Today

• Admin
• Updates on schedule and assignments for week 6-10
• Mini Quiz
• Design Recap
• Team Projects and Project Management
Schedule for Week 6-10 (Thursdays)

• Thursdays: work with instructors on your project

• Thu Nov 8+15 (W6+7)
  Work with TAs on Project Milestones and Deliverables

• Thu Nov 22 (W8)
  No class (Thanksgiving)

• Thu Nov 29 (W9)
  In class project testing

• Thu Dec 6 (W10)
  Work with TAs on Final Project Development and Report

Attendance is REQUIRED (we will mark down absent people)
Schedule for Week 6-10 (Tuesdays)

- **Tuesdays: present your work and updates**
- **Tue Nov 13 (W7)**
  Teams Project Report. CSE 118 to report on technical architecture and features
- **Tue Nov 20 (W8)**
  Time for teams to work together; Open Office Hours with TAs
- **Tue Nov 27 (W9)**
  Ethics Exercise
- **Tue Dec 64 (W10)**
  Time for teams to work together; Open Office Hours with TAs

Attendance is REQUIRED (we will mark down absent people)
Assignments Week 6-10

- **Mini Quiz**: last quiz is next week
- **Reading Summary**: last one is next week
- **Annotations**: no more annotations
- **Discussions**: no more discussions
- **Project**: no more prototipAR submissions
Assignments Week 6-10

- **Project Updates**: Weekly Agile/Scrum report on the team and individual progress on Google Docs
  - Due on Tuesdays at 11am
- **Final Demo**: no more annotations
  - Demo table and live demonstration on Tue Dec 11 (CSE 1202, 2-5pm)
- **Final Poster**: Project poster to be used for final demo
  - Due on Friday Dec 7, 11.59pm, we will print it for you.
- **Final Report**: Written report detailing motivation, related work, your design and development work, the system including all features, and any testing you did
  - Due on Friday Dec 14, 11.59pm
## Schedule

<table>
<thead>
<tr>
<th>WEEK</th>
<th>DATE</th>
<th>TOPIC</th>
<th>READINGS</th>
<th>ASSIGNMENTS</th>
<th>NOTES</th>
</tr>
</thead>
</table>
| 1    | Tue Oct 02| Introduction to the Course and Introduction to Ubicomp.             | 1. "How to Read an Engineering Paper"                                    |                                                                            | CSE 118  
- 11a-12:20p in CSE 2154  
CSE 218  
- 12:30p-1:50p in CSE 2154                                                    |
|      |           | [slides CSE 118]  
[slides CSE 218]                                                                |                                                                          |                                                                            |                                                                      |
2. V. Bush, "As We May Think", Atlantic Monthly, July 1945 |                                                                            |                                                                            | CSE 118  
- 11a-12:20p in CSE 2154  
CSE 218  
- 2p-3:20p in CSE 2154                                                    |
|      |           | [slides CSE 118]  
[slides CSE 218]                                                                |                                                                          |                                                                            |                                                                      |
| 2    | Tue Oct 09| Depth Cameras and Computer Vision:                                   |                                                                          |                                                                            | CSE 118  
-In-class Mini-Quiz (Week 1)  
CSE 218  
-Submit a 1-page essay (Week 1)  
-> Deadline 7pm                                                     |
• Paper Prototyping:  
-Optional:  
  - Transition to virtual and the depth                                                    |                                                                            |                                                                            |                                                                      |
|      |           | [slides CSE 118]  
[slides CSE 218]                                                                |                                                                          |                                                                            |                                                                      |
|      |           | Low-Fidelity Prototyping - Depth Cameras and Computer Vision               |                                                                          |                                                                            |                                                                      |
|      |           | [slides CSE 118]  
[slides CSE 218]                                                                |                                                                          |                                                                            |                                                                      |

Check the schedule page:  [https://ubicomp.ucsd.edu/cse118-218/schedule/](https://ubicomp.ucsd.edu/cse118-218/schedule/)
Mini Quiz on Week 5
On Google Classroom

https://docs.google.com/forms/d/1yLwYN1Y1auVEPWCGwcZsZANvcV5LMbu4kbRvKd2FHk/edit
Design Recap
What is Design?
DESIGN IS A PROCESS...

