

**EMPRESA MUNICIPAL DE INFORMÁTICA S.A. - IPLANRIO
PROGRAMA DE DESENVOLVIMENTO DE LIDERANÇAS – PDL**

**INSTITUTO FUNDAÇÃO JOÃO GOULART – FJG
COORDENADORIA DO PROGRAMA DE CAPACITAÇÃO EM GESTÃO – CPCG**

JORNADA DEVOPS NA IPLANRIO

**Adriana Soares Morales, matrícula 45/620.932-4
Katia Leituga Martins de Carvalho, matrícula 45/621.070-3
Luciana Nascimento Santos, matrícula 45/622.373-4
Simone Ferreira Machado da Silva, matrícula 69/623.027-1
Tatiane Carrilho Ribeiro, matrícula 45/622.021-6**

**Rio de Janeiro,
13 de novembro de 2020.**

EMPRESA MUNICIPAL DE INFORMÁTICA – IPLANRIO
INSTITUTO FUNDAÇÃO JOÃO GOULART – FJG
COORDENADORIA DO PROGRAMA DE CAPACITAÇÃO EM GESTÃO – CPCG

PROGRAMA DE DESENVOLVIMENTO DE LIDERANÇAS – PDL
PROJETO FINAL – TURMA 1

JORNADA DEVOPS NA IPLANRIO

Trabalho orientado pelo Prof. Vinícius de Oliveira, especialmente elaborado como Projeto Final para a obtenção do certificado de conclusão do Programa de Desenvolvimento de Liderança – PDL da IPLANRIO.

Trabalho elaborado por:

Adriana Soares Morales, matrícula 45/620.932-4
Katia Leituga Martins de Carvalho, matrícula 45/621.070-3
Luciana Nascimento Santos, matrícula 45/622.373-4
Simone Ferreira Machado da Silva, matrícula 69/623.027-1
Tatiane Carrilho Ribeiro, matrícula 45/622.021-6

Rio de Janeiro,
13 de novembro de 2020.

DEDICATÓRIA

Dedicamos este trabalho aos nossos familiares que demonstraram compreensão com as nossas horas de ausência.

AGRADECIMENTOS

Agradecemos a todos os professores que influenciaram a nossa jornada, em especial ao nosso padrinho Carlos Henrique, nosso orientador Vinícius, a coordenadora do Programa de Capacitação em Gestão da Fundação João Goulart Bárbara e aos colegas de trabalho que de alguma forma contribuíram para a realização desse projeto.

RESUMO

A adoção de novas tecnologias e o ritmo de inovação no mundo atual tem se acelerado exponencialmente. O que é válido hoje, pode não ser em alguns meses, especialmente no contexto atual de um mundo de incertezas, dentro e fora das empresas, ou seja, um ambiente de volatilidade (*volatility*), incerteza (*uncertainty*), complexidade (*complexity*) e ambiguidade (*ambiguity*). Em um mercado cada vez mais exigente que precisa lidar com mudanças constantes, o DevOps se encaixa perfeitamente. A sua capacidade de integrar os diferentes papéis dentro de uma equipe de tecnologia, fazendo com que todos trabalhem de forma colaborativa com entregas em menor tempo, com mais qualidade e menor custo, é crucial neste contexto. Esse modelo de atuação tem um foco muito forte em questões comportamentais e culturais, envolvendo iniciativas para que haja integração entre as equipes. Este trabalho tem como finalidade entender o nível de maturidade da empresa e propor um modelo de disseminação da cultura DevOps na Iplanrio. Diante disso, foi realizada uma abordagem qualitativa exploratória, onde foram coletados dados por meio de pesquisa e entrevistamos profissionais de perfis diversos dentro da empresa. Como resultado deste trabalho, identificamos que na Iplanrio existem iniciativas na implementação de métodos ágeis, mas que ainda temos um grande caminho a percorrer na disseminação da cultura DevOps, pois existe uma baixa maturidade. Em complemento a esta avaliação, um conjunto de ações foram mapeadas e priorizadas para a elaboração do plano de ação para execução do projeto piloto evoluindo a maturidade DevOps dentro da empresa e que este sirva como base para ações futuras de expansão dessa cultura para outras equipes.

SUMÁRIO

1. INTRODUÇÃO	10
1.1 O Problema	10
1.1.1	12
1.1.2 Riscos de uma Implantação inadequada	12
1.2 Hipótese	13
1.3 Justificativa	13
1.4 Escopo	14
1.5 Objetivos	14
1.5.1 Objetivos Gerais	14
1.5.2 Objetivos Específicos	15
2. METODOLOGIA	16
2.1 Referências Literárias	16
2.2 Benchmarking	17
2.2.1 Promove Soluções	17
2.2.2 Tribunal de Justiça do Estado do Rio de Janeiro	18
2.2.3 RedHat	20
2.3 Café com DevOps	21
2.3.1 Café com DevOps 1 - Apresentação	22
2.3.2 Café com DevOps 2 - Revisão do Processo de Desenvolvimento	23
2.3.3 Café com DevOps 3 - Avaliação do nível de maturidade DevOps	25
2.4 Talk com especialistas	28
2.4.1 Arquitetura de Soluções Java	28
2.4.2 Qualidade de Software	30
2.4.3 Workshop de Testes Automatizados	32
2.4.4 UX Designer	34

2.4.5 Operações	36
2.4.6 Desenvolvimento	38
2.4.7 Analistas de Negócio	39
3. RESULTADOS	42
3.1 Assessment	42
3.1.1 Perfil dos Participantes	42
3.1.2 Resultados por Práticas DevOps	43
3.1.3 Radar de Maturidade	67
3.2 Mapeamento das Dores e Necessidades	69
3.2.1 Cultura (Culture)	69
3.2.2 Automação (Automation)	72
3.2.3 Fluxo (Lean)	75
3.2.4 Medição (Measurement)	78
3.2.5 Compartilhamento (Sharing)	80
3.3 Plano de Ação	83
3.3.1 Projeto Piloto	85
3.3.2 Ações Evolutivas	89
3.3.3 Ações para Empresa	89
4. CONSIDERAÇÕES FINAIS	91
4.1 CONCLUSÃO	91
4.2 RECOMENDAÇÕES	94
5. REFERÊNCIA BIBLIOGRÁFICA	95
GLOSSÁRIO	97
APÊNDICE A - Framework Calms	101
APÊNDICE B - Feedback das Reuniões	103
APÊNDICE C - Revisão do processo de desenvolvimento atual - Equipe Java	115
APÊNDICE D - Assessment	121

APÊNDICE E - Mapeamentos das Dores e Necessidades	154
APÊNDICE F - Pipeline da Equipe Java	163
APÊNDICE G - Indicadores de Performance e Métricas	166
APÊNDICE H - Planning Poker	168

LISTA DE FIGURAS

Figura 1 - Benchmarking com a Consulta da Promove Soluções Analia Irigoyen	18
Figura 2 - Tribunal de Justiça do Estado do Rio de Janeiro	19
Figura 3 - Equipe Devops	20
Figura 4 - Exemplo de feedback coletado nas reuniões.	22
Figura 5 - Visão Geral do Processo de Desenvolvimento Java	24
Figura 6 - Exemplo de Questão do Assessment do MIRO	26
Figura 7- Exemplo de questão preenchida no Formulário do Assessment	27
Figura 8 - Arquitetura de Soluções Java	29
Figura 9 - Ferramentas utilizadas no Pipeline de Implantação	30
Figura 10 - Qualidade de Software	31
Figura 11 - Workshop de Testes Automatizados	33
Figura 12 - UX Designer	35
Figura 13 - Operações	36
Figura 14 - Desenvolvimento	38
Figura 15 - Analista de Negócio	40
Figura 16 - Participação por Diretoria	42
Figura 17 - Perfil por Participantes	43
Figura 18 - Questão 1 - Você conhece os princípios básicos de DevOps e seus benefícios?	44
Figura 19 - Questão 2 - O processo de desenvolvimento adotado na sua equipe está alinhado com os princípios DevOps ?	44
Figura 20 - Questão 3 - Os membros dos times de desenvolvimento, qualidade e operações se conhecem?	45
Figura 21 - Questão 4 - Há eventos regulares na organização que integrem as equipes?	45
Figura 22 - Questão 5 - Existe colaboração para a construção de conhecimento compartilhado?	46
Figura 23 - Questão 6 - Como os Requisitos Funcionais são coletados junto ao cliente?	47
Figura 24 - Questão 7 - Como é realizado o acompanhamento do projeto?	47
Figura 25 - Questão 8 - Como são priorizados os requisitos do projeto?	48

Figura 26 - Questão 9 - Como o trabalho é priorizado entre a equipe?	48
Figura 27 - Questão 10 - Todos os envolvidos conseguem visualizar os requisitos e o status do projeto em um dashboard integrado?	49
Figura 28 - Questão 11 - Existe alguma análise, orientação ou relatório produzido para criar um feedback para os times?	50
Figura 29 - Questão 12 - Quando o feedback é coletado?	50
Figura 30 - Questão 13 - Como você avalia a experiência do usuário?	51
Figura 31 - Questão 14 - Como o feedback do usuário é analisado e otimizado para melhoria contínua?	51
Figura 32 - Questão 15 - Existe compartilhamento do feedback entre os times? Como isto é praticado?	52
Figura 33 - Questão 16 - Como o seu código-fonte é gerenciado e como é feito o controle de versão?	53
Figura 34 - Questão 17 - Como são controladas e medidas as dívidas técnicas?	53
Figura 35 - Questão 18 - Como é o processo de integração, construção, deploy e entrega?	54
Figura 36 - Questão 19 - Como é o processo de gerenciamento de qualidade do código e build?	54
Figura 37 - Questão 20 - É realizado teste unitário durante a fase de desenvolvimento?	55
Figura 38 - Questão 21 - Como é o processo de implantação?	56
Figura 39 - Questão 22 - O processo de implantação da release é replicável?	56
Figura 40 - Questão 23 - Como é o processo de controle de atualização (promoção) de release em produção?	57
Figura 41 - Questão 24 - Utiliza alguma a técnica de deploy para a implantação, como Canary Release, Blue-Green, Featured Toggle?	57
Figura 42 - Questão 25 - Com que frequência são implantadas novas funcionalidades em produção?	58
Figura 43 - Questão 26 - Qual o tempo necessário para disponibilizar o ambiente?	59
Figura 44 - Questão 27 - Como são definidos os ambientes para as fases de Desenvolvimento, Homologação e Produção?	59
Figura 45 - Questão 28 - Qual o nível de automação do processo de provisionamento de ambientes?	60

Figura 46 - Questão 29 - A entrega dos ambientes é considerada durante o planejamento do projeto?	60
Figura 47 - Questão 30 - Como os problemas são informados aos times de desenvolvimento e operações?	61
Figura 48 - Questão 31 - Como é feito o gerenciamento de log, eventos e incidentes?	62
Figura 49 - Questão 32 - O pipeline permite identificar rapidamente as falhas para tomada de decisão do time?	62
Figura 50 - Questão 33 -A equipe de Operações é envolvida no início do projeto, para opinar sobre ambientes, implantação e planejamento de capacidade?	63
Figura 51 - Questão 34 - A sua aplicação e infraestrutura são auto escaláveis?	63
Figura 52 - Questão 35 - As equipes são capazes de analisar métricas de performance para solucionar problemas antes que eles afetem os usuários?	64
Figura 53 - Questão 36 - Qual o percentual de testes automatizados realizados em todas as fases de desenvolvimento?	65
Figura 54 - Questão 37 - Em que momento são elaborados os casos de testes?	65
Figura 55 - Questão 38 - Com que frequência é realizado o teste de funcionalidade?	66
Figura 56 - Questão 39 - Quando são realizados os testes não funcionais (testes de vulnerabilidade, desempenho, sonar, etc)?	66
Figura 57 - Questão 40 - O desenvolvedor compartilha o resultado dos testes com os QAs?	67
Figura 58 - Radar de Maturidade nas Práticas DevOps	68
Figura 59 - Mapeamento dos feedbacks relacionados à Cultura	70
Figura 60 - Dores e Necessidades identificadas relacionadas à Cultura	71
Figura 61 - Resultado do assessment - Cultura	71
Figura 62 - Plano de Ação Cultura	72
Figura 63 - Mapeamento dos feedbacks relacionados à Automação	73
Figura 64 - Dores e Necessidades identificadas relacionadas à Automação	73
Figura 65 - Resultado do assessment - Automação	74
Figura 66 - Plano de Ação Automação	75
Figura 67 - Mapeamento dos feedbacks relacionados à Fluxo (Lean)	76
Figura 68 - Dores e Necessidades identificadas relacionadas à Fluxo (Lean)	76

Figura 69 - Resultado do assessment - Fluxo (Lean)	77
Figura 70 - Plano de Ação Fluxo (Lean)	77
Figura 71 - Mapeamento dos feedbacks relacionados à Medição	78
Figura 72 - Dores e Necessidades identificadas relacionadas à Medição	79
Figura 73 - Resultado do assessment - Medição	79
Figura 74 - Plano de Ação Medição	80
Figura 75 - Mapeamento dos feedbacks relacionados à Compartilhamento (Sharing)	81
Figura 76 - Dores e Necessidades identificadas relacionadas à Compartilhamento (Sharing)	82
Figura 77 - Resultado do assessment - Compartilhamento (Sharing)	82
Figura 78 - Plano de Ação Compartilhamento (Sharing)	82
Figura 79 - Planning Poker	84
Figura 80 - Plano de Ação Projeto Piloto	85

LISTA DE TABELAS

Tabela 1 - Plano de Ação do Projeto Piloto

86

LISTA DE ABREVIATURAS E SIGLAS

CTP – Coordenadoria Técnica de Gestão de Pessoas

DAF – Diretoria de Administração e Finanças

DOP – Diretoria de Operações

DPN – Diretoria de Planejamento e Novos Negócios

DSI – Diretoria de Sistemas

1. INTRODUÇÃO

DevOps é uma cultura fortemente colaborativa entre equipes com ações e práticas de automação cujo objetivo é acelerar a entrega de produtos com qualidade, considerando o ponto de vista de todos os envolvidos, fomentando que estes sejam genuinamente proprietários do resultado final. Adotar a abordagem DevOps é uma tarefa que exige que a organização realize mudanças culturais, tecnológicas e adequações em seu processo de desenvolvimento de software. O caminho para uma adoção bem sucedida é único para cada organização, portanto qualquer padrão deve ser adaptável e flexível.

O conceito de DevOps pode ser complexo, mas em essência, trata-se de uma prática que envolve cultura, ferramentas e processos ágeis. Ela resulta em um rearranjo de responsabilidades e colaboração entre setores de desenvolvimento e operações, que até então trabalhavam de forma isolada. Como resultados temos maior agilidade nos processos, entrega contínua de software com qualidade e colaboração apoiada por ferramentas.

DevOps parte do entendimento de que repressar mudanças implica em erros, problemas frequentes e recorrentes. Com uma nova proposta, adotar DevOps significa buscar constantemente a automação de processos repetitivos ao longo de toda a cadeia de produção de software. Isto implica no provisionamento “self-service” de recursos para as equipes, esteiras de construção, testes automatizados, entrega contínua, ferramentas que facilitem a comunicação, colaboração e monitoração, entre outros.

O termo “DevOps” surgiu na Conferência Velocity, em 2009, durante a apresentação de John Allspan e Paul Hammond intitulada “10+ Deploy Per Daye Dev and Ops Cooperation at Flickr”, que apresentou os resultados e desafios da maior aproximação entre as equipes de desenvolvimento e operações no Flickr. O impacto desta conferência resultou no primeiro DevOps Day, realizado na Bélgica, que formou a base para a disseminação da cultura DevOps pelo mundo. (“DevOps”).

1.1 O Problema

A Iplanrio é a empresa municipal responsável pela administração dos recursos de Tecnologia da Informação e Comunicação da cidade do Rio de Janeiro e tem como

missão prover soluções tecnológicas para melhorar a vida na cidade. Possui em seu portfólio diversas aplicações que atendem os serviços de todos os órgãos do município e como toda empresa de tecnologia enfrenta desafios no seu dia a dia.

Historicamente, e isto não é uma exclusividade da empresa, as áreas de infraestrutura e desenvolvimento possuem objetivos diferentes e muitas vezes conflitantes. A equipe de desenvolvimento está focada em entregar as funcionalidades dentro do prazo enquanto a equipe de infraestrutura está preocupada com a disponibilidade e segurança.

Eventualmente a aplicação é entregue para publicação em produção apresentando defeitos ou falhas de configuração, o que pode ocorrer por diversos motivos, por exemplo, divergências de ambientes.

Em muitos casos, a aplicação funciona de forma adequada no ambiente de desenvolvimento, mas quando em produção, ocorrem incidentes e conseqüentemente desgaste entre as equipes. Frases como *“na minha máquina funciona”* ou *“os desenvolvedores não codificam e não testam direito”* são comuns de serem ouvidas.

Um outro ponto a ser destacado é o tempo gasto atualmente com os testes das aplicações, que muitas vezes são realizados de forma manual. Estes processos poderiam ser automatizados, reduzindo os riscos associados à falha humana, otimizando o tempo e eliminando o retrabalho.

A seguir apresentamos alguns dos problemas comuns ao se implantar DevOps em uma empresa (FERNANDES, 2019):

- Resistência de profissionais a mudanças: como em qualquer mudança, os profissionais precisam estar preparados para que evidenciem de forma clara os benefícios para eles e para a empresa;
- Falta de investimento da empresa: o principal problema é o investimento da alta direção, de forma que os profissionais que implementam o projeto tenham suporte e treinamentos necessários;
- Falta de profissionais com conhecimento necessário;
- Equipes segregadas, trabalhando separadamente com projetos e prioridades distintas e conflitantes;
- Processos de trabalho não automatizados, incluindo a infraestrutura;
- Falta de feedbacks, comunicação e colaboração entre as equipes (Negócios, Desenvolvimento, Qualidade e Operações).

1.1.1 Benefícios

- Os custos de DevOps não são altos pois a maioria das ferramentas é open source;
- Com a integração e colaboração entre as equipes iremos promover a difusão do conhecimento, passagem de experiências, treinamentos e cursos internos, melhorando os problemas de comunicação;
- Padronização dos ambientes: com o trabalho em conjunto entre as áreas é possível que todos trabalhem com as mesmas ferramentas;
- Realizar gestão eficiente, simplificando os processos e utilizando ferramentas do mundo Agile;
- Medir e monitorar os processos, para identificar o desempenho atual e o que pode ser melhorado;
- Realização de entregas contínuas com maior automação e qualidade dos produtos entregues;
- Capacidade de publicar novas funcionalidades, corrigir problemas com agilidade, segurança no rollback das versões do sistema, entre outras, agregam uma importante vantagem competitiva ao negócio;
- Redução de erros humanos por conta da automação de processos;
- Com a automação é possível descobrir o erro mais cedo e corrigir rápido.

1.1.2 Riscos de uma Implantação inadequada

- Não compreender o conceito DevOps: o conceito se estende muito além de uma metodologia de desenvolvimento, é como uma cultura operacional que cria uma linha única de colaboração entre desenvolvedores e operações. É necessária uma visão geral estratégica e a construção de um ambiente colaborativo, desde o planejamento até o usuário final. Esse entendimento é importante porque a adoção do DevOps deve ser plena, envolvendo várias equipes diferentes (principalmente de desenvolvedores e de infraestrutura);
- Resistência da adoção de DevOps: a resistência pode ocorrer entre os envolvidos, pois alguns não querem perder o controle total daquilo que

fazem. Evitam passar o conhecimento acumulado durante os anos na empresa, com medo de perder a influência e poder;

- Choque cultural dos envolvidos: o choque cultural pode ocorrer na integração das equipes quando parte dos processos enraizados foram modificados.

1.2 Hipótese

A disseminação da cultura DevOps na empresa promoverá união das áreas de tecnologia e irá melhorar a performance das equipes, com times multidisciplinares e autossuficientes. Cada membro deve entender o valor do produto para a organização e ser capaz de passar por todo o processo de criação e desenvolvimento trabalhando em harmonia. Com a integração das áreas, o tempo de execução das tarefas e os custos de cada fase serão reduzidos, possibilitando um feedback contínuo e melhorando a comunicação. A automatização dos processos diminuirá o retrabalho, aumentará o controle das aplicações desenvolvidas e permitirá que o time possa focar em melhorias e inovação. Desta forma, as entregas serão mais rápidas e com menos riscos. O time mais motivado, com uma maior compreensão dos detalhes do projeto, gera produtos com maior qualidade e aumenta a eficiência.

Com a capacitação dos profissionais nas melhores práticas do mercado, pretende-se motivar as equipes, aumentar a capacidade de inovar e melhorar os resultados para o cliente.

1.3 Justificativa

Segundo Muniz (2019, p.3), 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 filosofia DevOps tem como grande motivador eliminar os silos entre áreas de tecnologia (desenvolvimento, operações, qualidade e negócio) e fazer com que todos os envolvidos se sintam donos do produto final, trabalhando de forma colaborativa.

