

GLOSSÁRIO

Para um melhor entendimento do projeto, é importante conceituar alguns termos, metodologias e frameworks de TI adotados para elaboração deste projeto.

Ambiente auto escalável: a característica desejável em todo o sistema, rede ou processo, que indica a capacidade de manipular uma porção crescente de trabalho de forma uniforme, ou estar preparado para crescer. Por exemplo, isto pode-se referir à capacidade de um sistema suportar um aumento de carga total quando os recursos (normalmente do hardware) assim requisitam.

Ambiente em Nuvem: é um termo para a disponibilidade sob demanda de recursos do sistema de computador, especialmente armazenamento de dados e capacidade de computação, sem o gerenciamento ativo direto do utilizador.

Análise de Causa Raiz: processo de descobrir a causa raiz de problemas para identificar as soluções adequadas.

Análise Estática de Código: método de depuração, examinando o código-fonte antes da execução de um programa. Isso é feito analisando um conjunto de códigos em relação a um conjunto (ou vários conjuntos) de regras de codificação. Ela verifica a qualidade do código-fonte. Essa verificação é realizada antes mesmo que haja execução do software.

API REST: também chamada de API RESTful, é uma interface de programação de aplicações que segue conformidade com as restrições da arquitetura REST. A sigla REST significa Representational State Transfer (Transferência Representacional de Estado, em português).

API: A interface de programação de aplicações (API) é um conjunto de ferramentas, definições e protocolos usados para integrar serviços e aplicações. É por meio dela que suas soluções e serviços podem se comunicar com outras soluções e serviços, sem a necessidade de criar constantemente uma infraestrutura de conectividade.

Backlog: é uma lista contendo todas as funcionalidades desejadas para um produto. O backlog não precisa estar completo no início de um projeto. Pode-se começar com tudo aquilo que é mais óbvio em um primeiro momento. Com o tempo, o backlog cresce e muda à medida que se aprende mais sobre o produto e seus usuários

Behavior Driven Development: BDD ou ainda uma tradução (Desenvolvimento Guiado por Comportamento) é uma técnica de desenvolvimento Ágil que encoraja colaboração entre desenvolvedores, setores de qualidade e pessoas não-técnicas ou de negócios num projeto de software, relaciona-se com o conceito de verificação e validação. Foi originalmente concebido em 2003, por Dan North [1] como uma resposta à Test Driven Development (Desenvolvimento Guiado por Testes), e tem se expandido bastante nos últimos anos. Os focos do BDD são a linguagem e as interações usadas no processo de desenvolvimento de software. Desenvolvedores usam sua língua nativa em combinação com a linguagem ubíqua (ubiquitous language), que lhes permite concentrar nas razões pelas quais o código deve ser criado, e não em detalhes técnicos, além de minimizar traduções entre a linguagem técnica na qual o código é escrito e outras linguagens de domínio, usuários, clientes, gerência do projeto etc.

Blue-Green: Técnica que tem como objetivo permitir que o lançamento de uma nova versão seja feito de forma mais eficiente e reduzindo riscos, todo o tráfego é direcionado para uma versão, que passa a ser o ambiente de utilização, enquanto a outra versão passa a servir como ambiente de desenvolvimento.

Build: é uma versão "compilada" de um software ou parte dele que contém um conjunto de recursos que poderão integrar o produto final.

Canary Release: Técnica desenvolvida para reduzir os riscos da introdução de uma nova versão de software no ambiente de produção, a técnica de Canary Release pode ser resumida pela liberação gradual das mudanças para um pequeno subgrupo de usuários, antes que as modificações sejam aplicadas em toda a plataforma ou infraestrutura.

Casos de Testes: é um conjunto de condições usadas para teste de software. Ele pode ser elaborado para identificar defeitos na estrutura interna do software por meio de situações que exercitem adequadamente todas as estruturas utilizadas na codificação; ou ainda, garantir que os requisitos do software que foi construído sejam plenamente atendidos.

Casos de Uso: são narrativas em texto, descrevendo a unidade funcional, e são amplamente utilizados para representar requisitos funcionais nos sistemas

Cerimônias do Scrum: encontros responsáveis por garantir a transparência (geralmente através da comunicação, mas também envolvendo artefatos) e proporcionar momentos de inspeção e adaptação tal qual pregam os pilares do Scrum. Estas cerimônias são: a Sprint Planning, a Daily Meeting, a Sprint Review e a Sprint Retrospective.

Codificação (Code): segue padrões e um processo de desenvolvimento de software para entregas parciais ou integrais de um sistema conforme os requisitos funcionais definidos na fase de planejamento.

Containers: são tecnologias que permitem empacotar e isolar aplicações com todo o ambiente de execução delas, ou seja, com todos os arquivos necessários para executá-las. Dessa forma, fica mais fácil migrá-las de um ambiente para outro (desenvolvimento, teste, produção etc.) sem perder funcionalidades.

Definição de Pronto: é um artefato do Scrum que reforça a transparência como pilar principal da metodologia uma vez que torna global entre os membros do time o que DONE significa. A DoD define um vocabulário comum, que indica que quando um item está DONE, ele de fato está pronto tecnicamente para ir para produção.

Deploy: são tecnologias que permitem empacotar e isolar aplicações com todo o ambiente de execução delas, ou seja, com todos os arquivos necessários para executá-las. Dessa forma, fica mais fácil migrá-las de um ambiente para outro (desenvolvimento, teste, produção etc.) sem perder funcionalidades.

Dívida técnica: O termo foi definido por Ward Cunningham, programador americano que desenvolveu a primeira ferramenta wiki, e descreve a dívida que a equipe responsável assume quando escolhe um design ou abordagem fácil de implementar no curto prazo, mas com grande impacto negativo no longo prazo. Em algum momento, o efeito será exposto e a dívida técnica precisará ser tratada. No entanto, resolver o problema geralmente será mais complexo e caro, dadas as alterações e a evolução em que o sistema passou desde que o risco foi incorrido.

Docker: é um conjunto de produtos de plataforma como serviço (PaaS) que usam virtualização de nível de sistema operacional para entregar software em pacotes chamados contêineres. Os contêineres são isolados uns dos outros e agrupam seus próprios softwares, bibliotecas e arquivos de configuração. Eles podem se comunicar uns com os outros por meio de canais bem definidos. Todos os contêineres são executados por um único kernel do sistema operacional e, portanto, usam menos recursos do que as máquinas virtuais.

Endomarketing: é uma estratégia de marketing institucional voltada para ações internas na empresa. É também chamado de Marketing Interno e visa melhorar imagem da empresa entre os seus colaboradores, culminando em uma equipe motivada e reduzindo o turnover

Entrega contínua: entrega contínua é a capacidade de disponibilizar mudanças de forma segura e rápida garantindo que o código esteja sempre pronto para implantação, mesmo diante de diversos desenvolvedores fazendo alterações diariamente. Visa a automatização do processo de aprovação e liberação de forma íntegra, segura e com qualidade controlada pela integração contínua.
experiência do usuário

Feature Toggle: Técnica usada para permitir mudanças de comportamento de um sistema sem necessidade de um novo deployment. Neste contexto, podemos entender por mudança de comportamento uma adição/alteração/remoção de uma funcionalidade, que pode ser aplicada desde uma UI até um serviço backend (API, banco de dados etc.).

Feedback contínuo: a prática do feedback do usuário possibilita que o time possa melhorar o ambiente e as funcionalidades do software, e que a área de negócios possa rever os requisitos com foco na experiência do usuário. A participação de todos os envolvidos no processo deve ser incentivada, tanto do ponto de vista do usuário quanto do gestor do negócio. Dessa forma, as expectativas são gerenciadas da melhor forma agregando valor às funcionalidades desenvolvidas, garantindo agilidade e minimizando os riscos mediante avaliação por todos os envolvidos.

Ferramentas Open Source: ferramentas desenvolvidas de forma descentralizada e colaborativa e contam com a revisão e a produção pela comunidade. Costumam ser mais baratos, mais flexíveis e mais duradouros do que as opções proprietárias, já que são desenvolvidos por comunidades, e não por um único autor ou empresa.

Framework: framework é um template com diversas funções que podem ser usadas pelo desenvolvedor. Com ele, é desnecessário gastar tempo para reproduzir a mesma função em diferentes projetos, auxiliando em um gerenciamento ágil de projetos. Em outras palavras, ele é uma estrutura base, uma plataforma de desenvolvimento, como uma espécie de arcabouço. Ele contém ferramentas, guias, sistemas e componentes que agilizam o processo de desenvolvimento de soluções, auxiliando os especialistas de TI em seus trabalhos.

GIT: é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo.

Histórias de Usuários: Uma história do usuário é uma explicação informal e geral sobre um recurso de software escrito a partir da perspectiva do usuário final. Seu objetivo é articular como um recurso de software pode gerar valor para o cliente.

Infraestrutura como código: o objetivo é provisionar rapidamente a infraestrutura, e toda a atividade manual das equipes passa a ser realizada automaticamente com testes para garantia dos ambientes. A utilização de um repositório único, ferramentas

comuns e scripts de comandos que gerenciam a configuração, implantando o software no ambiente de produção sem causar impacto nas atividades diárias.

Integração contínua: é o processo de construção do software, integrando as etapas de codificação, construção e testes de todos os componentes para tornar a entrega um produto funcional automatizado. A cada mudança no software, toda a aplicação é compilada, são realizados os testes automatizados e, em caso de falha em alguma etapa do processo, o time é acionado para resolver o problema imediatamente. A grande vantagem é a identificação e correção dos erros de forma rápida.

Kubernetes: é um sistema de orquestração de contêineres open-source que automatiza a implantação, o dimensionamento e a gestão de aplicações em contêineres. Ele foi originalmente projetado pelo Google e agora é mantido pela Cloud Native.

Lean: trata-se de um método que institui o uso de nada além do que os recursos necessários para a realização de um determinado trabalho, etapa ou processo, evitando desperdícios. Um dos grandes benefícios é que essa metodologia, aplicada à gestão de negócios, minimiza desperdícios em diversas frentes. Seguindo os princípios lean, líderes e gestores são capazes de aproveitar melhor o tempo dos colaboradores e aumentar a produtividade de suas equipes tomando algumas ações como, por exemplo: identificar e eliminar constantemente problemas que travam os processos; garantir que o fluxo de trabalho esteja claro para todos, eliminando ruídos na comunicação; indicar sempre quais demandas e entregas são prioritárias e fornecer o suporte necessário para que a equipe possa efetuar entregas contínuas e de qualidade. A metodologia lean está associada ao Manifesto Ágil, que também prevê uma série de práticas para enxugar e dar mais leveza aos processos de desenvolvimento

Logs: uma expressão utilizada para descrever o processo de registro de eventos relevantes num sistema computacional. Esse registro pode ser utilizado para restabelecer o estado original de um sistema ou para que um administrador conheça

o seu comportamento no passado. Um arquivo de log pode ser utilizado para auditoria e diagnóstico de problemas em sistemas computacionais.

Máquina Virtual: Uma máquina virtual é um arquivo de computador (normalmente chamado de imagem) que se comporta como um computador de verdade. Em outras palavras, é a criação de um computador dentro de um computador.

Melhoria Contínua: prática adotada por diversas empresas que visa atingir, ininterruptamente, resultados cada vez melhores – sejam eles nos produtos e serviços da empresa, ou então em seus processos internos. A melhoria contínua pode ser atingida através de diversas metodologias e boas práticas organizacionais.

Métodos Ágeis: os métodos ágeis surgiram para sanar os problemas de engenharia de software onde os projetos eram muitos complexos e com longo prazo de desenvolvimento. A metodologia busca promover um processo de gerenciamento de projetos que incentiva a inspeção e adaptação frequente. É uma filosofia que acaba por incentivar o maior trabalho em equipe, a auto-organização, a comunicação frequente, o foco no cliente e a entrega de valor. Basicamente, os métodos ágeis são um conjunto de práticas eficazes que se destinam a permitir a entrega rápida e de alta qualidade do produto, tendo uma abordagem de negócios que alinha o desenvolvimento do projeto com as necessidades do cliente e os objetivos da empresa. Os quatro princípios básicos do manifesto ágil mostram o que se espera de qualquer método de desenvolvimento desta categoria: indivíduos e interações sobre processos e ferramentas; software funcionando sobre documentação extensiva; colaboração com o cliente sobre negociação de contrato; responder às mudanças sobre seguir um planejamento.

MGM3.0: O Management 3.0. foi desenvolvido para as empresas de TI, no entanto, outros segmentos estão adotando o conceito como uma nova abordagem de liderar equipes objetivando o treino de gerentes e líderes para promover organizações mais ágeis e colaboradores mais produtivos. É formado por 6 visões: energizar as pessoas; empoderar times; alinhar restrições; desenvolver competências; crescer a estrutura e aprimorar para melhorar sempre.

Mindset Ágil: um mindset ágil é o conjunto de atitudes que sustentam um ambiente de trabalho ágil. Estas incluem respeito, colaboração, melhoria e ciclos de aprendizagem, orgulho na propriedade, foco na entrega de valor e capacidade de adaptação à mudança. Este mindset é necessário para cultivar equipes de alta performance, que, por sua vez, oferecem um valor incrível a seus clientes.

Monitoração contínua: faz uso de métricas em todas as etapas envolvidas na entrega do software, para identificar a progressão do time, eventuais gargalos no processo e problemas de performance. Deve ser utilizada uma linguagem comum para as métricas que serão utilizadas pelos times. Para a monitoração da aplicação devem ser desenvolvidas funcionalidades de para criação de gatilhos a serem acionados automaticamente para atingir diversos objetivos. Existem alguns exemplos de métricas na telemetria que cobrem: a lógica do negócio, o aplicativo e o sistema operacional. Esses componentes geram eventos, registros e métricas que são capturados pelo roteador de eventos para a geração de informação que será utilizada na inspeção e investigação de logs, geração de gráficos e lançamentos de alertas configuráveis.

Obeya: é uma prática Lean de gerenciamento visual. É um lugar de colaboração intensiva e procedimentada. As informações estão disponíveis e são gerenciadas coletivamente através de reuniões curtas e periódicas. É uma ferramenta simples e eficaz para enfrentar os desafios de desempenho.

Openshift: é um produto de software de computador da Red Hat para implantação e gerenciamento de softwares baseados em containers. Ele é uma distribuição suportada do Kubernetes usando Docker e ferramentas DevOps para desenvolvimento acelerado de aplicações.

Orquestração: é a configuração, o gerenciamento e a coordenação automatizada de serviços, aplicações e sistemas de computador. A orquestração ajuda a TI a gerenciar fluxos de trabalho e tarefas complexas com mais facilidade.

PDCA: é um método mundialmente reconhecido como uma ferramenta de melhoria contínua composta pelas seguintes etapas: P (do inglês – Plan) = Planejamento; D (do inglês – Do) = Execução; C (do inglês – Check) = Verificação; A (do inglês – Act) = Atuar/Agir.

Pessoas e cultura: mais do que um conceito, é importante destacar que DevOps é uma jornada de aproximação entre as pessoas com ações práticas de automação para acelerar as implantações com qualidade, considerando o ponto de vista de todos os envolvidos, a tão falada empatia. A dimensão mais importante está relacionada a pessoas e à mudança de cultura da empresa, podendo ser necessária uma reavaliação de sua estrutura organizacional. É importante uma maior valorização de profissionais que atuem em várias áreas.

Pipeline de Implantação: o pipeline de Implantação é uma manifestação automatizada do seu processo para levar o software do controle de versão para as mãos de seus usuários.

Planejamento contínuo (Plan): compreende o gerenciamento do projeto, dos requisitos, prazos e prioridades. Trata da distribuição de atividades e entregas do software, com foco no estabelecimento de metas e nos ajustes constantes, visando responder de forma ágil aos feedbacks dos clientes no decorrer do desenvolvimento do software.

Planning Poker: técnica baseada no consenso para estimar, é um jogo e ao mesmo tempo um exercício. Através dessa técnica podemos estimar o esforço necessário para determinada quantidade de trabalho, tendo como base informações recolhidas do cliente, normalmente sendo, através de histórias de usuário.

Product Owner: é o representante máximo dos desejos do cliente dentro do Time Scrum. É ele que conhece o problema, as "dores", o retorno esperado pelo projeto e ele é o papel responsável por usar toda sua expertise para guiar o Time de Desenvolvimento para que o seu trabalho seja o mais otimizado possível, do ponto de vista de ROI (Return Over Investment - Retorno Sobre Investimento). É um membro

full-time do Time Scrum, estando sempre à disposição dos desenvolvedores para resolver todo tipo de impedimento de negócio, dúvidas de priorização de atividades e principalmente, para trabalhar constantemente no refinamento do Backlog de Produto e nos seus desdobramentos como roadmaps e releases.

Push: atualização do repositório remoto com o conteúdo do repositório local.

Refatoração: é o processo de modificar um sistema de software para melhorar a estrutura interna do código sem alterar seu comportamento externo.

Release: são versões de um software. Quando estamos desenvolvendo uma aplicação, é comum que a disponibilizamos entre versões. Essas versões são chamadas de Releases.

Repositórios: é que uma pasta onde estão todos os arquivos do nosso projeto, incluindo nosso versionamento.

Requisitos Funcionais: são as funções que o sistema/aplicativo deve possuir para atender o negócio e suas regras.

