Serverless + Modern Agile

@mikebroberts

https://symphonia.io/
Serverless Architectures

Serverless architectures refer to applications that significantly depend on third-party services (known as Backend as a Service or "BaaS") or on custom code that’s run in ephemeral containers (Function as a Service or "FaaS"). The best-known vendor host of which currently is AWS Lambda. By using these ideas, and by moving much behavior to the front end, such architectures remove the need for the traditional ‘always on’ server system sitting behind an application. Depending on the circumstances, such systems can significantly reduce operational cost and complexity at a cost of vendor dependencies and (at the moment) immaturity of supporting services.
Coming in June!

What is Serverless?
Understanding the Latest Advances in Cloud and Service-Based Architecture

Mike Roberts & John Chapin
Resources for this talk:


@mikebroberts
Continuous Experimentation
I - From Feature Factories to Product Labs

@mikebroberts
What is the point of software engineering?
“The goal of software is to sustainably minimize lead time to business impact”

— Dan North ‘Beyond Features’
How has software lead time changed over the years?
Back in the old days…

- Releasing software was expensive
- Infrastructure was expensive
- *Changing* infrastructure was expensive
“We used to ship our product every 12–18 months, in a 3–5 year period we shipped 3 or 4 releases. Now we ship 3 releases every week.”

— Sree Kotay, CTO, Comcast Cable
Project → Product

Output → Outcome
“We’re starting to see applications be built in ridiculously short time periods.”

— Adrian Cockroft, AWS
We don’t know upfront what the best solution is
“Most of our ideas suck!”

— Jeff Patton ‘Output vs. Outcome & Impact’
All companies use technology, but true tech companies are different: technology is at the core of their business. It is the catalytic engine that drive their competitive advantage.

– Joanna Parke
THE LEAN STARTUP

How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses

ERIC RIES

THE NEW YORK TIMES BESTSELLER
Feature Factory → Product Lab
II - Challenges of this approach
Technical Challenges

1. Minimize incidental complexity
2. Reduce infrastructure cost and commitment of new systems
3. Minimize the tech debt impact of highly fragmented ecosystem
Process and Culture Challenges

4. Minimize initial deployment time
5. Culture of learning
6. Spread ownership of product innovation
Safety Challenges

7. Safe to fail
8. Budgetary safety
9. Data and Security Safety
III - Serverless as an experimentation platform

@mikebroberts
A Quick Serverless Reminder
Attributes of a Serverless Product

• No management of Server Hosts or Processes
• Self auto provision & auto-scale based on load
• Costs based on actual, precise, usage
1 - Minimize incidental complexity

• Pre-built generic logic
• Lightweight model for custom code (FaaS)
• Handles system administration
• Automatic horizontal scalability
2 - Reduce infrastructure cost

• Costs based on actual usage, not any commitment

• Costs scale up from, and down to, zero, automatically

• Removes cost-saving incentive to co-locate features
3 - Minimize Tech Debt

Encourages a modular, well-encapsulated, loosely coupled architecture
If you’re only using Serverless for cost savings you’re missing a big trick
A technical platform is not sufficient - our organizations need to change too.
IV - Continuous delivery, continuous value
DevOps Culture
4 - Minimize initial deployment time

• More than Continuous Deployment…

• … need to be able to bring up *entirely new* stacks of components in minutes or seconds
5 - Culture of Learning

• Our customer
• Our tools
• Each other
Tear up the Backlog!
“Software, Faster”
6 - Spread ownership of innovation

Product ownership is a team’s responsibility, not an individual’s
V - A new approach to safety

@mikebroberts
7 - Safe to Fail

“Google’s data indicated that psychological safety, more than anything else, was critical to making a team work”

— New York Times
“If you think that’s a big failure, we’re working on much bigger failures right now — and I am not kidding. Some of them are going to make the Fire Phone look like a tiny little blip.”

— Jeff Bezos, on the Fire Phone
“By removing blame, you remove fear; by removing fear, you enable honesty; and honesty enables prevention.”

— Bethany Macri, Etsy c/o DevOps Handbook
8 - Budgetary safety

- Team visibility and ownership
- Automation and tooling
9 - Data and Security Safety

• Make it part of a team’s regular work
• Build cross-organizational tooling
• Team autonomy for many access controls…
• But embrace change request process too
VI - Modern Agile

@mikebroberts
MODERN AGILE

- Make People Awesome
- Experiment & Learn Rapidly
- Deliver Value Continuously
- Make Safety a Prerequisite

http://modernagile.org/
Make People Awesome
Experiment & Learn Rapidly
Deliver Value Continuously
Make Safety A Prerequisite
Serverless + Modern Agile + DevOps + Software, Faster = Continuous Experimentation
How will you transform your organization from a feature factory to a product laboratory?
Resources for this talk:


@mikebroberts
mike@symphonia.io
https://symphonia.io/