Universidade Federal de Pernambuco
Centro de Informática

Bacharelado em Engenharia da Computação

Pesquisa sobre práticas de CI e CD em Recife

Mateus Valgueiro Teixeira

Trabalho de Graduação

Recife
05 de maio de 2021

Universidade Federal de Pernambuco
Centro de Informática

Mateus Valgueiro Teixeira

Pesquisa sobre práticas de CI e CD em Recife

Trabalho apresentado ao Programa de Bacharelado em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: *Paulo Henrique Monteiro Borba*
Co-orientadora: *Klissiomara Lopes Dias*

Recife
05 de maio de 2021

DIGITE A DEDICATÓRIA AQUI

Agradecimentos

DIGITE OS AGRADECIMENTOS AQUI

DIGITE AQUI A CITAÇÃO
—AUTOR (NOTA)

Resumo

[WIP - Retirado da proposta TG]

As práticas de integração, entrega e implantação continuas (*Continuous Integration, Continuous Delivery* e *Continuous Deployment*) 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 pesquisam a respeito do estado da arte destas práticas nas empresas, principalmente no contexto da cidade de Recife.

Há também um grande equívoco em relação a nomenclatura das práticas e a real utilização das mesmas em vários aspectos no processo de desenvolvimento software. A utilização de termos e softwares voltados para práticas que não são seguidas de fato por muito projetos que os utilizam levam a um ambiente não saudável de desenvolvimento.

Neste contexto, o presente trabalho objetiva a realização de uma pesquisa quantitativa para entender como as práticas de integração, entrega e implantação continuas foram implantadas no contexto recifense, assim como avaliar o entendimento dos termos utilizados com os desenvolvedores locais.

Palavras-chave: DIGITE AS PALAVRAS-CHAVE AQUI

Abstract

This is the abstract

Keywords: DIGITE AS PALAVRAS-CHAVE AQUI

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

[WIP - Retirado da proposta de TG]

As práticas de integração, entrega e implantação continuas (*Continuous Integration, Continuous Delivery* e *Continuous Deployment*) já são muito difundidas e utilizadas por empresas de tecnologia em todo o mundo. Segundo [1], 95% dos participantes reportaram que sua organização pratica metodologias ágeis e 55% utiliza a técnica de integração continua. Também nesta mesma pesquisa foi encontrado que 41% utilizam a técnica de *Continuous Delivery*, e 36%, *Continuous Deployment*.

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 ágeis [5] e o seu uso traz maior qualidade ao software que está sendo produzido [9].

Ainda há, no entanto, uma grande falta de estudos a respeito de como essas práticas se exportaram para as grandes empresas de Recife [8], 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 [3].

Percebe-se também uma quantidade notável de indivíduos equivocados a respeito dos conceitos e do uso das práticas supracitadas [4]. O estudo [6] identificou que 60% dos projetos não utilizam a técnica de *commits* contínuos, mesmo utilizando uma ferramenta que tem como objetivo implantar a prática de integração contínua. Os autores chamam este fenômeno de *Continuous Integration Theather* (Teatro da integração contínua), e comentam que este processo produz um ambiente não saudável de desenvolvimento ágil.

Neste contexto, este trabalho tem como objetivo entender melhor, através de uma pesquisa quantitativa, como as práticas de integração, entrega e implantação continuas foram implantadas no âmbito das empresas de Recife, assim como avaliar se os termos utilizados são corretamente compreendidos pelos desenvolvedores destas, procurando assim possíveis "Teatros" recifenses.

CAPÍTULO 2

Motivação

Neste cápitulo será abordada a motivação por trás da pesquisa produzida. Na primeira seção, será falado um pouco sobre a grande adoção das práticas ágeis pela indústria. Já na segunda seção, será abordada a falta de estudos voltados para o contexto Recifense. A terceira discorre sobre o artigo base que serviu de motivação para este trabalho. A última seção contém as perguntas de pesquisa que este trabalho tenta responder.

2.1 A grande adoção de CI e CD

As práticas de integração e *deployment* contínuos (*Continuous Integration* e *Continuous Deployment*) já são muito difundidas e utilizadas por empresas de tecnologia em todo o mundo. Segundo [1], 95% dos participantes reportaram que sua organização pratica metodologias ágeis de desenvolvimento e 55% utilizam a técnica de integração continua. Também nesta mesma pesquisa foi encontrado que 36% utilizam a técnica de *deployment* contínuos.