Reuniões de Retrospectiva: é uma cerimônia que deve acontecer sempre no último dia da sprint, para "fechar" a mesma. É uma cerimônia com duração máxima de três horas para cada trinta dias de sprint e é facilitada pelo Scrum Master. Nesta cerimônia que todo o Time Scrum deve participar, o objetivo é analisar tudo o que aconteceu nesta sprint que está encerrando e propor melhorias para a próxima.

Rollback: encerramento da transação desfazendo todas as alterações (updates, deletes, inserts) realizadas durante a transação.

Sequência Fibonacci: É uma sucessão de números que, misteriosamente, aparece em muitos fenômenos da natureza. Descrita no final do século 12 pelo italiano Leonardo Fibonacci, ela é infinita e começa com 0 e 1. Os números seguintes são

sempre a soma dos dois números anteriores. Portanto, depois de 0 e 1, vêm 1, 2, 3, 5, 8, 13, 21, 34...

Sprint: etapas de um projeto. Toda sprint tem uma meta a ser atingida e consiste no desenvolvimento e na entrega de uma versão utilizável do produto ao cliente. Elas têm um tempo de duração predeterminado e geralmente são curtas, não excedendo o período de um mês — o que diminui os riscos e a propagação de erros. Além disso, as sprints ocorrem de forma iterativa, ou seja, assim que uma delas termina, uma nova é iniciada, de forma que o projeto vá sempre progredindo em direção à sua versão ideal.

Telemetria: A telemetria é uma forma avançada de monitoração de aplicações em ambiente de produção. Ela permite conhecer mais profundamente os padrões de uso de uma aplicação, variações de carga, acesso, entre outras questões. A Telemetria conta também com um mecanismo intrínseco de capacidade de análise de uso (analytics) que permite aos times conhecer os padrões de acesso e assim evoluir o produto tecnicamente e do ponto de vista de negócio com mais sabedoria.

Test Driven Development: ou TDD é uma técnica de desenvolvimento de software que se relaciona com o conceito de verificação e validação e se baseia em um ciclo curto de repetições: Primeiramente o desenvolvedor escreve um caso de teste automatizado que define uma melhoria desejada ou uma nova funcionalidade. Então, é produzido código que possa ser validado pelo teste para posteriormente o código ser refatorado para um código sob padrões aceitáveis.

Teste de Carga: Testa o software sob as condições normais de uso. Ex.: tempo de resposta, número de transações por minuto, usuários simultâneos etc.

Teste de Integração: Testa se um ou mais componentes combinados funcionam de maneira satisfatória. Há quem diga que o teste de integração é composto por vários testes de unidade.

Teste de Regressão: Reteste de um sistema ou componente para verificar se alguma modificação recente causou algum efeito indesejado, além de, certificar se o sistema ainda atende os requisitos.

Teste de Stress: Testa o software sob condições extremas de uso. Grande volume de transações e usuários simultâneos. Picos excessivos de carga em curtos períodos.

Teste de Usabilidade: Teste focado na experiência do usuário, consistência da interface, layout, acesso às funcionalidades etc.

Teste de Vulnerabilidade: é identificar se existem vulnerabilidades exploráveis conhecidas na camada da aplicação web, o que normalmente é feito internamente, sobre o servidor da aplicação web no ambiente de desenvolvimento ou de produção. testes automatizados

Testes Contínuos: compreende os testes unitários, de integração, de aceitação, de desempenho, de vulnerabilidade e qualidade de código, visando promover o correto funcionamento da aplicação. Os testes devem ser realizados antes e continuamente, durante o desenvolvimento, proporcionando redução de custos e tempo de entrega, além de elevar a qualidade do código.

Testes Exploratórios: normalmente o testador não tem informações detalhadas sobre o que vai testar e como vai testar. O testador se baseia na sua experiência, assim como no conhecimento que ele vai adquirindo sobre o aplicativo durante a execução do teste exploratório. A partir dessa perspectiva, podemos afirmar que o teste exploratório é uma atividade iterativa e empírica de exploração que exige idas e vindas num processo de investigação contínuo onde a intuição, a criatividade e a experiência do testador são indispensáveis para garantir a eficiência do teste

Testes Funcionais: Testa os requisitos funcionais, as funções e os casos de uso.

Testes Unitários: Testa um componente isolado ou classe do sistema

Time-box: A técnica de Timebox no Scrum, consiste em definir o tempo máximo para atingir as metas, tomar uma decisão ou executar um conjunto de tarefas. A ideia é maximizar o tempo fazendo o melhor que a equipe pode neste intervalo. Assim, em vez de começar a trabalhar em algo até sua finalização de forma indefinida, é acordado de antemão o tempo limite para cada tarefa de projeto.

Yaml: acrônimo recursivo para “YAML Ain’t Markup Language” é um formato de serialização de dados legível por humanos, sendo bastante utilizado para arquivos de configuração, assim como o JSON e o XML. Possui uma sintaxe simples e legível, podendo ser mapeada facilmente pelos tipos de dados mais comuns na maioria das linguagens de alto nível, sendo criada especificamente para funcionar bem em casos de uso comum, como, por exemplo: arquivos de configuração; compartilhamento de dados entre linguagens; depuração de estruturas de dados complexas e mensagens entre processos, entre outros.

APÊNDICE A - Framework Calms

1. CALMS

O CALMS é um acrônimo para Culture (Cultura), Automation (Automação), Lean (Metodologia Lean), Measurement (Mensuração) e Sharing (Compartilhamento).

Quando foi criado, sua ideia era quebrar gargalos existentes entre os setores de Desenvolvimento e Operações, um dos principais pilares do DevOps. Permite a análise da conjuntura atual, o apontamento do que deve ser feito e as diretrizes para uma implementação segura, a fim de conseguir os melhores resultados.

- **Culture (Cultura):** valorização das pessoas, comunicação interpessoal eficiente e rápida, colaboração entre os times e um mindset ágil para contribuir com os objetivos comuns da organização. Representa o relacionamento entre pessoas, processos e ferramentas que formam a cultura organizacional de desenvolvimento.
- **Automation (Automação):** promove a automação e padroniza o fluxo de trabalho “pipeline” dando agilidade com qualidade e segurança em todas as entregas. O foco é automatizar tarefas internas, principalmente as etapas do fluxo de criação, minimizando erros humanos que podem causar problemas e prejuízos para seu negócio. A automação também realiza a documentação de cada processo, auxiliando na padronização dos passos.
- **Lean (Metodologia Lean):** relacionado a melhoria no desempenho dos processos da empresa, eliminando desperdícios, com foco no valor e no cliente, entregas menores e contínuas, aumentar a aprendizagem e redução de custos.
- **Measurement (Mensuração):** a mensuração gera conhecimento, bem como a previsibilidade sobre problemas imprevistos e incidentes que possam ocorrer. Dessa forma os gestores podem se prevenir evitando falhas e reduzindo prejuízos com a utilização de logs e dashboard. Essa mensuração deve ser feita em todos os níveis, inclusive nas regras de negócio. Alguns de seus pilares

são: meçaure tudo o que for possível, adote a melhoria contínua, adote o monitoramento e controle e realize análise de desempenho constantemente.

- **Sharing (Compartilhamento):** compartilhar é essencial para melhorar o fluxo de comunicação e o trabalho em equipe. O feedback deve ser estabelecido em todos os estágios de fluxo de valor. Alguns dos pilares dessa etapa são: cultura do compartilhamento, adesão de maior colaboração entre os membros, cultura da transparência e feedback contínuo para os times.

APÊNDICE B - Feedback das Reuniões

1. Reunião Café Com DevOps 1 - Apresentação



2. Reunião Café com DevOps 2 - Revisão do Processo de Desenvolvimento





3. Café com DevOps 3 - Avaliação do nível de maturidade DevOps



4. Qualidade de Software



<p>Teste Funcional com pelo menos o caminho feliz automatizado</p> <p>Agência</p>	<p>1ª evolução, seria um teste unitário de boa qualidade, sonar e vulnerabilidade. Acréscimos de bom tamanho</p> <p>Comércio</p>	<p>O problema do sonar é que toda semana sai uma versão nova e os indicadores mudam. Como não usamos internamente não conseguimos estes resultados.</p> <p>Fora do Trabalho</p>	<p>2ª evolução (do pipeline), Testes Funcionais e desempenho.</p> <p>Comércio</p>	<p>Antes do Teste de Vulnerabilidade tem que ter restore do bd. Pode ser um passo do pipeline</p> <p>Comércio</p>
<p>Hoje não temos nem a progressão de um sistema. Não temos métrica nenhuma.</p> <p>Desenvolvimento</p>	<p>Sonar, são avaliados 3 parâmetros: linhas de código, complexidade de classe e cobertura de testes. Existe um quarto indicador que é uma média ponderada destes 3, mas nunca na última versão</p> <p>Comércio</p>	<p>Nós não temos costume de ter métricas</p> <p>Desenvolvimento</p>	<p>Há muitos anos, eu tento medir o trabalho da qualidade e não consigo</p> <p>Fora do Trabalho</p>	<p>Deveríamos ter 3 pipelines: java, .net, maker.</p> <p>Agência</p>
<p>Se todos os nossos sistemas tivessem testes unitários, não teríamos problemas</p> <p>Agência</p>	<p>teste unitário de boa qualidade</p> <p>Comércio</p>	<p>É importante ter uma regra de todos os projetos não só seguir o pipeline com testes, para a DDD ter uma ideia por conta de todas as demandas em paralelo</p> <p>Fora do Trabalho</p>	<p>Não tem como avaliar um teste de unitário de boa qualidade</p> <p>Comércio</p>	<p>Os indicadores (do sonar) são avaliados pelo Terson</p> <p>Comércio</p>
<p>Time java está sempre na vanguarda das coisas. O movimento na planta é sempre de baixo para cima, nunca de cima para baixo.</p> <p>Agência</p>	<p>A Direção não propaga esses comportamentos, acabam deixando criar essas aldeias/quartais de desenvolvimento.</p> <p>Agência</p>	<p>Internamente não usamos o sonar. Cobramos as fábricas mas não usamos internamente</p> <p>Fora do Trabalho</p>	<p>Atividades que deveriam ter no build, análise estática, teste unitário, apis e testes funcionais no pipeline</p> <p>Desenvolvimento Agência</p>	<p>A qualidade nunca usou o sonar, quem faz é marcia e wolosker</p> <p>Comércio</p>
<p>Quando fazemos um sistema, desenvolvemos internamente, ele entra em produção e apresenta erro. Du tempo que se vê que tem de erro e esse, no seu erro de qualidade? de regra de negócio? de código?</p> <p>Desenvolvimento</p>	<p>Não temos controle dos defeitos, se algum que acontece, foi regra de negócio que não foi mapeada, qualidade não pegou ou foi o desenvolvimento</p> <p>Desenvolvimento</p>	<p>Análise de cobertura tem um baixo no desenvolvimento, já que não temos ainda este nível de maturidade, mas ao lançamento de release, podem ser corrigidos</p> <p>Comércio</p>	<p>Como fugimos de regras que violam as regras, como de regras de programação e erro ou estamos criando uma versão com erro de gramática? Definimos que se for para não e agora já temos todas as regras corrigidas</p> <p>Fora do Trabalho</p>	

5. Arquitetura de Soluções Java

<p>Os líderes das outras plataformas estão no mesmo conhecimento. Estão acostumado com aquilo e não querem sair</p> <p>Para os DevOps Comentários</p>	<p>Existem essa questão que não devemos incluir. O desenvolvedor precisa conhecer as ferramentas e estudar a realidade que vai pagar na mão e levar para</p> <p>Comentários</p>	<p>Às vezes eu sinto que os líderes do projeto ficam perdidos, muitas vezes por falta de comunicação da gente mesmo</p> <p>Comentários</p>	<p>A reunião que participamos arquitetos de outras tecnologias serve para um brainstorm, o trabalho é esse, mas não é para segurar na mão e levar</p> <p>Comentários</p>	<p>Nós sabemos o que temos que fazer (melhorias no pipeline), não fazemos porque não temos tempo.</p> <p>Comentários</p>
<p>Infelizmente é uma coisa rara na liderança, um perfil (autodidata) que corre atrás.</p> <p>Para os DevOps Comentários</p>	<p>Se tivéssemos uma equipe DevOps (para equipe DevOps) não seria necessário uma pessoa com o mais perfil ou de dentro da DS?</p> <p>Para os DevOps Comentários</p>	<p>Os nossos analistas de requisitos precisam evoluir, precisam de treinamento</p> <p>Comentários</p>	<p>Precisa ter uma reunião geral no perfil de análise de requisitos. Não tem mais como deixar de fazer, do que para fazer análise de requisitos, mas tem que ter um líder de projeto, precisa estar a um passo mais na direção técnica, confiamos em pessoas com o conhecimento de pipeline para ter uma outra visão na liderança de projetos</p> <p>Comentários</p>	<p>O analista de requisito tem que evoluir para a liderança do projeto</p> <p>Comentários</p>
<p>o nosso (pipeline) seria um pontapé inicial e eles (os outros arquitetos) adaptariam para a realidade deles.</p> <p>Comentários Comentários</p>	<p>O que vamos fazer na nossa equipe? Temos que melhorar monitoração. Mas, quem vai instalar o prometheus?</p> <p>Comentários Comentários</p>	<p>Não temos nada do processo do time documentado</p> <p>Comentários</p>	<p>Esquecer o pessoal, vai que alguém se empolga e resolve ajudar. Temos um problema de comunicação de mudança de valores as pessoas não sabem que existe essa gerência.</p> <p>Comentários</p>	<p>quando vamos ter um projeto, é desenvolver já tem que ter as coisas basicamente prontas, para se conectar a desenvolver aquele produto. O pipeline tem que estar criado para aquela plataforma.</p> <p>Comentários</p>
<p>o legado sempre vai existir e ele provavelmente não será migrado</p> <p>Para os DevOps</p>	<p>Pipeline é aplicado a linguagem, mas, pra cada poderia ter um pipeline em uma imagem docker, é só um questão de cada líder de cada plataforma seguir e levar</p> <p>Para os DevOps</p>	<p>A liderança não define o padrão e não treina os devs</p> <p>Para os DevOps</p>	<p>enquanto a prioridade for sempre precedida, a gente nunca vai conseguir priorizar nada</p> <p>Para os DevOps</p>	<p>Temos que pensar em uma maneira da gente evoluir e a fábrica evoluir junto</p> <p>Para os DevOps</p>
<p>isto está totalmente ligado a uma outra direção que trabalhamos, a DS. Então, quem tem a sua outra voltada especificamente para desenvolver software de apoio ao Desenvolvedor.</p> <p>Para os DevOps</p>	<p>Hoje as vezes há de encontro com a própria DS, porque as vezes é mais importante fazer um projeto para o cliente do que fazer um projeto que não é projeto para os desenvolvedores.</p> <p>Para os DevOps</p>	<p>devia ter uma equipe que tivesse produtos para dar suporte para o desenvolvedor, isso não está na DS e nem na DOP. Está lá não dá pra</p> <p>Para os DevOps Comentários</p>	<p>essa equipe (equipe devops) deveria ter autonomia, múltipla funcional, com pessoas com conhecimentos diferenciados</p> <p>Para os DevOps</p>	<p>devops "sangra", porque mostra todos os problemas da empresa</p> <p>Comentários</p>
<p>não sei como as empresas estão fazendo para sair dessa forma segregada para uma coisa mais unificada</p> <p>Para os DevOps Comentários</p>	<p>é ter uma equipe independente e com responsabilidade para definir parâmetros</p> <p>Comentários</p>	<p>Se a Diretoria não abraçar não vai funcionar</p> <p>Para os DevOps Comentários</p>	<p>Muito bom (claro) que estão hoje na gerência de configuração deveria estar nessa equipe, porque de padrões, ferramentas, etc.</p> <p>Para os DevOps Comentários</p>	<p>sempre que há um problema em alguma contribuição em outra direção há um trabalho de integração com essa outra ferramenta sobre os problemas de "tudo na mão" (mas a integração de valor na atuação de diferentes áreas é crucial)</p> <p>Comentários</p>



6. UX Designer

Nossa primeira experiência não foi, "uma experiência baseada na experiência do usuário" temos uma, mas é uma coisa muito diferente. "Construímos ali mais a gente de fora" Uma só imagem Comentários

Não teríamos o maior prazer em compartilhar este conhecimento, em ampliar esta conversação. Comentários

Já percebi que alguns gestores perceberam o valor que a gente consegue agregar Comentários

O problema clássico de comunicação é que não dá para falar tempo do lado para quem não tem. É um problema que acontece. Estamos a 10 anos na empresa e estamos lá. Tem quem tem um tempo dedicado a ler e analisar os dados e a fazer uma análise.

Não conseguimos avançar muito ainda, mas por questão de logística. Temos 2 profissionais para muitos projetos Uma só imagem

Estamos no processo, está sendo uma experiência que acontece que não é uma coisa que acontece que não é uma coisa que acontece. Quando isso acontece, não é mais uma coisa que acontece que não é uma coisa que acontece. Quando isso acontece, não é mais uma coisa que acontece que não é uma coisa que acontece.