A necessidade de melhorar a qualidade das entregas dos nossos produtos, bem como a importância de diminuir os conflitos de objetivos e interesses das equipes, consolidaram a escolha do tema. Posteriormente o grupo submeteu para validação e

aprovação do projeto final do curso de Programa de Desenvolvimento de Liderança (PDL) da IplanRio, a disseminação da cultura DevOps na empresa. Por estar totalmente alinhado ao Planejamento Estratégico da IplanRio (iniciativa estratégica “Desenhar e Implementar DevOps”), a Diretoria aprovou o projeto redefinindo sua abrangência.

1.4 Escopo

O presente escopo trata de mapear o nível de maturidade da empresa na cultura DevOps e propor um plano de ação para a execução de um projeto piloto que visa acompanhar a evolução da maturidade DevOps na IplanRio, focando a melhoria contínua do processo. O objetivo deste piloto é servir de base para a expansão da cultura para outras equipes da organização.

Como premissa, o projeto piloto deverá ser executado com a equipe de desenvolvimento Java, lotada na DSI/GDS, pois trata-se de um time que já possui conhecimento e expertise nas metodologias ágeis e que evoluiu na construção de um pipeline automatizado de implantação das aplicações nos ambientes.

As questões não contempladas neste escopo estão detalhadas no item 4 Considerações Finais, ao final do documento.

1.5 Objetivos

1.5.1 Objetivos Gerais

O objetivo deste trabalho é construir um plano de ação, usando como base o framework ‘CALMS’, para adoção de práticas DevOps, melhorando a integração e comunicação entre os times. Para isso será executado um projeto piloto permitindo que a equipe vivencie os novos conceitos e práticas, com monitoramento de desvios e implementação das mudanças necessárias.

O framework CALMS foi escolhido como base para o desenvolvimento deste trabalho, porque ele traz as principais diretrizes para a adesão do DevOps, transformando as estruturas internas para esse fim. Maiores detalhes podem ser consultados no APÊNDICE A - Framework Calms.

1.5.2 Objetivos Específicos

- Integrar as equipes das áreas técnicas;
- Mapear o nível de maturidade da empresa, entendendo as dores e necessidades;
- Revisar o processo atual de desenvolvimento, para entender os gargalos e entraves;
- Revisar o pipeline utilizado e propor melhorias;
- Definir um plano de ação para disseminar a cultura e as práticas DevOps entre todos os times.

2. METODOLOGIA

A metodologia aplicada neste projeto envolveu técnicas de pesquisa exploratória com pesquisas literárias para embasamento teórico e nivelamento do grupo; entrevistas com profissionais da área de atuação da empresa e cases de adoção em outras organizações.

Para uma visão mais abrangente da situação atual e identificação das barreiras e oportunidades de melhoria do processo, combinamos duas abordagens: a primeira foi uma abordagem quantitativa, através de um *assessment* usado para classificarmos e analisarmos a situação atual e onde queremos chegar com a disseminação da cultura DevOps. A segunda foi uma abordagem qualitativa, que ocorreu através de entrevistas com diversos perfis de profissionais da empresa, onde entendemos as dores e necessidades dos times.

A seguir contaremos a nossa jornada para a transformação que estamos propondo com a disseminação da cultura DevOps.

2.1 Referências Literárias

No início do trabalho, percebemos que o grupo possuía muitas dúvidas e tínhamos conhecimentos diversificados sobre o tema. A pergunta principal que todos faziam era: o que realmente é DevOps? É um framework, uma ferramenta ou um processo de trabalho? Por isso decidimos nivelar esse conhecimento através de pesquisa literária, que envolveu pesquisas nos seguintes sites: Microsoft Azure¹, Vertigo Tecnologia², Redhat³, Gaea Consulting⁴, DevOps Manifesto⁵, Estabil⁶, Tecnospeed⁷, Digital Stream and IT Innovation⁸, InfoQ⁹ e Código Fonte TV¹⁰.

A internet dispõe de um imenso material e fomos bombardeados com tanta informação que em determinado momento nos sentimos perdidos. Decidimos então

¹ <https://azure.microsoft.com/pt-br/overview/what-is-DevOps/>

² <https://vertigo.com.br/o-que-e-DevOps/>

³ <https://www.redhat.com/pt-br/topics/DevOps>

⁴ <https://gaea.com.br/o-que-e-DevOps-conceito/>

⁵ <https://sites.google.com/a/jezhumble.net/DevOps-manifesto/>

⁶ <https://blog.estabil.is/calms-conceito-DevOps/>

⁷ <https://blog.tecnospeed.com.br/DevOps/>

⁸ <https://leonardo-matsumota.com/2019/02/19/adotando-o-DevOps-e-calms/>

⁹ <https://www.infoq.com/br/news/2015/06/DevOps-definition/>

¹⁰ <https://www.youtube.com/watch?v=iwf6kcvxeD4>

usar como base o livro Jornada DevOps [MUNIZ, 2019]. Esse livro apresenta uma linguagem mais didática e organizada, tendo sido escrito de forma colaborativa por diversos autores especialistas no assunto. Essa leitura nos ajudou a dar um norte nos conceitos e práticas, criando uma visão mais clara do caminho que deveríamos seguir.

Entramos em contato com um dos autores do livro e decidimos iniciar o Benchmarking para entender como as empresas do mesmo setor estavam praticando a disseminação da cultura DevOps, seus desafios e os benefícios que obtiveram.

2.2 Benchmarking

2.2.1 Promove Soluções

O primeiro Benchmarking foi realizado com a sócia-fundadora da Promove Soluções, Analia Irigoyen, em 04/06/2020. Analia é uma das co-autoras do livro Jornada DevOps [MUNIZ, 2019] e atua na área de liderança de projetos há mais de 18 anos e possui uma grande experiência na implantação de práticas e disseminação da cultura DevOps em algumas empresas, como TopDown Sistemas, Comitê Olímpico do Brasil, Grupo Mult, Sonda, entre outros, conforme Figura 1.

Figura 1 - Benchmarking com a Consulta da Promove Soluções Analia Irigoyen



Fonte: elaborado pelos autores

Nesta entrevista, Analia nos apresentou os conceitos DevOps e as principais práticas adotadas pelo mercado. Foram apresentados alguns cases e lições aprendidas durante as consultorias realizadas, esclarecendo que a obtenção dos

resultados é lenta e a longo prazo, por conta da dificuldade da mudança cultural. Outro ponto importante, é que cada empresa tem suas peculiaridades e precisa de soluções distintas.

Este encontro foi fundamental para o direcionamento do trabalho, porque foi através dele que percebemos que não podíamos iniciar nossa jornada sem envolver as equipes. A colaboração é um dos principais pilares do DevOps e não conseguiríamos alcançar o sucesso sem a participação das pessoas. Não existe um roteiro pronto para a disseminação da cultura DevOps e precisamos adequá-lo à nossa realidade.

Pensando nisso, o grupo definiu uma estratégia para trazer o engajamento das pessoas e torná-las parte do processo de adoção desta cultura. Decidimos, então, reunir diversos profissionais da empresa numa manhã de segunda-feira e assim surgiu o Café com DevOps, que será detalhado no item 2.3 Café com DevOps.

Em paralelo aos encontros com os profissionais das empresas, demos continuidade ao nosso estudo para entender as melhores práticas adotadas por outras empresas na adoção da Cultura DevOps e entrevistamos a equipe do Tribunal de Justiça do Estado do Rio de Janeiro e a RedHat.

2.2.2 Tribunal de Justiça do Estado do Rio de Janeiro

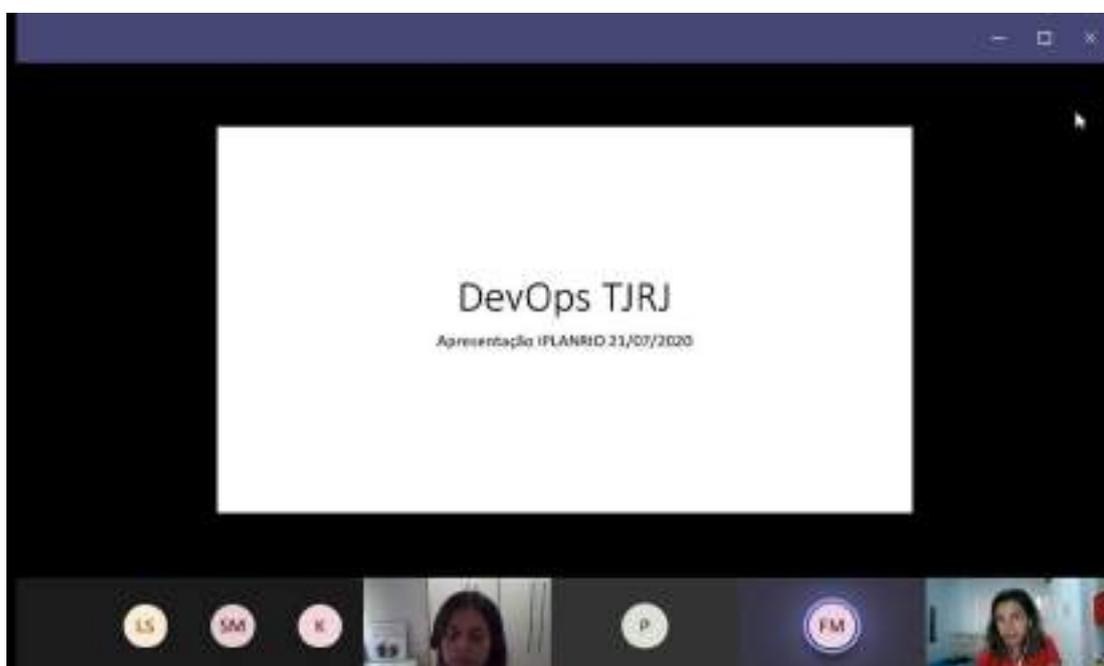
A Iplanrio, sendo uma empresa pública, possui características e limitações que a diferencia das empresas privadas e que muitas vezes podem ser um obstáculo em um processo de inovação. Por conta disso, decidimos procurar por empresas ou órgãos públicos e por indicação, convidamos o Tribunal de Justiça do Estado do Rio de Janeiro para uma entrevista.

Em 21/07/2020 entrevistamos o Diretor de Tecnologia, Flávio José Vazquez Maceira e seu Assessor Paulo Henrique Alexandre da Silva para entender os desafios e as estratégias adotadas na implantação da cultura DevOps no Tribunal de Justiça do Estado do Rio de Janeiro, conforme Figura 2. Flávio José compartilhou conosco a jornada do Tribunal de Justiça para disseminar a cultura DevOps e para que pudéssemos entender os desafios, ele inicialmente explicou a estrutura da área de tecnologia, que não possui desenvolvedores próprios.

Quando o Tribunal de Justiça decidiu adotar a cultura internamente, priorizou a contratação de desenvolvedores que já tivessem algum nível de experiência nas práticas DevOps. Algumas estratégias foram adotadas no início, como a reestruturação das Diretorias, com objetivo de melhorar a integração entre as equipes, bem como a execução de um projeto piloto de pequeno porte. Esse projeto está sendo importante para que eles possam experimentar e validar as práticas adotadas e corrigir pequenos desvios, sendo assim não tivemos como obter mais informações, pois o piloto ainda está em andamento. Para apoiar o piloto, o TJ adquiriu o pacote da Azure DevOps Server, que possui as ferramentas Azure Boards, Azure Pipelines e Azure Repos.

A troca de experiências nos permitiu ter a visão dos desafios que eles estão enfrentando e verificamos que conseguimos avançar em algumas práticas de integração contínua, como por exemplo, o pipeline de build automatizado integrado ao GIT. Isto significa que quando um desenvolvedor faz um *push*, no repositório, ele constrói e disponibiliza a aplicação no ambiente, sem precisar de intervenção humana.

Figura 2 - Tribunal de Justiça do Estado do Rio de Janeiro



Fonte: elaborado pelos autores

2.2.3 RedHat

Convidamos a RedHat, por ser o fornecedor da plataforma de containers Openshift adotada pela Empresa. A entrevista foi realizada no dia 15/10/2020 com o Arquiteto de Soluções, Luiz Bernardo Levenhagen Alarcon da Fonseca para que ele compartilhasse conosco a sua experiência na adoção da cultura DevOps e apresentasse para o grupo, a plataforma de containers Openshift¹¹.

Luiz, durante a adoção da cultura DevOps em uma startup em que trabalhou, ajudou na criação de uma equipe formada por profissionais multidisciplinares que eram responsáveis por gerenciar as ferramentas que davam apoio ao desenvolvimento. Cada membro desta equipe fazia parte de um time de desenvolvimento e era o responsável por disseminar a cultura dentro daquele time. Quando o time de desenvolvimento tinha algum problema, este integrante acionava a equipe DevOps, que discutia internamente a melhor solução para depois aplicar no seu time, conforme Figura 3.

Figura 3 - Equipe Devops



Fonte: apresentação RedHat

Os problemas eram solucionados, porém gerava um gargalo na equipe DevOps e os times de desenvolvimento não participavam da solução. A cultura DevOps preza

¹¹ <https://www.redhat.com/pt-br/technologies/cloud-computing/openshift?source=searchresultlisting>

pela colaboração, autonomia e a quebra das barreiras e eles perceberam que era necessária uma mudança na abordagem pois os silos continuavam existindo.

Como forma de melhorar o processo, a equipe DevOps passou a ser conhecida como “Accelerate”. Nesta nova forma de trabalho, todo o time (Desenvolvimento e Accelerate) era envolvido na solução de um problema, não apenas o representante DevOps do time. Desta maneira, eles garantiam que todos teriam um aprendizado comum, melhorando a comunicação e a responsabilidade era compartilhada por todos.

Seguindo este novo modelo, alguns resultados foram percebidos, como o aumento da melhoria da comunicação e da performance da aplicação, redução do índice de erros e de deployments falhos por conta da automação dos testes.

Ao final da apresentação ele fez um overview da ferramenta Openshift, mostrando as suas principais funcionalidades e um exemplo prático de aplicação.

2.3 Café com DevOps

Em paralelo a todas as etapas do Benchmarking, iniciamos o **Café com DevOps**, evento que reuniu profissionais de diversas áreas da empresa, selecionados por terem tido alguma experiência em projetos com o time de desenvolvimento Java da Diretoria de Sistemas.

O objetivo inicial do Café com DevOps era apresentar o projeto e envolver as pessoas na adoção e disseminação da cultura DevOps desde o início. Para isso, decidimos trabalhar de forma colaborativa para que as pessoas envolvidas se sentissem parte do todo e os nossos resultados fossem alinhados com as expectativas da empresa e dos principais impactados pelas mudanças que estariam por vir.

Tivemos a participação das áreas de Qualidade, Operação e Suporte, Desenvolvimento, Arquitetura, Administração de Dados, além de Líderes de Projeto, Analistas de Negócio e UX Designers.

Foram realizados 3 (três) encontros, utilizando a ferramenta Google Meets e cada um teve a duração de 2 horas. O grupo ficou responsável por moderar as dinâmicas, deixando claro para todos que aquele era um ambiente seguro e de respeito em que todos poderiam falar à vontade, sem julgamentos.

Todas as sugestões, dores, necessidades e opiniões, foram registrados conforme apresentado na Figura 4. Para organizar as informações coletadas utilizamos sticky notes (post-its) através da ferramenta de colaboração Miro¹², um quadro branco infinito on-line, que se transformou na nossa Obeya (grande sala) onde todas as informações do nosso trabalho foram registradas. No APÊNDICE E - Mapeamento das Dores e Necessidades, estão listadas as URLs do quadro e os “*stick notes*” (*post-its*) coletados durante as reuniões.

Figura 4 - Exemplo de feedback coletado nas reuniões.



Fonte: elaborado pelos autores

A adesão ao evento demonstrou o nível de engajamento, mobilização e comprometimento dos profissionais da empresa, que contribuíram muito para o entendimento do projeto.

2.3.1 Café com DevOps 1 - Apresentação

O primeiro encontro foi realizado no dia 02/06/2020, onde o grupo apresentou o objetivo do projeto e a importância da colaboração de todos para o sucesso da adoção da cultura DevOps. Pedimos que as pessoas compartilhassem o que entendiam por DevOps e ouvimos muitas sugestões de como poderíamos disseminar a cultura na empresa.

Estávamos bastante apreensivos em relação ao que ouviríamos e foi incrível perceber o quanto eles estavam à vontade para contribuir e dar sugestões. Neste encontro, pudemos perceber que as pessoas estavam entusiasmadas e desejavam participar desta mudança.

¹² <https://miro.com>

Um dos pontos que nos chamou a atenção durante o encontro foi que os envolvidos não tinham uma visão completa do processo de desenvolvimento utilizado pela equipe Java. Decidimos então organizar um segundo encontro, onde teríamos a oportunidade de nivelar o conhecimento sobre o assunto e identificar pontos de melhoria.

Apesar de não termos definido um modelo de feedback formal sobre o evento, recebemos diversos comentários positivos em relação à postura e a iniciativa do grupo na adoção deste modelo colaborativo de trabalho.

2.3.2 Café com DevOps 2 - Revisão do Processo de Desenvolvimento

Como no primeiro evento percebemos que muitos participantes não tinham uma visão completa do processo de desenvolvimento adotado pela equipe Java, realizamos no dia 29/06/2020 o segundo Café com DevOps, cujo objetivo foi apresentar uma visão geral do processo de desenvolvimento, enumerando as atividades realizadas em cada fase, desde o desenho até a publicação em produção, conforme Figura 5.

Figura 5 - Visão Geral do Processo de Desenvolvimento Java



Fonte: elaborado pelos autores

Neste encontro pudemos perceber que, além da falta de conhecimento sobre algumas fases deste processo, nem todos conheciam os papéis desempenhados por alguns perfis, como por exemplo, o papel do UX Designer e as funções de algumas gerências.

Os participantes enfatizaram a importância do envolvimento das diversas áreas da empresa (Operações, Qualidade, Desenvolvimento e Negócios) desde o início do projeto, porque isto poderia reduzir os problemas enfrentados no dia a dia.

Outro ponto levantado foi a diversidade de ferramentas que são utilizadas para gerenciamento, como SGRI¹³, Jira¹⁴, Project Builder¹⁵, DopWiki¹⁶, etc. Essas ferramentas não possuem integração e muitas vezes não são consistentes entre si, dificultando a visão geral do projeto.

Em destaque, identificamos que os Arquitetos Java são familiarizados com a ferramenta Openshift avançando na automação de alguns processos. Construíram um pipeline de build automatizado para publicação das suas aplicações, porém este pipeline não cobre os testes automatizados.

O detalhamento do processo de desenvolvimento da equipe Java pode ser consultado no APÊNDICE C - Revisão do Processo de Desenvolvimento Atual - Java, contendo os detalhes de cada fase descritos. As sugestões, dores, necessidades e opiniões, foram registrados em sticky notes (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

Além disso, iniciamos *Talks* com grupos de perfis distintos, tais como Arquitetura, Qualidade, Desenvolvimento, Operações, UX e Negócios para aprofundar a visão de onde queremos chegar em cada uma dessas disciplinas, através da adoção de DevOps.

¹³ SGRI: Sistema de Recursos de Informática é o cadastro de sistemas da PCRJ, independente se são construídos ou mantidos pela IplanRio, e suas relações com os Recursos Tecnológicos (aspectos de segurança, linguagens, SGBSs, modelos de dados e integrações) e humanos (Gestores e Equipes Técnicas).

¹⁴ Jira: ferramenta que permite o monitoramento de tarefas e acompanhamento de projetos garantindo o gerenciamento de todas as suas atividades.

¹⁵ Project Builder, ferramenta voltada para o gerenciamento de projetos, programas e portfólios de uma organização.

¹⁶ DopWiki: Ferramenta utilizada pela Área de Operações para documentar processos.

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

O 3º Encontro do Café com DevOps foi reservado para a realização do Assessment que tinha por objetivo identificar o nível de maturidade nas práticas DevOps de acordo com o ponto de vista dos profissionais que participaram dos encontros do Café com DevOps.

Para organizar a avaliação, o grupo realizou diversas pesquisas literárias com o objetivo de entender a estrutura de um assessment de maturidade DevOps e optou por utilizar o modelo proposto por Carlos de Amorim Levita, em Proposta de Modelo para Avaliação da Maturidade DevOps¹⁷. Este modelo foi escolhido por cobrir as várias práticas utilizadas na abordagem DevOps de desenvolvimento de software.

As questões foram agrupadas com base em oito práticas DevOps: pessoas e cultura; planejamento; integração contínua; testes; entrega contínua; infraestrutura como código; monitoração e feedback.

Como o nosso objetivo era realizar o assessment de forma colaborativa e interativa, decidimos aplicar, com algumas adaptações¹⁸, a técnica de facilitação conhecida como DotVoting¹⁹. As perguntas estariam disponíveis no quadro digital Miro e os participantes iriam interagir selecionando os ícones referentes a “Onde Estamos” e “Onde Queremos Chegar”, conforme Figura 6.