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 [1], 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 difundida na indústria. O estudo [5] comenta que desenvolvedores envolvidos se sentem mais produtivos quando utilizam a prática e dão mais valor aos testes automáticos. Já com relação a *deployment* contínuo, o estudo [9] mostra que seu uso traz maior qualidade ao software que está sendo produzido.

2.2 A falta de estudos no contexto Recifense

Ainda há, no entanto, uma grande falta de estudos a respeito de como essas práticas se exportaram para as grandes empresas [8], inclusive em Recife, 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 somente em 2019 [3].

2.3 O artigo base

Com o objetivo de entender um pouco mais sobre o estado das práticas de CI/CD no contexto de Recife, nos baseamos no estudo [8], que busca entender como as práticas geralmente associadas a *Continuous Deployment* acharam o seu caminho na indústria européia e norte-americana. Para tal, foi utilizado um método misto de estudo empírico baseado em um pré-estudo na literatura, entrevistas qualitativas com 20 participantes e uma entrevista quantitativa que recebeu 187 respostas. A ideia era questionar até que ponto a sabedoria na area estava dominada por peculiaridades de um pequeno grupo de grandes empresas, como Facebook e Google.

Durante o trabalho os autores definem a *stairway to heaven* (escada para o céu, em tradução livre), que tem como objetivo definir um caminho de evolução das empresas para um estágio de entregas sofisticado. Esta permeia práticas de integração contínua, *deployment* contínuo e entregas parciais e

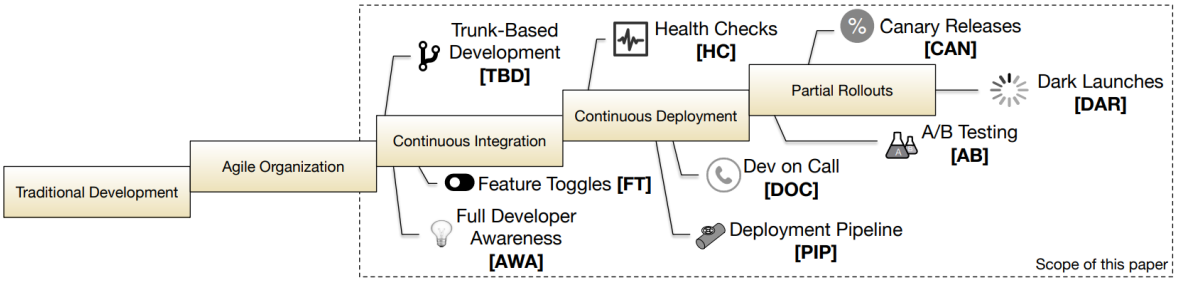

Figura 2.1 A escada de evolução denominada "Stairway to Heaven"proposta pelo artigo base. Fonte: Schermman et al (2016)

Com este caminho definido, os autores do artigo montaram as entrevistas qualitativas e quantitativas baseadas nas práticas que surgiram em cada degrau da evolução. O interessante da abordagem dos autores foi o roteiro consistir de perguntas sobre o processo utilizado, sem perguntar diretamente se o entrevistado utiliza determinada prática. Isso faz com que os resultados baseiem-se em um conjunto contido de definições, excluindo concepções diversas que pessoas diferentes possam ter a respeito de um mesmo assunto.

2.4 Perguntas de pesquisa

Assim, a grande adoção da indústria mundial das práticas de CI/CD, aliado a pequena quantidade de trabalhos a respeito no contexto recifense e a excelente metodologia proposta pelo artigo de Schermman et al [8] foram as principais motivações para o desenvolvimento deste trabalho. Para este as seguintes perguntas de pesquisa foram formadas:

1. Quais as práticas de CI/CD são utilizadas pelas empresas em Recife?
2. O cenário de CI/CD nas empresas em recife segue o *stairway to heaven* proposto no artigo?

3. Quais são os princípios e práticas subjacentes que governam a adoção de CD na indústria?

Para responder as perguntas acima foi decido reaplicar a pesquisa qualitativa do artigo [8] 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. Vale salientar que não foi replicado a pesquisa quantitativa presente no artigo base, mas esta pode servir como um trabalho futuro a este.

