



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

MAX WENDEL FERREIRA FREIRE ARAÚJO

**ANÁLISE DA UTILIZAÇÃO DA INTEGRAÇÃO CONTÍNUA NO AMBIENTE DE
DESENVOLVIMENTO DO NPI**

QUIXADÁ
2019

MAX WENDEL FERREIRA FREIRE ARAÚJO

ANÁLISE DA UTILIZAÇÃO DA INTEGRAÇÃO CONTÍNUA NO AMBIENTE DE
DESENVOLVIMENTO DO NPI

Monografia apresentada ao curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação. Área de concentração: Computação.

Orientador: Prof. Me. Carlos Diego Andrade de Almeida.

QUIXADÁ

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A69a Araújo, Max Wendel Ferreira Freire.

Análise da utilização da integração contínua no ambiente de desenvolvimento do NPI / Max Wendel Ferreira Freire Araújo. – 2019.

55 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2019.

Orientação: Prof. Me. Carlos Diego Andrade de Almeida.

1. Integração Contínua. 2. Gerenciamento de configuração de software. 3. Gitlab. I. Título.

CDD 005

MAX WENDEL FERREIRA FREIRE ARAÚJO

ANÁLISE DA UTILIZAÇÃO DA INTEGRAÇÃO CONTÍNUA NO AMBIENTE DE
DESENVOLVIMENTO DO NPI

Monografia apresentada ao curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.
Área de concentração: Computação.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

Prof. Me. Carlos Diego Andrade de Almeida (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Regis Pires Magalhães
Universidade Federal do Ceará (UFC)

Ma. Virginia Farias de Sousa
Universidade Federal do Ceará (UFC)

A Deus.

A minha mãe e avó Ivaneide e Neide,

A minha companheira Andreza.

AGRADECIMENTOS

A minha família, em especial minha mãe, Ivaneide Ferreira Freire e minha avó Neide Ferreira Freire, pelo apoio, suporte, carinho e amor incondicional.

A Andreza de Souza Brito pela amizade verdadeira, companheirismo e por ter me incentivado nos estudos.

Aos meus amigos pela amizade e apoio.

Ao Prof. Me. Carlos Diego Andrade de Almeida, por ter aceitado me guiar nesta caminhada, pela paciência e pela excelente orientação.

Aos professores participantes da banca examinadora Dr. Régis Pires Magalhães e Ma. Virginia Farias de Sousa pelo tempo, pelas valiosas colaborações e sugestões.

Aos supervisores entrevistados, pelo tempo concedido nas entrevistas.

Aos colegas, pela amizade e ajuda em todos os momentos.

A todos os alunos que participaram da pesquisa.

A todos que direta ou indiretamente contribuíram para a realização deste trabalho.

“Vigie seus pensamentos, eles tornam-se
palavras.
Vigie suas palavras, elas tornam-se ações.
Vigie suas ações, elas tornam-se hábitos.
Vigie seus hábitos, eles formam seu caráter.
Vigie seu caráter, ele se torna seu destino.”
(Frank Outlaw).

RESUMO

O NPI conta com vários projetos, e seus integrantes são, em sua maioria, alunos do Campus da UFC de Quixadá, o que indica a inexperiência das equipes em seu ambiente de desenvolvimento. Contando com isso, o NPI oferece o suporte de professores e supervisores para orientar esses alunos em suas atividades com uma organização de um ambiente de desenvolvimento real de uma fábrica de *software*. Para auxiliar o desenvolvimento e automatizar boa parte de seus processos o NPI adicionou em seus projetos a Integração Contínua, que vem como uma ajudante nas questões de avaliação de tarefas e como uma monitora da qualidade dos produtos desenvolvidos pelo núcleo. No entanto, não foi avaliado se a adição desta ferramenta realmente contribuiu ou prejudicou a equipe, sabe-se somente que segundo a literatura, a contribuição é presenciada, mas não comprovou-se ainda eficácia. Neste contexto, com o objetivo verificar se a ferramenta adicionada nesse ambiente descrito anteriormente trouxe algum benefício ou dificuldades, este trabalho propõe uma avaliação do impacto da Integração Contínua no ambiente de desenvolvimento do NPI. O trabalho consiste em avaliar a experiência e adaptação dos participantes com a adição da ferramenta, analisar os dados dos repositórios para verificar as mudanças nas frequência de submissão de atividades, resolução de problemas e indicadores da produtividade da equipe, comparando um período antes a depois da aplicação da Integração Contínua, para que por fim possa indicar o real impacto obtido. Para isso foi realizado entrevistas com alunos e supervisores, para identificar a experiência e adaptação, e foi realizado inspeções nos repositórios dos projetos que aderiram a CI, para verificar os dados dos repositórios explicados anteriormente. A partir dos resultados obtidos, foi possível indicar um impacto positivo na adoção da ferramenta no ambiente de desenvolvimento do NPI.

Palavras-chave: Integração contínua. Gerenciamento de configuração de software. Gitlab.

ABSTRACT

NPI has many projects, and its members are mostly students from the UFC Quixadá Campus, which indicates the inexperience of the teams in their development environment. With this in mind, NPI offers the support of teachers and supervisors to guide these students in their activities with organizing an actual software factory development environment. To help development and automate most of its processes, NPI has added Continuous Integration to projects, which comes as a helper on task assessment issues and as a monitor of the quality of products developed by NPI. However, it was not evaluated whether the addition of this tool really contributed or harmed the team, it is only known that, according to the literature, the contribution is witnessed, but it has not yet been proven effective. In this context, aiming to verify if the tool added in this environment described above brought any benefits or difficulties, this paper proposes an evaluation of the impact of Continuous Integration in the development environment of NPI. The work consists to evaluate the participants' experience and adaptation with the addition of the tool, analyzing repository data to verify changes in activity submission frequency, problem solving and team productivity indicators, comparing a period before and after to addition of the Continuous Integration, so that it can finally indicate the actual impact obtained. To this end, interviews were conducted with students and supervisors to identify experience and adaptation, and inspections were conducted on project repositories adhering to the IC, to verify data from the repositories explained above. From the results obtained, it was possible to indicate a positive impact on the adoption of the tool in the NPI development environment.

Keywords: Continuous integration. Software Configuration Management. Gitlab.

LISTA DE FIGURAS

Figura 1 – Etapas de execução do trabalho.....	27
--	----

LISTA DE GRÁFICOS

Gráfico 1 – Histórico de violações	25
Gráfico 2 – Média dos pontos a serem avaliados	33
Gráfico 3 – Dados obtidos com a inspeção do projeto Proficiência	40
Gráfico 4 – Dados obtidos com a inspeção do projeto Gestão por Competências	40
Gráfico 5 – Dados obtidos com a inspeção do projeto Gestão de Riscos	41
Gráfico 6 – Dados obtidos com a inspeção do projeto Dados Abertos	41
Gráfico 7 – Dados obtidos com a inspeção do projeto Prontuário Eletrônico Odontológico	42
Gráfico 8 – Status das <i>pipelines</i> dos projetos	42

LISTA DE QUADROS

Quadro 1 – Comparação dos trabalhos	26
Quadro 2 – Atributos das inspeções	30
Quadro 3 – Conjunto de datas utilizados nas inspeções	30
Quadro 4 – Projetos selecionados	34
Quadro 5 – Impacto dos pontos a serem avaliados observados pelos estagiários	36
Quadro 6 – Impacto dos pontos a serem avaliados observado pelos supervisores	38

LISTA DE ABREVIATURAS E SIGLAS

NPI	Núcleo de Práticas em Informática
UFC	Universidade Federal do Ceará
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicações)
CI	<i>Continuous Integration</i> (Integração Contínua)
<i>job</i>	Conjunto de operações que se executam simultaneamente
TCLE	Termo de Consentimento Livre e Esclarecido
<i>rebase</i>	União de dois ramos de códigos que diferencia-se por manter a linearidade do histórico de desenvolvimento
<i>merging</i>	União de dois ramos de códigos que mantém os ramos “pais” no histórico de desenvolvimento

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivo geral	15
1.2	Objetivo específico	15
1.3	Estrutura do trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Integração Contínua	17
2.2	Núcleo de Práticas em informática	18
2.3	GitLab	19
3	TRABALHOS RELACIONADOS	21
3.1	Experienced Benefits of Continuous Integration in Industry Software Product Development: A Case Study	21
3.2	Quality and Productivity Outcomes Relating to Continuous Integration in GitHub	22
3.3	Implantação de uma ferramenta de Integração Contínua em um Núcleo de Práticas em informática: relato de experiência	23
4	METODOLOGIA	27
4.1	Aplicação do questionário	27
4.2	Seleção dos pontos a serem avaliados	28
4.3	Seleção dos projetos	28
4.4	Aplicação do teste piloto	29
4.5	Realização das entrevistas	29
4.6	Realização das inspeções nos repositórios dos projetos selecionados	30
4.7	Avaliação dos resultados	31
5	RESULTADOS	32
5.1	Resultados do Questionário	32
5.2	Resultados do roteiro piloto	33
5.3	Projetos selecionados	34
5.4	Resultados das entrevistas	35
5.4.1	Estagiários	35
5.4.1.1	<i>Experiência e Adaptação</i>	35
5.4.1.2	<i>Pontos a serem avaliados</i>	36

5.4.1.3	<i>Sugestões</i>	37
5.4.2	Supervisores	37
5.4.2.1	<i>Experiência e adaptação</i>	37
5.4.2.2	<i>Pontos a serem avaliados</i>	38
5.4.2.3	<i>Sugestões</i>	39
5.5	Inspeção	39
5.5.1	Proficiência	40
5.5.2	Gestão por Competências	41
5.5.3	Gestão de Riscos	41
5.5.4	Dados Abertos	42
5.5.5	Prontuário Eletrônico Odontológico	42
5.5.6	Avaliação das pipelines dos projetos	43
5.6	Avaliação dos resultados	43
6	CONCLUSÃO	44
6.1	Pontos que afetam a conclusão da pesquisa	45
6.2	Trabalhos futuros	45
	REFERÊNCIAS	46
	APÊNDICE A – ROTEIRO PILOTO	49
	APÊNDICE B – ROTEIRO FINAL (ESTAGIÁRIOS)	50
	APÊNDICE C – ROTEIRO FINAL (SUPERVISORES)	51
	APÊNDICE D – QUESTIONÁRIO	52
	APÊNDICE E – TCLE	53

1 INTRODUÇÃO

Uma característica comum de muitos projetos de software é que por longos períodos de tempo, durante o processo de desenvolvimento, o aplicativo não está em um estado de funcionamento (HUMBLE e FARLEY, 2010). De fato, a maioria dos softwares desenvolvidos por grandes equipes gastam uma proporção significativa de seu tempo de desenvolvimento em resoluções de problemas no código, tratamento de *builds* que falharam e em falhas na comunicação com a equipe. É desejável que o projeto esteja, sempre que possível, em um bom estado de funcionamento, que seja avaliado os padrões de desenvolvimento adotados pela equipe e que seja mantida a qualidade do produto que está sendo desenvolvido.

A Integração Contínua é uma ótima solução para tais casos, pois possibilita executar frequentes integrações de *builds*, compilações, reconstrução do banco de dados, execução de testes automatizados e inspeções, implantação da aplicação e o obter um feedback (DUVALL, MATYAS e GLOVER, 2007). Com isso, é possível garantir que a equipe de desenvolvimento está seguindo o padrão de código e que o código está em um bom estado de funcionamento.

Atualmente, para a implantação da CI, existem várias ferramentas que tornam esse processo mais simples, além dessas ferramentas serem automatizadas e não necessitarem da intervenção de um desenvolvedor nesse processo (POLKHOVSKIY, 2016). Um exemplo de ferramenta é o GitLabCI que é a solução de Integração Contínua do Gitlab.

