

Um Estudo Exploratório sobre Práticas de Integração, Entrega e Implantação Contínua no Contexto das Empresas do Recife

Resumo

As práticas de integração [CI], entrega [CDE] e implantação [CD] já estão presentes no cotidiano de grandes empresas de todo o mundo. E isto é devido, entre outras coisas, aos comprovados benefícios que a utilização destas técnicas trazem para a equipe e para o produto em desenvolvimento. Mesmo assim, poucos são os estudos que investigam o estado da arte destas práticas nas empresas, principalmente no contexto da cidade de Recife. Com isso, o presente trabalho objetiva entender como as técnicas de integração, entrega e implantação contínuas foram importadas para as empresas recifenses, assim como identificar princípios e práticas adjacentes que governam a adoção destas técnicas. Para tanto, foi realizada uma pesquisa qualitativa por meio de entrevistas com 11 desenvolvedores de software de empresas de tecnologia sediadas na cidade do Recife. Foi descoberto que a amostra não segue o *Stairway to Heaven*, teoria que sugere que a adoção deve seguir uma sequência específica de práticas. Não obstante, a maioria entrevistada integra o código de uma nova funcionalidade para a *branch* principal apenas no final da *Sprint*, não diariamente, o que não é consistente com as definições mais comuns de integração contínua. Além disso, a prática de Testes A/B não foi encontrada em nenhum dos times entrevistados, em razão da baixa quantidade de usuários ativa ou por não se aplicar ao contexto da aplicação.

Keywords

Integração Contínua, Implantação Contínua, Desenvolvimento Colaborativo, Engenharia de Software

ACM Reference Format:

. 2021. Um Estudo Exploratório sobre Práticas de Integração, Entrega e Implantação Contínua no Contexto das Empresas do Recife. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 Introdução

As práticas de integração, entrega e implantação contínuas [7, 8] – originalmente conhecidas pelos termos em inglês *Continuous Integration* [CI], *Continuous Delivery* e *Continuous Deployment* [CD] ¹, respectivamente – já são muito difundidas e utilizadas por empresas de tecnologia em todo o mundo. Segundo o 14º relatório anual do

¹Neste trabalho, assim como no estudo original, a sigla CD se refere às práticas de entrega e implantação contínuas

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

estado das metodologias ágeis [3], 55% dos participantes reportaram que sua organização utiliza a técnica de integração contínua. Também nesta mesma pesquisa foi encontrado que 41% utilizam a técnica de entrega contínua e 36%, implantação.

Estes números elevados são causados principalmente pelos benefícios que a utilização destas técnicas trazem para a equipe e para o produto em desenvolvimento. Estudos comprovam que os desenvolvedores envolvidos se sentem mais produtivos quando usam as práticas de CI/CD [10] e o seu uso traz maior qualidade ao software que está sendo produzido [18].

É possível, no entanto, perceber uma grande falta de estudos a respeito de como essas práticas foram importadas para as empresas ao redor do mundo [17], incluindo Recife – as pesquisas são geralmente voltadas apenas para as corporações globais como Facebook [18] e Google [15]. A capital pernambucana é um dos grandes pólos tecnológicos do Brasil: somente no Porto Digital, localizado na cidade, há cerca de 330 empresas e 11 mil trabalhadores, com faturamento anual de R\$ 2,3 bilhões em 2019 [2].

Um estudo a respeito de como as técnicas de CI/CD migraram para a indústria de Recife serve como ponto de partida para pesquisas relacionadas ao levantamento de dores sentidas pelos desenvolvedores locais que inibem a utilização destas técnicas.

Neste contexto, este trabalho tem como objetivo entender, através de uma pesquisa qualitativa, quais as práticas e técnicas subjacentes de integração, entrega e implantação contínua adentraram nas empresas de Recife. Não obstante, o trabalho deseja descobrir que princípios e práticas subjacentes governam a adoção destas técnicas.

2 A Adoção de Integração, Entrega e Implantação Contínuas nas Empresas

As práticas de integração e implantação contínuas (*Continuous Integration* e *Continuous Deployment*) já são muito difundidas e utilizadas por empresas de tecnologia em todo o mundo. Segundo o 14º relatório anual do estado das metodologias ágeis [3], 55% dos participantes reportaram que sua organização pratica a técnica de integração contínua. Também nesta mesma pesquisa foi encontrado que 36% utilizam a técnica de integração contínua.

Estes números elevados são causados principalmente pelos benefícios que a utilização destas técnicas trazem para a equipe e para o produto em desenvolvimento. Ainda de acordo com [3], entre as razões para adoção de CI/CD, as principais são aceleração de entregas de software (71%), e aumentar a produtividade (51%) e a qualidade do software (42%).

Especificamente sobre integração contínua, a prática hoje em dia já é bastante estudada e difundida na indústria. O estudo [10] comenta que desenvolvedores envolvidos se sentem mais produtivos quando utilizam a prática e dão mais valor aos testes automáticos. Já com relação a implantação contínua, o estudo [18] mostra que seu uso traz maior qualidade ao software que está sendo produzido.

Apesar da popularização de CI e CD no desenvolvimento colaborativo de software, ainda há uma grande escassez de estudos que demonstrem como essas práticas foram importadas para a maioria das empresas [17], especialmente as não inseridas entre as empresas globais de desenvolvimento de software.

Uma das tentativas mais recentes de estudo sobre o tema, é o desenvolvido por Schermann et al. [17], que busca entender como as práticas geralmente associadas a implantação contínua acharam o seu caminho nas indústrias européia e norte-americana. Nesse estudo os autores utilizaram um método misto de estudo empírico baseado em um pré-estudo na literatura, entrevistas com 20 participantes e um *survey* que recebeu 187 respostas. A ideia era questionar até que ponto o conhecimento na área estava dominado por peculiaridades de um pequeno grupo de grandes empresas, como Facebook e Google.

Os autores também definem a chamada *stairway to heaven* (escada para o céu, em tradução livre), presente na Figura 1. Ela tem como objetivo definir um caminho de evolução das empresas para um estágio de entregas sofisticado. A escada permeia práticas de integração contínua, implantação contínua e entregas parciais.

Através desta metodologia os autores descobriram que, no contexto estudado, problemas arquiteturais são geralmente uma das maiores barreiras para a adoção de CD. Não obstante, a técnica de *Feature Toggles* [11] como forma de realizar entregas parciais adiciona uma complexidade demasiada e não saudável ao código. Por fim, eles concluem que os desenvolvedores necessitam também de um protocolo baseado em princípios que estabeleçam quando deve-se utilizar técnicas de entregas parciais, por exemplo, que funcionalidades e métricas devem ser testadas por um teste A/B [13]. Não obstante, os autores também descobriram que as empresas do estudo não seguem a escada de evolução proposta por eles.

