Lecture 12: Presentations, user experience

Lecture goals
- Give effective presentations to stakeholders
- Solicit feedback on user interface designs

Presentations

Presentations in software engineering
- Important in every project phase
  - Marketing to potential clients
  - Reporting progress to senior management
  - Reports and demonstrations to clients
  - Communication with colleagues on dev team
- Important for career growth
  - Unlikely to achieve leadership position if you cannot give decent presentations
- Not everyone is born a great presenter, but everybody can be well-prepared
  - If you are uncomfortable, take every opportunity to gain experience

Presentations in CS 5150
- Two required presentations:
  - Progress update during 3rd session
  - Preliminary delivery during 5th session
- Every team member must present a portion of one presentation
  - Less experienced presenters will be more comfortable presenting things they personally worked on
- Audience: your client
  - May not be technical (internal projects: client is manager, not developer)
- Course staff will evaluate presentation contents and technique

Planning for presentations
- Know your purpose, audience, and resources
- What is the presentation meant to achieve?
  - Confirm understanding?
  - Obtain client approval?
  - Propose new feature?
  - Solicit feedback on prototype?
  - Build excitement/buy-in?
  - Request assistance?
  - Report progress?
Train users?

- Who must attend the presentation for it to achieve its purpose?
  - Prospective clients?
  - Project management?
  - System users?
  - Other developers?

How is your presentation constrained?

- Time available
- Projector/screen sharing?
- Internet access?

Time management

- CS 5150: 45 min for presentation, 15 min for questions
  - Expect interruptions (presentation must serve the audience; is not an end in itself)
- Have an agenda that fulfils the presentation’s purpose
- Rehearse your presentation on the clock!

Remote presentations

- Good audio is essential
  - Make a practice recording with all presenters in their anticipated locations/positions
- Good video needs good lighting
- Client must be able to see all demonstrations and visual aids
  - Screen share
  - Whiteboard/annotations
  - Auxiliary camera
- Beware multiple computers in one room
- In-person presentations preferred (but rooms are hard to book)

Topics

- Topics on agenda should serve purpose of meeting
  - Description of what you have agreed to deliver to your client (shared definition of success)
  - Summary of progress since last presentation/report
  - Unexpected events and risks
  - Overview of remaining plan to complete and deliver project
  - Test plan and test cases
  - Results of user testing
  - Technical hurdles (if client is technical)
- Demonstrations are always welcome
CS 5150 topics

• Early-stage topics
  o Confirm agreement on scope and goals
    “The project will be a success if ...”
  o Progress to date
    “This is our understanding of your requirements...”
  o Mock-ups, prototypes, designs, etc.
  o Schedule and plan
    “The main risks are...”
  o What has changed since feasibility study?

• Mid-stage topics
  o Demonstration of operational prototype or delivered features
  o Results of user studies

Visual aids

• Slides
  o Common, but not required (and can be a liability)
  o Keep things simple (purpose is conveying information, not entertainment)
  o Must be legible
    ▪ Audience may have poor eyesight, projectors are lower resolution, screens are farther away
    ▪ Large fonts (including in figures!) – 20pt minimum
    ▪ Dark text on light background
  o Use to facilitate presentation, not as a reference source
    ▪ Slides are not controlled documents. Lack version control, hyperlinking

• Handouts
  o Can accommodate more simultaneous detail than a slide
  o Beware potential for distraction
    ▪ Distribute handouts ahead of time, or after meeting, or else be explicit about when they should be referenced

Preparations

• Must have a rehearsal
  o Include all demos and visual aids; don’t skip anything
  o Use same laptops as you plan to use later
  o Any unrehearsed changes are a risk – minimize them
  o Time each section

• Plan presenter coordination
  o Option 1: Moderator calls on each presenter
  o Option 2: Each presenter introduces the next
• Test equipment in location if possible
  o Projector connection, network connection, power availability

Presentation behavior
• Presenter (1) should stand; others should sit
• Appoint a recorder
• Briefly introduce each team member
• When asked a question,
  o If presenter knows answer, answer it
  o Presenter may ask another team member to respond
  o Okay to make note and reply later
• Never interrupt your colleagues
  o If you have information to add, raise your hand, allow presenter to decide if/when to call on you

Demonstrations
• Require preparation and practice to be successful
• Technical preparations:
  o Load and configure all software before presentation. Test it, then change nothing
  o If you need test data or accounts, create them in advance
  o If complex commands must be typed, create a cheat sheet or shell script. Ensure they work verbatim
• Prepare a script
  o Include setup, list of examples, task assignments, and cleanup
• Tell audience what they are seeing
  o Production-ready code? Mock-up? Proof-of-concept?

