Planning

Idaho State University | Computer Science

Isaac Griffith

CS 2263
Department of Informatics and Computer Science
Idaho State University

ROAR

Outcomes

After today's lecture you will be able to:

- Describe the basic concepts of feature prioritization
- Understand how to prioritize features
- Understand how to plan out your iterations
- Select tools to use when planning

Inspiration

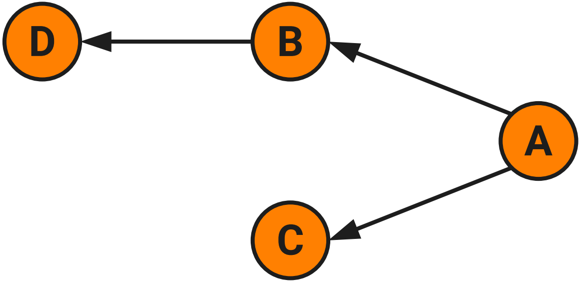
“Even the best planning is not so omniscient as to get it right the first time.”
– Fred Brooks

vfill Iteration planning

Prioritizing Features

- All applications have obvious dependencies between features in terms of what needs what before it can be built
  - If there is no game board you can’t think about moving pieces on the board yet
  - If there is no database persistence layer it would be a waste of time to work on saving the game state.
- The feature dependency graph has an edge from a feature to a feature it depends on: you can’t get the feature running until all its dependencies are working.

Dependency Graph



- Feature A depends on features B and C
- Feature B depends on feature D
- Thus, D must be built before B or A, and C must be built before A
- So the highest priority features are D and C

Prioritizing Features

- From the feature dependency graph there are multiple successful paths to building your final app over time: the onion of how it will appear layer by layer over time
- Reify the onion into a series of deadlines, regular (two-week for us) iterations with concrete goals of features/etc implemented in each successive layer/iteration
- Your job as a software developer is to find the most natural route through the dependency maze to your final app.

How to prioritize

- Define a subset of features to start with, the key features, which when implemented will give bare-bones functionality
- Since there are multiple people on a team, you need a “parallel programming plan” so you can proceed simultaneously with minimal blocking/conflicts.
- Make sure to “program to interfaces, not implementations” (a basic OO design principle) between team members: have a known meet-point
    - RESTful server APIs are one example, also distinct models and views in UI programming, distinct page views, etc.

Iterations

An iteration of a project is a planned global step in the development of a piece of software.

- Its one layer in the onion above
- An iteration should not be too big: add some features, modify the design to do one aspect differently, etc.
- Iterations give you many little deadlines to successfully hit – every two weeks in this class.

Planning Your Iterations

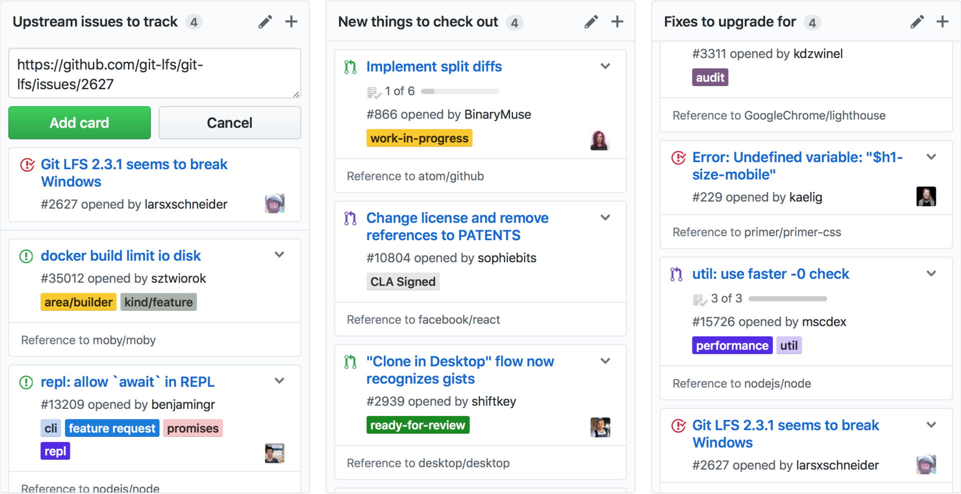
How to plan iterations in practice?

- An iteration plan maps feature implementation and other tasks on to which iteration they should be implemented in.
- Have (as in **write out**) a detailed plan for the next iteration and a fuzzier one for more distant iterations
- Continually revise your iteration plan as you go
  - Maybe some things in the previous iteration proved too hard – bump up to current iteration or divide into smaller problems over several future iterations.
  - Make clear what the new set of features you want to add in the next iteration is - take your fuzzy ideas from the previous iteration plan and refine them.

Planning Your Iterations

- Use a tool to help write and keep the plan updated and for everyone to know the plan.
  - There are many tools to help you do this now, Trello is a big one; also see ZenHub, kanban, waffle.io, etc
  - For your projects you will be required to use GitHub's version, the Project boards, starting in iteration 3. A picture from their tutorial:

Planning Your Iterations



Upstream issues to track 4

https://github.com/git-lfs/git-lfs/issues/2627

Git LFS 2.3.1 seems to break Windows
#2627 opened by larsxschneider

docker build limit io disk
#35012 opened by sztwiorok
area/builder kind/feature
Reference to moby/moby

repl: allow `await` in REPL
#13209 opened by benjamingr
cli feature request promises repl
Reference to nodejs/node

New things to check out 4

Implement split diffs
1 of 6
#866 opened by BinaryMuse
work-in-progress
Reference to atom/github

Change license and remove references to PATENTS
#10804 opened by sophiebits
CLA Signed
Reference to facebook/react

"Clone in Desktop" flow now recognizes gists
#2939 opened by shiftkey
ready-for-review
Reference to desktop/desktop

Fixes to upgrade for 4

#3311 opened by kdzwinel
audit
Reference to GoogleChrome/lighthouse

Error: Undefined variable: "$h1-size-mobile"
#229 opened by kaelig
Reference to primer/primer-css

util: use faster -0 check
3 of 3
#15726 opened by mscdex
performance util
Reference to nodejs/node

Git LFS 2.3.1 seems to break Windows
#2627 opened by larsxschneider

Planning Your Iterations

- Cards can be linked to GitHub issues in the issue tracker system - see key issues in graphical format, move between iterations by drag-and-drop.
- We will look at the project board of a past projects

Are there any questions?