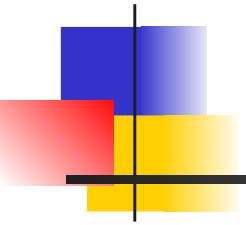


Team Software Project

Getting started

Let's Not Do Waterfall Projects

How to Get Started?

You Already Created a Plan

Your project proposal isn't a complete plan, but it contains:

- development approach (iterative)
- initial time line with goals & deliverables
- technical approach

What Will You Do in Iteration 1?

Group exercise

- each team brainstorm and show result
- can use Google Doc or project wiki
- share screen with class

10 minutes to create a google doc or Markdown with everything you want to do in Iteration 1.

Your Results

Done in class.

Milestone

For lack of a better definition, mine is:

Milestone:

an indicator that shows tangible progress toward completing a project, along with objectively verifiable criteria that show the milestone has been achieved (or not).

A milestone generally relates to achieving some goal or major work on the project. The criteria are how to evaluate that the goal or work has been satisfactorily done.

Not Objectively Verifiable Criteria

These are obviously bad, but I have see them repeatedly:

[ ] Study Django [*done or not? what's the result?*]

[ ] Write Use Cases [*when are they done?*]

[ ] Team meeting [*so what? how did it help project?*]

[ ] Task Board [*this isn't a work product or activity*]

Binary Milestones

A milestone is either 100% done or not done.

There are no "90% done" milestones.

-- *Steve McConnell, Software Project Survival Guide*

Milestones According to Agile

"Working software is the primary measure of progress."

They are not fans of documentation as milestones.

Create Your Own Milestones

Create milestones that are specific to the way your project is actually performed. Not a copy of examples.

Each milestone should have objectively verifiable ("done" or "not done") criteria.

In KU Polls Iteration 2, some students just copied my description as their iteration plan (it's not a plan). That is C- level work.

Project Initiation Milestone

- [ ] Vision statement reviewed, approved, and published
- [ ] Business case for project reviewed, approved, and published
- [ ] Initial timeline with iteration goals & features reviewed and published
- [ ] Technical approach agreed on and published
- [ ] Initial set of mock-ups of UI and screen-flow reviewed and published
- [ ] Project home page on VCS and wiki created.
- [ ] ... *what is significant work <u>your team</u> needs to do?*

