UMA ANÁLISE DO USO DE FERRAMENTAS EM DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE PARA ATUALIZAÇÃO DA PLATAFORMA ANDROID

Hualter O. Barbosa, Bruno A. Bonifacio, Thiago M. Menezes, Luis Felipe Uebel,
Fernando B. Pires e Andre F. Neto
Sidia Instituto de Ciência e Tecnologia
Av. Darcy Vargas, 654 - Parque 10 de Novembro, Manaus – AM, Brasil

RESUMO

Com a globalização do processo de desenvolvimento de software, a academia e a indústria têm pesquisado o desenvolvimento de novas tecnologias para assegurar maior competitividade em novos mercados. Apesar de empresas buscarem por profissionais qualificados para manter a eficácia e a eficiência do processo. É necessário adotar ferramentas que ajudem no ciclo de vida do software para facilitar seu gerenciamento e execução, bem como para seguir prazos e a utilização de recursos. Neste artigo, exploramos a experiência durante o uso de uma ferramenta, chamada de CLCATCH, para automatizar uma das atividades do ciclo de desenvolvimento de software distribuído, realizado no Sidia Instituto de Ciência e Tecnologia, localizado na Amazônia. Este artigo apresenta: (1) os desafios encontrados pelo Sidia em Desenvolvimento Distribuído de Software (DDS) para projetos de atualização da plataforma Android; (2) a solução aplicada, por meio do desenvolvimento e implantação da CLCATCH e; (3) os resultados obtidos, considerando os indicadores de tempo e falhas mostrando relevância na execução dos projetos do Sidia. Espera-se incentivar a indústria e a comunidade científica a fim de melhorar os processos de desenvolvimento de software através do uso de tecnologias.

PALAVRAS-CHAVE

DDS, Relatório de Experiência, Sistema Operacional Android, Atualização do Android

1. INTRODUÇÃO

Diante de novos mercados no desenvolvimento de software, o aumento da competitividade cria a necessidade de resultados cada vez mais rápidos. Dada a complexidade envolvida no desenvolvimento de software, algumas atividades acabam sendo propensas a enganos pelo desenvolvedor, o que pode impactar na qualidade de um produto (Stawnicza, 2015). Por essa razão, as organizações têm buscado alternativas para auxiliar na produtividade dos desenvolvedores de software e na qualidade dos artefatos de software gerados.

Neste contexto, várias ferramentas têm sido propostas com o objetivo de facilitar atividades intermediárias do desenvolvimento, como: sistemas de integração contínua, sistemas de controle de versão, sistemas de gestão de atividades, gestão de mudanças dentre outras. O uso dessas ferramentas pode facilitar a gestão das atividades de desenvolvimento de software, o acompanhamento de prazos, a gestão de equipes e o consumo de recursos (Weimann, 2013). Nesse sentido as ferramentas tornam-se obrigatórias na busca por qualidade e agilidade no desenvolvimento de novos produtos, especialmente com o mercado cada vez mais globalizado.

O Sidia é um Instituto de Ciência e Tecnologia, localizado na Amazônia, que possui inúmeros projetos com Institutos de Pesquisa e Desenvolvimento (P&D) e outras empresas ao redor do mundo. O processo de desenvolvimento dos projetos em colaboração com o Sidia é feito utilizando Desenvolvimento Distribuído de Software (DDS). O DDS tem como grande vantagem a colaboração entre equipes de desenvolvimento que trabalham juntas, no mesmo projeto, mas estão localizados em países ou cidades diferentes (Audy e Prikladnick, 2007). No entanto, o Sidia enfrentava dificuldades para alinhar o processo de desenvolvimento no contexto de DDS. Este cenário cria um grande desafio para gerenciar o processo de desenvolvimento de software, especialmente em projetos com outras instituições parceiras que possuem diferenças na forma como conduzem seus processos de desenvolvimento. Como alternativa, o Sidia tem criado iniciativas para

melhorar o processo de desenvolvimento, com o objetivo de diminuir a ocorrência de erros, por desenvolvedores, durante a integração de códigos, controles de versão e gestão de novos requisitos. O propósito é para automatizar atividades específicas do ciclo de desenvolvimento dos seus projetos em ambiente de DDS.

