Customer Requirements Preparation

Goals

- Apply various tools and agile principles utilizing concepts (user stories, behavior-driven development) to build quality software.
- Prepare to gather, critically analyze and evaluate qualitative information to produce a SaaS app.

Effort: Individual CS3300 Academic Integrity
Points: 20 pts
Deliverables: As separate Word or pdf files (NOT ZIP FILE) include the following documents

- Document containing answers to questions
- Interview Worksheet

Description

Explore how to do qualitative research to define a problem statement and gather information from stakeholders to develop requirements and acceptance criteria using Agile approaches.

Base Requirements for App Project

You will come up with an idea of an app but **your app should have at least two different stakeholders and must**

1. One stakeholder: Requires a login with a username and password to
    - Create a new item
    - Edit item
    - Remove item
    - Think about what the first item is that will be created.
2. Another stakeholder: Should be able to view items without logging in
    - Display condensed list of items created on one page
    - Display detailed information about each item individually
    - Should have at least three attributes associated with the item. **For example** in portfolio user can add a project that includes required title and description and optional image

3. **You can not make your project the portfolio app.**

1 Problem Statement and Qualitative Research

You are going to make an app that solves a need and conduct qualitative research to understand the users and their needs. You should interview 2 to 3 people. You only need to interview them one time.

[What is a Problem Statement in UX? (And How to Write One)]

2.1 Describe the components of a problem statement. How do you go about defining your problem statement and understanding the needs of the users? Who should you include when gathering information? A problem statement covers the idea of understanding your client and what they need. Before any of the work, doing some research for what is needed by your client and understanding what they may be lacking is key before diving into an interview with them. This could mean doing separate research but also simple questions to the client such as “what isn’t working”. This general idea covers the idea of who is being impacted by the problem, what is the problem, where will this product get used, and why is it important to solve? Following this work with more questions about the client with answers you think of help build this baseline understanding for the problem which allows you to create an effective statement. Finally knowing this information you can build your problem statement which needs a user, a need, and a goal. This creates a statement which has needs as well as problems in mind to create a concise statement. Below I provided an example of a great problem statement that covers the idea of doing prior research and then creating the solution to the problem. "Hard-working professionals need a convenient, quick way to eat healthy food because they often work long hours and don't have time to go grocery shopping." (Entrepreneur)
2.2 Describe an idea for the app you will develop this semester. An idea for the app I wanted to create was a fitness app which allows people to log on and upload their workouts they may be doing. (Stakeholder 1) Other users could go to this website and see the workouts people uploaded. (Stakeholder 2) In general the details of the app include more in depth items such as having videos or even links to other places but the base groundwork is the general idea of being able to login and upload workouts as well as see what others are doing.
2.3 Create your worksheet to interview users to develop your problem

statement, users stories and acceptance criteria in another document.

*Check Submitted Document

2.4 How could surveys be used to gather information? Give an example of some survey questions you could ask?

Surveys are a good way of gathering a large sample of data from people for bigger ideas. An example of this would be looking at the portfolio app which helps thousands of students which you can't all interview. The survey provides the option to hear a large amount of feedback. I think good questions on a survey should be shorter but valuable. The longer the survey response the harder it is to make correlations in needs, if you're reading through 300 paragraphs of things people want it wastes the purpose of the survey. Good questions cover the idea of "Do you have a place to upload your work so job employers can see?". "Would you use a portfolio app for your school projects?". These are more basic questions but the idea of getting solid answers is so important when dealing with a large sample size. Being able to gauge the majority of what people want, need, and feel like would be valuable is useful in the app building process. It is also still important to interview some clients in person to hear the interques of what they might need.

2.5 Include at least 3 resources you used. You must have at least one resource not in the lectures.

- https://www.velvetech.com/blog/33-questions-to-challenge-mobile-developers/ *(used to look at outside of the box questions that I would not think of getting asked or asking)*
- L07 Customer Requirements: BDD, User Stories and Acceptance criteria slides *(used to understand user stories, customer requirements, and having a BDD approach)*
- https://careerfoundry.com/en/blog/ux-design/problem-statement-ux/ *(used to understand problem statements)*
- https://www.betterup.com/blog/problem-statement *(more information on problem statements)*
- https://www.entrepreneur.com/growing-a-business/3-excellent-problem-statement-examples/452703 *(used to see examples of good problem statements)*
- https://www.surveymonkey.com/mp/writing-survey-questions/ *(used to understand what makes a survey better)*

2 User Stories

User stories for our class will be

- Written in “Connextra” format
- SMART

3.1 Describe the components of a user story and summarize what user stories should and shouldn’t be. Give examples.

A user story should be written in a way that the user can easily understand what you are trying to implement.

IMPORTANT
- No technical writing
- Short ~3 sentences
- Customer oriented
- Solving 1 issue

BAD
- Longggg
- Technical writing but also way to vague

The user story when written should coder the 5 major ideas of SMART.

S- specific: what are we doing?
M- measurable: what has changed?
A- achievable: what feature are we focused on?
R- relevant: why are we doing this?
T- timeboxed: how long is it really going to take?

Good Example:

“As a manager, I want to be able to understand my colleagues' progress, so I can better report our success and failures.”

Bad Example:

“As a project manager, I want more features in the project management tool, so that it’s better.”

