Battlerines: The War of Submarines

Prepared by

VMVA Development Group

Viktor Kirillov
Mykyta Parovyi
Volodymyr Vakhniuk
Aleksandra Dmitrieva

for use in CS 440 at the
University of Illinois Chicago

September 2020

Table of Contents

I Project Description

1 Project Overview

Battlerines: The War of Submarines is an extension to a popular, classic game *Battleship*. The main goal of an extension is to create an environment where the game can be played by more than two people at the same time. This extension provides game rules and mechanics for two - four players, giving them new and unique board layout, ability to move their ships and a new goal: destroy opponents bases instead of ships in order to defeat another player.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Current implementation of the *Battleship* game provides a game mechanics suited for two players only. That makes it impossible to play in a group of three or four players without a significant amount of changes in both game logic and board layout.

2b Goals of the Project

We want to make it possible to play a classic *Battleship* game with more than two players at a time.

2c Measurement

The project is considered as completed when it provides an ability for two, three and four users to play a fair game, considering that the game lasts for at least 10 minutes.

3 The Scope of the Work

The work that is covered by the project encapsulates the activities associated with the user's entertainment while playing the *Battleship* game with more than two players.

3a The Current Situation

The game industry provides a lot of games that can be played with more than two players at a time, but if the user wants to play a classic or an old game (*Battleship* in our case) with more than one of the friends, they currently have no ability at all to do that.

3b The Context of the Work

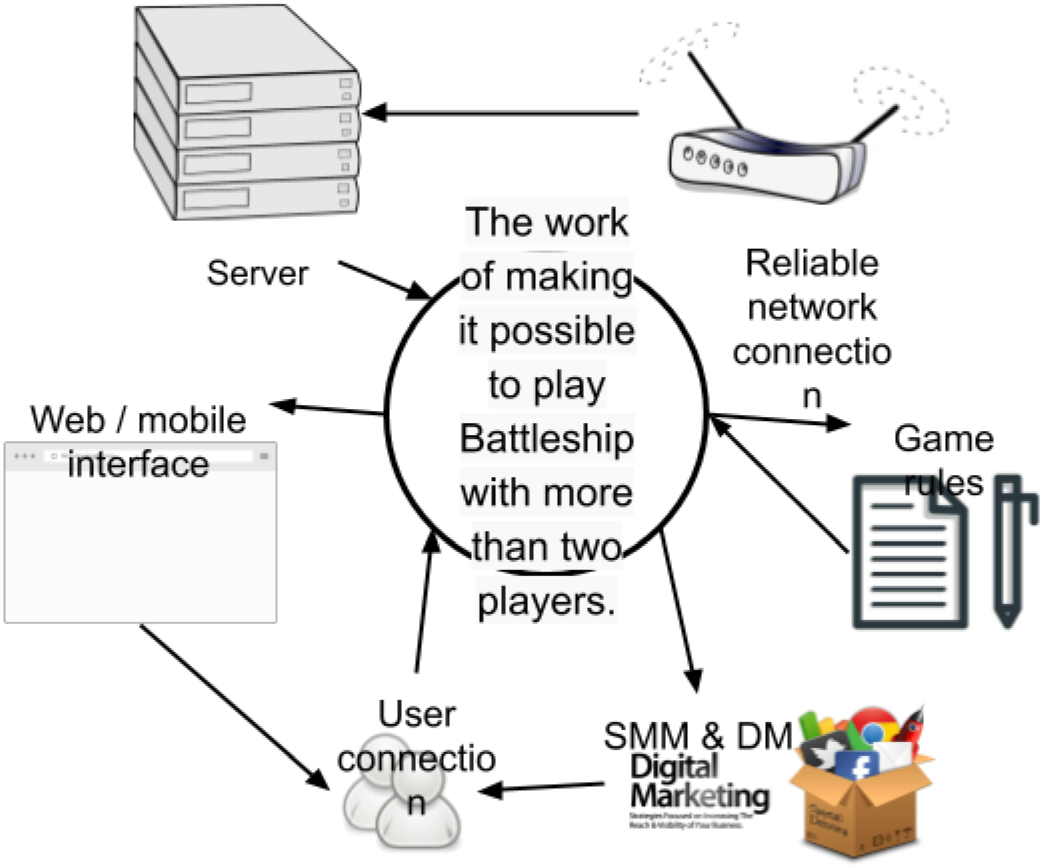

The work of making it possible to play Battleship with more than two players.
Server
Reliable network connection
Game rules
Web / mobile interface
User connection
SMM & DM
Digital Marketing

Figure 1 - The context of the work

Besides creating game rules, the scope of the work also includes providing the interface for the user to be able to connect and interact with. In order to make it possible for multiple players to interact with each other within the game, the reliable server should be created and configured to handle user connections and send the game gata between all players.

3c Work Partitioning

Business Event List:

Event Name	Input & Output	Summary
Server warning or error	Server (in) Web / mobile interface (out)	Warn users about the risk of losing the game progress and start the maintenance.
User connects to the game	User connection (in) Web / mobile interface (out)	Display the game interface to the user.

Lack of users	SMM & DM (out)	Target potential users with a promotion on social media or game-related web pages.
User reports a problem or suggests a feature	Game rules (out) Game rules (in)	Adjust the game rules and implement them in the game.