Além disso, as perguntas 1 e 3 são bem parecidas com as perguntas de pesquisa do artigo base, mas incluindo a técnica de *Continuous Integration*. Não obstante, uma terceira pergunta foi adicionada (*RQ2*), 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 outro, os próprios autores já refutaram esta definição com os seus resultados.

CAPÍTULO 3

Metodologia

Para o estudo, foi realizada uma pesquisa qualitativa através de entrevistas semi-estruturadas utilizando como base a entrevista desenvolvida pelo artigo base [8]. A primeira seção abordará uma visão geral sobre os primeiros passos da pesquisa. Já a segunda seção discorre sobre como as entrevistas foram conduzidas. Na terceira, encontra-se todo o processo utilizado para a análise dos dados obtidos na entrevista. Por último, a quarta seção mostra informações a respeito dos participantes da pesquisa.

3.1 Pré-estudo

Com o objetivo de entender melhor sobre como as práticas de CI/CD migraram para as empresas sediadas em recife, Mateus Valgueiro (autor deste trabalho) e Heitor Samuel realizaram um pré-estudo. Primeiramente, os dois discutiram a respeito do artigo base: *An empirical study on principles and practices of continuous delivery and deployment* [8]. Na discussão foi definido que seria replicado apenas o estudo baseado em entrevistas semi-estruturadas para garantir o entendimento dos termos pelos entrevistados e a adequação com as definições do artigo base.

Após a discussão, os entrevistadores acessaram o apêndice online deste [2] para obter o guia de entrevista utilizado. 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 haja alguma dúvida de definição de tópicos entre os entrevistadores.

Após isso, o guia de entrevistas do estudo base [2] também foi traduzido para português e utilizado durante as entrevistas. 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. O documento de consulta e a guia de entrevistas estão presentes no apêndice deste trabalho, ambos em sua versão traduzida.

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. Gerenciamentos de Problemas
5. Avaliação de Entrega

No guia, todas as sessões iniciam com uma questão aberta. As entrevistas seguiram os tópicos abordados em cada uma delas, mas sem ordem específica, respeitando o desenrolar da conversa. A primeira entrevista foi guiada pelos dois pesquisadores para assegurar que as próximas seriam feitas de forma semelhante pelos dois. Esta entrevista foi considerada como válida na análise de dados visto que não houve nenhuma mudança no questionário de entrevistas; o próprio 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: 5 conduzidas por Heitor e outras 5 conduzidas por Mateus.

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 50 minutos, duração bem parecida com os tempos obtidos no artigo base (35 a 60 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 de informações pessoais como nome ou empresa da qual o profissional se referia.

3.3 Análise dos Dados

A análise de dados foi feita apenas pelo autor deste artigo, Mateus Valgueiro. O processo escolhido foi baseado nas fases de *Coding* e *Thematic Analysis* da metodologia *Grounded Theory* [7]. 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

É importante salientar que os códigos gerados sofreram um certo enviesamento visto que este trabalho é uma replicação de um estudo, então o autor tinha em mente que assuntos estavam sendo procuradas na fala durante o levantamento de códigos.

Então, 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 quebrados em múltiplos. Depois, eles foram agrupados, quando compatíveis, em cada uma das 9 práticas descritas pelo artigo base e foi escolhida uma nota entre 0 e 2, representando não utiliza, utiliza parcialmente e utiliza completamente baseado nas definições do artigo base traduzidas. Este processo foi então replicado para cada uma das 11 entrevistas.

Ao final do processo ainda haviam 3 ligações entre a prática de *Dark Launch* e entrevistados que não tinham recebido nenhum código em comum. Para estes casos, foi enviado um email diretamente para cada um dos entrevistados com a definição do artigo base 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.

O agrupamento entre códigos e práticas gerou a Tabela T1, que mostra uma visão geral de cada relação entre prática e entrevistado, contendo a nota dada e os códigos que justificam a nota. Esta tabela contém 147 códigos.

De posse da tabela T1 foi gerada a Tabela T2, que contem uma visão reduzida da primeira, contendo apenas as notas de cada uma das práticas para cada um dos entrevistados com as colunas ordenadas de acordo com o *Stairway to Heaven* descrito no artigo base. Com a Tabela T2, também foi criada uma nova Tabela T3 com o mesmo formato da primeira, mas com as colunas ordenadas pela utilização de cada prática em ordem decrescente. Esta tem como objetivo replicar a Tabela 1 do artigo base [8].

