Review So Far

+

Testing Levels

Reminders

- Iter 0-3 due Sunday
  - Double check your GitHub access!
- Iteration 0 demo day- Tuesday!!!
  - Be ready
  - Submit your Heroic link under the demo day assignment
  - Everyone must talk.
- Test 1- Thursday
  - Covers everything up till end of lecture today
  - Review guide listed in the Test section on Canvas

Overall Development Processes

- What are the differences between the Waterfall, Spiral, and Agile Processes?
- Time? Documentation? Emphases? When do each work well?
- When is each good? When is each bad?

Silo vs. SOA

- There are siloed and SOAed applications. What are these? What are the benefits and drawbacks of each? When would you use one over the other?

- How might you design a SOA for airline company? (For example.)

Introducing SaaS

- What is a SaaS application? What are the benefits to developers and customers in creating SaaS applications? Can anything be a SaaS app? Why or why not?
    - How does SaaS differ from traditional software?
    - What kinds of applications are NOT well suited for SaaS? Why?

High level architectures
3 tiered, shared nothing

- Three-tier Shared-Nothing Architecture
  - What are the three tiers and how are they “shared-nothing?”
  - Why is this a good fit for large-scale SaaS applications?
  - Which tier doesn’t scale as well as the others and why not?

Picking Languages
How to decide?

- What are some considerations to take when picking a language to use?

- What are first steps that you take in learning about a language to see if it’s the right tool for the job at hand?

- Why are we using Ruby to create SaaS applications (other than because I told you to)?

High level architectures
How to decide?

- What is the Model/View/Controller design pattern?
- What are alternatives to MVC and when might you want to use them?
- Why are we using Rails to create SaaS applications (other than because I told you to)?
  - Active Record vs DataMapper design patterns
  - CRUD – what is it?
- Benefits and drawbacks of these design patterns? Why choose one over the other?

Developing a new project: Customer interaction

- What are 4 principles of the agile development process?
    - User stories
    - Why good? What drawbacks?
    - Connextra format (4 parts)
    - SMART (Specific, Measurable, Achievable, Relevant, Time-boxed)
        - Good read: http://www.chiefsimplicityofficer.com/2010/04/user-stories-its-smart-to-invest/
        - A summary: http://programmers.stackexchange.com/questions/184137/how-to-write-smart-objectives-as-an-agile-developer
- LoFi UI diagrams and story boards
    - Why good? What drawbacks?
    - What are they?

Self and Product Management

- What is velocity?
  - What are burndown charts?
  - Why are tools like Zenhub helpful in project management?
    - For the developer
    - For the manager

Versioning

- What are the benefits of using versioning tools?
- What can you do with them and why is their use so important when working in teams?

Debugging Techniques

- You have an error. What do you do? (Semi-automated)
  - Debugger
  - Log files
  - Print statements
- Beyond that- RASP
  - Read, Ask, Search, Post

Behavioral Driven Design

- Definition
- Why is BDD especially applicable when using the agile process?
- Do you think it’d be effective in a waterfall or spiral process?

Test Driven Design

- Definition
- Why good?
- In what development processes is it applicable? How would TDD change when using a different process?

Validation vs Verification

- What’s the difference as you’ve experienced in your projects?
  - Validation is the process of checking whether the specification captures the customer's needs. (Is it doing what they want?)
  - Verification is the process of checking that the software meets the specification. (Is it actually working?)
- Define and describe an example of each of the following:
  - Unit Tests - testing individual methods or lines of methods
  - Module or Functional Tests - (fuzzy… tests that go between functions)
  - Integration Tests - (everything you’re doing with Cucumber)
  - System or Acceptance Tests - you have the tools to do this (just need a contract)
- Think back on your homeworks: What was being tested? Why did those tests help? Is testing always the answer? Is it foolproof? When is more needed?