CS 410: INTRODUCTION TO SOFTWARE ENGINEERING

Project Scoping

Nam Chu Hoai
**SAAS**

**PROS**

- Easy monetization
- Easier maintenance, immediate updates
- Only one platform (web)
- Rich ecosystem of open source tools, libraries and services
- Little config/setup costs

**CONS**

- Hacky on top of HTML/CSS/JS
- Browser Idiosyncrasies
TRADE-OFFS

"Software Engineering is a constant exercise in making tradeoffs that rewards knowledge of short-/long term cost/benefits"
STARTED AS A SIMPLE CHAT APPLICATION …
... THEN OUR CLIENTS WANTED A DASHBOARD
... THEN WE NEEDED CUSTOM FUNCTIONALITY

Backend as a Service

Parse

Dashboard

Web Server

Mobile App
… THEN WE WANTED A DASHBOARD

- Backend as a Service
- Mobile App
- Web Server
- Internal Dashboard
- Dashboard

www.wellframe.com
TRADE-OFFS

“Hidden cost of abstraction layers"
TRADE-OFFS

- Be aware of the limitations of your choices
- Understand which components are critical
- Where are your core competencies?
TRADE-OFFS

- That complexity over time is known as technical debt
- Mitigation strategies:
  - Post-damage:
    - Find natural product features at which to refactor
    - Refactor sprints
  - Pre-damage:
    - Solid Product Management
... THEN WE NEEDED CUSTOM FUNCTIONALITY
... THEN WE WANTED A DASHBOARD
PRODUCT MANAGEMENT

- Crucial for efficient complexity management
- Coordinates all moving pieces, delivers context for effective decision making and aligns all teams
I want my patients to remind themselves of medications they need to take every day.
LIFECYCLE OF A FEATURE

Are there other things you want them to remind themselves of?
Good call, actually I also want to remind them of appointments with their PCPs.
Hey a feature coming up is to remind patients of various things like medications or their appointments. See any issues?
Nah, but what kind of recurring schedules do we have to support? One-off or other complex schedules? The latter will add a bunch of complexity.
LIFECYCLE OF A FEATURE

What’s the kind of schedules you’d want?
LIFECYCLE OF A FEATURE

Daily or one-off is fine.
LIFECYCLE OF A FEATURE

- Stakeholder
- Engineer
- Product Manager
Software Engineering is a constant exercise in managing complexity as you make changes to a system.
LANGUAGE/ECOSYSTEM

• Typed vs. Untyped Language
• Nature of problem domain (data, computation, IO, business logic)
• Package Ecosystem
• Community
SQL VS NOSQL

- SQL is good for basic, low to medium throughput applications
- Expressive queries, matches many people’s views of databases, tables records
- NoSQL is good for everything else:
  - Scale
  - Unstructured data
  - Graph Data
  - Realtime/Streams
SECURITY

- Salt Passwords
- CSRF
- XSS [https://xss-game.appspot.com/level1](https://xss-game.appspot.com/level1)
- Principle of Least Privilege
- Social Engineering
MISC

• Do outside projects
• Contribute to Open Source
• Code is read more than it is written
THANK YOU!