Managing Software Engineering Project

Meeting Customer Requirements

Failure in software is an unfortunately common occurrence. Roughly 70% of software projects fail to meet their targets, and 50-60% of those failures can be attributed to poor requirements. The problem is that requirements cannot be simply “gathered” from the customer by listing out their demands verbatim. Often, the customer does not know the best way to solve a problem they have, or even what the problem is in the first place. Therefore, it is crucial to approach requirements from different angles, first by developing user stories and user journeys, then translating these into requirements and mock-ups. User stories can be gathered from various means, such as surveys, brainstorming sessions, and artifact analysis; however, direct one-on-one interviews are still one of the best ways to get to the heart of the customer’s goals.

Learning Outcomes

When completed successfully, this project will enable you to:

- Conduct one-on-one requirements gathering sessions with customers
- Build user stories, epics, and user journeys
- Design and define system requirements for a potential client
- Distinguish between functional and nonfunctional requirements
- Build software UX wireframe mockups based on requirements and user feedback
- Incorporate the needs of users into both the requirements gathering and interface design aspects of the software development process

Project Description

For this project, you will interview a potential client to elicit information, generate user stories, and then translate these into a list of system requirements. The strength of these requirements will depend upon how well you can understand the explicit and implicit nature of the customer's needs. You can complete this project either individually or as a group of no more than **three** people.

Lily's Florist Shop has reached out to you with a request for a new website to handle increased customer orders and manage inventory. The proprietor, Lily, has conveniently made herself available via online chat.

Conduct a **requirements-gathering interview** with Lily to try to understand the pain points for the customer and eliminate ambiguity. Consider a typical scenario with no errors, as well as divergent paths with dead ends and issues. Once you have a grasp of user expectations, you will need to develop user stories highlighting the proposed system's goals. Be sure to introduce yourself and build rapport, and then consider asking the following questions as examples:

- What features should your website have?
- What are your target goals?
- How do users interact with your site?
- How should users checkout?
- What images do you want to use?
- What pages should be included?
- Do you have a rewards program?
- What systems do you currently use?
- Do you use social media?
- Is your site hosted somewhere?
- What is your competition?
- What products do you sell?
- What types of events do you have?
- Do you offer promotions?
- What is causing you the most frustration?
- What kind of database would you like to use?
- What does an order require?
- What security features do you want?
- How many orders do you receive?

From this interview, generate **fifteen total user stories** and outline them in a working document. Next, take a goal defined by these user stories and draft a **user journey**. A user journey is a way of mapping out the different steps a user may take to reach their ultimate goal. Note: redundancy is not necessarily a bad thing when it comes to interfaces and user journeys. For example, clicking on the “Copy” button in a word processor, and using “Ctrl-C” as a shortcut may achieve the same goal, but offer alternative methods for different types of users.

Once the user journey has been generated, you will now create requirements for Lily’s Florist Shop. Develop a requirements specification document with a minimum of **twenty requirements**. Define in very clear, detailed terms what your system does (functional requirements) and any constraints that define how it should be done (non-functional requirements). While your user stories might describe the website as allowing users to log in to an account, a requirements document will provide more quantifiable detail. For example:

- “A user shall be allowed three login attempts before the system locks them out for 30 minutes”
- “Customer information shall be saved in a database for up to 5 years if no activity occurs on this account”

Requirements should be numbered and categorized (e.g., functional/nonfunctional, by interface, by purpose, etc.). You should consider a tool for gathering and reviewing requirements. While simply tracking these in a spreadsheet or document is sufficient, a requirements tool can have added value, such as traceability, user tracking, and configuration management. Your requirements should have some form of traceability back to the original user stories, either in a separate matrix or as a footnote in your documentation.

Finally, once requirements are established, create **mockup wireframes** of the user journey you defined earlier using Figma. These mockups must address some of your requirements, and you should include notes/annotations that describe how the interface elements are tied into these respective requirements. Don’t worry about your artistic skills; the purpose of the mockup is to provide a quick, rough sketch of the user journey. This will help you determine if elements are too complex or impossible to implement in a user format. They also serve as a useful visual aid when relaying your plans to the customer. Be sure to incorporate best practices in usability and the principles covered in the *Managing User Needs* course. Your mockup should include annotations that explain your design choices and what happens when certain features are interacted with.

Tips & Resources

- If the customer has competition, you may also want to investigate how companies in that field address their users' needs.
- Consider distinct user types when writing your stories. You may even want to develop specific personas with unique behaviors and attitudes.
- You may want to begin with an epic—essentially a large story with a broad narrative—then break it down into its constituent parts.
- Review your requirements for language ambiguity or unnecessary constraints. Are requirements too vague? Too specific?
    - Consider how these requirements are going to be verified at the end of development. Is there an obvious way to demonstrate that the requirement has been met?
    - Do your requirements address both functional and non-functional attributes?
- AI tools will increasingly support requirements gathering. The use of transcription services, including ChatGPT in audio mode can facilitate the turning of conversations into textual records. Various AI-enabled tools are available:
    - IBM Engineering Requirements Management
    - Miro AI | Miro
    - Dovetail
    - ChatGPT
- Tools for requirements management:
    - MindManager
    - Accompa
    - OpenText Dimensions RM
- Effective Requirements Management: https://www.pmi.org/learning/library/effective-requirements-management-project-success-8181
- Collaborative practices for software requirements gathering in software startups: https://ieeexplore.ieee.org/document/8817046
- User Journey Mapping 101: https://www.nngroup.com/articles/journey-mapping-101/
- Requirements Gathering Tools: https://blog.mindmanager.com/requirements-gathering-tools/
- Figma: Mockup Tutorial: https://www.youtube.com/watch?v=c9Wg6Cb_YlU

