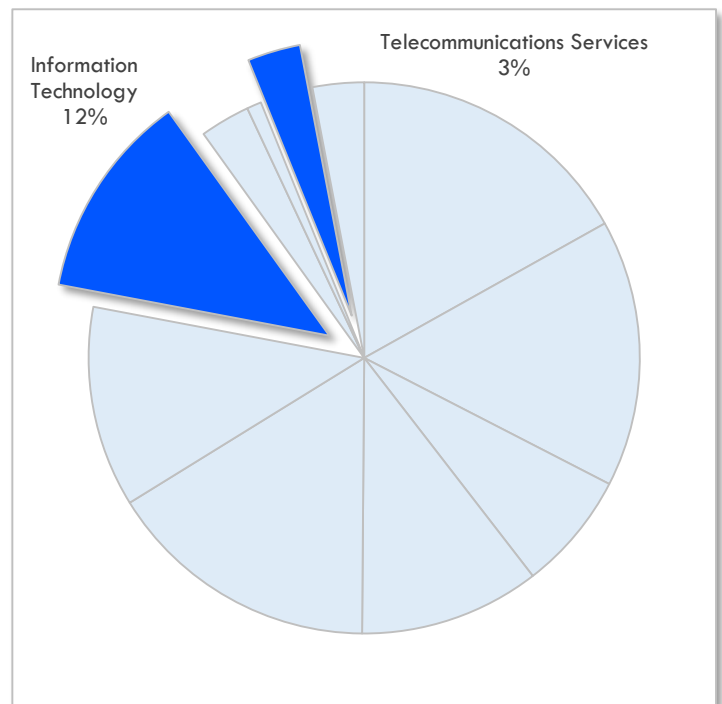


Software Factories: DevOps as the New Six Sigma

The Problem

In 2018, IT matters – half of the top 10 companies in the S&P 500 are tech companies and almost a quarter of the revenue generated by S&P 500 companies comes from tech. Just like companies aren't able to survive without HR or finance, many businesses won't be able to stay afloat without a technology competency. With technology touching every aspect of business from product delivery to supply chain management, leaving technology behind is a recipe for failure. To solve this, companies must create a software factory to manufacturer value.



S&P 500 Sector Breakdown by Revenue – Jan 2018

From Six Sigma to DevOps

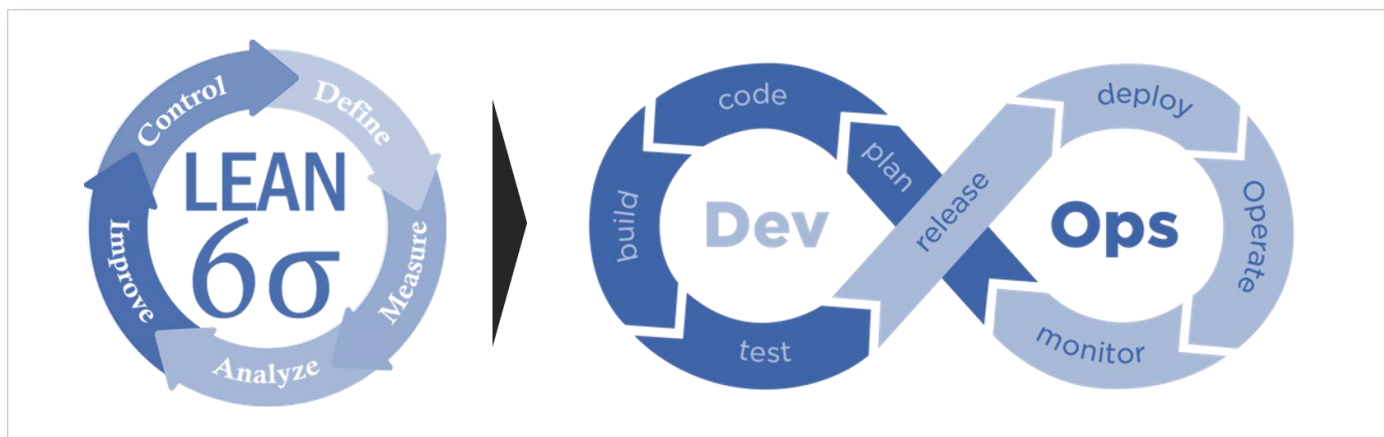
The idea of software development as a manufacturing process is not new. Just as one of the 90,000+ copies of The Phoenix Project to understand the need to treat software development like a manufacturing floor. But to many, this goal seems allusive. There have been waves of literature around this, from the old work around bimodal IT, which became taboo in 2017, to digital practices discussing DevOps for hardware. And just as Six Sigma was a continuous cycle to improve manufacturing processes, DevOps provides a framework to optimize the software factory through a continuous, iterative process. Theory is great but to practically implement solutions that make IT a high throughput process, companies must identify the most critical bottlenecks they face. The devil is in the details for companies looking to get ahead in manufacturing value through software development.