Figura 6 - Exemplo de Questão do Assessment do MIRO



Fonte: elaborado pelos autores

¹⁷ Proposta de Modelo para Avaliação da Maturidade DevOps. Carlos de Amorim Levita. Pontifícia Universidade Católica de São Paulo, Brasil. Recebido: 14/10/2017 – Aceito: 06/11/2017

¹⁸ Decidimos utilizar ícones mais amigáveis.

¹⁹ <http://www.funretrospectives.com/dot-voting/>

Realizamos uma dinâmica para testar o processo com algumas pessoas externas, porém não se mostrou muito prática, sendo assim optamos por uma outra abordagem. Decidimos adotar o *SurveyAnyPlace*²⁰, uma ferramenta online de criação de formulários, para realizar o *assessment* durante o encontro. Esta ferramenta foi escolhida por oferecer a criação de formulários visualmente agradáveis e customizados, permitindo mais de uma resposta para a mesma pergunta, com regras de validação e um controle sobre o preenchimento das questões, diferente de outras ferramentas de mercado.

Queríamos que a experiência da avaliação fosse visual e de fácil preenchimento e, por isso, nas questões, optamos por uma abordagem baseada no modelo da Agility Health²¹ utilizando a escala de maturidade Sit, Crawl, Walk, Run e Fly. Esta escala nos permitiu fornecer mais clareza em nossas respostas tornando mais fácil para as equipes identificarem onde estão e onde querem chegar, a partir de uma perspectiva de maturidade e fluência em cada prática DevOps. Por ser uma escala gradual, as pessoas poderiam perceber a evolução necessária para cada grau de maturidade, conforme Figura 6.

Sabemos que chegar ao nível mais alto de maturidade não é simples, muito menos rápido e nem sempre necessário, pois isto pode variar de acordo com a necessidade da organização. Porém, como não tínhamos uma noção da nossa maturidade atual, decidimos por realizar esta avaliação.

Para isso utilizamos um formulário com 40 questões onde os participantes foram orientados a marcar duas respostas em cada questão, uma referente a “onde nos encontramos” e a outra indicando “onde queremos chegar”, conforme Figura 7.

²⁰ <https://surveyanyplace.com/>

²¹ <https://agilityhealthradar.com/why-does-agilityhealth-use-pre-crawl-crawl-walk-run-fly/>

Figura 7- Exemplo de questão preenchida no Formulário do Assessment



Fonte: elaborado pelos autores

Os resultados do assessment foram tabulados em uma planilha Excel onde realizamos uma análise por questão e por prática DevOps. As análises por esses dois domínios nos dariam uma visão do grau de maturidade que os participantes percebem da empresa.

Para análise das questões, usamos como critério as respostas mais votadas, pois tivemos participantes com diversos níveis de conhecimento e alguns não se sentiram confortáveis em opinar sobre os assuntos que não dominavam e estas respostas foram desconsideradas. Como critério de desempate, fizemos análise pontual da questão, entrevistamos os especialistas naquela prática, e definimos em conjunto a resposta que fazia mais sentido, para o contexto.

A análise do grau de maturidade por Prática DevOps, considerou a média de todas as respostas e para gerar o Radar de Maturidade, usamos a ferramenta *Visual Paradigm Online*²².

²² <https://online.visual-paradigm.com/pt/>

Os resultados desta análise, assim como as ações derivadas dele, serão apresentados no Capítulo 4 Resultados

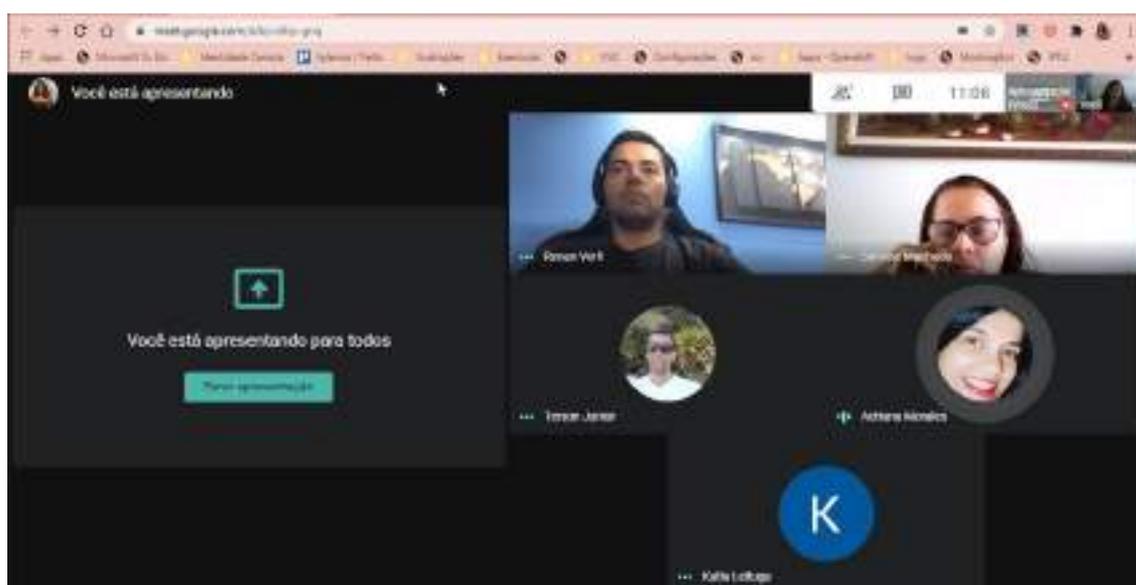
2.4 Talk com especialistas

Em paralelo a realização e análise do assessment, entrevistamos grupos de perfis distintos, tais como Arquitetura, Qualidade, Desenvolvimento, Operações, Negócios e UX para aprofundar a visão de onde queremos chegar em cada uma dessas disciplinas, através da adoção de DevOps. E assim surgiu o Talk com Especialistas.

2.4.1 Arquitetura de Soluções Java

Iniciamos nossas entrevistas com a equipe de Arquitetura de Soluções Java, formada por Terson Rigaud e Renan Verli. Esta equipe, atualmente, é responsável por definir os padrões e ferramentas utilizadas pela equipe de desenvolvimento Java. O encontro foi realizado no dia 07/08/2020 com objetivo conhecer o funcionamento do Pipeline de Implantação utilizado pela equipe Java, as ferramentas utilizadas, entender as dores e necessidades, e os pontos de melhoria que podem ser aplicados, conforme Figura 8.

Figura 8 - Arquitetura de Soluções Java

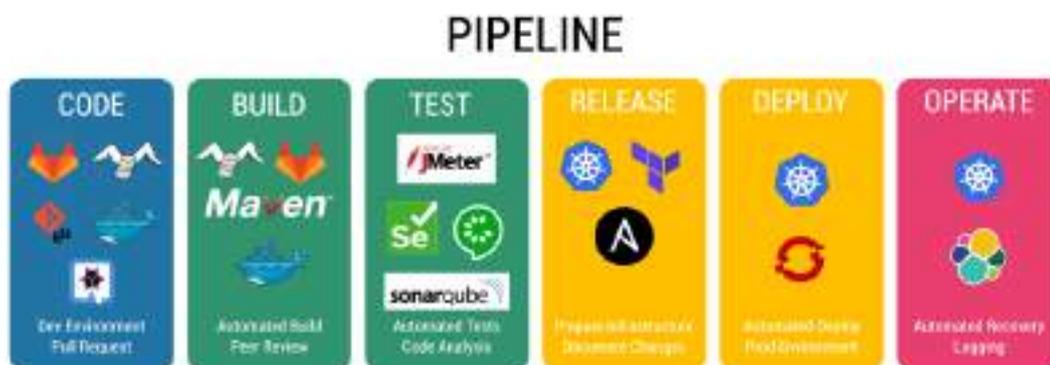


Fonte: elaborado pelos autores

Neste encontro ficou claro o avanço da equipe em relação a automação dos processos e o uso de algumas práticas de Integração Contínua e Entrega Contínua. Atualmente o pipeline só executa o processo de construção da aplicação.

Os arquitetos apresentaram todo o fluxo do pipeline e as ferramentas utilizadas em cada fase, conforme Figura 9.

Figura 9 - Ferramentas utilizadas no Pipeline de Implantação



Fonte: adaptado de Hacker Noon²³

Ficou evidente a importância dos testes automatizados, que ajudarão a reduzir as possibilidades de erros e a quantidade de esforço gasta em testes manuais. Para o projeto piloto, sugeriram iniciar pela automação dos testes unitários, pois estes dependem exclusivamente do desenvolvedor.

As sugestões, dores, necessidades e opiniões, foram registrados em sticky notes (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

2.4.2 Qualidade de Software

Este encontro aconteceu no dia 17/08/2020 com a participação de Cesar Teixeira, Gladys Magali e Márcia Fontes. O objetivo foi conhecer o processo de trabalho, entender os diferentes tipos de testes que podem ser automatizados e as

²³ Disponível em <<https://hackernoon.com/delivery-pipelines-as-enabler-for-a-devops-culture-ebc45963f703>>. Acesso em 03 nov. 2020

ferramentas utilizadas. Também aproveitamos para ouvir sugestões, entender as dores e as suas necessidades, conforme Figura 10.

Figura 10 - Qualidade de Software



Fonte: elaborado pelos autores

Atualmente a equipe de Qualidade realiza diversos tipos de testes. Os testes Funcionais e de Integração são executados de forma manual, já os testes de Vulnerabilidade, Performance/Carga e Estresse são executados com um certo nível de automação, porém não integrados ao Pipeline.

A equipe de Qualidade fez um overview sobre o processo de trabalho interno e questionou qual seria a estratégia adotada para aplicarmos as práticas de automação de testes. Explicamos que o projeto irá resultar na execução de um piloto na equipe Java e, diante disto, a equipe de Qualidade sugeriu que fosse estudada a possibilidade de que o piloto contemplasse a automação de dois tipos de testes: análise estática de código e vulnerabilidade.

Como a empresa utiliza a ferramenta SonarQube²⁴, para análise estática de código, nos contratos em execução com as fábricas de Software, a equipe de Qualidade sugeriu que utilizássemos internamente a mesma ferramenta. A sugestão inicial era que a análise não impedisse a execução do pipeline, mas sim prover

²⁴ <https://www.sonarqube.org/>

insumos para que os desenvolvedores pudessem avaliar o seu código e, se necessário, implementar ações para melhorar a qualidade da equipe.

Para automação dos testes de vulnerabilidade, a equipe de Qualidade sugeriu o uso da ferramenta Acunetix²⁵, que possui alta cobertura de detecção de vulnerabilidades, executa testes de penetração automatizados permitindo uma análise eficiente da segurança da aplicação.

Durante o encontro também conversamos sobre os benefícios esperados com a adoção dos testes automatizados, que requerem muito menos tempo para serem executados, são reutilizáveis e menos suscetíveis a erros em comparação com a intervenção humana.

As sugestões, dores, necessidades e opiniões, foram registrados em sticky notes (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

2.4.3 Workshop de Testes Automatizados

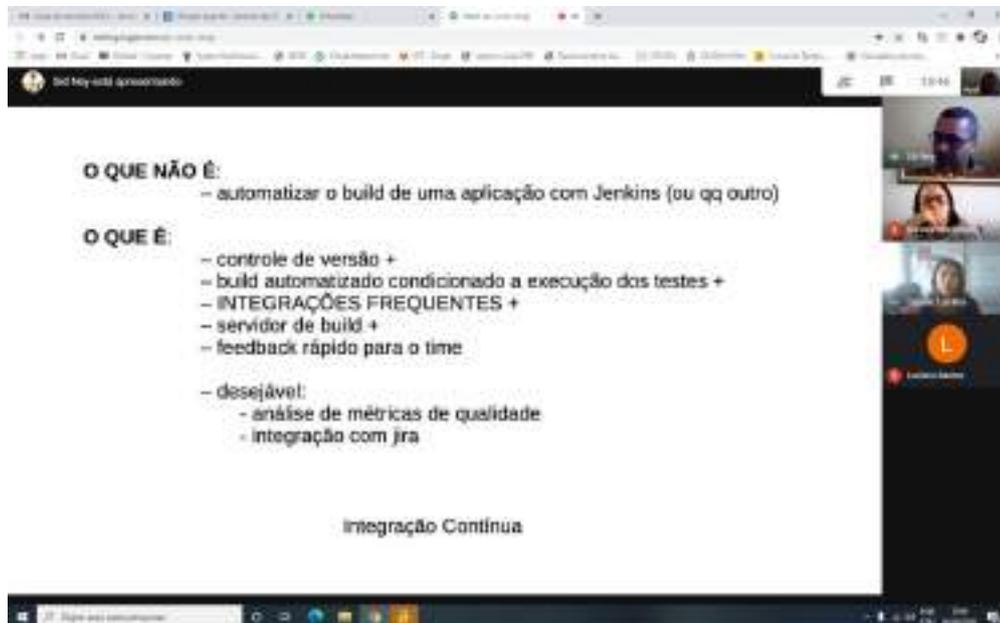
A entrevista com a equipe de Qualidade rendeu *insights* positivos e ótimas sugestões para considerar no projeto piloto. A equipe de Arquitetura de Soluções Java considerou importante evoluirmos o pipeline com a automação dos testes unitários, enquanto a equipe de Qualidade de Software, via valor na automação da análise estática do código e nos testes de vulnerabilidade, pois estes testes são realizados atualmente com as Fábricas de Software contratadas.

Para compreender melhor as diferenças de abordagens sugeridas e termos mais embasamento na definição do caminho a seguir, convidamos Sidney Amaral, da equipe de desenvolvimento Java e um entusiasta das práticas de automação de testes, conforme Figura 11.

Foi realizado um *workshop* com o grupo no dia 26/08/2020, onde Sidney apresentou o tema com uma explicação inicial dos seguintes tipos de teste: teste de usabilidade, teste de carga, teste de vulnerabilidade, teste de funcionamento, teste de regressão, testes exploratórios, testes unitários, teste de integração e teste *end-to-end*. Foram explicadas as diferenças, principais ferramentas e como estes podem ser integrados ao pipeline.

²⁵ <https://www.acunetix.com/>

Figura 11 - Workshop de Testes Automatizados



Fonte: elaborado pelos autores

Sidney reforçou a importância de disseminar a cultura de testes dentro da equipe Java e para toda empresa. Independentemente da existência ou não de um pipeline automatizado, o desenvolvedor poderia executar os testes unitários no seu próprio ambiente local, usando a sua IDE²⁶. Falta uma visão clara dos benefícios que o desenvolvimento orientado a testes pode trazer e existe a crença de que testes aumentam o tempo de desenvolvimento. A realização de *workshops*, mentorias ou outras ações poderiam auxiliar a disseminar a importância da cultura de testes.

Além dos testes unitários, um outro tipo de teste abordado no *workshop* que traz muitos benefícios, se for feito de forma automatizada, é o Teste Funcional. A automação destes testes evita o trabalho manual repetido, economiza tempo de execução e assegura que estamos sempre executando testes de forma consistente com as mesmas pré-condições e expectativas. Outro ponto positivo é a rapidez do *feedback*. Executar testes manuais repetidas vezes pode ser cansativo e suscetível a

²⁶https://pt.wikipedia.org/wiki/Ambiente_de_desenvolvimento_integrado#:~:text=IDE%2C%20do%20in gl%C3%AAAs%20Integrated%20Development,objetivo%20de%20agilizar%20este%20processo.

erros. Entretanto, mesmo com todos esses benefícios, ele não anula a necessidade de executar testes exploratórios na aplicação em algum momento.

Por conta da reunião sobre testes funcionais, a qualidade das histórias de usuários, escritas pelos analistas de requisitos e em alguns casos, o UX designer, foi abordada. As histórias precisam ser escritas de forma objetiva e com critérios de aceitação claros, porque muitas vezes o que é óbvio para quem está escrevendo pode não ser para quem está desenvolvendo.

Tivemos também um overview sobre Test Driven Development (TDD) e Behavior Driven Development (BDD).

Sidney reforçou que DevOps requer Integração Contínua e que esta prática requer testes automatizados. A cobertura mínima de testes deve ser discutida entre a equipe para juntos definirem o percentual de testes aceitável.

Ao final foram apresentadas algumas ferramentas que podem ser utilizadas para automação de testes, como: Maven, Gradle, JUnit, Selenium e Apium.

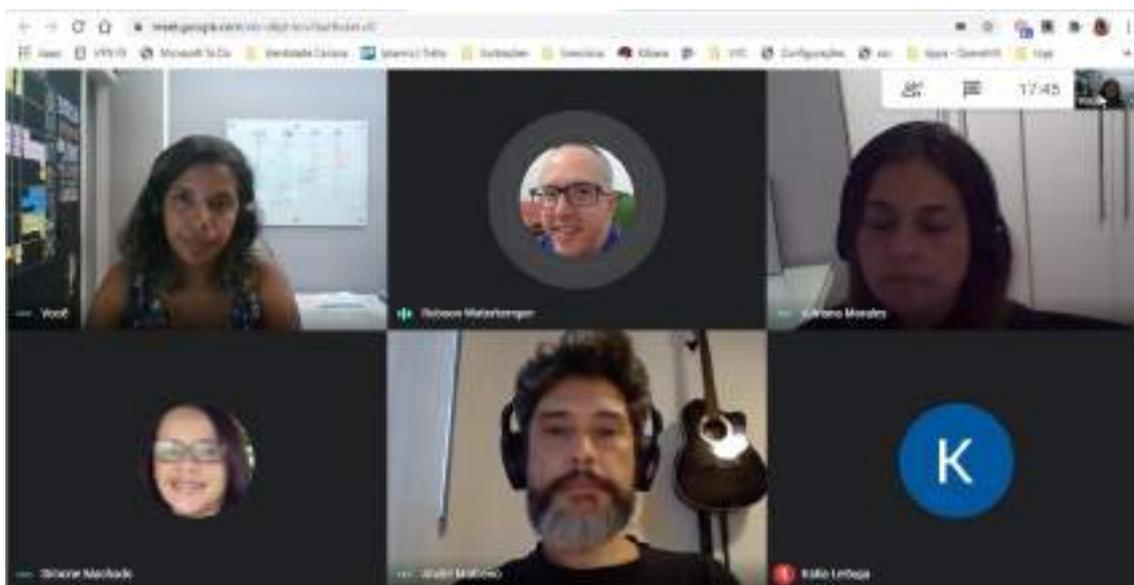
As sugestões, dores, necessidades e opiniões, foram registrados em sticky notes (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

2.4.4 UX Designer

Um dos maiores questionamentos durante o Café com DevOps foi sobre as funções do UX Designer e para conhecer um pouco mais sobre este perfil convidamos André Malheiros e Robson Waterkemper, para um bate-papo no dia 03/09/2020, conforme Figura 12.

O objetivo da reunião foi conhecer as funções e responsabilidades do UX Designer na empresa, entender o valor agregado com foco na experiência do usuário para o cidadão e para a gestão. Ouvimos as dores e necessidades dos UX Designers e pontos de melhoria que poderiam ser aplicados no processo de desenvolvimento.

Figura 12 - UX Designer



Fonte: elaborado pelos autores

André e Robson explicaram que o UX Designer tem responsabilidades que vão muito além de escrever histórias. O UX Designer se preocupa com a melhor experiência dos usuários, com a maior eficiência no uso de um serviço ou produto e com a marca da empresa.

Entender as necessidades e desejos do usuário é essencial para criar ou aprimorar serviços que gerem altos níveis de satisfação. E é esse o foco principal do trabalho deles, trazer satisfação para o usuário durante sua experiência com o produto ou serviço.

Por conta das dinâmicas realizadas no Café com DevOps, eles perceberam que existem muitas dúvidas do papel do UX Designer e por acharem importante que a empresa tenha mais pessoas com este perfil, estão abertos a compartilhar o conhecimento deles com os demais profissionais da empresa e abrir espaço para a troca e para o crescimento profissional.

Além da realização de palestras, eles sugeriram que os analistas de negócio pudessem fazer uma imersão profissional, acompanhando o dia a dia de um projeto e tendo assim um novo olhar nos produtos que estão sendo desenvolvidos pela empresa.

Apesar de não termos conseguido extrair assuntos referentes à cultura DevOps que impactem diretamente no projeto piloto, o encontro rendeu boas sugestões e nos

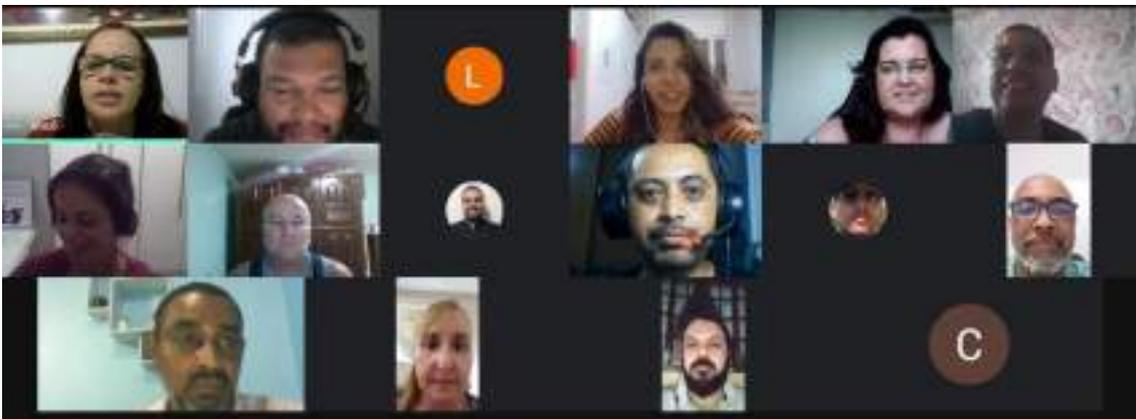
fez refletir como empresa, que precisamos adotar uma cultura de compartilhamento e integração entre as equipes.