Aprender mais sobre as dificuldades que outras empresas não globais sofrem no processo de adoção de práticas de CI/CD pode gerar mais estudos a respeito de como solucionar tais problemas, assim como mostrar possíveis oportunidades de melhoria e revisão dos processos utilizados. Um trabalho que utilize uma adaptação da metodologia aplicada a uma amostra de um outro contexto pode revelar ainda discrepâncias e semelhanças entre os dois ambientes de estudo. As discrepâncias levantariam pontos de questionamento, pesquisa e até possíveis melhorias aos ambientes envolvidos. Já as semelhanças podem assegurar que as práticas utilizadas já acharam o seu lugar na indústria e funcionam bem assim como a teoria propunha.

Nesse sentido, o presente trabalho procura entender sobre a utilização das práticas de CI/CD nas empresas de Recife para que esses dados sirvam como objeto de pesquisa para levantamento de possíveis dores sentidas pelos desenvolvedores locais que justifiquem a adoção ou não destas práticas. Recife é um dos grandes polos tecnológicos do Brasil: somente no Porto Digital, localizado na capital pernambucana, há cerca de 330 empresas e 11 mil trabalhadores, com faturamento anual de R\$ 2,3 bilhões em 2019 [2]. Um estudo a respeito de como as técnicas de integração e implantação contínua migraram para a indústria de Recife funciona ainda como um *benchmark* de como as empresas estão se portando em relação às novidades presentes na literatura nos últimos anos, assim como levantar possíveis discrepâncias entre empresas situadas na área e as grandes corporações.

2.1 Perguntas de pesquisa

Com o intuito de entender como as técnicas de integração, entrega e implantação contínua foram importadas para as empresas recifenses, as seguintes perguntas de pesquisa foram formadas:

PP1: *Quais as práticas de CI/CD são utilizadas pelas empresas em Recife?*

PP2: *O cenário de CI/CD nas empresas em Recife segue o stairway to heaven proposto no estudo original?*

PP3: *Quais são os princípios e práticas subjacentes que governam a adoção de CI/CD na indústria?*

Para responder às perguntas acima foi decidido replicar a pesquisa qualitativa do estudo [17] com desenvolvedores recifenses, baseando-se também na mesma lista de definições de cada uma das práticas envolvidas no *Stairway to Heaven*. A pesquisa qualitativa neste contexto foi considerada fundamental para garantir que os entrevistados entendessem claramente as perguntas e excluir possíveis entendimentos errados de termos em inglês, linguagem não nativa de todos os participantes. A pesquisa quantitativa realizada pelo estudo original também não foi replicada em função da delimitação de escopo deste trabalho, cujo objetivo principal é obter um panorama inicial da adoção de CI/CD nas empresas de Recife, de modo a permitir a o levantamento de indicadores sobre o tema no contexto investigado, que permitam a evolução da pesquisa iniciada com esse estudo.

Além disso, as perguntas 1 e 3 são uma adaptação das perguntas 1 e 2 do estudo original [17], respectivamente, incluindo a técnica de integração contínua. Não obstante, uma terceira pergunta foi adicionada (PP2), que foca na escada de evolução proposta pelo artigo. Ela tenta responder se a *stairway to heaven* é seguida no contexto de Recife, visto que, no estudo original, os próprios autores já refutaram esta definição com os seus resultados.

3 Metodologia

Para responder as perguntas de pesquisa apresentadas na seção anterior, foi realizada uma pesquisa qualitativa através de entrevistas semi-estruturadas utilizando como ponto de partida o roteiro desenvolvido pelo estudo original [17]. As entrevistas foram feitas com desenvolvedores de empresas sediadas em Recife.

3.1 Pré-estudo

Com o objetivo de entender melhor como as práticas de CI/CDE/CD migraram para as empresas sediadas em Recife, essa fase do estudo contou com a participação de um segundo pesquisador para discussão do estudo original [17]. Em função do objetivo principal desse estudo, para fins de delimitação de escopo, foi definido que seria aplicado apenas o estudo baseado em entrevistas semi-estruturadas.

Como forma de manter a consistência, foi utilizado o mesmo guia de entrevistas, assim como as mesmas definições utilizadas pelos autores. No início foi montado um arquivo com a definição de cada um dos termos envolvidos em língua portuguesa para ser utilizado como consulta caso houvesse alguma dúvida de definição de tópicos entre os entrevistadores.

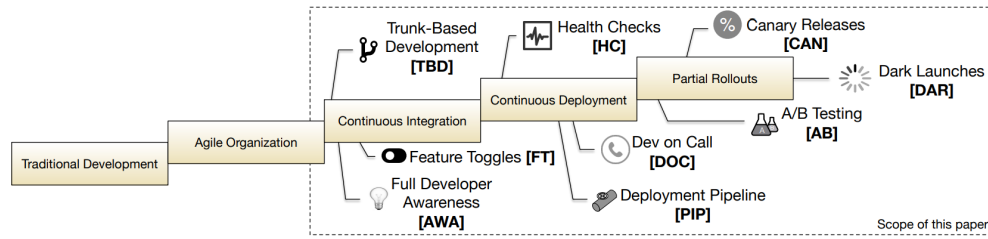


Figura 1: A escada de evolução denominada *Stairway to Heaven* [17]

Após isso, o guia de entrevistas do estudo original também foi traduzido para português e utilizado durante as entrevistas ². Um ponto importante a respeito da tradução é o fato de que alguns termos ainda foram mantidos na língua inglesa devido ao fato de serem conhecidos mundialmente nesta língua. Por exemplo, *Continuous Integration*, *Canary Releases* e *Health check*.

3.2 Estrutura da Entrevista

O guia de entrevistas se baseia na estratégia de evitar perguntas diretas (exemplo: “determinada prática está sendo utilizada?”). Esse modelo é essencial para garantir que a comparação entre participantes a respeito do uso ou não de práticas está sendo feita de maneira concisa, e não baseada nos conhecimentos prévios de cada participante. Assim, através de perguntas sobre o processo utilizado pelo entrevistado é possível, com uma certa margem de erro associada, afirmar que práticas ele utiliza. A margem de erro surge do fato de o autor ter que refletir sobre as informações recebidas e as definições para inferir o uso ou não de certa prática.

A entrevista é dividida em 5 sessões:

- (1) Processo de entrega no geral
- (2) Papéis/Responsabilidades
- (3) Garantia da Qualidade (Quality Assurance)
- (4) Gerenciamento de Problemas
- (5) Avaliação de Entrega

Apesar de o título das sessões não mencionarem aspectos de integração e implantação contínuas, as perguntas nela contidas permitiram tratar do tema com os entrevistados, por perguntar sobre características gerais do processo utilizado pela equipe.

