

Casos Seleccionados do Livro- O Manual do DevOps

Fonte: Livro Manual DevOps, de Gene Kim

Google

- Empresa: Google
- Problema: Gerenciamento de um grande volume de *commits* de código e implantações de produção.
- Prática: Desenvolvimento baseado em tronco e entrega contínua, com revisões obrigatórias de código.
- Resultado: Mais de 13.000 desenvolvedores trabalhando em uma única árvore de código-fonte, realizando mais de 5.500 confirmações (*commits*) de código por semana.

Target

- Empresa: Target
- Problema: Dificuldades em obter ambientes e executar implantações, além de acesso restrito aos dados necessários.
- Prática: Implementação de uma equipe de Habilitação de API para permitir que as equipes de desenvolvimento entregassem novos recursos rapidamente.
- Resultado: Habilitação de 53 novos recursos de negócios, incluindo Ship to Store e Gift Registry, e atendimento a mais de 1,5 bilhão de chamadas de API por mês em 2014.

Salesforce

- Empresa: Salesforce
- Problema: A capacidade de desenvolver e lançar novas funcionalidades parou, apesar de ter contratado mais engenheiros.
- Prática: Implementação de princípios e práticas de DevOps, incluindo testes automatizados em todos os estágios da criação do aplicativo e do ambiente.
- Resultado: Redução dos prazos de implantação de seis dias para cinco minutos até 2013, permitindo-lhes processar mais de um bilhão de transações por dia.

Etsy

- Empresa: Etsy
- Problema: O processo de implantação era motivo de estresse e medo.
- Prática: Implementação do Deployinator, uma ferramenta desenvolvida internamente para facilitar o processo de implantação.
- Resultado: O processo de implantação tornou-se uma operação de rotina, acontecendo de vinte e cinco a cinquenta vezes por dia.

Google

Estudo de caso Code Reviews no Google (2010)

O Google é um excelente exemplo de empresa que emprega desenvolvimento baseado em tronco e entrega contínua em escala. Como observado anteriormente neste livro, Eran Messeri descreveu que, em 2013, os processos no Google permitiram que mais de treze mil desenvolvedores trabalhassem fora do tronco em uma única árvore de código-fonte, realizando mais de 5.500 confirmações de código por semana, resultando em centenas de implantações de produção por semana. Em 2010, havia 20+ alterações sendo verificadas no tronco a cada minuto, resultando em 50% da base de código sendo alterada a cada mês.

Isso requer uma disciplina considerável dos membros da equipe do Google e revisões obrigatórias de código, que abrangem as seguintes áreas:

- Legibilidade de código para idiomas (impõe o guia de estilo)
- Atribuições de propriedade para subárvores de código para manter a consistência e a correção
- Transparência de código e contribuições de código entre equipes

A Figura 42 mostra como os prazos de entrega de revisão de código são afetados pelo tamanho da alteração. No eixo x é o tamanho da alteração e no eixo y é o lead time necessário para o processo de revisão de código. Em geral, quanto maior a alteração enviada para revisões de código, maior o prazo de entrega necessário para obter as aprovações necessárias. E os pontos de dados no canto superior esquerdo representam as mudanças mais complexas e potencialmente arriscadas que exigiram mais deliberação e discussão.