As sugestões, dores, necessidades e opiniões, foram registrados em sticky notes (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

2.4.5 Operações

A reunião com a equipe de Operações aconteceu no dia 07/10/2020 com a participação de Carlos Sampaio, Sócrates Quintanilha, Aguinaldo Teixeira, Marcos Paiva, Douglas Brum, Claudia Costa, Leonardo Faccini, Nataly Abreu, Carlos Pedro Garcez, Marcelo Fabricio e Carlos Santos. O objetivo foi ouvir as suas expectativas sobre a adoção Devops, conhecer seu processo de trabalho, entender as dores, necessidades e os pontos de melhoria que podem ser aplicados, conforme Figura 13.

Figura 13 - Operações



Fonte: elaborado pelos autores

A equipe de Operações reforçou como é importante a colaboração para a resolução de problemas e que o grande desafio na disseminação de DevOps será a mudança cultural da empresa. A visão de “dono” precisa ser compartilhada e difundida entre os envolvidos no projeto, para que todos se sintam parte e responsáveis pelo sucesso do produto.

Embora a relação com a equipe Java na publicação dos novos projetos possua um processo bem definido, por conta do pipeline de build automatizado, eles ainda enfrentam problemas com outras equipes. Informações desencontradas e processos confusos acabam tornando a publicação dos projetos destas equipes extremamente custosa e por isso consideram importante uma revisão dos papéis e responsabilidades.

Isto levou a uma preocupação, sobre o que será feito com o Legado. Existe uma diversidade de sistemas desenvolvidos com diferentes tipos de tecnologias, algumas obsoletas, que tornam custoso a sua manutenibilidade. Como a empresa está caminhando para o ambiente de computação em nuvem é preciso que seja estudado o que será feito com estas aplicações legadas.

A equipe de Banco de Dados reforçou a importância de ser envolvida no projeto, pois isso daria a oportunidade de contribuírem para mitigar possíveis problemas e evitar situações em que eles só são acionados quando a aplicação já está disponível em produção, como por exemplo, problemas de performance em queries.

Um ponto destacado durante a entrevista, foi que a área de banco de dados está muito pulverizada dentro da empresa, distribuída entre as 3 (três) Diretorias e foi sugerido um estudo com uma revisão de papéis e responsabilidades entre as áreas de Administração de Dados, Banco de Dados e *Business Intelligence*.

A monitoração da aplicação é um ponto crítico dentro da Empresa e a equipe entende que esta é uma responsabilidade que deve ser compartilhada por todos. Como eles não possuem uma visão completa da aplicação, isto dificulta a identificação dos problemas e estes poderiam ser resolvidos de forma mais rápida se todos colaborassem para melhorar este processo.

A equipe de Operações também demonstrou o desejo de conhecer outras áreas e equipes da empresa, e abrir um espaço para trocar experiências.

As sugestões, dores, necessidades e opiniões, foram registrados em *sticky notes* (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

2.4.6 Desenvolvimento

A reunião com a equipe de Desenvolvimento aconteceu no dia 08/10/2020 com a participação de André Márcio, Terson Rigaud, Eduardo Medeiros, Pedro Cláudio, Renan Verli, Sidney Amaral, Rodrigo Bartole, Eugenio Cunha, Bruno Ferreira e Marcelo Magalhães, conforme Figura 14. O objetivo era ouvir as expectativas deles com a adoção DevOps, conhecer o processo de trabalho, entender as dores e necessidades e ouvir sugestões.

Figura 14 - Desenvolvimento



Fonte: elaborado pelos autores

Durante a reunião a equipe informou que gostaria de entender como será feita a disseminação da cultura DevOps, uma vez que o modelo atual da empresa é segregado em silos e eles entendem que isto pode dificultar a adoção. Além disso, eles entendem que o apoio da alta gestão será fundamental para derrubar barreiras e divisões, favorecer a integração entre as equipes e o amadurecimento da cultura DevOps.

Foram sugeridas algumas estratégias para serem aplicadas no piloto, como por exemplo: a criação de uma equipe DevOps, formada por profissionais que seriam responsáveis por definir os padrões, pipelines e dariam apoio às equipes de desenvolvimento; a formação de uma equipe de profissionais multidisciplinares, onde

cada um tem a sua especialidade, mas conhece um pouco de cada assunto e formar times por produtos.

Outro assunto abordado foi o desejo de retomar os encontros técnicos que aconteciam quinzenalmente. Era um momento importante e único entre os desenvolvedores, pois eles podiam trocar experiências, adquirir novos conhecimentos e tirar dúvidas de ferramentas. Percebe-se que muitos ainda têm dúvidas básicas, como usar o Git e a API REST, entre outros. Muitos não conhecem a estrutura do pipeline atual. Enquanto o time falava sobre o pipeline, foi levantada a necessidade de reavaliar a definição de pronto de uma atividade, no momento da inclusão dos testes automatizados no pipeline.

Um ponto levantado pela equipe foi quando se inicia um projeto, as cerimônias são realizadas de forma regular. Porém, aos poucos, alguns ritos importantes acabam não sendo executados e seria importante manter as cerimônias. Por exemplo, as reuniões de retrospectiva servem para avaliar os resultados da sprint que foi finalizada.

A equipe também abordou a qualidade da escrita das histórias de usuários. Em alguns projetos, eles ainda recebem histórias nem sempre tão claras e objetivas, o que dificulta o entendimento e acaba gerando retrabalho.

Todos questionaram o que será feito com o legado, e sugeriram que seja incluído no projeto um plano de ação para tratar estas aplicações.

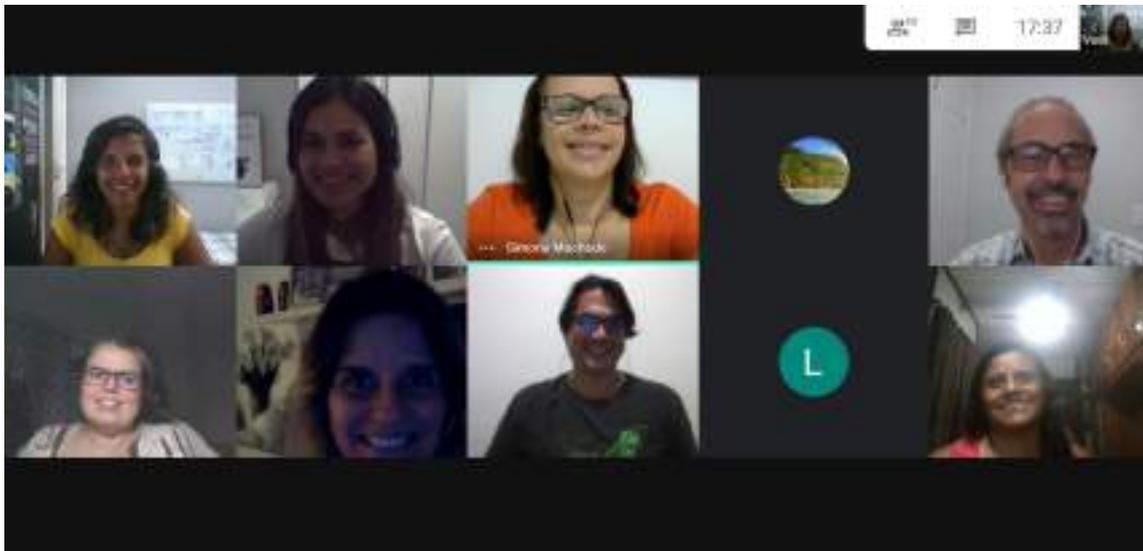
As sugestões, dores, necessidades e opiniões, foram registradas em *sticky notes* (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

2.4.7 Analistas de Negócio

A reunião com a equipe de negócios aconteceu no dia 16/10/2020 e teve a participação de Elaine Silva, Isabel Coelho, Tereza Cristina, Maria Isabel Albuquerque, Ricardo Lima e Marco Antônio da Silva, conforme Figura 15. O objetivo foi avaliar o relacionamento dos Analistas de Negócio com as equipes de Desenvolvimento e Operações e identificar melhorias no processo. Também foi abordada a experiência dos analistas que já participaram de projetos com um

profissional de UX, além de mapear as dores, necessidades e ouvir sugestões sobre a adoção de DevOps na Empresa.

Figura 15 - Analista de Negócio



Fonte: elaborado pelos autores

Como o projeto piloto será realizado com a equipe de desenvolvimento Java, explicamos que o foco da entrevista seria no relacionamento dos analistas com esta equipe, e seria uma oportunidade de melhorarmos nossos processos de trabalho, através do *feedback* deles.

Alguns apontaram uma dificuldade em se adaptar aos processos ágeis, utilizados pela equipe Java, mas que foram aprendendo de acordo com o progresso do projeto.

Os analistas de negócio informaram que o relacionamento deles com os times (Desenvolvimento & Operações), no geral, é feito de forma colaborativa pois eles são solidários às prioridades e às situações de exceção no processo.

Em relação a adoção da cultura DevOps pela Empresa, os analistas mostraram preocupação às diversas tribos tecnológicas que existem atualmente e como isso pode ser um grande desafio, quando a disseminação for estendida para as outras equipes. Algumas GTIS possuem equipes exclusivas com tecnologias obsoletas, processos de desenvolvimento distintos e não estão acompanhando as evoluções das tecnologias mais atuais. Por conta disso, mesmo sabendo que o Legado não faz parte

do escopo do projeto, eles questionaram se iremos propor um estudo ou recomendação para estas aplicações e seus processos não automatizados.

É importante que a cultura DevOps também seja disseminada entre os Gestores, pois é uma quebra de paradigma e é preciso que eles estejam abertos à mudança.

Os analistas de negócio que tiveram experiência com um profissional de UX Designer informaram como foi perceptível a diferença que um profissional com esta capacitação trouxe para o projeto. Como a visão deles é centrada no usuário final, passamos a desenvolver produtos que realmente atendem a necessidade dos usuários. A comunicação simples e clara, facilitou a validação das histórias com os Gestores, reduziu o retrabalho e aumentou a qualidade das entregas. Reforçaram também a necessidade de se capacitar outros profissionais da empresa com esta visão.

Os Analistas que ainda não tiveram essa parceria, informaram que em seus projetos eles precisam fazer papel de Líder de Projeto, Analista de Negócio, Analista de Requisito, Product Owner, o que acaba gerando um acúmulo de responsabilidades e impactando na qualidade do produto. Seria importante rever estes processos, definir os papéis e responsabilidades destes perfis, incluindo capacitação.

Quanto ao gerenciamento do projeto, eles reforçaram que existem muitas ferramentas não integradas e que o ideal seria um *workflow* com uma visão geral para todos, para assim conseguir gerir e classificar melhor a demanda quanto à sua criticidade.

Todos se mostraram receptivos e favoráveis à mudança e estão abertos a colaborar.

As sugestões, dores, necessidades e até mesmo os desabafos, foram registrados em sticky notes (post-its) no Miro e podem ser consultadas no APÊNDICE E - Mapeamento das Dores e Necessidades.

3. RESULTADOS

Neste capítulo serão apresentados e analisados os resultados obtidos a partir dos dados coletados do *Assessment*, mapeamento das Dores e Necessidades obtidas através das dinâmicas realizadas no Café com DevOps e no *Talk* com Especialistas e apresentaremos o Plano de Ação para o Projeto Piloto, bem como as ações para a Empresa.

3.1 Assessment

Foi aplicado um questionário com participação de 24 pessoas das Diretorias Técnicas, com objetivo de identificar o nível de maturidade nas práticas DevOps e entender “onde estamos” e “onde queremos chegar”. Detalhes do Assessment podem ser consultados no item 2.3.4 Café com DevOps - Avaliação do Nível de Maturidade DevOps.

3.1.1 Perfil dos Participantes

Como podemos verificar na Figura 16, 56% dos entrevistados faziam parte da Diretoria de Sistemas, com 24% da Diretoria de Negócios e 20% da Diretoria de Operações.

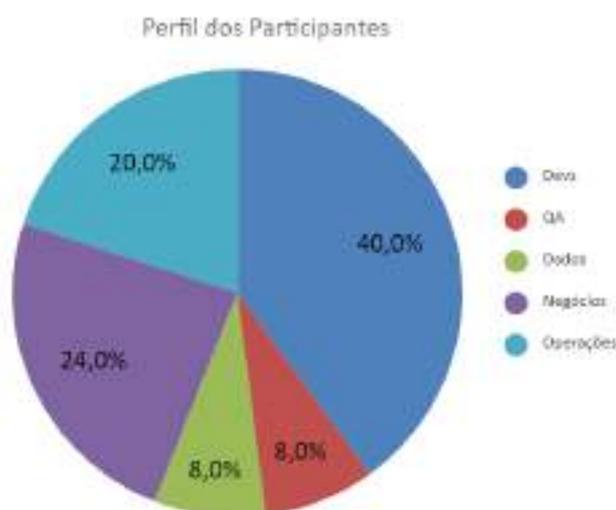
Figura 16 - Participação por Diretoria



Fonte: elaborado pelos autores

No que diz respeito ao perfil dos participantes, conforme Figura 17, podemos verificar que 40% dos entrevistados eram desenvolvedores, 20% da área de Operações e 8% da área de Qualidade, ou seja, 68% dos participantes era composta pelos principais envolvidos com a disseminação da cultura Devops.

Figura 17 - Perfil por Participantes



Fonte: elaborado pelos autores

3.1.2 Resultados por Práticas DevOps

Os resultados por Prática DevOps serão analisados apresentando a visão de “onde estamos”, com as práticas que implementamos atualmente e “onde queremos chegar”, que contempla a adoção de todas as práticas adotadas nas fases anteriores.

O detalhamento destes resultados pode ser verificado no APÊNDICE D - Assessment.

3.1.2.1 Pessoas e Cultura

De acordo com Spafford e Haight (2014), a dimensão mais importante para DevOps está relacionada a pessoas e tem a ver com a mudança de cultura da empresa, que necessita rever o papel da TI, o que muitas vezes acaba levando a uma reavaliação da sua estrutura organizacional e a uma maior valorização de profissionais que conheçam várias áreas, como desenvolvimento, teste e operações.

Por conta da sua importância, Pessoas e Cultura é considerado um dos pilares da nossa pesquisa.

Analisando os resultados aferidos, conseguimos verificar que 54% dos entrevistados possuem pouco conhecimento e experiência nos princípios básicos DevOps, mas 60% dos entrevistados gostariam que as práticas fossem difundidas em toda Empresa, conforme Figura 18.

Figura 18 - Questão 1 - Você conhece os princípios básicos de DevOps e seus benefícios?



Fonte: elaborado pelos autores

Também foi observado que embora 57% dos entrevistados tenham apontado que apenas a área de desenvolvimento utiliza Metodologia Ágil, o uso desta prática não é comum na Empresa e por isso 64% dos entrevistados gostariam que as práticas ágeis fossem difundidas, conforme Figura 19.

Figura 19 - Questão 2 - O processo de desenvolvimento adotado na sua equipe está alinhado com os princípios DevOps ?



Fonte: elaborado pelos autores

Quanto à interação entre os times de Desenvolvimento, Qualidade e Operação, 67% dos entrevistados informaram que todos se conhecem, porém apenas interagem na entrada em produção do projeto ou em problemas pontuais. Para 79% dos entrevistados seria importante que todos trabalhassem desde o início do projeto de forma coesa e colaborativa, conforme Figura 20.

Figura 20 - Questão 3 - Os membros dos times de desenvolvimento, qualidade e operações se conhecem?



Fonte: elaborado pelos autores

Em relação a eventos técnicos, 58% dos entrevistados informaram que raramente a organização promove esses encontros e para 60% seria importante que *talks*, *workshops*, entre outros acontecessem de forma regular, conforme Figura 21.

Figura 21 - Questão 4 - Há eventos regulares na organização que integrem as equipes?



Fonte: elaborado pelos autores

Dos entrevistados, 50% afirmaram que não há colaboração para a construção de conhecimento compartilhado entre os times e 53% gostariam que essa prática fosse adotada de forma regular, conforme Figura 22.

Figura 22 - Questão 5 - Existe colaboração para a construção de conhecimento compartilhado?



Fonte: elaborado pelos autores

De forma resumida, disseminar a Cultura DevOps é fundamental para garantir o sucesso, sendo a colaboração entre as pessoas o fator mais importante. Com base nos resultados percebemos que há entre os participantes o desejo de estar em um ambiente colaborativo, e que esta cultura seja difundida dentro da empresa com encontros técnicos frequentes. As ações que refletem os desejos e necessidades relacionados a este tema serão apresentadas no Plano de Ação ao final do Capítulo.

3.1.2.2 Planejamento

Segundo Levita(2017), o planejamento visa o estabelecimento de metas e o seu constante ajuste, decorrentes da necessidade de responder de forma ágil aos *feedbacks* dos clientes, que são recebidos durante o desenvolvimento do software.

Analisando as respostas sobre o Planejamento, identificamos que 71% dos entrevistados utilizam a ferramenta Jira para organizar os requisitos funcionais do projeto levantados com o cliente, sejam eles como casos de uso ou histórias de usuário e que 79% gostariam que Desenvolvedores, Qualidade e Operações tivessem uma ferramenta integrada, que desse uma visão completa do requisito, conforme Figura 23.

Figura 23 - Questão 6 - Como os Requisitos Funcionais são coletados junto ao cliente?



Fonte: elaborado pelos autores

Em relação ao acompanhamento do projeto, 50% dos entrevistados utilizam a ferramenta Jira para gerenciar o Backlog, trabalhando com sprints de *time-box* bem definidos e 69% dos entrevistados gostariam que este processo de trabalho pudesse abranger outras equipes, conforme Figura 24.

Figura 24 - Questão 7 - Como é realizado o acompanhamento do projeto?



Fonte: elaborado pelos autores

A priorização dos requisitos entre os Gestores e Líderes, ocorre, para 74% dos entrevistados sem o envolvimento do time e 69% gostariam de ter uma maior participação na priorização e para ter um melhor entendimento do negócio, conforme Figura 25.

Figura 25 - Questão 8 - Como são priorizados os requisitos do projeto?



Fonte: elaborado pelos autores

Para 43% dos entrevistados, ainda existe a dependência de uma pessoa distribuindo as atividades, porém 75% gostariam de ser parte de um time auto gerenciável tendo autonomia para organizar e priorizar as suas atividades, conforme Figura 26.

Figura 26 - Questão 9 - Como o trabalho é priorizado entre a equipe?



Fonte: elaborado pelos autores

O acompanhamento do projeto ocorre em um quadro próprio, para 48% dos entrevistados e este não é compartilhado com os demais envolvidos, porém todos afirmaram a importância de termos um *dashboard* integrado que forneça uma visão completa do projeto e que este seja compartilhado para todos, conforme Figura 27.

Figura 27 - Questão 10 - Todos os envolvidos conseguem visualizar os requisitos e o status do projeto em um dashboard integrado?



Fonte: elaborado pelos autores

Com base nas informações coletadas no *assessment*, podemos considerar que existe um desejo das pessoas de ter uma maior participação no início do projeto para melhorar o entendimento do negócio, e uma maior integração das ferramentas de gestão dos projetos, para prover uma visão única e compartilhada entre as equipes.

3.1.2.3 Feedback

Farroha e Farroha (2014, p. 289) afirmam que é necessária uma mudança cultural, de forma a incentivar a participação de todos os envolvidos no processo, inclusive do usuário ou área de negócios da empresa, pois isto permite um melhor gerenciamento das suas expectativas, viabilizando que as funcionalidades desenvolvidas sejam as que realmente importam para o cliente.

Analisando as respostas sobre o Feedback, identificamos que 60% dos entrevistados informaram que as ferramentas atuais não possuem análise e não provém relatório com as orientações necessárias para os times. Para 77% é importante que seja criado e incentivado o uso de ferramentas que forneçam *feedback* para diminuir as interrupções no fluxo do processo de desenvolvimento e melhorar a qualidade dos entregáveis, conforme Figura 28.

Figura 28 - Questão 11 - Existe alguma análise, orientação ou relatório produzido para criar um feedback para os times?



Fonte: elaborado pelos autores

Para 45% dos entrevistados, as reuniões de feedback para lições aprendidas só ocorrem quando há uma crise a ser resolvida, porém 85% acreditam que é importante que estas reuniões façam parte da nossa rotina e aconteçam a qualquer momento, independente se é um momento de crise ou não, conforme Figura 29.