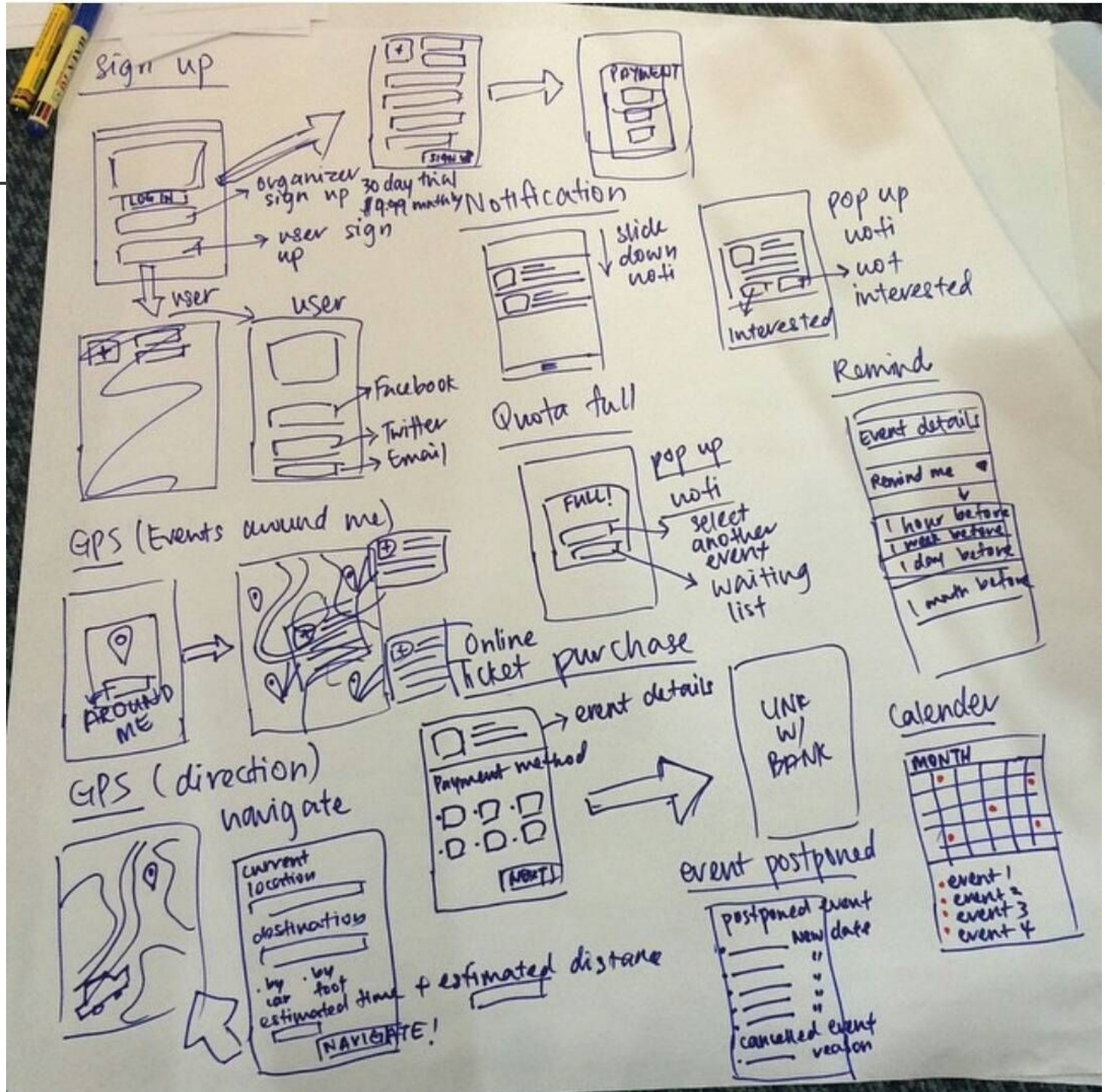
published = published on wiki or Google Drive, with link on project home. approved = team and TA/instructor all agree

Ideas for Iteration 1

Detailed mock-up with screen flow.

For each usage scenario do this:

- one screen on one sheet of paper
- one person "clicks" or enters on paper screen
- another person decides what to do next
- third person draws next screen
- repeat until user goal is achieved
- record the screen flow + processing in a file


Sign up
LOG IN
organizer sign up
30 day trial
$9.99 monthly
user sign up
PAYMENT
user
user
Facebook
Twitter
Email
Notification
slide down noti
pop up noti
not interested
interested
Remind
Event details
Remind me
1 hour before
1 week before
1 day before
1 month before
Quota full
FULL!
pop up noti
select another event
waiting list
GPS (Events around me)
AROUND ME
Online Ticket purchase
event details
Payment method
NEXT
UNI W/ BANK
Calender
MONTH
event 1
event 2
event 3
event 4
GPS (direction) navigate
current location
destination
by car
by foot
estimated time + estimated distance
NAVIGATE!
event postponed
postponed event
New date
cancelled event
vocation