Por fim, para identificar padrões e abstrações nos códigos agrupados na Tabela T1, foi feito um trabalho de agrupamento semântico, gerando por fim a tabela T4 apresentando as novas super categorias geradas no processo de *Thematic Analysis*. Neste, 17 super categorias foram geradas.

É importante salientar que ainda durante o processo de levantamento de códigos surgiram tópicos relevantes que não se relacionavam diretamente com as práticas descritas, mas que são perguntados pelo questionário e relevantes para o tema tratado. Surgiram então 2 novas colunas que agregavam códigos sobre as práticas de *Code Review* e de testes automáticos. Para estas foram relacionados 29 códigos, e duas super categorias foram geradas.

3.4 Participantes

No total foram entrevistados 11 desenvolvedores (P1 a P11) de 7 empresas diferentes sediadas em Recife. Destes, 3 eram mulheres. Pode-se ver a distribuição dos participantes entre gênero na Figura 3.1. Dentro desse grupo, 9 trabalhavam com aplicações Web, enquanto 1 trabalhava com sistemas embarcados e o último, com jogos.

A escolha dos entrevistados foi feita baseado na rede de conhecidos dos entrevistadores, com o propósito de agregar pessoas de que trabalhavam em empresas de tamanhos distintos para garantir uma variedade de parâmetros envolvidos. Como é possível perceber na Figura 3.2,

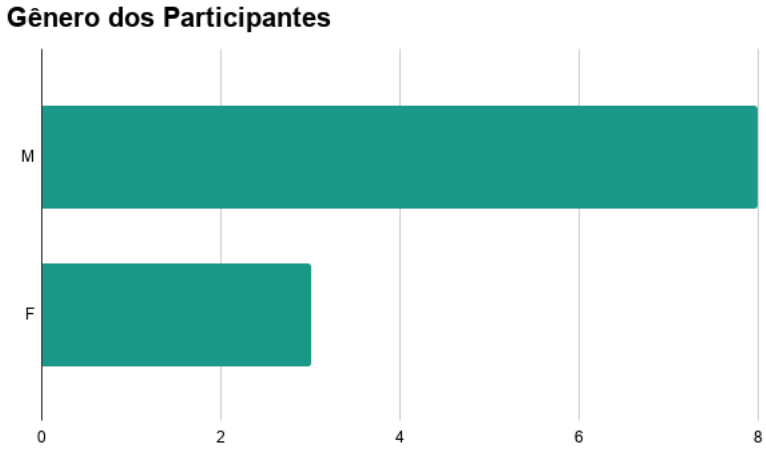

Figura 3.1 Distribuição dos Participantes por gênero

com relação a esse aspecto a amostra está bem distribuída.

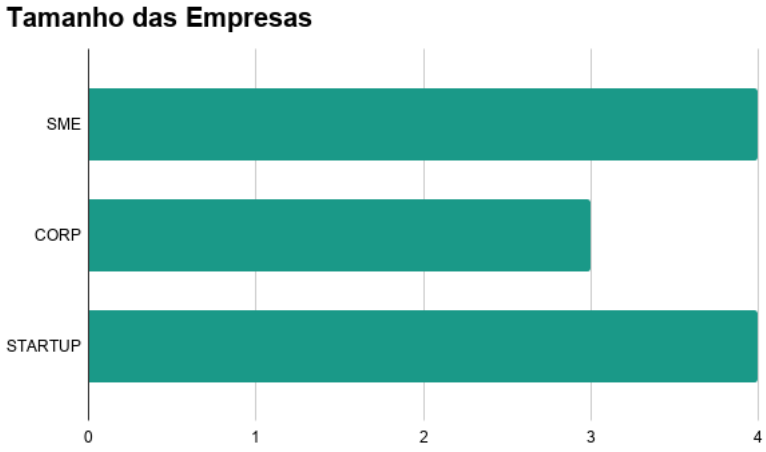

Figura 3.2 Distribuição dos Participantes por tamanho da empresa.