3.3 Participantes

No total foram entrevistados 11 desenvolvedores (P1 a P11) de 7 empresas diferentes sediadas em Recife. A Figura 2 mostra todos os dados demográficos obtidos de cada um dos entrevistados. Esses dados foram obtidos pelos entrevistadores no começo de cada entrevista e agregados nesta figura. O tamanho da empresa é definido pela quantidade de colaboradores: *Startup* define empresas com menos de 50 trabalhadores; *SME* são empresas que contêm entre 50 e 500 colaboradores; e *CORP* são empresas com mais de 500 colaboradores.

Da amostra, 3 eram mulheres. Pode-se ver a distribuição dos participantes por gênero na Figura 3. Dentro desse grupo, 9 trabalhavam com aplicações Web, enquanto 1 trabalhava com sistemas embarcados e o último, com jogos.

ID	Empresa	Tipo de Aplicação	Tamanho da Empresa	Experiência (Anos)	Sexo
P1	E1	Web	SME	2,5	M
P2	E2	Web	CORP	22	M
P3	E2	Web	CORP	2	F
P4	E7	Hardware	CORP	3	F
P5	E5	Games	STARTUP	2	M
P6	E6	Web	STARTUP	1	M
P7	E3	Web	STARTUP	0,58	M
P8	E1	Web	SME	2	M
P9	E3	Web	STARTUP	1	M
P10	E4	Web	SME	1	F
P11	E4	Web	SME	8	M

Figura 2: Detalhamento dos entrevistados da amostra, contendo todos os dados demográficos coletados.

A escolha dos entrevistados foi feita baseado na rede de conhecidos dos entrevistadores, com o propósito de agregar pessoas que trabalhavam em empresas de tamanhos distintos para garantir uma variedade de parâmetros envolvidos. Como é possível perceber também na Figura 3, com relação a esse aspecto a amostra está bem distribuída.

3.4 Coleta dos Dados

No guia, todas as sessões iniciam com uma questão aberta. Por exemplo, a pergunta aberta da sessão intitulada Processo de entrega no geral traz como pergunta aberta:

“Dado uma funcionalidade recém-implementada, como é o processo de entrega da sua empresa a partir do

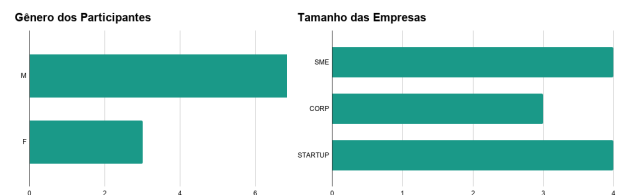


Figura 3: Distribuição dos Participantes por gênero e tamanho da empresa

²O guia de entrevistas traduzido está disponível em <http://bit.ly/ArtigoQuestoesEntrevista>

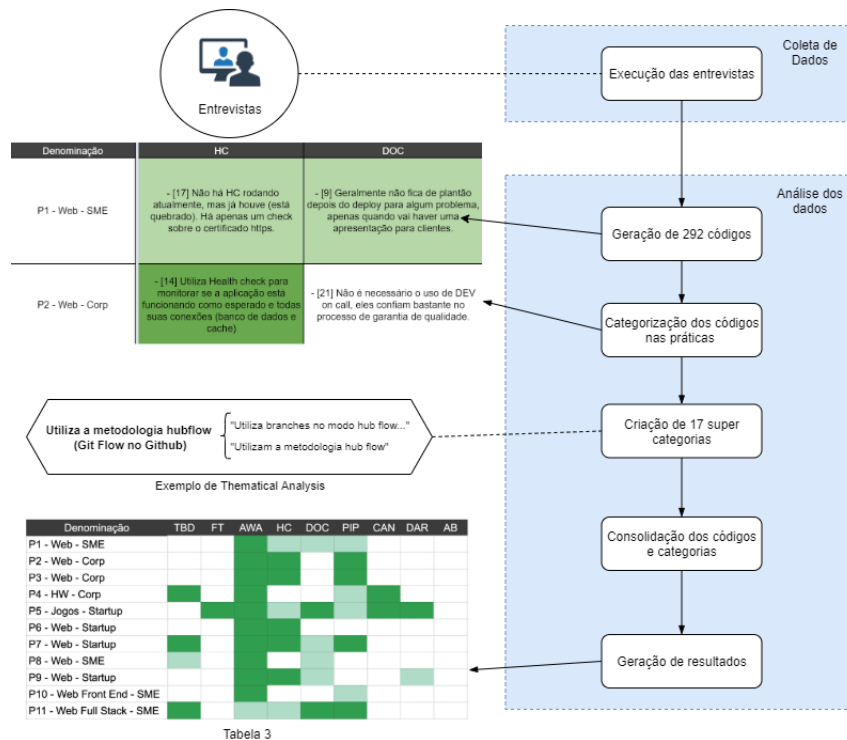


Figura 4: Visão geral dos processos de coleta e análise de dados.

commit da funcionalidade até chegar ao ambiente de produção e, portanto, aos clientes?”.

As entrevistas seguiram os tópicos abordados em cada uma delas, mas sem ordem específica, respeitando o fluxo natural da conversa. A primeira entrevista foi guiada pelos dois pesquisadores para assegurar que as seguintes seriam conduzidas de forma semelhante pelos dois pesquisadores. Esta entrevista foi considerada como válida na análise de dados visto que não houve nenhuma mudança no questionário de entrevistas: ele já tinha sido validado no artigo utilizado de base para este trabalho. As outras 10 – totalizando 11 entrevistas feitas – foram guiadas por apenas um entrevistador, sendo 5 feitas por cada um dos dois pesquisadores. Na Figura 4 o processo de entrevistas é representado pela sessão de Coleta de Dados.

Todas as entrevistas ocorreram de forma online, através do Google Meet, e aconteceram entre os meses de Setembro e Outubro de 2020 em português brasileiro. As entrevistas levaram entre 30 e 65 minutos, duração bem parecida com os tempos obtidos no estudo original (35 a 60 minutos), totalizando 7 horas e 15 minutos. Cada uma delas foi gravada pela plataforma para futura análise e 4 delas foram transcritas para o uso em citações neste trabalho, escolhidas através da relevância da entrevista e da forma como certos termos e processos foram apresentados pelo entrevistado. Foi deixado claro em cada uma das conversas a respeito da gravação e que estas seriam utilizadas apenas pelos entrevistadores, respeitando assim o anonimato do participante e confidencialidade de demais

informações pessoais, bem como da empresa onde o participante trabalha.

