



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

Proposta de framework iOS para implementação de Feature Toggle

Proposta de Trabalho de Graduação

Aluna:

ISABELLY CRISTINA FERREIRA DAMASCENA

Orientador:

LEOPOLDO MOTTA TEIXEIRA

Recife, 2018

Sumário

Contexto	2
Objetivo	4
Cronograma	5
Avaliadores	6
Referências	7
Assinaturas	8

Contexto

Segundo Martin Fowler, *Continuous Delivery* (CD) é uma maneira de construir software de forma que este possa ser entregue a qualquer momento. [1]. E para que isso seja possível, é necessário que o time de desenvolvimento do software se muna de algumas ferramentas.

Uma dessas ferramentas é a *Continuous Integration* (CI). Esta é uma prática que visa múltiplas integrações do software a ser construído diariamente. Cada integração é verificada por uma *build* automatizada, incluindo seus testes para a detecção de erros rapidamente, diz Martin Fowler [2]. Essa ferramenta facilita a junção de códigos feitos por diferentes pessoas desenvolvedoras, e, no melhor cenário todo código integrado seria um código entregável.

Em um time, ao ser estabelecida uma metodologia ágil como Scrum, algumas práticas são utilizadas para receber feedback de forma rápida dos *stakeholders* envolvidos no desenvolvimento do software: *Product Owner*, *Customer*, etc. Muitas vezes, é responsabilidade da PO representar o *customer*, decidindo se será feita a *release* de uma *feature* que o time construiu.

"The Product Owner must continuously engage the customer and stakeholders to ensure the Team is building the right product and therefore delivering the ROI expected of it. The Product Owner has the opportunity to steer the team in a different direction at the end of every Sprint, so he/she must be ready to do just that if necessary.[3]"

O conceito de CI foi construído desprendido de modelos de *branches* ou controle de versionamento, contudo, a intenção original é que as múltiplas integrações sejam feitas em um ponto único, remetendo e casando quase que perfeitamente com o desenvolvimento baseado em *trunks* [4]. Ao utilizar-se desta desse modelo de *branch*, o time de desenvolvimento encontra alguns desafios relacionados a *release* e seu maior *stakeholder*, a PO.

Um cenário que deixa claro um dos desafios é: Dado um conjunto de *cards/features* apresentadas a PO apenas um subconjunto destas são aceitas como feature que podem entrar na *release* da nova versão do *software*. *Como o time entrega somente as features que foram aceitas visto que todo o código está integrado? O time desfaz o código já feito, entrega e tem retrabalho depois?*

De fato, este momento é um ponto crítico para um time Ágil. No contexto, em que a PO aprova um subset das *features* que foram demonstrados e o time está construindo o software usando *Continuous Integration* usando como modelo de branch *Trunk-Based*, é preciso haver uma forma de separar o código que foi aceito pela PO do código que foi rejeitado com o mínimo de retrabalho

possível. É nesta situação em que a abordagem *Feature Toggle* toma forma, sendo uma ferramenta que de maneira simplista aplica *switches* dentro do código já feito; podendo assim, definir se um determinado trecho de código será executado ou não.

"The basic idea is to have a configuration file that defines a bunch of toggles for various features you have pending. The running application then uses these toggles in order to decide whether or not to show the new feature.[5]"

Voltando a visão para o desenvolvimento mobile, mais precisamente para o desenvolvimento iOS; nota-se uma falta de suporte quando se trata do desenvolvimento de apps usando Swift. A proposta deste trabalho de graduação é criar um framework intuitivo e de fácil acoplamento a um código existente.

Objetivo

No cenário de desenvolvimento utilizando da Entrega Contínua (CD) e Integração Contínua (CI) é necessário um levantamento sobre o contexto atual da área, analisando o status quo, os frameworks e/ou SDKs em evidência que possibilitam o uso de Feature Toggle.

Este trabalho tem como objetivo principal a criação de um framework iOS, construído em Swift, que possibilite a implementação de Feature Toggles de maneira fácil e intuitiva.

Faz-se necessário enfatizar que *Swift* é uma linguagem open source que vem sendo atualizada e melhorada de acordo com as contribuições da comunidade.

Como objetivos específicos, quer-se:

- Levantar frameworks que possibilitam a implementação de Feature Toggle;
- Implementar e demonstrar um caso de uso do framework, através de uma aplicação prática;
- Comparar o framework criado com os frameworks levantados;
- Disponibilizar o código desenvolvido como código aberto para a comunidade, estimulando a contribuição;
- Disponibilizar em pelo menos dois gerenciadores de dependência existentes - Carthage, CocoaPods, etc.

Cronograma

A *Tabela 1* descreve o cronograma previsto para a construção, preparação e entrega deste trabalho de graduação ao longo desse semestre.

	<i>Agosto</i>				<i>Setembro</i>				<i>Outubro</i>				<i>Novembro</i>				<i>Dezembro</i>			
<i>Definição de Escopo</i>	■	■	■																	
<i>Formulação de Proposta</i>			■	■	■															
<i>Implementação do Framework</i>					■	■	■	■	■	■	■	■	■							
<i>Elaboração do Documento</i>					■	■	■	■	■	■	■	■	■	■						
<i>Preparação para Defesa</i>														■	■	■	■			

Tabela 1 - Cronograma de Trabalho de Graduação

Avaliadores

Os seguintes professores são considerados como potenciais avaliadores deste trabalho de graduação:

- Kiev Santos Gama (kiev@cin.ufpe.br)
- Fernando José Castor de Lima Filho (fjclf@cin.ufpe.br)
- Paulo Borba (phmb@cin.ufpe.br)

Referências

- [1] MARTIN, Fowler. Continuous Delivery. 2013. Acesso em: 20/08/2018. Disponível em: <https://martinfowler.com/bliki/ContinuousDelivery.html>.
- [2] MARTIN, Fowler. Continuous Integration. 2006. Acesso em: 20/08/2018. Disponível em: <https://martinfowler.com/articles/continuousIntegration.html>.
- [3] MILUNSKY, Jack. Continuous Integration. 2009. Acesso em: 20/08/2018. Disponível em: <https://dzone.com/articles/product-owner-top-10>.
- [4] HAMMANT, Paul; SMITH, Steve. Trunk-Based Development: Continuous Integration (CI). 2017. Acesso em: 20/08/2018. Disponível em: <https://trunkbaseddevelopment.com/continuous-integration/>.
- [5] MARTIN, Fowler. Feature Toggle. 2009. Acesso em: 02/09/2018. Disponível em: <https://martinfowler.com/bliki/FeatureToggle.html>.

Assinaturas

O aluno e orientador assinam abaixo, comprometendo-se com o desenvolvimento do projeto descrito neste documento.

Isabelly Cristina Ferreira Damascena
Aluno

Leopoldo Motta Teixeira
Orientador