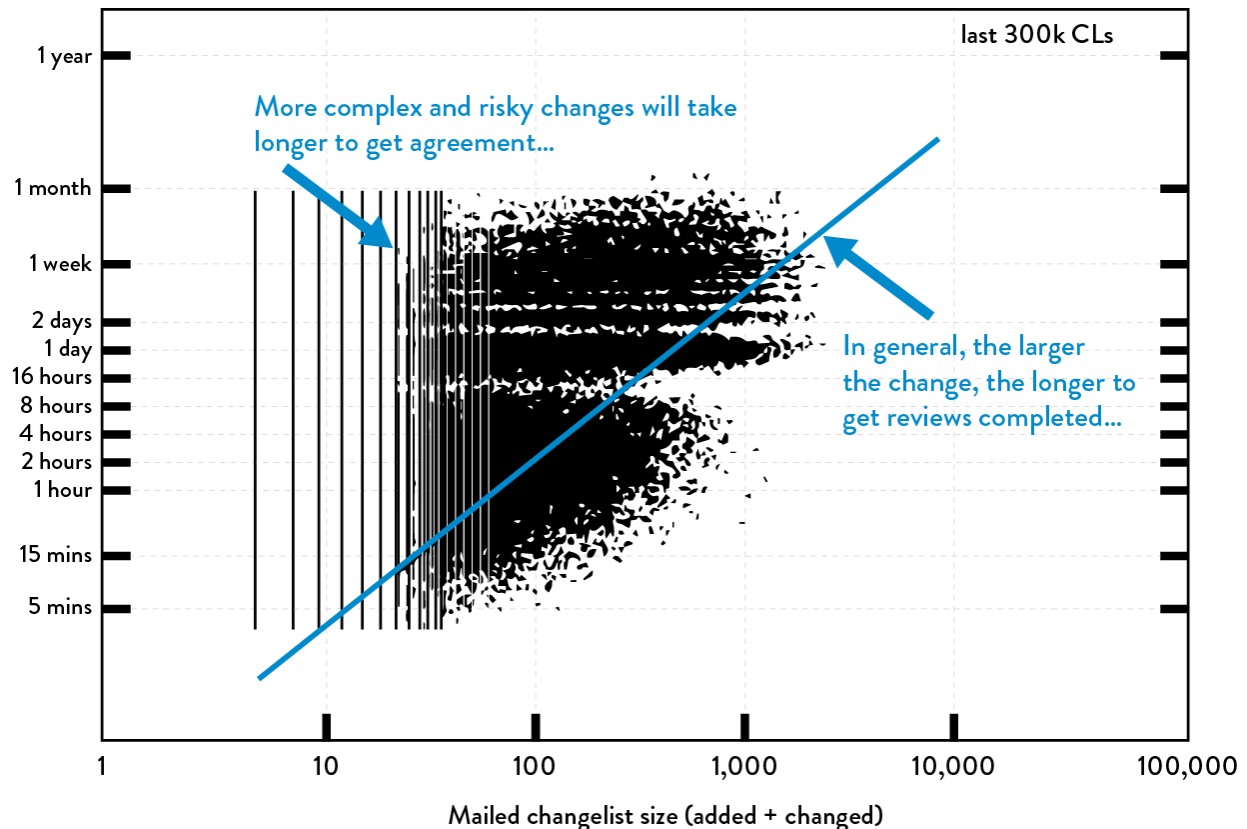


Figura 42: Tamanho da mudança vs. lead time para revisões no Google (Fonte: Ashish Kumar, "Development at the Speed and Scale of Google", apresentação na QCon, San Francisco, CA, 2010, https://qconsf.com/sf2010/dl/qcon-sanfran-2010/slides/AshishKumar_DevelopingProductsattheSpeedandScaleofGoogle.pdf.)

Enquanto trabalhava como diretor de engenharia do Google, Randy Shoup iniciou um projeto pessoal para resolver um problema técnico que a organização estava enfrentando. Ele disse: "Trabalhei nesse projeto por semanas e finalmente consegui pedir a um especialista no assunto para revisar meu código. Foram quase três mil linhas de código, que levaram dias de trabalho do revisor para passar. Ele me disse: 'Por favor, não faça isso comigo de novo'. Fiquei grato por esse engenheiro ter tido tempo para fazer isso. Foi também quando aprendi a fazer com que as revisões de código fizessem parte do meu trabalho diário."

Habilitação de API na Target

A Target é a sexta maior varejista dos EUA e gasta mais de US\$ 1 bilhão em tecnologia anualmente. Heather Mickman, diretora de desenvolvimento da Target, descreveu o início de sua jornada de DevOps: "Nos maus velhos tempos, costumávamos levar dez equipes diferentes para provisionar um servidor na Target e, quando as coisas quebravam, tendíamos a parar de fazer alterações para evitar mais problemas, o que, claro, piora tudo".