Este artigo apresenta um relato de experiência, sobre o aprimoramento de uma das atividades importantes no ciclo de DDS do Sidia, por meio de uma ferramenta para automatizar o processo de desenvolvimento de atualização da plataforma Android. A ferramenta proposta, chamada de CLCATCH, foi desenvolvida com o objetivo de automatizar a busca de pacotes (*patches*) de atualização e correções de software, em projetos Android. Para avaliação do impacto da ferramenta proposta, no processo de gestão dos projetos do Sidia, foi feita uma análise qualitativa, por meio de questionários, baseado no Modelo de Aceitação de Tecnologia (TAM). O objetivo deste artigo consiste em apresentar os desafios encontrados, a motivação para construção da solução adotada e os resultados obtidos, analisando o custo-benefício com a adoção da ferramenta. Os resultados iniciais mostram importante contribuição para melhorar o processo de DDS executado no Sidia. Espera-se com isso incentivar a indústria e a comunidade científica a criar novas formas de melhorar os processos de desenvolvimento de software, por meio do uso de novas tecnologias de software.

Este trabalho está organizado da seguinte maneira: a Seção 2 apresenta uma visão geral de ferramentas utilizadas para auxiliar o DDS, e apresenta a ferramenta construída para automatizar uma das atividades do desenvolvimento de projetos de desenvolvimento da plataforma Android, feita no Sidia; a Seção 3 apresenta os resultados obtidos. Na Seção 4 as lições aprendidas são apresentadas, e finalmente, a Seção 5 apresenta as considerações finais e perspectivas de trabalhos futuros.

2. FERRAMENTAS PARA O DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE

Com a globalização no processo de desenvolvimento de software, a indústria e a academia tem buscado alternativas para responder com maior rapidez na construção de novas tecnologias. Neste sentido, o Desenvolvimento Distribuído de Software (DDS) tem se tornado um elemento estratégico para garantir maior competitividade (Lima et al. 2016). O cenário no qual o DDS é feito, onde equipes trabalham juntas, mas não estão localizados geograficamente no mesmo ambiente, permitiu o crescimento desta área, devido a economia de recursos e redução de custos (Audy e Prikladnick, 2007).

Por isso, muitas empresas têm investido na adoção e desenvolvimento de tecnologias de software para auxiliar em DDS, especialmente em atividades de gestão dos projetos. As soluções desenvolvidas nesta área têm focado em gestão de atividades, consumo de recursos e acompanhamento de prazos (Weimann, 2013). No entanto, ainda existem desafios importantes a serem superados, especialmente na gerência de atividades de projetos que usam DDS (Bosnic et al. 2011).

Lima et al. (2016) propõe uma ferramenta que apoia automatização para a gerência de processos de software, chamada WebAPSEE. Esta ferramenta auxilia na modelagem e execução de processos de software. Além disso, a abordagem apresenta outras características, em desenvolvimento, para auxiliar simulação, reutilização e instanciação de processos.

Weimann et al (2013), propõem a utilização de uma ferramenta para maximizar o desempenho de equipes, na execução de tarefas, em ambientes distribuídos. O propósito foi identificar características, como tempo de resolução de defeitos e demandas novas de desenvolvimento, que podem influenciar em tarefas específicas do ciclo de desenvolvimento. Os resultados apresentados no estudo de Weimann et al (2013), mostram que a falta de ferramentas, para atividades intermediárias, pode influenciar no desempenho dos desenvolvedores, bem como o monitoramento de informações entre as equipes.

Diante desse contexto, este artigo apresenta a experiência na construção de uma ferramenta para automatizar uma atividade específica do time de desenvolvimento do Sidia. O propósito foi avaliar o impacto da realização da atividade com o uso da ferramenta, além da experiência de uso com a implantação desta nova abordagem, aplicados em projetos de atualização do sistema operacional Android. A próxima Seção apresenta o contexto, a motivação e uma visão geral da ferramenta, chamada CLCATCH.

