6																				
Test 7																				
Test 8																				
Test 9																				
Test 10																				
Test 11																				
Test 12																				
Test 13																				
Test 14																				
Test 15																				

Table 1 - Requirements - Acceptance Tests Correspondence

13b 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 . . .

III Design

1 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 . . .

2 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 . . .

3 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.

3a 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 . . .

3b 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 . . .

3c 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 . . .

3d 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 . . .

4 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.

4a 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 . . .

4b 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 . . .

4c 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 . . .

4d 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 . . .

4e 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 . . .

4f 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 . . .

4g 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 . . .

5 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 . . .

6 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.

6a 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 . . .

6b Subsystem I

Your text goes here . . .

6c Subsystem II

Your text goes here . . .

6d etc.

Your text goes here . . .

IV Project Issues

1 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 . . .

2 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 . . .

2a 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 . . .

2b 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 . . .

2c 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 . . .

3 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.

3a 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 . . .

3b 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 . . .

3c 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 . . .

3d 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 . . .

3e 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 . . .

4 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.

4a 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 . . .

4b 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 . . .

5 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 . . .

6 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 . . .

7 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 . . .

8 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 . . .

9 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 . . .

V 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 . . .

VI 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.

VII 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