

The top challenges with microservices

by Charles Green - Thursday, June 08, 2017

<http://blog.belatrixsf.com/the-top-challenges-with-microservices/>

Over the past 18 months we've seen increasing attention placed on microservices. Standard bearers such as Netflix and Facebook have highlighted how microservices can be used to create more nimble and responsive software applications.

At Belatrix we have just published a [whitepaper](#) examining microservices and outline some of their many benefits for tech organizations. However it is also clear that despite the benefits of modularization and containerization, many organizations continue to struggle with microservices. In this blog post therefore, I want to highlight what we've seen as some of the most common challenges among organizations and their typical pain points with microservices:

- **Challenges of reimagining and rearchitecting a software product.**

Probably the biggest challenge for the software architect considering microservices is the need to reimagine and re-think how the application will work. It is one thing to implement microservices in a Silicon-Valley start-up, and another for an enterprise with complex, legacy systems.

- **Testing can become challenging.**

Particularly with integration tests, it becomes necessary for the quality assurance engineer to clearly understand each of the different services in order to write the test cases effectively. Debugging meanwhile can mean the QA engineer having to analyze logs across different microservice environments.

- **Effective management and teamwork required to prevent additional complexity.**

While one of the key benefits of microservices are that in theory they should help technical organizations lower the complexity of dealing with monolithic applications, in reality the opposite can occur as teams struggle to coordinate their work with so many different, rapidly moving elements. This is why hiring effective project managers will be key to your success.

- **Databases need to be completely decoupled from each other.**

We have sometimes seen software architects sharing the same database model across different microservices. This is something which shouldn't happen. When transitioning to cloud microservices, you need database models 100% decoupled from each other.

My advice to organizations considering microservices is the same as in every conversation about software architecture. There are choices which the software architect needs to make, and those choices will be influenced by your specific business requirements. Make sure to evaluate the possible benefits that microservices can provide for each application, and then make the call on whether to proceed. For some

applications, it may simply not be necessary to break them down, and far easier to keep them as they are. However, as we point out in the whitepaper, winning companies from Nike to Netflix are rapidly moving to a microservices architecture because they recognize the benefits they can provide, in particular as they seek to become more customer-centric and responsive to customer demands.

PDF generated by Kalin's PDF Creation Station