3.5 Análise dos Dados

A Figura 4 apresenta também uma visão geral de como funcionou o processo de análise das entrevistas. A análise das entrevistas foi conduzida apenas pelo autor principal deste estudo. Os demais pesquisadores auxiliaram no processo de discussão dos resultados para a geração de consenso e consolidação das informações. O processo escolhido foi baseado nas fases de *Coding* e *Thematic Analysis* da metodologia *Grounded Theory* [9]. No processo de *Coding* a ideia é levantar rótulos ou tags relevantes para o texto e, tradicionalmente, é feito baseado na transcrição das entrevistas. No entanto, neste trabalho o autor gerou códigos através da escuta das entrevistas. Então, como um exemplo, a seguinte citação:

“Como somos uma equipe muito pequena, todos os desenvolvedores são meio que DEVOPS. Quando tem que tomar alguma decisão nós entramos em discussão e definimos por nós.” — P5

Gerou o código: “*Todos os desenvolvedores são devops.*” — P5_15, onde P5 identifica o número do participante do qual a entrevista gerou o código, e o 15 define a ordem de criação deste, além de servir como identificador único.

Após levantados todos os códigos de uma entrevista, estes foram revisados para garantir semântica e sintaxe adequadas. Alguns códigos nessa fase foram eliminados por redundância, enquanto outros foram desmembrados, gerando mais de um código. Depois,

eles foram agrupados, quando compatíveis, em cada uma das 9 práticas descritas pelo estudo original e foi escolhida uma nota entre 0 a 2, representando não utiliza, utiliza parcialmente e utiliza completamente, seguindo a mesma escala utilizada no estudo original.

A Figura 4 ilustra um trecho dos agrupamentos realizados. Por exemplo, o códigos [17] do entrevistado P1 e o código [14] do entrevistado P2, foram agrupados à prática HC. Similarmente, os códigos [9] de P1 e [21] de P2 foram agrupados à prática DOC. Este processo foi então replicado para cada uma das 11 entrevistas.

Ao final do processo ainda haviam 3 entrevistas que não geraram nenhum código relacionado a prática de *Dark Launch*. Para estes casos, foi enviado um email diretamente para cada um dos entrevistados com a definição do estudo original da técnica e foi perguntado se o entrevistado utilizava ou não a mesma, o que gerou mais 3 códigos. Ao final, 292 códigos surgiram ao todo; A lista de todos os códigos está disponível em <http://bit.ly/ArtigoCodigos>.

Após a etapa de geração e revisão dos códigos, estes foram então agrupados, onde aplicável, na prática específica com a qual aquele código se relacionava. Este agrupamento gerou a Tabela Entrevistado-Pratica-Codigos³, que mostra uma visão geral de cada relação entre prática e entrevistado, contendo a nota dada e os códigos que justificam a nota. A nota é demonstrada pela cor presente em cada célula: branco significa não utiliza, verde claro mostra que o participante utiliza parcialmente, e verde mais escuro, totalmente. A tabela contém 147 dos 292 códigos gerados. A Figura 5 apresenta um trecho da Tabela Entrevistado-Pratica-Codigos para os entrevistados P1 e P2 e os códigos de cada um relacionados às práticas HC e DOC.

Denominação	HC	DOC
P1 - Web - SME	- [17] Não há HC rodando atualmente, mas já houve (está quebrado). Há apenas um check sobre o certificado https.	- [9] Geralmente não fica de plantão depois do deploy para algum problema, apenas quando vai haver uma apresentação para clientes.
P2 - Web - Corp	- [14] Utiliza Health check para monitorar se a aplicação está funcionando como esperado e todas suas conexões (banco de dados e cache)	- [21] Não é necessário o uso de DEV on call, eles confiam bastante no processo de garantia de qualidade.

Figura 5: Trecho da Tabela Entrevistado-Pratica-Codigos, com os códigos derivados das entrevistas agrupados por prática. As cores de cada célula representam a nota dada para determinada prática, de acordo com sua adoção pelo participante. 0: Branco (prática não é utilizada), 1: Verde Claro (utilizada parcialmente) e 2: Verde Escuro (utilizada totalmente).

Para identificar padrões e abstrações nos códigos agrupados na Tabela Entrevistado-Pratica-Codigos, foi feito um trabalho de agrupamento semântico, gerando por fim 17 novas super categorias⁴ através do processo de *Thematic Analysis* [9]. Esse processo é ilustrado pela etapa de “Criação de super categorias” na Figura 4. As

³A Tabela Entrevistado-Pratica-Codigos está disponível em <http://bit.ly/ArtigoTabelaEPC>

⁴A tabela com todas as super categorias geradas pode ser encontrada em <http://bit.ly/ArtigoSupCategorias>

super categorias tinham como objetivo agrupar códigos semelhantes ou de mesma semântica para facilitar na reflexão dos dados. Um exemplo de super categoria relacionada à prática de *Trunk Based Development* [TBD] pode ser encontrado na Figura 6.

É importante salientar que ainda durante o processo de codificação surgiram códigos relacionados às práticas de Revisão de Código [1] e Testes Automáticos [14]. Para estas foram relacionados 29 códigos⁵ – 17 para a prática de Revisão de Código e 12 para Testes Automáticos – dos 292 identificados, e duas novas super categorias foram geradas.

Por fim, a consolidação da análise dos códigos e categorias geradas deram origem a diferentes tabelas como a apresentada na Figura 4 para facilitar a visualização dos resultados sobre diferentes perspectivas. Por exemplo, a Tabela 3 ilustrada na Figura 4 mostra um mapa de utilização das práticas de CI/CD considerando os dados agregados na amostra utilizada nesse estudo. Este e os demais resultados desse estudo serão apresentados na próxima seção.

4 Resultados

Esta seção responde às perguntas de pesquisa apresentadas na Seção 2, de acordo com a metodologia descrita na seção anterior.

4.1 Estudo de Predomínio das Práticas

Com o intuito de responder a pergunta de pesquisa PP1 – *Quais as práticas de CI/CD são utilizadas pelas empresas em Recife?* foi montada a Tabela 1, que apresenta a utilização de cada uma das práticas investigadas para cada um dos participantes, de acordo com a escala de uso apresentada na seção anterior. Assim, o verde mais escuro significa que o participante utiliza totalmente, enquanto o verde claro significa utiliza parcialmente e, por fim, o branco denota a não utilização.

Denominação	AWA	HC	PIP	DOC	TBD	CAN	DAR	FT	AB
P1 - Web - SME									
P2 - Web - Corp									
P3 - Web - Corp									
P4 - HW - Corp									
P5 - Jogos - Startup									
P6 - Web - Startup									
P7 - Web - Startup									
P8 - Web - SME									
P9 - Web - Startup									
P10 - Web Front End - SME									
P11 - Web Full Stack - SME									

Tabela 1: Nível de utilização de cada uma das práticas, com as colunas ordenadas em ordem decrescente de uso.