O GitLab é uma ferramenta de gerenciamento de repositórios *Git*, que integra recursos de *wiki*, rastreamento de *issues* e *pipelines* de CI (entre outros recursos) em projetos públicos e privados (BAARSEN, 2014). A mesma é usada nos projetos do Núcleo de Práticas em Informática (NPI), sendo que alguns projetos usam, especificamente, o seu recurso de CI.

O NPI é um programa que fornece soluções tecnológicas para o sertão central do Ceará. Localizado na Universidade Federal do Ceará (UFC) - Campus de Quixadá, o seu objetivo é atender a comunidade, fornecendo soluções de TI, e aos próprios alunos da universidade, provendo estágios (NPI, 2019). O NPI aderiu recentemente o uso das práticas de CI no seu ambiente de desenvolvimento de projetos, com o uso das ferramentas, do GitLab que dá suporte a essas práticas. Devido a implantação dessa ferramenta ser algo recente no NPI, é importante que seja realizada uma avaliação do seu uso, para verificar o impacto.

Como exemplos de avaliação do impacto de tais ferramentas há trabalhos como os de Stâhal e Bosh (2013) e Vasilescu, Yue Yu et al. (2015) que avaliaram o impacto da CI em

projetos que possuíam equipes de desenvolvedores. Ambos mostraram que a utilização da CI nos ambientes de projetos geraram um impacto positivo tanto na qualidade quanto na produtividade. Ambos os trabalhos abordam cenários em que há equipes de desenvolvimento cujos desenvolvedores são experientes, tornando mais fácil a adaptação de tais ferramentas.

Este trabalho tem como objetivo avaliar o impacto da Integração em um ambiente com uma equipe pequena e pouco experiente no ambiente de desenvolvimento do NPI da UFC – Campus Quixadá, e avaliar também o processo de adaptação e experiência obtida pelos integrantes do NPI com a ferramenta.

1.1 Objetivo geral

Este trabalho tem como objetivo avaliar o impacto da utilização da Integração Contínua no ambiente de desenvolvimento do Núcleo de Práticas em Informática da Universidade Federal do Ceará – Campus Quixadá.

1.2 Objetivo específico

- Selecionar os critérios de avaliação a serem utilizados como indicadores do impacto da Integração Contínua a partir das necessidades do NPI.
- Analisar o uso da Integração Contínua em projetos que a utilize em seus repositórios a partir dos critérios de avaliação.
- Avaliar a experiência e adaptação dos integrantes do NPI com a adoção da ferramenta.
- Realizar uma comparação com o antigo modelo de desenvolvimento sem a Integração Contínua.
- Indicar o impacto gerado com a adoção da Integração Contínua no processo de desenvolvimento do NPI da UFC – Campus Quixadá.

1.3 Estrutura do trabalho

Esse trabalho foi dividido, além desse capítulo introdutório, por 6 capítulos descritos a seguir:

- O capítulo 2 aborda uma breve descrição sobre Integração Contínua, seus benefícios e seus desafios, sobre o NPI, seus objetivos e propósito, e sobre o Gitlab, e seus suportes;
- O capítulo 3 descreve alguns trabalhos relacionados ao tema listados no decorrer

do levantamento bibliográfico, indicando os pontos aos quais esta pesquisa se diferencia das demais;

- No capítulo 4 é descrito a metodologia que foi utilizada nessa pesquisa, caracterizando-a e apresentando a descrição das etapas ao qual a pesquisa foi dividida;
- No capítulo 5 é apresentado os resultados obtidos com a pesquisa;
- E finalmente, no capítulo 6 são apresentados a conclusão, os pontos que podem afetar a validação do trabalho e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção explana sobre os pontos mais importantes desse trabalho. Na Subseção 2.1 é explicado o que é *Continuous Integration*, seus benefícios e dificuldades enfrentadas na sua adoção em ambientes de desenvolvimento. A subseção 2.2 fala sobre o Núcleo de Práticas em Informática, seu cenário atual e antigo de desenvolvimento. Por fim, na subseção 2.3, há uma explicação sobre o Gitlab, suas ferramentas e seu *workflow*.

2.1 Integração Contínua

Segundo o que Martin Fowler (2006) descreveu em seu blog, *Continuous Integration* é uma prática de desenvolvimento de software em que os membros de uma equipe integram seu trabalho com frequência, geralmente por dia – tendo várias integrações por dia. Cada integração gera um processo de *build* novo, que é compilado, analisado e testado, sendo possível detectar os erros se o *build* falhar.

O processo da CI é dividido em vários passos, onde cada passo é responsável por executar diferentes tipos de análises no código, os mais comuns são os passos de compilação, análise e teste, sendo possível adicionar mais passos, isso vai de acordo com a necessidade da equipe e com a qualidade do produto. Se houver um erro em qualquer passo, a CI gerará um *build* quebrado (ou falho), se todos os passos forem executados e não existir erros, a CI gerará um *build* integro (ou sucesso), tais passos estão definidos em uma *pipeline* de execução. Esta, por sua vez, contém os passos a serem executados, a ordem das execuções, e os recursos que devem ser usados e salvos, assim sendo possível ter a automação da integração (DUVALL, MATYAS e GLOVER, 2007).

Segundo Fowler (2006) os benefícios mais abrangentes da *Continuous Integration*, em geral são:

- Redução de Risco, pois é possível saber o estado do projeto, de cada funcionalidade, os bugs que o projeto possui a todo momento.
- Facilitação da detecção de *bugs* relativa à qualidade da suíte de testes, pois é possível detecta-los ao submeter as alterações.
- Visibilidade do projeto para a equipe, pois todos poderão ver o que cada um fez e que mudanças foram feitas no projeto.

- Automatização do *deploy*, é possível programa-lo na *pipeline* de execução da CI.

Seguindo esses pontos Polkhovskiy (2016) descreve que, com a *Continuous Integration*, é possível ter:

- Redução de Risco, pois há um monitoramento frequente da saúde do projeto.
- Redução da repetição do processo manual, como compilação, testes, integração com o banco de dados, inspeção, implantação e feedback.
- Geração de um código implantável, pois é garantido pela CI, de acordo com os testes e as inspeções, que o código gerado é implantável.
- Permite a equipe melhor visibilidade e transparência do projeto, pois torna-se mais fácil a comunicação devido a atualização das informações do projeto feitas pela equipe.

E por fim, os trabalhos de Stâhal e Bosh (2013) e Vasilescu, Yue Yu et al. (2015) mostraram que com a CI é possível obter:

- Qualidade, pois toda alteração é testada e inspecionada.
- Produtividade, pois é possível ter um feedback rápido sobre o as alterações feitas, e assim, mais rápido para mudar novamente até passar.

No entanto, trabalhos como o de Hilton, Nelson *et al* (2016), que entrevistaram 105 desenvolvedores, mostra que há desenvolvedores que não adotam o uso da CI porque o projeto não possuem testes suficientes para implementação da CI, outros não adotam porque não possuem equipes familiarizadas com tais ferramentas, e outros responderam que não valiam a pena, ou que o custo era muito alto. Contudo, as ferramentas que implementam a CI oferecem bastante suporte e pouco custo, e, é possível garantir a qualidade do projeto utilizando análises estáticas e inspeção bugs, sem precisar de testes.

2.2 Núcleo de Práticas em informática

O Núcleo de Práticas em Informática foi criado com o objetivo de atender a comunidade acadêmica e a sociedade do sertão central com soluções de TI. O mesmo,

posiciona-se como um programa de extensão com múltiplas perspectivas, pois oferece soluções de TI para a comunidade e provimento de estágio para os alunos.

Segundo Gonçalves, Bezerra, *et al* (2013): “O NPI-UFC foi criado no ano de 2009, com o objetivo de suprir as necessidades de sistemas para uso interno do campus. Porém, com sua evolução percebeu-se outras possibilidades para este, como o provimento de estágio para estudantes dos cursos de graduação do Campus Quixadá.”, o que futuramente se expandiu para o atendimento da comunidade do sertão central, atendendo demandas de sistemas a serem desenvolvidos, atendendo o que era desejado, que era modelá-lo como uma fábrica de *software*.

Localizado no campus da UFC em Quixadá, o NPI possui várias equipes divididas em 2 turnos (manhã e tarde), cada equipe formada por alunos e supervisores. O supervisor é responsável por avaliar as atividades executadas de cada aluno, por liderar reuniões diárias e semanais, por comunicar-se com os clientes.

O processo de desenvolvimento do NPI é baseado, mais especificamente, no *Scrum*, pois há poucos membros nas equipes, as equipes são multidisciplinares, há uma comunicação frequente com os clientes e há documentações necessárias do projeto, devido a necessidade de cumprir prazos curtos, retirando o excesso de documentação de relatórios (GONÇALVES, BEZERRA, *et al.*, 2013).

2.3 GitLab

O Gitlab é um sistema de gerenciamento de repositórios, lançado em 2011 sob a licença do MIT, que se tornou uma ferramenta muito poderosa por conta de seus diversos recursos, e ótimo suporte, que auxiliam no desenvolvimento e no gerenciamento de projetos (HETHEY, 2013). O Gitlab oferece ferramentas de suporte a DevOps para todo o ciclo de vida do projeto tais como, *Continuous Integration*, *Continuous Deployment* e *Continuous Delivery*, Configuração do *Kubernetes*, *Auto DevOps*, Monitoramento da aplicação Segurança, entre outros recursos.

O Gitlab também oferece recursos de gerenciamento de projeto e auditoria, como Gestão de Atividades (*Boards*), Rastreamento de *Issues*, Wiki e gerenciamento de usuários, grupos e permissões (BAARSEN, 2014). O Gitlab também oferece suporte a uma hospedagem local de repositórios, com autorizações via LDAP.

Onde as 3 primeiras atividades referem-se à ideia e planejamento do projeto, com a

ferramenta de bate papo, Mattermost, e as ferramentas de criação e gerenciamento de *issues*. Nas atividades de *Code* e *Commit* há o processo de hospedagem e as ferramentas que auxiliam na auditoria dos *commit's*. Nas atividades de *Test* e *Review* entra o Gitlab CI com a realização da *Continuous Integration*, realizando o *build* da aplicação e rodando os testes automatizados, para depois serem revisados pelas ferramentas de *code review* do próprio Gitlab, ou usar outros analisadores estáticos. E por fim, as 3 últimas atividades são auxiliadas com o uso do Gitlab CD, que realiza o processo de *Continuous Deployment* (ou *Delivery*) para empacotar, implantar e monitorar a aplicação em um servidor, e por fim, o recebimento do *feedback* realizado pela ferramenta Cycle Analytics.

O NPI utiliza a hospedagem local do Gitlab no servidor do próprio campus, sendo possível ter a autenticação e autorização via LDAP. O gerenciamento de projeto e auditoria do NPI também são realizados pelo Gitlab, dividido e organizados por projeto. No entanto, os recursos oferecidos pelo Gitlab CI/CD estão começando a serem utilizados no ambiente de desenvolvimento do NPI.

O GitlabCI é uma solução de CI feita pela mesma equipe que criou o Gitlab, sendo assim, ele há uma ótima integração com o próprio sistema do Gitlab, é mais fácil de configurar e está totalmente integrado a interface do Gitlab, eliminando o credenciamento, ou a execução, de outro sistema com o mesmo objetivo (BAARSEN, 2014).

3 TRABALHOS RELACIONADOS

Esta Seção contém um relato sobre alguns trabalhos que possuem objetivos semelhantes a este, além de conter uma comparação com cada um, esclarecendo as diferenças e semelhanças (Quadro 1).