Quanto mais essa conversação for desenvolvida dentro da empresa, e não for uma conversa de fora, é melhor, a nossa interação fica muito melhor, a gente consegue ir mais longe e ter a interação mais rica. Uma só imagem

A gente tem que tentar passar um pouco dessa cultura do lado do produto para os colegas de uma maneira orgânica e natural. Com mudanças culturais que é mais lenta e gradual, tem resultados a longo prazo Uma só imagem

Trazer de problemas complexos, é uma das especialidades de uma de UI, desde metodologia para isso, mas a metodologia não é um template Uma só imagem

Existem algumas coisas que são mais intuitivas que é primeiro entender os requisitos prévios que você entregou, depois voltar a visão do UI, fazer uma apresentação para eles Uma só imagem

A última parte de uma reunião de trabalho é quando o projeto começa a ser desenvolvido. Quando isso acontece, não é mais uma coisa que acontece que não é uma coisa que acontece. Quando isso acontece, não é mais uma coisa que acontece que não é uma coisa que acontece.

Porque normalmente o que acontece no projeto, é que a primeira a reação é partir no ferramenta e depois pensar no que estamos fazendo Comentários

É sempre assim, a gente pensa do importante. Nós para isso não temos ferramentas. Mas não chamamos a gente para trabalhar Uma só imagem

Declarar que a proposta de empresa, ao gerar novos sistemas, deve ser considerado na seguinte ideia "Qual valor estamos entregando para a cidade?"

O mercado não é tão diferente da iplan. Não estamos tão fora do mercado assim Comentários

Uma coisa que a iplan poderia colocar como uma missão estratégica é tentar transformar os órgãos em serviços Uma só imagem

É preciso de ter aquela mentalidade de órgão, pensar em serviço de saúde de cidade, sempre de educação. E dar suporte para esses órgãos, talvez um pouco mais visível de modo que a gente não tenha

Um grande caso de sucesso na iplan é o Banius. Tem uma equipe de atendimento, tem um RO que abraça o produto. Uma só imagem

Tem um caso de sucesso de cidade, de uma cidade que pode ser um serviço para a cidade, para a cidade, desde o ponto, podemos pagar serviços para atender mais coisas, para ajudar a racionalizar o órgão Comentários

É mais difícil, mas não estamos tão fora do mercado assim Uma só imagem

No trabalho, sabemos a razão, trabalho e a competitividade existindo. Podemos pagar mais do que os outros, para fazer coisas que o mercado de mercado que desafiou a gente. Mas isso faz com que a gente tenha a capacidade de fazer coisas que os outros não podem Uma só imagem

Como uma parte de serviço que precisamos entregar e precisamos nos conectar e ajudar os outros, que é o entendimento que não é uma coisa que acontece que não é uma coisa que acontece Uma só imagem

Entendi que é um tipo de uma maneira mais ampla, e não chegou a ser feito, e de fazer uma apresentação para a empresa em que a gente precisa de uma reunião organizada a este respeito. "Qual é a ideia aqui?" Uma só imagem Comentários

Não organizamos uma conferência para 1000 pessoas, na americana latina latina, tem ao lado da iplan e ninguém sabe disso. Uma só imagem Comentários

Não seria importante ali para essa direção. Hoje a sua vontade de ser trazer esse perfil para dentro da empresa, mas ficou entender o espaço que não faz para dentro da empresa Uma só imagem

Estamos buscando sempre encontrar isso fora do produto, por exemplo, a gente não é só colocar no site. A gente é só uma parte do iplan, tem muitas coisas por trás Uma só imagem

Seria interessante que a iplan tivesse um plano de estratégia em plano de estratégia cultural da Prefeitura, com esse tipo de produto para os órgãos Uma só imagem

É uma coisa que não dá para falar tempo do lado para quem não tem. É um problema que acontece. Estamos a 10 anos na empresa e estamos lá. Tem quem tem um tempo dedicado a ler e analisar os dados e a fazer uma análise Uma só imagem

A primeira etapa, é se informar para dentro, analisar este entendimento e ter um discurso mais estruturado para ajudar essa geração de valor neste projeto que inicia Uma só imagem



7. Operações



Os envolvidos devem se sentir responsáveis pelo sucesso do projeto desde o início	Enquanto tá funcionando é meu. Deu problema, é seu!	Qual é o log importante para ser guardado e por quanto tempo. (LGPD)	Temos que analisar os problemas e ter métricas bem definidas	Legado com tecnologia ultrapassadas é um problema
Monitoração não é só com a DOP	As pessoas não são colaborativas. Tem que compartilhar a informação e os setores tem ser colaborativos entre eles.	<small>Os dados tem que ser analisados quando se faz uma documentação. Não somente para desenvolver o sistema, com intuito de dar atenção para o cliente e poder ter melhor feedback em caso de erro com suas funcionalidades, para o comprometimento do atendimento e documentação.</small>	Falta colaboração pra o tratamento dos problemas. Todas devem resolver juntos.	Conhecer os times de outras áreas
Produtos internos são melhores do que os de fábrica	Reunião de Alinhamento seria um retrocesso para Devops	Tenho certeza (que a disseminação da cultura Devops) vai dar certo.	É importante que todos entendam e saibam a importância do produto desenvolvido para alcançar os objetivos	É preciso transparência e integração
Equipe de BDS tem muitos profissionais competentes	É preciso um processo definido que seja usado por todos, evitando exceções.	DOP não tem visão completa das aplicações, integrações e dos bancos acessados	Mudança deve ser em como as pessoas pensam em tecnologia	É necessária a mudança de cultura
Só porque o Dynatrace está na DOP, não significa que somos os únicos responsáveis.	Devemos ter um Projetista de Dados, que vai indicar melhor tipo de dado e onde vai ser utilizado	Junção das áreas (AD + BD + BI). Area de banco está muito pulverizada	As áreas da empresa não se conhecem	Olhar de segurança por conta da LGPD
Ter processo, padrões e disseminar isso também para fora (Fábricas)	Precisamos disseminar o processo e o mesmo tem que ser seguido.	Ter clareza, transparência de como as coisas funcionam. E estarmos aberto a colaboração	Parece que trabalhamos em empresa diferentes	A documentação (do projeto), que é gerada por cada área, é fundamental para atender a demanda.

A solução na infra é dada para resolver um problema da aplicação

Capacitar a equipe que está no legado.

DevSecOps? Existe algum planejamento para isso.

Estamos sempre correndo contra o tempo, com pressão, é muito sofrido, doloroso, cansativo.

No final das contas, colaboração é o sentimento de dono.

Inquisições incompletas, muito generalizadas. A falta de informação sobre requisitos de atividades e tarefas pode ocasionar não funcionamento de aplicação.

8. Desenvolvimento

Sempre temos aquela tarefa que a gente começa a fazer um trabalho, tenta progredir, porém no final, no meio por parte da própria empresa, tem alguma barreira para que o trabalho continue.

Se não tiver apoio da Diretoria, o projeto, como tanto outros trabalhos podem ir por ralo abaixo.

É uma realidade comum, mas se desconsideramos, quando você tenta colocar um projeto em prática, a diretoria adota como o feedback, se você é área de produção, não tem muito a responder, ou não se vêem por cima de pouco feedback.

Os novos temas algo a solucionar, é aprimorando a eficiência de uma ação da área operacional e a aplicação sobre a funcionalidade. A gente pergunta e a resposta é "não foi feito nada", é estranho isso.

Com esta nova cultura, não vai exigir que as pessoas tenham um maior conhecimento entre os assuntos?

3 - E a outra estratégia, é manter como está. Uma equipe de Desenvolvimento e outra de Operações

Uma das medidas, é o que é o sentido da responsabilidade. A responsabilidade pelo produto é da equipe do produto. Dentro da equipe tem vários profissionais para fazer o produto funcionar.

Muito da realidade que se apresenta, é que a gente vai ter que sair do ambiente de desenvolvimento para cima. Um exemplo, é sempre ter que sair do ambiente de desenvolvimento para cima. Um exemplo, é sempre ter que sair do ambiente de desenvolvimento para cima.

A barreira maior é disseminar a cultura devops na empresa. É fazer com que a Diretoria entenda a importância.

Talvez não seja tão complicado sair com o piloto e com um produto, quanto disseminar a cultura devops na empresa.

A diretoria (board) que precisa na parte de consultoria, não ocorre somente entre Operações e Desenvolvimento. Duas áreas que DevOps tem que está lá no início do projeto.

Algo que o time não tem a ideia, que se dá bem, no início, o trabalho com negócios de DevOps, porém não tem feedback com o time. Então, quando se fala de DevOps, não tem a ideia de como é o trabalho.

Quando você tem um profissional que consegue trabalhar, com qualquer área de negócio, é um ganho muito grande.

Poderia incluir no projeto, uma proposta de como lidar com o legado.

Podemos ter um profissional que não tem a ideia de como é o trabalho, porém não tem a ideia de como é o trabalho. Então, quando se fala de DevOps, não tem a ideia de como é o trabalho.

Também é bom lembrar, que não é simples. Tem muita coisa que se dá bem, no início, o trabalho com negócios de DevOps, porém não tem feedback com o time. Então, quando se fala de DevOps, não tem a ideia de como é o trabalho.

A questão estrutural é muito relevante. Existe aquela máquina que fala que o modelo de desenvolvimento de software tem a capacidade de lidar com a organização.

Temos vivenciando um pouco do DevOps, há um tempo, já tivemos um salto grande da maneira de trabalhar, com containers, etc. É um caminho que traz muitos benefícios e é interessante.

Não sabemos o dia de amanhã. Mas é que podemos fazer para aproveitar a oportunidade de trabalhar, aprender e ter mais qualidade como profissionais.

<p>Todos tem que se sentir Dono do Produto.</p>	<p>Papéis e responsabilidades são importantes! Precisam ser mapeados</p>	<p>Temos falta de pessoal.</p>	<p>Atualmente, a estrutura de DTE não funciona. Desenvolvedores e QA's estão trabalhando de forma desorganizada.</p>	<p>Precisamos padronizar as ferramentas de desenvolvimento.</p>
<p>O que é a Definição de Pronto de uma tarefa? Horas registradas, testes unitários. Temos que ter uma definição de pronto de cada tarefa.</p>	<p>Sentimos falta da Terça Tech, que acabou mas deveria voltar por ser um momento importante.</p>	<p>Como desenvolvedor, eu perco muita tempo refletindo, analisando, a analista de negócio ao invés de ajudar, prejudica por influenciar bastante no desenvolvimento.</p>	<p>Os analistas estão fazendo levantamento como antigamente. Falta adequação às novas ferramentas e técnicas.</p>	<p>Se não conseguirmos resolver os requisitos mal definidos no início, o custo do projeto é grande porque pode ter muita redefinição.</p>
<p>Temos poucas definições a DDT. Temos uma dificuldade com todas relacionadas a design, a responsabilidade de testes de unidade, o complexo, complicado, complexo, tem o Senar em inglês, coisas desse tipo são muito pouco em um sistema DDD.</p>	<p>Estamos trabalhando com Gerenciamento de Prioridades e isto pode ser um risco para o projeto piloto.</p>	<p>A GDT é a prioridade de muitos. Isso é válido, mas falta de gente, isso acabou por ser tecnologia e legado. A sala pouco de cada duas substituições que existem por aí.</p>	<p>A GDT é difícil e padronização das tecnologias de desenvolvimento. Temos a eu claro, não muito grande da tecnologia.</p>	<p>É importante colocar no piloto que as empresas externas tem que se alinhar ao novo processo.</p>
<p>Precisamos primeiro adotar como obrigatório em todas as entregas, os testes unitários.</p>	<p>É preciso ter uma documentação mais clara, mais simples, mais fácil de entender, que seja mais clara, mais simples, mais fácil de entender, que seja mais clara, mais simples, mais fácil de entender.</p>	<p>Compartilhar informação contagia, faz o time crescer junto.</p>	<p>É importante ter todos envolvidos que está trabalhando, tem tudo a ver com o tempo, mais para a tecnologia. Ainda que seja o Senar, vai estar oferecendo e mudar a vida piloto.</p>	
<p>Equipe multidisciplinar é diferente de profissional multidisciplinar</p>	<p>A equipe multidisciplinar, é cada um na sua especialidade e apoiando o outro</p>	<p>Se a empresa estiver com uma estrutura de trabalho que não seja mais a mesma, um time de trabalho que não seja mais o mesmo, a estrutura de trabalho que não seja mais o mesmo.</p>	<p>As 3 principais estratégias para começar uma iniciativa de trabalho, no trabalho, no trabalho, no trabalho. 1 - Que seja uma área de trabalho, uma área de trabalho, uma área de trabalho. Um processo de trabalho que seja mais claro, mais simples, mais fácil de entender.</p>	<p>2 - Outra estratégia é ter profissionais multidisciplinares na equipe, e a equipe toma conta de tudo</p>
<p>Qualquer mudança gera conflito. Principalmente uma mudança tão grande.</p>	<p>Uma mudança sempre não é fácil, mas sempre é necessária. Um exemplo é o trabalho. Uma sempre que se trata a gente sempre esperando que a mudança seja feita de repente, que seja feita de repente, que seja feita de repente.</p>	<p>Para isso dar certo, a gente precisa de uma plataforma de trabalho. Um processo de trabalho que não seja mais o mesmo, a estrutura de trabalho que não seja mais o mesmo.</p>	<p>Quando fazemos um trabalho multidisciplinar, precisamos ter uma equipe multidisciplinar, precisamos ter uma equipe multidisciplinar, precisamos ter uma equipe multidisciplinar.</p>	<p>Seja a mudança, seja a mudança, seja a mudança. Uma sempre que se trata a gente sempre esperando que a mudança seja feita de repente, que seja feita de repente, que seja feita de repente.</p>

<p>Se não entendem DevOps, começa lá no início do desenvolvimento do projeto. Temos um problema sério quando a gente trabalha com SFTU, com o usuário de equipe. Muitas vezes não conseguimos entender os nossos erros.</p> <p>Conhecimento</p>	<p>De Análise de Negócio (de pessoas que estão falando direto com o Cliente) tem muita gente se tornando um impacto no negócio. Ao lado o trabalho de análise, a gente precisa pensar mais do que o cliente.</p> <p>Conhecimento</p>	<p>Hoje o Análise de Negócio, está muito aquém do processo. Precisa de uma psicologia. Precisa ter uma visão mais baseada na experiência do usuário e talvez cobrar para agregar e formação dele.</p> <p>Conhecimento</p>	<p>A organização (plan, difícil um pouco a implementação de Devops, pq é trabalho em silos, isso dificulta bastante a implementação.</p> <p>Conhecimento</p>	<p>Desenvolvimento de Gerência de Produto de Desenvolvimento que engloba todo o Devops. Cabe todo mundo, ambiente do gerenciamento.</p> <p>Conhecimento</p>
<p>É impossível que todo mundo saiba tudo. A gente não precisa saber tudo, mas precisamos saber o suficiente. Não precisa ser um DBA para fazer algo que é performático.</p> <p>Conhecimento</p>	<p>Quando falamos a frase, "a gente não precisa saber tudo", isso é um fato. Mas a gente precisa saber o suficiente. A única coisa que temos a confiança e certeza é de que o mesmo hoje é muito coisa.</p> <p>Conhecimento</p>	<p>A gente tem nível de tudo. Todos o cara que entende não aprendeu, o que dá para muito básico. Mas tem que ter outros skills que ele precisa desenvolver.</p> <p>Conhecimento</p>	<p>Hoje em TI, nunca houve tanta oferta de conteúdo. Não precisa nem pagar um Alura da vida.</p> <p>Conhecimento</p>	<p>Não tem jeito, temos que embarcar nesse maré. É a nossa profissão. Temos que encerrar que é desafiador.</p> <p>Conhecimento</p>
<p>As prioridades se deslocam demais. Como melhorar isso?</p> <p>Conhecimento Conhecimento</p>	<p>Eventos de compartilhamento de conhecimento poderiam ser gravados e compartilhados no YouTube.</p> <p>Conhecimento</p>	<p>Alguns problemas e atividades, como Pair Programming, eventos como a Tech Talk, integração com os membros da equipe, fazer as coisas mais ágeis, integrar entre os projetos, fazerem tudo.</p> <p>Conhecimento Conhecimento</p>	<p>Muitos projetos acontecendo ao mesmo tempo. Acelerou muito a mudança de um projeto para o outro.</p> <p>Conhecimento Conhecimento</p>	<p>Para poder criar um Pipeline é preciso ter mais conhecimento. É preciso nivelar o conhecimento.</p> <p>Conhecimento</p>
<p>Existe ainda esta cultura de tentar segurar a equipe, criando requisitos que não agregam valor.</p> <p>Conhecimento</p>	<p>Análise de negócios na mesma situação que todos os outros, mas a gente tem que ter uma visão mais ampla para ter um trabalho e desenvolver a ideia de negócio de negócio, precisa saber negociar e gerar.</p> <p>Conhecimento</p>	<p>O analista de negócio precisa saber negociar para trazer o que tem valor para o sistema.</p> <p>Conhecimento</p>	<p>Hoje que ainda fazemos coisas, que em um pipeline Devops, não passamos. Um exemplo, o trabalho de testes automatizados. Ainda não sabemos se foi via SaaS, mas sabemos que temos uma abordagem.</p> <p>Conhecimento</p>	<p>Como temos desafios na relação a gente, ainda precisamos pensar separados, porque a não ser que tenhamos efetivamente como usar direito as coisas no git. Quando coisas locais não se tem necessidade.</p> <p>Conhecimento</p>
<p>Uma empresa que trabalha com o usuário de equipe, precisa ter uma cultura de desenvolvimento que seja mais performático e que seja mais ágil e que seja mais colaborativo. Não tem que ser uma cultura que seja mais performático e que seja mais ágil e que seja mais colaborativo.</p> <p>Conhecimento</p>	<p>Seria interessante que o projeto do PDI, poderia documentar as etapas de desenvolvimento, as condições, formas de se trabalhar e aprender para a gente.</p> <p>Conhecimento</p>	<p>De um tempo para cá, tem um trabalho muito a mudança de um projeto para outro. A gente acaba não conseguindo acompanhar as mudanças de tecnologia.</p> <p>Conhecimento</p>	<p>Precisamos cobrar a empresa, para poder ter tudo isso. Também temos que ter um trabalho para não ser muito do usuário, só para aprender mais para poder ajudar em trabalho mais performático.</p> <p>Conhecimento</p>	<p>Quando o Devops, tudo isso acontece, não se precisa ter de novo pipeline, não tem um pouco diferente de outros, para poder a adoção. Com o tempo, podemos ganhar um processo de adoção e isso acaba.</p> <p>Conhecimento</p>

APÊNDICE C - Revisão do processo de desenvolvimento atual - Equipe Java

1. Visão Geral



A seguir apresentamos o processo de desenvolvimento atual da Equipe Java e os pontos de melhoria identificados.

