

The Forge: A Reforge Companion App

# Analysis Report

Requirements Engineering & Analysis

Rachel Matthews  
Spring 2021

## Table of Contents

<b>The Process of Requirements Analysis .....</b>	<b>2</b>
<b>Characteristics of individual Requirements: .....</b>	<b>2</b>
<b>Characteristics of a set of Requirements:.....</b>	<b>3</b>
<b>Analyzing Requirements: The Forge .....</b>	<b>5</b>
<b>Requirement Attributes: .....</b>	<b>5</b>
<b>Conceptual Modeling: .....</b>	<b>6</b>
<b>Critical Thinking .....</b>	<b>7</b>
<b>Table of Requirements .....</b>	<b>7</b>
<b>Ethical and Professional Responsibilities in Engineering .....</b>	<b>8</b>

## The Process of Requirements Analysis

Requirements analysis is the process of defining the needs or conditions to meet the new or altered project/product. It considers possibly conflicting requirements of various stakeholders. The tasks include analyzing, documenting, validating, and managing software and systems requirements. Individual requirements are looked at in detail, making sure that they adhere to the characteristics of individual requirements. This includes if they are necessary, appropriate, unambiguous, complete, singular, feasible, verifiable, correct, and conforming. Each one of these attributes is described in the table below.

### Characteristics of individual Requirements:

<b>Necessary</b>	<ul style="list-style-type: none"><li>• Requirement defines a capability, characteristic, constraint, and/or quality factor that is essential.</li><li>• Planned expiration dates or applicability dates are clearly identified.</li></ul>
<b>Appropriate</b>	<ul style="list-style-type: none"><li>• The amount of detail and the specific intent of the requirement is appropriate to the level of the entity that it is referring to.</li><li>• Avoid unnecessary constraints on the design or architecture and allowing implementation independence to the extent possible.</li><li>• You do not want so much detail in the requirement describing a solution that you might miss other possible solutions to the requirement. Do not cause unintentional constraints.</li></ul>
<b>Unambiguous</b>	<ul style="list-style-type: none"><li>• The way that the requirement is stated is simple, easy to understand.</li><li>• Can only be interpreted in one way.</li></ul>
<b>Complete</b>	<ul style="list-style-type: none"><li>• Adequately describes the necessary capability, characteristic, constraint, or quality factor to meet the entity's need.</li><li>• Does not need other information to understand the requirement.</li></ul>
<b>Singular</b>	<ul style="list-style-type: none"><li>• Only states one capability, characteristic, constraint, or quality factor.</li><li>• Can have multiple conditions under which the requirement is met.</li></ul>
<b>Feasible</b>	<ul style="list-style-type: none"><li>• The requirement can be realized within system constraints, with acceptable risk.</li></ul>

<b>Verifiable</b>	<ul style="list-style-type: none"> <li>Structured and worded so that it can be verified to the customer's satisfaction at the level the requirement exists.</li> <li>Enhanced when the requirement is measurable.</li> </ul>
<b>Correct</b>	<ul style="list-style-type: none"> <li>An accurate representation of the entity's need that it was derived from.</li> </ul>
<b>Conforming</b>	<ul style="list-style-type: none"> <li>Individual items conform to an approved standard template and style for writing the requirements when possible.</li> </ul>

Requirements must be analyzed after elicitation to 1.) discover the bounds of the software and how it must interact with its operational and organizational environment. 2.) detect and resolve and conflicts between the requirements. 3.) to elaborate on system requirements to be able to derive the software requirements. After individual requirements are analyzed they must be viewed as a set to ensure that they form a complete, consistent, feasible, comprehensible, and can be validated as a full set of requirements set.

