

Aplicação do PSP (*Personal Software Process*) integrado ao BDD (*Behavior-Driven Development*) no desenvolvimento de um aplicativo Android

Gianfranco B. Gerbaldo¹, Joyce Miranda dos Santos¹

¹ Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM)
AV. Sete de Setembro, 1975 Caixa Postal 69020-120 – Manaus – AM – Brasil

giangerbaldo@gmail.com, jsantos@ifam.edu.br

***Abstract.** Software companies use development processes along with methodologies or techniques to support the development and improve the quality of their final product. This paper will present the main concepts of the PSP (*Personal Software Process*) and the BDD (*Behavior-Driven Development*) with the goal of proposing a strategy of integration between them. At the end, the expected result is to provide a reference material about how BDD can be integrated to the phases of planning and test automation of the PSP, showing how this strategy can be adapted to any software development process. The proposed strategy will be validated through a case study of Android application development.*

1. Introdução

Todo desenvolvedor utiliza algum processo para construir softwares. Esse processo pode ser aleatório, modificar-se constantemente, ser eficiente ou não. O PSP (*Personal Software Process*) provê métodos bem definidos para o aumento da produtividade individual durante o processo de desenvolvimento de software [Humphrey 2000]. Seu foco está em estabelecer um mecanismo para melhorar, no nível do desenvolvedor, sua capacidade de planejamento, acompanhamento e qualidade dos resultados [Pressman 2011]. Este processo utiliza um conjunto de métodos, formulários e roteiros para planejar, medir e gerenciar o trabalho realizado. O PSP possui como objetivo geral produzir produtos de software sem defeitos, respeitando prazos e custos planejados, a partir de um comprometimento pessoal com a qualidade e com a melhoria contínua do processo [Koscianski e Soares 2006].

O BDD (*Behavior-Driven Development*) foi proposto como forma de auxiliar o desenvolvedor a criar testes e integrar regras de negócio com a linguagem de programação, mantendo o foco no comportamento do software [North 2006]. Sua aplicação é baseada no modelo de entradas-eventos-saídas (*given-when-then*) que tem como objetivo capturar cenários que reflitam os critérios de validação de um comportamento esperado para o software. O exemplo apresentado na Figura 1 mostra que cada cenário descrito deve possuir um contexto inicial (dado), condições (quando), resultado esperado (então) e seus conectores (e). Desta forma, é possível atender aos requisitos, realizar testes automatizados nos mais diversos cenários, reduzindo assim o número de defeitos encontrados. Neste sentido, a utilização de técnicas como o BDD

durante o processo de desenvolvimento de software tende a aumentar a qualidade do produto final.

Dado que o usuário está na tela principal Quando o usuário preencher o nome do município E clicar em pesquisar Então deve ser exibida a lista de barcos do município desejado pelo usuário
--

Figura 1. Exemplo do modelo de entradas-eventos-saídas do BDD.

Atualmente, não existem estudos relacionados à integração da técnica de BDD às fases definidas pelo PSP. Partindo deste princípio, os objetivos deste trabalho são: (i) propor uma estratégia de integração do BDD na atividade de Codificação e Teste do processo PSP e (ii) definir diretrizes de como utilizar o BDD na automatização de testes. Como forma de validar a estratégia proposta, esta será aplicada no desenvolvimento de um aplicativo Android, cujo objetivo é gerenciar a compra de passagens de viagens fluviais realizadas pelo porto de Manaus.

2. Proposta de projeto de pesquisa

Neste trabalho, a solução proposta para a integração entre o PSP e o BDD consiste inicialmente na adaptação do modelo original do PSP, no que diz respeito às fases de Codificação e Teste. Ambas ocorriam em momentos distintos do processo. Com a adaptação, as fases foram unificadas para fazer melhor proveito do BDD conforme ilustra a Figura 2.

