

CS 398: Application Development

Week 02 Lecture: Requirements

Interviews; Personas; Affinity diagrams; User stories

Team Meetings

- Team organization
 - Team of two people that posted on Piazza — <https://piazza.com/class/kxkw2lud9cf5ah?cid=41>
 - Please let me know if your team changes! No changes after this week please!
- Lectures
- Everyone is expected to attend ALL lectures - general lectures plus team breakouts.
 - Stay until the end of the class Mon/Wed, for the first 50 mins Fri.
 - You are welcome to stay longer if you want! (Jeff will often stick around longer when possible).
 - Please take meeting minutes.
 - 5% of your grade is based on showing up!
- We'll start check-ins this Friday.

This Week

Wed: Software Requirements

- Before Class: [requirements video](#), [slides](#) (notes)
- Lecture: slides
- Activities: Determine software requirements.
 1. Review the project specification.
 2. Brainstorm with your team.
 3. Interview users; generate personas.
- Resources:
 - Interaction Design Foundation. 2021. <https://www.interaction-design.org/ux-fundamentals/requirements-engineering/>
 - Nielsen Norman Group. 2018. **Affinity Diagrams and Design Ideas.** <https://www.nngroup.com/articles/affinity-diagrams/>
 - Patrick Thornton. 2019. **How to conduct user interviews.** <https://uxdesign.cc/how-to-conduct-user-interviews-fe4b8c34b0b7>

<https://student.cs.uwaterloo.ca/~cs398/01-syllabus/1-weekly-schedule/#week-2-planning--requirements>

Fri: Kotlin Part 1

- Before Class: [kotlin 1 video](#), [slides](#) (notes)
- Lectures: slides
- Activities: Continue requirements analysis.
 1. Interview users; generate personas (cont'd).
 2. Create an affinity diagram to isolate requirements.
 3. Document user stories.
- Resources:
 - JetBrains. **IntelliJ IDEA.** <https://www.jetbrains.com/idea>
 - JetBrains. **Kotlin Documentation & Tutorials.** <https://kotlinlang.org>
 - Steve Yegge. 2017. **“Why Kotlin is better than whatever dumb language you’re using”.** <http://steve-yegge.blogspot.com/2017/05/why-kotlin-is-better-than-whatever-dumb.html>

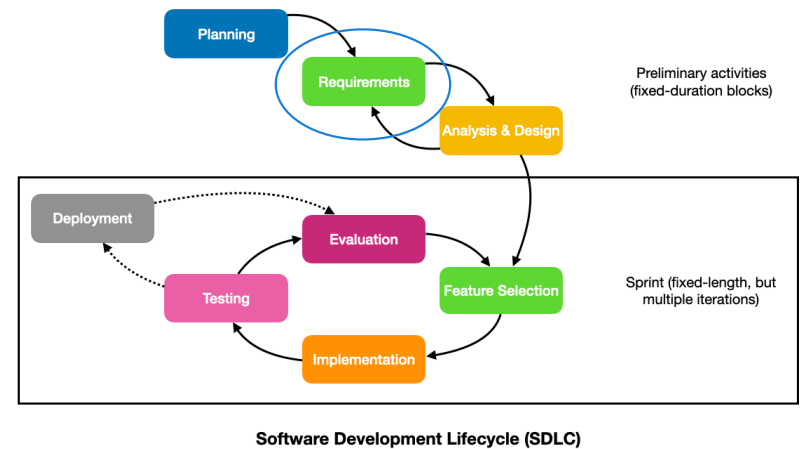
Requirements Review

Phase 2: Requirements

The requirements stage is where we explore the problem that we wish to address, and determine what to build. The output of this phase is a detailed list of product features that you would implement.

This focus on users-first is a critical tenet of [User-Experience Design \(UXD\)](#).

“User Experience Design (UXD or UED) is **a design process whose sole objective is to design a system that offers a great experience to its users**. Thus UXD embraces the theories of a number of disciplines such as user interface design, usability, accessibility, information architecture, and Human Computer Interaction.” — www.usertesting.com




Step 1: Determine Users

Your first step is to determine which users you want to target.

Pick a ROLE or CATEGORY of user e.g. Software Developer, Students, Teacher, Lawyer, Doctor, Engineer, Parent, ...

- If you are CERTAIN this this will be a good match, you can immediately start targeting that particular set of users.
- If you are NOT CERTAIN about the fit for your product, you can broadly interview users (across a bunch of different roles) to determine if they might be good candidates.



Look at competing products and brainstorm - you might identify a great fit right away!


Step 2: Interviews

Locate people that match your target audience (“potential users”).

Interview 3-5 of these people [Nielsen 2000].

Goal is to define

- Who are my users?
- What is important to them?
- What are they trying to accomplish?
- What features will be important to my users?
- Do these features work together?



Course staff can be interviewed! We represent **students, instructors, developers, (gamers, musicians...)**

Do NOT interview one another.