3.1 Experienced Benefits of Continuous Integration in Industry Software Product Development: A Case Study

Realizado por Stâhal e Bosch (2013), este trabalho apresenta múltiplos estudos de caso sobre a experiência de profissionais na indústria de software, que trabalham em projetos de larga escala, com a *Continuous Integration*, tendo como objetivo, validar os múltiplos benefícios da CI, segundo a literatura. Os autores investigaram se a CI realmente afeta os efeitos citados abaixo e se são realmente vivenciados no desenvolvimento de software na indústria:

- Maior produtividade dos desenvolvedores;
- Maior previsibilidade e
- Adoção de testes ágeis no projeto

O estudo envolve quatro produtos diferentes, e independentes, níveis de maturidade de CI, na empresa Ericson AB. Em cada um desses produtos, desenvolvedores, testadores, gerentes de projeto e gerentes de linha foram entrevistados. Suas experiências em relação ao uso da CI foram avaliadas e discutidas quantitativamente em comparação com os benefícios de CI propostos em trabalhos relacionados.

Stâhal e Bosch (2013) chegaram à conclusão de que não existe só um, mas vários benefícios com o uso da *Continuous Integration*. As hipóteses levantadas foram confirmadas com os dados coletados. Os dados também mostram um efeito positivo na comunicação da equipe e entre equipes, maior produtividade devido a facilidade de fazer um *rebase* e um *merging*, auxiliando o desenvolvimento paralelo. Foi concluído também que a CI possui uma relação com métodos ágeis de automação de testes de aceitação do cliente e de testes unitários escritos em conjunto com o novo código de produção.

Este trabalho se assemelha ao trabalho de Stâhal e Bosch (2013) por também realizar uma entrevista com cada membro da equipe de cada projeto e se diferencia por contar com

uma equipe menor e os projetos não serem de alta escala, diferencia-se também por realizar uma inspeção nos repositórios.

3.2 Quality and Productivity Outcomes Relating to Continuous Integration in GitHub

Realizado por Vasilescu, Yue Yu *et al* (2015), o objetivo do trabalho de foi avaliar se o uso da *Continuous Integration* melhora a produtividade das equipes em projetos *open source*, verificando se eles podem integrar mais contribuições externas sem ter uma diminuição observável na qualidade do software.

Os pontos avaliados por este projeto foram produtividade e qualidade provenientes de contribuições de projetos que usam o método de desenvolvimento baseados em *pull request*.

A partir de um conjunto de dados coletados do uso da CI em projetos do *GitHub*, este trabalho analisou a engenharia de software que se apresentam de forma diferencial com a introdução da CI versus sem. Em particular, este trabalho contribuiu com:

- A coleta de um conjunto abrangente de dados de 246 projetos do *GitHub*, em que algum momento de sua história, introduziram as funcionalidades do *Travis-CI* para o processo de desenvolvimento.
- A descoberta de que, após adicionar a CI, mais *pull request* dos principais desenvolvedores foram aceitas e menos foram rejeitadas. Menos envios de desenvolvedores secundários foram rejeitados. Isso foi devido ao fato de que a CI provê uma melhora nos tratamentos de *pull request* de membros interno, ao mesmo tempo tendo um efeito positivo global na qualidade inicial de submissões externas.
- A observação de que o uso da CI não diminui os relatos de bugs dos usuários, sugerindo que a CI não afeta negativamente a qualidade experimentada pelos usuários. Foi observado também, um aumento de bugs relatados pelos desenvolvedores, o que sugere que a *Continuous Integration* está ajudando os desenvolvedores a descobrirem mais defeitos.

O trabalho está dividido em duas etapas de execução, a coleta de dados e análise dos dados.

A etapa de coleta de dados foi dividida em 4 passos:

- Seleção de projetos: nesse passo, houve uma seleção de projetos que possuem um longo histórico de registros, linguagens de programação populares, e tivessem mais que 200 *pull request*.
- Coleta de dados na fonte e dos arquivos de testes: neste passo houve a coleta dos dados sobre os repositório fonte dos projetos, tais como tamanho da equipe, quantidade de *merges* aceitos e rejeitados, *issues* abertas pela CI, entre outras informações.
- Coleta de dados de produtividade: neste passo foi realizado uma inspeção nos repositórios para mensurar a quantidade de *merges* aceitos por mês em um intervalo de um período de 24 meses.
- Coleta de dados de qualidade: neste passo foi feito uma inspeção para identificar a quantidade de *bugs* nos projetos, através das mensagens de *commit* que possuíam palavras como “*fix*”, “*bugs*”, “*erro*”, através também de rastreamento de *issues* que possuíam as tags “*bug*” e “*featured request*”.

Na etapa de análise dos dados foi usado um modelo de regressão múltipla para descrever a relação entre o conjunto de variáveis explanatórias para predição, como o uso da CI, e respostas para saídas, como a quantidade de *bugs* relatados pelo time. Em seguida os dados passaram por cálculos estatísticos para obterem informações de distribuições, precisões e outras informações para os resultados.

Por fim, Vasilescu, Yue Yu *et al* (2015) concluíram que os projetos *open source* que utilizam *Continuous Integration* processaram mais *pull request*, estão sendo mais aceitas e mescladas ao projetos, e mais estão começando a serem rejeitadas.

Este trabalho assemelha-se pelo o uso de inspeções nos repositórios para a coleta de dados, mas difere em relação a quantidade de projetos, aos tipos de projetos (*open source*), e ao modelo de análise dos dados.

3.3 Implantação de uma ferramenta de Integração Contínua em um Núcleo de Práticas em informática: relato de experiência

Este trabalho é um estudo de caso sobre a implantação de uma ferramenta de CI no NPI, tendo como sua finalidade a avaliação dos relatos e resultados provenientes da introdução de tal ferramenta no modelo de processo do NPI. Neste trabalho, Cabral (2014) realizou um

estudo sobre as ferramentas disponíveis e selecionou a que melhor se adapta ao NPI, levando em consideração o modelo de processo definido do NPI, descrito por Gonçalves, Bezerra, *et al* (2013).

Os seguintes passos foram executados:

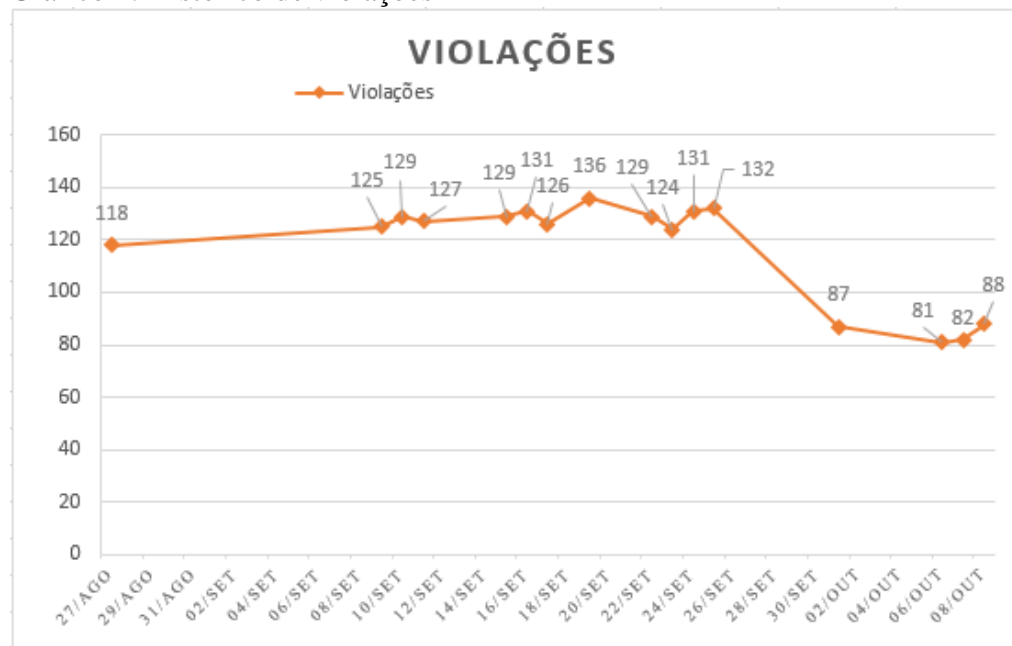
- **Análise das atividades do NPI:** identificar como as atividades são executadas no NPI, a fim de obter informações que auxiliará na decisão de escolha da ferramenta de CI.
- **Pesquisa e seleção da ferramenta de CI:** a partir de uma pesquisa realizada pelo autor, de acordo com os requisitos estabelecidos anteriormente no próprio trabalho, houve a seleção da ferramenta de CI.
- **Absorção do perfil dos estagiários do NPI:** coletar informações sobre relatos de experiência dos usuários em relação a CI e ferramentas de CI. Para isso, foi aplicado um questionário online, de escopo fechado, com todos os participantes do projeto.
- **Implantação da ferramenta de CI:** essa etapa refere-se a implantação da ferramenta de CI escolhida, no segundo passo, em um projeto piloto, de modo a avaliar o impacto da ferramenta no desenvolvimento da equipe. No início da realização desta etapa, será realizado um treinamento com os participantes para entender as funcionalidades e a finalidade da ferramenta no projeto.
- **Coleta de métricas de código:** por fim, foi realizado uma coleta de informações dos códigos que estão sendo submetidos através de uma inspeção contínua, realizada pela ferramenta de análise estática de código *Sonarqube* integrada com a ferramenta *Jenkins*, no processo de *post-build*.

Os resultados dos passos executados foram:

- Identificação do SCRUM como processo de realização das atividades e metodologia ágil aplicada no NPI.
- A ferramenta escolhida foi o *Jenkins*, pelo fato de ser uma ferramenta *open source* e que fornece suporte a vários tipos de linguagens e tecnologias diferentes, e por preencher os requisitos estabelecidos pelo trabalho.

- Fora coletado informações dos 23 participantes que participaram da pesquisa. Em resumo, foi observável um bom conhecimento sobre CI, mas pouco uso em projetos, como também foi identificado que 19 participantes desconhecem ou nunca utilizaram ferramentas de CI.
- Os pontos positivos obtidos com a implantação da ferramenta foram: boa receptividade da equipe; fácil entendimento da ferramenta; coleta de métricas constantes e melhoria na comunicação da equipe. Já os seguintes pontos negativos foram observados: ausência de um gerente de configuração; problemas estruturais e ausência de testes automatizados.
- Foi realizado um balanço entre a quantidade de violações registradas pelo tempo em que a ferramenta de CI estava em execução. Com isso, foi gerado o seguinte gráfico:

Gráfico 1: Histórico de violações



Fonte: Cabral (2014).

Por fim, foi concluído que este trabalho permitiu a criação de uma cultura de CI no NPI, possibilitando que os envolvidos visualizassem o impacto que esta ferramenta exerce no desenvolvimento do software e relatar as lições aprendidas que poderão ser utilizadas em futuras implantações (CABRAL, 2014).

Este trabalho se assemelha como o relatado anteriormente pelo fato de realizar uma análise do impacto da ferramenta de CI no NPI. Porém, o foco do relato apresentado foi a implantação das ferramentas, o que difere deste trabalho, que o foco é a análise do impacto da

CI no ambiente de desenvolvimento do NPI.

No trabalho de Cabral (2014), foi utilizado a ferramenta *Jenkins*, que atualmente foi substituída pela ferramenta de CI e CD do Gitlab, o que torna este trabalho diferente do mesmo, pois, o Gitlab oferece tais ferramentas integradas ao gerenciador de repositórios, onde há pontos positivos e negativos a serem descobertos com essa adoção da ferramenta.

O Quadro 1, é feita uma comparação entre os trabalhos relacionados e este trabalho, onde “X” indica que o trabalho aplica tal atividade e “-” indica que o trabalho não aplica.

