THE SATURN TIMES

FRIDAY, APRIL 27, 2019

A FINE GRAINED CATASTROPHE

How Microservices Brought Down an Entire Industry

By Michael Keeling | @michaelkeeling | http://neverletdown.net

Software start-ups in the first decade of the 21st century were known for their playful open floorplans, self-organizing teams, and pancake flat org charts. To Marie and her moving team, the open floorplan of Quust Software Studio meant they stood a chance of finishing their job before happy hour was over. Like many software start-ups in recent months, Quust was forced to declare bankruptcy and liquidate assets following the rapid decline of the globe's tech economy. Marie and her team were taking the furniture.

> "WE HAD NO IDEA WHAT WE NEEDED TO DELIVER … MICROSERVICES LET US SHIP CODE LITERALLY 100s OF TIMES A DAY."

Lando Studebaker, a talkative thirty something with an affinity for tea over coffee, was the chief architect at Quust Software Studio. He orders an aromatic black tea at the cafe where we met on a chilly February afternoon a few days before the Quust offices were to be emptied.

"I'm still not quite sure what went wrong. One day we were planning the company trip to Cuba, the next day, we were shutting down the shop." Studebaker grimaces as he recalls the decisions his development team made over the past year that led to the company's decline.

Like many now bankrupt start-ups, Quust Software adopted a microservices architecture and hosted their services in Amazon's cloud. For a company selling mobile apps meant to assist buzzword-compliant flash mobs jazzed about Scala, microservices was a perfectly reasonable architecture choice.

Studebaker explained, "We were a start-up! I mean, shit! We had no idea what we needed to deliver but we needed to deliver it yesterday. Adopting microservices let us ship code literally 100s of times a day."

At the time Quust and Studebaker decided to adopt Microservices it was a loosely defined architectural style meant to promote business agility. The idea was that using small, "micro" web services, each a self-contained unit, would be easier to understand and faster to deploy, thus cheaper to maintain over time.

The number of start-ups using microservices skyrocketed shortly after the name was coined in 2013.

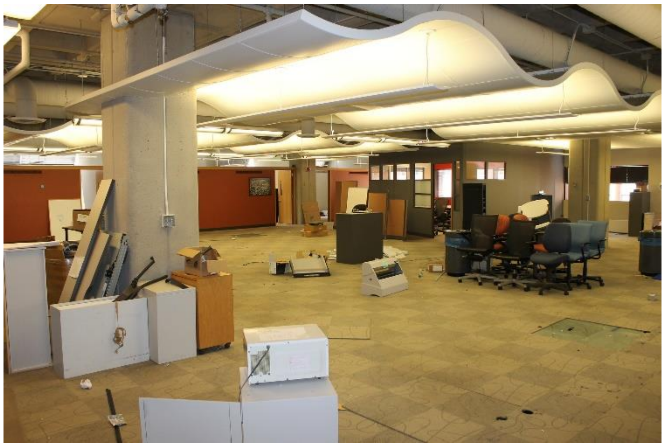
LANDO STUDEBAKER
(PHOTO CREDIT: DANIEL FOSTER, FLICKR)

While the core principles behind microservices were basically defined as “fine grained SOA” with newer technologies (for example REST over SOAP), few organizations who used the style actually knew what they were doing.

Microservices require a level of sophistication that few development shops at the time could handle. Another problem was that while the basic principles and technologies were relatively easy to use, the promise of microservices could really only be realized by mature organizations with the right culture.

Quust ran into problems early, burning close to six months of venture money setting up basic infrastructure before they shipped their first services. Eventually Quust was able to release services to their Amazon cloud using the Netflix OSS Stack released by Netflix, one of the pioneers and poster children of the microservices style.

Technology was not the only problem Quust faced. Individuals were largely allowed to grow the system organically without clear direction from Studebaker or other technical leaders. In spite of Quust’s flat org chart and start-up culture, company politics and a lack of visible progress began to create friction between the development team and management.



Quust Software Studios the day of our interview. (Photo Credit: Corey Seeman, Flickr)

At one point, management barred the use of automated quality enforcement tools that run in the production cloud. These tools create minor disruptions in the production system to prove that specific, desired properties are present in the deployed software. One example is the Netflix OSS Chaos Monkey. Without these tools and without other means of architectural enforcement, the deployed system architecture quickly deteriorated.

Quust’s experience is not unique among the software industry. So many companies adopted cloud and microservices so fast and with so little understanding for what they were getting into that the style is often cited as one of the top reasons for the Tech 2.0 Crash. One analyst estimates that upwards of 2 out of 3 failed start-ups were using microservices at the peak of Bubble 2.0.

With so many inexperienced adopters, it was only a matter of time until a Chernobyl-like explosion nuked the industry. The 2016 “Rouge Lemur Incident” that decimated the entire Eastern Amazon AWS cloud region for a week forced everyone to adopt even higher high availability strategies.

With increased availability came increased operating costs. When combined with poor engineering and monitoring practices, and questionable business plans, “pay as you go” became a fast and easy way to bankrupt cloud-based start-ups.

Towards the end of our interview, Studebaker took a deep breath and admited what many in his position have also concluded, “We didn’t really *need* to use microservices. It’s just that everyone else was doing it and it sounded like fun.” It’s difficult not to make the obvious allusion to lemmings jumping off a cliff in droves.

This "article from the future" is my position paper for the SATURN 2015 microservices workshop. It's a satire depicting an extreme (hopefully) alternate reality that could result from our current, popular fascination with microservices architecture. Can the misguided use of an architectural style bankrupt entire companies and cause an economic collapse? Hopefully not, but at the same time we've seen several process and technology related incidents in the past decade or so with long and lasting economic impacts.

Let's try to nail some of these ideas about microservices down so we can provide appropriate and positive guidance for implementers and future maintainers.

Michael Keeling
http://neverletdown.net
@michaelkeeling
April, 2015

The characters and story depicted in this position statement is completely fiction and not intended to represent any real people or companies, living or dead, historical or otherwise.