Step 2: Interview Questions

- Do you use an application to take notes? If so, what do you use?
 - What do you like about it? Are there things that you dislike about it?
 - If there is one thing that you could change with this application, what would it be?
 - What would it take for you to consider moving from your current application to a new application?
- If you've never used an application like this: is there a particular reason?
 - How do you store information like X currently?
- If this application could do X would that be interesting to you?
- Do you use other similar applications for (fill-in-idea-here)?
 - Have you tried doing X with a note-taking application?

**Be careful with
hypotheticals b/c users
can get hung up on them**

Step 3: Generate Personas


One issue with trying to understand users is that we always want to design from our own perspective. You don't want that - instead you want to be able to empathize with your users, and anticipate what *they* would want.

A **persona** is an archetypal description of a the user of your product. Creating personas can help you step out of yourself... [I]t can help you to identify with the user you're designing for.

Create one or more personas for your user roles. (Consider if you have different types of users in the same role? Maybe need multiple personas).

Jeff D. Instructor

BIO
Jeff is a faculty member in the Cheriton School of Computer Science. Previously, he worked in industry as a software developer, and engineering manager. He is passionate about teaching, and will evaluate any tool through the lens of "how it might make him a better teacher".



GOALS
Find better ways to organize information. Improve efficiency.

FRUSTRATIONS
Poorly designed software.
Hates vendor lock-in. Ironically, Apple user.

MOTIVATIONS

| | | |
|--------------|-----|------|
| Incentive: | Low | High |
| Fear: | Low | High |
| Achievement: | Low | High |
| Growth: | Low | High |
| Power: | Low | High |
| Social: | Low | High |

QUOTES
"Using your computer should be pleasant, even *joyful*."
"Build something that will make you proud. If you wouldn't show it off to your parents, rewind and try again."

| | |
|--|--|
| DOMAIN AWARENESS Self-proclaimed "productivity geek". Has strong opinions in this domain. | TECH KNOWLEDGE Expert in software development. Broad general knowledge. |
|--|--|

Age: 50
Occupation: Lecturer
Status: Married
Location: Waterloo
Archetype: Educator
Personality: Friendly; Helpful.

Step 4: Affinity Diagrams

The outcome of your interviews will likely be a lot of qualitative data -- opinions, suggestions and ideas that the user expressed. The challenge is extracting ideas, themes from this narrative that you can use in your design.

Use an **affinity diagram** to expose critical ideas or themes from the user's ideas. (You can even do this across multiple users and look for which themes occur more frequently!).

To create an affinity diagram:

1. Record all notes or observations on individual cards or sticky notes
2. Look for patterns in notes or observations that are related and group them
3. Create a group for each pattern or theme
4. Give each theme or group a name
5. Create a statement of what you learned about each group (provide your analysis or key insight)



Affinity Diagram created in InVision.

<https://www.invisionapp.com/freehand/templates/detail/affinity-diagram-template>

User-Story Map: Mobile App Feature for Depositing Checks

1. Activities:

High-level tasks users can do in the digital product



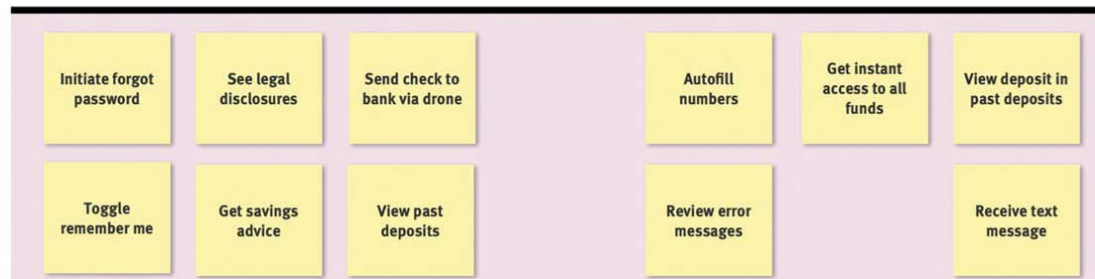
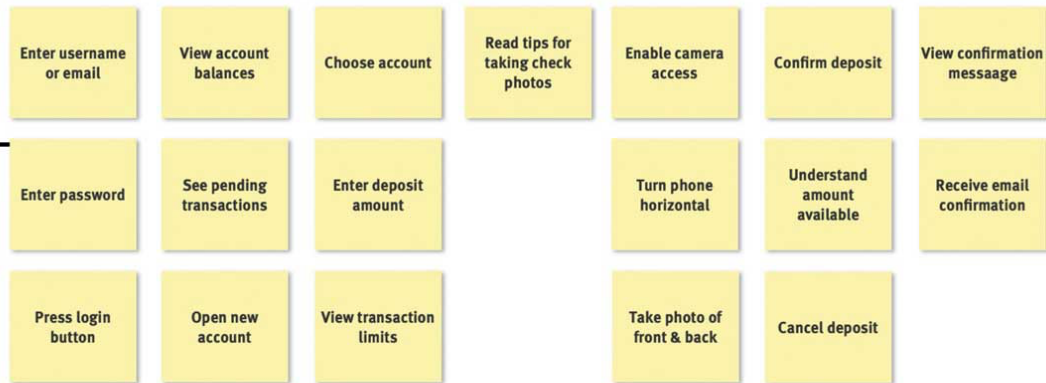
2. Steps:

Steps users go through to complete the activity above



3. Details:

Granular, discrete interactions to complete the step above



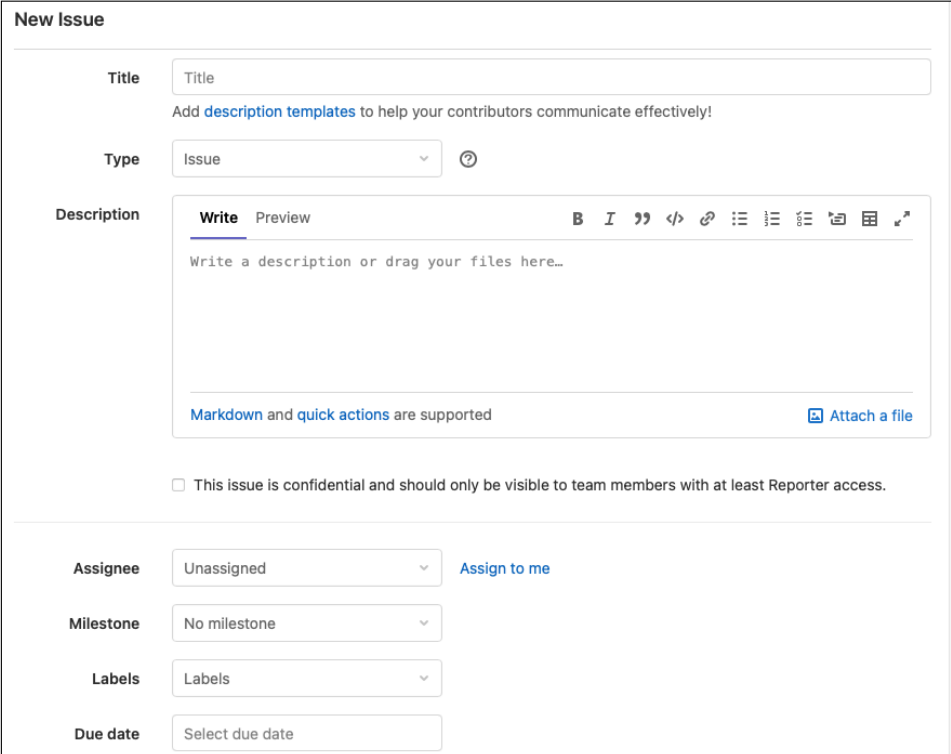
<https://www.nngroup.com/articles/user-story-mapping/>

Step 5: Product Backlog

Your final step should be to convert these requirements into actionable items.

- Log each item into your GitLab project as an Issue.
- Don't assign anything yet!
 - Assignee: Unassigned
 - Milestone: No milestone
 - Due date: Unselected

These unassigned issues constitute your Product Backlog. You only schedule this during Sprint planning, and only if the team agrees to take it on.



The screenshot shows the 'New Issue' form in GitLab. It includes a 'Title' field with the placeholder 'Title', a 'Type' dropdown set to 'Issue', and a 'Description' field with a rich text editor. The description field has a 'Write' tab selected and a 'Preview' tab. The rich text editor includes a toolbar with icons for bold, italic, quote, code, link, list, ordered list, table, and image. Below the description field, there is a checkbox for 'This issue is confidential and should only be visible to team members with at least Reporter access.' At the bottom, there are dropdown menus for 'Assignee' (set to 'Unassigned'), 'Milestone' (set to 'No milestone'), 'Labels' (set to 'Labels'), and 'Due date' (set to 'Select due date'). There is also an 'Assign to me' link next to the Assignee dropdown.

Activities

Activities for Today

Step 1: Pick users

- Decide which users you want to target. “Who are your users?”
 - e.g. Students, Developers, Teachers, (Other?) — pick what appeals to you

Step 2: Interview people that fall into your role

- Find 3-5 users that fit your target demographic. — **this can include course staff.**

Step 3: Create persona(s) — not today

- Can be different types of users, or more nuanced - e.g. front-end developer, mobile developer.

Step 4: Affinity Diagrams — not today

Step 5: Product backlog — not today

How to Setup a Team Call

Here's how to move from the public lecture (LEC 001 Morning or LEC 002 Afternoon) to your Team channel:

1. Click on **Leave** to exit the current meeting.
2. In MS Teams, select your Team's channel on the left-hand side.
3. If you are the first person, click on the **Meet** button. If there is a meeting in progress, click on the **Join** button.
4. You should now be in a team meeting! To quit the meeting, click **Leave**.

If you want to ask the instructor or TAs for help:

1. Do NOT leave your meeting!
2. Click on the public channel. You should see the chat window.
3. Ask for help and include your Team number. A team member will join your Team channel.
4. Click on your Team window to rejoin that call.

LEC 001 Morning channel

> [@morning-staff](#) can we meet in Team 101?

LEC 002 Afternoon channel

> [@afternoon-staff](#) can we meet in Team 201?