SE463
Software Requirements Specification & Analysis

Prototyping

Readings:
  Chapter 15: “Risk reduction through prototyping”

Prototyping

“A software prototype is a partial implementation constructed primarily to enable customers, users, or developers to learn more about a problem or its solution.”

Alan Davis

- Presentation Prototypes
  - used for proof of concept; explaining design features; etc.
  - explain, demonstrate and inform – then throw away

- Exploratory Prototypes
  - used to determine problems, elicit needs, clarify goals, compare design options
  - informal, unstructured and thrown away.

- Breadboards or Experimental Prototypes
  - explore technical feasibility; test suitability of a technology
  - typically no user/customer involvement

- Evolutionary (e.g. “operational prototypes”, “pilot systems”):
  - development seen as continuous process of adapting the system
  - “prototype” is an early deliverable, to be continually improved.
Throwaway Prototyping

Purpose:
• learn more about the problem
• discard after desired knowledge is gained

Use:
• early or late

Approach:
• horizontal - build only one layer (e.g. UI)
• “quick and dirty”

Advantages:
• learning medium for better convergence
• early feedback → less cost
• successful even if it fails!

Disadvantages:
• often replaces proper documentation of the specification
• may set customers’ expectations too high

Evolutionary Prototyping

Purpose:
• learn more about the solution
• reduce risk by building parts early

Use:
• incremental; evolutionary

Approach:
• vertical – partial implementation of all layers
• designed to be extended and adapted

Advantages:
• requirements are not frozen
• return to last version if problem encountered

Disadvantages:
• early architectural choice may be poor
• can end up with complex, unstructured system that is difficult to maintain
• optimal solutions not guaranteed
• lacks control and direction
SPRINT kickoff slides
https://www.slideshare.net/bikolabs/sprint-kickoff-slides-sesin-de-sprint-week
Sketching, Wireframing, Prototypes

Prototype as a façade of a solution, used to bait stakeholders into providing new requirements details

http://murdochcarpenter.com/portfolio/wireframes/
Sketches

Sketches of the UI focus on content: the scope of what to include on a screen, and how to lay it out.

http://murdochcarpenter.com/portfolio/wireframes/
Wireframing

A wireframe is a visual representation or mock-up of a user interface, using only simple shapes

- focuses on content, information hierarchy and flow
- defers details about presentation (colour, fonts, images)
Sketching vs. Low-Fid Wireframing
F-Layout Design

http://webdesign.tutsplus.com/articles/understanding-the-f-layout-in-web-design--webdesign-687

The heatmap shows where users tend to focus their gaze, reading left to right, and top to bottom (but less to the right the lower they gaze)
Crazy 8's

-youTravel search

- Get people to enough info and "feel" of a trip to decide to commit + get more info.
- Trips are visual: maps, photos, places, events.
- Time may matter: (winter = ski, fall = scenic drives)
- Sometimes location is already decided on (ground can you?) + sometimes not so much (the beach! a get-away!)
- What about $44?

Note: this is a completely fake project!
Sketchboard

User Flow, Storyboard
Rockit Chat List Screen [9]

Chat List Screen (Homescreen) [9]

- Settings button (button): sends user to Settings screen (screen 13)
- Specific module list items (list item buttons): sends user to chat screen for the specific module (screen 12 for example module screen). Each module sends a formatted query to the server.

You should have a way to order the modules. EX: By alphabetic order? By recently used? By favourite?

Might be a really long screen depending on the amount of modules the service provide
Summary

• Sprint prototypes enable fast learning about some aspect of a proposed solution

• Prototypes generally used to elicit requirements in the form of feedback on the value of proposed solutions.

• Sketching enables rapid generation of alternative UI designs
  • Lots of ideas to work from
  • Multiple iterations and refinements
  • Active input from all stakeholders
IMPORTANT!!!!!!!

Each team has to provide the contact of the five interviewees in Deliverable #4. I may contact some of them to be sure they are real early adopters 😊
Deliverable #5
Tuesday, June 25 at noon

• **(40%) Elicited Requirements**
  Using a variety of elicitation techniques, especially those that lead to innovative requirements, identify at least 2+ new innovative features or use cases for your project.
  • Generating ideas should involve concrete stakeholders outside of your team
  • Evaluating the business value of the new ideas should involve the project’s customer or champion stakeholder

• **(40%) Annotated Screen Sketches**
  • 2N sketches of screens (or pages, or user interfaces)
  • Annotations explaining GUI widgets on the screens
  • Stakeholder feedback on the screens

• **(20%) Screen Alternatives**
  • For 2 screens, sketch 6 alternative layouts
  • Team feedback (positive and negative) on the screens