2. Desenho



Nesta fase do processo, é desenhada a visão do produto que será desenvolvido, com discussões entre Gestor, Stakeholder e Líder de Projeto. Nesta fase, o Líder do Projeto inicia as etapas de criação do projeto e as histórias de usuários são criadas, para priorização do Gestor.

Pontos de melhorias identificados nesta fase:

- Envolvimento dos times de Qualidade, Desenvolvimento, Operações e Administração de Dados, nas reuniões do desenho da solução e arquitetura;
- Integração das diversas ferramentas de gerenciamento do projeto para evitar duplicação de informações e retrabalho de atividades.
- Definição das responsabilidades dos Líderes de Projetos e Analistas de Negócio, para evitar a sobrecarga dos papéis.
- Definir os indicadores para acompanhar o projeto, melhorar a qualidade das informações registradas no Jira e gerar dashboards que deem uma visão do acompanhamento do projeto.
- Conscientização do papel do Gestor e suas responsabilidades.
- Capacitar os analistas de requisitos para ter uma visão com foco na experiência do usuário.
- Envolvimento do time QA na elaboração das histórias para construção de casos de testes.
- Definir o mínimo de percentual de cobertura de testes funcionais para cada entrega.

3. Construção



Nesta fase, é adotado o framework Scrum com a realização de algumas cerimônias ao longo do processo. Após a fase de inicialização e priorização do projeto, as histórias são organizadas no backlog que são priorizadas na reunião de Planning. As sprints, no geral, possuem um time-box de 15 dias podendo ser adaptada de acordo com evolução ou necessidade do projeto. A equipe de desenvolvimento realiza as cerimônias de Daily Meetings, onde são informados ao Scrum Master a evolução das atividades e os impedimentos.

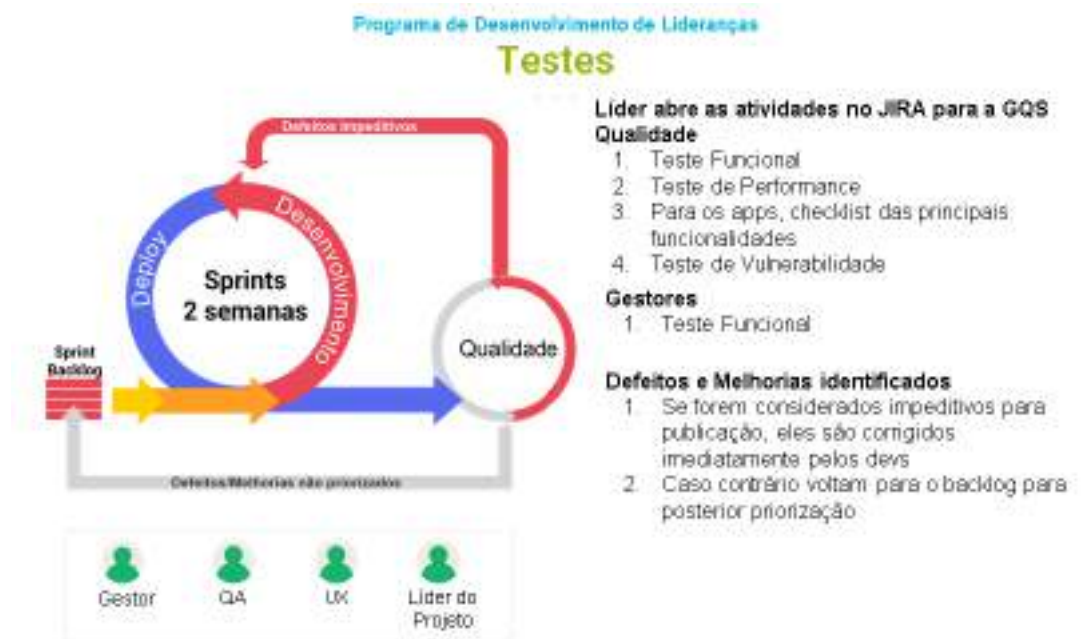
Ao final da sprint, a equipe apresenta as funcionalidades implementadas ao Líder do Projeto e, de acordo com sua avaliação, as funcionalidades podem ser liberadas para a Qualidade testar. Nesta mesma reunião, realizamos o planejamento da próxima sprint.

Pontos de melhorias identificados nesta fase:

- Complementar o quadro de desenvolvedores para atender a demanda reprimida;
- Conscientizar os desenvolvedores sobre a importância dos testes automatizados;

- Registrar, controlar e priorizar os débitos técnicos.
- Definir o “conceito de pronto” da história
- O SonarQube não é utilizado internamente para avaliação do código-fonte e nem vulnerabilidade de código.

4. Testes



Nesta fase, a equipe de Qualidade recebe as histórias que foram priorizadas na sprint para serem testadas. A Qualidade realiza os testes com base nos cenários de aceite da história. Os dados para teste são fornecidos pelo Líder do Projeto ou a Equipe de desenvolvimento. Os testes de performance, vulnerabilidade e carga são executados na entrada em produção.

Pontos de melhorias identificados nesta fase:

- Não são realizados testes para todos os projetos, fica a critério do gestor a decisão;
- Ausência de histórias com foco em testes, a equipe de testes não atua em paralelo e nem sempre participa das reuniões de **planning** para escrever os casos de testes;
- Time de Qualidade necessita de capacitação para automação dos testes e ajuda para integrar ao pipeline;

- Os testes não funcionais são realizados somente na entrada em produção, as atualizações e evoluções não realizam testes;
- Um perfil de Qualidade deveria participar da sprint para apoiar nos casos de testes e automatizar os testes funcionais das histórias.
- Implantar indicadores e métricas para avaliar a qualidade das entregas, dos testes, entre outros.

5. Entrega



Figura 8

A equipe Java utiliza a plataforma de Openshift para automação do processo de entrega, que é responsabilidade do Arquiteto de Soluções. Atualmente este perfil, cria o projeto no Openshift, configura o pipeline de build e prepara todos os ambientes.

Pontos de melhoria identificados nesta fase:

- O time de Arquitetura de Soluções é pequeno para atender todas as demandas;
- Necessita de aprovação para entrada em produção porque o pipeline atual não cobre todas as práticas de Entrega Contínua e não temos como garantir a qualidade e segurança da entrega. Além disso, o processo atual de

publicação em produção utilizado pela empresa requer a autorização do Gestor;

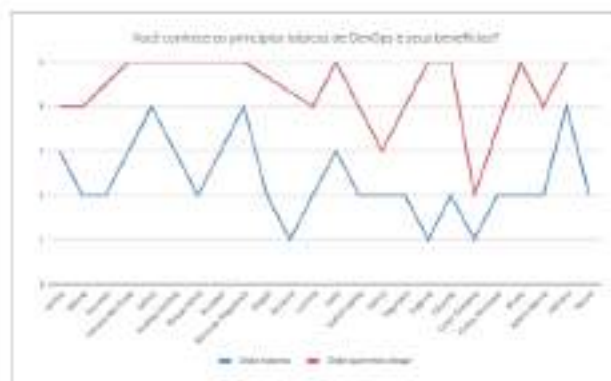
- Melhorar a comunicação com a área de Operações, pois com o uso do Openshift esta área ficou sem visibilidade da realização de testes de performance, vulnerabilidades e entrada em produção das aplicações;
- Implantar monitoração das aplicações, integrações e demais ativos/serviços envolvidos;
- Melhorar o armazenamento de variáveis, chaves secretas e demais parâmetros de configuração;
- Implementar gerenciamento de logs;
- Disseminar o conhecimento da plataforma, que atualmente é restrito aos arquitetos.

APÊNDICE D - Assessment

1. Análise por Questão

1.1 Pessoas & Cultura

Questão 1: Você conhece os princípios básicos de DevOps e seus benefícios?



Comentários

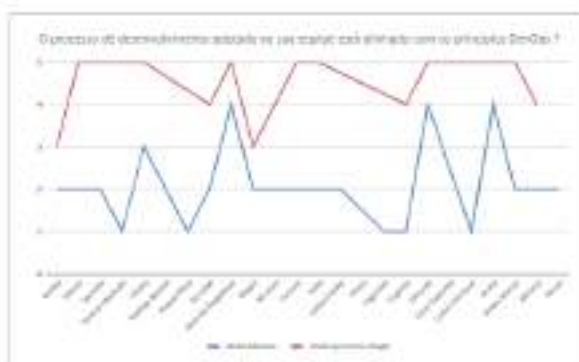
Difundir o conhecimento sobre o assunto e exemplificar com exemplos (cases) práticos.

Mais votadas



	Poucos possuem conhecimentos e experiência
	Práticas DevOps são institucionalizadas em toda a empresa

Questão 2: O processo de desenvolvimento adotado na sua equipe está alinhado com os princípios DevOps?





Comentários

Não se aplica a Q1 e a minha equipe não desenvolve software.

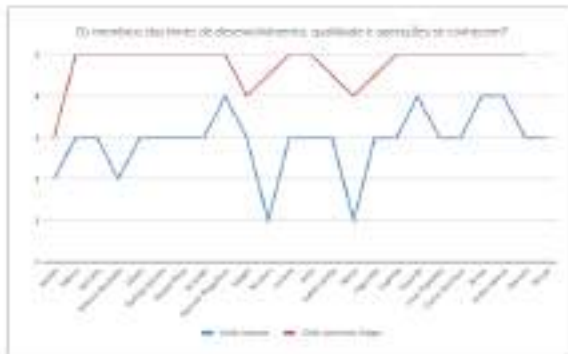
Mais votadas



Legenda

	Somente desenvolvimento de apl. GA e DevOps básico
	Agilidade fim-a-fim entre todas as equipes (aplicações, plataformas e infraestrutura)

Questão 3: Os membros dos times de desenvolvimento, qualidade e operações se conhecem?



Legenda

	Os membros se conhecem e interagem apenas no lançamento ou se ocorre um problema.
	Todos os membros se conhecem e trabalham de forma coesa e colaborativa.

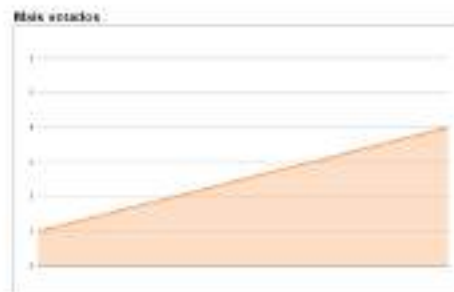
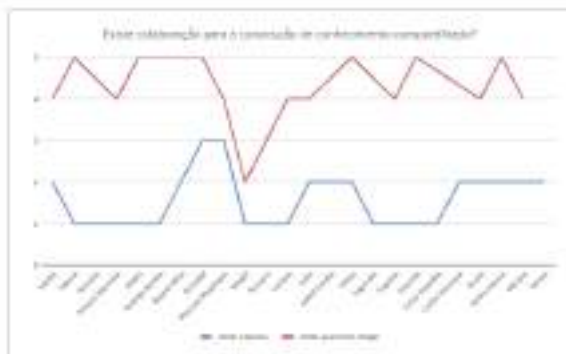
Questão 4: Há eventos regulares na organização que integrem as equipes?



Legenda

	Parceiros, apesar dos eventos, externos e não há compartilhamento de conhecimento.
	Os times realizam regularmente talks, workshops divulgados para toda empresa.

Questão 5: Existe colaboração para a construção de conhecimento compartilhado?

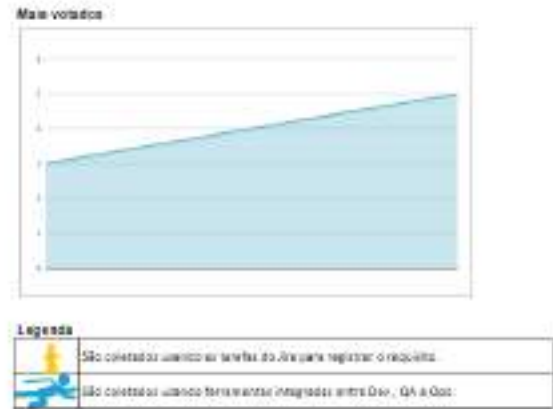
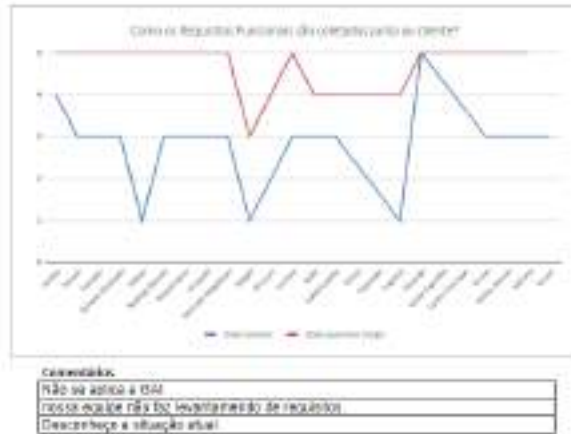


Legenda

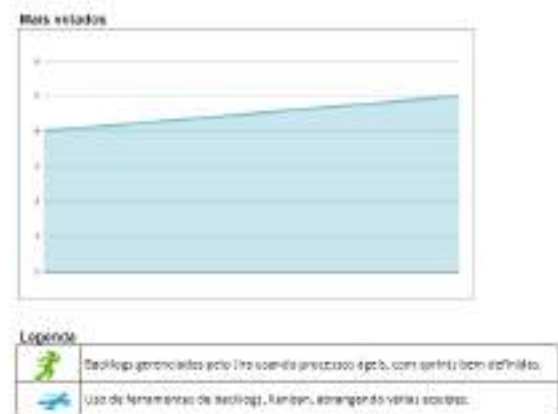
	Não existe compartilhamento de conhecimento entre as equipes Devs e Ops.
	Acontecem reuniões de compartilhamento regularmente.

1.2 Planejamento

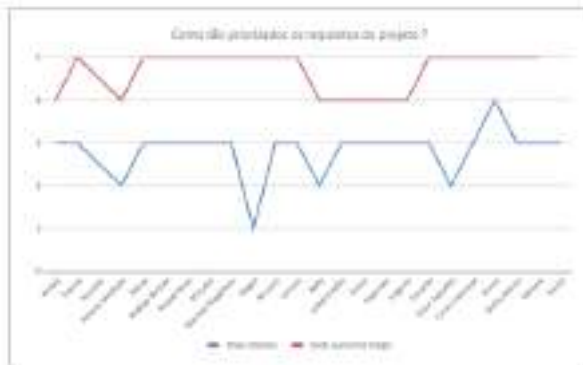
Questão 6: Como os Requisitos Funcionais são coletados junto ao cliente?



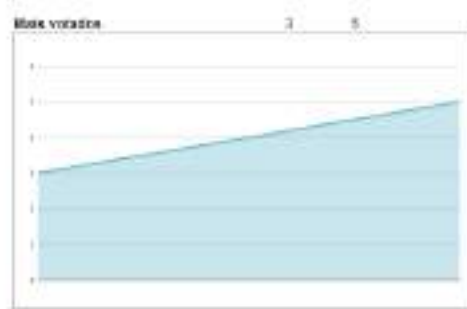
Questão 7: Como é realizado o acompanhamento do projeto?



Questão 8: Como são priorizados os requisitos do projeto?

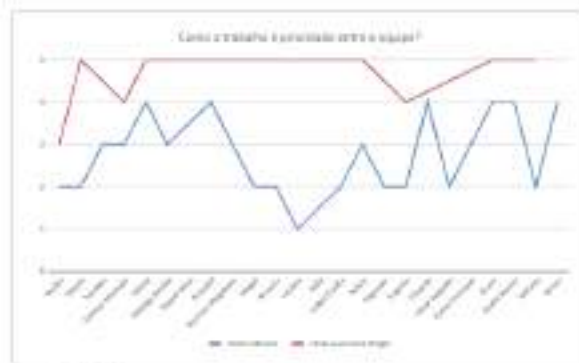


Comentários
 Não se aplica a QM
 Não se responder, mas acredito que seja a resposta marcada



Legenda
 Requisitos são priorizados entre os stakeholders, Líder do Projeto em um ambiente de sim.
 Todos os envolvidos (Dev+Ops+Qas) participam na priorização dos requisitos registrando no Jira.

Questão 9: Como o trabalho é priorizado entre a equipe?