Com a Tabela 1 é possível perceber que *Developer awareness* [AWA] foi a prática mais encontrada em toda a amostra, sendo totalmente utilizada pela grande maioria dos entrevistados; apenas um deles utiliza parcialmente. Logo após, pode-se encontrar as práticas de *Health Check* [HC] e *Pipeline de Implantação* [PIP] na segunda e na terceira colocações, respectivamente. No geral, também pode-se inferir que as técnicas de *Partial Rollouts* são ainda muito pouco utilizadas, com as 3 práticas do grupo entre as quatro

⁵A tabela com os códigos relacionados às descobertas adicionais está disponível em <http://bit.ly/ArtigoDescobertasAdicionais>

Super Categorias	Códigos
Utiliza a metodologia hubflow (Git Flow no Github)	P2 - [3] Utilizam a metodologia de hub flow.
	P5 - [11] Utiliza branches no modo hub flow. Apenas entra na branch master aquilo que está preparado para ir para produção.

Figura 6: Exemplo de super categoria gerada durante o processo de *Thematic Analysis*.

Participant	HC	PIP	AWA	DOC	CAN	FT	TBD	AB	DAR
P1 _{Web SME}									
P2 _{Enterpr SME}									
P3 _{Web SME}									
P4 _{Web SME}									
P5 _{Web SME}									
P6 _{Desktop SME}									
P7 _{Enterpr Corp}									
P8 _{Enterpr SME}									
P9 _{Enterpr SME}									
P10 _{Web SME}									
P11 _{Web SME}									
P12 _{Web Corp}									
P13 _{Web Startup}									
P14 _{Web Corp}									
P15 _{Web Corp}									
P16 _{Web SME}									
P17 _{Web Startup}									
P18 _{Web Startup}									
P19 _{Web SME}									
P20 _{Embedded SME}									

Figura 7: Utilização das práticas pelos participantes do estudo original [17].

últimas colocadas. Vale a pena citar que nenhum dos entrevistados utiliza, mesmo que precariamente, a técnica de Testes A/B [AB].

Quando comparamos os resultados da Tabela 1 com os resultados do estudo original (Figura 7), podemos perceber que há diferenças de posição entre práticas, mas que não há mudanças exorbitantes. Com a ajuda da Tabela 2, é possível perceber que há uma diferença de no máximo 2 posições na ordem de predomínio, ao comparar os resultados obtidos neste trabalho com os do estudo original. Por exemplo, é possível perceber que as práticas de AWA, TBD, DAR e FT tiveram mudanças de 2 posições, enquanto HC, PIP, CAN e AB tiveram apenas diferença de 1 unidade de posição. Apenas *developer on call* permaneceu no mesmo local dentro das duas amostras. Por fim, é interessante perceber que o conjunto das três primeiras práticas é o mesmo, mas em posições trocadas nos dois estudos.

4.2 Stairway to heaven

Com o objetivo de responder a pergunta de pesquisa PP2 – *O cenário de CI/CD nas empresas em Recife segue o “stairway to heaven” proposto no artigo?* – foi produzida a Tabela 3. Esta contém a visualização de utilização das práticas, com o mesmo conteúdo da Tabela 1 e utilizando a mesma escala de grau de utilização, mas com uma

Prática/Trabalho	Contexto Recifense	Artigo Base	Diferença de Posição
AWA	1º	3º	2
HC	2º	1º	1
PIP	3º	2º	1
DOC	4º	4º	0
TBD	5º	7º	2
CAN	6º	5º	1
DAR	7º	9º	2
FT	8º	6º	2
AB	9º	8º	1

Tabela 2: Diferença entre o estudo original e este trabalho a respeito da ordem de predomínio das práticas.

outra perspectiva de visualização dos resultados que consiste na apresentação com as colunas ordenadas pela escada definida pela sequência de práticas do *Stairway to Heaven* apresentada na Figura 1.

Denominação	TBD	FT	AWA	HC	DOC	PIP	CAN	DAR	AB
P1 - Web - SME									
P2 - Web - Corp									
P3 - Web - Corp									
P4 - HW - Corp									
P5 - Jogos - Startup									
P6 - Web - Startup									
P7 - Web - Startup									
P8 - Web - SME									
P9 - Web - Startup									
P10 - Web Front End - SME									
P11 - Web Full Stack - SME									

Tabela 3: Nível de utilização de cada uma das práticas, com as colunas ordenadas na ordem do *Stairway to Heaven*.

Com a Tabela 3 é possível perceber que a amostra deste estudo, assim como a amostra do estudo original, não segue a evolução proposta pelos seus autores. Isso fica claro quando percebe-se que não há, em nenhum dos entrevistados, uma relação clara entre a coloração da coluna com a sua anterior. É possível notar também que há uma lacuna nas duas primeiras práticas, seguido de uma grande utilização da terceira, reforçando que a *Stairway to Heaven* não foi identificada na amostra. O mais próximo dos participantes a atingir a escada definida é P5, mas ainda há nele a falta do pilar TBD do primeiro degrau da escada. É interessante destacar que na amostra do próprio estudo original também não foi possível identificar a escada de evolução.

4.3 Análise das Práticas

Nesta seção será abordado, para cada uma das práticas apresentadas na Tabela 3, as principais características encontradas na amostra. Esta tem como objetivo responder a pergunta PP3: *Quais são os princípios e práticas subjacentes que governam a adoção de CI/CD na indústria?*.

Para tanto, os códigos e super categorias criados durante a análise das entrevistas foram revisitados em busca de princípios que governam a adoção ou não de cada prática. Onde aplicado, ao longo desta seção, nós também destacamos como uma data prática se apresentou no estudo original, de modo a destacar as discrepâncias e congruências entre os contextos analisados por cada estudo.

4.3.1 Integração Contínua

Com a Tabela 1 é possível perceber que, apesar da grande disseminação a respeito de informações relacionadas à infraestrutura e manutenção de software – ligadas à prática de *Developer Awareness* [16] – estar em primeiro lugar, as outras práticas que compõem o conjunto de integração contínua ainda não estão disseminadas na nossa amostra.

4.3.1.1 Trunk Based Development [TBD]

Na amostra é possível perceber que a técnica de *Trunk Based Development* [TBD] [5] não é tão amplamente adotada, visto que apenas 4 entrevistados utilizam ao menos parcialmente. Destes, 2 comentam que entregam novas versões aos clientes baseado em novas funcionalidades, e não em *sprints*. Já entre os que não utilizam, 5 integram o código apenas no final da *sprint*. Deste grupo, 2 utilizam a metodologia *Git Flow* [4], que define uma maneira de manusear várias branches ao mesmo tempo de modo que os desenvolvedores deparem-se com o mínimo de conflitos possível e que software seja entregue em versões bem definidas.