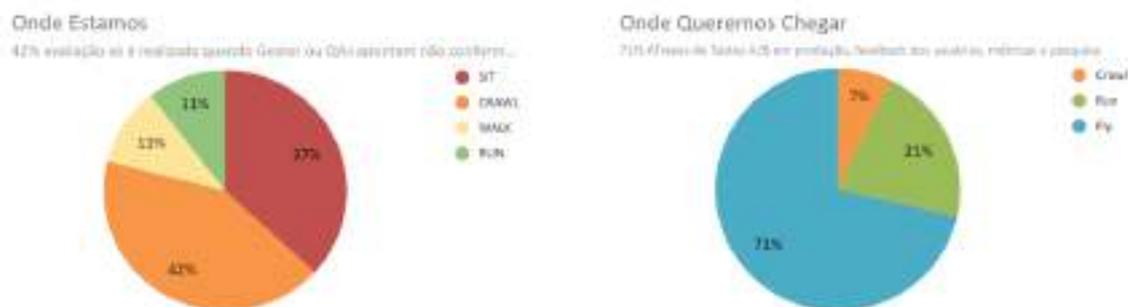
Figura 29 - Questão 12 - Quando o feedback é coletado?



Fonte: elaborado pelos autores

Em relação a avaliação da experiência do usuário nas aplicações desenvolvidas pela empresa, 42% dos entrevistados informaram que esta só acontece quando Gestor ou a equipe de Qualidade apontam alguma não conformidade e que para 71% dos entrevistados, esta avaliação deveria ser realizada através de testes de usabilidade, métricas e pesquisas de usuários, conforme Figura 30.

Figura 30 - Questão 13 - Como você avalia a experiência do usuário?



Fonte: elaborado pelos autores

Para 60% dos entrevistados, o feedback (seja para elogios ou reportar problemas) do usuário no uso das nossas aplicações só acontece quando estes entram em contato, por exemplo, através de canais de ouvidoria. Enquanto 53% dos entrevistados entendem que a empresa poderia utilizar técnicas como: análise de causa raiz, entrevistas e definição de métricas para prover melhoria contínua, conforme Figura 31.

Figura 31 - Questão 14 - Como o feedback do usuário é analisado e otimizado para melhoria contínua?



Fonte: elaborado pelos autores

Quanto ao compartilhamento de informações entre os times, 65% dos entrevistados informaram que não sabem o que acontece nos outros times e só são acionados quando há problemas, porém 71% acreditam que seria importante que os times trabalhassem de forma coesa e compartilhassem as informações referentes ao projeto de forma contínua, conforme Figura 32.

Figura 32 - Questão 15 - Existe compartilhamento do feedback entre os times? Como isto é praticado?



Fonte: elaborado pelos autores

De forma resumida, podemos entender que as equipes possuem o desejo de que ferramentas que forneçam feedback façam parte do fluxo completo de desenvolvimento, com cerimônias de lições aprendidas fazendo parte da nossa rotina, independente de crises. É importante que seja criado um processo de coleta do feedback dos usuários para evoluir nossos produtos, através de melhorias nos ambientes ou ajustes em funcionalidades.

3.1.2.4 Integração Contínua

Para Humble e Farley (2010, p. 55), o uso de práticas de Integração Contínua, permite que a cada mudança no software, toda a aplicação seja novamente compilada, testes automatizados executados e, caso ocorra alguma falha nos processos de teste ou compilação, todo o time seja acionado para resolver o problema imediatamente.

Analisando as respostas sobre o Integração Contínua, identificamos que 60% dos entrevistados informaram que o código-fonte dos seus projetos são armazenados em repositórios de versionamento, porém não possuem um pipeline automatizado, enquanto para 56% é importante a adoção de políticas para gerenciamento das bibliotecas e dependências com integração de um pipeline automatizado, conforme Figura 33.

Figura 33 - Questão 16 - Como o seu código-fonte é gerenciado e como é feito o controle de versão?



Fonte: elaborado pelos autores

Em relação a dívida técnica, 56% dos entrevistados informaram que esta não é revisitada para refatoração e 63% dos entrevistados consideram importante adotarmos um processo de melhoria contínua para identificar, avaliar, priorizar e mitigar as dívidas técnicas, conforme visto na Figura 34.

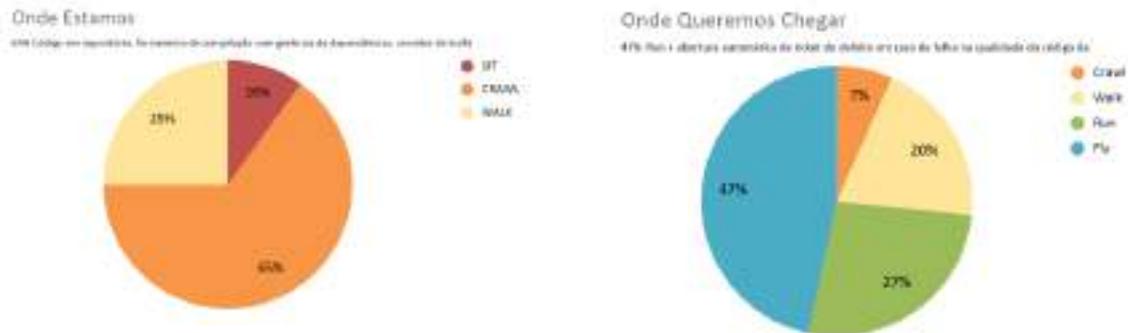
Figura 34 - Questão 17 - Como são controladas e medidas as dívidas técnicas?



Fonte: elaborado pelos autores

Atualmente, para 65% dos entrevistados, o processo de integração, construção, deploy e entrega é realizado a partir do código-fonte em repositório, com ferramentas de compilação e gerenciamento de dependências em um servidor de build, porém sem automação de testes. Sendo que 47% entendem que além de todas as fases anteriores, seria interessante incluir a abertura automática de ticket de defeito em caso de falha na qualidade do código, conforme Figura 35.

Figura 35 - Questão 18 - Como é o processo de integração, construção, deploy e entrega?



Fonte: elaborado pelos autores

Para 44% dos entrevistados, o gerenciamento da qualidade do código e do build é realizado de forma manual, sem ferramentas de avaliação da qualidade do código fonte ou de monitoração do build. Enquanto 69% entendem que para melhorar é necessário um processo automatizado integrado com a ferramenta de build e de revisão de código, conforme Figura 36.

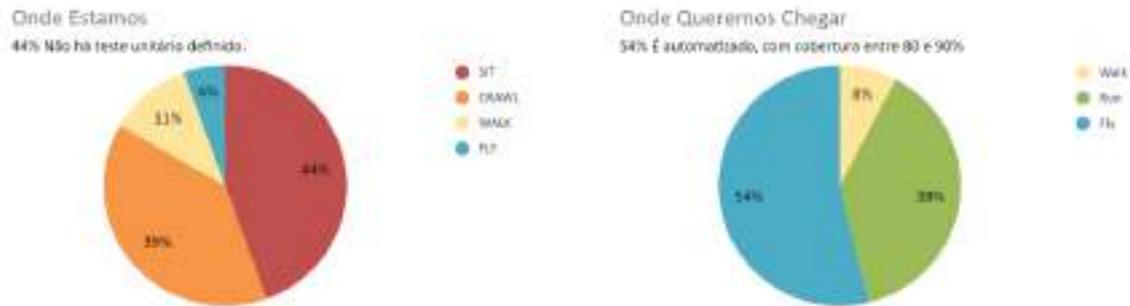
Figura 36 - Questão 19 - Como é o processo de gerenciamento de qualidade do código e build?



Fonte: elaborado pelos autores

Para 44% dos entrevistados, o processo de desenvolvimento não engloba os testes unitários, enquanto 54% entendem que os testes deveriam ser automatizados com cobertura acima de 80%, conforme Figura 37.

Figura 37 - Questão 20 - É realizado teste unitário durante a fase de desenvolvimento?



Fonte: elaborado pelos autores

Com base nos resultados coletados, podemos entender que os times possuem o desejo de ter um pipeline integrado com práticas de integração contínua como por exemplo: revisão de código, testes automatizados e geração de tickets automatizados para os defeitos encontrados. Percebemos também que pequenos passos para a Integração Contínua já foram dados e entendemos que ainda precisamos avançar em outras práticas que o pipeline não contempla.

3.1.2.5 Entrega Contínua

A Entrega Contínua, para Humble e Farley (2010, Prefácio), possibilita que o time de desenvolvimento consiga entregar o software em produção de forma confiável, previsível e com baixo nível de riscos.

Analisando as respostas sobre Entrega Contínua, para 53% dos entrevistados o processo de implantação ocorre de forma automatizada em alguns ambientes com toda configuração versionada. Para melhoria do processo de implantação, 40% dos entrevistados entendem que a implantação deveria ser gerenciada com processos testados com possibilidade de *rollback*, conforme Figura 38.

Figura 38 - Questão 21 - Como é o processo de implantação?



Fonte: elaborado pelos autores

Para 36% dos entrevistados, o processo de implantação da release é replicável, porém coordenado de forma manual. Enquanto 58% gostariam que o processo fosse totalmente replicável, medido e continuamente melhorado, conforme Figura 39.

Figura 39 - Questão 22 - O processo de implantação da release é replicável?



Fonte: elaborado pelos autores

Para 44% dos entrevistados, o processo de controle de atualização de release em produção acontece após a aprovação da Qualidade e 57% entendem a importância de um pipeline automatizado que garanta os *check-ins* para cada release de forma segura, conforme Figura 40.

Figura 40 - Questão 23 - Como é o processo de controle de atualização (promoção) de release em produção?



Fonte: elaborado pelos autores

Técnicas de *deploy* para implantação, como *Canary Release*, *Blue-Green* e *Feature Toggle*, ainda são desconhecidas para 47% dos entrevistados. A adoção destas técnicas deveria acontecer frequentemente, de acordo 78% dos entrevistados, conforme Figura 41.

Figura 41 - Questão 24 - Utiliza alguma a técnica de deploy para a implantação, como Canary Release, Blue-Green, Featured Toggle?



Fonte: elaborado pelos autores

A frequência de implantação de novas funcionalidades acontece pelo menos uma vez por mês, para 50% dos entrevistados. Com um pipeline automatizado, com práticas de Integração Contínua sendo executadas de forma efetiva, 62% dos entrevistados entendem que será possível implantar novas funcionalidades com qualidade, uma ou mais vezes por dia, conforme Figura 42.

Figura 42 - Questão 25 - Com que frequência são implantadas novas funcionalidades em produção?



Fonte: elaborado pelos autores

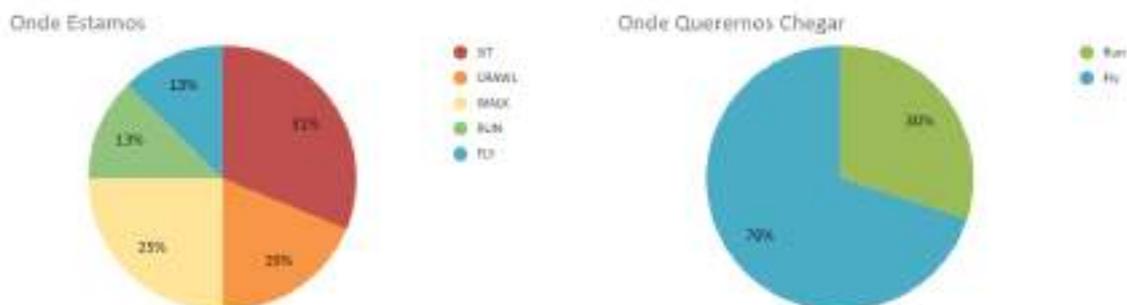
Com base nos resultados, entendemos que os times desejam expandir os benefícios da automação de testes e feedback imediato para os próximos estágios não cobertos pela Integração Contínua, com a adoção de técnicas de *deploy* para garantir qualidade e segurança na liberação das releases.

3.1.2.6 Infraestrutura como código

Para Oliveira (2020) a Infraestrutura como código torna o processo de administração de infra mais confiável, seguro e controlado, pois, todas as alterações no código ficam registradas, facilitando a rastreabilidade do conteúdo e proporcionando uma maior velocidade no processo de entrega de uma nova infraestrutura.

Analisando as respostas sobre Infraestrutura como Código, percebemos que para 31% dos entrevistados, o tempo para disponibilização de um ambiente físico é de até 40 horas. Enquanto 70% dos entrevistados informaram que seria ideal uma disponibilização do ambiente em nuvem de 2 a 4 horas, conforme Figura 43.

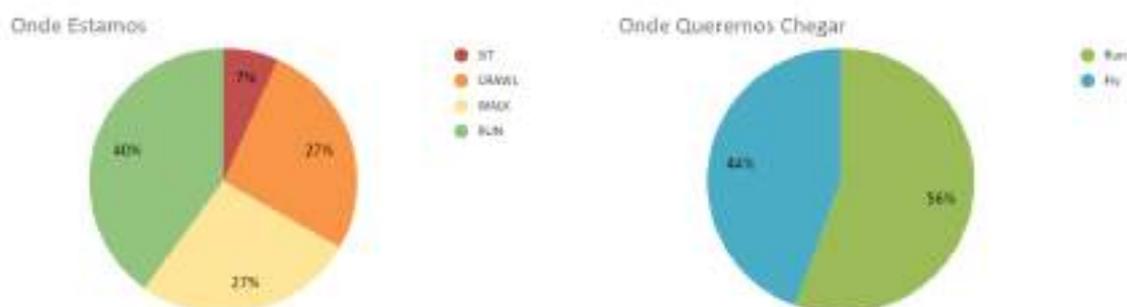
Figura 43 - Questão 26 - Qual o tempo necessário para disponibilizar o ambiente?



Fonte: elaborado pelos autores

Para os entrevistados nós já atingimos o nível de maturidade dentro da equipe Java, que será considerada para o piloto deste projeto. Esta equipe atualmente utiliza a plataforma *Openshift* para os ambientes de Desenvolvimento, Homologação e Produção. Os resultados podem ser vistos na Figura 44.

Figura 44 - Questão 27 - Como são definidos os ambientes para as fases de Desenvolvimento, Homologação e Produção?



Fonte: elaborado pelos autores

A automação do processo de provisionamento de ambientes, ainda é realizado através da criação de Máquina Virtual, para 36% dos entrevistados. Porém 70% dos entrevistados entendem que este processo deve ser totalmente automatizado e orquestrado pela plataforma *Openshift*, conforme Figura 45.

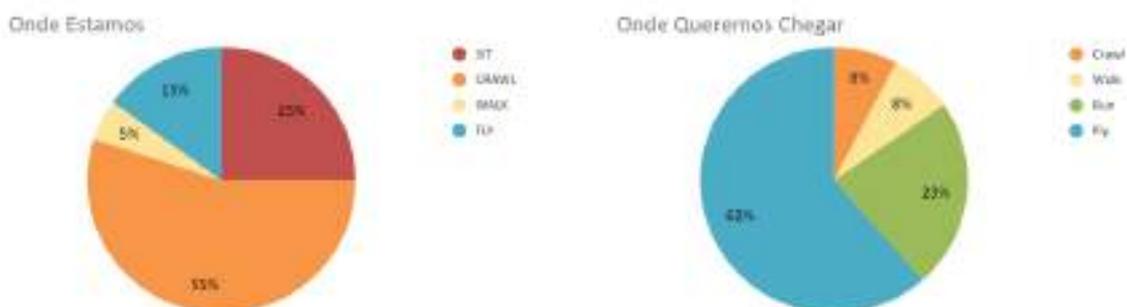
Figura 45 - Questão 28 - Qual o nível de automação do processo de provisionamento de ambientes?



Fonte: elaborado pelos autores

Quanto à disponibilidade dos recursos, 55% dos entrevistados afirmam que o planejamento e a entrega dos ambientes, da forma como é feito atualmente, pode afetar a data dos projetos. Enquanto 62% entendem que um bom planejamento e com o uso de arquitetura em nuvem, os serviços podem estar disponíveis e configuráveis sob demanda, conforme Figura 46.

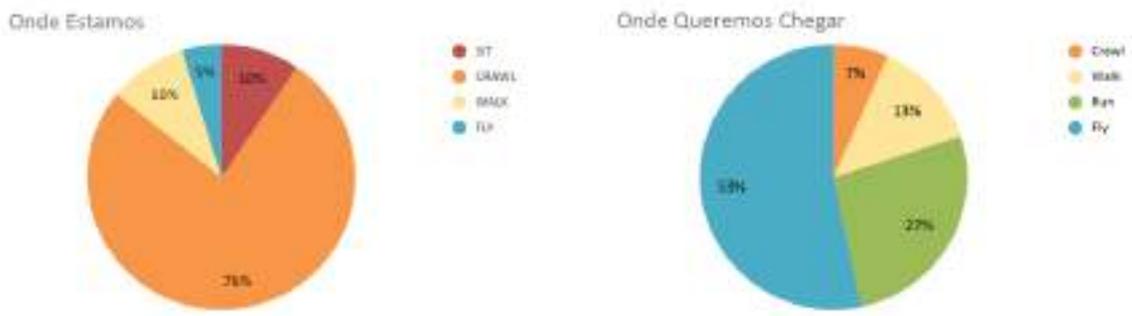
Figura 46 - Questão 29 - A entrega dos ambientes é considerada durante o planejamento do projeto?



Fonte: elaborado pelos autores

As equipes são notificadas dos problemas que acontecem com as aplicações em produção, através de reclamações dos usuários finais que podem chegar por telefone, e-mail, ouvidoria ou redes sociais, de acordo com 76% dos entrevistados. A adoção de ferramentas de Telemetria, é importante para 53% dos entrevistados, como forma de prever anomalias e antecipar os problemas antes que eles ocorram, conforme Figura 47 .

Figura 47 - Questão 30 - Como os problemas são informados aos times de desenvolvimento e operações?



Fonte: elaborado pelos autores

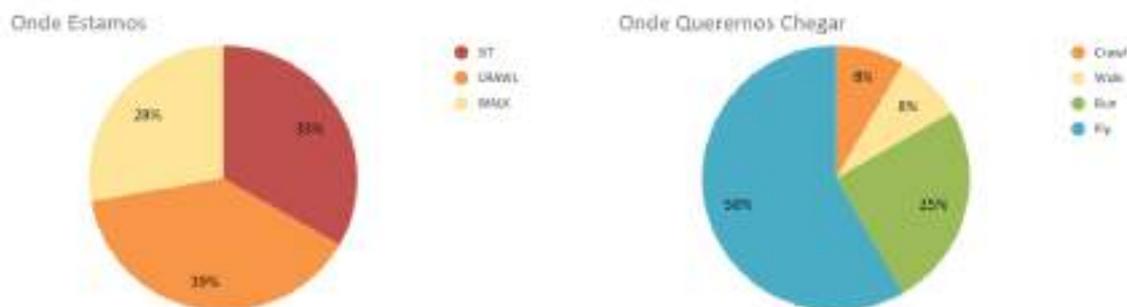
Com base nas informações coletadas, podemos concluir que os times desejam aplicar Infraestrutura como Serviço, para provisionar e gerenciar nossa infraestrutura e adotar ferramentas de telemetria, para acompanhar e mitigar os problemas antes que eles ocorram.

3.1.2.6 Monitoração

De acordo com Levita (2017, p. 5), a prática de Monitoração permite que métricas estejam disponíveis para todas as partes envolvidas na entrega do software, tenham uma visão clara de como o time está progredindo e identifiquem eventuais gargalos no processo.

Analisando as respostas sobre Monitoração, 39% dos entrevistados informaram que o gerenciamento de eventos e incidentes, acontece sob demanda com investigação em logs, através de comunicação à central. Enquanto 58% entendem a importância do uso de ferramentas centralizadas com integração automática às ferramentas de gestão, conforme Figura 48.

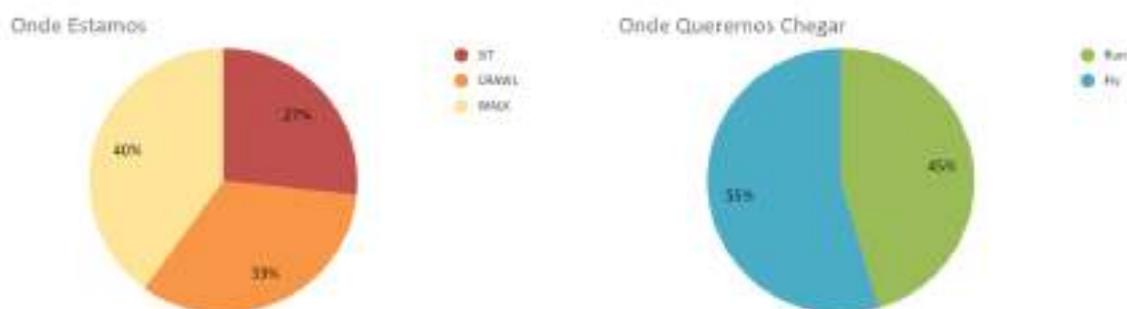
Figura 48 - Questão 31 - Como é feito o gerenciamento de log, eventos e incidentes?



