



Team Software Project

Getting started

Let's Not Do Waterfall Projects



How to Get Started?

You Already Have a [Partial] Plan

Your project proposal contains a partial project plan:

- development approach (iterative)
- initial time line with goals & deliverables
- technical approach
- some tools you're project will use

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

Shared 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** of the project. The criteria are how to evaluate that the work has been satisfactorily done.

Not Objectively Verifiable Criteria

These are obviously **bad**, but have been used before:

- [] Study Django [*when is it "done"? how does it show progress toward finishing the project?*]
- [] 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 your project plan & work.
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 your team 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

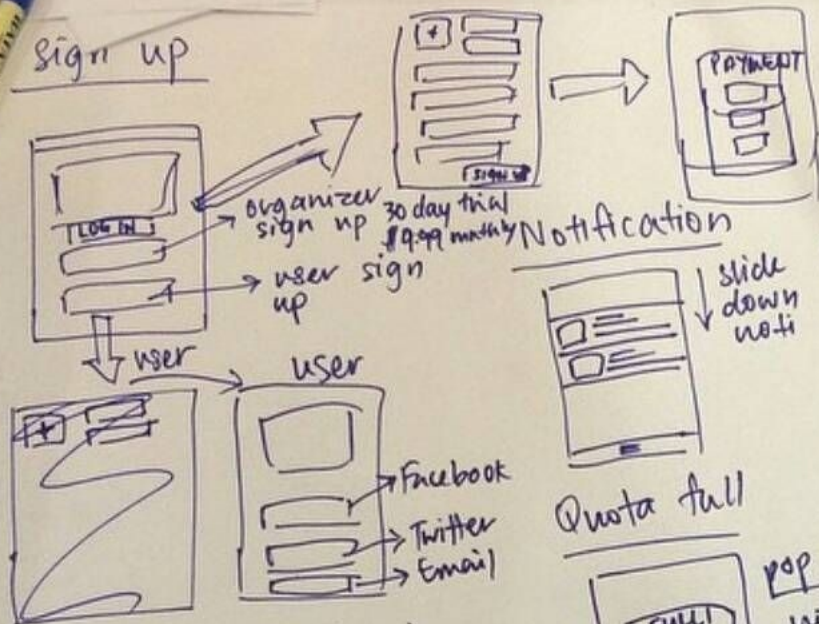
Detailed mock-up with screen flow.

For each usage scenario do this:

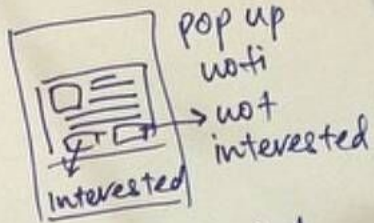
- one screen on one sheet of paper
- one person "clicks" or writes on paper screen
- another person decides what app should do next
- present user with the next paper screen
- repeat until user goal is achieved

record the screen flow + processing done in a file

sign up



Notification



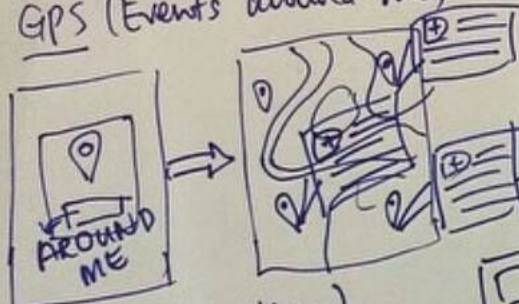
Quota full



Remind



GPS (Events around me)