#### Characteristics of a set of Requirements:

<b>Complete</b>	<ul style="list-style-type: none"> <li>The requirement set stands alone such that it sufficiently describes the necessary capabilities, characteristics, constraints, and/or quality factors to meet the entity needs without needing other information. In addition, the set does not contain any to be defined (TBD), to be specified (TBS), or to be resolved (TBR) clauses. (IEEE 29148-2018 5.2.6)</li> </ul>
<b>Consistent</b>	<ul style="list-style-type: none"> <li>The set of requirements contains individual requirements that are unique, do not conflict with or overlap with other requirements in the set, and the units and measurement systems they use are homogeneous. The language used within the set of requirements is consistent, i.e., the same word is used throughout the set to mean the same thing. (IEEE 29148-2018 5.2.6)</li> </ul>
<b>Feasible</b>	<ul style="list-style-type: none"> <li>The requirement set can be realized within entity constraints (e.g., cost, schedule, technical, legal, regulatory) with acceptable risk. (Note: Feasible includes the concept of "affordable".) (IEEE 29148-2018 5.2.6)</li> </ul>
<b>Comprehensible</b>	<ul style="list-style-type: none"> <li>The set of requirements must be written such that it is clear as to what is expected by the entity and its relation to the system of which it is a part. (IEEE 29148-2018 5.2.6)</li> </ul>

<b>Able to be validated</b>	<ul style="list-style-type: none"> <li>It must be able to be proven the requirement set will lead to the achievement of the entity needs within the constraints (such as cost, schedule, technical, legal, and regulatory compliance). (IEEE 29148-2018 5.2.6)</li> </ul>
-----------------------------	---

Classification of requirements can be done in many ways. They can be classified as functional and non-functional. By priority, where the higher the priority the more essential the requirement is for the main goals of the software. Classification allows for determining if the requirement is derived from high-level requirements or an emergent property, or if it is being imposed on the software by a stakeholder or other source such as a governing body. Also, if the requirement is on the product or the process. Requirements on the process are constraints on the software engineering process to be adopted, or the standards to adhered to.

- Product requirement is a **need or constraint on the software** to be developed.
- Process requirement is a **constraint on the development** of the software.
- **Functional requirements** - **describe what the software will do**. The functions that the software will execute. Also, known as features or capabilities. A finite set of test steps can be written to validate the requirements behavior. Originate from user stories, use cases, and business rules. Stated in the form of input to be given to the system, the operation performed, and the output expected.
- **Nonfunctional requirements** - **define some observable characteristic or quality of the system**. Also known as constraints or quality requirements. Can be classified further depending on if they are performance requirements, maintainability requirements, safety requirements, reliability requirements, security requirements, interoperability requirements, or one of many other types of software requirements.

Conceptual modeling is used to depict system boundaries. Where the boundary separates the users or systems in the external environment(actors) from the internal behavior of the system. Many modeling notations are part of the Unified Modeling Language (UML). One model that is developed are Use Case diagrams, each use case shows an example of the functionality

of the system. Other diagrams include data flow models, state models, user interactions, object models, and data models.

