Creating Effective User Stories

CS 485/698: AI-Assisted SE

Today’s Agenda

- Turn user discovery into user stories, integrate needs and feedback into user stories, develop a storyboard to communicate ideas
- Mobbing session
- Team meeting
  - Finalize project “one-line descriptions”
  - Setup team infrastructure + invite course staff

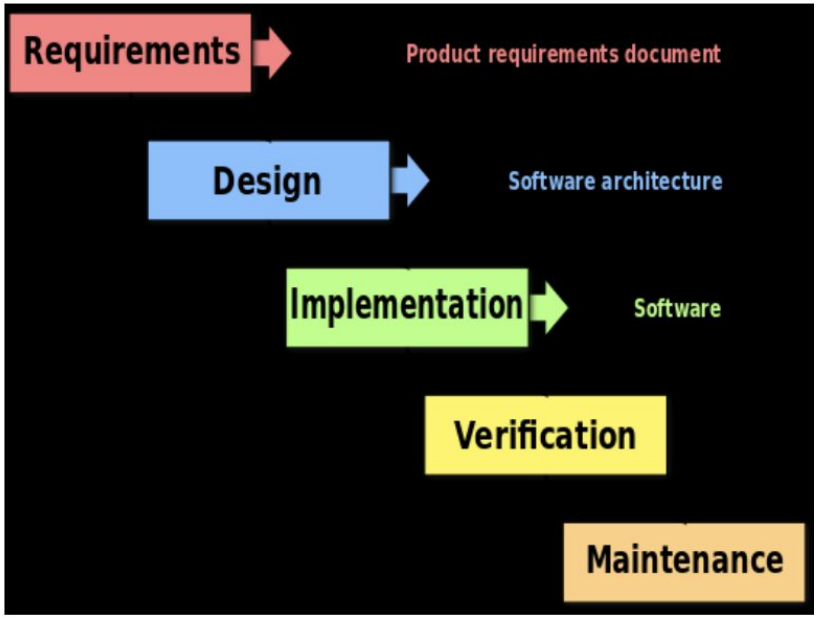
Today’s Agenda

- Turn user discovery into user stories, integrate needs and feedback into user stories, develop a storyboard to communicate ideas
- Mobbing session
- Team meeting
  - Finalize project “one-line descriptions”
  - Setup team infrastructure + invite course staff

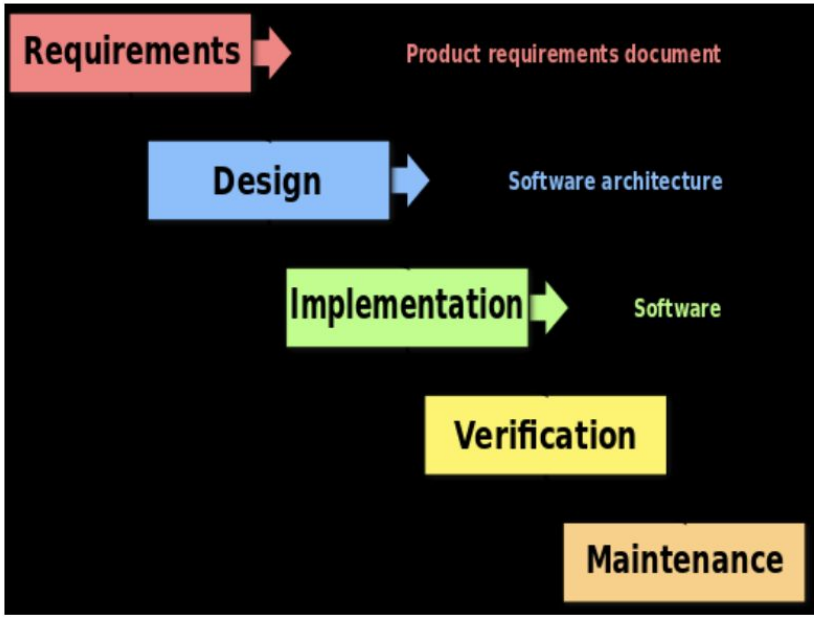
Don’t forget to sign up for A2 reflection topics!