3d Competing Products

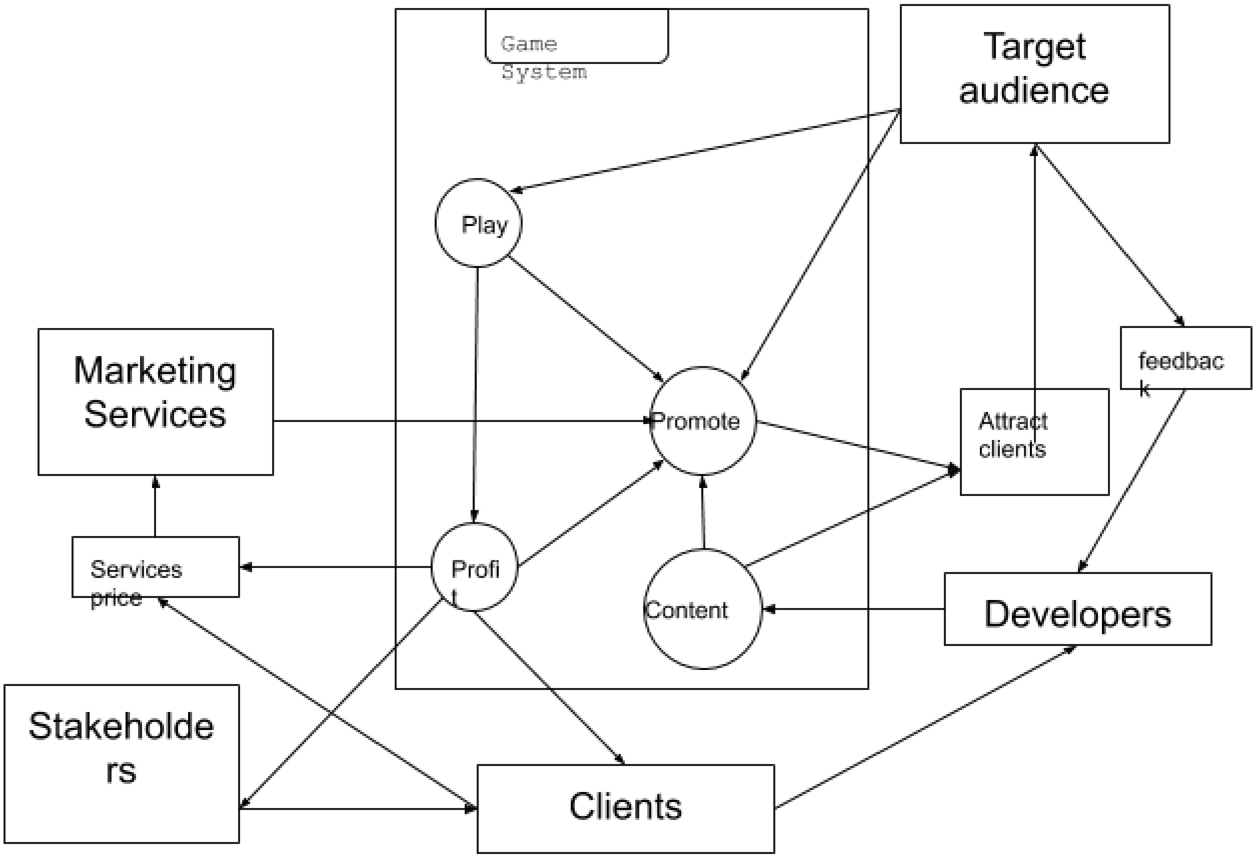
No products that give an ability to play classic or close to classic Battleship games with more than two players were found.

4 The Scope of the Product

Subset of work:

- Client Playing
- Promotion via Marketing Services
- Stakeholders Involvement
- Developers work

4a Scenario Diagram(s)



4b Product Scenario List

1. Game Playing - **Target Audience**
2. Promotion & Target Audience Attraction
   Marketing Services / Stakeholders / Developers / Target Audience / Client
3. Content Updating - **Developers / Target Audience / Stakeholders**
4. Profit Getting - **Stakeholders / Clients**
5. Feedback Creation - **Target Audience**

4c Individual Product Scenarios

- ***Game Playing:*** A Target Audience person who purchased the product is able to use it at any time by playing it. First thing they have to do is to log into their account and select to either play in a single or multiplayer mode. Secondly depending on what they chose, the game starts and they either begin their battle with a computer (if a single player mode was chosen) or with another enemy player (if a multiplayer mode was chosen). When the game ends, players can either play again or quit the game.

- ***Promotion & Target Audience Attraction:*** Target Audience, Stakeholders, Developers & Marketing services are all involved. Target Audience promote the

product by using it and occasionally telling other people about it.
Clients pay the Marketing services to do their job of promoting it. |
Marketing services promote the product by using social media ads, google ads, YouTube ads, etc. Developers promote the product by updating its content from Target Audience feedback of the product.Stakeholders promote the product via their website and social media platform announcements of funding.

- ***Content Updating***: Developers get the feedback from Target Audience and update content accordingly.
- ***Profit getting:*** Clients and Stakeholders will get their profit from Target Audience purchasing the product.
- ***Feedback Creation:*** Target Audience will generate feedback while using the product which will directly be addressed to developers to look at.

5 Stakeholders

There is an extremely large pool of groups that are interested in game development. Below listed all the groups that are considered the most important.

5a The Client

Battlerines: The War of Submarines is the original idea of VMVA Development Group Organization.