Comentários
 O processo de priorização em nossa equipe, vai do nível de gerência até a diretora



Legenda
 O líder do projeto define as atividades que cada membro irá trabalhar de acordo com
 O time envolvido (Dev+Ops+Qas) define jant e priorização das atividades

Questão 10: Todos os envolvidos conseguem visualizar os requisitos e o status do projeto em um dashboard integrado



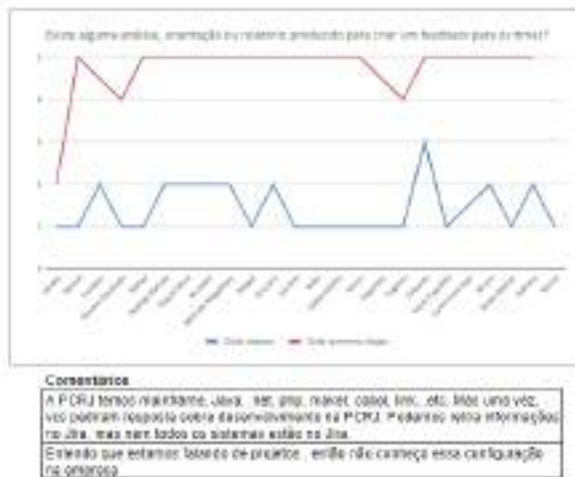
Comentários
 Não se aplica a QM
 Jantão é usado. Lembre de que as perguntas são sobre o desenvolvimento de POCs e não de uma determinada equipe.
 Não participo de projetos, portanto não posso que não conheço o assunto



Legenda
 Cada time possui seu próprio painel de controle.
 O quadro de status compartilhado entre DevOps/Colabor/Dev/Dev/Dev.

1.3 Feedback

Questão 11: Existe alguma análise, orientação ou relatório produzido para criar um feedback para os times?



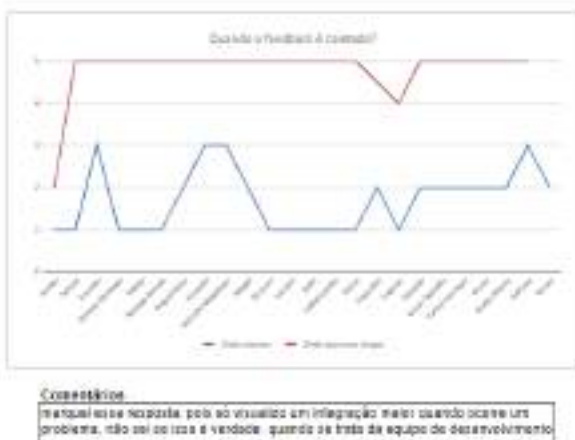
Mas volamos



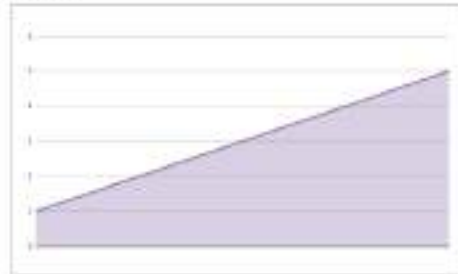
Legenda

	Não há relatório.
	Uso de ferramentas de análise provendo feedback para melhoria contínua.

Questão 12: Quando o feedback é coletado?



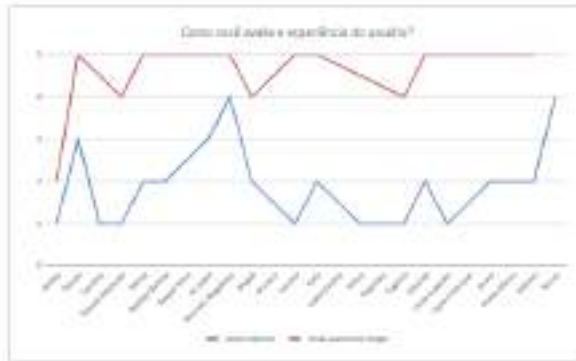
Mas volamos



Legenda

	Demora até que há um erro.
	Existe a opção de obter feedback a qualquer momento, durante todas as fases.

Questão 13: Como você avalia a experiência do usuário?



Comentários

Essa o tema de encerramento do projeto e na minha opinião a área superficial em termos de avaliação de usuário.
Não conheço como o assunto está neste momento na organização de forma mais geral.
Não tenho conhecimento sobre esse tema.

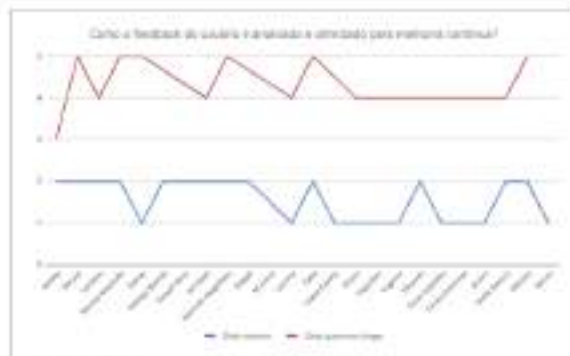
Mais votado



Legenda

	Avaliação de e realizado quando Gestor ou Qui apontar não contém dados.
	Através de testes A/B em produção, feedback dos usuários, métodos estatísticos.

Questão 14: Como o feedback do usuário é analisado e otimizado para melhoria contínua?



Comentários

Não conheço esta ação quando lançamos de projetos

Mais votado



Legenda

	O feedback é analisado quando há retorno dos usuários finais.
	Use de técnicas como Análise de Causas Raiz, entrevistas, métodos.

Questão 15: Existe compartilhamento do feedback entre os times? Como isto é praticado?



Comentários
 Não se aplica a GIN
 Não sei como isto é tratado atualmente, seja movimentos para migração e feedback mas no projeto só aqui quando há problemas, mas quando muitas modificações são realizadas sem o conhecimento de todos, mas não tenho conhecimento de como isto está sendo tratado pelo participantes desse do

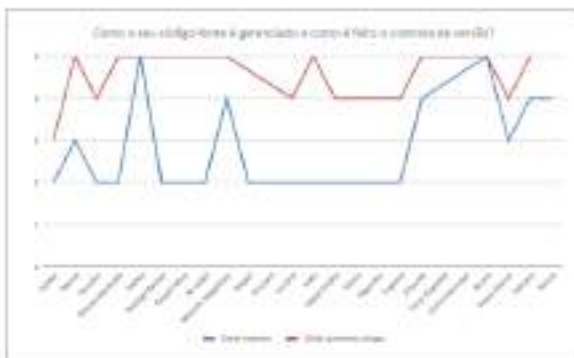


Legenda

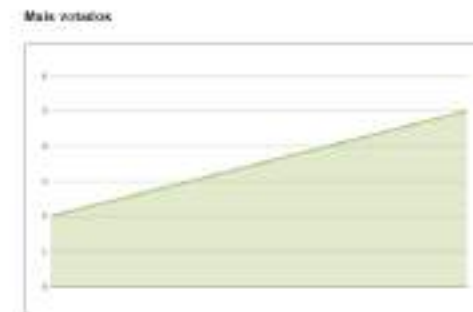
	Apenas usando e-mails tradicionais.
	Tem como recebendo feedback contínuo.

1.4 Integração Contínua

Questão 16: Como o seu código-fonte é gerenciado e como é feito o controle de versão?



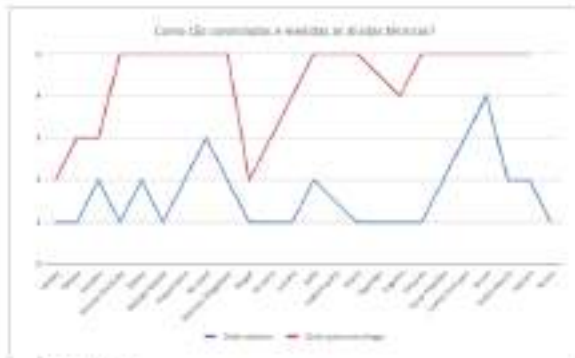
Comentários
 Nossa equipe não desenvolve, mas cada projeto faz o próprio do versão do código com o que há disponível e de seu conhecimento.
 No nosso caso, os modelos são versionados no SVN mas não estão integrados com o código fonte.
 Entendo que sim, havia um controle das modificações, o que é vital para manutenção as configurações de sistema, mas não consigo ser há outros projetos



Legenda

	Utiliza sistema de controle de versão (SVN/GIT) sem pipeline integrado.
	Utiliza pipeline para gerenciamento de integração e implantação.

Questão 17: Como são controladas e medidas as dívidas técnicas?



Constatantes

Me parece ser manual, mas não sei responder

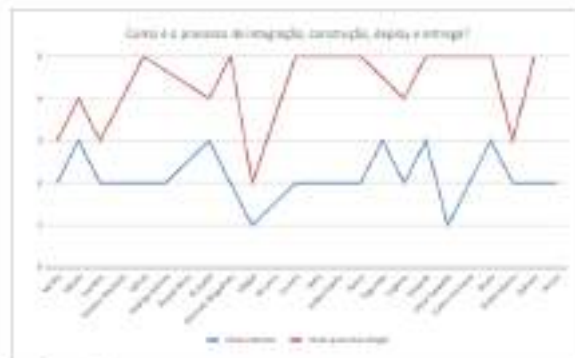
Mais votados



Legenda

	Dívidas técnicas não é revisado
	Dívidas técnicas

Questão 18: Como é o processo de integração, construção, deploy e entrega?

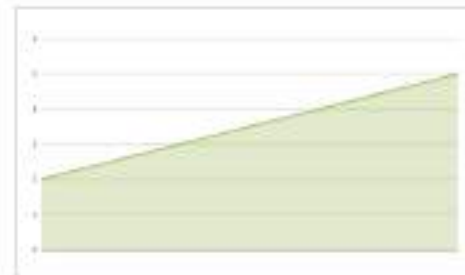


Constatantes

Dependência do projeto o próprio desenvolvedor faz todo o processo em vários projetos isso não ocorre. Então aqui por "processo não definido" não conheço o assunto para responder.

Não conheço a fundo estou tentando entender, mas vejo que muitas modificações não são conhecidas por todos, nem sei se deveria, mas muitas vezes só percebemos quando esta modificação causa alguma problema no ambiente

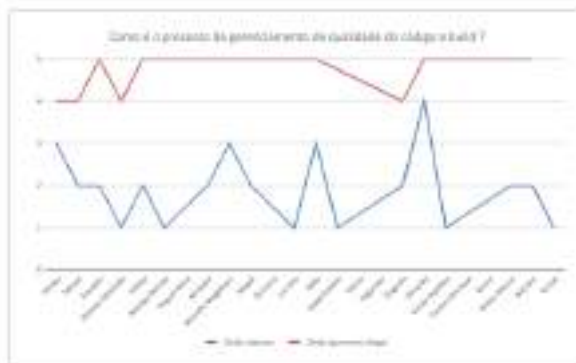
Mais votados



Legenda

	Código em repositório, ferramenta de construção com geração de dependências, servidor de build automatizado sem texto.
	Run + scripts automática de build de código em casa de teste e na qualidade do código de teste.

Questão 19: Como é o processo de gerenciamento de qualidade do código e build ?



Comentários

Não se aplica a QM

Depende do projeto. Tem projetos que usam SonarQube, Maven, etc. Mas maioria dos projetos não há um processo de gerenciamento de código. Como não identifiquei qualquer resposta nesse sentido, vou optar pela última resposta.

Não conheço o assunto

Não sei sobre centros especializados, mas a que percebo que muito anos são informados pela equipe de suporte, mas não sei se são tratados pois não vejo uma resposta para informação passada

Mais votados



Legenda

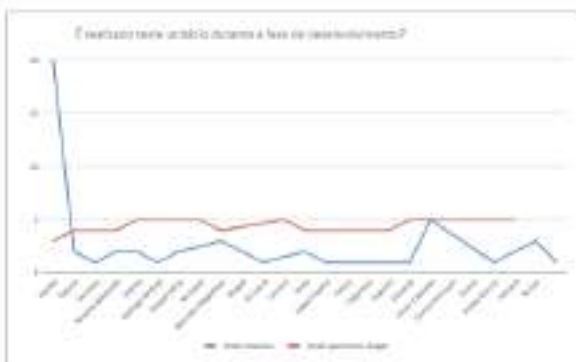


Processo manual de revisão de código, sem ferramentas de acompanhamento de alteração de código fonte.



Processo automático e integrado com o framework de build (Maven e revisão de código SonarQube).

Questão 20: É realizado teste unitário durante a fase de desenvolvimento?



Comentários

Não se aplica a QM

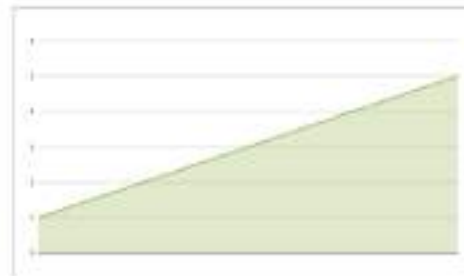
Não sei ao certo responder. Mas sabe que os perguntas são direcionadas para grupo específicos e não para a PDU como um todo. Eu não sei se os testes unitários são executados, mas deveria ser automatizados.

Desconheço situação atual

Não consigo avaliar como o assunto está de forma global.

Não tenho conhecimento de como isso ocorre em todos, o que torna complicado e o teste de vulnerabilidade que é realizado, mas também não sei se isso acontece o projeto de todo.

Mais votados



Legenda



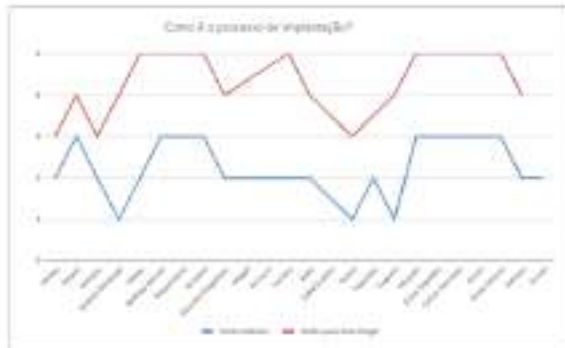
Não há teste unitário definido.



É automatizado, com cobertura entre 80 e 90%

1.5 Entrega Contínua

Questão 21: Como é o processo de implantação?



Comentários:
 Não sei se o processo de implantação é aplicado de forma igualitária para todos os projetos, mas acho que isso pode ser feito.

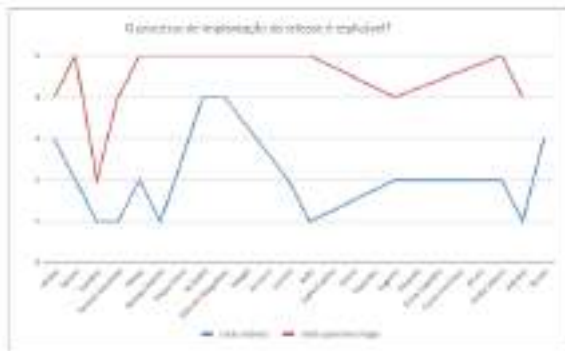
Mais robôes



Legenda

	Implantação automatizada por alguns projetos. Toda a configuração é manual.
	Implantações gerenciadas. Processos testados com possibilidade de rollback.

Questão 22: O processo de implantação da release é replicável?




Comentários:
 Não se aplica a CAJ
 Desconheço situação atual

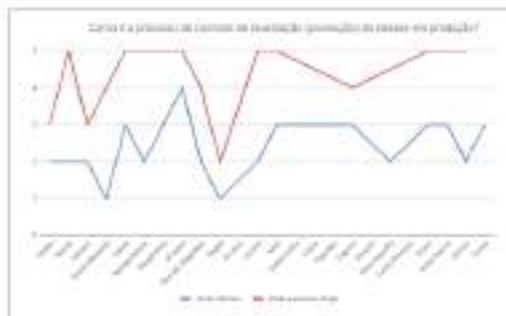
Mais vocádes



Legenda

	Replicável, mas com grande manualismo.
	Totalmente replicável, métodos e procedimentos mais maduros.

Questão 23: Como é o processo de controle de atualização (promoção) de release em produção?



Características
 Não se aplica a QM
 Dependem do projeto e de "se sabe com quem está falando". Vamos pensar para não dar para produção sem passar pela QA. A resposta não perto de produção é um pouco atrasado porque não QA. Devido que se for possível desenvolver para produção via QA.
 Exemplo que é o processo de gerenciamento e atualização em todos os ambientes, não tem conhecimento a isso é aplicado em todos ambientes.

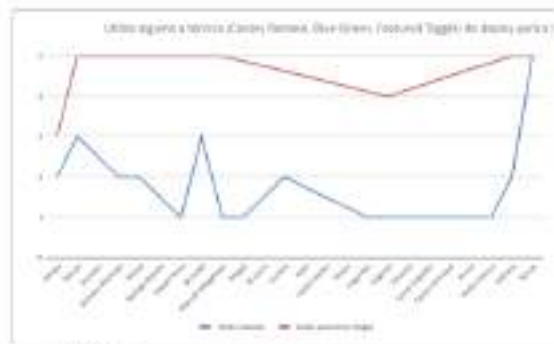
Meu voto(s)



Legenda

	Seguir o procedimento de QA para aprovação
	Fluxo automático, gerencia o check-in de toda release de forma segura.

Questão 24: Utiliza alguma a técnica (Canary Release, Blue-Green, Featured Toggle) de deploy para a implantação



Características
 Não consegue usar técnicas.
 Não se aplica a QM
 Nunca se aplica a QA ou QA. O outro não sabe, não conhece o assunto.