Quadro 1: Comparação dos trabalhos

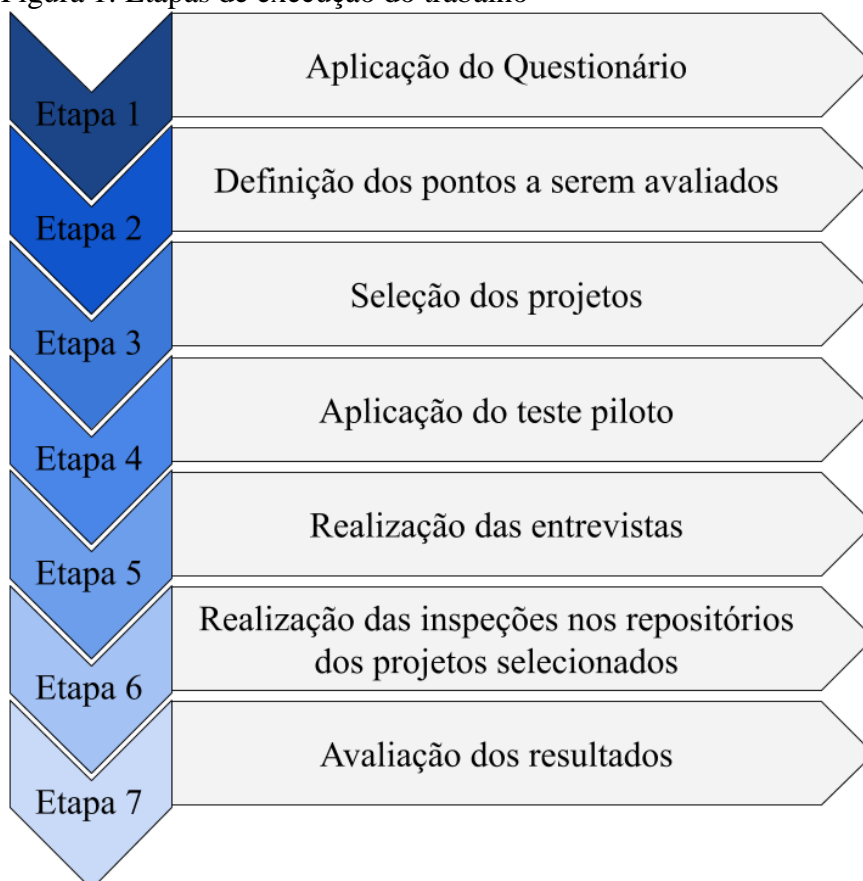
Trabalho	Análise do uso da CI	Realização de Entrevistas	Realização de Inspeções	Objetivo principal do trabalho
Stâhal e Bosch (2013)	X	X	-	Verificar se os benefícios da CI descritos na literatura são vivenciados na indústria de software
Vasilescu, Yue Yu <i>et al</i> (2015)	X	-	X	Analisar os benefícios da CI em projetos <i>open source</i> do Github.
Cabral (2014)	X	X	X	Introduzir uma ferramenta de CI no NPI e avaliar seu impacto.
Este trabalho	X	X	X	Analisar o impacto do uso da CI no ambiente de desenvolvimento do NPI.

Fonte: Autor (2019).

4 METODOLOGIA

Esta seção explana sobre as etapas que foram executadas para a realização deste trabalho. Como mostra a Figura 1, o trabalho foi dividido em 6 etapas.

Figura 1: Etapas de execução do trabalho



Fonte: Autor (2019).

4.1 Aplicação do questionário

Foi apresentado aos supervisores do NPI, pela plataforma *Slack*, um questionário online que teve como objetivo identificar os pontos a serem avaliados pela CI no NPI, de acordo com a resposta destes supervisores.

O questionário foi dividido em 3 partes:

- Identificação do supervisor,
- Pontos importantes, para o NPI, a serem avaliados pela CI e
- Sugestões.

A primeira parte do questionário buscou obter informações sobre o nome, função no NPI e o tempo de trabalho no NPI de cada supervisor. Essas perguntas identificaram os cargos que são executados no NPI e a experiência que cada supervisor possui no núcleo. O nome requisitado serviu apenas para identificação de cada supervisor, esta informação não foi utilizada em nenhuma etapa do trabalho.

A segunda parte do questionário definiu os pontos mais importantes, para o NPI, a serem avaliados pela CI. Foi solicitado aos participantes a definição de um nível de prioridade (que possui um peso de avaliação que vai de 1 a 5) de cada ponto, sendo eles: Muito baixa (peso 1), Baixa (peso 2), Média (peso 3), Alta (peso 4) e Muito alta (peso 5).

A terceira parte do questionário tratava-se de um campo opcional, onde foi questionado ao respondente se o mesmo tinha alguma sugestão ou comentário em relação a execução do trabalho. Cada comentário e/ou sugestão foram considerados e avaliados para a execução das próximas etapas do trabalho.

4.2 Seleção dos pontos a serem avaliados

Após a aplicação do questionário, foi realizado a seleção dos pontos a serem avaliados pela CI, pontos esses que se referem a segunda parte do questionário.

Como foi aplicado um peso para cada resposta, de cada ponto, foi possível calcular a média dos resultados. Foi selecionado os pontos que obtiveram uma média maior, ou igual, a 4 (quatro), que era a classificação de importância “Alta”, como é apresentado nos resultados, na Seção 5.1.

Alguns pontos foram sugeridos no campo de sugestões e comentários do questionário. Alguns destes pontos sugeridos já haviam sido apresentados no próprio questionário, então foram considerados somente os pontos sugeridos que não estavam entre os pontos do questionário.

4.3 Seleção dos projetos

Foram selecionados os projetos que tinham em seu repositório:

- Integração Contínua;
- Uma equipe de desenvolvimento organizada que realize modificações frequentes no projeto;
- Uma política de gerência de configuração para a padronização das entregas (abertura de uma *merge request* para resolução de uma *issue*, por exemplo).

Não é importante que a equipe esteja familiarizada com a CI, bastando que a mesma verifique apenas se suas alterações foram integradas corretamente, ou geraram um *build* falho.

4.4 Aplicação do teste piloto

O teste piloto é descrito como sendo uma atividade necessária para avaliar se o participante compreendeu corretamente o que foi apresentado, se o tempo de execução do teste é previsto e viável e se através das tarefas demandadas é possível avaliar os critérios desejados (PRATES e BARBOSA, 2003).

Foi realizado um teste piloto antes da execução das entrevistas, para verificar se as perguntas do roteiro piloto (APÊNDICE A) estão compreensíveis e estão atendendo a avaliação dos pontos desejados a serem avaliados. Foi avaliado também se seria necessário a alteração, ou remoção, de alguma pergunta que não ficou clara ao participante, ou que o mesmo não soubesse responder, afim de definir um roteiro padrão a todos os participantes, separados por funcionalidade no NPI (supervisor ou estagiário).

4.5 Realização das entrevistas

As entrevistas foram realizadas seguindo um roteiro pré-elaborado (APENDICE B e APENDICE C) de acordo com a função do entrevistado no NPI, estagiário ou supervisor.

Todas as entrevistas foram realizadas, e gravadas, individualmente e tiveram uma duração, em média, de 9 minutos para os estagiários e 20 minutos para os supervisores. Antes de realizar a entrevista, cada participante assinou o TCLE (APÊNDICE E), para que o mesmo esteja ciente de seus direitos e aos termos do pesquisador (Autor).

As respostas semiabertas foram classificadas como positiva e negativa, após a realização de uma análise de padrões nas respostas. São semiabertas as perguntas da seção referente aos pontos a serem avaliados, perguntas 7 a 12 do roteiro de supervisores e de 8 a 12 do roteiro de estagiários.

As respostas fechadas foram mensuradas, em formato numérico, em relação ao total de candidatos, exemplo de questões fechadas são as perguntas de número 2, 3, 4, 6, 7, 13 do roteiro dos estagiários e 2, 3, 5 e 6 do roteiro dos supervisores.

As respostas abertas foram avaliadas utilizando análise qualitativa simples, onde se busca por padrões a partir da familiarização dos dados que em seguida passam por uma confirmação e reconfirmação, para se certificar que as impressões iniciais não estão

influenciando a análise, gerando, por fim, um conjunto de resultados que são utilizados na análise para o objetivo final da entrevista (YVONNE, SHARP e PREECE, 2011). São perguntas abertas as questões 5 e 14 do roteiro de estagiários, e as questões 4 e 13, do roteiro dos supervisores.

4.6 Realização das inspeções nos repositórios dos projetos selecionados

As inspeções foram realizadas nos repositórios dos projetos selecionados, com a finalidade de avaliar, quantitativamente, a diferença do atual ambiente de desenvolvimento, com a Integração Contínua, e o antigo ambiente de desenvolvimento, sem a Integração Contínua, de acordo com os atributos listados no Quadro 2.

Quadro 1: Atributos das inspeções

Nome do atributo	Significado
p_name	Nome do projeto
n_commits	Quantidade de <i>commits</i> realizados
n_mergerequest_open	Quantidade de <i>mergerequest</i> abertos
n_mergerequest_merged	Quantidade de <i>mergerequest</i> aceitos
n_mergerequest_rejected	Quantidade de <i>mergerequest</i> rejeitados
n_issues_opened	Quantidade de <i>issues</i> abertas
n_issues_closed	Quantidade de <i>issues</i> fechadas
n_pipelines_success	Quantidades de <i>pipelanes</i> que foram aceitas pela CI
n_pipelines_failed	Quantidades de <i>pipelanes</i> que foram recusadas pela CI

Fonte: Autor (2019).

As inspeções foram realizadas com o auxílio da API *python-gitlab*¹, que foi utilizada para realizar requisições sobre as informações desejadas nos projetos, utilizando a linguagem de programação *Python*.

Foram armazenadas as informações (Quadro 2) de cada projeto em 2 períodos diferentes, antes e depois da utilização da CI. Avaliou-se um intervalo de 90 dias de cada período. Foi selecionado dois conjunto de datas, A e B, como é apresentado no Quadro 3.

Quadro 2: Conjunto de datas utilizados nas inspeções.

Conjunto	Período antes da CI		Período depois da CI	
	De:	Até:	De:	Até:
Conjunto A	01/04/2019 às 00:00	30/06/2019 às 23:59	01/08/2019 às 00:00	30/11/2019 às 23:59
Conjunto B	20/08/2018 às 00:00	20/12/2018 às 23:59	01/08/2019 às 00:00	30/11/2019 às 23:59

Fonte: Autor (2019).

As datas apresentadas no Conjunto A representam, respectivamente, as datas de

¹ Informações sobre a API: <https://python-gitlab.readthedocs.io/en/stable/index.html>.

funcionamento do NPI, após um mês do início de trabalho em cada semestre do ano. Essas datas foram escolhidas para os projetos que tiveram a CI adicionada em seus repositórios no meio do ano de 2019, tendo um mês após o início do funcionamento do NPI para ter um tempo de adaptação da equipe com a ferramenta.

As datas apresentadas no Conjunto B representam as datas do final do semestre do ano de 2018 e 2019, pois estas datas foram selecionadas para os projetos que adicionaram a CI em seus repositórios no início de 2019.

Após armazenar as informações dos dois períodos, antes da CI e de depois da CI, foi realizado uma comparação dos dados, para verificar o impacto que a Integração Contínua teve nos projetos.

Por fim, foi feito uma análise das *pipelines* dos projetos para identificar quais os projetos que mais estão adaptados com a CI.

4.7 Avaliação dos resultados

A última etapa da pesquisa foi a avaliação do que foi obtido com as inspeções e entrevistas. Nesta etapa, foi feito uma comparação dos dados do antigo ambiente de desenvolvimento do NPI, sem CI, com os dados do ambiente de desenvolvimento com CI.

Com os dados analisados será possível definir um indicador do impacto do uso da CI no ambiente de desenvolvimento do NPI.