VMVA Development Group takes full interest in development and realizing the game. The game is intended for common use. Main purpose of the project is entertainment.

5b The Customer

The game is intended to be placed on online platforms such as Kongregate.com, MmoGames.com for free and demo version. The game also will be available for purchase. Customers purchasing the game are hand-on users.

5c Hands-On Users of the Product

Hands-On Users of the product can be divided into several groups:

1) Gamers:

Gamers are the people who play the game. The concept of Battlerine: war of submarines is a new concept in the war games category. Individual gamers are responsible for the game set up and get themself familiar with additional rules the game might have. Since the Battlerine: war of submarines is an expansion of classic battleship games it is assumed that many gamers will have some knowledge of the rules. For every gamer who is unsure about the rules we provide a description of the game rules. The game is created for people of all ages but the target audience is children 7 yo and up.

2) Game Market:

The Battlerine: war of submarines is a new game to be released in the online games market. Therefore, the Battlerine: war of submarines game is to be a market research subject.

3) Parents:

The Battlerine: war of submarines is suitable for young age players and parents are often concerned on how the game is influencing their child behavior and education.

The Battlerine: war of submarines is a strategy type game. The hands on user is the one who is interested in intellectual entertaining games. The game is suitable for any gender and educational level user who has an access to the computer with ethernet connection.

5d Maintenance Users and Service Technicians

The VMVA Development Group takes full responsibility of the game as well as technical support, game maintenance and updates development.

5e Other Stakeholders

The Battlerine: war of submarines game product is focusing on entertaining. However, the game requires users to develop strategy and therefore can carry educational purpose. Teachers and other educators may use the game as a smart entertaining experience for their students. Game elements, problem-solving skills, and discussions with fellow team members aid in facilitating active learning amongst end-users in the global market.

5f User Participation

The Battlerine: war of submarines is a complex development game. There are several stages in the development process prior final release.

Stages include: requirements, design, implementation, verification, maintenance.

User participation is desirable at every stage of the process.

Several demo versions are to be developed and released for hand on users to use and give feedback on. The demo games will differ in design implementation as well as difficulty of the rules. Besides the user feedback VMVA Development Group is interested in performance evaluation of the game. Based on use;s r evaluation the version decision is to be made.

5g Priorities Assigned to Users

VMVA Development Group is a non-profit educational organization. However the development and maintenance process requires time and resources. VMVA Development Group prioritese customers purchasing the game from online platforms and hand-on users purchasing full game for personal use.