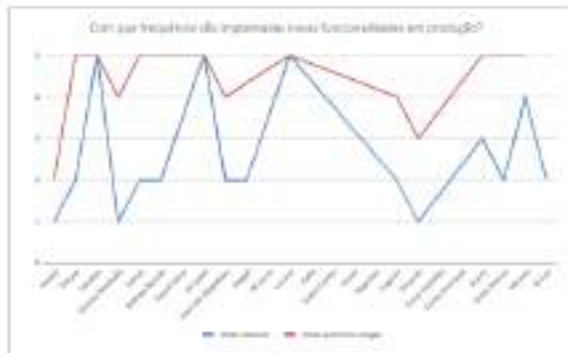
Meu voto(s)



Legenda

	Não pode usar QA
	Implementado, mas não se aplica técnica de implantação de QA

Questão 25: Com que frequência são implantadas novas funcionalidades em produção?



Considerações

Depende do projeto
 Não se aplica a GAJ
 Depende do projeto
 Não tenho esse tipo de controle, só preciso quando ocorre problemas no jenkins por exemplo

Mais vídeos

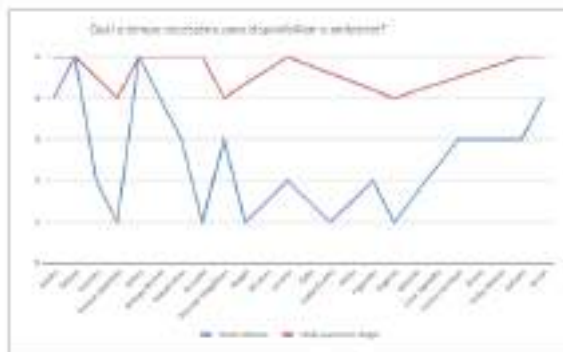


Legenda

	Pelo menos uma vez ao mês
	É possível implantar até 4 ou mais vezes ao dia

1.6 Infraestrutura como código

Questão 26: Qual o tempo necessário para disponibilizar o ambiente?



Considerações

Não se aplica a GAJ
 Não consigo o status de agenda
 Entendo que consiste o código é usado automaticamente por script, demais partes de configuração são feitas em separado para atender os requisitos do projeto (versões de serviços), vejo que não existe uma plataforma única e que faz com que não tentamos como criar todos ambientes de forma automática.

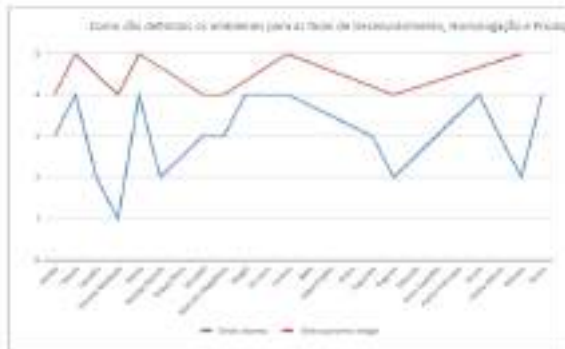
Mais vídeos



Legenda

	Ambiente físico até 40 horas
	Ambiente em cloud de 2 a 4 horas

Questão 27: Como são definidos os ambientes para as fases de Desenvolvimento, Homologação e Produção?



Comentários

Não se aplica a GAH

Não está respondida para o questionamento, mas muitos ambientes não são exclusivos e são compartilhados, mas alguns podem ter exclusivos dependendo do requisito da prova.

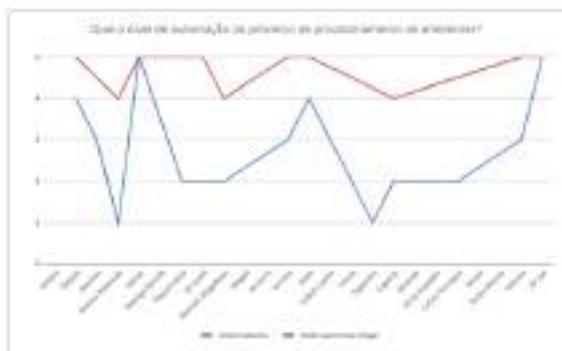
Mais votados



Legenda

	Uso de PaaS (OpenShift) - uso de imagens de produção, sem controle de rede/ouso
	Uso de PaaS (OpenShift) - uso de imagens de produção, sem controle de rede/ouso

Questão 28: Qual o nível de automação do processo de provisionamento de ambientes?



Comentários

Não se aplica a GAH

Não controla o status do serviço.
Entende que somente o serviço é criado automaticamente por script, demais

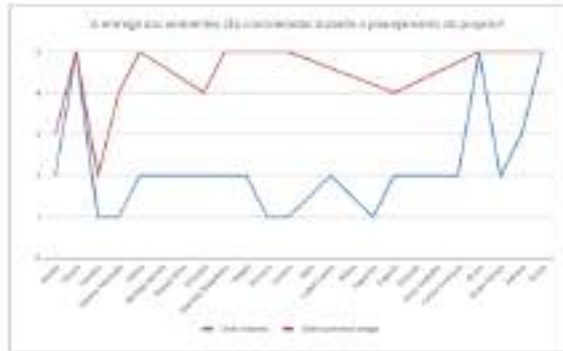
Mais votados



Legenda

	Orquestração de VM por scripts, demais tarefas manuais
	Totalmente automatizado e orquestrado (OpenShift)

Questão 29: A entrega dos ambientes é considerada durante o planejamento do projeto?



Comentários
Não se aplica a GAJ

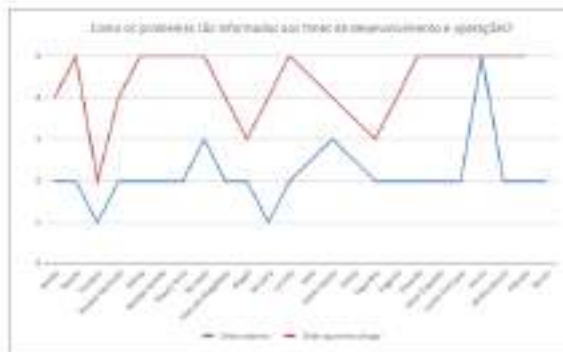
Mais votados



Legenda

	Pode ser afetado pela disponibilidade de recursos
	Serviço disponível e configurável pelas equipes sob demanda (Oportunidade)

Questão 30: Como os problemas são informados aos times de desenvolvimento e operações?



Comentários
Não se aplica a GAJ

Mais votados

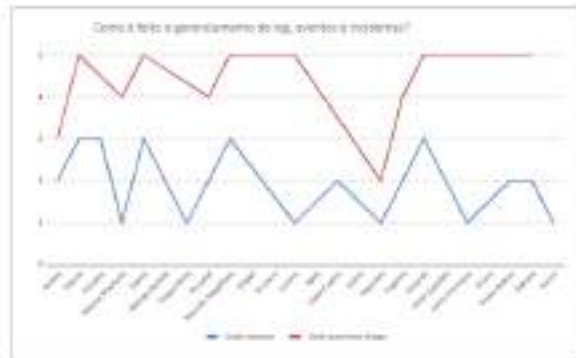


Legenda

	Alerta de resolução dos usuários final (telefone, email, tickets), tickets, notas de
	Por fechamento de tickets disponíveis para todos as equipes em tempo real

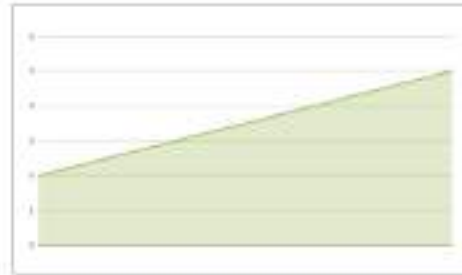
1.7 Monitoração

Questão 31: Como é feito o gerenciamento de log, eventos e incidentes?





Consistência
Não se aplica a GUI

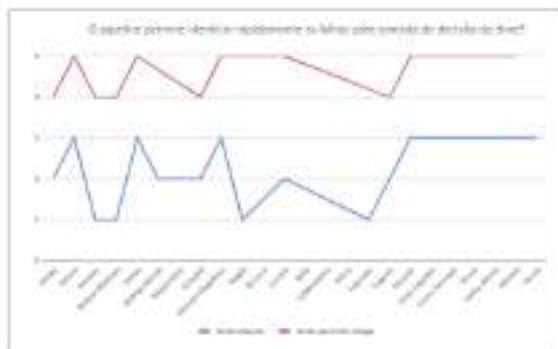
Mais votados



Legenda

	Comunicação à central e investigação de log por demanda
	Uso de ferramentas centralizadas com integração a sistemas com ferramentas de ps

Questão 32: O pipeline permite identificar rapidamente as falhas para tomada de decisão do time?



Consistência

Desconecta

Reconecta?

Há o carregamento do sistema

Mais votados



Legenda

	Comunicação à central e investigação de log por demanda
	Pipeline totalmente integrado para todos os testes, no nível de desenvolvimento e produção

Questão 33: A equipe de Operações é envolvida no início do projeto, para opinar sobre ambientes, implantação e planejamento de capacidade?



Comentários

Antigamente havia uma reunião no início dos projetos q envolvia todos os departamentos. Agora essa reunião só acontece quando há alguma falha no sistema. Por isso várias falhas e melhorias podem ser feitas nessa reunião.

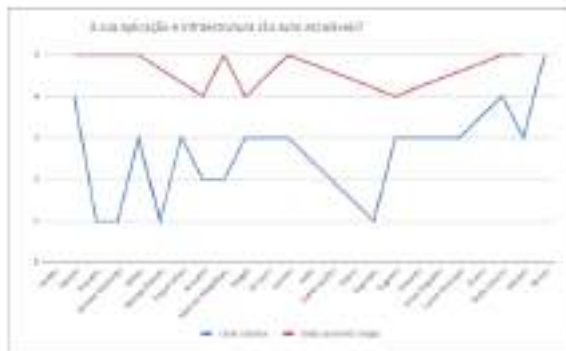
Mais votos



Legenda

	Apenas quando necessário, caso a caso
	É uma obrigação clara de forma case a case

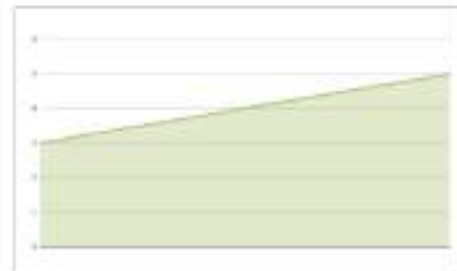
Questão 34: A sua aplicação e infraestrutura são auto escaláveis?



Comentários

Não se aplica o SSI

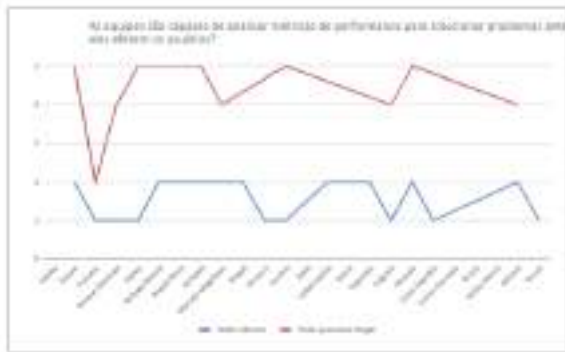
Mais votos



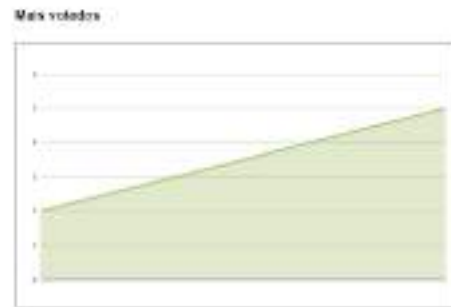
Legenda

	Requer ação de recursos a infraestrutura
	De forma automática, com base em parâmetros configuráveis, como CPU/memória.

Questão 35: As equipes são capazes de analisar métricas de performance para solucionar problemas antes que eles afetem os usuários?



Comentário:
 Não se aplica a D4
 Nunca nenhuma métrica sendo usada.

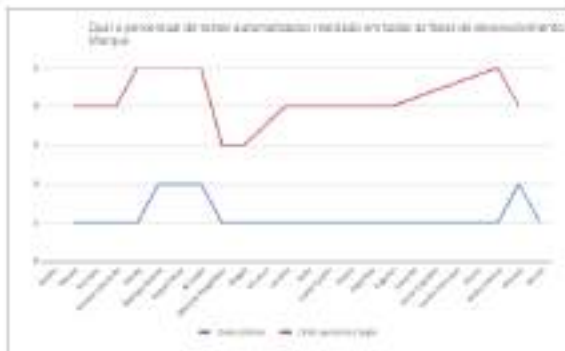


Legenda:

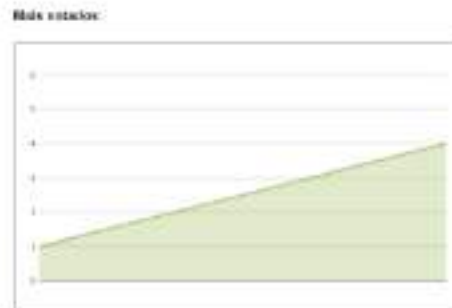
- Não afetam os usuários
- Soluções implementadas por equipes de suporte

1.8: Testes

Questão 36: Qual o percentual de testes automatizados realizados em todas as fases de desenvolvimento?



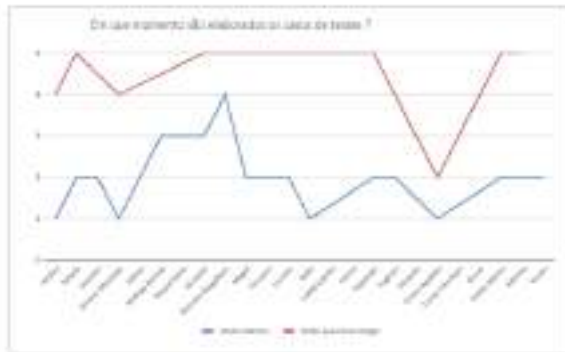
Comentário:
 Não se aplica a D4
 Não tenho essa visibilidade



Legenda:

- Não existem testes automatizados
- Até 50%

Questão 37: Em que momento são elaborados os casos de testes?



Comentário

Não se aplica a GAI
 Estes serviços de EDD TDD são visto como gatilho para a PCMI. Acredito que eles devam continuar com a desenvolvimento. Dependem do projeto. Os projetos menores onde não envolveram em participações desde o início, mas isso não é um fe grande.
 Os casos de teste, para os testes automatizados, deveriam ser extraídos dos requisitos/entregas pelos desenvolvedores. Portanto, os requisitos/testes devem estar "ligados" a cada fase o que necessita ser testado.

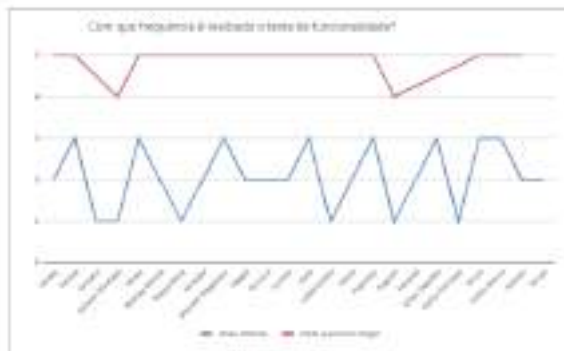
Mais votados



Legenda

	Qua participam sob demanda dos reuniões de planning para coletar os requisitos
	Qua participam desde o inicio, utilizando técnicas como EDD ou TDD com testes automát

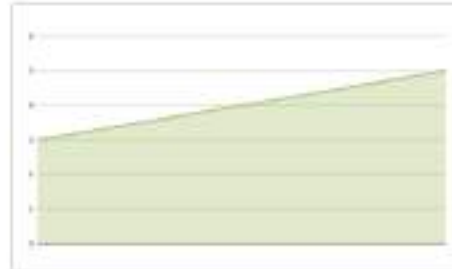
Questão 38: Com que frequência é realizado o teste de funcionalidade?



Comentário

Não se aplica a GAI
 Depende do projeto, normalmente quando o sistema vai entrar em produção "analisar".
 Mas nos projetos onde os CA's são elaborados desde o início os testes funcionais.
 Não tenho visibilidade de testes testes, muitas vezes minha visão é só quando ocorre os problemas.

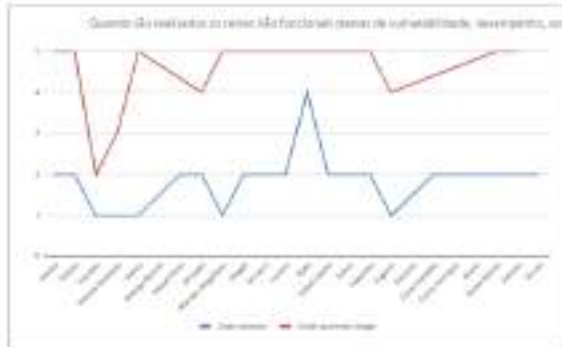
Mais votados



Legenda

	As entregas de cases sprint
	Qua faz parte do time e realiza os testes antes de entrega os sprint.

Questão 39: Quando são realizados os testes não funcionais (testes de vulnerabilidade, desempenho, sonar, etc)?



Comentários

NÃO QA!