5 RESULTADOS

Neste capítulo é apresentado os resultados obtidos com os questionários, as entrevistas e a inspeção.

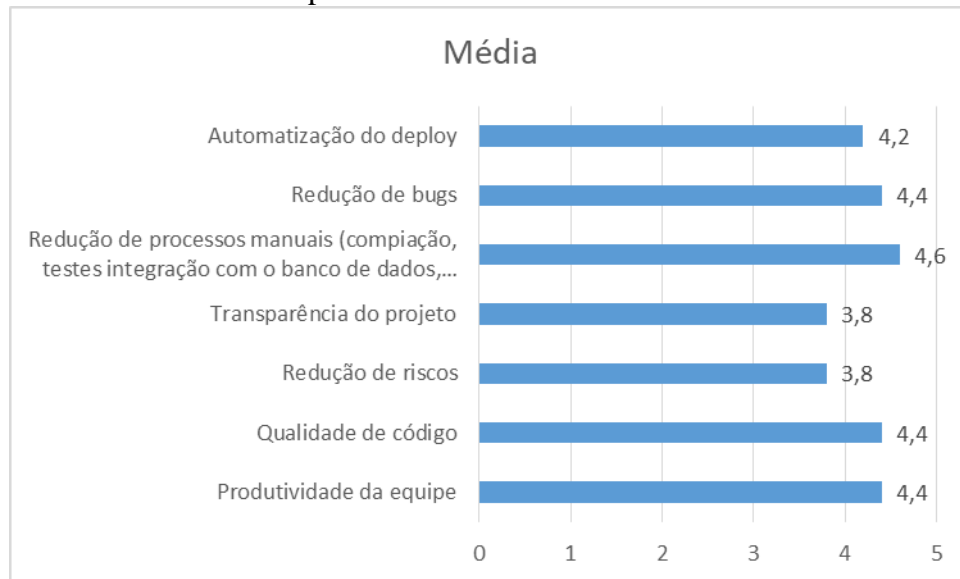
5.1 Resultados do Questionário

A princípio foram convidados 7 supervisores do NPI para responder o questionário (APENDICE D). Dos 7 supervisores 5 responderam ao questionário.

De acordo com as respostas temos que:

- As funções executadas atualmente pelos supervisores são:
 - Analista de Tecnologia da Informação,
 - Gerente de projetos,
 - Orientador de estágio,
 - Coordenador.
- Atualmente, o tempo de serviço de cada supervisor no NPI varia de 1 ano e 6 meses à 5 anos.
- Seguindo a média de notas dadas por cada participante aos pontos de avaliação proposto (Gráfico 2), temos por ordem de prioridade:
 - Produtividade da equipe,
 - Qualidade de código,
 - Redução de processos manuais,
 - Redução de bugs,
 - Automatização do *deploy*.

Gráfico 2: Média dos pontos a serem avaliados



Fonte: Autor (2019).

Durante o questionário foi possível obter as seguintes sugestões:

- A comunicação era um ponto importante a ser avaliado e,
- O NPI precisa, de modo geral, de melhorias e maiores automatizações em seus processos. Foi dito também que alguns projetos já estão bem automatizados, mas outros precisam melhorar nesse aspecto.

Com o questionário foi possível definir quais os pontos a serem avaliados durante a entrevista e a inspeção, que são:

- Automatização do *deploy*
- Comunicação da equipe
- Produtividade da equipe
- Qualidade de Código
- Redução de *bugs*
- Redução de processos manuais

5.2 Resultados do roteiro piloto

O roteiro piloto (APENDICE A) foi aplicado com 3 participantes (estagiários) e notou-se que os participantes não souberam responder a questão 13, devido ao fato de que os

mesmos não ficavam responsáveis por essa tarefa, ou por não participarem do processo de *deploy*.

O roteiro final dos estagiários (APÊNDICE B) foi aplicado aos outros participantes tendo a questão de número 13 retirada.

No roteiro final dos supervisores (APÊNDICE C) foi retirado as perguntas referentes a experiência passada com a Integração Contínua e a questão 14, que perguntava se o participante recomendaria essa ferramenta. E questões como a 2 e 6, tiveram uma mudança de sujeito, que antes recebia a ação e agora aplica a ação, devido ao fato de que as perguntas estão sendo feitas aos supervisores.

5.3 Projetos selecionados

O Quadro 4 apresenta os projetos selecionados e suas informações.

Quadro 3: Projetos selecionados

Nome do projeto	Data de criação *	Quantidade de membros ^{2*}	Informações do projeto
Proficiência	20/08/2018	14	Sistema de controle do exame de proficiência leitora das casas de cultura da UFC.
Gestão de Riscos	12/03/2018	14	Sistema para o controle de gestão de riscos da UFC.
Gestão por Competências	12/03/2018	17	Sistema de gestão de competências dos servidores da UFC ocupam cargos de gestão.
Dados Abertos	12/12/2017	15	Projeto de consultas e gráficos sobre os dados abertos de pagamentos e remunerações do Poder Executivo Federal.
Prontuário Eletrônico Odontológico	11/12/2017	13	Sistema de prontuário eletrônico das clínicas de odontologia da UFC

Fonte: Autor (2019).

(*) Dados obtidos com a API *python-gitlab*.

Os projetos selecionados possuem Integração Contínua em seus repositórios. A ferramenta de Integração Contínua utilizada é a GitlabCI. Os projetos possuem, no momento, as fases de *build*, análise estática e verificação de qualidade.

Os projetos possuem uma equipe de desenvolvimento organizada que realizam modificações frequentes no projeto. Os estagiários trabalham 10 horas semanais, divididas em

² A quantidade de membros obtidos soma todos os membros do projeto e suas funções, supervisor, estagiários e seus mantenedores (como o SonarQuebe por exemplo).

3 dias da semana: segunda, terça e quarta, com exceção dos bolsistas que trabalham 12 horas semanais. As atividades são executadas em turnos diferentes, manhã e tarde, de acordo com o curso do mesmo.

Os projetos possuem uma política de gerência de configuração para a padronização das entregas definidas. Cada projeto possui um *board* de atividades, e a partir disso, os estagiários realizam uma atividade, testam, e realizam um *merge-request*, que será avaliado pela CI e, se obtiver êxito em todos os jobs, é avaliado pelos supervisores e adicionado a *branch* principal.

Todos esses projetos estão disponíveis no repositório do Gitlab do NPI³, no qual utiliza o GitlabLDAP para a autorização do integrantes do NPI.

5.4 Resultados das entrevistas

Foi obtido um total de 11 entrevistas, sendo 7 estagiários e 4 supervisores. Cada uma das entrevistas foram gravadas e avaliadas após a entrevista. Essa diferença de tempo entre os estagiários e os supervisores se deve ao fato de que os supervisores possuíam mais comentários em cada resposta, por possuírem mais experiência com os projetos e perceberem os impactos presenciados por eles e pelos alunos.

5.4.1 Estagiários

Foi realizado 7 entrevistas com os estagiários de diferentes projetos que possuíam a Integração Contínua. Foi possível ter a participação de 4 participantes do projeto Gestão de riscos, 2 participantes do projeto Proficiência e 1 participante do projeto Prontuário Odontológico.

5.4.1.1 Experiência e Adaptação

Os estagiários possuem entre 8 e 9 meses de trabalho nesses projetos, sendo 5 participantes com 9 meses e 2 participantes com 8 meses. Esta diferença de 1 mês se deve ao fato de que há estagiários bolsistas, e estes executam suas atividades no mês de julho, um mês a mais que do que é executado normalmente;

A maioria dos estagiários nunca utilizaram Integração Contínua em seus projetos, somente 1 participante diz já ter usado, e explicou que teve dificuldades no começo, mas depois conseguiu executar normalmente. A ferramenta utilizada pelo mesmo foi o Gitlab, isso fez com que a adaptação fosse mais rápida no NPI;

³ Sitio do repositório: <http://git.quixada.ufc.br/>.

A experiência deles foi bastante positiva. Em seus relatos foi dito que melhorou muito a qualidade do código e que houve uma “limpeza” no código, referindo-se a redução de *bugs* e estilo de código. Foi dito também que os projetos estão mais “visíveis”, no que se remete a saúde do projeto, e que está sendo mais rápido e fácil corrigir os erros submetidos, sendo possível “prever os erros”.

Eles obtiveram uma rápida adaptação com a ferramenta.

Eles não receberam um treinamento/orientação sobre a ferramenta, apenas foi citado que eles começariam a usar e que os códigos submetidos por eles iam ser avaliados pela Integração Contínua e monitorados pelo *Sonarqube*.

5.4.1.2 Pontos a serem avaliados

O Quadro 5 apresenta o impacto em cada ponto, identificando se ele sofreu um impacto positivo, negativo ou que não houve impacto.

Quadro 4: Impacto dos pontos a serem avaliados observados pelos estagiários

Ponto a ser avaliado	Impacto positivo	Impacto negativo	Não houve impacto
Produtividade da equipe	X		
Qualidade do código	X		
Redução dos processos manuais	X		
Redução de <i>bugs</i>	X		
Comunicação			X

Fonte: Autor (2019).

Segundo os entrevistados, os que afirmaram que houve um impacto positivo, a **Produtividade da equipe** aumentou pois está sendo possível ter um *feedback* mais rápido do código entregue, e isso faz com que seja possível realizar alterações mais rápido no código e submete-lo logo, se necessário. No entanto, dois entrevistados que afirmam que não houve um impacto na produtividade, não sabem ao certo se a produtividade aumentou por conta da Integração Contínua ou por causa da maturidade da equipe, que aumentou ao passar do tempo.

Todos os entrevistados afirmaram que a **Qualidade do código** o melhorou bastante, devido ao fato de que o código está sempre sendo analisado e avaliado a todo momento e de forma rápida, e com isso é possível identificar irregularidades no código (erros, *bugs*, *code smells*, vulnerabilidades e etc) e consertar imediatamente. Alguns afirmam que atualmente está sendo possível “prever” os erros antes de enviar, indicando que houve um aprendizado da equipe em relação a identificação e correção de irregularidades no código.

Em relação a **Redução dos processos manuais**, a maioria dos entrevistados afirmaram que houve uma redução no processo de avaliação, realizado pelos supervisores, que antes avaliavam as irregularidades do código e ainda tinham que testar a funcionalidade para verificar se está executando o que foi solicitado. Agora os supervisores avaliam somente a execução da funcionalidade e verificar se está de acordo com o que foi solicitado. Apenas um entrevistado afirmou que não percebeu nenhum impacto neste ponto.

Com o monitoramento do *Sonarqube* vinculado com a Integração Contínua no repositório dos projetos é possível ter um monitoramento dos bugs do projeto, além de ter uma classificação dos mesmos, afirmam os entrevistados. Assim sendo possível identificar um *bug* e resolve-lo. Impactando positivamente a **Redução de bugs**.

A maioria dos entrevistados comentaram que não perceberam nenhum impacto em relação à **Comunicação**. Alguns disseram que na teoria faria sentido, mas não presenciaram isso na prática.

5.4.1.3 Sugestões

Todos os entrevistados afirmaram que usaria e/ou recomendaria em futuros projetos.

Em geral, os participantes sugeriram a realização de um treinamento/orientação da ferramenta de forma que pudessem aproveitar o máximo da mesma.

5.4.2 Supervisores

Foi realizado 4 entrevistas com os supervisores de diferentes projetos que possuíam a Integração Contínua. Foi possível ter a participação de 2 supervisores dos projeto Gestão de riscos e Gestão por Competências, 1 supervisor do projeto Proficiência e Prontuário Odontológico e 1 supervisor do projeto Dados Abertos.

5.4.2.1 Experiência e adaptação

Os supervisores possuem entre 1 e 2 anos de trabalho nesses projetos, sendo 1 com 1 ano e 3 com 2 anos.

