1. Introduction
   1. Overview of my (rather unique) rapid software development lab in the Hospital setting
      1. Collaboration With Clinicians, Radiologists: “Wouldn’t it be great if I had an app that…”
   2. Overview of technologies that are found in our lab.
      1. Why specific tools are preferred to keep the development lifecycle short.
      2. The importance of not writing something that someone else has already OpenSourced
   3. Discussion of the Projects that I was involved in.
      1. iOS Development,
      2. Rails;Ruby Development
2. **Software Development Best practices** (code review, readability, reusability)
   1. Usage of a ‘self documenting’ code style (Objective-C) to minimize redundant commenting.
   2. Incorporating ideas of the *Agile* model of software development.
   3. **Object Oriented Design** 
      1. How MVC makes many different coding challenges easier.
      2. Using Object oriented design to create novel solutions
         1. Case study: using one ‘web’ Controller for an “Sign-on” page, a “main application UI” view, and a “radiological Image View”
      3. Using the tremendous opensource world to do the heavy lifting
         1. Tools like Bundler, Cocoapods, homebrew, NPM make trying out new Opensource Controls/Objects/Frameworks a quick process. No need for git’s submodules.
3. **Software lifecycle** and **Industry Best practices** 
   1. Using Mockups and wire framing to dream about what a project might look like.
   2. The utility of a Version Control System like GIT to track progress, Collaborate, divide up work, and maintain reference to old code
      1. Also, Project Management/Issue Tracking tools as productivity springboards. (Both of which simply *must* integrate with GIT)
4. **- Ruby on Rails** (Learning OOP by doing , learning the value of a web development framework, SDK importance, HTML/CSS/Javascrpt)
   1. The Boilerplate that you don’t have to write.
      1. Built in MVC makes the framework
   2. The expressiveness of the Ruby language makes code highly maintainable
5. Conclusion
   1. I walk away with strong opinions on how IDEs, Frameworks and languages should work
   2. I walk away with experience taking an application that professionals use from concept to production. Learning good lifecycle/industry practices along the way.
   3. Learned Ruby, Objective-c (cocoatouch), Javascript, Coffeescipt, HTML/CSS