Não se aplica a QA!

Mas não é "contingência" dependo de quem quer ocorrer em produção e o usuário se atreva em produção. Há vários sistemas que entram em produção sem QA sabem pelo QD.

não posso afirmar, mas o que vejo é que esse teste só são feitos quando o sistema está prestes a entrar em produção.

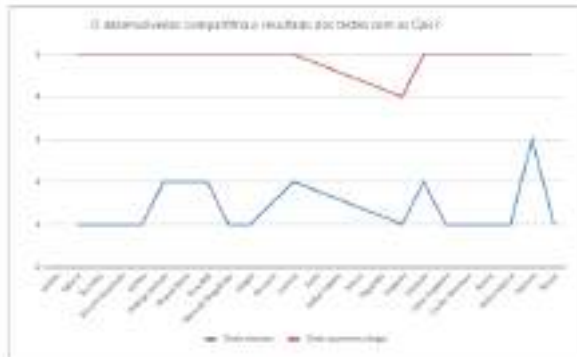
Mais votados



Legenda

	Original, sem que áreas de entrada em produção
	Testes não funcionais automaticamente no pipeline

Questão 40: O desenvolvedor compartilha o resultado dos testes com os QAs?



Mais votados















Legenda







	Não há casos de testes
	Executado automaticamente no pipeline e disponível em painel de controle.

2. Questionário







2.1 Pessoas & Cultura

1. Você conhece os princípios básicos de DevOps e seus benefícios?					
					 CONFUSED
Poucas conversas e nenhuma educação interna	Poucos possuem conhecimentos e experiência.	Existe um consenso, tem se construído atividades para aplicar.	Existe um consenso, tem se construído um processo para aplicar.	Práticas DevOps são institucionalizadas em toda a empresa	Não entendi a pergunta ou não conheço o assunto.







2. O processo de desenvolvimento adotado na sua equipe está alinhada com os princípios DevOps ?					
					 CONFUSED
Uso de metodologia tradicional (cascata).	Somente desenvolvimento usa ágil. QA e Ops separados	Agilidade integrada entre Dev e QA, mas Ops é separado.	Usam processo ágil fim-a-fim até a produção. Dev totalmente, mas Ops parcialmente.	Agilidade fim-a-fim entre todas as equipes, aplicações, plataformas e infraestrutura	Não entendi a pergunta ou não conheço o assunto.







3. Os membros dos times de desenvolvimento, qualidade e operações se conhecem?					
					 CONFUSED
Os membros não se conhecem pessoalmente.	Os membros se conhecem mas nunca trabalharam juntos.	Os membros se conhecem e interagem apenas no lançamento ou se ocorre um problema.	Agilidade fim-a-fim entre todas as equipes, aplicações, plataformas e infraestrutura	Todos os membros se conhecem e trabalham de forma coesa e colaborativa.	Não entendi a pergunta ou não conheço o assunto.

4. Há eventos regulares na organização que integrem as equipes ?					
					 CONFUSED
Não existem eventos de integração entre equipes	Raramente, apenas em eventos externos e não há compartilhamento do conhecimento.	Os times realizam eventualmente, talks, workshops internamente, mas não é compartilhado.	Os times realizam regularmente talks, workshops internamente e algumas pessoas de fora são convidadas.	Os times realizam regularmente, talks, workshops divulgados para toda empresa.	Não entendi a pergunta ou não conheço o assunto.







5. Existe colaboração para a construção de conhecimento compartilhado?					
					 CONFUSED
Não existe compartilhamento de conhecimento entre as equipes Devs e Ops.	O compartilhamento entre as equipes é limitado a e-mail e mensagens.	A equipe participa das Daily Meetings	Acontecem cerimônias de compartilhamento regularmente.	A gestão de conhecimento é unificada	Não entendi a pergunta ou não conheço o assunto.

2.2 Planejamento







6. Como os Requisitos Funcionais são coletados junto ao cliente?					
					 CONFUSED
São coletados em ferramentas Office e ficam de posse do Líder do Projeto.	São coletados e versionados manualmente em repositórios.	São coletados usando as tarefas do Jira para registrar o requisito.	São coletados usando ferramentas que permitem a rastreabilidade do requisito.	São coletados usando ferramentas integradas entre Dev, QA e Ops.	Não entendi a pergunta ou não conheço o assunto.

7. Como é realizado o acompanhamento do projeto ?					
					 CONFUSED
Cada projeto possui sua própria forma de realizar o acompanhamento.	Utilizando ferramentas Oce (xls, doc), email e whatsapps.	Acompanham ento através de cronogramas (Project Builder).	Backlogs gerenciados pelo Jira usando processos ágeis, com sprints bem definidas.	Uso de ferramentas de backlogs, Kanban, abrangendo várias equipes.	Não entendi a pergunta ou não conheço o assunto.

8. Como são priorizados os requisitos do projeto ?					
					 CONFUSED
De forma pontual para cada projeto, sem ferramenta.	Requisitos são priorizados pelo Líder do Projeto por não definição de Gestor.	Requisitos são priorizados entre os Gestores, Líder do Projeto sem envolvimento do time.	Requisitos são priorizados entre os Gestores, e Líder do Projeto com envolvimento dos devs.	Todos os envolvidos (dev+ops) participam da priorização dos requisitos registrando no Jira.	Não entendi a pergunta ou não conheço o assunto.







9. Como o trabalho é priorizado entre a equipe?					
					 CONFUSED
Não há um processo definido.	Cada equipe possui seu processo de priorização das atividades.	O líder do projeto define as atividades que cada membro irá trabalhar de acordo com as habilidades.	O líder do projeto define as atividades que cada membro irá trabalhar de acordo com as habilidades.	O time envolvido (Devs+Ops+Qas) define junto a priorização das atividades.	Não entendi a pergunta ou não conheço o assunto.

10. Todos os envolvidos conseguem visualizar os requisitos e o status do projeto em um dashboard integrado






					 CONFUSED
Não existe uma visualização dos requisitos e do status do Projeto.	Requisitos e Status do Projeto são visualizados através de planilhas compartilhadas.	Cada time possui seu próprio painel de controle.	Quadros digitais compartilhados entre Gestores/Lideres/Devs/Qas.	Quadros digitais compartilhados entre Gestores/Lideres/Devs/Qas/Ops.	Não entendi a pergunta ou não conheço o assunto.







2.3 Feedback







11. Existe alguma análise, orientação ou relatório produzido para criar um feedback para os times?







					 CONFUSED
Não há relatórios.	Coleta de dados e relatórios são manuais.	Envio de alertas por scripts, api, e-mail e outros, mas não há análises.	Uso de ferramenta de análise, mas o feedback não é usado para a melhoria contínua.	Uso de ferramenta de análise provendo feedback para melhoria contínua.	Não entendi a pergunta ou não conheço o assunto.

12. Quando o feedback é coletado?







					 CONFUSED
Somente quando há uma crise.	Apenas durante crises e pós-implantação.	Durante crises, pós-implantação e testes.	Durante crises, pós-implantação, testes e desenvolvimento.	Existe a opção de prover feedback a qualquer momento, durante todas as fases.	Não entendi a pergunta ou não conheço o assunto.

13. Como você avalia a experiência do usuário?					
					 CONFUSED
Não existe avaliação da experiência do usuário com nossos serviços.	Avaliação só é realizada quando Gestor ou Qas apontam não conformidades.	Apenas usabilidade é avaliada.	Prototipagem para validar e refinar conceitos.	Através de testes A/B em produção, feedback dos usuários, métricas e pesquisas.	Não entendi a pergunta ou não conheço o assunto.







14. Como o feedback do usuário é analisado e otimizado para melhoria contínua?					
					 CONFUSED
Não existe processo definido.	O feedback é analisado quando há retorno dos usuários finais.	São realizadas pesquisas pontuais com os usuários finais.	Uso de técnicas como Análise de Causa Raiz, entrevistas, métricas.	Uso de métodos como Design Thinking.	Não entendi a pergunta ou não conheço o assunto.

15. Existe compartilhamento do feedback entre os times? Como isto é praticado?					
					 CONFUSED
Não são compartilhados.	Apenas quando ocorrem problemas.	Periodicamente.	Somente durante as releases, mas NÃO são monitorados.	Time coeso recebendo feedback contínuo.	Não entendi a pergunta ou não conheço o assunto.






2.4 Integração Contínua







16. Como o seu código-fonte é gerenciado e como é feito o controle de versão?					
					 CONFUSED
Não existe controle de versão.	Utiliza sistema de controle de versão (SVN/GIT) sem pipeline integrado.	Políticas para gerenciamento das bibliotecas e dependências.	Desenvolvedores trabalham na master e são usadas branches para novas releases.	Alterações auditáveis, integração com Jira, para rastreabilidade e do código.	Não entendi a pergunta ou não conheço o assunto.

17. Como são controladas e medidas as dívidas técnicas?					
					 CONFUSED
A dívida técnica não é revisitada.	Em um período pré-definido, não vinculado às releases.	Medimos a cada release de produção.	Medimos a cada build	De forma contínua	Não entendi a pergunta ou não conheço o assunto.







18. Como é o processo de integração, construção, deploy e entrega?					
					 CONFUSED
Processo não definido.	Código em repositório, ferramenta de compilação com gerência de dependências, servidor de build automatizado sem testes.	Código em repositório, ferramenta de compilação com gerência de dependências, servidor de build automatizado com execução dos testes automatizado	Walk + Inspeção automatizada de qualidade do código (cobertura de testes + métricas básicas de qualidade de código)	Run + abertura automática de ticket de defeito em caso de falha na qualidade do código da feature.	Não entendi a pergunta ou não conheço o assunto.







		s e notificação ao time (devs) de eventuais quebras no build			
--	--	--	--	--	--







19. Como é o processo de gerenciamento de qualidade do código e build ?					
					 CONFUSED
Não há um processo de qualidade do código.	Processo manual de revisão do código, sem ferramenta de monitoramento de alteração de código fonte.	Processo manual de revisão do código, com ferramenta de monitoramento de alteração de código.	Processo automático de revisão de código, SEM integração com ferramentas de revisão de código (SonarQube).	Processo automático e integrado com a ferramentas de build (Maven) e de revisão de código (SonarQube).	Não entendi a pergunta ou não conheço o assunto.

20. É realizado teste unitário durante a fase de desenvolvimento?					
					 CONFUSED
Não há teste unitário definido.	Testes unitários são realizados apenas em funcionalidades definidas pelos desenvolvedores	Testes unitários são realizados apenas em funcionalidades críticas definidas pelo gestor.	É automatizado, com cobertura entre 40 e 50%	É automatizado, com cobertura entre 80 e 90%	Não entendi a pergunta ou não conheço o assunto.







2.5 Entrega Contínua

21. Como é o processo de implantação?					
					
Processo manual para implantação do produto nos ambientes.	Implementação automatizada para alguns ambientes. Toda a configuração versionada	Totalmente automatizado . Aperta apenas um botão para promover os ambientes.	Implementações gerenciadas. Processos testados com possibilidade de rollback.	Todos os ambientes geridos de forma eficaz. Provisionamento automatizado	Não entendi a pergunta ou não conheço o assunto.







22. O processo de implantação da release é replicável?					
					
Replicação restrita, processos separados.	Replicável, mas coordenado manualmente	Replicável, exceto para grandes mudanças.	Totalmente replicável para todas as releases	Totalmente replicável, medidos e continuamente melhorados	Não entendi a pergunta ou não conheço o assunto.

23. Como é o processo de controle de atualização (promoção) de release em produção?					
					
Não há controle	Segue o procedimento de QA para aprovação	Requer aprovação gerencial, promoção manual.	Uso de experiência para promoção gradual da nova release	Pipeline automático, garante os check-in para cada release de forma segura.	Não entendi a pergunta ou não conheço o assunto.

24. Utiliza alguma a técnica (Canary Release, Blue-Green, Featured Toggle) de deploy para a implantação







					 CONFUSED
Nunca ouvir falar	Conheço, mas não usamos	Implantamos tudo de uma vez e torcemos para dar certo	Fazemos algo parecido, mas de forma manual.	Implementamos pelo menos uma técnica de forma automatizada	Não entendi a pergunta ou não conheço o assunto.







25. Com que frequência são implantadas novas funcionalidades em produção?







					 CONFUSED
Demora mais de um mês	Pelo menos uma vez ao mês	Pelo menos uma vez a cada 15 dias	Pelo menos uma vez por semana	É possível implantar uma ou mais vezes ao dia	Não entendi a pergunta ou não conheço o assunto.







2.6 Infraestrutura como código

26. Qual o tempo necessário para disponibilizar o ambiente?







					 CONFUSED
Ambiente físico até 40 horas.	Ambiente virtualizado, até 16 horas	Ambiente virtualizado, até 8 horas	Ambiente virtualizado, de 4 a 6 horas	Ambiente em cloud, de 2 a 4 horas	Não entendi a pergunta ou não conheço o assunto.

27. Como são definidos os ambientes para as fases de Desenvolvimento, Homologação e Produção?					
					 CONFUSED
Máquinas físicas exclusivas.	Máquinas virtuais exclusivas	Ambiente virtual para laboratórios	Uso de Paas (openshift) , uso de imagens de produção, sem controle individual	Uso de infraestrutura com o código (Ansible, Puppet).	Não entendi a pergunta ou não conheço o assunto.

28. Qual o nível de automação do processo de provisionamento de ambientes?					
					 CONFUSED
Não é automatizado	Criação de VM por scripts, demais tarefas manuais.	Somente o sistema operacional, as demais configurações são manuais	Somente infraestrutura básica (sistema operacional e middleware) de forma automatizada em único processo de orquestração.	Totalmente automatizado e orquestrado (Openshift)	Não entendi a pergunta ou não conheço o assunto.







29. A entrega dos ambientes é considerada durante o planejamento do projeto?					
					 CONFUSED
O impacto não é considerado durante o planejamento do projeto	Pode ser afetado pela disponibilidade de recursos.	Requer gerenciamento entre Dev e Ops para disponibilização do ambiente	Ambiente virtual solicitado e configurado por Ops.	Serviço disponível e configurável pelos arquitetos sob demanda (Openshift).	Não entendi a pergunta ou não conheço o assunto.

30. Como os problemas são informados aos times de desenvolvimento e operações?







					 CONFUSED
Não há estratégia definida	Através de reclamação dos usuários finais (telefone, email, helpdesk, líderes, redes sociais)	Uso de ferramenta de service desk (iplanfácil)	Uso de ferramenta de monitoração	Por ferramenta de Telemetria disponível para todas as equipes em tempo real.	Não entendi a pergunta ou não conheço o assunto.







2.7 Monitoração







31. Como é feito o gerenciamento de log, eventos e incidentes?







					 CONFUSED
Não há um processo de gerenciamento de logs/eventos ou incidentes.	Comunicação à central e investigação de log sob demanda.	Uso de ferramentas distintas para logs e incidentes sem integração.	Uso de ferramentas centralizadas.	Uso de ferramentas centralizadas com integração automática com ferramentas de gestão.	Não entendi a pergunta ou não conheço o assunto.

32. O pipeline permite identificar rapidamente as falhas para tomada de decisão do time?







					 CONFUSED
Não existe um pipeline implementado	Existe um pipeline, mas sem validação da execução	Existe um pipeline com validações, mas sem testes automatizados	Existe um pipeline com validações com testes automatizados.	Pipeline totalmente integrado com todos os testes, rollback de banco e aplicação.	Não entendi a pergunta ou não conheço o assunto.







33. A equipe de Operações é envolvida no início do projeto, para opinar sobre ambientes, implantação e planejamento de capacidade?					
					 CONFUSED
Ops não são envolvidos.	Apenas quando necessário, caso a caso	Apenas para prover informações.	Em reuniões de planejamento, é envolvido ativamente.	O time DevOps trabalha de forma coesa e colaborativa.	Não entendi a pergunta ou não conheço o assunto.







34. A sua aplicação e infraestrutura são auto escaláveis?					
					 CONFUSED
A aplicação e infraestrutura não são escaláveis	A aplicação é escalável, mas a infra não permite	Requer adição de recursos a infraestrutura	De forma manual, com base em parâmetros de limite	De forma automática, com base em parâmetros configuráveis, como CPU/memória.	Não entendi a pergunta ou não conheço o assunto.







35. As equipes são capazes de analisar métricas de performance para solucionar problemas antes que eles afetem os usuários?					
					 CONFUSED
Não existem métricas implementadas.	Só é avaliado sob demanda.	Parcialmente implementado por código, mas avaliado sob demanda	Parcialmente implementado por código, com análises constantes.	Totalmente implementado por ferramenta de appMon	Não entendi a pergunta ou não conheço o assunto.







2.8 Testes

36. Qual o percentual de testes automatizados realizados em todas as fases de desenvolvimento?					
					 CONFUSED
Não existem testes automatizados.	Até 20%	Até 40%	Até 60%	Acima de 60%	Não entendi a pergunta ou não conheço o assunto.

37. Em que momento são elaborados os casos de testes?					
					 CONFUSED
Os requisitos são definidos sem envolvimento dos QAs	QAs participam sob demanda das reuniões de planning para conhecer os requisitos	QAs participam sob demanda das reuniões de planning para conhecer os requisitos	QAs faz parte do time e elabora os casos em paralelo ao desenvolvimento.	QAs participam desde o início, utilizando técnicas como BDD ou TDD com testes automatizados no pipeline.	Não entendi a pergunta ou não conheço o assunto.