6 Mandated Constraints

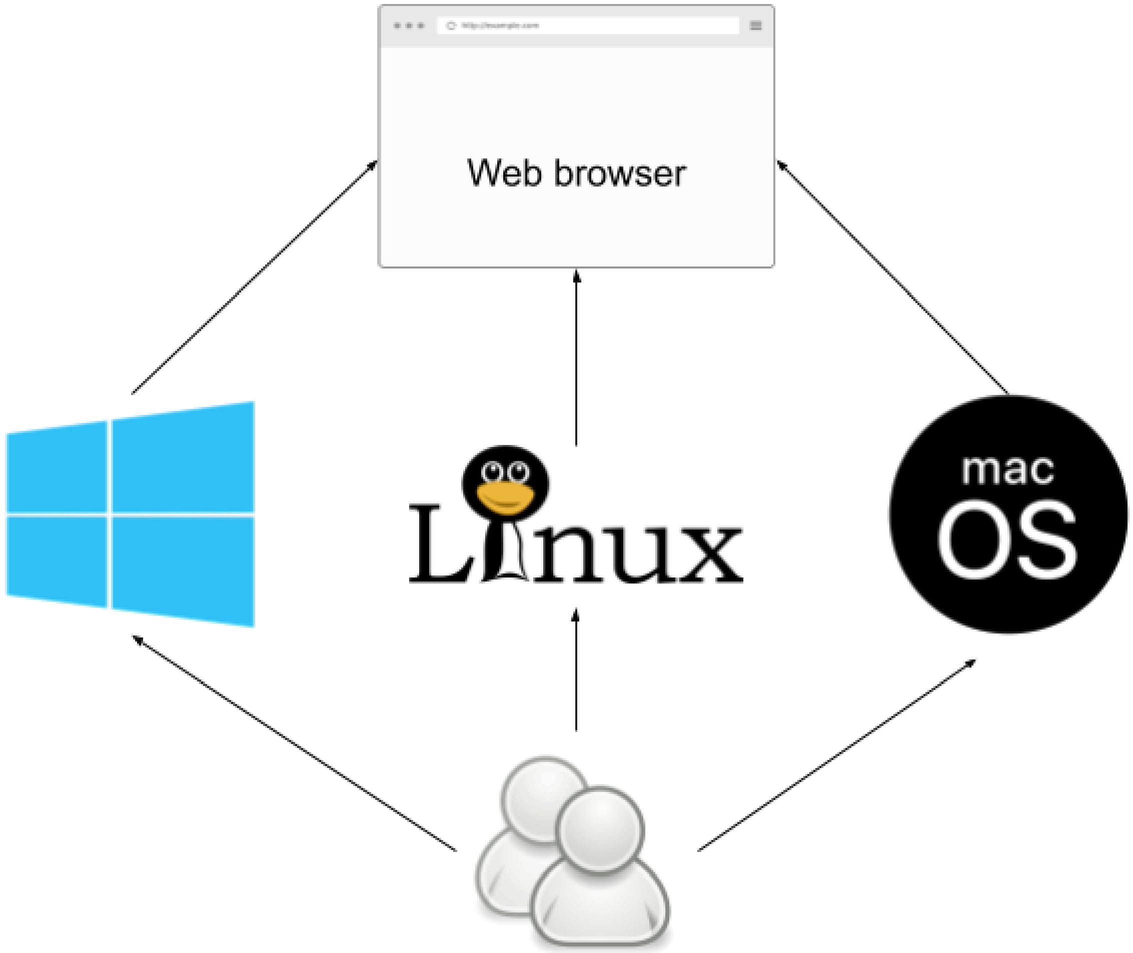
6a Solution Constraints

Description: The product should be a web-based game and function correctly in Google Chrome and Mozilla Firefox browsers.
Rationale: Users are likely to have different browsers and operating systems.
Fit criterion: The product shall be accessible from a browser.

Description: The product should properly handle the disconnection of any of the players and shouldn't crash for other players.
Rationale: Users are likely to have internet issues or the product may have unexpected code flow.
Fit criterion: The product shall continue operating correctly and count disconnected players as a defeated one.

Description: The product should operate correctly under a big number of users using it at a time.
Rationale: The product tends to be unique, therefore a big amount of users are expected.
Fit criterion: The product shall operate properly under 100,000 connected users.

6b Implementation Environment of the Current System



Users should be able to access the product no matter what operating system they use. The product should be able to function properly on Windows, MacOS and Linux operating systems. It is assumed that users have a good internet connection.

6c Partner or Collaborative Applications

The product should have an ability to create an account and/or login using both Google and Discord authentication services.
The product is not intended to have any of the data that should be used outside of itself.

6d Off-the-Shelf Software

The product has no requirements for any of the COTS software to be included. It is up to developers of the product to decide which software to use in the development and release phases.

6e Anticipated Workplace Environment

This product should be used in any environment regardless of surrounding conditions.

6f Schedule Constraints

There are no essential deadlines for this project.

6g Budget Constraints

Expected budget is 20,000$

7 Naming Conventions and Definitions

7a Definitions of Key Terms

This section is intended to clarify and define all terms used in the Battlerine: war of submarines game.
Base: base of operations for a naval fleet. Illustrated as buildings
Submarines: a watercraft capable of independent operation underwater.
Fleet: consist of submarines and belongs to the base
International waters: term used to describe territory that does not belong to any player and used to position a fleet.

7b UML and Other Notation Used in This Document

This document uses Graphical Notation Reference to build diagrams.

The Unified Modeling Language™ (UML®) is a standard visual modeling language intended to be used for

- modeling business and similar processes,
- analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.
Specifications Explained that process:

- provides guidance as to the order of a team's activities,
- specifies what artifacts should be developed,
- directs the tasks of individual developers and the team as a whole, and
- offers criteria for monitoring and measuring a project's products and activities.

Given some UML diagram, we can't be sure to understand the depicted part or behavior of the system from the diagram alone. Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams.

For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases.

Name of an abstract classifier is shown in italics while final classifier has no specific graphical notation, so there is no way to determine whether classifier is final or not from the diagram

8 Relevant Facts and Assumptions

8a Facts

According to “Strategy Games Market Research” conducted in 2018 (NEW YORK, Dec. 11, 2018 /PRNewswire) report on global board games market offers analysis on market size & forecast, market share, industry trends, growth drivers, and vendor analysis. The exponential growth of the entertainment and gaming industry is attributing to the rising demand for analog games in the market. According to the research strategy games influence factors such as problem-solving skills and critical thinking in team-oriented games allow players to build communication and relationship skills with consumers in the market. The global board games market is anticipated to reach values of more than $12 billion by 2023, growing at a CAGR of over 9% during 2017-2023.

8b Assumptions

The development of the Ballerine: war of submarines is based on strong assumption of the growing market of strategy games. The popularity of classic battleship games give us a strong belief that adding new features and expanding into multiplayer concepts will bring attention not only from hands-on users(gamers) but also will attract online game markets to research and invest into the development of the Battlerine: war of submarines.

Educational features of the game can not be underestimated. The game requires problem-solving skills and critical thinking as well as team-oriented skills and communication skills. Taking this fact into account it is assumed that games can be used for educational purposes.

Battlerine: war of submarines game will be created to serve entertaining purpose therefore it is to be associated with leisure and good time. The game design supports users' experience providing exceptional visual and sounds effects. Detailed design gives the user full experience.