Submission Guidelines

Your submission should consist of a single PDF of at least four pages, with at least one page dedicated to each of the following deliverables:

- a link to your requirements-gathering discussion with Lily the Florist (use the "Share" button)
- all user stories (15)
- user journey
- requirements (at least 20),
- mockup interfaces

To submit your project, please click on the "Submit Project" button on your dashboard and follow the steps provided in the Google Form. If you are submitting your Managing Software Engineering project as a group, **please ensure only ONE member submits on behalf of the group**. You will also be prompted to upload the final page of your Group Project Agreement, which must be completed and signed by all group members. Please reach out to msse+projects@quantic.edu if you have any questions. Project grading typically takes about 3-4 weeks to complete after the submission due date. There is no score penalty for projects submitted after the due date, however grading may be delayed.

Plagiarism Policy

Here at Quantic, we believe that learning is best accomplished by "doing"—this ethos underpinned the design of our active learning platform, and it likewise informs our approach to the completion of projects and presentations for our degree programs. We expect that all of our graduates will be able to deploy the concepts and skills they've learned over the course of their degree, whether in the workplace or in pursuit of personal goals, and so it is in our students' best interest that these assignments be completed solely through their own efforts with academic integrity.

Quantic takes academic integrity very seriously—we define plagiarism as: "Knowingly representing the work of others as one's own, engaging in any acts of plagiarism, or referencing the works of others without appropriate citation." This includes both misusing or not using proper citations for the works referenced, and submitting someone else's work as your own. Quantic monitors all submissions for instances of plagiarism and all plagiarism, even unintentional, is considered a conduct violation. If you're still not sure about what constitutes plagiarism, check out this two-minute presentation by our librarian, Kristina. It is important to be conscientious when citing your sources. When in doubt, **cite**! Kristina outlines the basics of best citation practices in this one-minute video. You can also find more about our plagiarism policy here.

Project Rubric

Scores 2 and above are considered passing. Students who receive a 1 or 0 will not get credit for the assignment and must revise and resubmit to receive a passing grade.

Score	Description
5	• Overall project clearly addresses the customer's needs • Provides 15+ user stories • User stories are written well, clearly presenting the user roles, needs, and respective outcomes. Stories include perspectives from multiple user types • Includes a comprehensive user journey that directly relates to previously generated user stories • User journey provides information on user roles, steps, and details on the interactions involved for each step • Includes 20 or more defined requirements • Requirements are well-written, unambiguous, and quantifiable • Requirements include numbering, categorization, and can be traced back to the original user stories • Includes both functional and non-functional requirements • Includes one or more mockups of the system interface • Mockup includes annotations that explain the reasoning behind design choices as well as traceability back to the requirements • Project documents are free of spelling, punctuation, and grammatical errors
4	• Overall project clearly addresses the customer's needs • Provides 10-15 user stories • User stories are clear, presenting user roles, needs, and respective outcomes. Stories include perspectives from multiple user types • Includes a user journey that relates to previously generated user stories. • User journey provides information on user roles, steps, and details on the interactions involved for steps • Includes 15-20 defined requirements • Requirements are clearly defined and numbered, outlining potential system functionality. Some ambiguity may be present • Includes both functional and non-functional requirements • Includes at least one mockup of the system interface • Mockup includes annotations that explain the reasoning behind design choices

	• A few mistakes in vocabulary and grammar and/or formatting errors are present
3	• Overall project addresses the customer's needs. • Provides 10-15 user stories • User stories are simply defined and do not address different types of users. Stories include rolls, needs, and outcomes • Includes a user journey that partially relates to some of the previously generated user stories • Includes fewer than 15 requirements • Requirements are defined, though may be vague or ambiguous, and in need of further clarification and development • Includes a mockup of the system interface. Mockup provides a basic wireframe, but may be missing additional context • Several mistakes in vocabulary and grammar, with a potential impact on overall comprehension
2	• Overall project partially addresses the customer's needs. • Provides 10 or fewer user stories • User stories are poorly defined and do not provide an adequate coverage of the customer's needs • Includes a user journey with a simple flow, and does not relate to the original stories. • Includes 10 or fewer requirements • Requirements are poorly written and difficult to quantify. Further clarification is necessary • Includes a mockup of the system interface, but missing additional context. Mockup a very simplistic UX design with few elements • Numerous mistakes in vocabulary and grammar, with a potential impact on overall comprehension
1	• Overall project barely addresses the customer's needs, if at all. • Overall submission may be missing one or more of the required components: stories, journey, requirements, and mockup • Provided documentation may be poorly written, ambiguous, and in need of further development • Difficult to understand due to significant usage of incorrect vocabulary and grammar.

0	• The student either did not complete the assignment, plagiarized all or part of the assignment, or completely failed to address the project requirements.