collaborate

accept

empathy

iterate

synthesize

define

show

stoke

prototype

select

ideate
Human-Centered Design
Design Methods
Principles of Human-Centered Design

- Early focus on users
- Empirical measurement
- Iterative design
Methods available at Different Stages…

<table>
<thead>
<tr>
<th>Planning</th>
<th>Context of Use</th>
<th>Requirements</th>
<th>Design</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability planning and scoping</td>
<td>Identify stakeholders Context of use analysis Survey of existing users Field study / user observation Diary keeping Task analysis</td>
<td>Stakeholder analysis User cost-benefit analysis User requirement interview Focus groups Scenarios of use Personas Existing system / competitor analysis Allocation of function..</td>
<td>Brainstorming Parallel design Design guidelines and standards Design Patterns Storyboarding Affinity diagrams Card sorting Paper prototyping Software prototyping Organizational prototyping</td>
<td>Participatory evaluation Assisted evaluation Heuristic or expert eval. Controlled user testing Satisfaction questionnaires Assessing cognitive workload Critical incidents Post experience interviews</td>
</tr>
</tbody>
</table>
User Center System Design
User Center Design
Designing Interactive Systems
Who will use the system?
User-centered Design (UCD)

• “The User-centered design (UCD) process outlines the phases throughout a design and development life-cycle all while focusing on gaining a deep understanding of who will be using the product” (Usability.gov)

• "Human-centred design is an approach to interactive system development that focuses specifically on making systems usable. It is a multi-disciplinary activity.“ (ISO 13407)

• “Framework of processes (not restricted to interfaces or technologies) in which usability goals, user characteristics, environment, tasks and workflow of a product, service or process are given extensive attention at each stage of the design process.” (Wikipedia)
User Center Design
Understanding users’ needs

- Need to take into account what people are good and bad at
- Consider what might help people in the way they currently do things
- Think through what might provide quality user experiences
- Listen to what people want and get them involved
- Use tried and tested user-centered methods
Understanding users’ needs
From problem space to design space
Refining
Ideas and Prototyping
Design right vs. Right design

• Getting the design right
  • Generate an idea
  • Iterate and develop it

• Getting the right design
  • Generate many ideas an variations (Generation)
  • Reflect and choose (Reduction)
  • Then iterate and develop your choice
Getting the design right

- Optimal design solution for one idea
Getting the right design

- Problem: final design can be only as good as that idea

- If the idea is not a good one, then the ‘best’ design solution will only be so-so
Prototyping

Iterative User Centred Design Process

- Sketch
- Prototype

Criteria Weight

Cost/Investment

UI Design
Usability
The Design Thinking process oscillates between divergent and convergent thinking modes. It can be helpful to be aware of the mode that corresponds to the design phase you are working through.
Problem Space

- Creating Choices
- Making Choices

Solution Space

- Creating Choices
- Making Choices

Steps:
1. Understand
2. Observe
3. Define Point-of-View
4. Ideate
5. Prototype
6. Test
Agile Development and Project Management
Agile Software Development

https://www.youtube.com/watch?v=OJflDE6OaSc
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
Sequential vs. overlapping development

Rather than doing all of one thing at a time...

...Scrum teams do a little of everything all the time

Scrum

Sprint goal

Return

Sprint backlog

Product backlog

Potentially shippable product increment

24 hours

Sprint 2-4 weeks
Product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

This is the product backlog
The ScrumMaster

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

—> One of the CSE 218 should be the ScrumMaster. Rotate between the three
The team

- Typically 5-9 people
- Cross-functional:
  - Programmers, testers, user experience designers, etc.
- Members should be full-time
  - May be exceptions (e.g., database administrator)
- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change only between sprints
Sprint planning meeting

**Sprint prioritization**
- Analyze and evaluate product backlog
- Select sprint goal

**Sprint planning**
- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours

Team capacity → Sprint goal
Product backlog → Sprint goal
Business conditions → Sprint goal
Current product → Sprint goal
Technology → Sprint goal

Sprint goal
Sprint backlog
The daily scrum

- **Parameters**
  - Daily
  - 15-minutes
  - Stand-up
- **Not for problem solving**
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- **Helps avoid other unnecessary meetings**

--- For your project: Google Hangout Scrum, at least 2-3 times per week
Everyone answers 3 questions

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

- These are commitments in front of peers

--> Add to the weekly Google Doc Report
The sprint review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world