A experiência que os supervisores obtiveram com a adoção da Integração Contínua foi bastante positiva. A melhora na automatização do deploy, na qualidade do código e no monitoramento foram os pontos mais presenciados no ambiente de desenvolvimento do NPI, segundo os supervisores. Houve uma melhora na segurança de entregar um projeto de

qualidade e uma facilidade de gerar uma versão nova do sistema, segundo relatos dos entrevistados.

Segundo os supervisores, a equipe teve uma rápida adaptação com o uso da Integração Contínua.

Os supervisores afirmaram que não houve um treinamento, apenas uma orientação sobre as ferramentas que começariam a avaliar e monitorar os códigos submetidos e como as mesmas entrariam no processo de desenvolvimento da equipe.

5.4.2.2 Pontos a serem avaliados

O Quadro 6 apresenta o impacto em cada ponto, identificando se ele sofreu um impacto positivo, negativo ou que não houve impacto.

Quadro 5: Impacto dos pontos a serem avaliados observado pelos supervisores

Ponto a ser avaliado	Impacto positivo	Impacto negativo	Não houve impacto
Produtividade da equipe			X
Qualidade do código	X		
Redução dos processos manuais	X		
Redução de <i>bugs</i>	X		
Comunicação	X		
Automatização do Deploy	X		

Fonte: Autor (2019).

Alguns dos entrevistados comentam que o impacto na **Produtividade da equipe** tornara-se positivo a longo prazo, com a adaptação das futuras equipes aos projetos, devido a melhora na qualidade do código, que implica em um código mais manutenível.

Segundo os entrevistados, a melhoria na **Qualidade do código** foi bem perceptível, sendo observado até uma maior preocupação dos estagiários ao entregar um código de qualidade.

Os maiores benefícios obtidos, em relação à **Redução dos processos manuais**, foram presenciados na avaliação dos códigos submetidos pelos estagiários e na implantação do sistema.

Segundo os entrevistados a **Redução de bugs** deve-se ao fato de que os estagiários estão recebendo um *feedback* rápido no seus códigos submetidos, além de monitorarem os *bugs* que o sistema possui através do *Sonarqube*.

Em relação à **Comunicação**, foi presenciado uma compartilhamento de resolução de

problemas entre os estagiários, além de uma transparência maior do projeto em relação aos supervisores.

Por fim, na **Automatização do Deploy**, para realizar a implantação dos sistemas era necessário a utilização de ferramentas para o empacotamento dos *web archive's (war's)*, além de ser necessário uma solicitação a outro setor dentro do campus para subir o sistema no servidor. Atualmente, houve uma adaptação para que houvesse a remoção da interferência de outro setor no processo de implantação. O Gitlab quem é o novo responsável por subir o sistema, através do repositório do projeto é gerado o *war*, gerado por um *job* da Integração Contínua, que é instalado no servidor ao clicar no botão *run deploy*.

5.4.2.3 Sugestões

Foi sugerido que houvesse um melhor aproveitamento da Integração Contínua, fosse adicionado mais testes, e que houvesse uma melhoria na análise estática e no feedback que ela gera para o Gitlab, que por algumas vezes gera relatórios grandes de problemas como erros de estilo de código e *code smells*. Foi relatado pelos supervisores que as vezes o resumo dos erros feito pelo Gitlab não são muito explicativos, sendo necessário acessar a ferramenta do *Sonarqube* para obter mais detalhes.

A adição da etapa de deploy em produção e homologação junto a *pipeline* atual.

A utilização de ferramentas e *plug-ins* que auxiliem a Integração Contínua no processo de análise de código, como a utilização do *SonarLint* nas IDE's de cada usuário para que houvesse uma análise do código antes da submissão do mesmo ao repositório. Foi sugerido também a utilização de um ambiente *Docker* para simular o ambiente de produção na máquina dos estagiários.

A inserção da Integração Contínua em todos os projetos e uma orientação/treinamento às equipes desde o início do estágio.

5.5 Inspeção

Os projetos foram analisados com 2 conjuntos de datas diferentes, como foi explicado na Seção 4.6, pois há projetos que adicionaram a CI em seus repositórios no início do ano de 2019 e outros que adicionaram na metade do ano de 2019.

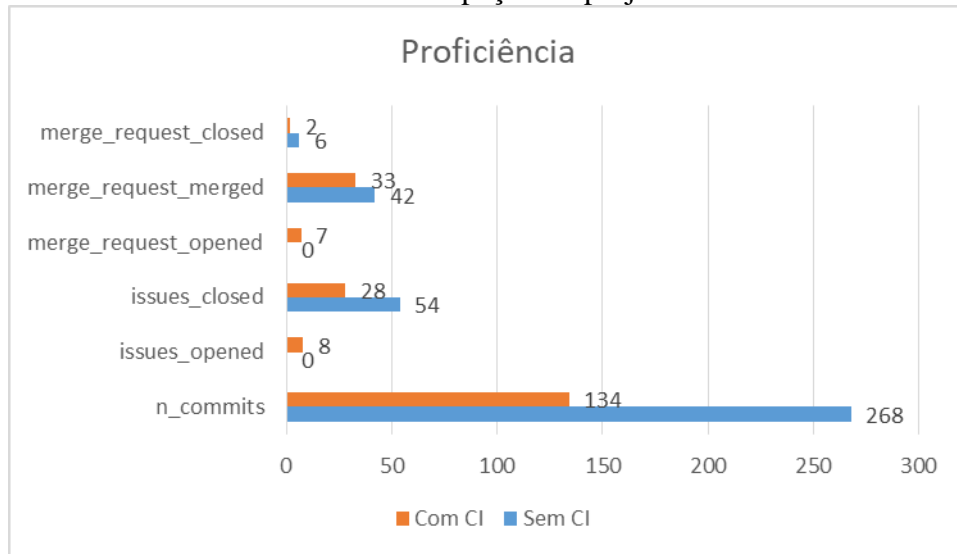
Os projetos que utilizaram o Conjunto A foram: Gestão por Competências, Gestão de Riscos e Dados Abertos. Os projetos que utilizaram o Conjunto B foram os projetos

Proficiência e Prontuário Eletrônico Odontológico.

Como foi citado por Vasilescu, Yue Yu et al (2015), a quantidade de *mergerequests* aceitas por mês é um indicador de produtividade da equipe, a quantidade de *issues* resolvidas indica maior resolução de problemas e a quantidade de *commits* submetidos indica uma maior frequência de alterações da equipe. Essas informações foram utilizadas para identificar os impactos apresentados nas inspeções de cada projeto.

5.5.1 Proficiência

Gráfico 3: Dados obtidos com a inspeção do projeto Proficiência

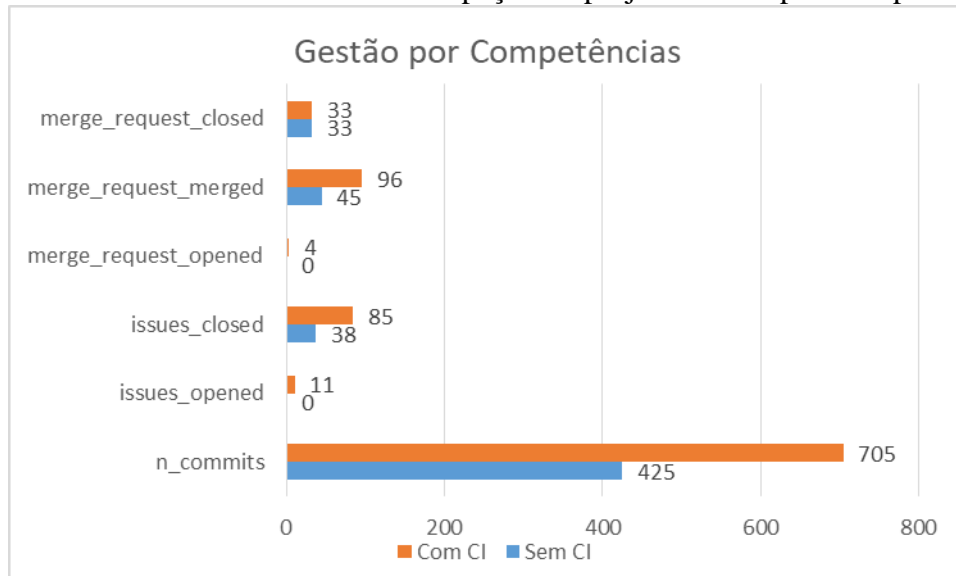


Fonte: Autor (2019).

A produtividade do projeto foi maior antes da CI devido ao fato de que as datas selecionadas coincidem com o período de criação do projeto, onde a produtividade é mais presenciada, como podemos ver pelos dados (Gráfico 3). No entanto, mesmo considerando essa hipótese, temos uma quantidade de *mergerequests* com status *merged* aproximado do que foi obtido no início do projeto, o que indica que, paulatinamente haverá um impacto positivo da CI referente a produtividade.

5.5.2 Gestão por Competências

Gráfico 4: Dados obtidos com a inspeção do projeto Gestão por Competências

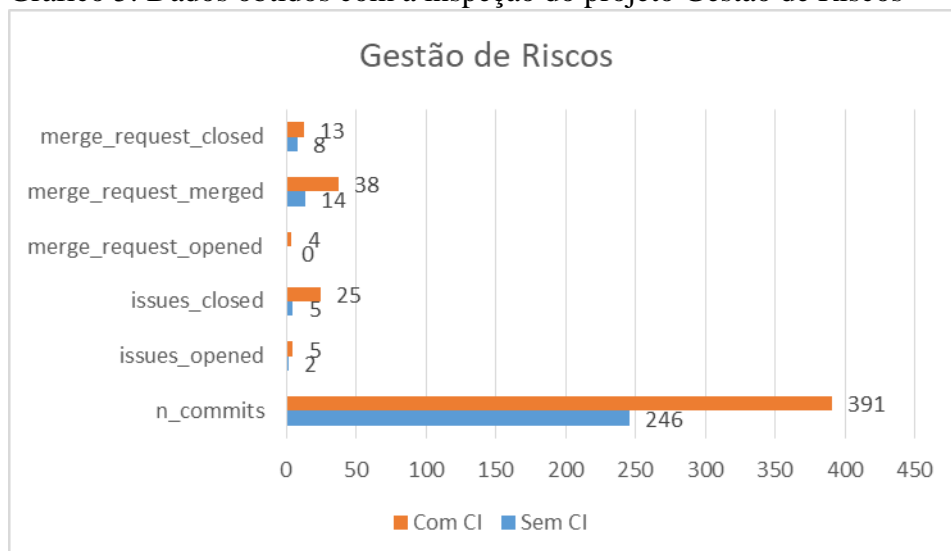


Fonte: Autor (2019).

Nesse projeto, podemos verificar (Gráfico 4) que houve um aumento significativo nos atributos *n_commits*, *issues_closed* e *merge_request_merged*, o que indica que houve um aumento na frequência de submissões, resoluções de problemas e entregas de atividades, o que indica que a CI impactou positivamente na produtividade deste projeto.