As dificuldades associadas à obtenção de ambientes e à execução de implantações criaram dificuldades significativas para as equipes de desenvolvimento, assim como o acesso aos dados de que precisavam. Como Mickman descreveu:

O problema era que muitos de nossos dados principais, como informações sobre estoque, preços e lojas, estavam bloqueados em sistemas legados e mainframes. Muitas vezes tínhamos múltiplas fontes de verdades de dados, especialmente entre o e-commerce e nossas lojas físicas, que eram de propriedade de equipes diferentes, com estruturas de dados diferentes e prioridades diferentes. O resultado foi que, se uma nova equipe de desenvolvimento quisesse criar algo para nossos hóspedes, levaria de três a seis meses para construir as integrações para obter os dados de que precisavam. Pior, levaria mais três a seis meses para fazer os testes manuais para garantir que eles não quebrassem nada crítico, por causa de quantas integrações ponto a ponto personalizadas tínhamos em um sistema muito bem acoplado. Ter que gerenciar as interações com as vinte a trinta equipes diferentes, juntamente com todas as suas dependências, exigiu muitos gerentes de projeto, por causa de toda a coordenação e transferências. Isso significava que o desenvolvimento estava gastando todo o seu tempo esperando em filas, em vez de entregar resultados e fazer as coisas.

Esse longo prazo para recuperar e criar dados em seus sistemas de registro estava comprometendo objetivos comerciais importantes, como a integração das operações da cadeia de suprimentos das lojas físicas da Target e de seu site de comércio eletrônico, que agora exigia levar o estoque para as lojas e casas dos clientes. Isso empurrou a cadeia de suprimentos da Target muito além do que foi projetada, que era apenas para facilitar a movimentação de mercadorias dos fornecedores para os centros de distribuição e lojas.

Em uma tentativa de resolver o problema de dados, em 2012 Mickman liderou a equipe de Habilitação de API para permitir que as equipes de desenvolvimento "entregassem novos recursos em dias, em vez de meses". Eles queriam que qualquer equipe de engenharia dentro da Target fosse capaz de obter e armazenar os dados de que precisavam, como informações sobre seus produtos ou suas lojas, incluindo horários de funcionamento, localização, se havia Starbucks no local e assim por diante.

As restrições de tempo desempenharam um grande papel na seleção da equipe. Mickman explicou que:

Como nossa equipe também precisava entregar recursos em dias, não meses, eu precisava de uma equipe que pudesse fazer o trabalho, não entregá-lo a empreiteiros – queríamos pessoas com habilidades de engenharia impressionantes, não pessoas que soubessem gerenciar

contratos. E para garantir que nosso trabalho não estivesse na fila, precisávamos possuir toda a pilha, o que significava que assumimos os requisitos de operações também.... Trouxemos muitas novas ferramentas para apoiar a integração contínua e a entrega contínua. E como sabíamos que, se conseguíssemos, teríamos que escalar com um crescimento extremamente alto, trouxemos novas ferramentas, como o banco de dados Cassandra e o corretor de mensagens Kafka. Quando pedimos permissão, nos disseram que não, mas fizemos mesmo assim, porque sabíamos que precisávamos.

Nos dois anos seguintes, a equipe de Habilitação de API habilitou cinquenta e três novos recursos de negócios, incluindo Ship to Store e Gift Registry, bem como suas integrações com o Instacart e o Pinterest. Como Mickman descreveu, "Trabalhar com o Pinterest de repente se tornou muito fácil, porque acabamos de fornecer nossas APIs".

Em 2014, a equipe de Ativação de API atendeu mais de 1,5 bilhão de chamadas de API por mês. Em 2015, isso cresceu para dezessete bilhões de chamadas por mês, abrangendo noventa APIs diferentes. Para dar suporte a esse recurso, eles rotineiramente realizavam oitenta implantações por semana.