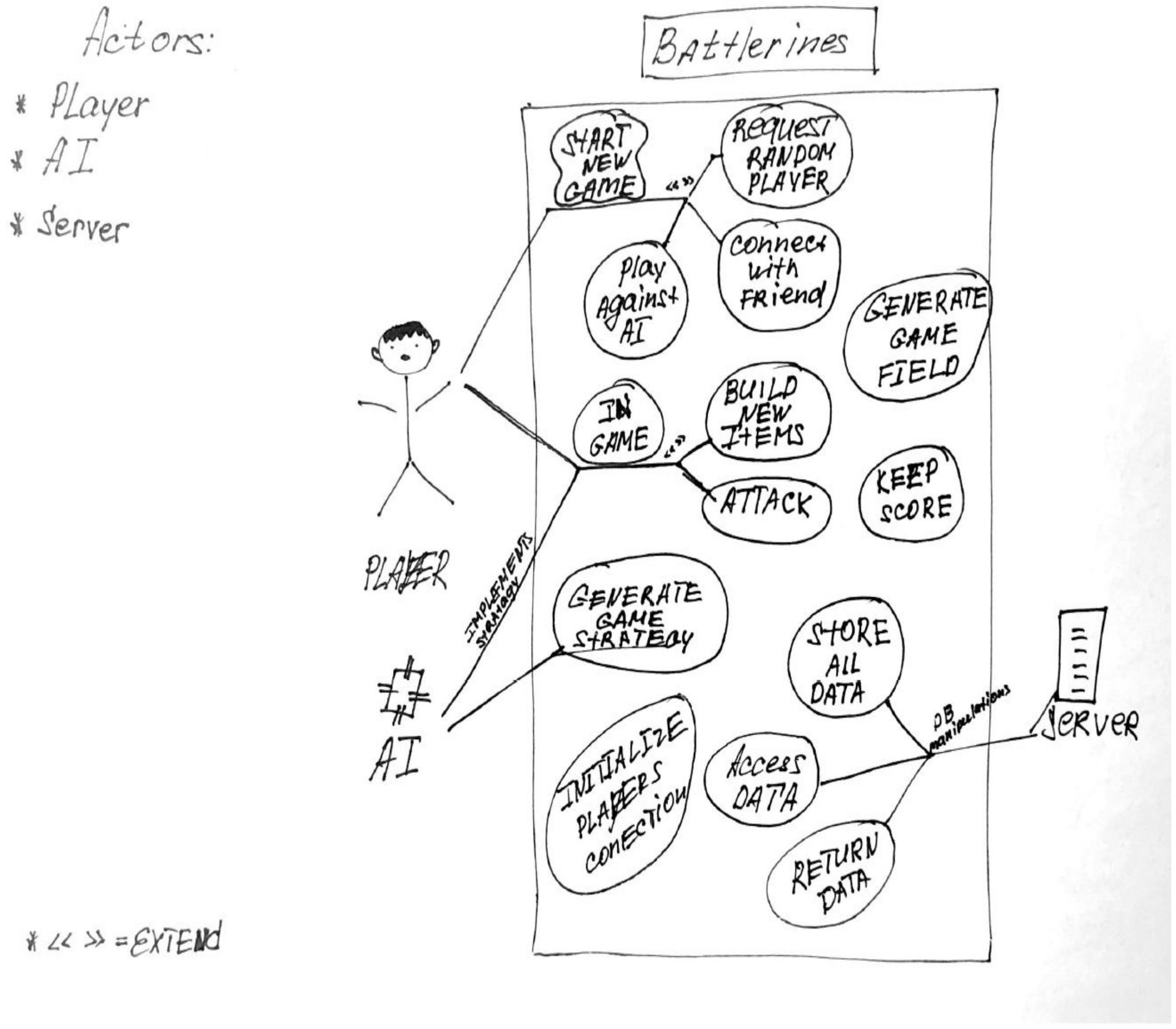
The Battlerine: war of submarines is a war-themed board game for multiplayer therefore there is a large number of players who are going to play the game at one time. It is assumed that every user will have ethernet access and suitable technology such as smartphones, tablets or laptops.

The game can be a platform for commercial advertising.

II Requirements

1 Product Use Cases

1a Use Case Diagrams



1b Product Use Case List

- User account manipulations:
    - Register
    - Login
    - Charge user

- Start New Game
    - Two Players OR MultiPlyer
        - Request Random Player
        - Connect With Friend
        - Play Against AI
- In Game
    - Build new Game Items
    - Attack the opponent

- DB
    - Store new Data
    - Request Data
    - Update Data
    - Return Data

- Game System
    - Generate New Game Field
    - Keep Score
    - Initialize players connection

1c Individual Product Use Cases

Use Case #2	**<u>RequestRandomPlayer</u>**

Description	*Players request to start a game with randomly picked player available on the server*
Actors	*Player1, Player2, server, game system*
Assumptions	*The game request and player connection is successful. Enters new scene .GameField implements .TwoPlayerGame*
Steps	*user request game by pressing "Two player game" button*
Flow of Events	*.twoPl waits for user to press* *.twoPl sends server request to initialize connection to opponent* *.gameState updated to two player game state* *server initialize connection between players*
Exit conditions	*Game was successfully loaded and system exited .PickTheMode scene*

Use Case #3	**<u>ConnectWithFriend</u>**
Description	*Players request to start a game with a friend*
Actors	*Player1, server, game system*
Assumptions	*The game request and player connection with friends is successful. Enters new scene .GameField implements .TwoPlayerGame*
Steps	*user request game by pressing "Friend game" button*
Flow of Events	*.friendG waits for user to press* *.friendG sends server request to initialize connection* *.gameState updated to friends game state* *random code generated and displayed on screen. when code entered by player's friend connection initialized*
Exit conditions	*Game was successfully loaded and system exited .PickTheMode scene*

Use Case #4	**<u>InGame</u>**

Description	*Game was successfully initialized*
Actors	*Player1, Player2, (or more up to 4 when extends .MultiPlayer) server, game system*
Assumptions	*Game system successful keeps score while sending game state to the server*
Enter Conditions	*successful game field initialization*
Flow of Events	*server initialize a connection, game logic starts*
Exit conditions	*.GameState.currentState changes to GAME_OVER flag*