4.3.1.2 Feature Toggles [FT]

No estudo foi possível perceber que, na amostra, a prática de *Feature Toggles* [FT] [11] é raramente utilizada, assim como no estudo original. Esta técnica foi encontrada apenas na equipe do participante P5, que a utiliza para esconder funcionalidades enquanto testes manuais ainda estão sendo feitos. Geralmente esta técnica é utilizada para auxiliar a integração de códigos ainda não finalizados quando a equipe utiliza técnicas como o *trunk based development*, mas este não é o caso de P5. É interessante notar que este participante também é um dos poucos que utiliza a técnica de *Canary Releases* [CAN].

Entre o grupo dos que não utilizavam, 7 só enviam código novo para produção quando a funcionalidade está concluída. Deste grupo, 2 comentam que fazem uso de conceito de épicas, onde uma história de usuário se prolonga para além de apenas uma *sprint*. É também interessante notar que P3 está em vias de utilizar esta técnica para reduzir conflitos de *merge* e *rebase* devido ao grande número de desenvolvedores em seu time, como podemos ver na citação:

“O meu time tem 23 [pessoas]. [...] a gente tá trabalhando em funcionalidades muito distintas, então às vezes acontece de termos um paralelismo de branches

muito grande. [...] é muito complicado ‘mergear’ e fazer *rebase* de tudo. Realmente dá muitos conflito” – P3 – Web – CORP

4.3.1.3 Developer Awareness [AWA]

Na amostra é possível identificar que a prática de *Developer Awareness* [AWA] é amplamente adotada nas companhias, assim como na amostra do estudo original. Em 6 entrevistas foi possível perceber que o time de desenvolvimento era o mesmo responsável pela entrega e manutenção da aplicação. Em outras 2, há um time específico de *DevOps*, mas não havia grandes silos entre este e a equipe de desenvolvimento. Um caso interessante foi o de P11, que, apesar de um conhecimento espalhado dentro da equipe, há receio e insegurança por parte de alguns a respeito de questões de infraestrutura no geral.

4.3.2 Implantação Contínua

É possível inferir, baseado principalmente na Tabela 1, que as técnicas ligadas ao processo de implantação contínua estão presentes na maioria das equipes da amostra: as 3 estão nas quatro primeiras colocações.

4.3.2.1 Health Checks [HC]

Sobre a prática de *Health Checks* [HC] [14], é possível perceber que, apesar de não ser o mais adotado – como foi no estudo original (ver Figura 7), se destaca na nossa amostra, em relação à maioria das práticas, presente na segunda colocação. Na amostra, 7 entrevistados adotam pelo menos uma forma rudimentar de verificação e alertas, e 2 entre eles fazem verificações não muito complexas. Um ponto interessante que surgiu foi o fato de P4 achar desnecessário o uso dessa técnica em função do projeto se encontrar, à época da entrevista, em fase de prototipação.

4.3.2.2 Developer on Call [DOC]

Na amostra é possível perceber que a técnica de *Developer on Call* [DOC] [5] é mais adotada de forma implícita do que de fato definida – 3 entrevistados estão em times que funcionam desta forma. Contudo, 5 entrevistados não utilizam esta prática: alguns comentaram que a confiança nos testes automáticos faz com que não utilizem a prática, enquanto outro comentou que a prática é inclusive mal vista pela empresa.

Uma dicotomia interessante foi encontrada entre os resultados da amostra deste trabalho e o do estudo original. Este último comenta que a prática já está sendo largamente aceita nas organizações atualmente, e inclusive um dos entrevistados comenta que essa responsabilidade de ficar até mais tarde para resolver problemas leva os desenvolvedores a escreverem e testarem seus códigos mais veementemente. Tal argumentação tem como base a seguinte citação retirada do artigo e traduzida:

“Se você não tem testes suficientes e faz deploy de um código ruim isso vai se voltar contra você pois você estará de plantão e terá que dar suporte a isto.” – P14 (do estudo original) – Web – CORP

Contudo, com a seguinte citação de P2, é possível perceber que há um sentimento contrário: confia-se no processo de qualidade e, por isso, não necessitam de plantão.

“Temos o ciclo de QA, se encontrar alguma coisa a gente vê [...] não precisa isso de plantão não.” – P2 – Web – CORP

4.3.2.3 Pipeline de Implantação [PIP]

Em relação à prática de Pipeline de Implantação [PIP] [14] é possível inferir que ela é amplamente adotada pela amostra, visto que apenas 2 entrevistados obtiveram nota 0 (não utiliza). No estudo original é possível perceber um resultado semelhante a este. Uma grande parte dos entrevistados segue o mesmo padrão para qualquer tamanho da mudança. Outros 2 tinham alguns processos automatizados, mas a *pipeline* era diferente dependendo do tamanho da mudança.

É interessante perceber que alguns times ainda demonstram falta de automação dos processos da *pipeline*. Isto acontece em decorrência da complexidade da automação, devido às tecnologias e ferramentas utilizadas, ou da falta de prioridade do time para tal.

4.3.3 Entregas Parciais

Na amostra foi possível perceber que todas as técnicas relacionadas a Entregas Parciais são muito pouco utilizadas: todas estão entre as 4 últimas colocações.

4.3.3.1 Canary Releases [CAN]

A técnica de *Canary Releases* [CAN] [12] apareceu nos contextos de jogos e sistemas embarcados. Na equipe do entrevistado P5, que trabalha no domínio de jogos eletrônicos, os principais jogadores – conhecidos como “baleias” – são escolhidos para participar de um *early access* de novas funcionalidades. Esses testes levam em torno de 1 semana.

Já na equipe de P4, que trabalha com sistemas embarcados, a prática é necessária devido ao contexto de atuação e às tecnologias utilizadas. Como o sistema que está em fase de prototipação servirá para o contexto médico, vários testes de campo deveriam ser feitos para garantir que todas as funcionalidades estivessem de acordo com o esperado. Para os testes, o cliente que contratou a empresa de P4 escolhia a quantidade de pessoas e local que serviria como validação de funcionalidades.

Entre os entrevistados que não utilizam a prática de *Canary Releases*, 3 têm um ambiente de homologação para testes e validação de requisitos, mas este utiliza dados diferentes dos de produção. Outros 3 comentam que as features são sempre entregues para todos os usuários ao mesmo tempo.