Essas mudanças criaram grandes benefícios comerciais para a Target – as vendas digitais aumentaram 42% durante a temporada de férias de 2014 e aumentaram outros 32% no 2º trimestre. Durante o fim de semana da Black Friday de 2015, mais de 280 mil pedidos de retirada nas lojas foram criados. Até 2015, a meta é permitir que 450 de suas 1.800 lojas sejam capazes de atender pedidos de e-commerce, ante cem.

"A equipe de ativação de API mostra o que uma equipe de agentes de mudança apaixonados pode fazer", diz Mickman. "E isso nos ajuda a nos preparar para o próximo estágio, que é expandir o DevOps em toda a organização de tecnologia."

Sales Force

Estudo de caso Mudanças automatizadas de infraestrutura como alterações padrão na Salesforce.com (2012)

A Salesforce foi fundada em 2000 com o objetivo de tornar a gestão de relacionamento com o cliente facilmente disponível e entregue como um serviço. As ofertas da Salesforce foram amplamente adotadas pelo mercado, levando a um IPO bem-sucedido em 2004. Em 2007, a empresa tinha mais de cinquenta e nove mil clientes corporativos, processando centenas de milhões de transações por dia, com receita anual de US\$ 497 milhões.

No entanto, na mesma época, sua capacidade de desenvolver e lançar novas funcionalidades para seus clientes parecia parar. Em 2006, eles tiveram quatro grandes lançamentos de clientes, mas em 2007 eles só foram capazes de fazer um lançamento de cliente, apesar de ter contratado mais engenheiros. O resultado foi que o número de recursos entregues por equipe continuou diminuindo e os dias entre os principais lançamentos continuaram aumentando.

E como o tamanho do lote de cada lançamento continuou aumentando, os resultados da implantação também continuaram piorando. Karthik Rajan, então vice-presidente de Engenharia de Infraestrutura, relata em uma apresentação de 2013 que 2007 marcou "o último ano em que o software foi criado e enviado usando um processo em cascata e quando fizemos nossa mudança para um processo de entrega mais incremental".

No DevOps Enterprise Summit de 2014, Dave Mangot e Reena Mathew descreveram a transformação de DevOps de vários anos resultante que começou em 2009. De acordo com Mangot e Mathew, ao implementar princípios e práticas de DevOps, a empresa reduziu seus prazos de implantação de seis dias para cinco minutos até 2013. Como resultado, eles foram capazes de escalar a capacidade com mais facilidade, permitindo-lhes processar mais de um bilhão de transações por dia.

Um dos principais temas da transformação do Salesforce foi tornar a engenharia de qualidade o trabalho de todos, independentemente de fazerem parte do Desenvolvimento, Operações ou Infosec. Para fazer isso, eles integraram testes automatizados em todos os estágios da criação do aplicativo e do ambiente, bem como no processo contínuo de integração e implantação, e criaram a ferramenta de código aberto Rouster para conduzir testes funcionais de seus módulos Puppet. Eles também começaram a realizar rotineiramente *ensaios destrutivos*, um termo usado na fabricação para se referir à realização de testes de resistência prolongada sob as condições operacionais mais severas até que o componente que está sendo testado seja destruído. A equipe do Salesforce começou a testar rotineiramente seus serviços sob cargas cada vez mais altas até que o serviço quebrou, o que os ajudou a entender seus modos de falha e fazer as correções apropriadas. Sem surpresa, o resultado foi uma qualidade de serviço significativamente maior com cargas normais de produção.

A Segurança da Informação também trabalhou com a Engenharia da Qualidade nos estágios iniciais de seu projeto, colaborando continuamente em fases críticas, como arquitetura e projeto de teste, bem como integrando adequadamente ferramentas de segurança no processo de teste automatizado.

Para Mangot e Mathew, um dos principais sucessos de toda a repetibilidade e rigor que projetaram no processo foi ser informado por seu grupo de gerenciamento de mudanças que "as

mudanças de infraestrutura feitas por meio do Puppet agora seriam tratadas como 'mudanças padrão', exigindo muito menos ou até mesmo nenhuma aprovação adicional do CAB". Além disso, eles observaram que "mudanças manuais na infraestrutura ainda exigiriam aprovações". Ao fazer isso, eles não apenas integraram seus processos de DevOps com o processo de gerenciamento de mudanças, mas também criaram mais motivação para automatizar o processo de mudança para mais de sua infraestrutura.

