

Team Structure & Software Architecture



Architectural is Organizational

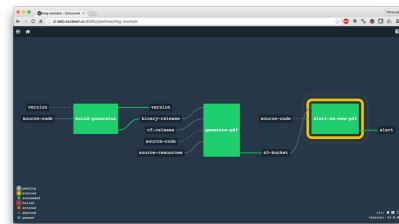
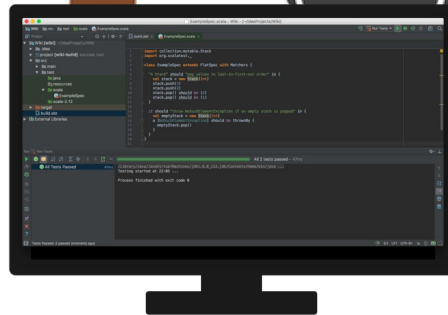
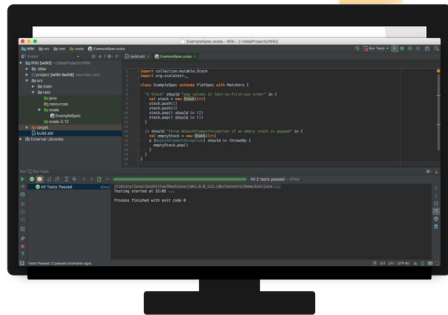
Team Structure & Software Architecture

1. Talk about the ways teams are structured
2. What it has to do with Architecture
3. What we should do about it.

Product Starts



[illegible]

An illustration of a man and a woman. The man on the left has short black hair, thick black eyebrows, and a slight smile. The woman on the right has shoulder-length black hair, wears a purple headband with a pink bow, and large pink-rimmed glasses. She is smiling broadly. Both characters have a light skin tone and are wearing simple necklines.

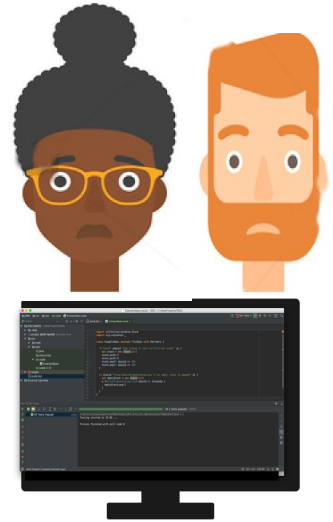
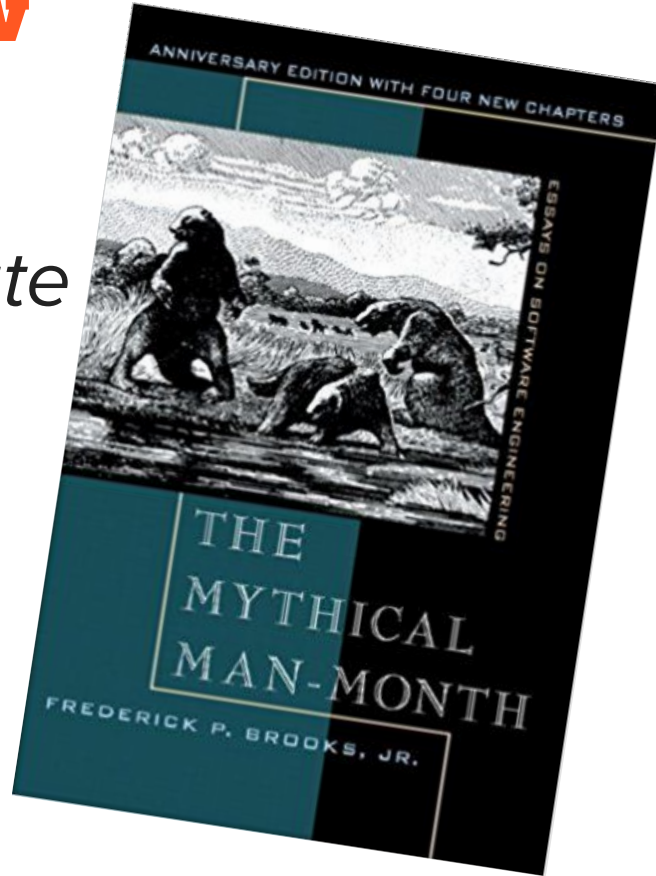


Brooke's Law?