Fonte: elaborado pelos autores

O pipeline atual permite identificar algumas falhas, para 40% dos entrevistados, porém não identifica falhas em testes automatizados, por não ter essa cobertura. Para 55% é importante que o pipeline esteja integrado com uma suíte de testes automatizados, *rollback* de banco e aplicação, para identificar falhas e auxiliar a tomada de decisão do time, conforme Figura 49.

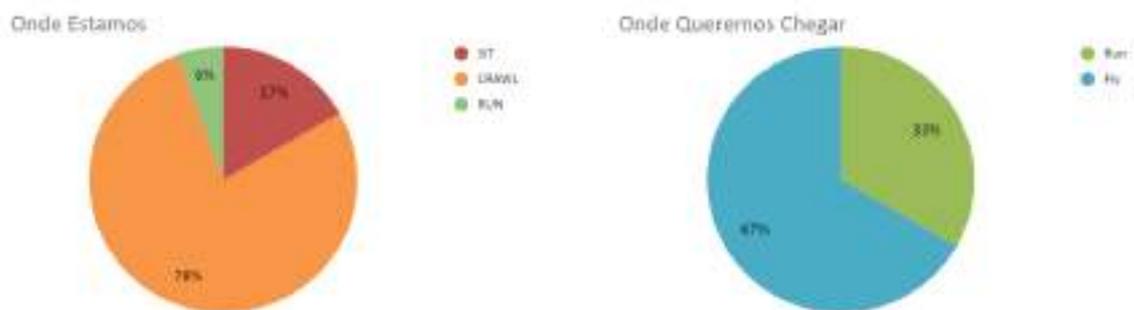
Figura 49 - Questão 32 - O pipeline permite identificar rapidamente as falhas para tomada de decisão do time?



Fonte: elaborado pelos autores

O envolvimento da equipe de Operações, para opinar sobre arquitetura, ambientes, implantação e planejamento da infraestrutura, ocorre apenas quando necessário de acordo com o projeto, para 78% dos entrevistados. Enquanto 67% gostariam de trabalhar de forma colaborativa, com o time DevOps focado desde o início do projeto, conforme Figura 50.

Figura 50 - Questão 33 -A equipe de Operações é envolvida no início do projeto, para opinar sobre ambientes, implantação e planejamento de capacidade?



Fonte: elaborado pelos autores

Atualmente, para 44% dos entrevistados, a aplicação e infraestrutura dos seus projetos não é escalável. Enquanto 70% afirmaram que é importante termos um ambiente auto escalável, baseado em parâmetros configuráveis, conforme Figura 51.

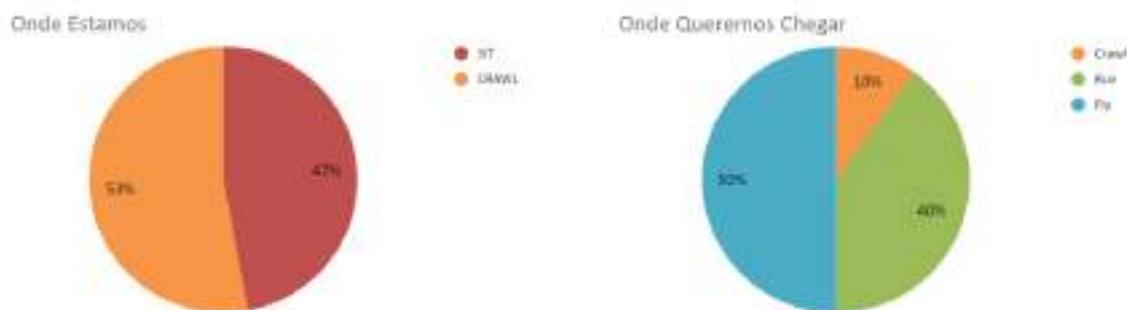
Figura 51 - Questão 34 - A sua aplicação e infraestrutura são auto escaláveis?



Fonte: elaborado pelos autores

Atualmente, a avaliação das métricas de performance para solucionar problemas só ocorrem sob demanda, para 53% dos entrevistados. Enquanto 50% entendem a importância do uso de ferramentas de telemetria, conforme Figura 52.

Figura 52 - Questão 35 - As equipes são capazes de analisar métricas de performance para solucionar problemas antes que eles afetem os usuários?



Fonte: elaborado pelos autores

Com base nos dados coletados, entendemos que os times desejam feedback rápido para evitar que falhas aconteçam em ambiente de produção, uso de ferramentas de telemetria para análise de métricas e um envolvimento prévio no projeto, para um melhor planejamento.

3.1.2.7 Testes Automatizados

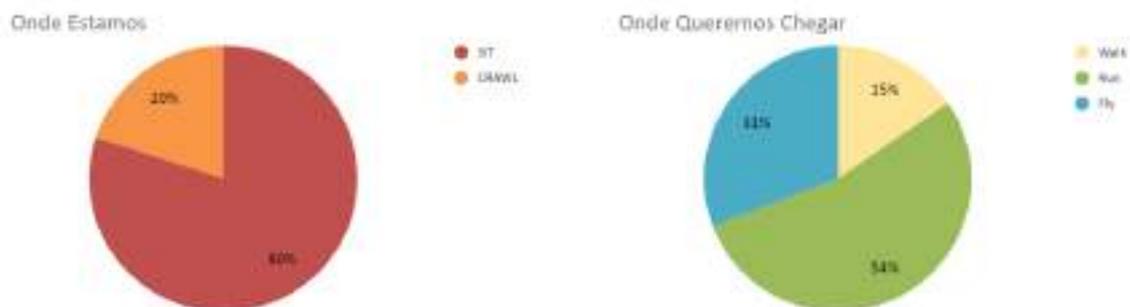
Para Duvall, Matias e Glover (2007, p. 15), não existe integração contínua sem a implementação da prática de Testes Automatizados, pois é por meio deles que os desenvolvedores e demais partes envolvidas no projeto têm confiança nas mudanças feitas no software.

Sharma e Coyne (2015, p. 12) explicam que o objetivo é testar o quanto antes e continuamente, durante o ciclo de vida do desenvolvimento, o que leva a uma redução nos custos e nos tempos de testes, bem como uma melhor qualidade do software.

Analisando as respostas sobre Testes Automatizados, identificamos que 80% dos entrevistados informaram que não existe automação dos testes nos projetos que atuam e que para 54% é importante que os projetos possuam uma cobertura de no mínimo, 60% dos testes automatizados, conforme Figura 53.

Figura 53 - Questão 36 - Qual o percentual de testes automatizados realizados em todas as fases de desenvolvimento?

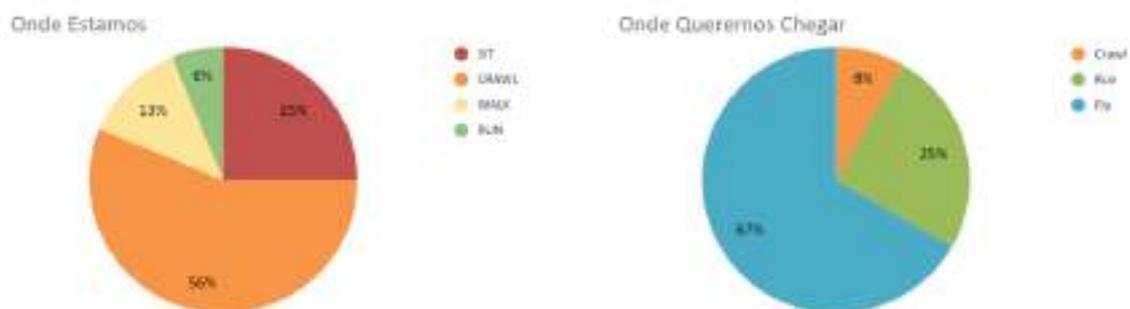
53 -



Fonte: elaborado pelos autores

Para 56% dos entrevistados, os casos de testes só são elaborados sob demanda pela equipe de Qualidade e como melhoria, 67% entendem que o time de Qualidade deveria participar ativamente do projeto, apoiando na escrita das histórias, auxiliando o desenvolvedor na criação dos cenários de testes e aplicação de práticas como BDD e TDD, conforme Figura 54.

Figura 54 - Questão 37 - Em que momento são elaborados os casos de testes?

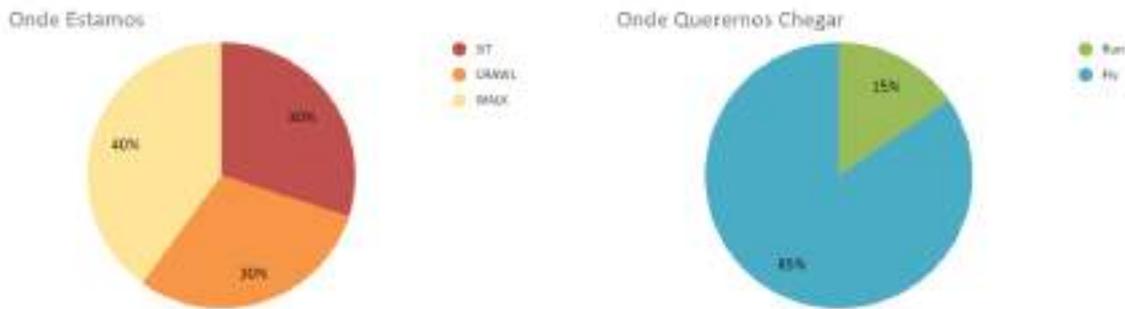


Fonte: elaborado pelos autores

Atualmente, os testes de funcionalidade acontecem após a entrega da sprint, para 40% dos entrevistados. Para 85% a participação de um profissional de Qualidade

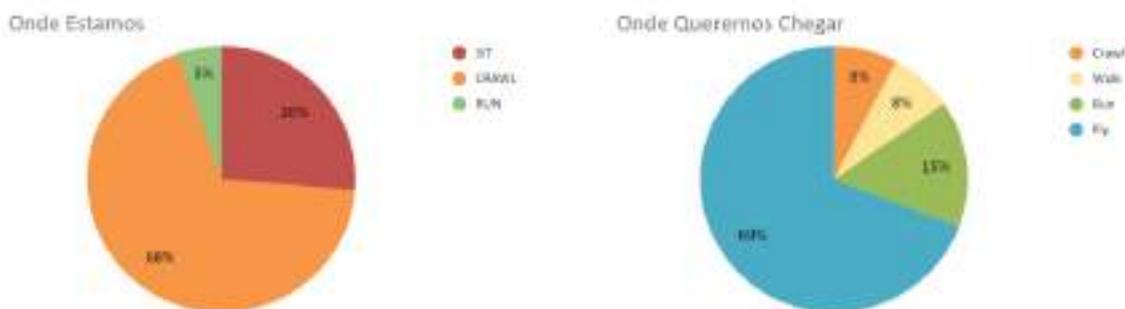
durante a sprint é fundamental, pois assim poderíamos realizar os testes antes da entrega, conforme Figura 55.

Figura 55 - Questão 38 - Com que frequência é realizado o teste de funcionalidade?



Atualmente, os testes não funcionais como performance, vulnerabilidade, carga e estresse, são obrigatórios e acontecem sempre na entrada da aplicação em produção, de acordo com 68% dos entrevistados. Porém, 69% entendem que estes poderiam ser automatizados no pipeline, conforme Figura 56.

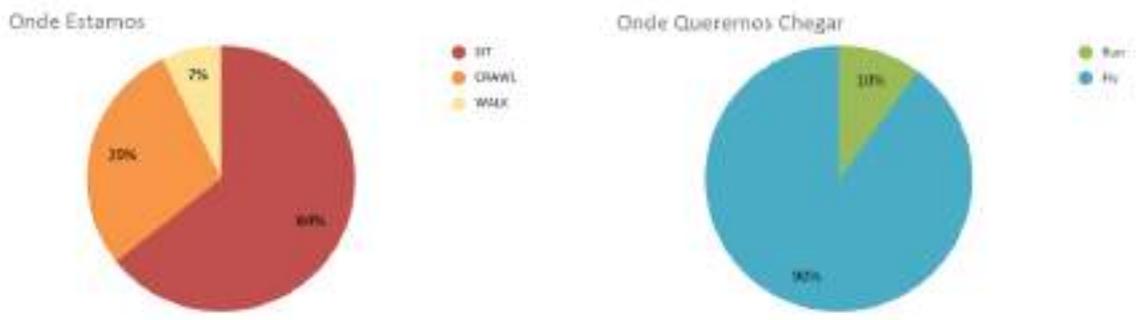
Figura 56 - Questão 39 - Quando são realizados os testes não funcionais (testes de vulnerabilidade, desempenho, sonar, etc)?



Fonte: elaborado pelos autores

Atualmente, não existe compartilhamento dos resultados dos casos de testes funcionais entre as equipes, porque estes não são automatizados, de acordo com 64% dos entrevistados. Porém, 90% informaram que os casos de testes poderiam ser executados automaticamente pelo pipeline e os resultados disponibilizados em um *dashboard*, compartilhado para todos, conforme Figura 57.

Figura 57 - Questão 40 - O desenvolvedor compartilha o resultado dos testes com os QAs?



Fonte: elaborado pelos autores

Sendo assim, com base nos resultados coletados, podemos concluir que os times desejam que os perfis de Qualidade participem ativamente do ciclo do desenvolvimento, para dar apoio a construção dos casos de testes, das histórias, da automação dos testes funcionais e não funcionais e assim apoiar na melhoria da qualidade das entregas.

3.1.3 Radar de Maturidade

A partir da análise dos resultados por Práticas DevOps, foi possível gerar o Radar de Maturidade que mostra o panorama atual, exibindo de forma visual o ponto “Onde Estamos” e para “Onde Queremos Chegar”, conforme Figura 58.

Observamos que o ponto mais maduro do Radar é na prática de Infraestrutura Como Código (Maturidade Nível 3), uma vez que a equipe java utiliza a Plataforma *Openshift* para provisionamento dos ambientes e criação de pipelines de build automatizados.

Em contraste a isto, percebemos que as práticas de Testes são o nosso nível de maturidade mais baixo (Maturidade Nível 1), por conta de não termos testes automatizados e da ausência de uma cultura que fortaleça a importância da automação dos testes.

Os demais itens, Pessoas & Cultura, Planejamento, Feedback, Integração Contínua, Entrega Contínua e Monitoração, apesar de existirem se encontram em uma fase bastante embrionária com iniciativas pontuais e incentivo para evolução.

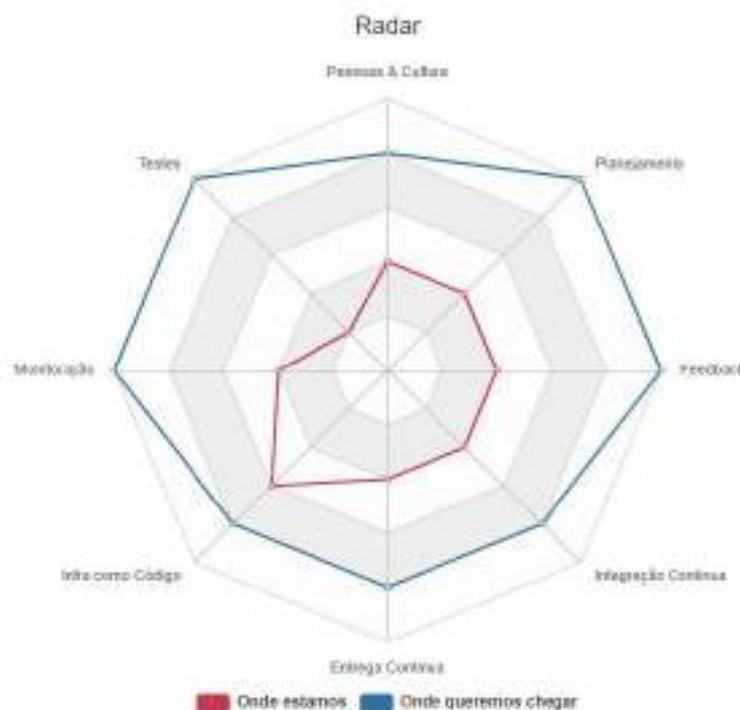
Percebemos que é desejo dos times atingir a excelência na realização das práticas com a cultura DevOps adotada em toda Organização e para isso precisamos evoluir em pequenos passos, para ganharmos experiência.

O Radar de Maturidade nos indica quão distante estamos da maturidade DevOps que queremos implantar na Empresa e as principais práticas que precisam ser evoluídas.

Importante ressaltar que atingir a Maturidade Nível 5, não significa que chegamos no nosso objetivo final, porque a melhoria contínua é um processo cíclico sem fim, pois sempre há novas oportunidades de melhoria para serem identificadas e colocadas em prática.

As ações que nos levarão para mais perto dos nossos objetivos serão apresentadas no item 3.3 Plano de Ação.

Figura 58 - Radar de Maturidade nas Práticas DevOps



Fonte: elaborado pelos autores

3.2 Mapeamento das Dores e Necessidades

Com os feedbacks coletados durante as reuniões do Café com Devops e Talk com Especialistas, fizemos o mapeamento das Dores e Necessidade, classificando-os com base nos conceitos do framework CALMS, apresentando no Capítulo 1 Introdução.

Entender de fato, qual é a dor é um trabalho muito complexo, pois as pessoas podem ter dores iguais ou semelhantes, mas a causa raiz de cada um pode ser diferente. Era importante realizar este mapeamento para termos certeza de que estávamos no caminho certo e que o nosso projeto atenderia os anseios dos nossos colaboradores. Os feedbacks que recebemos tinham um valor inigualável e não poderiam ser ignorados.

Cada comentário, opinião e sugestão coletado, foi transportado para o quadro virtual do Miro e analisado pontualmente, para identificarmos a Dor ou Necessidade e assim descobrir a causa-raiz do problema.

Estudar a dor é uma tarefa subjetiva e para nos guiarmos definimos como Dor, aquele contratempo, preocupação ou problema que os participantes possuem e Necessidade, como aquilo que é indispensável e que não se pode deixar de ter.

Identificando a Dor e a Necessidade real, verificamos se estava alinhado com as respostas obtidas no Assessment e se algo que não foi mapeado surgiu. Com isso, foi possível gerar um Plano de Ação que correspondesse às expectativas dos participantes.

O mapeamento foi realizado apenas com os feedbacks que estavam relacionados ao escopo do nosso projeto. Os feedbacks fora de escopo estão destacados no APÊNDICE B - Feedback das reuniões, para que possam servir eventualmente para outras ações.

Apresentaremos a seguir como foi o processo de avaliação dos feedbacks e a identificação das Dores e Necessidades, para cada item do framework CALMS.

3.2.1 Cultura (Culture)

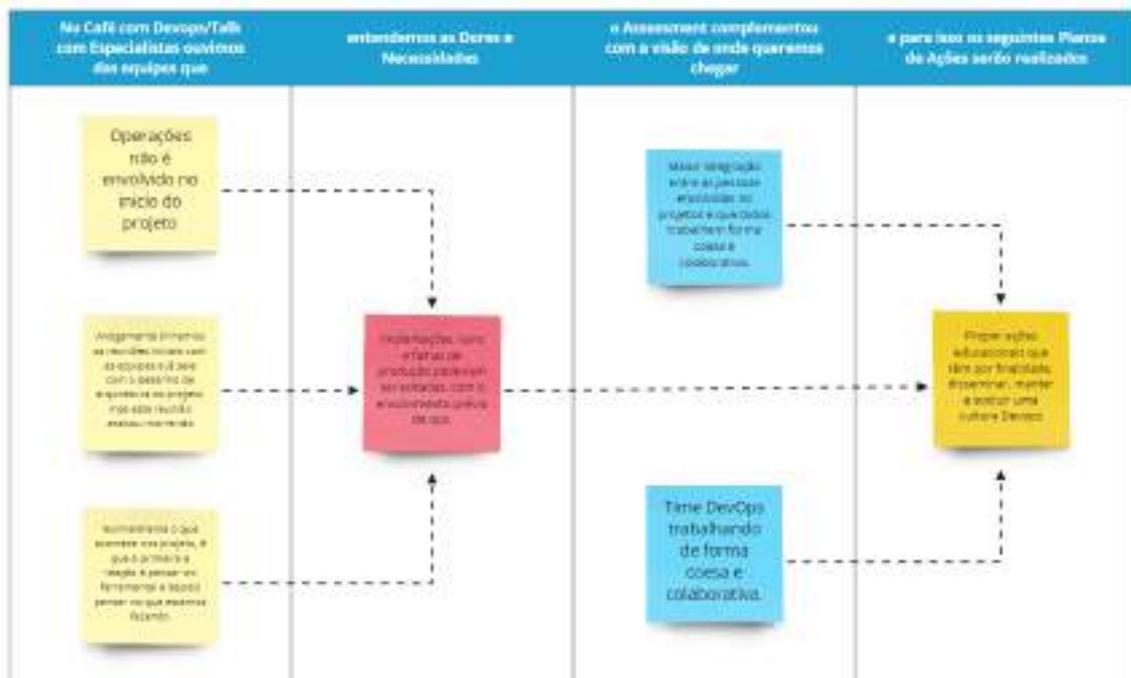
A cultura é o fator fundamental para o sucesso da adoção do DevOps. Portanto, tem a ver como os times se relacionam, como se organizam, como compartilham experiências e informações. A cultura é um ponto de atenção em qualquer processo

de transformação da empresa, pois uma cultura ruim influencia diretamente na qualidade dos resultados. Alguns dos pilares desta etapa são:

- Foco nas pessoas e em como elas pensam;
- Adesão de uma cultura da experimentação;
- Promoção de uma cultura de integração entre desenvolvimento e operações;
- Mudança de Comportamento;
- Bom relacionamento entre as áreas;
- Responsabilidade compartilhada entre equipes;
- Adesão de uma cultura livre de culpa por erros nos processos.