## Analyzing Requirements: The Forge

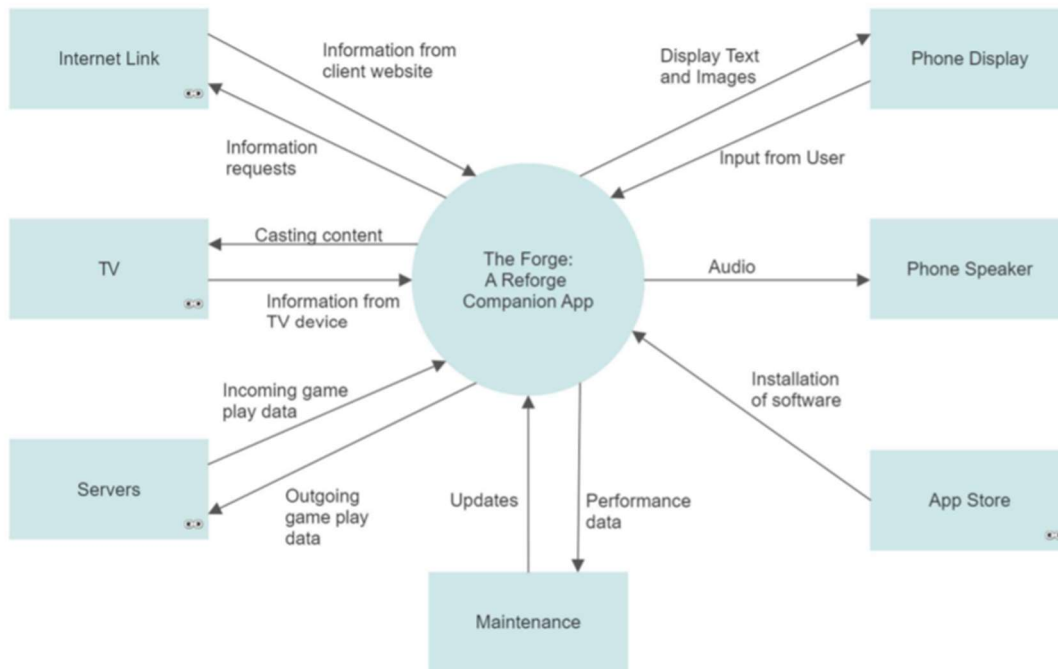
After eliciting the requirements for my project, I set about the task of analyzing them. First each requirement was compared to the desired characteristics that were previously mentioned. I made sure that each requirement had the following attributes such as identification, a version number, stakeholder priority, and a type. This made it easier to understand how the requirements fit together as a set. Different conceptual models were created to help visualize the product in situations of use. They reflected real-world relationships and dependencies and lead to the discovery of missed requirements. The first model that was created was a use case diagram. I used draw.io to create it. The second was a level 0 data flow diagram that depicted the application and its interactions with all outside entities. The process of creating my diagram started with related readings and videos to get an understanding of what a context diagram should look like and what needs to be included. I then looked over my system requirements to better understand how the system will function in its environment. To create my context diagram, I started out by defining the functional boundary of the system, noting its relationship to other related products and systems. I then identified the interfaces between the system and outside entities. Once I understood what was to be included, I started working on the actual diagram. I decided to use SmartDraw, which was a departure from draw.io which I normally use to create diagrams, because the video on it really sold me on being able to nest next level diagrams.

### Requirement Attributes:

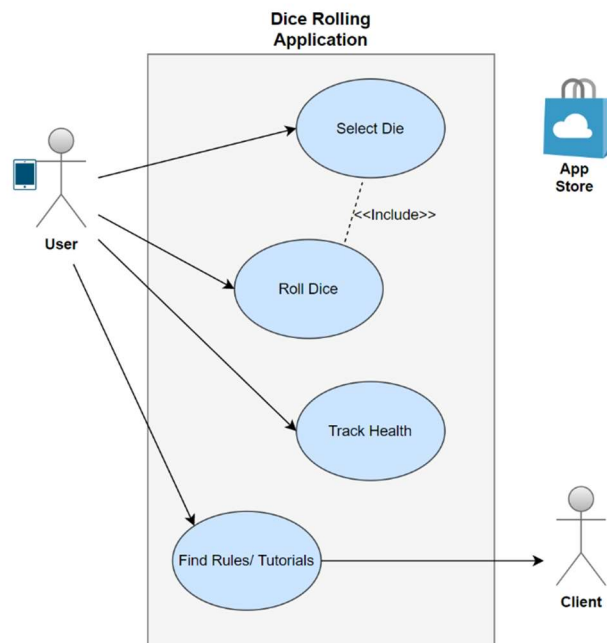
<b>Identification</b>	<ul style="list-style-type: none"><li>• This requirement was given a unique identifier of Dice_1 so that it can be easily traced and referenced throughout the development process.</li></ul>
<b>Version number</b>	<ul style="list-style-type: none"><li>• This requirement was given a version number to ensure that the correct form of the requirement is used and to track changes.</li></ul>
<b>Stakeholder Priority</b>	<ul style="list-style-type: none"><li>• This requirement was given a priority level of high to reflect the importance of the requirement to the overall system. This will make it easier to address issues when having to compare it to other requirements in the case of needed tradeoffs.</li></ul>
<b>Type</b>	<ul style="list-style-type: none"><li>• This requirement is a product requirement because it deals with the need on the software system. It is a functional requirement because it describes what the software will do.</li></ul>