CAPÍTULO 4

Resultados

resultados

4.1 Estudo de Predomínio das Práticas

bla bla bla

4.2 Stairway to heaven

4.3 Analise das Práticas

4.3.1 Integração Contínua

4.3.1.1 Trunk Based Development [TBD]

4.3.1.2 Feature Toggles [FT]

4.3.1.3 Developer Awareness [AWA]

4.3.2 Deployment Contínuo

4.3.2.1 Health Checks [HC]

4.3.2.2 Developer on Call [DOC]

4.3.2.3 Deployment Pipeline [PIP]

4.3.3 Entregas Parciais

4.3.3.1 Canary Releases [CAN]

4.3.3.2 Dark Launches [DAR]

4.3.3.3 Testes A/B [AB]

4.4 Descobertas adicionais

4.4.1 Code Review

4.4.2 Testes Automáticos

CAPÍTULO 5

Conclusão

Está é a minha conclusão

APÊNDICE A

Questionário Traduzido

A.1 Demografia

- Qual seu cargo no trabalho atualmente?
- Qual o tamanho da empresa para o qual você trabalha?
- Qual é o tamanho típico das equipes dentro da empresa que você trabalha?
- Quantos anos de experiência profissional em <cargo do entrevistado> você tem?
- Qual o domínio (contexto: educação, saúde, etc) da empresa ou projeto no qual você trabalha?

A.2 Processo de entrega em geral

Dado uma funcionalidade recém-implementada, como é o processo de entrega da sua empresa a partir do commit da funcionalidade até chegar ao ambiente de produção e, portanto, aos clientes?

Possíveis questões complementares:

- O processo é o mesmo para toda mudança no código, isto é, é o mesmo para uma funcionalidade totalmente nova bem como para uma mudança pequena no código? Quem define/decide sobre esse "impacto" das mudanças?
- As mudanças são lançadas diretamente para todos os usuários?
- O quão automatizado é esse processo?
- Quanto tempo leva desde o commit de uma funcionalidade até que ela esteja disponível para os usuários?
- Qual a frequência típica de entrega da sua empresa? A cada commit, uma vez por dia, uma vez por semana, etc? Como isso se relaciona com os processos de desenvolvimento de software? Por exemplo, você está fazendo entregas "sempre" ou só no fim de uma sprint?

- Como os empregados (incluindo desenvolvedores) ficam informados acerca das entregas, por exemplo, qual versão está atualmente implementada, quais versões/funcionalidade estão sob teste em certos ambientes? Como essa comunicação é gerenciada entre equipes e nas equipes?
- Como você julga pessoalmente seu processo de entrega? O que funciona bem, e o que não funciona?

A.3 Papéis/Responsabilidades

Quais são, tipicamente, os papéis envolvidos no processo de entrega da sua empresa e quem é responsável pelo que?

É uma abordagem mais colaborativa com times contendo pessoal de operações e pessoal de desenvolvimento, ou existem equipes dedicadas, por exemplo: o time de operações assume a partir do momento que as mudanças feitas pelo time de desenvolvimento passaram os testes e estão prontas para serem entregues?

Possíveis questões complementares:

- Quem é responsável por monitorar a entrega uma vez que ela chegou em produção?
- Como você julga, pessoalmente, as definições de papéis na sua equipe? Elas fazem sentido? Algo está faltando, ou não está muito claro?
- Caso haja papel de DevOps: Como DevOps é realizado dentro da sua empresa?

A.4 Garantia da Qualidade

Quando é feito o commit de uma mudança no sistema de controle de versão (git, por exemplo) da sua empresa ou do seu projeto, como vocês garantem que commits errôneos ou commits que não seguem certas orientações/padrões não cheguem em produção? Como vocês garantem a qualidade do software na empresa?

Possíveis questões complementares:

- Caso seja um processo em estágios, quais estágios existem?
- Caso seja em estágios: Testes críticos são selecionados e executados em estágios iniciais para obter feedback mais rapidamente?
- É feito o build do software exatamente uma vez durante o processo todo, ou cada estágio requer uma build com configuração diferente? Tais arquivos de configuração são gerenciados pelos sistemas de controle de versão?
- Existem estágios/barreiras de qualidade que exigem uma aprovação manual explícita?
- Existe um ambiente parecido com o de produção ou alguma forma para que vocês tenham a certeza de que as mudanças vão funcionar em produção também?

- As revisões de código (code reviews) são parte do seu processo de análise de qualidade(testes)? Quais as razões para que isso seja feito/não seja feito?
- Você acha que o processo de análise de qualidade de vocês funciona bem? O que poderia ser melhorado, na sua opinião?