Some Ideas for Iteration 1

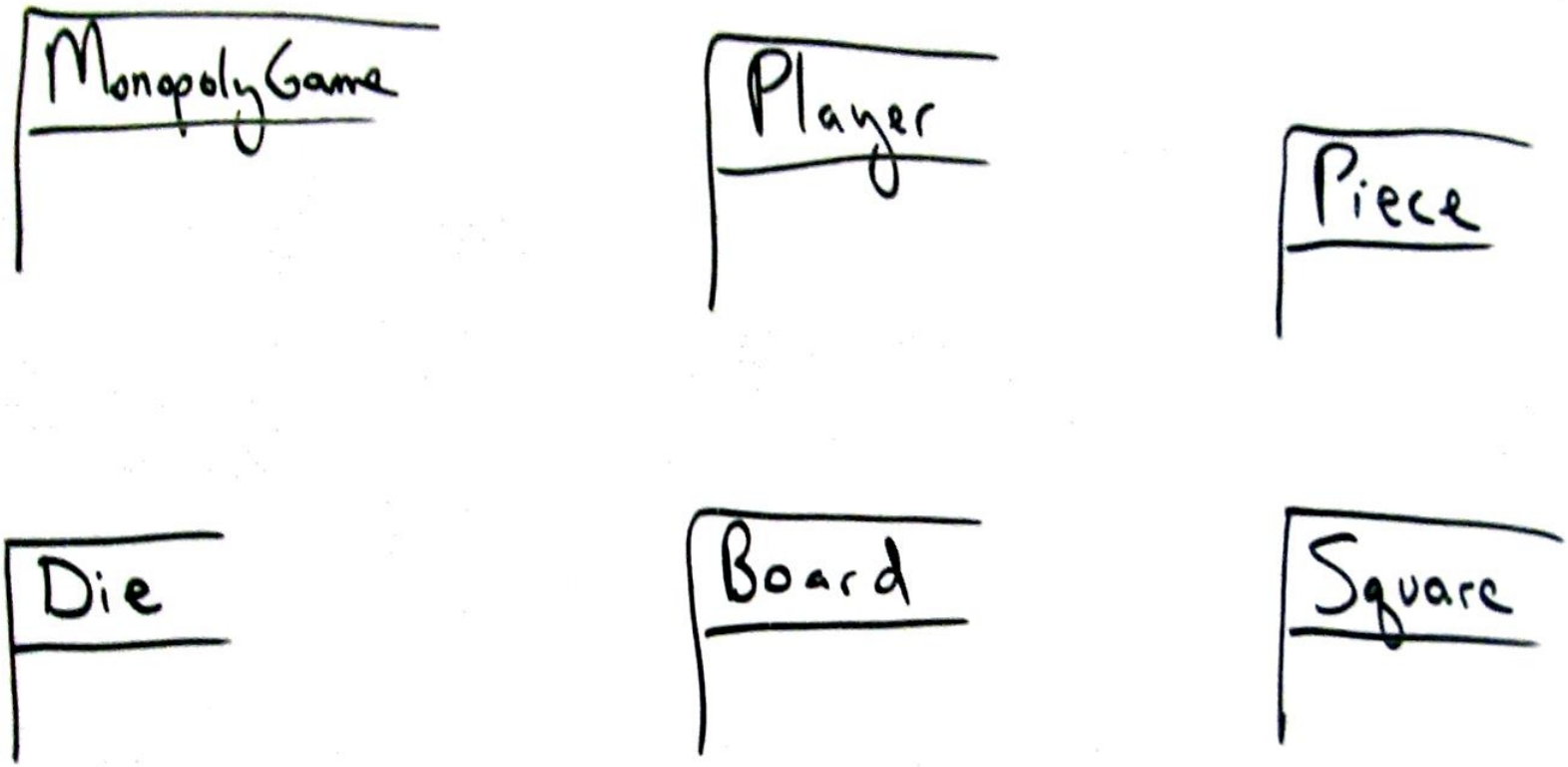
Domain Model

Create a "Domain Class Diagram" showing...