Etsy

Ao contrário do Facebook, onde as implantações são gerenciadas por engenheiros de versão, no Etsy as implantações são realizadas por qualquer pessoa que queira executar uma implantação, como Desenvolvimento, Operações ou Infosec. O processo de implantação na Etsy se tornou tão seguro e rotineiro que novos engenheiros realizam uma implantação de produção em seu primeiro dia de trabalho!

Como Noah Sussman, arquiteto de testes da Etsy, escreveu: "Quando as 8h rolam em um dia útil normal, cerca de 15 pessoas estão começando a fazer fila, todos eles esperando implantar coletivamente até 25 conjuntos de mudanças antes que o dia termine".

Os engenheiros que desejam implantar seu código primeiro vão para uma sala de chat, onde os engenheiros se adicionam à fila de implantação, veem a atividade de implantação em andamento, veem quem mais está na fila, transmitem suas atividades e obtêm ajuda de outros engenheiros quando precisarem. Quando é a vez de um engenheiro implantar, ele é notificado na sala de chat.

O objetivo da Etsy tem sido tornar mais fácil e seguro a implantação em produção com o menor número de etapas e a menor quantidade de cerimônia. Provavelmente antes mesmo de o desenvolvedor verificar o código, eles executarão em sua estação de trabalho todos os 4.500 testes de unidade, o que leva menos de um minuto. Todas as chamadas para sistemas externos, como bancos de dados, foram canceladas.

Depois que eles verificam suas alterações no tronco no controle de versão, mais de sete mil testes de tronco automatizados são executados instantaneamente em seus servidores de integração contínua (CI). Sussman escreve: "Por tentativa e erro, estabelecemos cerca de 11 minutos como o mais longo que os testes automatizados podem executar durante um push. Isso deixa tempo para executar novamente os testes uma vez durante uma implantação [se alguém quebrar algo e precisar consertá-lo], sem ir muito além do limite de tempo de 20 minutos."

Se todos os testes fossem executados sequencialmente, Sussman afirma que "os 7.000 testes de tronco levariam cerca de meia hora para serem executados. Então, dividimos esses testes em subconjuntos e os distribuímos nas 10 máquinas em nosso cluster Jenkins [CI].... Dividir nosso conjunto de testes e executar muitos testes em paralelo nos dá o tempo de execução desejado de 11 minutos."