Use Case #5	**BuildNewItem**
Description	*Players request to place new object on the .GameField*
Actors	*Player1, .GameField scene*
Assumptions	*The object was successfully placed on the .GameField .GameState is updated and server recorded changes*
Flow of Events	*.twoPl waits for user to press* *.twoPl sends server request to initialize connection* *.gameState updated to two player game state* *server initialize connection between players*
Exit conditions	*New item appears on the field*

Use Case #6	**Attack**
Description	*Players request to make an Attack*
Actors	*Player1,Player2, server, game system*
Assumptions	*The attack was made and successfully analyzed. the result of the attack was recorded and displayed to all players*
Steps	*player 1 picks location to attack and initialize attack by pressing "Attack" btn*
Flow of Events	*attack was initialize, implemented and result was*

	recorded and displayed
Exit conditions	*Attack was implemented Attack finished results displayed*

Use Case #6	**Charge Customer**
Description	This use case charges the order currently being placed to a credit card.
Actors	*User, server, game system, bank System*
Assumptions	*The purchase is successfully made, or user is notified of frailer.*
Steps	1. The use case begins when a user selects "Credit Card" as a payment option, while in use case *Place Order* 2. The system requests the credit card number, type and expiration date. 3. The user enters the information. 4. The system verifies that the credit card is valid for the amount to be charged and completes the credit card transaction. The system stores the payment details and returns a success message
Errors to handle	If the credit card cannot be validated the use case ends, returning a failure message
Exit conditions	The customer has been charged for the order.

2 Functional Requirements

- **Database**: Server shall have full database functionality for data manipulation and storage. That functionality must include: *Requesting, Setting, Updating and Returning* data.
    - Fit Criterions:

        - *If a specific data was requested, then the db should only return that data, not anything else.*
        - *All queries should be processed appropriately, giving the right data as the result*

- **Server**:
    - Server shall be able to handle two or more users and all interactions between them (shooting, moving the ships, placing new ones, etc.) as well as keeping a steady socket connection with each client in the game in order to pass the data back and forth instantly, without losing any client connection.
    - Fit criterions:
        - *Passing updated client data to the server should not cause any errors, but reflect them on all other clients by sending updated data through the server to them.*
        - *Connecting two or more players to a game should not cause any errors, but successfully assign them to an instance of game and keep a steady socket connection.*
        - *If there are 2 players waiting for the game to be found, the server should connect them without any errors and start the game session.*
- **Auth**:
    - Server should have an Authentication system that stores personal game information of each user as well as their login credentials they filled during the registration form.
    - Fit criterions:
        - *Requests for a specific user data should only return that user's data not any other one.*
        - *Setting or Updating a particular user data entry in the database should not cause any other users' updates or sets apart from that particular user entry.*
        - *Users should be able to login with their credentials without any errors, apart from those that include mistyping and other similar ones.*
- **Client**:
    - Client shall display the right map to a user.
    - Client shall generate new game fields during each new game the user starts.
    - Client shall handle all server incoming data and correctly reflect that updated data on the game field during each game.
    - Client shall keep a score of each player during the game.

    - Fit criterions:
        - *Client should not reveal the enemy map to a user at any time during the game.*
        - *Client should not generate any errors while processing server incoming data from the other players in an on-going game*

3 Data Requirements

- System should store:
    - Encrypted User Credentials
        - Rationale:
          Privacy and security of the personal data.
          This also allows the system to have login/register functionality.
        - Fit Criterion:
          Original data incoming from the user should be ensured that it should not be visible in its raw (decrypted format) anywhere in the system. User data should strictly be encrypted in the whole system.
    - Game Entities Data
        - Rationale:
          This allows us to avoid getting modified data from the client on the server which would mean cheating. Thus, keeping every entity specific data on the server such as (damage of a submarine dealt, displayed part of the map, max number of available units on the map, etc.) keeps us safe and away from data modifications.
        - Fit criterion:
          Make sure there is no crucial data coming from the client to the server that can be modified in order to get some sort of an advantage.
    - Game Sessions
        - Rationale:
          In order for a system to maintain properly a game process between players it should store its game session along with players entities coordinates, status of bases and submarines on the map.
        - Fit criterion:
          In case if a player disconnects and connects back due to the temporal internet connection loss, a player should be placed back to the game process with the same state they had before they got disconnected.

4 Performance Requirements

4a Speed and Latency Requirements

The purpose of this section of the document is to outline the Software Performance Goals for Product Battlerines: The War of Submarines. These are the goals that are based on general advice on response time from Jakon Nielsen book on Usability. Where the 10 seconds limit for response is considered maximum to keep user focused/uninterrupted.

Any interface between a user and the automated system shall have a maximum response time of 3 seconds.

- Server needs to be available at any time
- Connection between random players needs to be initialized within 10 seconds.
- Connection with friend limit is 1.5-3 second
- The response shall be fast enough to avoid interrupting the user’s flow of thought.
- The database request should be fulfilled within 1.5 seconds

The system should limit all wait times to 10 seconds, which is about the limit for keeping the user’s attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

Considerations: Response time may vary based on the ethernet connection on the user's end.

4b Precision or Accuracy Requirements

This section of report is dedicated to accuracy of game system performance.

(1) It is expected that the game would perform the logic correctly.

(2) AI systems must have an ability to generate game strategy based on the players level of performance determined by experience and games previously won.