Presentation tips
• Not a lecture!
  o Also not an advertisement
• You are not the audience
  o Try to imagine the client’s perspective
• Not an end in itself
  o Be able to articulate its purpose
• Not a controlled document
  o Should not serve as primary documentation
• Not about showing off
  o Don’t mislead audience or overpromise
• Explain purpose of topics, figures
  o Why should the audience pay attention to this?
Looking ahead: CS 5150 final presentation

- **Goals**
  - Personal & team satisfaction from handing over good work to client
  - Complete course in good style with good grade
  - Clean handover without loose ends
  - A good basis for future involvement with client, team, or project

- **Audience interests**
  - Client: has invested effort in this project
    - Is it ready for production?
    - Should they invest more to deploy/maintain it?
    - Should this approach be abandoned?
  - Course staff
    - What has been accomplished?
    - What has been learned?
    - Is the client satisfied?
    - Are you handing over a maintainable system?

Final presentation components

- **Demonstration of operational system**
  - Walk through scenario
  - Be honest about gaps, weaknesses

- **Presentation**
  - Brief review of context, goals
  - Honest summary of achievements and misses
  - Summary of what is being delivered

- **Time for discussion**
- **Must fit within 45 min**
  - Cannot walk through everything

User experience

**Overview**

- A system is only as good as the interface it provides to users
  - Symptoms of poor usability:
    - Failure to attract, retain market share (users give up in disgust)
    - Users fail to find or misinterpret important information
    - System can be operated in an unsafe manner (example: 737 MAX)
  - Usability aspects that improve system effectiveness:
    - Appropriate functionality
    - Easy navigation
    - Fast response times
Elegant, organized design
- Supporting users is more than a cosmetic flourish
  - Developing good UI takes skill and time

Terminology
- User Interface (UI)
  - Look and behavior of system’s controls
- User experience (UX)
  - All factors that contribute to usability of computing system
  - Encompass entire usage lifecycle, from discovery to accomplishing goals
  - Focus on user satisfaction
- Human-Computer Interaction (HCI)
  - Academic discipline studying how people interact with computers
  - Many courses and research programs in Information Science and Communications departments

Usability requirements and evaluation tools
- Client’s opinions
- Competitive analysis
- Expert opinion
- Focus groups
- Observing users
- Measurements

Focus group
- Group interview – helps generate ideas that would not have occurred individually
- Participants: 5-12 potential users with similar viewpoints
- Interviewer
  - Ask a structured set of questions
  - Encourage group discussions
  - May show mock-ups
  - Summarize conclusions
- Recorder takes notes
- Repeat with contrasting user groups

Internal project users
- Internal projects are code review tools; users are software developers
- Your classmates are candidate users!
  - Not including your team members
  - Not including teams working on the same feature
- Recruit classmates for focus groups, user testing
Can coordinate on Ed Discussion
• May be easiest to pair teams
• Documentation of user studies will be expected in future report

Accessibility
• Users have varying ability to interact with computer interfaces
  o Color blindness (1/12 men, 1/200 women)
  o Poor or no vision
  o Lack of hearing
  o Poor manual dexterity
  o Limited language skills, domain vocabulary
  o Sensitivity to flashing light, motion sickness
• Accessibility requirements constrain the user interface
  o Many systems have a legal requirement to support users with disabilities
  o Example: Compliance with Section 508 of US Rehabilitation Act
    https://www.section508.gov/
  o Some technologies may not be suitable
    ▪ Examples: Flash, untagged PDF, immature widget toolkits

Equipment requirements
• Software runs on wide variety of devices, with diverse configurations, in many environments
  o Screen size
  o Graphics performance
  o Network bandwidth, latency, stability
  o Peripheral hotplugging
• Be explicit about equipment assumptions/requirements
• Be explicit about failure handling
• Test on variety of equipment (including extremes)

Dark patterns
• Many of our experiences with UI are in a marketing context
  o Goal is to maximize engagement and manipulate user decisions
  o Being commonplace and effective in marketing goals does not make a design pattern good
    ▪ Avoid simply aping features of slick websites (even if libraries make it easy to do so)
• User-centric design
  o Interface should facilitate, not redirect, users’ objectives
• https://cacm.acm.org/magazines/2020/9/246937-dark-patterns/fulltext
Analyze/design/build/evaluate loop
Development of user interface is always iterative

Development processes
- Written requirements poor fit
  - Requirements benefit from sketches, comparison with existing systems
  - Designs should include graphical elements, benefit from prototypes
- UI must be tested with users; expect requirements and design changes
  - Schedules must include time for testing and time to make changes

UI prototypes
- Preliminary version used to iterate rapidly between requirements and design
  - Minimize polishing effort to maximize iteration speed
- Paper sketches
  - Lowest effort, so amenable to major changes
- Wireframe
  - Outline layout
- Mock-up
  - Graphic designs with detailed layout, color
- Operational prototype