

CESAR SCHOOL

Unidade de Educação do Centro de Estudos e Sistemas Avançados do
Recife (CESAR)

**MESTRADO PROFISSIONAL em Engenharia de
Software**

Proposta de Pesquisa

Autor:

Jeferson Barros Alves

Linha de Pesquisa:

Fábrica de Software

Sumário

1	Apresentação	2
2	Motivação para a pesquisa	2
3	Descrição do problema a ser resolvido	3
4	Metodologia	3
5	Conclusões e considerações finais	4

1 Apresentação

Este documento tem como objetivo propor um projeto de estudo a ser desenvolvido durante o Mestrado Profissional em Engenharia de Software, com intuito de trabalhar com conteúdos práticos de necessidade do autor, adquirindo como resultado uma evolução nas experiências na área de atuação profissional e construção de competências que agregam maior valor profissional.

2 Motivação para a pesquisa

Em empresas com alta demanda de produção de software, pode-se observar uma busca por redução de custos, tempo de produção, melhoria na qualidade, manutenibilidade de código, dentre outros fatores. Uma das estratégias adotadas pelas empresas para alcançar tais objetivos é a reutilização de software.

De acordo com [Sametinger 1997], o reuso de software foi introduzido com uma das alternativas para superar a crise de software, termo criado para descrever o aumento da carga de frustração do desenvolvimento e manutenção de softwares. Impactando positivamente em aspectos como qualidade, custos e produtividade.

Segundo [Sommerville 2011], a disponibilidade de softwares reusáveis tem aumentado significativamente. Algumas grandes empresas fornecem uma variedade de componentes reutilizáveis para seus clientes. Padrões, como *web service*, tornaram mais fácil o desenvolvimento de serviços gerais e reuso destes em uma variedade de aplicações.

Ao longo dos últimos vinte anos, muitas técnicas foram desenvolvidas para oferecer suporte ao reuso de software. Essas técnicas exploram os fatos de que os sistemas, no mesmo domínio de aplicação, são semelhantes e têm potencial para reuso. O reuso é possível em diferentes níveis, desde funções simples até aplicações completas, e normas para componentes reusáveis facilitam o reuso [Sommerville 2011]. Dentre essas técnicas podemos salientar a linha de produto de software, que segundo [Sommerville 2011] é um tipo de aplicação que é generalizado em torno de uma arquitetura comum para que esta possa ser adaptada para diferentes clientes.

Em [Orozco et al. 2008], foi proposto uma aglutinação dos requisitos comuns das abordagens de teste baseados em modelos através de uma arquitetura de linha de produtos de software, proporcionando assim o reuso dos elementos arquiteturais comuns destas abordagens, facilitando a implementação de ferramentas, diminuindo o tempo de desenvolvimento e aumentando a qualidade destas ferramentas, sendo estas vantagens promovidas pelos con-

ceitos de linha de produtos de software.

No trabalho de [Pontes 2017], observamos que é proposto um método de extração e evolução de linha de produto de software a partir de sistemas existentes implementados na linguagem Java no domínio de sistemas de controle de espaço físico atendendo novos requisitos durante o processo de evolução.

3 Descrição do problema a ser resolvido

O teste de software muitas vezes requer mais trabalho de projeto do que qualquer outra ação da engenharia de software. Se for feito casualmente, perde-se tempo, fazem-se esforços desnecessários, e, ainda pior, erros passam sem ser detectados. Portanto, é razoável estabelecer uma estratégia sistemática para teste de software [Pressman e Maxim 2011].

De acordo com [Bertolino 2007] teste é uma atividade essencial em Engenharia de Software. Em termos, simples é a observação da execução de um sistema de software para validar se a execução ocorre como o esperado e identifica potenciais defeitos. Testes são amplamente usados na indústria para medida de qualidade.

A automação de teste de software pode reduzir drasticamente o esforço requerido para as atividades de teste. Através da automação, os testes podem ser realizados em minutos ao invés de demorarem horas para serem executados manualmente, podendo alcançar uma diminuição de esforço em mais de 80

O objetivo principal desta proposta é a instanciación e automatização de um processo de Linha de Produto de Software orientado a família de produtos no domínio de testes automatizados para o sistema operacional *Android*.

4 Metodologia

Uma das abordagens mais eficazes para o reúso é criar linhas de produto de software ou famílias de aplicações. Uma linha de produto de software é um conjunto de aplicações com uma arquitetura comum e componentes compartilhados, sendo cada aplicação especializada para refletir necessidades diferentes. O núcleo do sistema é projetado para ser configurado e adaptado para atender às necessidades de clientes diferentes. Isso pode envolver a configuração de alguns componentes, implementação de componentes adicionais e a modificação de alguns dos componentes para refletir novos requisitos [Sommerville 2011].

5 Conclusões e considerações finais

Conclusão da proposta de pesquisa.

Referências

- [Bertolino 2007]BERTOLINO, A. Software testing research: Achievements, challenges, dreams. In: IEEE COMPUTER SOCIETY. *2007 Future of Software Engineering*. [S.l.], 2007. p. 85–103.
- [Fewster e Graham 1999]FEWSTER, M.; GRAHAM, D. *Software test automation: effective use of test execution tools*. [S.l.]: ACM Press/Addison-Wesley Publishing Co., 1999.
- [Orozco et al. 2008]OROZCO, A. M. S. et al. Linha de produtos de testes baseados em modelos. Pontifícia Universidade Católica do Rio Grande do Sul, 2008.
- [Pontes 2017]PONTES, E. S. R. d. Um método para extração e evolução de linhas de produto de software a partir de sistemas web existentes. 2017.
- [Pressman e Maxim 2011]PRESSMAN, R.; MAXIM, B. *Engenharia de Software, uma abordagem profissional—7ª Edição*. [S.l.]: McGraw Hill Brasil, 2011.
- [Sametinger 1997]SAMETINGER, J. *Software engineering with reusable components*. [S.l.]: Springer Science & Business Media, 1997.
- [Sommerville 2011]SOMMERVILLE, I. *Engenharia de Software - 9ª Edição*. [S.l.]: Pearson Prentice Hall, 2011.