Classificamos os feedbacks que entendemos estar relacionados à Cultura e fomos analisando para chegar a dor ou necessidade. Em alguns casos, tivemos feedbacks relacionados a uma ou mais dores ou necessidades. A Figura 59 apresenta um exemplo de como foi realizado este processo, até chegarmos ao Plano de Ação.

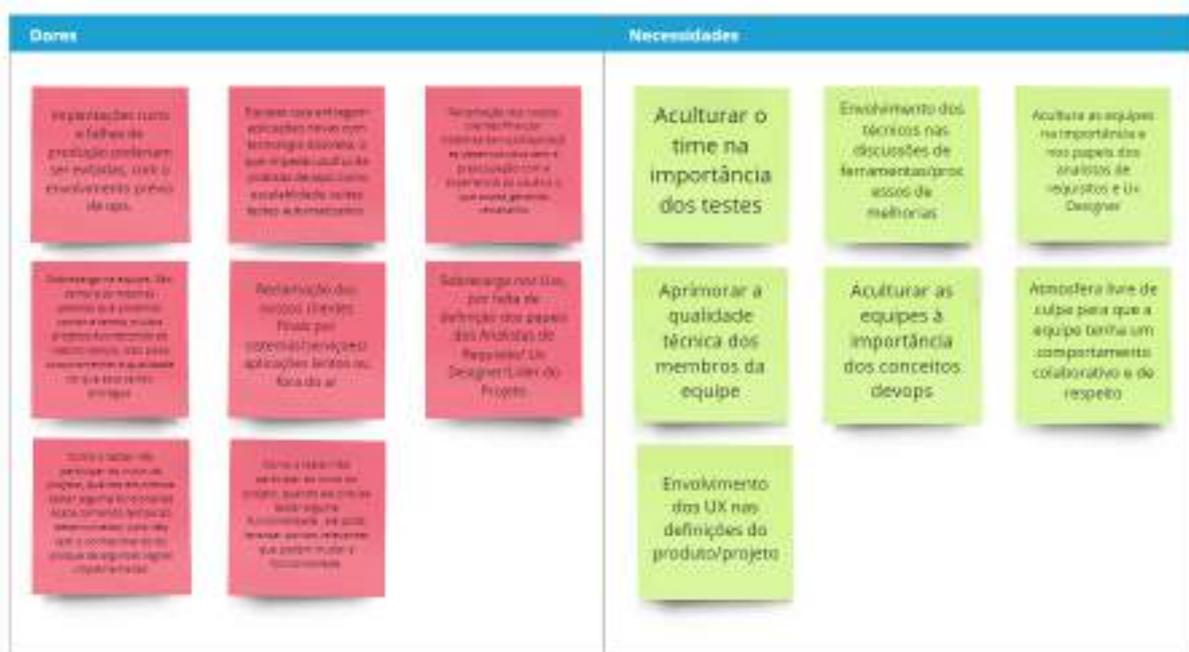
Figura 59 - Mapeamento dos feedbacks relacionados à Cultura



Fonte: elaborado pelos autores

Com este fluxo, conseguimos identificar as seguintes Dores e Necessidades, conforme Figura 60.

Figura 60 - Dores e Necessidades identificadas relacionadas à Cultura



Fonte: elaborado pelos autores

Com o resultado do assessment, tivemos a visão de onde o time deseja chegar, conforme Figura 61.

Figura 61 - Resultado do assessment - Cultura



Fonte: elaborado pelos autores

Com todas as informações coletadas, identificamos os seguintes Planos de Ação, conforme Figura 62.

Figura 62 - Plano de Ação Cultura



Fonte: elaborado pelos autores

O processo completo que foi realizado para identificar as Dores e Necessidades relacionadas à Cultura, apresentadas acima, encontra-se no APÊNDICE E - Mapeamento das Dores e Necessidades.

3.2.2 Automação (Automation)

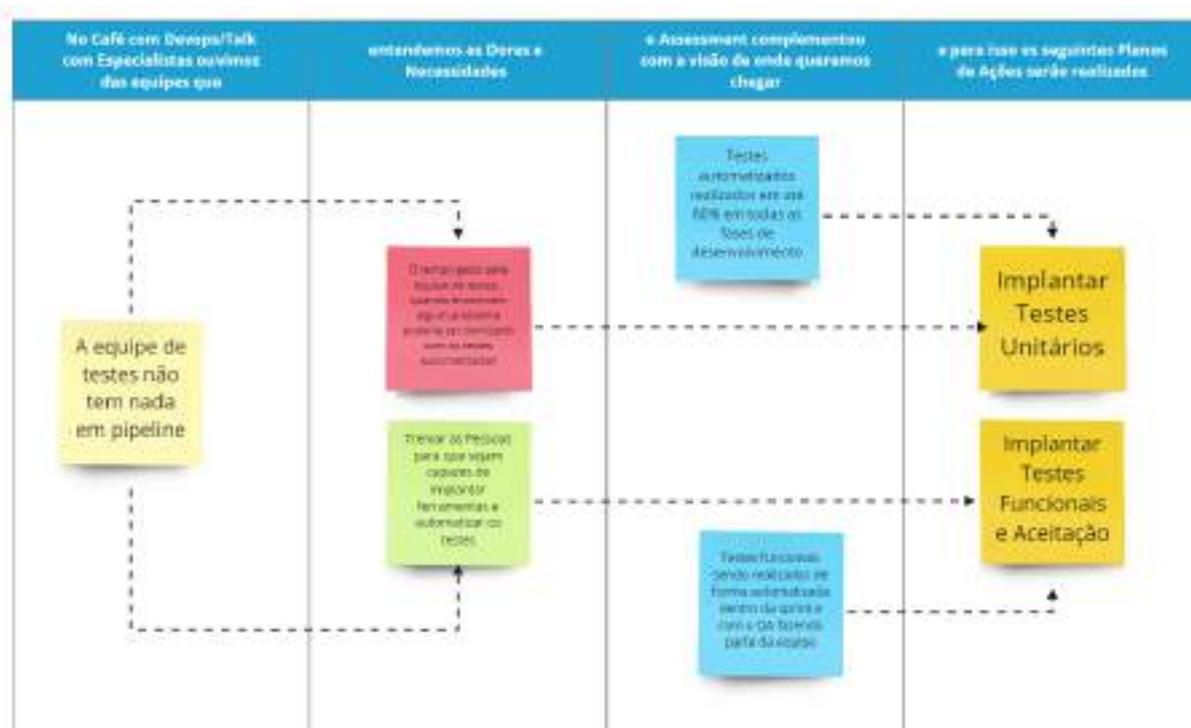
A automação traz grandes benefícios para a empresa, não só pela redução dos erros humanos. A automação também realiza a documentação de cada processo, auxiliando na padronização dos passos. Esta etapa deve ser executada com cuidado e atenção, pois caso seja feito com gaps e gargalos, pode acabar prejudicando o processo de desenvolvimento, entregando soluções com defeitos e problemas.

Alguns dos pilares desta etapa são:

- Automatizar tudo que for possível;
- Adotar ferramentas para manter tudo aquilo que for necessário para que o processo de desenvolvimento, testes e releases seja um sucesso;
- Realizar prática de Entrega Contínua e Infraestrutura como código;
- Utilizar ferramentas de Orquestração.

Classificamos os feedbacks que entendemos estar relacionados à Automação e fomos analisando para chegar a dor ou necessidade. Em alguns casos, tivemos feedbacks relacionados a uma ou mais dores ou necessidades. A Figura 63 apresenta um exemplo de como foi realizado este processo, até chegarmos ao Plano de Ação.

Figura 63 - Mapeamento dos feedbacks relacionados à Automação



Fonte: elaborado pelos autores

Com este mapeamento, conseguimos identificar as seguintes Dores e Necessidades, conforme Figura 64.

Figura 64 - Dores e Necessidades identificadas relacionadas à Automação



Fonte: elaborado pelos autores

Com o resultado do assessment, tivemos a visão de onde o time deseja chegar, conforme Figura 65.

Figura 65 - Resultado do assessment - Automação



Fonte: elaborado pelos autores

Com todas as informações coletadas, identificamos os seguintes Planos de Ação, conforme Figura 66.

Figura 66 - Plano de Ação Automação



Fonte: elaborado pelos autores

O processo completo que foi realizado para identificar as Dores e Necessidades relacionadas à Automação, apresentadas acima, encontra-se no APÊNDICE E - Mapeamento das Dores e Necessidades.

3.2.3 Fluxo (Lean)

Lean está ligado à melhoria no desempenho dos processos da empresa, analisando, identificando e redesenhando os pontos que podem estar causando maiores gargalos e índices de retrabalho. O foco é no valor, na redução de custos e no cliente.

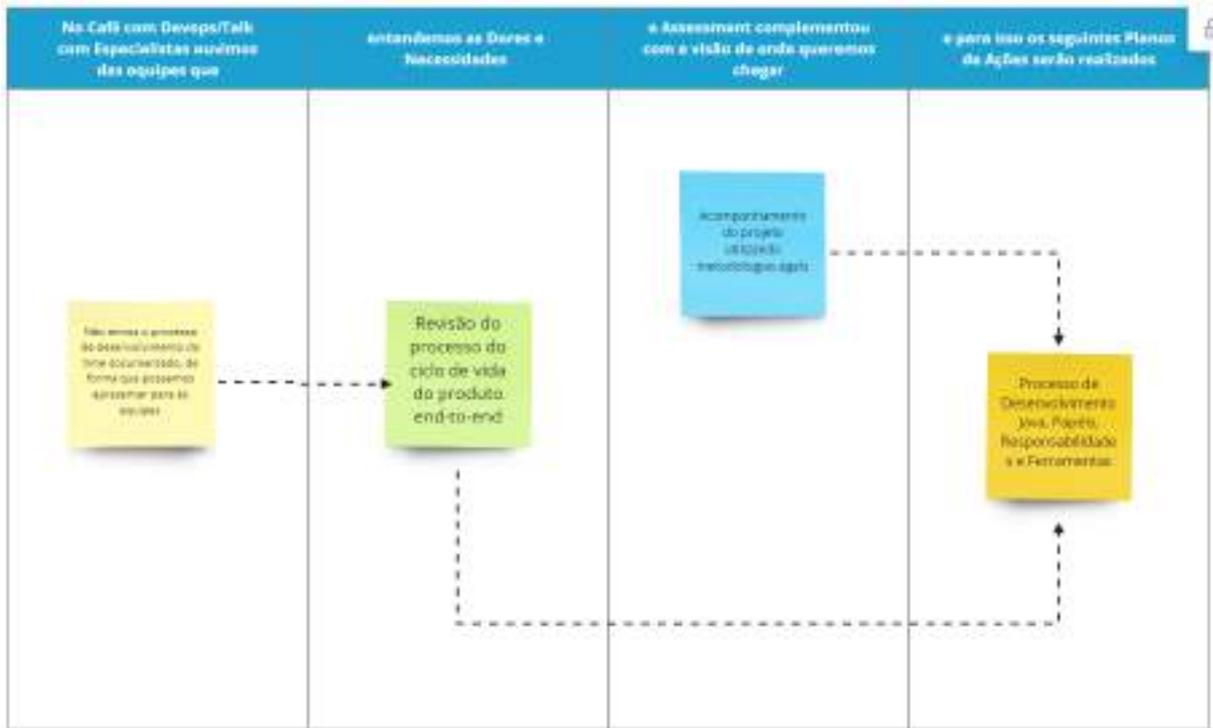
Alguns dos pilares desta etapa são:

- Mantenha o foco no cliente e no valor gerado para ele;
- Elimine gradualmente os focos de desperdício;
- Adote a entrega contínua;
- Promova entregas pequenas e contínuas;
- Faça o simples;
- Foque no aumento de aprendizagem.

Classificamos os feedbacks que entendemos estar relacionados ao Fluxo (Lean) e fomos analisando para chegar a dor ou necessidade. Em alguns casos, tivemos feedbacks relacionados a uma ou mais dores ou necessidades. A Figura 67

apresenta um exemplo de como foi realizado este processo, até chegarmos ao Plano de Ação.

Figura 67 - Mapeamento dos feedbacks relacionados à Fluxo (Lean)



Fonte: elaborado pelos autores

Com este mapeamento, conseguimos identificar as seguintes Dores e Necessidades, conforme Figura 68.

Figura 68 - Dores e Necessidades identificadas relacionadas à Fluxo (Lean)



Fonte: elaborado pelos autores

Com o resultado do assessment, tivemos a visão de onde o time deseja chegar, conforme Figura 69.

Figura 69 - Resultado do assessment - Fluxo (Lean)



Fonte: elaborado pelos autores

Com todas as informações coletadas, identificamos os seguintes Planos de Ação, conforme Figura 70.

Figura 70 - Plano de Ação Fluxo (Lean)



Fonte: elaborado pelos autores

O processo completo que foi realizado para identificar as Dores e Necessidades relacionadas a Fluxo (Lean), apresentadas acima, encontra-se no APÊNDICE E - Mapeamento das Dores e Necessidades.

3.2.4 Medição (Measurement)

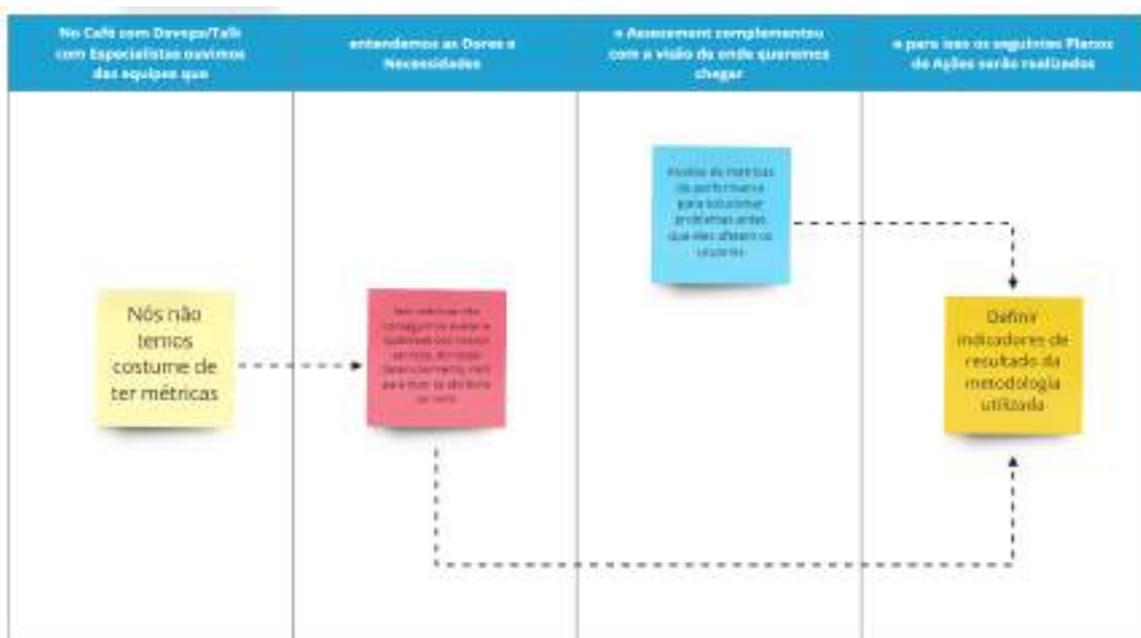
A Medição permite adquirir o conhecimento sobre a empresa, gerar previsibilidade sobre problemas, imprevistos e acidentes que possam vir a ocorrer, identificando suas possíveis causas e os primeiros sinais antes que aconteçam. Dessa forma, é possível se prevenir, evitando as falhas e criando planos de contingência, a fim de reduzir os prejuízos.

Alguns dos pilares desta etapa são:

- Medir tudo que for possível;
- Apresentar melhorias constantes;
- Adotar a melhoria contínua em seu negócio;
- Adotar o monitoramento e controle;
- Realizar análises de desempenho constantemente.

Classificamos os feedbacks que entendemos estar relacionados a Medição e fomos analisando para chegar a dor ou necessidade. Em alguns casos, tivemos feedbacks relacionados a uma ou mais dores ou necessidades. A Figura 71 apresenta um exemplo de como foi realizado este processo, até chegarmos ao Plano de Ação.

Figura 71 - Mapeamento dos feedbacks relacionados à Medição



Fonte: elaborado pelos autores

Com este mapeamento, conseguimos identificar as seguintes Dores e Necessidades, conforme Figura 72.

Figura 72 - Dores e Necessidades identificadas relacionadas à Medição



Fonte: elaborado pelos autores

Com o resultado do assessment, tivemos a visão de onde o time deseja chegar, conforme Figura 73.

Figura 73 - Resultado do assessment - Medição



Fonte: elaborado pelos autores

Com todas as informações coletadas, identificamos os seguintes Planos de Ação, conforme Figura 74.

Figura 74 - Plano de Ação Medição



Fonte: elaborado pelos autores

O processo completo que foi realizado para identificar as Dores e Necessidades relacionadas à Medição, apresentadas acima, encontra-se no APÊNDICE E - Mapeamento das Dores e Necessidades.

3.2.5 Compartilhamento (Sharing)

O conhecimento, quando compartilhado, consegue auxiliar os times a trabalharem de forma mais organizada e integrada, evitando que os processos sejam dependentes dos demais departamentos e, principalmente, de determinados colaboradores.

Alguns dos pilares desta etapa são:

- Cultura do compartilhamento;
- Adesão de maior colaboração entre membros;
- Cultura da transparência;
- Feedback contínuo para os times;

- Compartilhamento das descobertas, Lições Aprendidas e Ferramentas, para criar um ambiente de aprendizagem organizacional.

Classificamos os feedbacks que entendemos estar relacionados ao Compartilhamento (Sharing) e fomos analisando para chegar a dor ou necessidade. Em alguns casos, tivemos feedbacks relacionados a uma ou mais dores ou necessidades. A Figura 75 apresenta um exemplo de como foi realizado este processo, até chegarmos ao Plano de Ação.

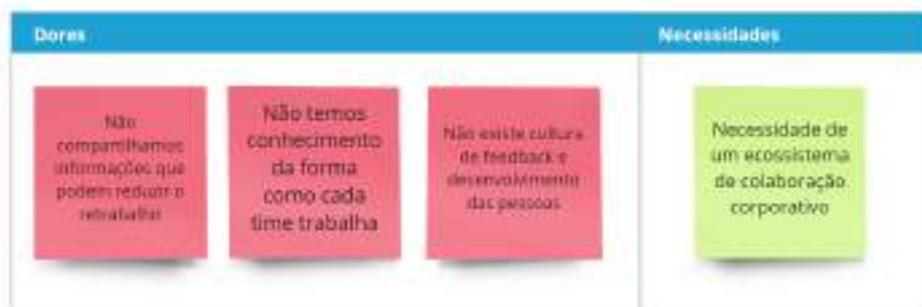
Figura 75 - Mapeamento dos feedbacks relacionados à Compartilhamento (Sharing)



Fonte: elaborado pelos autores

Com este mapeamento, conseguimos identificar as seguintes Dores e Necessidades, conforme Figura 76.

Figura 76 - Dores e Necessidades identificadas relacionadas à Compartilhamento (Sharing)



Fonte: elaborado pelos autores

Com o resultado do assessment, tivemos a visão de onde o time deseja chegar, conforme Figura 77.

Figura 77 - Resultado do assessment - Compartilhamento (Sharing)



Fonte: elaborado pelos autores

Com todas as informações coletadas, identificamos os seguintes Planos de Ação, conforme Figura 78.

Figura 78 - Plano de Ação Compartilhamento (Sharing)



Fonte: elaborado pelos autores

O processo completo que foi realizado para identificar as Dores e Necessidades relacionadas à Compartilhamento (Sharing), apresentadas acima, encontra-se no APÊNDICE E - Mapeamento das Dores e Necessidades.

3.3 Plano de Ação

Após o resultado do Assessment, visto no item 3.1 e o mapeamento das Dores e Necessidades, visto no item 3.2, foi possível identificarmos as macro-ações necessárias para iniciarmos a adoção da cultura Devops na empresa e planejar o Projeto Piloto. Estas análises foram a base para gerarmos um backlog com as ações para serem priorizadas.