2.1 A Ferramenta CLCATCH

O Sidia atualmente possui projetos focados no desenvolvimento e melhoria do Sistema Operacional (S.O.) Android, para os dispositivos da Samsung. Estes projetos são divididos em desenvolvimento e atualização de S.O. em dispositivos já lançados no mercado e novos modelos. As atualizações de S.O. podem ser tanto para inclusão de novos requisitos de desenvolvimento quanto para pacotes de aprimoramento de segurança, que são um conjunto de pacotes contendo funcionalidades para melhoria do S.O. Cada pacote é armazenado em um sistema de controle de versão, de acordo com cada modelo de dispositivo que são enviados em lotes, chamados de *patches*. O uso de pacotes, ou patches, visa a otimização do processo de desenvolvimento de software, uma vez que este ciclo de desenvolvimento é feito em conjunto com outros institutos de P&D, distribuídos geograficamente no Brasil. O processo de desenvolvimento foca na geração de *releases*, que consiste em uma versão de software, produzida pelo Sidia contendo correção de falhas, *patches* de segurança e novos requisitos de software das operadoras para o sistema operacional Android.

O processo de *release* é mantido por uma equipe específica, formada por engenheiros de software que ocupam o papel de líder de projeto (LP). Esta equipe é responsável pela gestão das demandas de cada projeto, tendo como atividades principais a inclusão de *patches* de atualizações de segurança, monitoramento de novos requisitos das operadoras de telefonia dentre outras. As demandas de desenvolvimento são solicitadas pelas operadoras, ou pelo Google, por meio de novos requisitos de software. O processo de desenvolvimento executado no Sidia é representado na Figura 1.

Figura 1. Visão Geral do Processo de Desenvolvimento da Plataforma Android

Os projetos de desenvolvimento de plataforma Android iniciam com a criação de repositório, feito por um time de Gerência de Configuração. Após a criação do repositório do projeto, o time de LPs realiza o porte do Android, identificando as melhorias que devem ser aplicadas (Figura 1.1) e os requisitos solicitados para geração de uma nova *release* (Figura 1.2). Cada novo requisito recebe um rótulo único que passa pela análise e classificação de prioridade de desenvolvimento até chegar aos desenvolvedores, que por sua vez, atendem aos requisitos, mapeando suas respectivas implementações a um identificador e uma descrição com o rótulo do requisito (Figura 1.3).

Para melhor organização dos times distribuídos, o Sidia emprega uma abordagem que utiliza pacotes (*patches*) de atualização, chamadas de *Propagation Patch* (PP). Essa abordagem adota um processo de qualidade que começa com as demandas dos clientes, para construção do produto, e percorre todo fluxo processual da empresa até chegar aos respectivos responsáveis por atender às demandas, que são os grupos de desenvolvedores da arquitetura Android (Conti, 1993).

Para criar uma nova *release*, o time de LPs precisa compilar implementações e novas solicitações e aplicar as mudanças para versão de S.O. em cada modelo, gerando modificações que são agrupadas em *patches*, integradas dentro de cada PP. A vantagem da utilização de PPs é o aproveitamento de código, que pode ser aplicado em diferentes modelos com a mesma arquitetura de hardware.

Após a geração de PPs é criada uma *release* que posteriormente é enviada para testes (Figura 1.4). Caso sejam detectadas falhas nos testes de software, estas são analisadas pelo time de LP de cada projeto e enviadas para o time responsável (Figura 1.5). Em caso de não haver falhas a *release* é enviada para os usuários finais, por meio de notificação de atualização da plataforma no celular de cada usuário.

Como o time de LP realiza a gestão das demandas no processo de desenvolvimento, a identificação das PPs é crítica, pois torna-se necessário identificar quais novos rótulos e *patches* precisam ser integrados nas PPs. Munidos dessa informação, os analistas acessam o sistema de controle de versões de software e buscam manualmente os rótulos para encontrar o identificador da respectiva implementação. Em projetos de atualização de S.O.s a busca acaba dificultando o processo de geração de *release*, pois é necessário identificar a *release* mais recente, para identificar quais os novos requisitos precisam ser integrados na nova versão.