5.5.3 Gestão de Riscos

Gráfico 5: Dados obtidos com a inspeção do projeto Gestão de Riscos

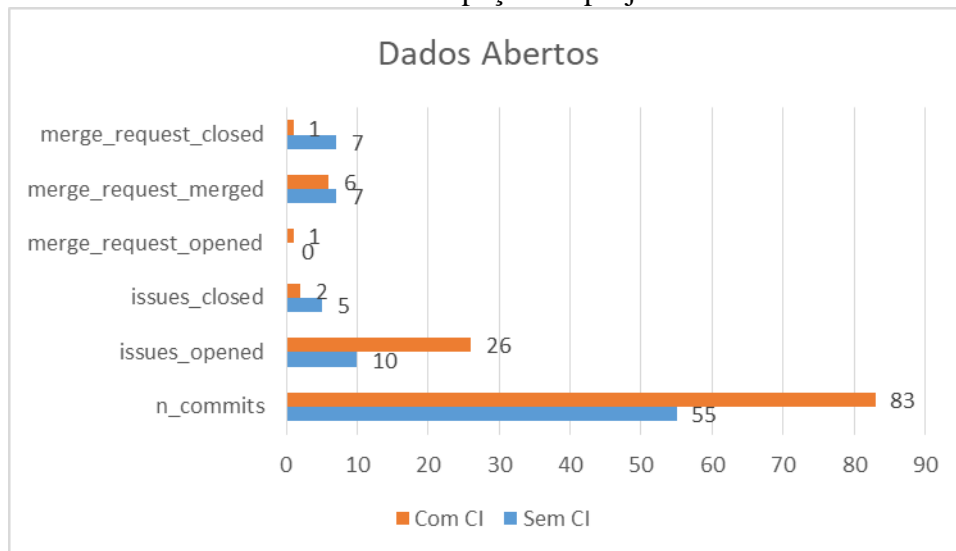


Fonte: Autor (2019).

Como podemos observar no gráfico acima (Gráfico 5), este projeto também teve um impacto positivo em relação a produtividade, pois foi possível observar que a quantidade de *mergerequests* aceitos aumentou, junto com a resolução de *issues* e também um aumento significativo na submissão de alterações.

5.5.4 Dados Abertos

Gráfico 6: Dados obtidos com a inspeção do projeto Dados Abertos

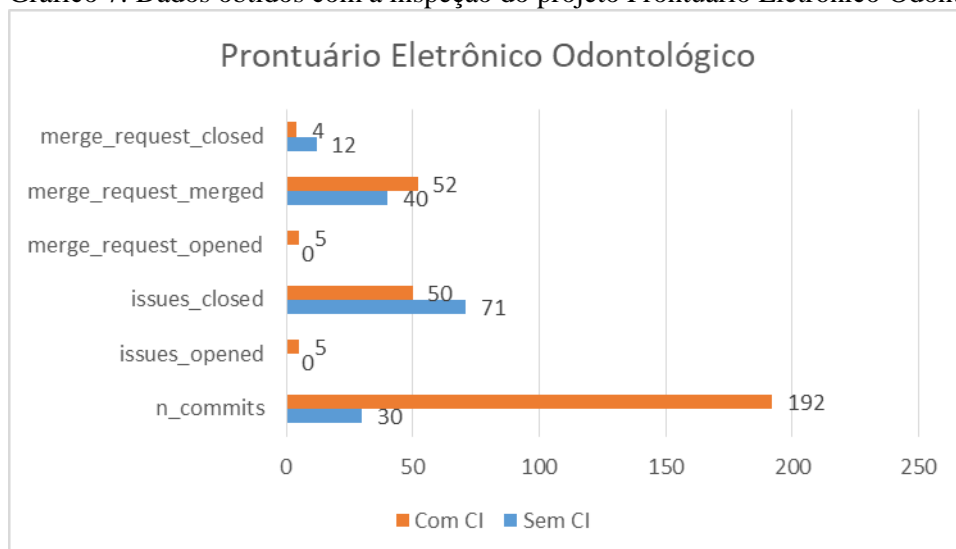


Fonte: Autor (2019).

Este projeto teve um aumento maior na submissão de alterações, no entanto não houve um aumento na quantidade de *mergerequests* e na quantidade de *issues* resolvidas. A CI não impactou a produtividade deste projeto.

5.5.5 Prontuário Eletrônico Odontológico

Gráfico 7: Dados obtidos com a inspeção do projeto Prontuário Eletrônico Odontológico



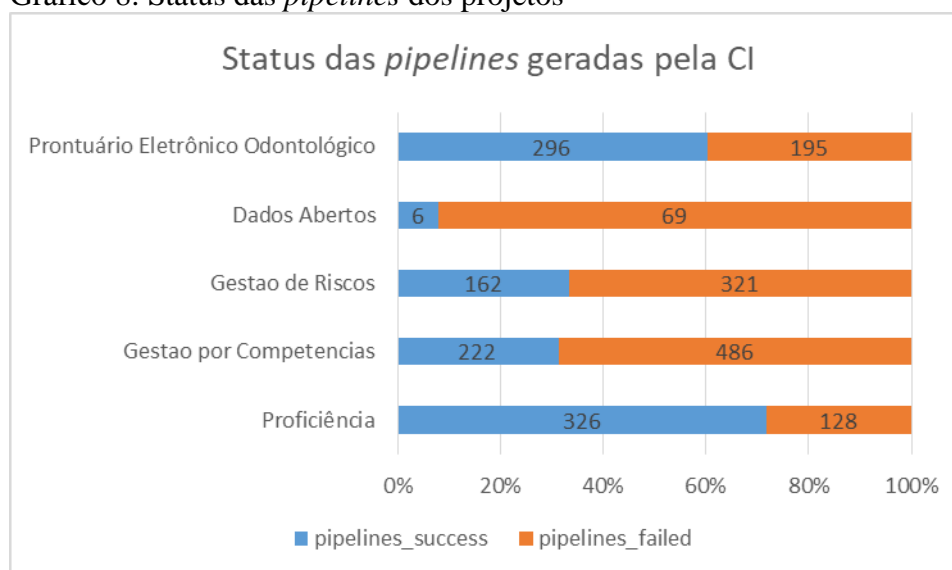
Fonte: Autor (2019).

Como é apresentado acima (Gráfico 7), a quantidade de *mergerequests* e *commits* aumentaram, mas não houve um ganho em relação a resolução de *issues*. O impacto que a CI teve neste projeto foi positivo, pois, além do aumento citado anteriormente, foi possível identificar que mais *mergerequest* foram criadas, algumas não foram aceitas (*merge_request_closed*) e outras ainda precisam ser avaliadas (*merge_request_opened*), isso

indica que a equipe realizou mais submissões a serem avaliadas.

5.5.6 Avaliação das pipelines dos projetos

Gráfico 8: Status das *pipelines* dos projetos



Fonte: Autor (2019).

Como podemos observar (Gráfico 8), os projetos que mais obtiveram *pipelines* com êxito foram os projetos que possuíam a CI por mais tempo (Prontuário Eletrônico Odontológico e Proficiência), o que identifica uma maior adaptação com a ferramenta e uma aprendizagem da equipe.

O projeto que obteve o maior número de *pipelines* com falhas foi o projeto Dados Abertos, o que indica que a equipe ainda está passando por um processo de adaptação e aprendizado da CI adicionada ao repositório do projeto.

Os projetos Gestão de Riscos e Gestão por Competências obtiveram resultados semelhantes, também indicando que o processo de adaptação e aprendizado estão parecidos, mesmo sendo uma quantidade baixa eles obtiveram resultados melhores que o projeto Dados Abertos.

5.6 Avaliação dos resultados

Com os resultados das entrevistas e das inspeções foi possível obter um *feedback* positivo em relação a adição da CI nos projetos do NPI. Os pontos que a CI mais impactou positivamente foram:

- Qualidade de código,
- Redução de *bugs*,

- Redução de Processos Manuais,
- Automatização do *deploy*.

Alguns projetos obtiveram um aumento na produtividade, o que indica que a CI convencionalmente afeta neste ponto seja direta ou indiretamente, no entanto será possível perceber mais sobre este impacto com o passar do tempo, já que o tempo analisado (com a CI) foi curto.

É importante ressaltar que os projetos Dados Abertos e Prontuário Eletrônico Odontológico possuem uma quantidade menor de desenvolvedores, o que pode afetar nos resultados das inspeções em relação a produtividade.

O *feedback* rápido dos códigos submetidos gerou uma rapidez nas correções de alterações que foram rejeitados pela CI, como também gerou uma rapidez ao analisar os códigos que foram aceitos pela CI, para a aceitação de *merge requests*.

Foi observado uma evolução da equipe em relação a preocupação de entregar um código de qualidade, como foi dito nos resultados das entrevistas, eles estão “prevendo” seus erros antes de submeter alguma alteração.

Foi observado também que a CI serviu de aprendizado para os alunos, ensinando tipos de erros, *bugs*, boas práticas e, acima de tudo, uma preocupação com a importância na entrega de um produto de qualidade. Esta observação é importante pois vemos que algo que é ensinado nas disciplinas, que acaba sendo esquecido com a ausência da prática, está sendo cobrado a cada submissão, assim temos um ambiente formado por alunos, o que indica que o NPI está preparando melhor os profissionais que passam por ele.

Por fim, é importante mencionar que todos os estagiários entrevistados afirmaram que recomendariam/usariam a Integração Contínua em futuros projetos, o que aponta uma aceitação do uso ferramenta, outros afirmam até que atualmente está sendo imprescindível a utilização de uma ferramenta de Integração Contínua para a garantia de uma manutenibilidade, organização e monitoramento de um projeto.

6 CONCLUSÃO

Foi possível concluir que com as entrevistas e as inspeções realizadas obteve-se uma análise qualitativa e quantitativa dos dados, respectivamente, sendo possível ter uma cobertura maior de informações acerca do objetivo do trabalho, que é identificar o impacto da CI no

ambiente de desenvolvimento do NPI.

Com os resultados obtidos foi possível observar que o objetivo apresentado no capítulo introdutório, avaliar o impacto da Integração Contínua no ambiente de desenvolvimento do NPI, foi alcançado.

Foi concluído também que os resultados observados estavam de acordo com o que era a ser esperado, segundo a literatura, sendo assim, esta pesquisa acaba sendo também mais um caso de uso a ser explorado.

Esta pesquisa também concluiu que a implantação desta ferramenta em um ambiente de desenvolvimento pequeno, e com integrantes que não possuem uma experiência de mercado considerável, também traz resultados positivos.

É importante ressaltar que os alunos obtiveram um aprendizado a mais trabalhando com a CI, isto acaba reforçando que o objetivo do NPI, citado por Gonçalves, Bezerra, *et al* (2013), está sendo alcançado, que é a formação de alunos em profissionais qualificados sendo treinados em um ambiente de desenvolvimento similar ao de uma fábrica de *software*.

Por fim, foi concluído também que os integrantes do NPI tiveram uma adaptação rápida da ferramenta e uma experiência bastante positiva, o que indica a aceitação da utilização da Integração Contínua no processo de desenvolvimento de *software* como também uma harmonia dos integrantes ao trabalharem com a ferramenta.

6.1 Pontos que afetam a conclusão da pesquisa

O fato de que o tempo de utilização da CI avaliado foi curto, por ser uma adoção recente, gera poucos dados para garantir o real impacto da ferramenta no ambiente de desenvolvimento.

O aumento da maturidade da equipe em desenvolvimento, que ao passar do tempo seu desempenho evolui, aumenta a produtividade e a qualidade de código gerado pela equipe, interferindo nos resultados das inspeções, pois não há garantia se a produção aumentou por conta da CI ou da evolução dos participantes.

6.2 Trabalhos futuros

Atualmente alguns dos projetos do NPI não contam com testes, o que seria muito importante para garantir a qualidade do projeto com a CI. É importante que os projetos possuam *Jobs* para a execução desses testes. Como trabalho futuro, é desejável reexecutar as

inspeções e as entrevistas com os testes

Um ponto a melhorar seria a inspeção, realizando uma análise mais profunda dos dados, como por exemplo categorizar as *issues* para identificar se é uma ocorrência de *bug*, correção de erro ou uma melhoria no código, filtrando por palavras chaves (“*hotfix*”, “*bug encontrado*”, “*correção de um erro*” e etc), para mensurar a qualidade do projeto a partir de quantas *issues* de *bugs*, *erros* ou *improvement* foram resolvidas por mês (VASILESCU, YUE YU, *et al.*, 2015).