4.3.3.2 Dark Launches [DAR]

Do grupo das práticas associadas a Entregas Parciais, *Dark Launches* [DAR] [5] foi a segunda mais utilizada, mas a menos conhecida entre os entrevistados. Isto se confirma com o fato de que 6 entrevistados dizem nunca ter utilizado, e 3 destes afirmam explicitamente que não conhecem a técnica. Importante notar que no estudo original [17] esta é a prática menos utilizada.

Dark Launches é utilizado totalmente por P5 no contexto de jogos para testes manuais, e apenas parcialmente no contexto de WEB por P9 para validações de alguns cenários que dependiam de dados de produção.

“A gente implementa a funcionalidade mas condiciona a não aparecer para o usuário até que a gente queira.”
– P5 – Jogos – Startup

4.3.3.3 Testes A/B [AB]

A utilização da prática de Testes A/B [AB] [13] não foi identificada entre os participantes. Entre as principais causas levantadas para a não utilização, 4 entrevistados dizem que não utilizam pela baixa quantidade de usuários ativos no sistema. Outros 2 consideram que a técnica não se aplica ao contexto da aplicação. É interessante notar que esses dois motivos estão presentes no estudo original [17], contudo, ao contrário deste, ninguém na amostra comentou sobre problemas na arquitetura como causa para não utilização.

“O sistema da gente – apesar de lidar com uma massa de dados muito grande – não têm tantos usuários, então não faz muito sentido [utilizar testes A/B]...” – P2 – WEB – Corp

Outro motivo importante foi levantado por P8, que comenta que aparentemente não existe na empresa o interesse em investir nessa prática. P3 comenta ainda que o time está mais focado em entregar novas funcionalidades, pois a demanda por parte do cliente é muito grande.

4.4 Descobertas adicionais

Esta subseção abordará sobre os dois tópicos marginais que foram identificados durante a análise dos dados obtidos pelas entrevistas: *Code Review* e testes automáticos.

4.4.1 Code Review

A prática de *Code Review* [1] é bem vista pela maioria dos entrevistados, considerado uma técnica importante e essencial para o controle de qualidade de código. As razões para o uso são diversas, desde de impor boas práticas – por P7 – até o compartilhamento e nivelamento do conhecimento – por P11.

“... a partir do momento que o sistema vai crescendo, o próprio desenvolvedor não tem a noção de que aquela sua mudança não é a melhor forma de fazer e que pode quebrar outras partes do sistema...” – P2 – WEB – Corp

Ainda na amostra, 3 entrevistados acreditam que a prática funciona principalmente para troca de conhecimento. Sobre isso, a entrevistada P10 comenta que agrega muito valor para ela como novata no time, mas acha que adiciona um tempo desnecessário na entrega de funcionalidades por pessoas mais experientes, visto que – para ela – a técnica não faria sentido neste caso.

Outro ponto importante foi a distinção entre dois relatos a respeito do engajamento do time com a prática. Enquanto P3 comenta que a equipe dela é bem aberta ao debate e está engajada com o processo, P7 fala que o processo funciona “mais ou menos”, dependendo do humor dos revisores.

4.4.2 Testes Automáticos

Testes automáticos são, no geral, bem vistos e extremamente recomendados pela maioria dos entrevistados como forma de prevenir erros em tempo de execução. Contudo, o participante P2 comenta que é contrário ao fato da garantia de qualidade depender exclusivamente desses testes.

“... automação [de testes] não resolve todos os problemas [relacionados a garantia de qualidade], ele vai identificar muita coisa, mas tem várias outras que precisamos do olhar de um testador...” — P2 – WEB – Corp

Na amostra, 5 entrevistados acreditam que o aumento da cobertura de testes automáticos pode diminuir a quantidade de bugs em produção. Dentro desse grupo, o participante P9 comenta que isto poderia, no entanto, diminuir a velocidade de entregas do time. P5 adicionou ainda que sente que a *sprint* é muito curta e não consegue tempo dentro destas para adicionar testes automáticos.

5 Discussão

Sobre o processo de integração contínua, pode-se inferir que as técnicas ligadas ao *merge* contínuo estão pouco presentes, apesar de *Developer Awareness* ser a prática mais utilizada da amostra. A maioria entrevistada integra o código na *branch* principal apenas no final da *sprint*. Este comportamento pode trazer vários problemas de integração de código quando atualizado concorrentemente, causa principal para a criação da prática de *Trunk Based Development* [5]. É válido um trabalho mais quantitativo na região que busque razões específicas para a não utilização desta, aliado a uma divulgação maior dos benefícios que essa técnica pode trazer para times de desenvolvimento.

É possível perceber ainda que todas as práticas ligadas à implantação contínua estão muito presentes na amostra. Mesmo com a prática de *Developer on Call* sendo mais utilizada de forma implícita do que de fato definida, e com algumas empresas interpretando-a como uma má prática, ela está em 4º lugar no ranking de utilização montado e é a de menor colocação do grupo. Isso demonstra principalmente que, na amostra, há um trabalho bem evoluído de otimização e automação do processo de entrega, além de utilização de ferramentas para garantir que a aplicação está rodando corretamente em produção.

Por fim, foi possível perceber que os entrevistados utilizam muito pouco as práticas de entregas parciais. Foi possível identificar técnicas desconhecidas por um grupo razoável de participantes – *Dark Launches* – e até técnicas que não foram utilizadas por nenhum deles – Testes A/B. É necessário um trabalho maior de disseminação de como realizar e das vantagens que essas técnicas podem trazer para produtos produzidos em Recife.

Todos esses pontos demonstram uma certa congruência da amostra de Recife com a amostra europeia e norte-americana do estudo original. Assim, podemos afirmar que a amostra está no mesmo patamar no sentido de utilização de outras partes do mundo, mas é necessário uma pesquisa quantitativa para poder generalizar esse achado para a capital pernambucana.

Para ampliar e obter novos pontos de discussão para o presente estudo, podem ser realizadas entrevistas com um grupo maior de

pessoas. A reaplicação poderia levantar novas barreiras e até identificar lacunas que não puderam ser vistas com a amostra deste trabalho. Uma destas lacunas se deve ao fato de que o formato utilizado para entrevistas impossibilitou que fossem identificadas razões para a não utilização das técnicas, uma vez que o questionário é estruturado de forma a obter informações sobre determinada prática de forma indireta. Outro ponto que também seria de grande valia seria a pesquisa quantitativa dentro deste mesmo tema, com o objetivo de entender melhor que práticas estão sendo utilizadas na indústria de Recife com um grupo maior de entrevistados, permitindo assim a generalização dos achados para a capital.

6 Ameaças a validade

Como em qualquer estudo, este também apresenta algumas limitações. Uma delas diz respeito ao número de desenvolvedores que concordaram participar do estudo. Para diminuir os efeitos disso, foi escolhido uma amostra mais diversa em relação a tamanho da corporação que trabalha.