A título de simplificação do texto, a palavra rótulo referenciar-se-á ao rótulo do requisito, enquanto a palavra identificador ao identificador da implementação do requisito, conhecidos como *patches*. Para poucos rótulos, dentro de cada *patch*, a compilação de identificadores é uma tarefa pouco impactante dentro do processo do Sidia. Contudo, para muitas mudanças solicitadas, torna-se uma tarefa repetitiva, monótona e sujeita a erros, que pode gerar impacto negativo significativo no processo de desenvolvimento.

Nesse contexto, foi identificado como oportunidade a criação de uma ferramenta, com o propósito de minimizar o impacto da detecção dos rótulos, no repositório dos projetos. Assim, foi proposta a CLCATCH, apresentado na Figura 2.

Figura 2. Visão Geral da Ferramenta CLCATCH

A CLCATCH é uma aplicação web, desenvolvida com as linguagens de programação Java e PHP, que consiste no formulário, com os seguintes campos:

- Endereço IP do servidor de controle de versão SVN (*Subversion*);
- Porta do servidor de SVN;
- Usuário do SVN;
- Senha do SVN;
- Arquivo XML resultante do PP;
- Arquivo CSV resultante de *Bug Tracking*.

O objetivo da CLCATCH é identificar os *patches*, dentro de cada PP, que são armazenadas em arquivos XML e CSV e mapeá-las com alterações dentro do SVN. Com o envio desses dados, a CLCATCH lê os rótulos no arquivo das PPs, autentica o usuário e pesquisa os rótulos no SVN que por sua vez, retorna uma lista de identificadores. A CLCATCH distingue os rótulos com identificadores daqueles não agrupado na PP. Os identificadores encontrados são formatados para um padrão inteligível à geração da *release*, ordenados e tem as duplicatas removidas. Em seguida, a CLCATCH exibe uma lista de identificadores já formatados para aplicar na *release* e outra de rótulos não encontrados.

A motivação para a construção da CLCATCH para atividades de detecção de *patches*, por meio dos rótulos foi aumentar a produtividade dos LPs e agilizar a disponibilização das atualizações dos projetos de desenvolvimento de S.O.s, dentro do Sidia. Este relato apresenta um estudo, no uso da CLCATCH nas atividades do time de LPs. E apresenta o impacto do uso desta ferramenta no processo de geração de *release* dos S.O.s para os dispositivos móveis da Samsung.

Neste artigo, apresenta-se o impacto do uso da CLCATCH nos projetos desenvolvimento de plataforma, que focam especialmente em atualizações de segurança do Android presentes nos dispositivos Samsung, uma das parceiras do Sidia. Como forma de avaliar o impacto da ferramenta CLCATCH no processo de busca de novas mudanças no repositório, foi feita uma análise qualitativa, considerando as buscas de *patches* e rótulos em PPs. Assim, foi realizada uma pesquisa de satisfação no uso da ferramenta.

3. ESTUDO DE CASO PARA O USO DA CLCATCH

Para validar a eficiência da CLCATCH foi escolhida uma amostra de desenvolvedores, considerando a atividade de busca de *patches* de cada PP, feita de forma manual e utilizando a ferramenta. Foram escolhidos nove LPs, como participantes do estudo. A amostra foi escolhida aleatoriamente, e os participantes realizaram a aceitação e caracterização do estudo, por meio de formulários. O contexto da avaliação foi buscar os requisitos e *patches* de segurança, que são itens que o Google disponibiliza publicamente para comunidade de desenvolvedores e parceiros da Open Hands Alliance (Android Security Bulletin, 2019).