3.2 Explain the value in using the user stories methodology with a customer.

A user story provides a clear cut plan that is understandable and measurable for a customer. The biggest part of it all is the fact that it is measurable; a customer is focused heavily on results so being able to see issues, solutions, and a general time period creates an environment that encourages collaboration. Being able to understand what the client wants and expressing it in manageable situations is what brings the value of user stories. It also provides a framework for us as the programmer; just asking the user what they want either ends up in unrealistic expectations or “you

just make the app I don't know what I want" which leads to this "isn't what I wanted" once it is complete. In general the value of user stories is the ability for understanding between the customer and the programmer.

"After reading a user story, the team knows why they are building, what they're building, and what value it creates."

3.3 Include at least 3 resources you used. You must have at least one resource not in the lectures.

- https://www.atlassian.com/agile/project-management/user-stories *(understanding user stories along with good examples)*
- https://guides.visual-paradigm.com/crafting-effective-user-stories-a-guide-to-good-and-bad-versions/#:~:text=Good%20user%20stories%20are%20specific,inefficiencies%20in%20the%20development%20process. *(understanding user stories along with bad examples)*
- L07 Customer Requirements: BDD, User Stories and Acceptance criteria *(understanding user stories, SMART approach, along with examples)*
- https://www.atlassian.com/agile/project-management/user-stories#:~:text=After%20reading%20a%20user%20story,and%20a%20better%20product%20overall. *(used to better understand what is so beneficial about user stories for a customer)*

3 BDD Scenarios

For this class we will follow

- User story should include "happy" and "sad" scenarios.
- Scenarios should have 3 to 8 steps written in the BDD form describing the process
    - Scenario: Title of the condition
    - Given: Represent state of world before event (preconditions)
    - When: Something happens
    - Then: This is the result (postcondition)
    - Optionally And, But

4.1 Explain Behavior Driven Development methodology and include the benefits.

BDD is a focus on how we want the app to work compared to implementation of the app. This includes multiple discussions with the customer about what they like and what they want to see. This helps prevent them getting the final product and being like "this isn't what I

asked for". It focuses on what the customer is seeing and how they would be interacting with the app. The idea of seeing the "happy path" and "sad path" of what might be occurring when interacting with the app. A couple benefits discussed to be involved with BDD methodology is simple language, focus, using scenarios, and efficiency. Scenarios are layed out which provides a base framework for everyone to work off of which eventually leads to a smooth transition into the automated test stage.

4.2 Describe the components of BDD scenarios and include examples.

Like discussed above, the idea of BDD scenarios covers the idea of how a user would interact with the application. The happy path and the sad path are two ideas which discuss the idea of something going right and when something might be going wrong with the website. The idea is to see how the user interacts and what the appropriate response would be from the system. Compared to the earlier TDD which focuses on implementation and then seeing what goes wrong this method allows more thought to go into the making of the app. The scenarios allow the user and programmer to discuss possible issues and what the solutions wanted would be.

Happy Path:

- User can log into their account with their username and password
- I login into the website and see the welcome page with all my saved work still being there

Sad Path:

- I try to log in and it says I got the wrong username and password
- When I go over to delete my work it is not there

*the sad path is not always user error but more just a general scenario of what could go wrong while working with the app

4.3 Include at least 3 resources you used. You must have at least one resource not in the lectures.

- https://www.agilealliance.org/glossary/bdd/ *(get an understanding of BDD)*
- https://www.departmentofproduct.com/blog/writing-bdd-test-scenarios/ *(used to better understand the difference between TDD and BDD)*
- L07 Customer Requirements: BDD, User Stories and Acceptance criteria *(used to understand the BDD and see scenarios)*
- https://www.biteinteractive.com/bdd-toolbox-happy-pathsad-path/ *(used to better understand the perks of having a sad and happy path approach to the development process)*

4 Lo-Fi UI Mockup

5.1 Explain the benefits in using Lo-Fi Mockups of the web page with the customer.

A Lo-Fi mockup allows you to sketch out the general idea of how everything would work together. When you click your username you get brought to a profile page and other general cause and effects while working in the app.

Benefits:

- Don't waste time on an elaborate plan or design
- Less technical making it easier to look at
- Focused more on how the thing will work rather than how pretty the presentation is
- Leaves a middle ground to discuss features that were good and allow for change
- Can quickly implement changes into the sketch (they want a resource page for example, you would just add another box)

5.2 Describe what the Lo-Fi storyboard should contain.

It should be a general sketch with how everything will work together, below is a list of what would be included in a good lo-fi mockup.

- Various pages that would be interacted with
- Forms and buttons
- Pop ups
- Arrows showing where one thing would lead into another
- General movement of the application between various items clicked

*IT IS A SKETCH!!!

5.3 Include at least 1 resource showing a lo-fi mockup for an app.

- https://www.lucidchart.com/blog/value-of-low-fidelity-mockups *(has examples of mobile apps working but also provides info on lo-fi mockup benefits)*
- L08 Lo-Fi UI Sketches and Storyboards *(provides examples of mockups but also explains the benefits)*
- https://bootcamp.uxdesign.cc/creating-wireframes-and-low-fidelity-prototypes-in-figma-for-the-google-ux-course-326a7e11fa60 *(provides a great example of a mockup for a mobile app)*