—> Once a week, in person before releasing the Google Doc weekly report
Sprint retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - ScrumMaster
  - Product owner
  - Team
  - Possibly customers and others

--> Integrate into your Sprint Review
Start / Stop / Continue

- Whole team gathers and discusses what they’d like to:
  - Start doing
  - Stop doing
  - Continue doing

This is just one of many ways to do a sprint retrospective.
A Scrum reading list

- *Agile and Iterative Development: A Manager’s Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen
- *Agile Software Development Ecosystems* by Jim Highsmith
- *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle
- *Scrum and The Enterprise* by Ken Schwaber
- *Succeeding with Agile* by Mike Cohn
- *User Stories Applied for Agile Software Development* by Mike Cohn
Agile Project Management and Reports
Weekly Agile Project Updates

Ubiquitous Computing / Software Engineering (CSE 118/218) - Fall 2017

Weekly Project Team Update

To be completed every week by Sunday 11:59pm

TEAM/PROJECT NAME:

TEAM WEB PAGE:

TEAM GITHUB PAGE:

EXTERNAL LINKS: (trello, gantter, ...)

TEAM MEMBERS:

List all members, specifying: First Name, Last Name (PID) - CSE 118 or CSE 218
Weekly Agile Project Updates
Milestones and Deliverables

<table>
<thead>
<tr>
<th>Week</th>
<th>Milestones</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Weekly Agile Project Updates
Team Updates

<table>
<thead>
<tr>
<th><strong>WHAT HAPPENED THIS WEEK</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TIME AND COMMUNICATION LOG</strong></td>
<td></td>
</tr>
<tr>
<td><strong>GOALS FOR NEXT WEEK</strong></td>
<td></td>
</tr>
</tbody>
</table>
**Weekly Agile Project Updates**

**Individual Updates**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHAT DID YOU DO?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GOALS FOR NEXT WEEK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WHAT WILL YOU DO?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WHAT IS IN YOUR WAY?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

53
Project Management Tools
Project Management Tools

- Trello
- Basecamp
- Jira
- Asana
- Github + ZenHub
- Tom’s Planner
- Gantter
- Github + Zenhub
Trello

- Highly visual
- Flat and simple
- Boards, Lists and Cards
- mobile apps available

https://trello.com
Basecamp

• More structure and PM tools
• to-dos with deadlines
• project templates
• time tracking
• invoicing tools
• file backup a
• synchronizing tools
• software development tools.
• Mobile

https://basecamp.com
Jira

- Project Tracking
- Agile, Development support
- Project Planning, Issue Tracking
- Code Integration
- Mobile
- Connect to LDAP and Active Directory
- Bug Tracking
- Git Integration
- 1000's of Add-ons
- OnDemand or Hosted
- Free for Open source projects
- eMail Notifications
Gantter

- Web-based, can add as chrome extension
- EASY to create Work Breakdown Structure
- Gantt chart with multiple people and external resources
- Google Drive integration
- Export capabilities
- FREE
Github + Zenhub

- Github for free code repository and source control
- Zen hub (FREE Chrome extension) adds kanban-style boards features to Github issues
Cloud-based “connectors”

- Zapier: https://zapier.com/
- Cloudwork: https://cloudwork.com/
- IFTTT: If this then that:  https://ifttt.com/
Recommendation

**Trello**
For capturing requirements and sorting them into priorities

**Gantter**
Turning requirements into a Work Breakdown Structures and scheduling w/ dependencies

**Github+Zenhub**
Source control + feature tracking linked to commits *
Advice

• Use deliverable and planning templates
• Use a system like Google Docs or Slack to record & document your team meetings, with special attention to **action items** and **decisions**
• Take the time to make a Gantt chart, and keep it updated based on your progress
• Keep your project plan fairly simple (not too detailed at the activity level), but set clear milestones (with dates!) and stick to them
• Learn the lingo… it may help you get a job!
References


Next Steps

• Readings for this week (no discussion on Thursday)

• Design of Ubicomp Systems


• Experience Prototyping

Next Steps

• Submit your reading-summary (no annotations) by Thu 12.30pm
• Read/Annotate all papers
• Submit Development Report #1 by Tue (Nov 13) at 11am
• Work on setting up Milestones and Deliverables
Thanks