

Thoughtful Software Design

Lincoln Startup Week



Andy Unterseher

- Software Engineer at Don't Panic Labs
- @unter
- @dontpaniclabs
- <http://dontpaniclabs.com/blog/>



Software is a “wicked” problem

A **wicked problem** is a problem that is difficult or impossible to solve because of incomplete, contradictory, and changing requirements that are often difficult to recognize.

https://en.wikipedia.org/wiki/Wicked_problem



Agile and Evolutionary Design

- “In its common usage, evolutionary design is a disaster. The design ends up being the aggregation of a bunch of ad-hoc tactical decisions, each of which makes the code harder to alter.”

Martin Fowler

<http://martinfowler.com/articles/designDead.html>



There is a lot of suffering...

"We're gonna need *another* hot fix"

"Yes, we're feature complete but we need 4 weeks to stabilize"

"That may *seem* like a small change, but..."



"He's the only one who knows how that code works."

"We should just re-write this whole thing"

"I hate release day"

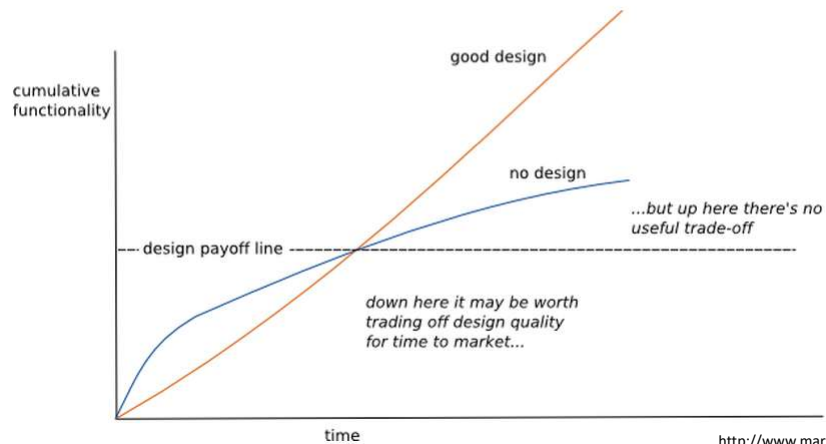


But we're just a startup...

- Can we afford to spend the time and money?



Design Stamina Hypothesis



<http://www.martinfowler.com/bliki/DesignStaminaHypothesis.html>



The cost...

Table 3-1. Average Cost of Fixing Defects Based on When They're Introduced and Detected

Time Detected					
Time Introduced	Requirements	Architecture	Construction	System Test	Post-Release
Requirements	1	3	5-10	10	10-100
Architecture	—	1	10	15	25-100
Construction	—	—	1	10	10-25

Source: Adapted from "Design and Code Inspections to Reduce Errors in Program Development" (Fagan 1976), *Software Defect Removal* (Dunn 1984), "Software Process Improvement at Hughes Aircraft" (Humphrey, Snyder, and Willis 1991), "Calculating the Return on Investment from More Effective Requirements Management" (Leffingwell 1997), "Hughes Aircraft's Widespread Deployment of a Continuously Improving Software Process" (Willis et al. 1998), "An Economic Release Decision Model: Insights into Software Project Management" (Grady 1999), "What We Have Learned About Fighting Defects" (Shull et al. 2002), and *Balancing Agility and Discipline: A Guide for the Perplexed* (Boehm and Turner 2004).

Code Complete (McConnell 2004)



Why are you writing your software?



The Zen of Architecture

For the beginner architect, there are many options

For the Master architect, there are only a few

Juval Lowy

<http://channel9.msdn.com/Events/TechEd/NorthAmerica/2010/ARC206>

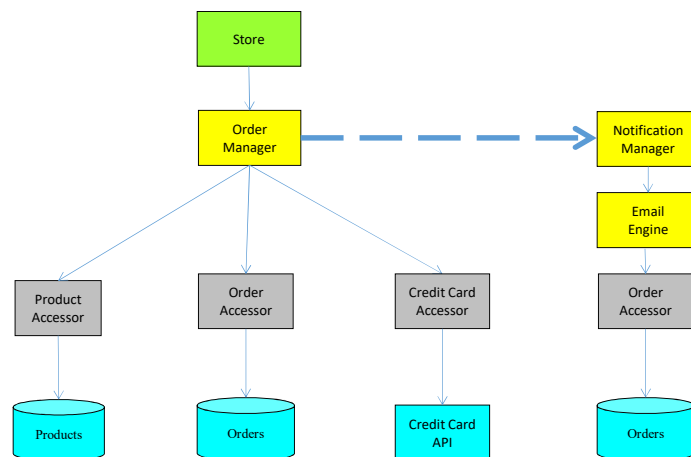


How do we do this process?