(3) Server is assumed to be provide strong and uninterrupted connection between the players

4c Capacity Requirements

This section specifies the volumes that the product must be able to deal with and the amount of data stored by the product. Our game logic is designed to be stored on cloud and be dynamically loaded as the game progresses and develops into the local

game directory. The game is not graphically heavy, therefore does not require large memory capacity.
To ensure that ethernet connectivity is required for games to be initialized.
HARD DRIVE: At least 5 GB of free space with at least 500 MB additional space for custom content and saved games.

5 Dependability Requirements

5a Reliability Requirements

- This product provides 90% confidence that all the data and game state would be saved in case of unexpected exit or disconnection.
- In case of server failure the game creates the local copy of the game state, so it would be saved in the server when connection is restored.
- This product provides an intuitive game navigation system.

5b Availability Requirements

Battlerines is a desktop game is available for everyone who meets the set up and connection requirements:

- The product shall be available for use 24 hours per day, 365 days per year.
- Ethernet connection is required.

5c Robustness or Fault-Tolerance Requirements

Battlerines: The war of submarines relies on storing game data into the server to insure player's account availability from any device. Therefore, connection to the server is required.

- Disconnection from the server may be considered an abnormal environment.
- Therefore, a product must be able to provide all of its functionality after or during some unusual situation.
- The product shall continue to operate in local mode whenever it loses its link to the central server.
- While disconnected from the server game system must create a copy of all game states changes for each player.

6 Maintainability and Supportability Requirements

6a Maintenance Requirements

- VMVA Development Group reserves the right to collect all the data related to system failure and use this information in maintaining purposes.
- Users would be notified via in game notifications.
- If an update is required for optimal system performance, the game should not be able to exit pop-up unless the update is installed.

6b Supportability Requirements

User support is available via email that can be found in the game settings.
email: vmvadev@support.com

6c Adaptability Requirements

The Battlerines: The war of submarines is expected to run on the systems meeting this requirement:

- Memory:2 GB
- Graphics Card:NVIDIA GeForce 8500 GT
- CPU:Intel Pentium 4 2.00GHz
- SIMPLE GAME File Size:200 MB
- OS:Windows 7 +

6d Scalability or Extensibility Requirements

- The system must be able to handle around 10,000–12,000 players, 4000–5000 active simultaneously.
- Game is expected to gain wide popularity within the first 5 years of release.
- If a players' number is expanded over the server limit, usage of an additional server is recommended.

6e Longevity Requirements

- During the first year, while the new game concept is being adopted by users, it is expected to have increased user support requests.

- Battlerines:War of Submarines is expected to operate without major updates and within next 3 years.
- One major update every 3 year is recommended.

7 Security Requirements

7a Access Requirements

VMVA Development Group reserves the right to access all the data related to the project as well as modify and update if necessary.

- No body has access to the users account I info except the user who own the account.
- Every user with account has ability to look and check previous scores and game results by query.
- Data base access is limited to the members of the development group

7b Integrity Requirements

The Batteries: war of submarines guaranties safety of personal and financial data. To ensure safety program must support two end encryptions. Consideration: detected fraud attempts will result in immediate consent to check the personal data.

7c Privacy Requirements

- Login and password information stays encrypted for all parties involved.
- User is guaranteed to have personal account where he can look up all personal information he provided, make changes or remove it.
- The result of the game between multiple players must stay private and available for users involved in it.
- All user information stays private unless consent was given. (System must request consent)

7d Audit Requirements

To maintain the system safe and efficient semi-yearly audit is required. An audit is “a systematic, independent and documented process for obtaining audit evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled”.

Audit criteria:

- Safe user environment.
- Ban on swearing words
- Encryption of personal data
- Efficient and informative notifications and emails

Consideration: Audit is required. Qualified personnel are allowed to do the audit. All audit information must be reported in unified form to VMVA director of quality control.

7e Immunity Requirements

The Battlerines: war of Submarines is a server-based game that is require ethernet connection. The system must be equipped with strong antivirus system. User are guaranteed to have cyber safe environment while playing the game.

8 Usability and Humanity Requirements

The Battlerines: war of submarines guarantees to have intuitive user interface following the UX standards. It is a game that excels by taking a simple, well-trod formula and executing it with incredible style. Every new scene of the game must follow the design standards. (Can be found in design standards section)

8a Ease of Use Requirements

The Battlerines: war of submarines must rely on intuitive based navigation. Which includes common UI patterns such as:

- Hamburger menu symbol (leaving context hidden until needed)
- Icons must reflect its meaning (Ex: magnifying glass symbol equals searching)
- Full screen navigation options for navigating between game styles and rules.
- Gesture based navigation is used for interacting while game is in a progress

8b Personalization and Internationalization Requirements

The Battlerines: war of submarines is internationally available game and must support different languages and cultural differences if necessary. System is adjustable for users with disability. In addition, game provides theme choice for user

- The system support at least 5 languages
- User has ability to pick the language of his choice.
- The system must provide enlarged objects for users with vision disabilities.
- The system must provide sound volume control

- The system is implemented in at least 3 color styles and theme adjustment is available for user.

8c Learning Requirements

The Battlerines: war of submarines system provides clear description of game rules for all users.

- The system is recommended for users of age of 6 and up.
- Updated game rules must be always available for user during the game for the reference if needed without interrupting the game flow.
- All users age of 6 and up must be able to learn the game rules within one hour.
- No educational background is necessary for the users.