Outro ponto de evolução é a realização dessas análises em mais projetos do NPI, avaliando os projetos com períodos maiores, buscando garantir resultados mais concretos.

REFERÊNCIAS

- BAARSEN, J. Van. **GitLab cookbook**: Quick answers to common problems. [S.l.]: Packt Publishing Ltd, 2014. 172 p. Disponível em: <https://books.google.com.br/books?id=ANIGBgAAQBAJ>. ISBN: 9781783986859. Acesso em: 12 Jun. 2019.
- CABRAL, Guilherme Tabosa. **Implantação de uma ferramenta de integração contínua em um núcleo de práticas em informática**: relato de experiência, Quixadá, Novembro 2014. 37. Universidade Federal do Ceará.
- DUVALL, P. M.; MATYAS, S.; GLOVER, A. **Continuous integration**: improving software quality and reducing risk. Boston: Pearson Education, 2007. ISBN 9780321630148. Disponível em: <https://books.google.com.br/books?id=PV9qfEdv9LOC>. Acesso em: 10 Abr. 2019.
- FOWLER, M. **Continuous integration**. 2006. Disponível em: <https://martinfowler.com/articles/continuousIntegration.html>. Acesso em: 19 Abr. 2019.
- GONÇALVES, E. J. T. et al. Núcleo de práticas em informática: contribuindo para a formação em sistemas de informação através do desenvolvimento de projetos de software. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 33., 2013, Maceió. **Anais [...]**. Maceió. 2013. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/0038.pdf>. Acesso: 24. Abr. 2019.
- HETHEY, J. M. **GitLab repository management**. Birmingham: Packt Publishing Ltd., 2013. 88 p. Disponível em: <https://books.google.com.br/books?id=tUYiAgAAQBAJ>. ISBN: 9781783281800. Acesso em: 12 Jun. 2019.
- HILTON, M. et al. **Continuous integration (ci) needs and wishes for developers of proprietary code**. Corvallis, OR : Oregon State University, Dept. of Computer Science, 2016. Disponível em: https://ir.library.oregonstate.edu/concern/technical_reports/2z10wr50b. Acesso em: 26 Mai. 2019.
- HUMBLE, J.; FARLEY, D. **Continuous delivery**: reliable software releases through build, test, and deployment automation. [S.l.]: Pearson Education, 2010. ISBN 9780321670229. Disponível em: <https://books.google.com.br/books?id=6ADDuzere-YC>. Acesso em: 14 Mai. 2019.
- NPI. Sobre o NPI. **Núcleo de práticas em informática (npi)**, 2019. Disponível em: http://www.npi.quixada.ufc.br/#_about. Acesso em: 04 Mai. 2019.
- POLKHOVSKIY, D. **Comparison between continuous integration tools**. Finland. Tampere University of Technology. 2016. Disponível em: <https://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/24043/polkhovskiy.pdf?sequence=1&isAllowed=y>. Acesso: 27 Mai. 2019.
- PRATES, R. O.; BARBOSA, S. D. J. **Avaliação de interfaces de usuário – conceitos e métodos**. 2003. Disponível em: https://www.researchgate.net/publication/242691294_Avaliacao_de_Interface_de_Usuario_-_Conceitos_e_Metodos. Disponível : Acesso em: 20 Out. 2019
- STÅHL, D.; BOSCH, J. **Experienced benefits of continuous integration in industry**

software product development: a case study., 2013. Disponível em:
https://www.researchgate.net/publication/266632251_Experienced_Benefits_of_Continuous_Integration_in_Industry_Software_Product_Development_A_Case_Study.
10.2316/P.2013.796-012. Acesso em: 20 Abr. 2019.

VASILESCU, et al. Quality and productivity outcomes relating to continuous integration in github. **ACM**, New York, 2015. In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015).

ROGERS, Y.; SHARP, H.; PREECE, J. **Interaction design: beyond human-computer interaction**. 3th ed. United Kingdom: John Wiley & Sons, 2011. 585 p. Disponível em:
https://books.google.com.br/books?id=b-v_6BeCwwQC. ISBN: 9780470665763 / 0470665769. Acesso em: dia mês ano. Acesso em: 15 Out. 2019.

APÊNDICE A – ROTEIRO PILOTO

Início da Entrevista

1. Qual o seu nome?
2. Em qual projeto você está trabalhando? *Proficiência, Gestão de riscos, Prontuário...*
3. A quanto tempo você está trabalhando nesse projeto?

Experiência e adaptação (antes do NPI)

4. Você já trabalhou em algum projeto que possuía Integração Contínua?

Se sim:

- a. Como foi sua experiência?
- b. Como foi sua adaptação? *Fácil, Moderada, Difícil*
- c. Qual era a ferramenta utilizada?

(Pule para a 5, e tenha em mente que o participante já possui experiência com a ferramenta)

Se não: (Pule para a 5)

Experiência e adaptação (atualmente)

5. Como está sendo sua experiência com a Integração Contínua no ambiente de desenvolvimento do NPI?
6. Como está sendo o processo de adaptação dessa ferramenta no ambiente de desenvolvimento do projeto? *Fácil, Moderado, Difícil.*
7. Sua equipe recebeu algum treinamento/orientação sobre o uso da Integração Contínua?
Sim Não.

Relação aos pontos a serem avaliados

8. Você percebeu algum impacto na **Produtividade da Equipe**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
9. Você percebeu algum impacto na **Qualidade do Código**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
10. Você percebeu algum impacto na **Redução de Processos Manuais**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
11. Você percebeu algum impacto na **Redução de Bugs**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
12. Você percebeu algum impacto na **Comunicação**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
13. Você percebeu algum impacto na **Automatização do Deploy**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.

Seção de comentários e sugestões (Área Livre)

14. Você recomendaria/usaria essa ferramenta em algum projeto pessoal/em organização?
Sim Não
15. Você possui alguma sugestão ou comentário?
Sim, qual? Não.

Fim da entrevista**APÊNDICE B – ROTEIRO FINAL (ESTAGIÁRIOS)****Início da Entrevista**

1. Qual o seu nome?
2. Em qual projeto você está trabalhando? *Proficiência, Gestão de riscos, Prontuário...*
3. A quanto tempo você está trabalhando nesse projeto?

Experiência e adaptação (antes do NPI)

4. Você já trabalhou em algum projeto que possuía Integração Contínua?

Se sim:

- a. Como foi sua experiência?
- b. Como foi sua adaptação? *Fácil, Moderada, Difícil*
- c. Qual era a ferramenta utilizada?

(Pule para a 5, e tenha em mente que o participante já possui experiência com a ferramenta)

Se não: (Pule para a 5)**Experiência e adaptação (atualmente)**

5. Como está sendo sua experiência com a Integração Contínua no ambiente de desenvolvimento do NPI?
6. Como está sendo o processo de adaptação dessa ferramenta no ambiente de desenvolvimento do projeto? *Fácil, Moderado, Difícil.*
7. Sua equipe recebeu algum treinamento/orientação sobre o uso da Integração Contínua?
Sim Não.

Relação aos pontos a serem avaliados

8. Você percebeu algum impacto na **Produtividade da Equipe**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
9. Você percebeu algum impacto na **Qualidade do Código**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
10. Você percebeu algum impacto na **Redução de Processos Manuais**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
11. Você percebeu algum impacto na **Redução de Bugs**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
12. Você percebeu algum impacto na **Comunicação**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.

Seção de comentários e sugestões (Área Livre)

13. Você recomendaria/usaria essa ferramenta em algum projeto pessoal/em organização?
Sim Não
14. Você possui alguma sugestão ou comentário?
Sim, qual? Não.

Fim da entrevista

APÊNDICE C – ROTEIRO FINAL (SUPERVISORES)

Início da Entrevista

1. Qual o seu nome?
2. Quais projeto você está supervisionando? *Proficiência, Gestão de riscos, Prontuário...*
3. A quanto tempo você está trabalhando nesse(s) projeto(s)?

Experiência e adaptação (atualmente)

4. Como está sendo sua experiência, com a Integração Contínua no ambiente de desenvolvimento do NPI?
5. Como está sendo o processo de adaptação dessa ferramenta no ambiente de desenvolvimento do projeto? *Fácil, Moderado, Difícil.*
6. Sua equipe aplicou algum treinamento/orientação sobre o uso da Integração Contínua?
Sim Não.

Relação aos pontos a serem avaliados

7. Você percebeu algum impacto na **Produtividade da Equipe**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
8. Você percebeu algum impacto na **Qualidade do Código**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
9. Você percebeu algum impacto na **Redução de Processos Manuais**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
10. Você percebeu algum impacto na **Redução de Bugs**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
11. Você percebeu algum impacto na **Comunicação**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.
12. Você percebeu algum impacto na **Automatização do Deploy**, após a utilização da Integração Contínua?
Positivo Negativo Nulo.

Seção de comentários e sugestões (Área Livre)

13. Você possui alguma sugestão ou comentário?

Sim, qual? Não.

Fim da entrevista

APENDICE D – QUESTIONÁRIO

Seção 1 de 3

Dados do respondente

Qual seu nome? _____.

Qual sua função no NPI? _____.

A quanto tempo você trabalha no NPI? _____.

Seção 2 de 3

Seleção de pontos importantes a serem avaliados, para o NPI

Selecione o nível de importância, de cada ponto abaixo, para o ambiente de desenvolvimento do NPI, na sua opinião.

	Muito Baixa	Baixa	Média	Alta	Muito Alta
Produtividade da equipe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Qualidade de código	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Redução de riscos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Transparência do projeto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Redução de processos manuais (compilação, testes, integração com o banco de dados, <i>code review</i>, <i>deploy</i> e <i>feedback</i>)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Redução de bugs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Automatização do <i>deploy</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Seção 3 de 3

Sugestões (Opcional)

Escreva uma sugestão para o trabalho, autor ou para o questionário:

APÊNDICE E – TCLE

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Declaro, por meio deste termo, que concordei em ser entrevistado(a) e/ou participar na pesquisa de campo referente ao projeto/pesquisa intitulado(a) **ANÁLISE DA UTILIZAÇÃO DA INTEGRAÇÃO CONTÍNUA NO AMBIENTE DE DESENVOLVIMENTO DO NPI** desenvolvida(o) por **MAX WENDEL FERREIRA FREIRE ARAÚJO**. Fui informado(a), ainda, de que a pesquisa é orientada por **CARLOS DIEGO ANDRADE DE ALMEIDA**.

Afirmo que aceitei participar por minha própria vontade, sem receber qualquer incentivo financeiro ou ter qualquer ônus e com a finalidade exclusiva de colaborar para o sucesso da pesquisa. Fui informado(a) dos objetivos estritamente acadêmicos do estudo.

Fui também esclarecido(a) de que os usos das informações por mim oferecidas estão submetidos às normas éticas destinadas à pesquisa envolvendo seres humanos, da Comissão Nacional de Ética em Pesquisa (CONEP) do Conselho Nacional de Saúde, do Ministério da Saúde.

Minha colaboração se fará de forma anônima, por meio de entrevista semiestruturada a ser gravada a partir da assinatura desta autorização. O acesso e a análise dos dados coletados se farão apenas pelo(a) pesquisador(a) e/ou seu(s) orientador(es).

Fui ainda informado(a) de que posso me retirar desse(a) pesquisa a qualquer momento, sem prejuízo para meu acompanhamento ou sofrer quaisquer sanções ou constrangimentos.

Atesto recebimento de uma cópia assinada deste Termo de Consentimento Livre e Esclarecido, conforme recomendações da Comissão Nacional de Ética em Pesquisa (CONEP).

Quixadá, ____ de _____ de _____.

Assinatura do(a) participante: _____.

Assinatura do(a) pesquisador(a): _____.