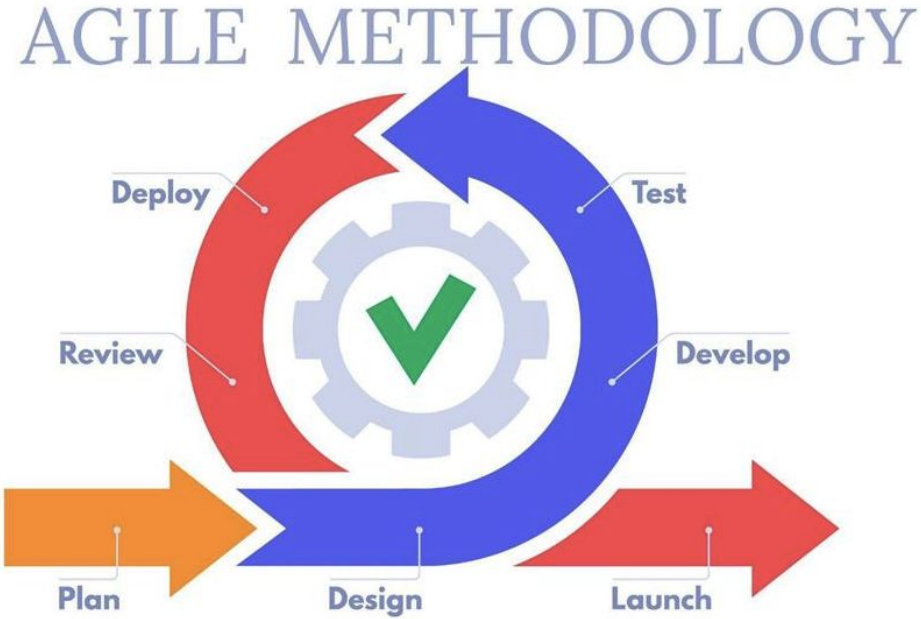
A Quick Overview of Software Process

A Quick Overview of Software Process



A Quick Overview of Software Process





Our Process: Scrum Meetings w/ LLManager

We'll have a short team meeting at the beginning of each class from now on. There are three kinds:

Our Process: Scrum Meetings w/ LLManager

We'll have a short team meeting at the beginning of each class from now on. There are three kinds:

- **Sprint Planning Meeting** (1st Monday of each unit)
    - LLM decides what to tackle for that sprint and team reviews

Our Process: Scrum Meetings w/ LLManager

We'll have a short team meeting at the beginning of each class from now on. There are three kinds:

- **Sprint Planning Meeting** (1st Monday of each unit)
  - LLM decides what to tackle for that sprint and team reviews
- **Daily Scrum Meeting** (2nd Monday/1st Wednesday)
  - Quick meeting to touch base on:
    - What have I done? What am I doing next? What am I stuck on/need help?

Our Process: Scrum Meetings w/ LLManager

We'll have a short team meeting at the beginning of each class from now on. There are three kinds:

- **Sprint Planning Meeting** (1st Monday of each unit)
  - LLM decides what to tackle for that sprint and team reviews
- **Daily Scrum Meeting** (2nd Monday/1st Wednesday)
  - Quick meeting to touch base on:
    - What have I done? What am I doing next? What am I stuck on/need help?
- **Sprint Retrospective** (2nd Wednesday)
  - LLM reviews sprint process and team discusses

Backlogs

- The ***product backlog*** is a list of all the features for the product that haven’t yet been completed

Backlogs

- The ***product backlog*** is a list of all the features for the product that haven't yet been completed
- The ***sprint backlog*** is all the features that will be worked on for that sprint. These should be broken down into discrete tasks:
  - Fine-grained
  - Estimated
  - Assigned to individual team members
  - Acceptance criteria should be defined

Backlogs: Kanban

https://en.wikipedia.org/wiki/Kanban_(development)

Backlogs

- The ***product backlog*** is a list of all the features for the product that haven't yet been completed
- The ***sprint backlog*** is all the features that will be worked on for that sprint. These should be broken down into discrete tasks:
  - Fine-grained
  - Estimated
  - Assigned to individual team members
  - Acceptance criteria should be defined
- ***User Stories*** are often used to express tasks in the backlog
  - Each user story is a unit of customer-visible functionality

User stories

“As a <**role**>, I can <**capability**>, so that I can <**receive benefit**>.”

User stories

“As a <**role**>, I can <**capability**>, so that I can <**receive benefit**>.”

- also requires a ***condition of satisfaction***, which is the measurement you will use to decide if the user story has been completed

User stories: examples

“As a <**role**>, I can <**capability**>, so that I can <**receive benefit**>.”

- “As a **computer user**, I want to **backup my entire hard drive** so that **my files are safe**”

User stories: examples

“As a <**role**>, I can <**capability**>, so that I can <**receive benefit**>.”

- “As a **computer user**, I want to **backup my entire hard drive** so that **my files are safe**”
- “As a **typical computer user**, I want to **specify folders to backup**, so that **my most important files are safe**”

User stories: examples

“As a <**role**>, I can <**capability**>, so that I can <**receive benefit**>.”