GPS (direction) navigate



Online ticket purchase



event postponed



Calendar



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

MonopolyGame

Player

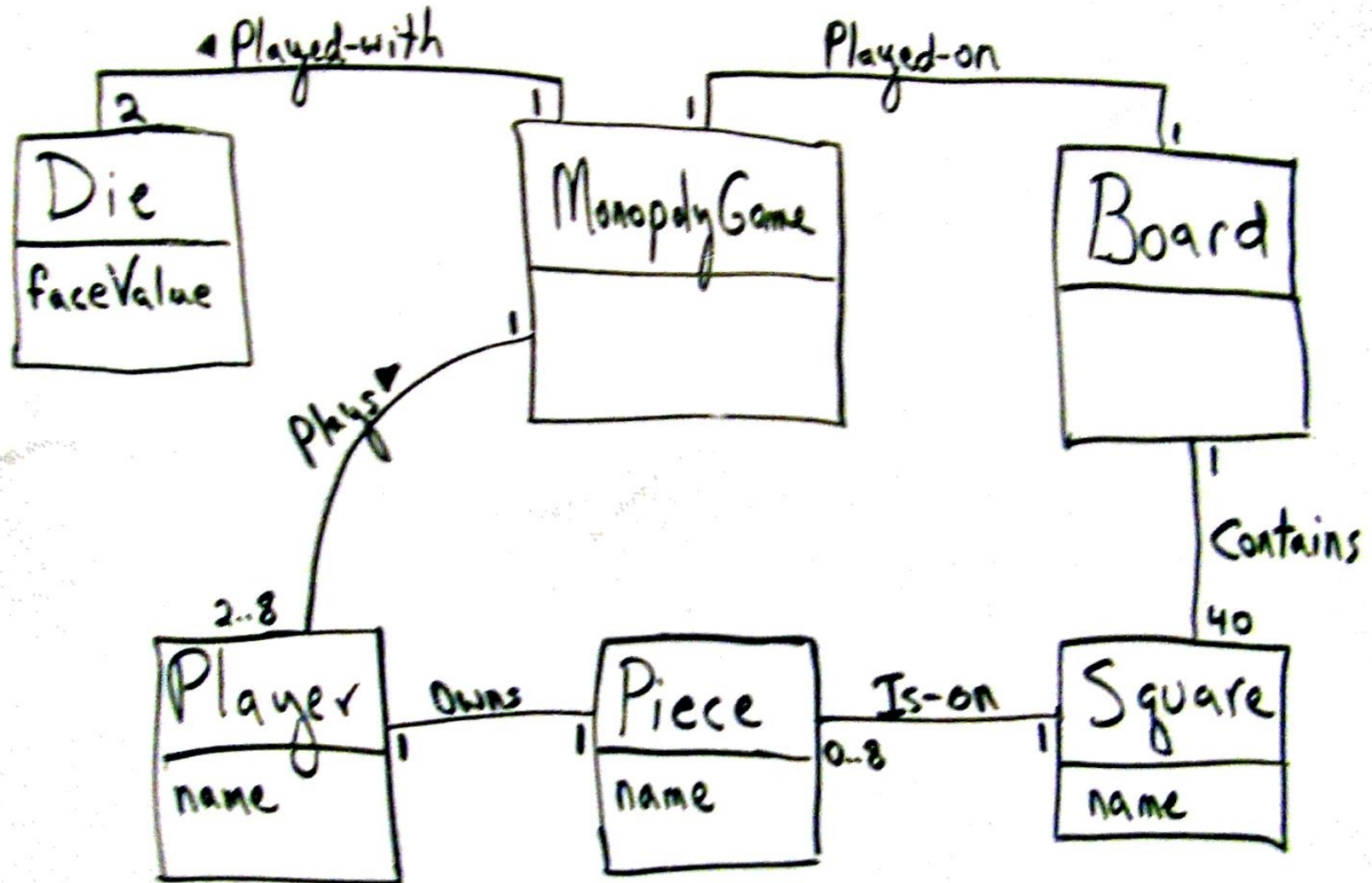
Piece

Die

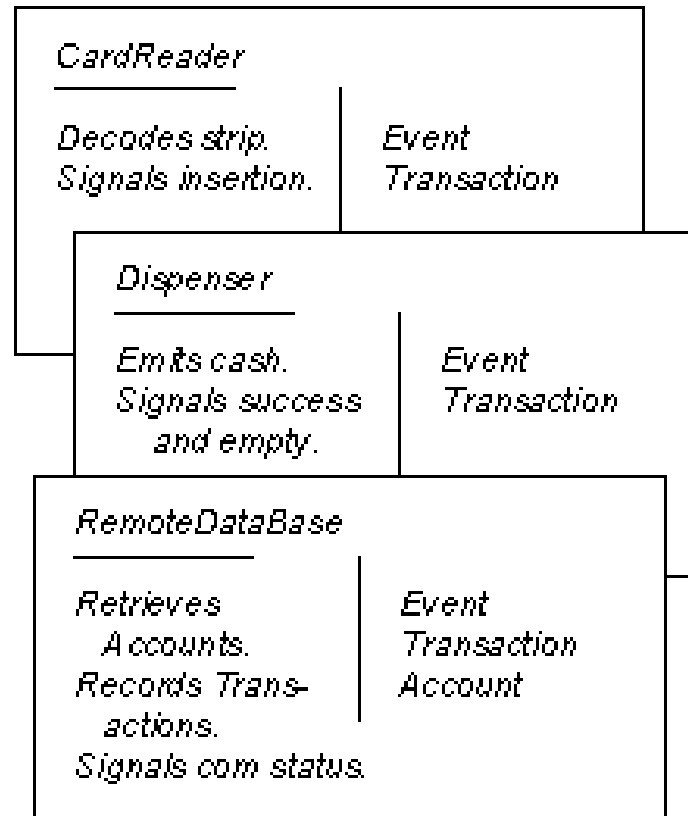
Board

Square

Identify relations and key attributes



Class-Responsibility-Collaborator Cards



Back Side: notes, important things to validate

Concentrate on dynamic aspects

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

-- Craig Larman

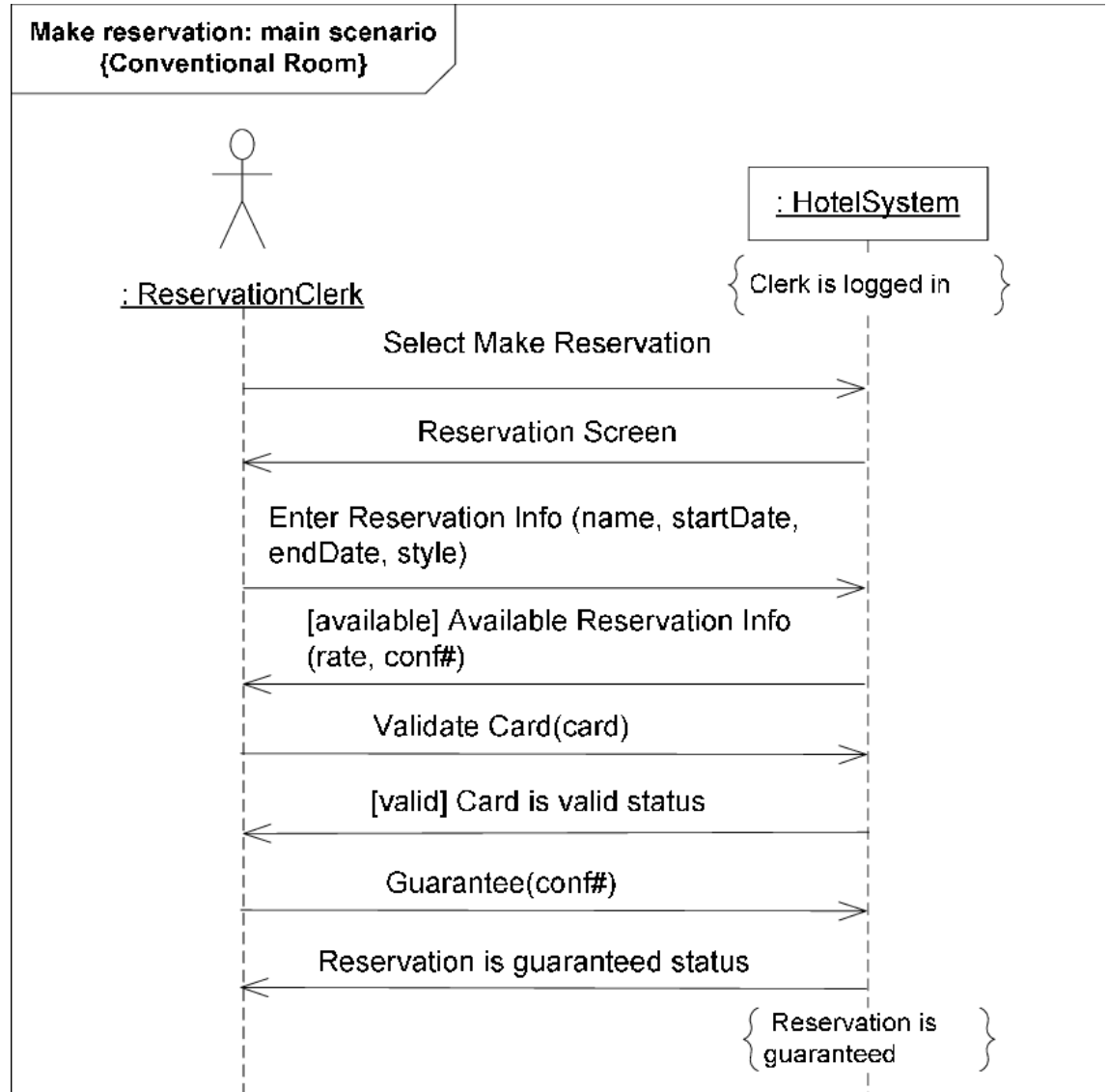
You should draw sequence diagrams, too.

System Sequence Diagram

Shows interaction between "user" & system for one usage scenario.

Use SSD to discover what system must do.

May show external services.



*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** Knowns

Examples:

- *How to send and receive JSON requests in Django?*
- *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!)

Tasks:

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

Work should be VISIBLE

*For any group project it is important
that **progress** is **visible**.*

Lack of visibility => miscommunication,
"surprises", defective or unusable work products.

- Your work should produce **work products**.
- **Work products** should be **shared** with team.
- Make it visible and online.

Agile Practice: "***Develop in plain sight***".

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 & 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