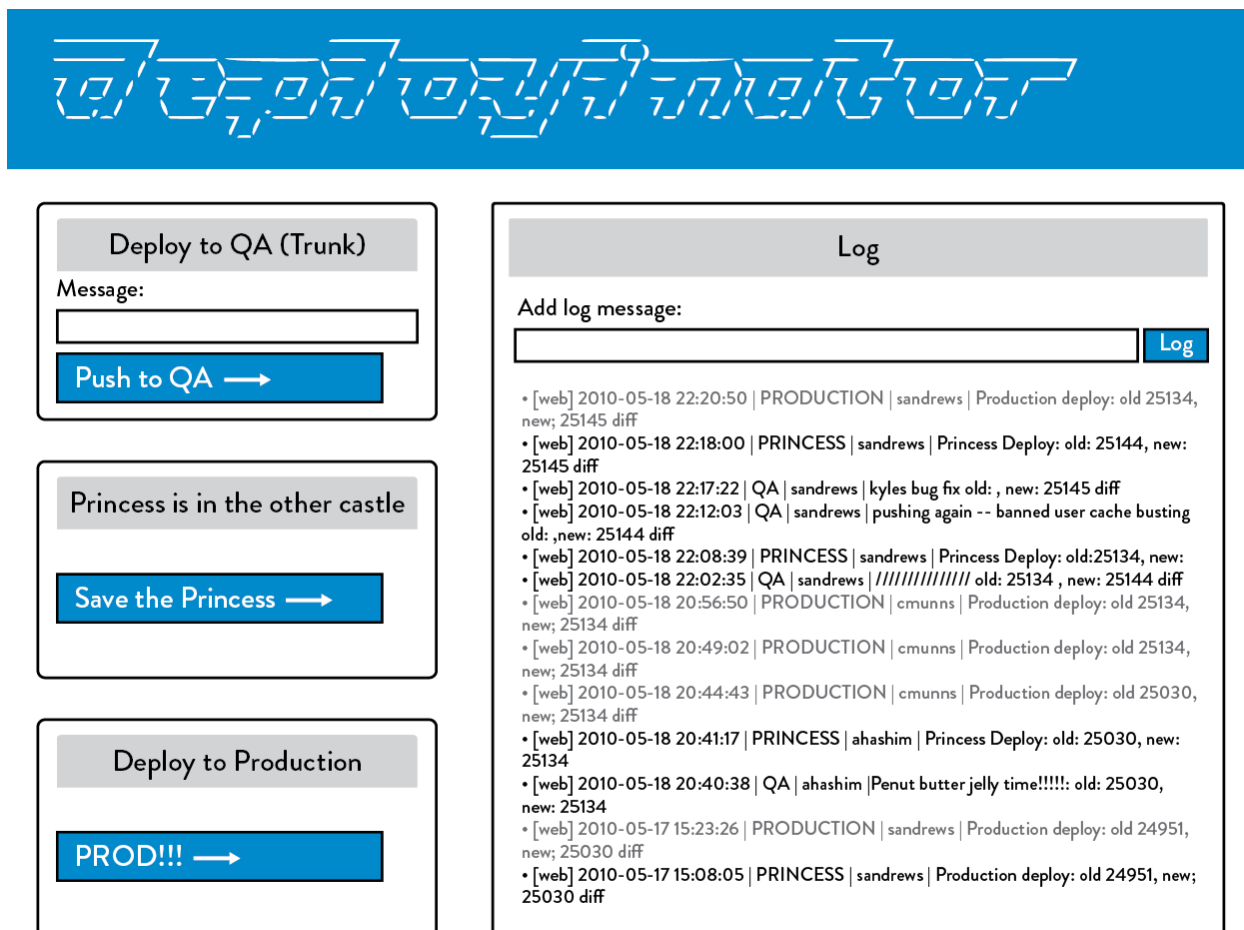


Figura 19: O console do Deployinator no Etsy (Fonte: Erik Kastner, "Quantum of Deployment", CodeasCraft.com, 20 de maio de 2010, <https://codeascraft.com/2010/05/20/quantum-of-deployment/>.)

Os próximos testes a serem executados são os *testes de fumaça*, que são testes de nível de sistema que executam cURL para executar casos de teste PHPUnit. Após esses testes, são executados os testes funcionais, que executam testes completos orientados por GUI em um servidor ativo — esse servidor é seu ambiente de controle de qualidade ou ambiente de preparo (apelidado de "Princess"), que na verdade é um servidor de produção que foi retirado da rotação, garantindo que ele corresponda exatamente ao ambiente de produção.

Quando é a vez de um engenheiro implantar, Erik Kastner escreve: "você vai ao Deployinator [uma ferramenta desenvolvida internamente, veja a figura 19] e aperta o botão para colocá-lo no controle de qualidade. De lá, visita a princesa.... Então, quando estiver pronto para entrar no ar, você aperta o botão "Prod" e logo seu código está ao vivo, e todos no IRC [canal de bate-papo] sabem quem empurrou qual código, completo com um link para o diff. Para quem não está no IRC, há o e-mail que todos recebem com a mesma informação."

Em 2009, o processo de implantação do Etsy foi motivo de estresse e medo. Em 2011, tornou-se uma operação de rotina, acontecendo de vinte e cinco a cinquenta vezes por dia, ajudando os engenheiros a colocar seu código rapidamente em produção, entregando valor aos seus clientes.