Segundo Jimenez *et al.* (2009), em DDS alguns fatores podem impactar na produtividade de desenvolvedores tais como: alto número de desenvolvedores, tempo, alta complexidade das tarefas, escopo, habilidades e tarefas manuais. Nesse contexto, para escopo do estudo foi escolhido o monitoramento do tempo, a eficácia para execução da tarefa e o total de PPs a serem detectadas, utilizando a forma manual e com a ferramenta. Após a implantação da ferramenta CLCATCH, foi feito o monitoramento do tempo de execução dos desenvolvedores no processo de *patches* presentes nas PPs, de forma manual. A Tabela 1 mostra o resultado considerando a identificação pelos desenvolvedores participantes no estudo.

Tabela 1. Desempenho dos Participantes do Estudo

LP	Tempo de busca (hh:mm:ss)	Detecção de <i>patches</i> na PP	Taxa de Eficácia
1	00:02:32	02	67%
2	00:01:34	03	100%
3	00:00:55	01	33%
4	00:02:28	03	100%
5	00:01:27	02	67%
6	00:03:09	03	100%
7	00:00:58	03	100%
8	00:01:02	02	67%
9	00:02:06	01	33%

Este estudo inicial foi a motivação para automatização da tarefa de buscas de PPs por meio de uma ferramenta de software. Diante desse cenário, foi construída a CLCATCH, para avaliar o impacto de contribuição da ferramenta foi feita uma pesquisa realizada qualitativa, realizada por meio de questionários. O objetivo foi identificar o grau de eficiência no processo de identificação. Assim, foi disponibilizado um projeto piloto, apenas para monitorar o tempo de busca manual, sem o uso da CLCATCH. A tarefa monitorada foi identificar três rótulos em três *patches* dentro de uma PP específica, utilizando o SVN como base para busca. Esta atividade foi feita antes da disponibilização da CLCATCH para a equipe de LPs.

O tempo médio para execução da tarefa foi de um minuto e 48 segundos. E a taxa de eficácia média, que é definida pela razão entre a detecção individual e o total de PPs (Brito *et al.* 2016; Fernandes *et al.* 2011; Coutinho, 2015), foi de 74%. O processo de detecção das PPs utilizando a CLCATCH demora em torno de cinco segundos para realizar a busca da PP e a extração das PPs correspondentes. Isto acaba influenciando no desempenho individual dos desenvolvedores para execução da atividade. Além de melhorar a precisão para identificação dos *patches* de atualização. Isto cria indícios de que a ferramenta melhora substancialmente a atividade no processo de *release* do Sidia.

Além dessa análise, foi feita a avaliação da percepção dos desenvolvedores participantes. Para coleta foi feito um questionário, no Google Forms. Os resultados são descritos na próxima Seção. Para coleta dos dados, foram elaboradas questões, usando como base o modelo TAM (*Technology Acceptance Model*). O TAM investiga a aceitação de uma tecnologia pelos usuários por meio da percepção sobre utilidade e percepção sobre facilidade de uso (Davis, 1989), este modelo foi utilizado como base para elaboração de questões sobre a interação dos usuários com a tecnologia.

4. RESULTADOS OBTIDOS

A análise qualitativa foi importante para identificar pontos de melhoria e avaliar a percepção da equipe, sobre o uso da ferramenta CLCATCH. A Tabela 2 apresenta os resultados do questionário TAM aplicado. Os resultados, coletados pelo formulário mostram resultados favoráveis ao uso da CLCATCH.

Tabela 2. Avaliação qualitativa dos participantes do estudo

Métrica	ID Perguntas	Itens de Verificação	Concordo totalmente	Concordo amplamente	Concordo parcialmente	Discordo parcialmente	Discordo amplamente	Discordo totalmente
Facilidade de Uso	Q1	Foi fácil de aprender a usar a CLCATCH	4	3	2	0	0	0
	Q2	Consegui usar a CLCATCH da forma que eu queria	4	1	3	1	0	0
	Q3	Eu entendia o que acontecia na minha interação com a CLCATCH	6	2	1	0	0	0
	Q4	É fácil ganhar habilidade no uso da CLCATCH	5	3	1	0	0	0
Desempenho	Q5	A ferramenta me permitiu realizar as buscas de forma mais rápida que as abordagens anteriormente utilizadas	6	2	1	0	0	0
	Q6	Considero que a CLCATCH melhora a minha atividade no processo de geração de <i>releases</i>	5	3	1	0	0	0
	Q7	Consegui buscar facilmente os pacotes que eu desejava	5	2	1	0	0	0
	Q8	CLCATCH melhora minha produtividade	6	2	1	0	0	0
Utilidade	Q9	Considero a CLCATCH útil	7	2		0	0	0
	Q10	Consegui indicar as modificações dos pacotes correspondentes identificadas pela ferramenta	6	2	1	0	0	0
	Q11	A CLCATCH aprimora a busca segura de pacotes em relação ao processo anteriormente utilizado	5	0	4	1	0	0
	Q12	A CLCATCH melhora o processo de geração de <i>releases</i>	5	2	1	1	0	0