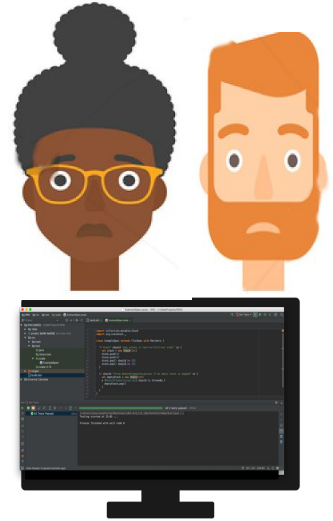
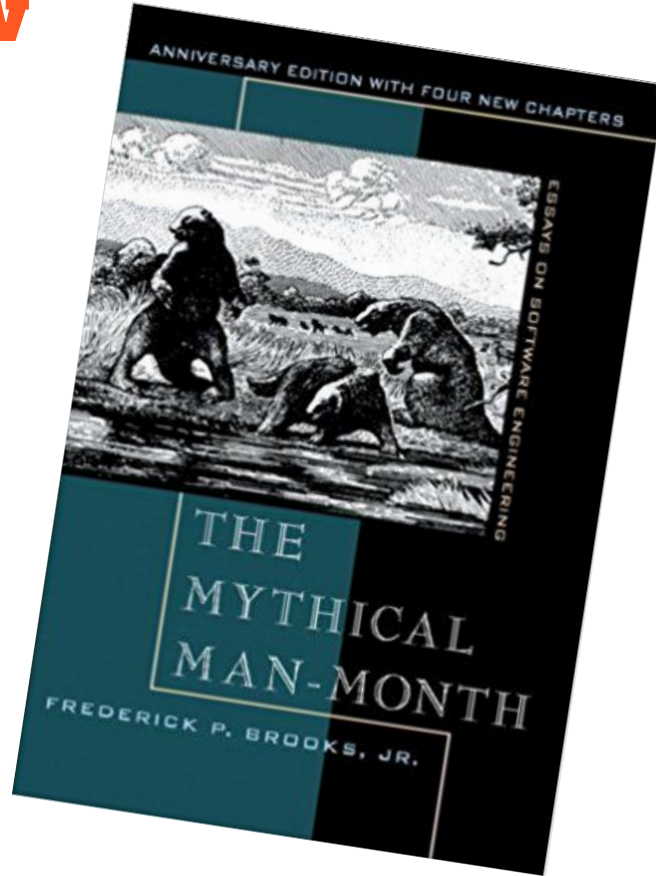
Brooke's Law

“Adding human resources to a late software project makes it later”



Brooke's Law

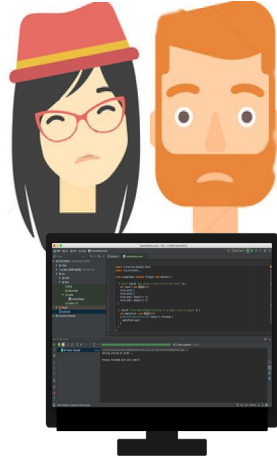
"Diminishing returns of adding new team members"



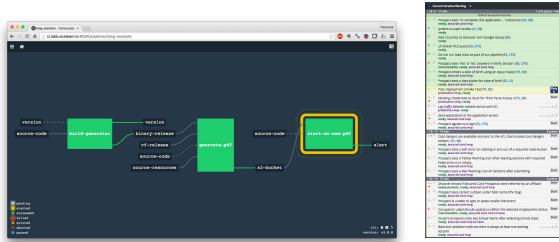
What does our software team do?



Velocity....



Split into smaller more focused teams.



But, how?