## Conceptual Modeling:

### Level 0 Data Flow Diagram



### Use Case Diagram












## Critical Thinking



When analyzing my requirements, the information that was gathered during the elicitation phase was reviewed. I had to keep in mind the overall purpose of the product in relation to the business goals that it was trying to solve. The previously created Business Requirement Specification document was very useful in outlining the overall goals and objectives of the project. It also kept the problem at issue at the forefront when priorities were given to the different requirements. Priority was considered based on what was needed to achieve the goal, with keeping in mind the implications and consequences that could arise if certain requirements were not met. Assumptions that were documented earlier were reviewed and addressed when putting the set of requirements together.

## Table of Requirements

### Requirements

Key	Summary	Description	T	Linke d Issues	Labels	Risk	Diffic ulty
RS-54	The system shall adhere to privacy laws.	The app shall disclose to the user if any information or data is being collected. The system shall adhere to all privacy laws.		RS-34	limitation		
RS-53	The system shall be maintainable by app updates.	Updates or maintenance shall be done by updates that can be downloaded through the app store.		RS-34	limitation		
RS-52	The system shall be operable when disconnected from the Internet.	The system will operate normally without an Internet connection.		RS-40	non-functiona l		Low
RS-51	The system shall adjust to fit different screen sizes.	To ensure that the layout is flexible and adapts to different screen sizes, "wrap_content" and "match_parent" for the width and height of most view components, instead of hard-coded sizes will be used.		RS-37	non-functiona l		Medium
RS-50	The system shall be no larger than 150MB.	The app must be 150MB or smaller to be published to the google play store using the Android App Bundle so that Google Play can use the app bundle to generate, sign and serve optimized APKs for each user's device, resulting in a smaller app to download and install.		RS-42 , RS-34	limitation		Low
RS-45	The system shall meet quality guidelines for the google play store.	<b>Test against the quality guidelines.</b> Quality guidelines for all apps plus specific criteria for tablet, TV, and Auto apps provide testing templates. Use these to confirm that the app offers the basic UI design, features, and functions expected by Android users		RS-44 , RS-38 , RS-49	limitation	Medium	
RS-44	The user shall be able to download the app from the app store.	The app must be available in the google play store, and the apple store to reach the target audiences.		RS-43 , RS-45 , RS-49		Medium	Medium
RS-43	The app shall target at least an Android 9.0 (API level 28).	Target a recent API level  Google Play requires that new apps target at least Android 9.0 (API level 28), and that app updates target at least Android 9.0 starting on November 1, 2019.		RS-34 , RS-44 , RS-49	non-functiona l	Low	Low
RS-40	The system shall provide the user with a link to the clients website, that can be accessed directly from the system.	A link to the Secret Boss games website will be accessible directly in-app.		RS-52 , RS-29	functiona l	Medium	Low