A priorização foi realizada utilizando a técnica baseada em consenso, conhecida como Planning Poker, conforme orientação do nosso padrinho Carlos Henrique. Algumas adaptações ao nosso contexto foram realizadas, como a troca da sequência Fibonacci por notas de 1 a 5, onde a nota máxima definiria se a macro ação seria detalhada para o plano de ação do projeto Piloto.

Para realizarmos o Planning Poker, utilizamos a ferramenta PlanITPoker²⁷, conforme Figura 79. Cada macro-ação foi apresentada para os integrantes do grupo, e cada um votou individualmente considerando a sua visão da importância daquele item para o piloto. Em caso de notas discrepantes, realizamos uma pequena discussão para entender a visão de todos e chegar a um consenso. Quando não conseguimos chegar a um acordo, pontuamos de acordo com a média das notas. O resumo do Planning Poker, pode ser consultado no APÊNDICE H - Planning Poker.

Figura 79 - Planning Poker



Fonte: elaborado pelos autores

²⁷ <https://www.planitpoker.com/>

Separamos os itens com nota máxima para o detalhamento, identificando as ações, pessoas envolvidas, objetivos e mês/ano inicial estimado. O Plano de Ação completo será apresentado no item 3.3.1 Projeto Piloto. Algumas ações que foram identificadas para o piloto, e que tiveram nota abaixo de 4, ficarão no backlog para uma próxima etapa.

Ao final do piloto, realizaremos um PDCA²⁸, para avaliar os resultados e realizar o planejamento do próximo ciclo com base no backlog de ações evolutivas. A priorização será feita de acordo com o que for considerado mais importante no momento, em termos de valor para a equipe responsável pelo projeto.

A cultura é um dos principais pontos a serem desenvolvidos para que a empresa conquiste resultados mais efetivos na adoção de DevOps, por isso, ações educacionais e culturais, listadas no item 3.3.2 Ações para Empresa, poderão ocorrer em paralelo ao projeto piloto e servirão para difundir os conceitos da cultura DevOps, colaborando para melhoria do relacionamento entre as equipes. Estas ações não precisam de uma priorização, porque elas irão beneficiar a todas as equipes, não apenas a equipe do projeto piloto.

A Figura 80, mostra como funcionará o Projeto Piloto e suas evoluções. O objetivo é evoluir até a adoção da cultura e práticas DevOps, usando como base os conceitos de agilidade, focando em entregas de valor e adaptando quando necessário.

O piloto servirá de baseline para as demais equipes, que se beneficiarão das lições aprendidas, dos desafios que enfrentamos e os caminhos que seguimos.

²⁸ 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
Fonte: <https://ferramentasdaqualidade.org/pdca/>

Figura 80 - Plano de Ação Projeto Piloto



Fonte: elaborado pelos autores

3.3.1 Projeto Piloto

Para garantir o melhor resultado do projeto piloto, ele deverá ser executado em um sistema não crítico, de pequeno porte e com prazo de implantação de até 90 dias. Dessa forma teremos melhor controle do ambiente, maior agilidade, segurança e facilidade para acompanhar as medições. Os primeiros passos devem contribuir para a interação entre os profissionais, utilização das novas ferramentas e revisão das boas práticas recomendadas.

O objetivo do projeto piloto é mensurar os benefícios com a adoção das práticas DevOps e capacitar uma equipe que servirá como disseminadora do conhecimento e experiência adquirida dentro da empresa.

A equipe do projeto piloto deverá ser multidisciplinar e recomendamos que todos os membros estejam completamente dedicados ao projeto, sem outras alocações.

A seguir, apresentaremos o Plano de Ação do Projeto Piloto com o detalhamento das macro-ações identificadas com nota 5 e o backlog de ações evolutivas.

Tabela 1 - Plano de Ação do Projeto Piloto

Plano de Ação do Projeto Piloto			
Atividades	Objetivo	Envolvidos	Mês/Ano previsto

1. Planejamento do Piloto			
1.1 Definir Projeto Piloto	Identificar no portfólio projeto que será utilizado para o piloto bem como definir o Patrocinador	Diretoria de Sistemas e Diretoria de Negócio	Dez/2020
1.2 Definir Equipe e papéis & responsabilidades	Montar equipe e definir os papéis e responsabilidades dos membros	Diretoria Técnicas	Dez/2020
1.3 Definir Macro-cronograma	Definir marcos de entrega	Líder do Projeto	Jan/2021
1.4 Definir métrica e indicadores	Identificar as métricas que irão compor o projeto piloto	Equipe do projeto piloto	Jan/2021
1.4.1 Definição das ferramentas	Selecionar as ferramentas que serão utilizadas para coleta dos indicadores selecionado	Equipe do projeto piloto	Jan/2021
1.4.2 Treinamento dos responsáveis pela coleta/avaliação dos indicadores	Treinar os responsáveis nas ferramentas e na avaliação dos indicadores.	Equipe do projeto piloto	Jan/2021
2. Propor ações educacionais que têm por finalidade, disseminar, manter e evoluir uma cultura DevOps.			
2.1 Programa de Disseminação da Cultura DevOps	Nivelar conhecimento sobre DevOps Público-alvo: todos os envolvidos no projeto piloto	Equipe do projeto piloto	Fev/2021

2.2 Fundamentos em Práticas de Integração Contínua	Nivelar conhecimentos sobre Integração Contínua Público-alvo: todos os envolvidos no projeto piloto	Equipe do projeto piloto	Fev/2021
2.3 Fundamentos e Práticas em Infraestrutura como Código	Nivelar conhecimentos em Infraestrutura como Código Público-alvo: todos os envolvidos no projeto piloto	Equipe do projeto piloto	Fev/2021
2.4 Fundamentos e Práticas em Entrega Contínua	Nivelar conhecimentos sobre Entrega Contínua Público-alvo: todos os envolvidos no projeto piloto	Equipe do projeto piloto	Fev/2021
2.5 Fundamentos e Práticas em Docker, Kubernetes e Orquestração de Containers	Nivelar conhecimentos sobre Docker, Kubernetes e Orquestração de Containers. Público-alvo: todos os envolvidos no projeto piloto	Equipe do projeto piloto	Fev/2021
3. Revisar Processo de Desenvolvimento Java, Papéis e Responsabilidades			
3.1 Revisão do Processo de Desenvolvimento Java	Revisar o processo atual de desenvolvimento e propor melhorias com foco nas práticas DevOps	Equipe do projeto piloto	Fev/2021
3.2 Definição de Papéis e Responsabilidades	Definir papéis e responsabilidades com foco nas práticas devops	Equipe do projeto piloto	Fev/2021
4. Propor ações educacionais que têm por finalidade, disseminar a importância dos Testes Automatizados.			

4.1 Programa de Disseminação da Cultura de Testes	Disseminar a importância e os benefícios da automação dos testes	Equipe do projeto piloto	Mar/2021
5. Implantar Testes Unitários			
5.1 Definir Ferramenta de Testes Unitário	Identificar ferramenta para automação dos testes unitários em tecnologia JAVA	Equipe do projeto piloto	Mar/2021
5.2 Definir Metodologia e Padrões para codificação dos Teste	Definir boas práticas e processos para codificação dos Testes.	Equipe do projeto piloto	Mar/2021
5.3 Preparar ambiente de testes de desenvolvimento	Preparar ambiente de desenvolvimento local	Equipe do projeto piloto	Mar/2021
5.4 Integrar Testes Unitário no Pipeline	Configuração do Pipeline	Equipe do projeto piloto	Mar/2021
5.4.1 Treinamento dos Desenvolvedores em Ferramentas e linguagem Yaml	Treinar a equipe na ferramenta selecionada e na linguagem YAML para codificação do pipeline.	Equipe do projeto piloto	Mar/2021
6. Avaliação do Projeto Piloto			
6.1 Avaliação do Piloto	Avaliação da evolução dos indicadores e resultados	Equipe do projeto piloto	Abr/2021

3.3.2 Ações Evolutivas

As macro-ações listadas abaixo fazem parte do backlog de ações evolutivas que poderão ser implementadas aos projetos da tecnologia Java, após análise e priorização seguindo o PDCA.

- Implantar Avaliação de Qualidade Código
- Implantar Testes Funcionais e Aceitação
- Implantar Testes de API
- Revisar Testes de Performance
- Revisar Testes de Vulnerabilidade
- Implantar Práticas de automatização do provisionamento dos ambientes internos
- Implantar cultura de melhoria contínua/medição/monitoramento
- Implantar ferramentas e técnicas para avaliação da Experiência do Usuário
- Implantar Ferramentas de Telemetria
- Implantar ferramentas para Gerenciamento de Log

3.3.3 Ações para Empresa

As macro-ações listadas abaixo, visam inserir novas equipes da empresa nas atividades para a disseminação da cultura DevOps.

- Propor ações educacionais que têm por finalidade, disseminar, manter e evoluir uma cultura DevOps
- 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
- Reforçar Metodologia de Desenvolvimento Ágil, papéis e responsabilidades e ferramentas
- Encontros periódicos de melhoria contínua para discussões de ferramentas e processos do time
- Reforçar cultura da Liderança com Respeito (Lean/MGM3.0)

- Definir e implantar um processo de melhoria contínua
- Definir e implantar um ecossistema de Colaboração Corporativo
- Definir processos de Gestão de Pessoas
- Propor a contratação de Consultoria para orientar os demais times sobre práticas DevOps
- Definir e contratar Treinamento especializado nas práticas DevOps para diversos times
- Definir, adquirir Ferramentas e Consultoria de práticas do DevOps conforme a tecnologia envolvida

4. CONSIDERAÇÕES FINAIS

4.1 CONCLUSÃO

Este trabalho apresentou um estudo sobre a adoção da cultura e das práticas DevOps na Iplanrio com o objetivo de montar um plano de ação para execução de um projeto piloto que permitisse experimentarmos os conceitos e as práticas DevOps e validar os seus benefícios.

A Iplanrio é uma empresa pública que tem como missão prover soluções tecnológicas para melhorar a vida na cidade do Rio de Janeiro, fornecendo serviços com alta qualidade e disponibilidade. Sendo assim, procuramos entender os principais problemas enfrentados pela empresa no seu dia a dia.

Descobrimos alguns problemas típicos de uma empresa de tecnologia, como por exemplo: falta de comunicação entre áreas; implantações ruins e falhas em produção; aplicações entregues com problemas; profissionais sobrecarregados.

Cada vez mais os serviços passam a depender de soluções tecnológicas inovadoras e esta demanda tem se acelerado exponencialmente. Em um mundo de imprevisibilidade, a empresa precisa se preparar para as constantes mudanças de cenários, e nesse mundo tão exigente, DevOps se encaixa perfeitamente.

Então, como disseminar a cultura DevOps em uma empresa pública? Os desafios são inúmeros. Porém, antes de começarmos a implantar ferramentas, técnicas de automação e trabalhar a colaboração, procuramos compreender o real significado do conceito DevOps. Para isso fizemos diversas pesquisas literárias, entrevistamos com outras empresas e contamos com o apoio de uma consultora que nos auxiliou no início da jornada.

O trabalho foi realizado de forma colaborativa, para que as pessoas envolvidas se sentissem parte do todo e que os nossos resultados fossem alinhados às suas expectativas. Por isso, iniciamos os encontros do Café com DevOps, onde reunimos diversos perfis de profissionais que estariam envolvidos de forma direta ou indireta na tecnologia java, para ouvir suas dores, necessidades e opiniões e o que entendiam sobre DevOps.

O foco dos encontros era a disseminação da cultura DevOps e no piloto que executaremos na tecnologia Java, porém as pessoas aproveitaram o momento e compartilharam o seu dia a dia, as suas percepções nas diferenças dos processos de

desenvolvimento e entregas. Tudo que foi compartilhado foi coletado para que possa servir para ações futuras.

Com os encontros, concluímos que ainda existem dúvidas do significado de DevOps, mas todos gostariam que a empresa divulgasse mais informações sobre o assunto a fim de promover uma cultura homogênea, colaborativa e coesa entre todas as áreas. Percebemos também que muitos não têm um conhecimento da estrutura da empresa, suas gerências, equipes, papéis e responsabilidades. Muitos colegas não sabem o que a equipe ao seu lado desenvolve.

Com o assessment, identificamos o nível de maturidade nas práticas DevOps e verificamos que conseguimos avançar em Infraestrutura como Código e que possuímos um nível baixo de maturidade em testes automatizados. Com isto concluímos que estamos no caminho da evolução e avançamos em práticas e técnicas atuais que muitas empresas no mercado ainda não adotaram. Precisamos avançar em automação de testes para melhorar a qualidade das entregas.

Com as entrevistas realizadas com os especialistas, concluímos que temos pessoas engajadas e que estão adotando algumas práticas DevOps de forma tímida, com esforços individuais ou em times isolados. Mesmo sem perceber, praticam princípios DevOps, como a colaboração e automação de alguns processos.

Essa visão foi importante pois entendemos os benefícios que teremos com o direcionamento da Diretoria, em executar o piloto na tecnologia java, uma vez que esta equipe está mais evoluída em algumas ferramentas e práticas ágeis.

Concluímos que as equipes desejam participar de forma mais ativa nos projetos, porém se encontram em diretorias distintas, com sobreposição de prioridades e em alguns casos trabalhando com tecnologias obsoletas. A diversidade de tecnologias existentes atualmente ficou evidente e é importante que as equipes compartilhem suas inovações para incentivar os demais.

Após a coleta dos feedbacks dos participantes foi possível mapear as dores e necessidades dos envolvidos e com o apoio do gráfico de Radar de Maturidade, geramos um Plano de Ação do Projeto Piloto, onde poderemos experimentar e avaliar a adoção de novas práticas.

A partir de todo estudo realizado durante este trabalho, concluímos que o Projeto Piloto deverá começar com os testes unitários porque eles são o início para o

caminho da automação dos testes, e seu custo de implantação é baixo, uma vez que serão utilizadas ferramentas open source.

Com esta estratégia começaremos a disseminar uma cultura focada nos testes dentro da equipe, além de nos beneficiar com as seguintes melhorias: redução dos erros; incentivo à refatoração; documentação técnica viva, executável e sempre atualizada.

O projeto piloto possui um backlog de ações evolutivas que serão implementadas conforme seu andamento e que contempla por exemplo: evolução na automação dos testes, Telemetria, Gerenciamento de Log, Automatização do Provisionamento dos ambientes internos e a disseminação da cultura da Melhoria Contínua.

Concluimos que é necessário que a empresa entenda a importância da adoção da Cultura DevOps e que ela não pode ser exclusiva de uma tecnologia ou time. Por isso identificamos macro-ações para empresa, a fim de envolver outras equipes no processo e fomentar os princípios de colaboração, compartilhamento e práticas de automação. As novas equipes poderão tomar como base os planos de ação definidos no projeto piloto, com as adaptações necessárias.

No início desta jornada, tínhamos dúvidas do que significa DevOps e hoje conseguimos compreender que DevOps não é só cultura, não é só prática e ferramentas. DevOps é um “estado de espírito”, que vai muito além da abordagem técnica, pois envolve toda a empresa. Todos fazem parte da mesma equipe, não apenas desenvolvedores e operações, e todos trabalham em direção a um objetivo comum: o sucesso de todo o projeto.

Esperamos que este trabalho sirva de apoio para adoção da cultura DevOps na empresa e fortalecimento da colaboração, compartilhamento e integração entre os times, visando agilizar as entregas com automação e alto nível de qualidade.

4.2 RECOMENDAÇÕES

A adoção de práticas DevOps pode trazer uma série de benefícios para empresa com impactos de agilidade e inovação nos negócios. É necessário entender que DevOps não é um projeto e sim uma jornada e os líderes devem estar preparados, pois é um desafio de longo prazo.

Apresentamos a seguir, uma série de recomendações que a empresa pode adotar, para apoiar essa jornada:

- Revisão da estrutura organizacional para reduzir os silos, melhorar a colaboração nos times e torná-los multifuncionais, trazendo qualidade, segurança e agilidade para as aplicações;
- Reforçar os conceitos de Melhoria Contínua e adotar o pensamento Lean, encarando os problemas de maneira aberta e entendendo que eles geram uma oportunidade de aprendizado;
- Promover a integração das equipes através de eventos internos, fóruns e workshops para a melhoria do processo de desenvolvimento;
- Criar ações de endomarketing para disseminar os resultados e produtos que são realizados pelas equipes, gerando um clima organizacional saudável;
- Incentivar a transparência, visibilidade, feedback e compartilhamento das ideias para que os times possam participar, trabalhando de forma integrada em busca de soluções inovadoras, rápidas e simples para os problemas;
- Nivelar o conhecimento de práticas DevOps entre as demais equipes, como forma de facilitar a transição para o desenvolvimento de aplicações baseadas em nuvem, tais como: integração contínua e entrega contínua; infraestrutura como código; gerenciamento de configuração; monitoramento contínuo, entre outros.

5. REFERÊNCIA BIBLIOGRÁFICA

AMAZON. **Aws**. 2020? Disponível em: <https://aws.amazon.com/pt/devops/what-is-devops/?nc1=f_cc>. Acesso em: 20 out. 2020.

ANTONIO, Muniz; SANTOS, Rodrigo; IRIGOYEN, Analia. Jornada DevOps: Unindo Cultura Ágil, Lean E Tecnologia Para Entrega De Software Com Qualidade. 2. ed. Brasil: Brasport, 2019. 320 p. v. 2. ISBN 978-8574529271.

DEVOPS.COM. **ANDREW DAVIS**. 2020. Disponível em: <<https://devops.com/measuring-devops-performance/>>. Acesso em: 15 out. 2020.

DIEGO CARDOSO. Cio.com. 2019. Disponível em: <<https://cio.com.br/gestao/como-medir-o-sucesso-das-iniciativas-de-devops/>>. Acesso em: 2 nov. 2019.

DUVALL, Paul; MATIAS, Stephen; GLOVER, Andrew. Continuous Integration: improving software quality and reducing risk. New Jersey: Addison-Wesley, 2007.

EVERTON MARTINS. **Mettzer**. 2019. Disponível em: <<https://blog.mettzer.com/coleta-de-dados/#Metodologias-para-a-coleta-de-dados>>. Acesso em: 15 out. 2020.

FARROHA, Bassam; FARROHA, Debora. A Framework for Managing Mission Needs, Compliance and Trust in the DevOps Environment. IEEE Military Communications Conference, p. 288–293, out. 2014. Disponível em: <<http://ieeexplore.ieee.org/document/6956773/references>>. Acesso em: 5 nov. 2017.

FERNANDES, A.A.; ABREU, V.F; DINIZ, J.S. **Governança Digital 4.0**. Rio de Janeiro: BRASPORT, 2019. p. 409.

HUMBLE, Jez; FARLEY, David. Continuous Delivery: reliable software releases through build, test and deployment automation. New Jersey: Addison-Wesley, 2010.

LEVITA, Carlos de Amorim **Proposta de modelo para avaliação da maturidade DevOps**:: Estudo de caso em empresas de grande porte. 2017. Dissertação (ESPECIALIZAÇÃO EM ...) - Pontifícia Universidade Católica de São Paulo, São Paulo.

KIM, Gene et al. **Como Obter Agilidade, Confiabilidade e Segurança em Organizações Tecnológicas**. [S.l.]: Alta Books, 2020.

JOHN ALLSPAWE. **Slideshare**. 2009. Disponível em: <<https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>>. Acesso em: 15 out. 2020.

JOSHUA PARTOGI. **Scrum.org**. 2018. Disponível em: <https://www.scrum.org/resources/blog/scrum-and-devops?utm_source=newsletter&utm_medium=email>. Acesso em: 13 out. 2020.

LEONARDO MATSUMOTA. **Digita Strategy and IT Innovation**. 2019. Disponível em: <<https://leonardo-matsumota.com/2019/02/19/adotando-o-devops-e-calms/>>. Acesso em: 15 out. 2020.

MARCELO COSTA. **Promove Soluções**. 2018. Disponível em: <<https://promovesolucoes.com/como-aplicar-uma-cultura-devops-na-empresa-entenda/>>. Acesso em: 15 out. 2020.

MANISH VIRMANI. **IEEE Explorer**. 2015. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/7173368/>>. Acesso em: 15 out. 2020.

OLIVEIRA, BEATRIZ. “Infrastructure as Code.” *Medium*, 01 08 2020, <https://medium.com/sysadminas/infrastructure-as-code-42537d03e021>. Acesso em: 31 10 2020.

SHARMA, Sanjeev; COYNE, Bernie. *DevOps for dummies*. 2. ed. New Jersey: John Wiley & Sons, 2015.

TOMER LEVY. **Devops.com**. 2020? Disponível em: <<https://devops.com/9-open-source-devops-tools-love/>>. Acesso em: 15 out. 2020.

VINEET CHATURVEDI. **Edureka.co**. 2020. Disponível em: <<https://www.edureka.co/blog/devops-tutorial>>. Acesso em: 15 out. 2020.

WIKIPEDIA. 2020. Disponível em: <<https://pt.wikipedia.org/wiki/DevOps#Hist%C3%B3ria>>. Acesso em: 15 out. 2020.