Em relação à facilidade de uso, os desenvolvedores participantes concordaram sobre a facilidade de aprendizado no uso da CLCATCH (Q1), tendo todos os participantes com aceitação positiva para o uso da CLCATCH. Em relação a Q2, um desenvolvedor discordou parcialmente sobre o uso da CLCATCH e outros oito concordaram positivamente. Como sugestão o desenvolvedor que discordou parcialmente relatou “*Eu adotaria como funcionalidade um sort para patches encontradas na ferramenta*”. Em relação a Q3, houve aceitação positiva sobre interação com a ferramenta, tendo apenas um dos desenvolvedores concordado parcialmente com o uso da CLCATCH.

A análise qualitativa permitiu como sugestão capturar sugestões de melhoria como: “*Inserir hints para o campo de pesquisa das PPs*” outro desenvolvedor sugeriu “*Poderia retirar o campo de download e realizar esta tarefa de forma automática*”. Em relação a Q3 e Q4, da Tabela 2, os desenvolvedores tiveram facilidade no uso e ganho de habilidade na interação com a ferramenta proposta. No entanto, houve uma sugestão para melhorar a aplicação como: “*Relacionar os PPs com os patches correspondentes e ordená-las*” proposta por um participante do estudo.

Em relação ao desempenho, os desenvolvedores participantes foram parcialmente favoráveis, em relação a Q5, seis participantes concordaram amplamente que a ferramenta permite realizar a atividade de buscas de forma mais eficiente. Em relação a Q5 e Q6, os participantes concordaram que a ferramenta melhora a atividade dos LPs. No entanto, um dos participantes sentiu dificuldade no uso da CLCATCH (Q7), pois segundo o mesmo *“removeria a necessidade de download dos arquivos, presentes na versão atual da CLCATCH”*. Considerando a utilidade da ferramenta, os resultados mostram importante contribuição da CLCATCH. Os participantes relataram como pontos favoráveis *“Reduz a necessidade de outras ferramentas para filtrar patches em PPs”*. Outro participante relatou *“A ferramenta melhora consideravelmente, pois concentra toda a atividade dentro da mesma interface”*. Um dos participantes relatou ainda que *“A ferramenta reduz o tempo de busca, e elimina as falhas causadas pela ação humana nas buscas”*. Além disso, o estudo foi útil, pois permitiu homologar a ferramenta oficialmente para atividade de buscas de patches, para geração de releases de segurança. Este estudo permitiu novas funcionalidades como: 1) adicionar um tutorial a ferramenta; 2) importância da ferramenta para as atividades de LPs do Sidia. 3) reduzir a lista e agrupar os patches de atualização.

5. CONCLUSÃO

Este artigo apresenta um relato de experiência, sobre a adoção de uma ferramenta para melhoria de uma das atividades de LPs, em projetos de desenvolvimento de plataforma Android, executados no Sidia. Este processo de desenvolvimento de S.O. é feito pelo Sidia em conjunto com outros Institutos de P&D da Samsung, separados geograficamente e representa grande importância para o crescimento da empresa e da região Amazônica.

A CLCATCH foi criada em razão da grande demanda de projetos que o Sidia dispõe atualmente, como forma de automatizar uma atividade do processo de buscas de PPs, que contém modificações armazenadas patches. Tais modificações são de diversas características que podem ser desde correção de módulos do Android, para os dispositivos Samsung. Até atualizações de segurança homologadas pelo Google, que disponibiliza novos patches para os parceiros da Open HandSet Alliance.