Frontend Team



Option 1: Horizontally



Backend Team



Option 2: Divide Teams by Feature

Auth Team



Checkout Team



Option 3: Divide by Business Capability

Shipping Team

Shipping
Service



API

Orders Team

Orders
Service

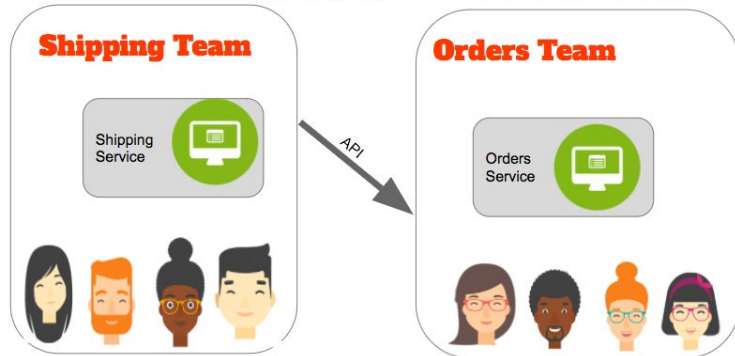


So which option is the best to do?

Option 1: Horizontally



Option 3: Divide by Business Capability





Kent Beck

@KentBeck



Follow

any decent answer to an interesting question begins, "it depends..."



RETWEETS

429

FAVORITES

263



10:45 AM - 6 May 2015

1967

1967

Apollo 1 results in Disaster

1967

Apollo 1 results in Disaster

Ronald Reagan elected Governor of
California

1967

Apollo 1 results in Disaster

Ronald Reagan elected Governor of
California

Israel Six Day War

1967

Apollo 1 results in Disaster

Ronald Reagan elected Governor of California

Israel Six Day War

Conway's Law was introduced

1967

What is Conway's Law?

1967

What is Conway's Law?

"Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure."

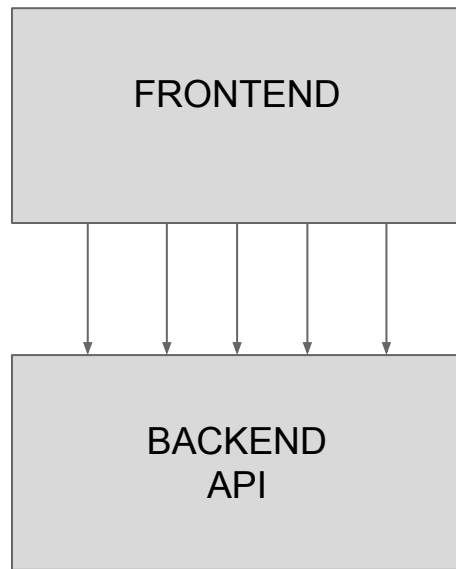
Option 1: Horizontal

Pros:

- Allows technological specialization

Cons:

- Requires features to go through two teams to be implemented



Option 1: Horizontally



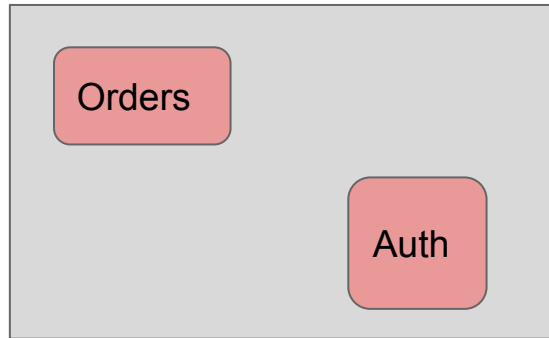
Option 2: Divide Teams by Feature

Pros:

- Architectectural Simplicity

Cons:

- Lack of team autonomy
- Unclear team boundaries



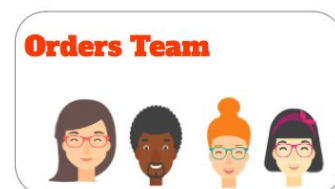
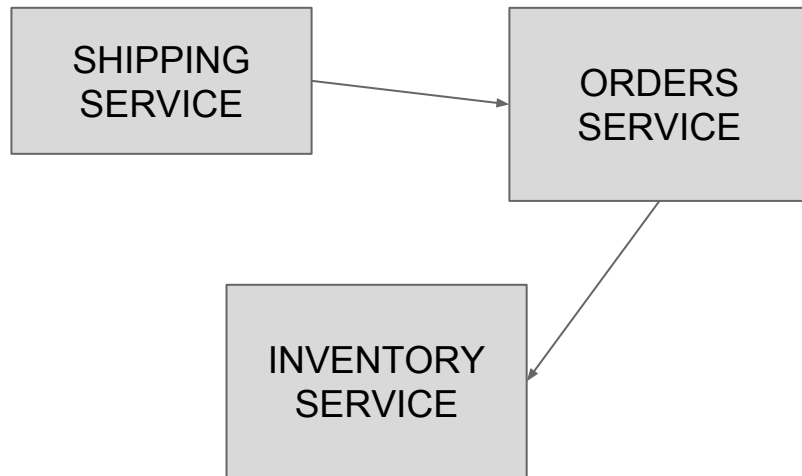
Option 3: Microservices