RS-38	If the app is accidentally closed and then reopened, the system shall store the current information upon the closure of the app and display the information when the app is reopened.	The system should allow for the closure of the app while storing the information in case the user needs it at a later time.		RS-45	functiona l	Medium	High
RS-31	The user shall be able to select a die from the following die choices, four-sided, six-sided, eight-sided, ten-sided, or twelve-sided.	The user shall be able to select from the specified die choices.  Relates to issues RS - 10 to RS - 16.		RS-10 , RS-12 , RS-11 , RS-14 , RS-46 , RS-13 , RS-16 , RS-15	functiona l	Low	Medium
RS-30	The system shall allow the user to change the color the dice.	The system will allow the users to change the color of the dice from a set of choices, to match the starting player colors in the Reforge games. (Red, Blue, Green, Purple, Orange)  Related to <a href="https://rachelaliko.atlassian.net/browse/RS-12">https://rachelaliko.atlassian.net/browse/RS-12</a>		RS-12	functiona l	Low	Medium
RS-29	The system shall enable the user to access rules and tutorials for the Reforge games.	Relates to <a href="https://rachelaliko.atlassian.net/browse/RS-18">https://rachelaliko.atlassian.net/browse/RS-18</a>  The system shall have a rules/tutorials page that users can access.		RS-40 , RS-18	functiona l	Low	Medium
RS-28	The system shall allow users to change the theme of the display.	Relates to <a href="https://rachelaliko.atlassian.net/browse/RS-19">https://rachelaliko.atlassian.net/browse/RS-19</a>  The user has the ability to change the theme of the display.		RS-19	functiona l	Medium	High
RS-27	When the User clicks the plus button on the side of the health tracker, the system shall increase the health count by 1 for each click, until it reaches a health count of 20.	Relates to <a href="https://rachelaliko.atlassian.net/browse/RS-17">https://rachelaliko.atlassian.net/browse/RS-17</a>  The system shall increase the health count by 1 for each click until it reaches max health of 20. Any clicks afterward will have no effect.		RS-11 , RS-17	functiona l		Low
RS-26	When the User clicks the minus button on the side of the health tracker, the system shall decrement the health count by 1 for each click, until it reaches zero.	Relates to <a href="https://rachelaliko.atlassian.net/browse/RS-17">https://rachelaliko.atlassian.net/browse/RS-17</a>  The health count will decrement by 1 for each click until it reaches zero, but will not go any lower.		RS-11 , RS-17	functiona l	Low	Low
RS-25	The User shall be able to click a button to cycle through the available die choices.	Relates to user story <a href="https://rachelaliko.atlassian.net/browse/RS-20">https://rachelaliko.atlassian.net/browse/RS-20</a> . Enables the user to cycle through die choices.		RS-20 , RS-46	functiona l	Low	Medium
RS-24	When the simulated die is rolled the system shall generate a random number within the range of the corresponding die selected.	Relates to User stories RS - 10 to RS - 14.  Needed to allow the user to use the app as a substitute for physical die.		RS-10 , RS-14 , RS-46 , RS-13 , RS-16 , RS-15	functiona l	Low	Low

## Ethical and Professional Responsibilities in Engineering

To recognize ethical and professional responsibilities in engineering situations you must be aware of the impact your actions and how the solutions you create will impact global, economic, environmental, and societal contexts. The easiest way to be aware of these responsibilities is to learn and adhere to the professional standards and guidelines. These guidelines include The Software Engineering Code of Ethics and Professional Practice published by the ACM, as well as the NSPE Code of Ethics for Engineers. Below is a condensed version of the ACM code of ethics taken from <https://sites.google.com/site/profvanselow/success/ethics>

### Software Engineering Code of Ethics and Professional Practice (short version)

<b>1. Public</b>	Software engineers shall act consistently with the public interest.
<b>2. Client and Employer</b>	Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
<b>3. Product</b>	Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
<b>4. Judgement</b>	Software engineers shall maintain integrity and independence in their professional judgment.
<b>5. Management</b>	Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
<b>6. Profession</b>	Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
<b>7. Colleagues</b>	Software engineers shall be fair to and supportive of their colleagues.
<b>8. Self</b>	Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

The Software Engineering Code of Ethics and Professional Practice (ACM full):

<https://ethics.acm.org/code-of-ethics/software-engineering-code/>

NSPE Code of Ethics for Engineers:

<https://www.nspe.org/resources/ethics/code-ethics>