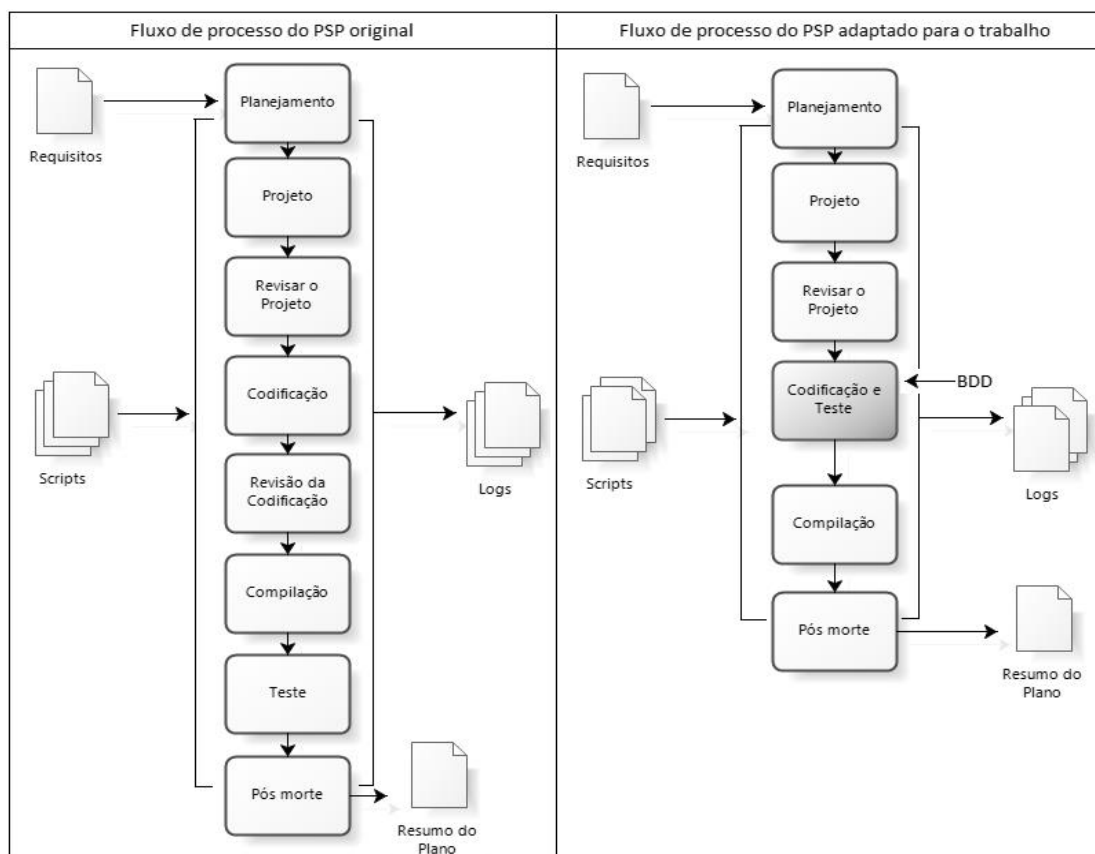


Figura 2. Adaptação do modelo PSP para integração com BDD.

A Figura 3 ilustra as atividades de Codificação e Teste adaptadas para o trabalho e onde irá ocorrer a integração com o BDD. No início da fase serão elaborados os cenários relacionados às funcionalidades que serão implementadas. A elaboração dos cenários seguirá o modelo de entradas-eventos-saídas (*given-when-then*) proposto pelo BDD, onde deverão ser descritos os critérios de aceitação para cada cenário. Após essa atividade, será feita a codificação da funcionalidade que irá atender ao comportamento requisitado.

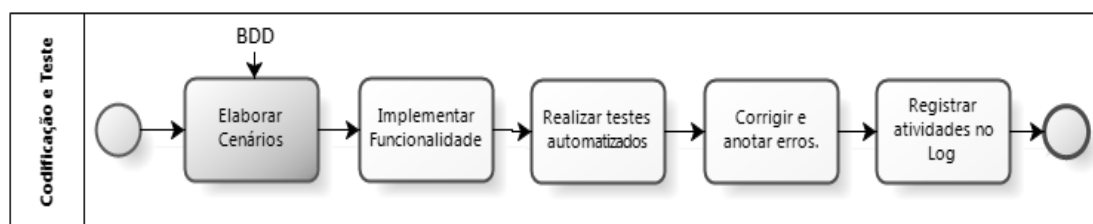


Figura 3. Esquema proposto de integração do PSP com o BDD.

Uma vez implementadas as funcionalidades, o próximo passo consiste na execução dos casos de testes utilizando os cenários elaborados a partir do BDD. Para automatizar a execução dos testes, será utilizado o *framework Calabash-Android* [Krukow 2012] como forma de estabelecer a integração entre os casos de teste e o Android. Por fim, as correções dos erros encontrados serão anotadas e registradas no documento de *log*.

3. Conclusão

Ao final deste trabalho, espera-se disponibilizar um material de referência sobre como o BDD pode ser aplicado para fazer o planejamento e automatização de testes propostos no PSP, apontando as vantagens da aplicação do modelo proposto, além de mostrar como este modelo de integração pode ser adaptado a qualquer processo de desenvolvimento de software. Espera-se também apresentar as principais vantagens do BDD em relação à utilização dos casos de testes tradicionais.

Referências

- Humphrey, W. (2000) “The Personal Software Process (PSP)”, <http://www.sei.cmu.edu/reports/00tr022.pdf>, Março.
- Koscianski, A. e Soares, M. (2006), Qualidade de Software, Novatec, 2th edição.
- Krukow Karl (2012) “Calabash: Functional Testing for Mobile Apps”, <http://blog.lesspainful.com/2012/03/07/Calabash/>, Março.
- North, D. (2006) “Introducing: Behaviour-driven development”, <http://dannorth.net/introducing-bdd/>, Março.
- Pressman, R. (2011), Engenharia de Software, Pearson Addison, 5th edição.