Pros:

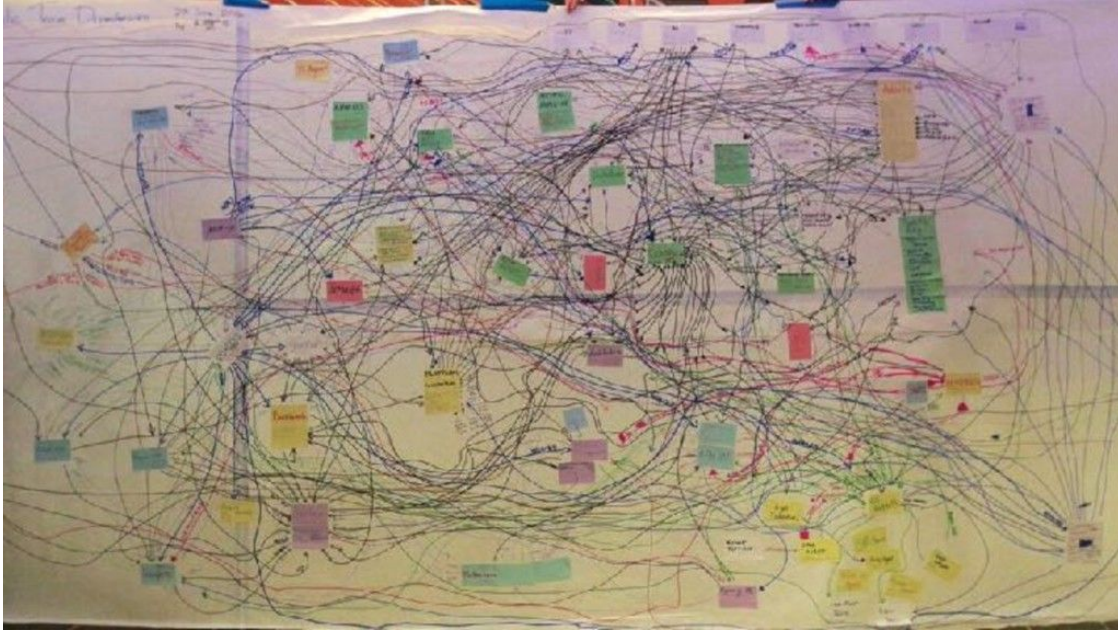
- Independent Teams

Cons:

- Complexity of a distributed system



Option 3: Microservices



Shipping Team



Orders Team



Understand Conway's Law

Understand Conway's Law

Avoid the Consequences of Conway's Law

**Understand
Conway's Law**



**Avoid the
Consequences of
Conway's Law**



**Use Conway's
Law to your
Advantage**



Inverse Conway Maneuver



If you want....

Small Microservices built around business domains

Then structure...

Small teams around business domains



If you want....

Well defined interfaces that encompass Biz Logic so
new UIs can be built quickly and independently

Then structure...

Independent backend and UI teams



If you want....

Formalized interface for a component so that other systems teams can interact with it.

Then structure...

Separate teams for the component and the current consumers



If you want....

Well structured organization with clear responsibilities

Then build...

A well structured software architecture with clear responsibilities





Tammer Saleh

@tammersaleh

Follow



Remember: Conway's Law works in both directions. Monolithic codebases produce monolithic companies.

8:30 AM - 10 Sep 2015



In Conclusion

Recognize that team organization decisions are
Architecture decisions

Software Architecture has the power to help
refactor organizations themselves

Any Questions?