38. Com que frequência é realizado o teste de funcionalidade?					
					 CONFUSED
Apenas sob demanda.	Somente ao final da release.	Na entrega de cada sprint.	QA faz parte do time e realiza os testes antes da entrega da sprint.	QA faz parte do time e realiza os testes antes da entrega da sprint.	Não entendi a pergunta ou não conheço o assunto.

39. Quando são realizados os testes não funcionais (testes de vulnerabilidade, desempenho, sonar, etc)?					
					 CONFUSED
Apenas quando solicitado.	Obrigatório, sempre antes da entrada em produção.	Antes da implantação com automação parcial	A cada release com automação parcial	Testes não funcionais automatizados no pipeline	Não entendi a pergunta ou não conheço o assunto.

40. O desenvolvedor compartilha o resultado dos testes com os QAs?					
					 CONFUSED
Não há casos de testes	Somente quando solicitados, não há casos formais	Executado manualmente pelo Dev e não gera relatório	Executados manualmente pelo Dev, compartilha um relatório em cada release.	Executado automaticamente pelo pipeline e disponível em painel de controle.	Não entendi a pergunta ou não conheço o assunto.

APÊNDICE E - Mapeamentos das Dores e Necessidades

O que podemos resolver com Devops, baseado no **escopo** do nosso projeto. O quadro está disponível no seguinte link:

https://miro.com/app/board/o9J_kpRZrMk=/

1. Cultura

Calms

CULTURE (Cultura)

- Foco nas pessoas e em como elas pensam
- Adesão de uma cultura de experimentação
- Promoção de uma cultura de integração entre desenvolvimento e operações
- Mudança de Comportamento
- Bons relacionamentos entre as áreas
- Responsabilidade compartilhada entre equipes
- Barreiras: adesão de uma cultura livre de culpa por erros nos processos

No Café com Devops ouvimos das equipes que

1. Otimizar o fluxo de trabalho	2. Melhorar a comunicação entre equipes	3. Reduzir o tempo de entrega	4. Aumentar a transparência	5. Melhorar a segurança	6. Melhorar a qualidade do código	7. Melhorar a colaboração	8. Melhorar a automação	9. Melhorar a documentação	10. Melhorar a formação de novos membros
11. Melhorar a segurança	12. Melhorar a qualidade do código	13. Melhorar a colaboração	14. Melhorar a automação	15. Melhorar a documentação	16. Melhorar a formação de novos membros	17. Melhorar a segurança	18. Melhorar a qualidade do código	19. Melhorar a colaboração	20. Melhorar a automação
21. Melhorar a documentação	22. Melhorar a formação de novos membros	23. Melhorar a segurança	24. Melhorar a qualidade do código	25. Melhorar a colaboração	26. Melhorar a automação	27. Melhorar a documentação	28. Melhorar a formação de novos membros	29. Melhorar a segurança	30. Melhorar a qualidade do código
31. Melhorar a colaboração	32. Melhorar a automação	33. Melhorar a documentação	34. Melhorar a formação de novos membros	35. Melhorar a segurança	36. Melhorar a qualidade do código	37. Melhorar a colaboração	38. Melhorar a automação	39. Melhorar a documentação	40. Melhorar a formação de novos membros
41. Melhorar a segurança	42. Melhorar a qualidade do código	43. Melhorar a colaboração	44. Melhorar a automação	45. Melhorar a documentação	46. Melhorar a formação de novos membros	47. Melhorar a segurança	48. Melhorar a qualidade do código	49. Melhorar a colaboração	50. Melhorar a automação

entendemos as Dores e Necessidades



o Assessment complementou com a visão de onde queremos chegar



e para isso os seguintes Planos de Ações serão realizados



2. Automação

Calms

AUTOMATION (Automação)

- Automatize quase tudo que for possível;
- Dê ferramentas para manter tudo aquilo que for necessário para que o processo de desenvolvimento, testes e releases seja um sucesso;
- Entrega Contínua
- Integração Contínua
- Infra como código
- Orquestração

No Café com Devops ouvimos das equipes que

The sticky notes contain the following text:

- 1. A maioria dos sistemas são MV, testes unitários
- 2. DevOps é uma cultura, não é uma ferramenta
- 3. Não se trata de ferramentas, mas de processos e cultura
- 4. A maioria de testes não tem nada em comum
- 5. Não se trata de ferramentas, mas de processos e cultura
- 6. DevOps é uma cultura, não é uma ferramenta
- 7. Não se trata de ferramentas, mas de processos e cultura
- 8. Não se trata de ferramentas, mas de processos e cultura
- 9. Não se trata de ferramentas, mas de processos e cultura
- 10. Não se trata de ferramentas, mas de processos e cultura
- 11. Não se trata de ferramentas, mas de processos e cultura
- 12. Não se trata de ferramentas, mas de processos e cultura
- 13. Não se trata de ferramentas, mas de processos e cultura
- 14. Não se trata de ferramentas, mas de processos e cultura
- 15. Não se trata de ferramentas, mas de processos e cultura

entendemos as Dores e Necessidades

The sticky notes contain the following text:

- 1. A maioria de testes não tem nada em comum
- 2. Não se trata de ferramentas, mas de processos e cultura
- 3. Não se trata de ferramentas, mas de processos e cultura
- 4. Não se trata de ferramentas, mas de processos e cultura
- 5. Não se trata de ferramentas, mas de processos e cultura

o Assessment complementou com a visão de onde queremos chegar



e para isso os seguintes Planos de Ações serão realizados



3. Lean

Calms

LEAN
(Fluxo)

- Mantenha o foco no cliente e no valor gerado para ele
- Elimine todos os focos de desperdício
- Adote a entrega contínua
- Entregas pequenas e contínuas
- Fazer o simples
- Foque no aumento de aprendizagem.

No Café com Devops ouvimos das equipes que

1. Definir o propósito do trabalho e o valor que será entregue ao cliente

2. Definir a estratégia de entrega contínua

3. Definir o modelo de trabalho e o fluxo de trabalho

4. Definir o modelo de trabalho e o fluxo de trabalho

5. Definir o modelo de trabalho e o fluxo de trabalho

6. Definir o modelo de trabalho e o fluxo de trabalho

7. Definir o modelo de trabalho e o fluxo de trabalho

8. Definir o modelo de trabalho e o fluxo de trabalho

9. Definir o modelo de trabalho e o fluxo de trabalho

10. Definir o modelo de trabalho e o fluxo de trabalho

11. Definir o modelo de trabalho e o fluxo de trabalho

12. Ter uma definição de pronto

13. Definir o modelo de trabalho e o fluxo de trabalho

14. Definir o modelo de trabalho e o fluxo de trabalho

15. Definir o modelo de trabalho e o fluxo de trabalho

16. Definir o modelo de trabalho e o fluxo de trabalho

entendemos as Dores e Necessidades

1. Revisão do processo do ciclo de vida do produto end-to-end

2. Definir o modelo de trabalho e o fluxo de trabalho

3. Definir o modelo de trabalho e o fluxo de trabalho

4. Definir o modelo de trabalho e o fluxo de trabalho

5. Definir o modelo de trabalho e o fluxo de trabalho

6. Definir o modelo de trabalho e o fluxo de trabalho

o **Assessment** complementou com a visão de onde queremos chegar

Planejamento



e para isso os seguintes **Planos de Ações** serão realizados



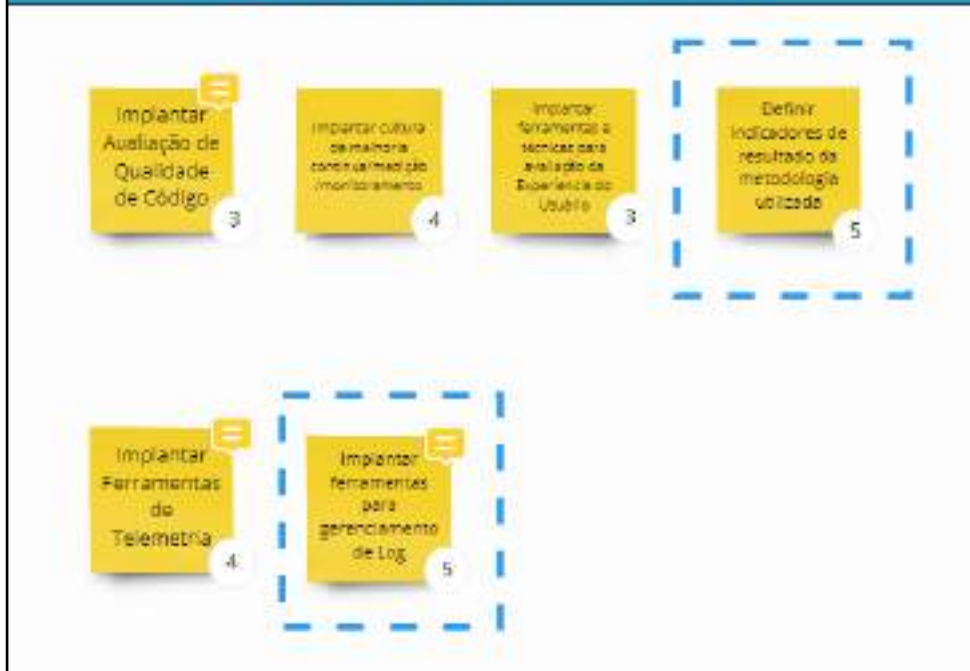
4. Medição

Calms

MEASUREMENT (Medição)

- Mensure tudo que for possível
- Apresente melhorias constantes
- Adote a melhoria contínua em seu negócio
- Adote o monitoramento e controle
- Realize análises de desempenho constantemente

e para isso os seguintes Planos de Ações serão realizados



5. Compartilhamento

Calms

SHARING (Compartilhamento)

- Cultura do compartilhamento
- Adesão de maior colaboração entre membros
- Cultura da transparência
- Feedback contínuo para os times
- Compartilhe descobertas, Lições Aprendidas e Ferramentas, para criar um ambiente de aprendizagem organizacional.

No Café com Devops ouvimos das equipes que



entendemos as Dores e Necessidades



o Assessment complementou com a visão de onde queremos chegar



e para isso os seguintes Planos de Ações serão realizados



APÊNDICE F - Pipeline da Equipe Java

Durante a reunião com os Arquitetos de Soluções, foi apresentado o pipeline e suas ferramentas, que é utilizado atualmente na equipe Java.

De forma bem simplificada pode-se dizer que a Pipeline consiste em todas as etapas realizadas de forma automatizada para levar o código do repositório até produção. A Pipeline torna o trabalho da equipe mais efetivo, reduzindo significativamente os riscos e as taxas de erros em produção, pois evita o acúmulo de bugs e qualquer não conformidade evita que o código avance uma etapa da esteira. Com um processo amadurecido é possível garantir que as entregas não sejam mais semanais, quinzenais ou mensais e sim frequentes durante o dia.



1. Código

Nesta etapa, o Time de desenvolvimento desenvolve as features em branches. Quando a funcionalidade está pronta ela é integrada no master (push). Esta ação inicia o GitLab CI, que é o ambiente de Integração Contínua do GitLab que dá suporte a Integração Contínua e Entrega contínua.

Ferramentas usadas

Desenvolvimento: Java + Wildfly ou Quarkus, Flyway

Ambiente local: Docker, subindo imagens com os servidores, banco de dados.

Repositório de código-fonte e artefatos: Gitlab

2. Gitlab CI

O GitLab CI possui três estágios padrões, porém não obrigatórios. Os nomes que você escolhe aqui são arbitrários, mas a ordenação determina a ordem de execução dos passos que se seguirão. Stages ou estágios são tags que você pode aplicar a jobs individuais. O GitLab vai executar jobs do mesmo estágio em paralelo e vai esperar para executar o próximo estágio até que todos os jobs do estágio atual estejam completos. Se nenhum estágio for definido, o GitLab usará três estágios chamados build, test e deploy.

Estas etapas ficam configuradas no arquivo `.gitlab-ci.yml`, que basicamente é um script com todas as etapas que o pipeline deve rodar. Quando o desenvolvedor faz o push na master, este evento inicia o script. Se não houver runners disponíveis, a execução da CI será definida como “pending” ou pendente. Os runners executam os Jobs utilizando as imagens docker para provisionar os ambientes e retornar o resultado para o gitlab.



2.1 Build

O primeiro passo do script do pipeline é realizar o build da aplicação usando o *Maven*, que é responsável por compilar todo código usando o Nexus Repository para gerenciar as dependências. Por fim, a imagem é criada e enviada para o *Container Registry*. Sempre que há alguma alteração na *branch master* ou em uma **tag**, esse processo é iniciado, uma nova imagem é criada e enviada para o *Container Registry* do Gitlab.

Os scripts de banco, também conhecidos como *migrations*, são executados. Para isso é usado o Flyway que é responsável pela organização dos scripts SQL

executados no banco de dados. As *migrations* possuem apenas a estrutura das tabelas e dessa forma o código e estrutura de banco passam a ser uma coisa só.

Colocando ambos no pipeline, mitigamos o problema de atualizar a versão e não ter a atualização no banco de dados necessária aplicada.

Ferramentas utilizadas

- Build: Maven
- Dependências: Nexus Repository
- Scripts: Flyway
- Container: Docker
- Container Repository: Container Registry

2.2 Testes

Atualmente o pipeline não executa os testes automatizados, indo direto para a fase de release e deploy da aplicação. Esse é um ponto de melhoria definido para o projeto.

2.3 Release/Deploy

Após o build da aplicação, a estrutura do projeto é montada utilizando *Kubernetes* que é responsável por orquestrar e gerenciar os containers. Por fim é feito o deploy no ambiente Openshift e a aplicação está disponível para uso.

Ferramentas utilizadas

- Terraforms
- Ansible
- Kubernetes

APÊNDICE G - Indicadores de Performance e Métricas

1. Indicadores de performance e métricas

É preciso pensar com cuidado sobre como as métricas e indicadores de performance (KPIs) podem ser interpretados pelos times técnicos, executivos da empresa e clientes gestores. Uma vez que as estatísticas geralmente não são acompanhadas por uma narrativa, elas podem ficar sujeitas a interpretações imprecisas.

As métricas são projetadas para facilitar a visão sobre a qualidade e como melhorá-la, por isso, elas precisam medir efetivamente o impacto dos produtos e serviços para os clientes e os negócios. As três métricas críticas para atender às expectativas incluem velocidade, eficiência e mitigação de risco, ou seja: melhorar a velocidade com a qual sua organização entrega mudanças para reduzir o tempo de ciclo com objetivo de obter vantagem competitiva; melhorar e aumentar a eficiência das equipes e garantir que as alterações sejam aderentes às políticas de conformidade.

Medir é importante, mas tenha ciência de que algumas métricas são difíceis de coletar e mensurar, como por exemplo a cultura e a colaboração, que não podem ser medidas diretamente.

Abaixo seguem algumas métricas de mercado que podemos avaliar e experimentar futuramente para medir o quanto DevOps pode ajudar a equipe ou a empresa na resolução de problemas:

- **Produtividade/Eficiência operacional:** custo por release; custo por transação; tempo médio para resolução de um problema; tempo de resposta para um problema.
- **Qualidade:** taxa de sucesso de deploy; bugs na produção; tempo médio por atividade (task/issue/bug); quantidade de Incidentes.
- **Velocidade:** tempo de Ciclo (Sprint + Test + Deploy); frequência de deploy; valor por release; performance por recurso (diária e por release).
- **Efetividade organizacional:** colaboração; motivação (felicidade, satisfação dos funcionários); retenção; número de lições aprendidas;

número de pesquisa e inovação aplicadas em projetos; número de workshops/apresentações de DevOps/Inovação/Transformação Digital.

- **Valor para o cliente:** tempo de espera para solução de problemas; quantidade de histórias entregues versus quantidade de histórias em backlog.
- **Desempenho dos negócios:** lucros por funcionalidade entregue; fatia de mercado; novos clientes; margem bruta; novos mercados; objetivos estratégicos alcançados; reputação da marca; experiência do usuário.

No plano de ação temos atividades para definição de indicadores e métricas que serão utilizados para o projeto piloto.

APÊNDICE H - Planning Poker

1. Cultura



Reforçar Metodologia de Desenvolvimento e papéis e responsabilidades



Propor ações educacionais que têm por finalidade, disseminar a importância dos Testes Automatizados



Propor ações educacionais que têm por finalidade, disseminar a importância da Experiência do Usuário



- 3
25 % (1 player)
- 4
25 % (1 player)
- 5
50 % (2 players)

Propor ações educacionais que têm por finalidade, disseminar, manter e evoluir uma cultura Devops



- 3
25 % (1 player)
- 4
25 % (1 player)
- 5
50 % (2 players)

2. Automação



Integrar Testes de API no Pipeline



Integrar Testes Funcionais e Aceitação no Pipeline



Integrar Testes Unitários no Pipeline



Implantar práticas de automação do provisionamento dos ambientes



Implantar Modelo de Entrega Contínua



Definir e implantar um processo de melhoria contínua



3. Lean



4. Medição



Implantar Ferramentas de Telemetria



Implantar ferramentas/tecnicas para avaliação da Experiencia do Usuário



Implantar cultura de melhoria continua e medição/monitoramento:



- 4
33.3 % (1 player)
- 5
66.7 % (2 players)

Implantar Avaliação de Qualidade de Código:



- 4
33.3 % (1 player)
- 5
66.7 % (2 players)

4. Compartilhamento

