



Tokyo-based Rakuten is one of the largest electronic commerce and Internet companies in Japan. Rakuten's Ichiba e-commerce platform is among the world's largest by sales.

Large, established enterprises like Rakuten must address issues such as monolithic architecture, silo mentality and long release and QA cycles in order to remain competitive against smaller, leaner startups.

Small Teams, Microservices and DevOps

Rakuten realized that enterprises are still investing into long QA cycles simply because they have the resources to do so. Long release cycles, however, are preventing them from competing against leaner, more agile startups that started with a DevOps culture. Small, 3-5 person cross-functional teams are able to use

“DevOps is a turning point for the ICT Industry.”

Yasunobu Kawaguchi (Agile Coach)

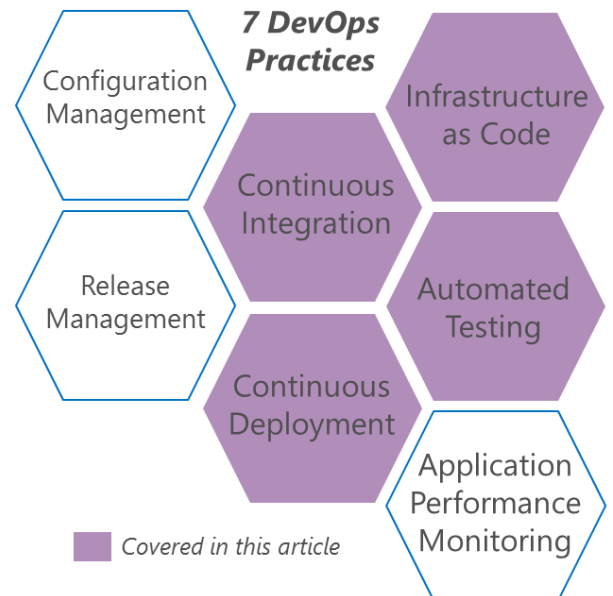
advances in technology to rapidly build profitable services, and enterprises need to change their established culture by embracing DevOps practices if they wish to remain competitive.

“DevOps is a turning point for the ICT Industry,” says Yasunobu Kawaguchi, and Agile Coach in Rakuten who is influential in Agile adoption in Japan. “The world is about to change – and it requires company-level judgement.”

Rakuten already has experience in navigating watershed moments. Before virtualization became mainstream, Rakuten had a Service division and an Operations division. Servers were set up by Operations, while Services configured the OS. Virtualization forced the teams to work together, since both teams provided key input for effectively provisioning and managing Virtual Machines.

In much the same way, DevOps is bringing similar challenges to traditional Development and Operations divisions. For example, Agile practices enable Developers to release at a rapid cadence. This puts pressure on Operations to provision and update environments far more frequently, while still maintaining stability. Ops has had to change the way they work – for example, instead of manually provisioning environments, Ops needed to learn how to treat infrastructure as code, which enabled automated provisioning. Infrastructure as Code is forcing Dev and Ops to collaborate to solve common problems.

Rakuten is also embracing a microservices architecture. They are decomposing large system into smaller, loosely-coupled microservices each managed by small, cross-functional teams. The teams are embracing agile practices such as automation, allowing rapid cadences. Rakuten now enjoys the agility of a startup with the resources of an Enterprise organization, allowing them to remain competitive in the market.



Merging Dev and Ops

Kotaro Ogino, a Test Automation engineer, speaks about the journey of merging of Dev and Ops at Rakuten. "Historically, the two divisions had plenty of conflict. The Developers 'kaizen' by demonstrating positive results."

Ogino's team triggered change by automating simple "smoke tests" that tested Japanese character handling for their search engine. Historically this testing was not performed effectively, since the Ops team were non-Japanese. Once the tests had been automated, they were able to rapidly detect localization issues. The Dev and Ops teams now collaborate around infrastructure as code as well as test automation, and so are able to implement Continuous Integration for their system tests – a process that Ogino calls "Continuous System Testing".

The more the teams collaborated, the more the need for a regular DevOps meeting grew. The teams now meet regularly and are rapidly developing mutual understanding. Ogino recalls how they started referring to "non-functional" testing as "operability testing". The change in name shifted the mindset – now Ops was empowered to feed operational requirements to the Devs as requirements early on, enabling the Devs to gain insight into operational aspects of their applications.

Continuous Testing

"When I automate something, I consider the value stream," says Ogino. "Testing is traditionally considered a dedicated process for Dev or Ops, but it should rather be considered as a part of the value flow from requirements analysis to release. Test automation provides value to both Dev and Ops, enabling rapid and reliable releases."

“ Testing should
be considered a
part of value flow.

Kotaro Ogino (Test Automation engineer)

In the past, Developers had to request environments for automated testing from Ops, causing delays and bottlenecks in the QA process. Once they had collaborated with Ops to define environments as Chef recipes, it's become common for automated builds to fire Chef recipes, automatically deploying and configuring

infrastructure for test automation, removing the need for manual intervention. QA cycles that used to take up to 5 days are now run automatically every night. "Using Jenkins and 20 – 30 concurrent tasks, we can run deep integration testing in under 2 hours." This enables daily deployments that have been deeply tested.

"Our web API systems are primarily Java, PHP and Ruby based. We use Junit, Selenium and CucumberJVM for our testing," says Ogino. "We run performance, data-consistency, availability, search-quality and index testing, as well as recovery and operability testing. Since our systems run in many different regions across the world, we leverage Azure for performance testing across multiple regions."



Kotaro Ogino from Rakuten