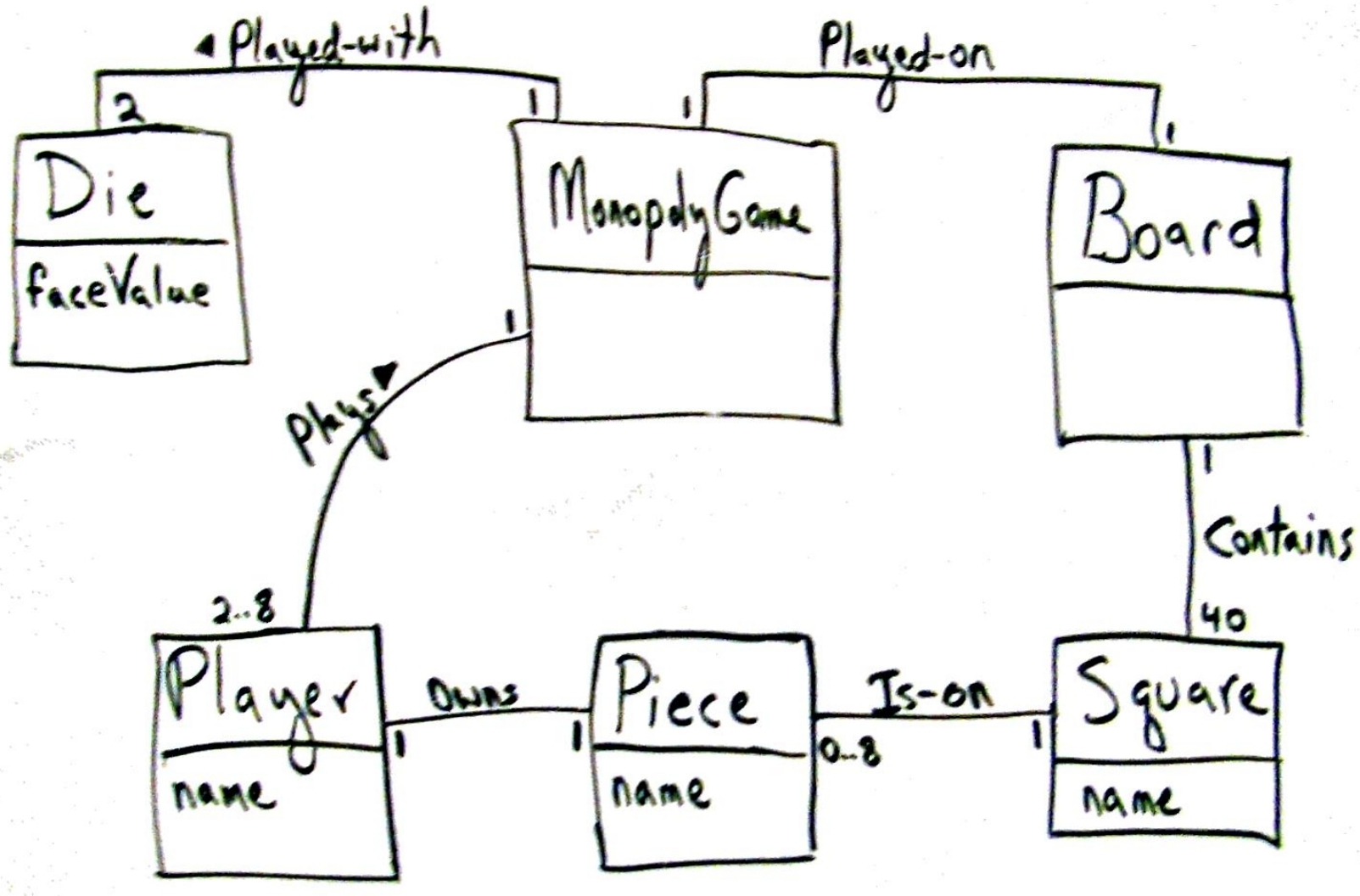
- important classes in the domain
- what they know (attributes)
- major responsibilities

Another tool for this is CRC cards.

Model: Identify classes



Identify relations and key attributes



Concentrate on dynamic aspects

Beginners spend too much time on static structure,
not enough time on dynamic (behavior)

-- Craig Larman

You should draw sequence diagrams, too.

*Identify what do you **Not** know*

Things you don't know could be risks.

Identify important things you don't know.

Domain or business related "don't knows":

what are the important terms and conditions of an apartment rental?
(so we can include them in the rental app UI)

- learn more about app domain, what users want
- ask people who do know

*Technical **Not** Know*

Examples:

- Using React with Django: *How to send and receive JSON requests?*

- *How to integrate Google Calendar or existing Calendar framework into our app?* (It's not useful to write your own calendar -- no one wants another calendar!)

- Study possible alternatives
- Work through a tutorial or sample app
- Create your own proof-of-concept app

Let's Not Do Waterfall Projects

Typical course project...

- Project presentation during last week.

- Project code & docs submitted during final.

- Instructor finds problems after semester ends.

Missed opportunity to learn.

Let's not do this.

Iteration Review & Demo

After each iteration:

- review progress and plan with TA or instructor.
- what did you do?
- any changes to the product or project plan?

Every iteration after the 1st one:

- Demo running software
- It should do more than the previous demo

Iteration Review & Demo

Who and where to demo?

- Each iteration a different team member must present
- Can be in-person or online

Key Project Elements from *Ship It!*