É importante salientar também que os códigos gerados durante o processo de codificação das entrevistas são passíveis de enviesamento visto que este trabalho replica parte de um estudo anterior. O viés ocorre porque os autores já tinham em mente que assuntos estavam sendo procurados na fala durante o levantamento de códigos.

Outra ameaça que deve ser levada em conta é o processo de *coding* realizado. Ele foi feito baseando-se no áudio das entrevistas, e não nos textos transcritos, como geralmente é feito [9]. Isso pode tornar os códigos enviesados ou até mesmo significar a falta de códigos importantes que poderiam ter sido levantados através da análise de transcrições. É válido citar que há algumas vertentes que defendem a codificação através do áudio [19]. Entre os benefícios de realizar o processo de codificação por essa técnica estão a preservação da entonação e intenção do usuário, que são aspectos difíceis de capturar por meio de transcrições. Contudo ainda devem ser feitos trabalhos quantitativos na área para validação e melhor definição do processo.

7 Trabalhos Relacionados

Há vários trabalhos relacionados a benefícios e barreiras que existem na adoção de integração contínua em contextos variados. É possível perceber que no geral os desenvolvedores gostam de utilizar CI por garantir um desenvolvimento de uma forma mais segura e confiável [15] e por se sentirem mais produtivos [10]. É possível perceber também que os desenvolvedores ainda acham que as ferramentas de CI são complicadas e difíceis de configurar [10].

É interessante levantar ainda no contexto de integração contínua que utilizar as ferramentas voltadas para esta técnica sem adequar a cultura de desenvolvimento para tal leva a práticas não saudáveis de desenvolvimento, tais como *builds* que levam muito tempo para concluir ou permanecem quebrados por longos períodos de tempo [6]. Outro estudo encontrou que a utilização da técnica apresenta um *trade-off* entre velocidade e certeza no que diz respeito a garantia de qualidade do software [10].

Já a respeito de implantação contínua, foi encontrada a visão dos desenvolvedores do Facebook e de OANDA⁶ a respeito desta técnica [18]. O artigo mostra que os desenvolvedores preferem *deployments* mais rápidos por gerar maior qualidade de software e maior produtividade. Contudo, eles acreditam que há uma instabilidade maior e é uma metodologia inviável para sistemas críticos.

Traçando um comparativo com o estudo original [17], para a construção do questionário utilizado nesse estudo, houve uma adaptação das perguntas 1 e 2 do questionário original para que as questões 1 e 3 fossem elaboradas. Adicionalmente, foi proposta uma nova pergunta de pesquisa (PP2) para que se possa tentar entender se a *stairway to heaven* proposta é seguida no contexto de Recife. No que diz respeito aos resultados, o que foi obtido com a amostra deste trabalho se comportou de forma condizente àquela encontrada no estudo original [17]. Tanto no ranking de utilização de cada prática, bem como nas barreiras enfrentadas pelos desenvolvedores, houve uma congruência razoável entre os dois resultados, com apenas algumas diferenças no que tange a permutação de alguns elementos entre si. Isto serve como indicador de que tais práticas encontraram seu lugar na indústria e estão sendo implementadas de forma eficiente.

8 Conclusão

Com a realização deste estudo, percebeu-se dentro da amostra a presença de pelo menos alguma das técnicas de CI/CD presentes no cotidiano das empresas Recifenses. Com a análise da entrevista qualitativa, foi possível entender um pouco melhor como tais técnicas foram importadas e também algumas das razões do porque foram adotadas. Ainda que a amostra não siga fielmente o *Stairway to Heaven* proposto no estudo original [17], observou-se certa congruência no que diz respeito ao resultado de ambos os estudos.

Além disso, percebeu-se que testes A/B não eram utilizados, além de que *trunk based development* se mostrou pouco utilizado, visto que a maioria entrevistada relatou que a integração do código de uma nova funcionalidade para a *branch* principal de desenvolvimento era feita apenas no final da *Sprint*.

Referências

- [1] Bacchelli et al. 2013. Expectations, outcomes, and challenges of modern code review. (2013). <https://doi.org/10.1109/icse.2013.6606617>
- [2] Porto Digital. 2020. O que é o porto digital. <https://www.portodigital.org/parque/o-que-e-o-porto-digital>.
- [3] Digital.ia. 2020. 14th Annual State of Agile Report. <https://stateofagile.com/#ufhi-615706098-14th-annual-state-of-agile-report/7027494>.
- [4] Abhishek DWARAKI et al. 2015. GitFlow: Flow revision management for software-defined networks. (2015).
- [5] D. G. Feitelson et al. 2013. Development and Deployment at Facebook. *IEEE Internet Computing* 17(4):8-17 (2013), 8–17.
- [6] Wagner Felidré et al. 2019. Continuous Integration Theater. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–10.
- [7] Martin Fowler. 2006. Continuous Integration. <https://martinfowler.com/articles/continuousIntegration.html>.
- [8] Martin Fowler. 2013. Continuous Delivery. <https://martinfowler.com/bliki/ContinuousDelivery.html>.
- [9] Barney G. Glaser. 1992. Basics of Grounded Theory Analysis: Emergence Vs. Forcing. (1992).
- [10] Michael Hilton et al. 2016. Continuous Integration (CI) Needs and Wishes for Developers of Proprietary Code. (2016).

- [11] Pete Hodgson. 2017. Feature Toggles. <https://martinfowler.com/articles/feature-toggles.html>.
- [12] Jez Humble and David Farley. 2010. *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison-Wesley Professional.
- [13] Ron Kohavi et al. Agosto 2007. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. (Agosto 2007), 956–967.
- [14] Ingo Weber Len Bass and Liming Zhu. 2015. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional.
- [15] John Micco. 2012. Tools for Continuous Integration at Google Scale. https://www.youtube.com/watch?v=KH2_sB1A6lA.
- [16] Akond Ashfaq Ur Rahman et al. 2015. Synthesizing continuous deployment practices used in software development. (2015).
- [17] Gerald Schermann et al. 2016. An empirical study on principles and practices of continuous delivery and deployment. (2016).
- [18] Mitchel Douglas Tony Savor and Michael Gentili. 2016. Continuous Deployment at Facebook and OANDA. (2016).
- [19] Jade L. Ozawa-kirk Uma D. Parameswaran and Gwen Latendresse. 2020. To live (code) or to not: A new method for coding in qualitative research. (2020), 630–644.

⁶OANDA é uma pequena empresa privada que provê informações de moeda e troca de moeda como um serviço. Diariamente os serviços da empresa movimentam bilhões de dólares.