A.5 Gerenciamento de Problemas

Suponha que uma nova funcionalidade ou mudança chegou em produção, mas não se comporta como esperado(*). Quais são os passos executados e por quem são executados para gerenciar problemas em caso de:

1. Bugs devidos a erros no código
2. Defeitos devido a desvios dos requisitos, resultando em, por exemplo, performance ruim, mantenabilidade ruim, usabilidade ruim, etc.
3. Problemas com a disponibilidade de certas funcionalidades/serviços/partes da sua aplicação.

Adicionalmente: Como vocês identificam ou mensuram se o sistema está exibindo um comportamento inesperado?

Possíveis questões complementares:

- Tais problemas são comuns?
- Como são os problemas típicos que ocorrem em tempo de execução?
- O lançamento de uma funcionalidade requer que os desenvolvedores envolvidos estejam em plantão para lidar com tais problemas?
- Como tais problemas são identificados/descobertos? Existe suporte para isso fazendo uso de alguma ferramenta?
- Reversões (rollbacks) são utilizados? Caso afirmativo, quanto a compatibilidade de versões, como vocês lidam com problemas quando revertendo para versões incompatíveis?
- Você tem quaisquer sugestões para como prevenir uma boa parcela de problemas em tempo de execução?
- Em caso de uma arquitetura baseada em serviço: O quão comum é que novas versões de certos serviços sejam incompatíveis com outros serviços já existentes?

A.6 Avaliação da entrega

Dado a entrega contínua de novas versões, como vocês comparam se a versão mais recente é "melhor que" ou traz benefícios comparada com suas predecessoras? Como vocês avaliam se uma certa feature/mudança:

- Está satisfazendo as demandas dos usuários
- Tem o impacto desejado no lucro, performance, mantenabilidade, usabilidade, ... ?

Suponha que você quer comparar duas versões da sua aplicação que foram entregues e estão em execução. Quais as métricas que você consideraria?

Possíveis questões complementares caso live testing (Canary, A/B, ...) for usado:

- Qual a duração de tais testes?
- Qual a quantidade? Quantos testes rodam em paralelo?
- Cada um desses experimentos testa exatamente uma feature?
- Como esses testes são avaliados, quando, e em quais intervalos?
- Quem define as métricas e as limiares que se deve observar?
- Como você se certifica que os experimentos paralelos não estão influenciando um ao outro?
- Qual o escopo dos testes? Quais usuários são selecionados para tais testes e como eles são selecionados?

A.7 Finalização

Caso não tenha sido perguntado ou mencionado durante a entrevista:

- Que tipo de arquitetura tem o sistema? Monolítica, baseada em serviços (microsserviços)?
- Caso teste A/B não seja utilizado: Quais são as razões para não utilizar técnicas como teste A/B?
- Como você lida com funcionalidades que ainda não estão prontas para serem entregues, especialmente se elas necessitam de mudanças mais complexas através da base de código? Você faz uso de condicionais no código que previne tais funcionalidades de serem executadas até estarem prontas?
- Você tem processos de entrega diferentes para projetos diferentes?
- Na sua opinião, o quão automatizado os processos de entrega devem ser? Por exemplo, deve haver uma aprovação manual antes de fazer uma reversão automática para uma versão anterior (rollback) quando certos limiares não são atingidos?

Referências Bibliográficas

[1] 14th annual state of agile report, 2020.

[2] Continuous deployment study - online appendix, 2020.

[3] O que É o porto digital, 2020.

[4] Adam Debbiche, Mikael Dienér, and Richard Berntsson Svensson. Challenges when adopting continuous integration: A case study. In *International Conference on Product-Focused Software Process Improvement*, pages 17–32. Springer, 2014.

[5] Michael Hilton et Al. Continuous integration (ci) needs and wishes for developers of proprietary code. 2016.

[6] Wagner Felidré et Al. Continuous integration theater. 2019.

[7] Barney G. Glaser. Basics of grounded theory analysis: Emergence vs. forcing. 1992.

[8] et al. Schermann, Gerald. An empirical study on principles and practices of continuous delivery and deployment. 2016.

[9] Mitchel Douglas Tony Savor and Michael Gentili. Continuous deployment at facebook and oanda. 2016.