**Group 1 Investigation Requirements Engineering**

**GitHub Wiki:** [**https://github.com/djgamekid/GDP-Group-I-bearcatmanager/wiki/Investigation-Requirements-Engineering-%E2%80%90-Updated**](https://github.com/djgamekid/GDP-Group-I-bearcatmanager/wiki/Investigation-Requirements-Engineering-%E2%80%90-Updated)

Manikala Chevitipalli

Sai Konda

Anthony Jack

Darren Ross

1. What is requirements engineering?

**Requirements engineering is the overall process of gathering information and requirements for a software project from a client. This process is the core principle of software development and allows development teams to transform the idea into an overall program or product. [1]**

1. In an agile/iterative software development setting, will you undergo this process once or multiple times?

**In an agile/iterative setting, this process will be completed multiple times as features are added and discussed. Requirements are completed in a “sprint” which is typically a period of work and then reflected and maintained by the development team. The development team’s project leader will be in contact with the stakeholders to ensure the requirements are implemented as closely as they can, which is typically done with more requirement engineering activities. [6]**

1. What are the different requirements engineering activities? Who takes part in those activities?

**In terms of software engineering, you have the project definition and initiation along with gathering requirements. You should then design and mock the flow of the product after requirements are gathered. Implement and test the software and finally, the team maintains or repeats per new feature/requirement.**

**The software team leader, client, and stakeholders would be part of the project definition and gathering requirements. The software team developers and lead would design and mock the flow of the product. Sometimes the stakeholders are a part of this for visualization. Implementing will be the developers, and the testing will be the stakeholders and software team leader. [3]**

1. Define each of the participants in the above activities (example: developer, stakeholder)

**Software Developers: These members of the team are the essential programmers who have extensive knowledge of producing software and code for the overall product. They will work together to collaborate and produce efficient code and design the product.**

**Stakeholders: These members are company employees or clients who are the main overseers of the idea/product that is requested of the software teams. They should know the requirements and purpose of the software and be in communication with the team with updates or if anything changes with the requirements.**

**Software Lead: This person is the leader of the software team guiding their team to be on track with deadlines. This person also works to assist the development team with understanding information from the stakeholders. This person should also have extensive knowledge about development and coding to ensure the product is efficient. Lastly, the lead should have great communication skills as they are the liaison between both the software team and client/stakeholder. [5]**

1. For each of the roles in the above two questions, identify who fills those roles (for example, your team is the developer(s), etc). Remember that one person/group can play multiple roles!

**Our team would be the software developers. Our software team leader would be Darren, as he is responsible for our work and would communicate with the stakeholders. The stakeholder in our class would be professor Chai as he’ll help us understand what is needed to produce this final product. Anthony would be a designer and developer on our software team. Sai would be a tester and developer, and Manikala would be too.**

1. Define functional requirement.

**The Functional Requirements are basically that the end user specifically demands as a basic facility that the system should offer, all these functionalities need to be incorporated into the system as a part of the data flow. They are represented or stated in the form of input to be given to the system, the operation performed, and the output expected. They are the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements. [1]**

1. Define non-functional requirement.

**A non-functional requirement defines the quality attribute of a software system. Non-functional requirements are specified by technical peoples e.g. Architect, Technical leaders and software developers. It is captured as a quality attribute. It is applied to the system as a whole. It helps you to verify the performance of the software. The Performance Testing, Stress testing, Usability testing, Security Testing are outcomes under non-functional requirements. [1]**

1. What project artifacts (documents, etc) should result from requirements engineering?

**Project artifacts that should come from requirements engineering are the product/software description, design expectations, software requirements, target users, deadlines, and data/process workflow. All of these would come from detailed requirements engineering and produce various documents and diagrams that will be used by both the stakeholders and software development team. [4]**

1. Define requirements elicitation

**Requirements elicitation is the process of creating and defining a software project's requirements. This is done through communication with the users, clients, and whoever else has any kind of stake in the project. [2]**

1. What are the sequence of steps one should take during requirements elicitation? Which step is potentially the hardest
   1. **Identify all the stakeholders, so this is the users, the clients, the developers, the customers**
   2. **List all requirements given to you from the customer/client**
   3. **Assigning a value to each of the requirements you just listed out which will give you a sense of things that need to be done first and what things can be put on the backburner, some call this the degree of importance**
   4. **Categorize the list into a couple different sections**
      * **It's possible to achieve**
      * **It should be deferred and list the reason for deferring**
      * **It is impossible to complete and should be taken of the list of requirements [2]**

**The hardest step is potentially the second step when you are writing down what the client wants: it's where most of the communication errors can happen, they can say they want one thing when they want a completely different thing.**

1. What are key things you should try to identify when working with the client? What kinds of questions should you ask?

**What they want from you exactly how they are thinking it, most of the problems from these kinds of projects come directly from the miscommunication from the client/developer interaction and can lead to a project going off the rails very early on in development. Some questions that would be great to ask are “What are some absolute must haves for this project?”, “What’s the number one thing is you are hoping to achieve with this project?”, and “Will this project be open for the public or is this an inhouse private software?” [5]**

# Citations

[1] GeeksforGeeks. (2024, July 24). Requirements Engineering Process in Software Engineering. [https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/#](https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/)

[2] GeeksforGeeks. (2024, May 29). *Requirements elicitation - software engineering*. <https://www.geeksforgeeks.org/software-engineering-requirements-elicitation/>

[3] Hani, A. (2023, September 6). *Software engineering process activities*. Software Engineering Process Activities. <https://dev.to/anmarhaniv1/software-engineering-4h8b>

[4] *What is Requirements Engineering? An Overview and Intro into Requirements Engineering*. Jama Software. (2023, September 6). <https://www.jamasoftware.com/requirements-management-guide/requirements-gathering-and-management-processes/requirements-engineering>

[5] Shekhovtsov, V. A., Mayr, H. C., & Kop, C. (2013). Towards conceptualizing quality-related stakeholder interactions in software development. In *Information Systems: Methods, Models, and Applications: 4th International United Information Systems Conference, UNISCON 2012, Yalta, Ukraine, June 1-3, 2012, Revised Selected Papers 4* (pp. 73-86). Springer Berlin Heidelberg. <https://link.springer.com/chapter/10.1007/978-3-642-38370-0_7>

[6] Srinivasan , R., & Maloney, B. (Eds.). (n.d.). *What is scrum: Guide to the most popular agile framework*. What is scrum | Guide to the most popular agile framework. <https://www.scrumalliance.org/about-scrum>