- 2 Phases
 - Requirements Gathering
 - Time is variable
 - Uncovering core use cases of the system
 - Architecture and Design
 - Time boxed to 1 week
 - Pen and paper architecture
 - Testing Core Use Cases against it



User submits credit card order

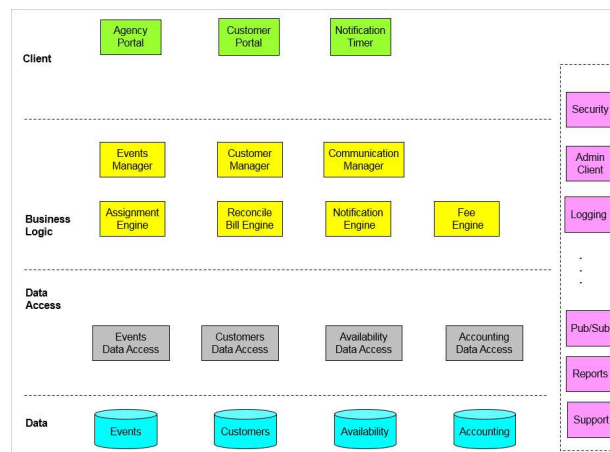


Designing for Volatility

- How will my business change for the same customer over time?
- How will it change at the same time across customers?



End Result



Establish a consistent design identity

“I will contend that Conceptual Integrity is the most important consideration in system design. It is better to have a system omit certain anomalous features and improvements, but to reflect one set of design ideas, than to have one that contains many good but independent and uncoordinated ideas.”

Fred Brooks (1975)



Open vs. Closed Architecture

- We always strive for a closed architecture
 - Less choices for developers
 - More consistent design



Testing

- We highly value testable code
 - Loose coupling allows for testing in Isolation



Scalability

- Hopefully your startup is wildly successful
 - Are you prepared to scale from the start?



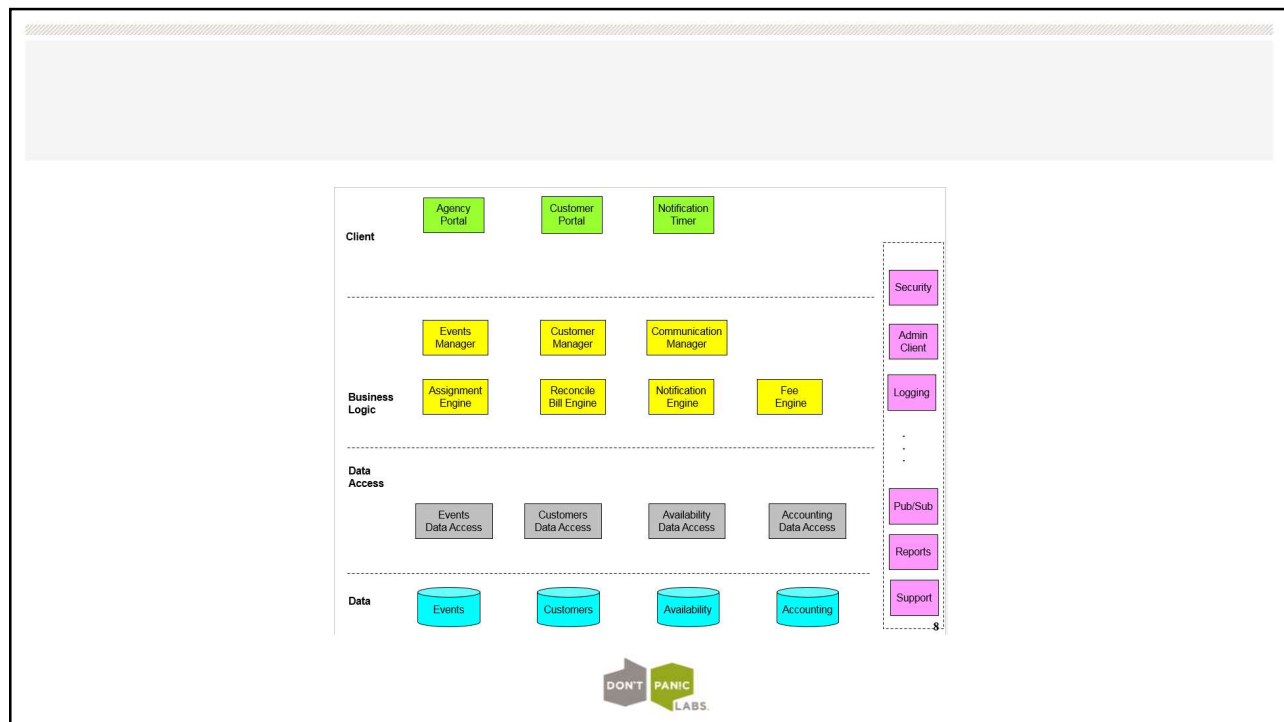
Prototyping and Spike Releases

- These are our tools for rapidly developing to prove a business concept or software technology:
 - Done outside of the current architecture
 - Time boxed
 - Ported into the primary system after acceptance



Is this a microservice architecture?





Recap

- Benefits
 - Upfront planning allows us to uncover issues in the beginning
 - Development can go faster
 - Confidence of a proven design to guide the process
 - Less paralysis by analysis on designing components
 - Tested and Scalable system
 - Avoid the dreaded re-write of the system

<https://github.com/unter/Presentations>