Process Drives Outcomes

When creating a software factory, it is important to understand the implications of partial implementation. In a highly automated manufacturing process, the bottlenecks can make or break the effectiveness of the process. Applying this to DevOps, this means accelerating development effort but leaving backend processes manual and underfunded is a recipe for disaster. Though development effort can be a good starting point, constantly iterating and improving the process is vital for DevOps to yield the outcomes businesses want. Without this approach, companies might as well go back to waterfall development.

So what does this mean in terms of improving the process. First off, it means companies looking to incrementally improve processes they've already radically transformed are missing out on solving the big bottlenecks. It also implies that companies must look to emerging technologies to eliminate these pain points to remain competitive. Since most enterprises don't have the resources to allocate to building tools to enable development, they should look at how they can adopt new technologies to build their factory, not on creating the machinery.

Finally, DevOps – just like Six Sigma– requires constant improvement to get better. This means that leaving the same, correctable mistakes to repeat is unacceptable. Monitoring and modifying the process as a whole is what leads to a successful factory. And just like manufacturing, the process is an means to enabling a business goals, not a goal in its own right.



Quick Wins Don't Need to Come at the Cost of Long Term Gains

Understanding the pitfalls of not constantly iterating doesn't mean quick wins don't exist. As mentioned, it is important that companies keep a razor sharp focus on building the software factor, not the machinery. This doesn't mean that radical transformation can't come through quick wins. Just as Six Sigma is an iterative process, DevOps allows for quick wins to be impactful – which is the same thinking that companies used when moving from waterfall development to agile.

As long as they have a long term goal guiding them (which should be redefined once they reach it), they'll find that some quick wins are just as transformative as giant transformation processes. Keeping in mind that the software factory is made up of interactions between people, processes, and technologies, small changes in any of these three categories has the potential to radically improve output from the software factory.

So Where to Start

In our humble opinion, we see QA as the low hanging fruit for companies to pick off. Why? It's simple:

- Software factories require high levels of automation and monitoring to be successful.
- Companies frequently funnel the majority of their funds into development teams, leaving the rest of the factory behind.
- QA processes need to focus on the intent of testing, not the repetitive functions such as generating and maintaining scripts, provisioning data, or executing tests.
- AI is robust and economic enough to make QA teams far more productive

Looking at these factors, it's clear that QA can be radically transformed through AI to eliminate this bottleneck in the software factory.

AutonomIQ is a cloud platform that enables product and IT teams to autonomously test, release and deploy software, thereby increasing velocity of software releases without compromising quality. With pre-built integrations to common web applications and SaaS providers, customers can instantly create test cases, generate test scripts and test data, and execute tests. Using deep-learning and AI algorithms, AutonomIQ detects changes, enables self-healing for test assets and provides advanced diagnostics. In real world situations, AutonomIQ has been shown to provide over ~50% improvement in speed and quality compared to existing tools and techniques. For more information, please visit www.AutonomIQ.io