Constrains: Users not familiar with no basic computer skills might find some difficulties in navigation and might need extra training. VMVA Development Group can provide expanded navigation and rules guide for such user via request.

8d Understandability and Politeness Requirements

The Battlerines: war of submarines is internationally available game and must support different languages and cultural differences if necessary.

- The game system must use symbols and words that are only socially acceptable.
- The symbol usage across the system must be checked for international political and ethical correctness.
- The game system must not use "swear" words
- The game system must guarantee safe environment for user at any location of the word.

8e Accessibility Requirements

The game system does not need to require users to sign up with account in order to have full access to the game. However, users are able to keep track of all the games played by signing up with an account.

- User reserves the right to play without creating an account
- Only signed users are able to see games history and scores
- All in game purchases are available for users with confirmed accounts
- Only users with confirmed account can participate in game chats

Considerations: Users without an account can make one-time purchases, however, the purchase would be gone when cycle is ended.

8f User Documentation Requirements

The Battlerines: war of submarines' game system must provide an access for the user where he can find all the information related to the game rules and environment navigation.

- User is quarantined to have access to brief game rules at any point of the game.
- Accessing rules must not affect the game flow.
- Documentation must be written in friendly and clear language
- Documentation must provide information on symbols used in the game.

8g Training Requirements

The Battlerines: war of submarines does not require additional training that is not specified in user documentation provided for every user who has access to the game.

Constrains: If additional training is necessary: expanded documentation can be requested via email (vmva.dev@support.com).

9 Look and Feel Requirements

9a Appearance Requirements

- The Battlerine: War of submarines is an educational multiplayer game and must be designed in warm colors.
- The color-scheme must be appealing for all genders.
- The game must not have bright/neon colors.
- The game must be visually pleasing and need to avoid sharp figures and lines

9b Style Requirements

The game design and style must follow all the design requirements provided in "Design" documentation.

- The game needs to give an impression of cartooned battlefield
- Warm color-scheme is used to give user positive feeling from the game navigation and the game itself.
- The game should avoid loud, disturbing sounds
- The game style must be appealing for all users of different age (6+)

10 Operational and Environmental Requirements

10aExpected Physical Environment

The game requires to have computer with mouse navigation and keyboard

*no specific requirements are necessary.

* User is responsible for physical safety while playing the game

10b Requirements for Interfacing with Adjacent Systems

- The products shall work on the last four releases of the five most popular browsers
- The product must be cross platform and available on Linux systems and Windows systems

10cProductization Requirements

- The product must be downloaded from the server and installed by the user locally on the device of his choice.

10d Release Requirements

- Each release shall not cause previous features to fail
- User should be notified of every new release.
- Installation of new release is done via the server.
- If update is necessary for optimal performance, it can be forced.

11 Cultural and Political Requirements

11aCultural Requirements

- The product shall not be offensive to religious or ethnic groups.
- The product shall keep a record of public holidays for all countries in the European.
- The product must be available in different languages.

11b Political Requirements

The Battlerine: war of submarines shall stay apolitical.

12 Legal Requirements

12aCompliance Requirements

- Personal Information (username/password) should not be known to anyone and could not be accessed anywhere in the system, only in encrypted form.
  - Rationale: To avoid personal data thefts
  - Fit criterion: Ensure that it is impossible to access personal data and get it in decrypted form from anywhere in the system.

12b Standards Requirements

- Product should comply with GDPR standards.
- Product should be developed using agile software development methodologies.
- Product should be developed using any version control software to avoid delays during development.
- Product should comply with ISO/IEC/IEEE 29119-1 standards.

III Design

1 System Design

1a Design goals

Content

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

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

Your text goes here . . .

2 Current Software Architecture

SV:

Your text goes here . . .

3 Proposed Software Architecture

3a Overview

SV:

Your text goes here . . .

3b Class Diagrams

SV:

Your text goes here . . .

3c Dynamic Model

SV:

Your text goes here . . .

Content

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

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

3d Subsystem Decomposition

SV:

Your text goes here . . .

3e Hardware / software mapping

SV:

Your text goes here . . .

3f Data Dictionary

SV:

Your text goes here . . .

3g Persistent Data management

SV:

Your text goes here . . .

3h Access control and security

SV:

Your text goes here . . .

3i Global software control

SV:

Your text goes here . . .

3j Boundary conditions

SV:

Your text goes here . . .

4 Subsystem services

SV:

Your text goes here . . .

5 User Interface

SV:

Your text goes here . . .

6 Object Design

6a Object Design trade-offs

SV:

Your text goes here . . .

6b Interface Documentation guidelines

SV:

Your text goes here . . .

6c Packages

SV:

Your text goes here . . .

6d Class Interfaces

SV:

Your text goes here . . .

IV Test Plans

1 Features to be tested / not to be tested

SV:

Your text goes here . . .

2 Pass/Fail Criteria

SV:

Your text goes here . . .

3 Approach

SV:

Your text goes here . . .

4 Suspension and resumption

SV:

Your text goes here . . .

5 Testing materials (hardware / software requirements)

SV:

Your text goes here . . .

6 Test cases

SV:

Your text goes here . . .

7 Testing schedule

SV:

Your text goes here . . .

V 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.

<u>Content</u>

Any type of requirement.

<u>Motivation</u>

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

<u>Considerations</u>

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 . . .

VI 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 . . .

VII 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.

VIII 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. }