Para a coleta de dados, foi feito questionários com o time de LPs, que realiza a gestão das tarefas, entregas e prazos para atualização dos dispositivos, bem como as tarefas necessárias para identificação dos requisitos de atualização dos S.Os. Os resultados mostram importante contribuição da ferramenta, com melhoria no tempo de atividades e taxa de eficácia média de 74%. Neste sentido, o estudo foi importante para identificar os problemas que são mais críticos no time de LP, bem como a criação de uma alternativa para otimizar uma das atividades de gestão das tarefas no ciclo de desenvolvimento de um dos produtos disponíveis nos dispositivos Samsung.

Como limitação podemos destacar a falta de um design experimental mais detalhado e um comparativo, focado nas atividades com e sem a ferramenta. Sabe-se que o uso de ferramentas em comparação à atividade manual possui grande disparidade (Fernandes, et al. 2011). No entanto, o estudo qualitativo foi útil para identificar as percepções de melhoria obtida com o uso, pelos LPs participantes.

Espera-se com isso contribuir para incentivar outras empresas e institutos de P&D a compartilhar suas experiências na área de Engenharia de Software. E com isso criar alternativas que permitam estreitar parcerias com a academia para melhoria de processo de software.

AGRADECIMENTO

Os autores agradecem a SAMSUNG Eletronics of Amazon Ltda, nos termos da Lei de Informática N° 8387/91.

REFERÊNCIAS

- Android Security Bulletin – Link: <https://source.android.com/security/bulletin/> access: 2019-08-08.
- Audy, J. Prikladnick, R., 2007. *Desenvolvimento Distribuído de Software - Desenvolvimento de Software com Equipes Distribuídas*. Série Campus. Elsevier Editora, Rio de Janeiro, Brasil.
- Bosnic, I. et al. 2011. *Avoiding Scylla and charybdis in distributed software development course. Proceedings of the 2011 Community Building Workshop on Collaborative Teaching of Globally Distributed Software Development*. New York, United States, 26–30.
- Brito, R.B.S et al. 2016. *Aprimorando um Jogo para Ensino do Método de Avaliação de Comunicabilidade por Meio de Estudos Experimentais*. XV Simpósio Brasileiro de Qualidade de Software / XV Brazilian Symposium on Software Quality Artigos Técnicos / Research Paper. Maceió, Brasil, 94-105.
- Coutinho, et al. 2011. *Improving an Educational Mobile Application through Usability Evaluation*. 2015. 9th International Technology, Education and Development Conference. Madrid, Espanha, 5812-5820
- Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 319-339.
- Fernandes, P et al. 2014, *A Usability Inspection Approach to Assist in the Software Development Process*. The Ninth International Conference on Software Engineering Advances. Nice, France, 651 -657.
- Jimenez, M. et al.2009. *Challenges and Improvements in Distributed Software Development: A Systematic Review* – In *Advances in Software Engineering Vol 2009*, 1-14, doi:10.1155/2009/710971
- Lima, A. Costa, B. França, 2016, *Gerência Flexível de Processos de Software com o Ambiente WebAPSEE*. XIII Sessão de Ferramentas–XX SBES.
- Neto, C.R.L. et al., 2012. *Avaliando a Qualidade do Software de um Projeto Distribuído: um Relato de Experiência*. XI Simpósio Brasileiro de Qualidade de Software. 373-380, Fortaleza, Brasil,
- Stawnicza. O. 2015. *Distributed team cohesion – not an oxymoron. The Impact of information and communications technologies on teamness in globally distributed IT projects*. *International Journal of Information Systems and Project Management*. Available in: <http://www.sciencesphere.org/ijispm/archive/ijispm-030202.pdf>
- Weimann, M. P. et al. 2013 *Enhancing team performance through tool use: How critical technology related issues influence the performance of virtual project teams*. *IEEE Transactions on Professionals Communication*.