- “As a **computer user**, I want to **backup my entire hard drive** so that **my files are safe**”
- “As a **typical computer user**, I want to **specify folders to backup**, so that **my most important files are safe**”
- “As a **power user**, I want to **specify subfolders and filetypes NOT to backup**, so that **my backup doesn’t fill up with things that I don’t need to preserve**”

Mobbing Activity: How It Works

- We will randomly select a “mobber” and a “navigator” each time that we do a mobbing activity
  - Let’s do it now: link to randomizer
- ***Mobber***: actually operates the LLM in front of the class
- ***Navigator***: takes notes in the `#mobbing` channel on Discord
  - Mobber and navigator will switch halfway through
- Everyone else: give suggestions to the mobber about what to do
  - Up to the mobber to decide who to listen to
- You’ll be required to do this at most once this semester

Mobbing Activity

- Let's get an LLM to make user stories for *Zoom w/ better muting*.
- Everyone take your user discovery transcripts from Monday's class and copy their text into the **`#mobbing`** channel on Discord.
  - Mobber will come to the hot seat.
  - Notetaker sits in the front of the class with their laptop to take notes in the **`#mobbing`** channel.
- Now, everyone tell the mobber what to say to the model to get it to create 10 user stories. Be sure it incorporates what you all learned from user discovery

How do we evaluate a user story?

How do we evaluate a user story?

Remember “**INVEST**”

User stories should be:

Independent

- = “can be scheduled in any order”
- This isn’t always possible, but dependencies should be:
  - minimized
  - clearly labeled when actually necessary

Negotiable

- A good story captures the *essence*, not the details
  - Details can be negotiated during development
- Goal: avoid overly-constraining the implementation
  - e.g., “the light blinks regularly” rather than “the light blinks every two seconds”

Valuable

- This story needs to have value to someone (hopefully the customer!)
  - It's easy to forget why you are doing what you are doing
- Make sure that each story's ***benefit*** (i.e., 3rd part of the story) clearly expresses its value to the user

Estimable

- Each story should provide enough details to estimate the amount of effort needed to implement it
  - Helps keep the size small
- Best practice: think about how you'd implement the user story as you write it
  - Break down the story if you're not sure how you'd approach it

Small

- Traditional definition: story fits on 3” x 5” index card
- Modern definition: at most two person-weeks of work (one sprint)
- Too big => unable to estimate

Testable

- Each story must have a clear criterion for when we can mark task “Done”
  - Ensures understanding of task
  - Unable to test == do not understand well enough, need to think more
- Ideally, using some measure that can be objectively tested
  - Avoid: “50% of the code is done”
  - OK: “Pressing the button makes *something* happen”
    - (even if you don’t specify what)

Mobbing Activity

- Ask the LLM to evaluate the user stories we generated using these criteria
  - One criterion at a time
  - Think critically: is the LLM correct?
- Ask the LLM to improve the user stories
  - Does it work?
- Ask the LLM to regenerate the user stories from scratch, with the criteria in the prompt
  - Does that work better?

Storyboards

- Next activity: we will use Figma Make to create a storyboard for one of these user stories
- A ***storyboard*** is a sketch of the user's journal through a specific workflow.
  - Shows user navigation and actions
  - Can be used for usability testing
  - Can show to potential customers to get feedback

Storyboards: Mobbing Activity

- Let's create a flowchart for the user's journey using Figma's AI
  - Think about our user's background. How much context should we give the model?
- First, swap mobber and notetaker roles.
  - Notetaker will again take notes in the `#mobbing` channel
- Everyone, tell the mobber how to make Figma's AI make a flowchart for one of the "Zoom w/ better muting" user stories.

Storyboards: Personas, UX, etc

- The user story on its own probably doesn't have enough detail to make a good storyboard. We might also want:
  - a user persona: who is she? What job does she have? What are her goals? What challenges does she face?
    - consider premade personas, e.g., https://gendermag.org/
  - UX: we need to see how the app should look to understand how we want it to work. Did Figma's AI do this well? Why or why not?
- Discussion: what went well/poorly? What lessons did we learn about getting Figma to behave?

Project: Development Hygiene Expectations

- Create issues for feature improvements or bug fixes
  - Do work on feature/bug branches, not on `main`
- When creating issue, assign team members and tag with appropriate labels
- When you create pull requests, reference the issue it resolves
- Provide feedback on pull requests (code review)
- Use the Kanban board to track your workflow
- Whenever checking in code, we suggest always checking in your LLM logs too. That's the real code in this class, right?

Project: Development Hy

Overall: act like a professional software engineer

- Create issues for feature improvements or bug fixes
  - Do work on feature/bug branches, not on `main`
- When creating issue, assign team members and tag with appropriate labels
- When you create pull requests, reference the issue it resolves
- Provide feedback on pull requests (code review)
- Use the Kanban board to track your workflow
- Whenever checking in code, we suggest always checking in your LLM logs too. That's the real code in this class, right?

Rest of class: Team meetings

Don't forget: P1 due on Sunday

- What's the one-sentence description of your project?
  - e.g., "Zoom with better muting behavior"
- Communication:
  - agree with your team on a primary communication channel
  - invite instructors: any communication that we can't see "didn't happen" (for grading purposes)
- GitHub:
  - set up a repo for your code now (private ok)
  - invite your team members + instructors as collaborators
    - instructor GH usernames: `kelloggm`, `Chun-Jie-Chong`