

Elementos que Impactam o Planejamento de Testes em Ambientes de Desenvolvimento Distribuído no Contexto de Ecosystemas de Software

Nayane Maia¹, Rodrigo Santos², Elisa Huzita³, Arilo Dias-Neto¹, Claudia Werner²

¹Instituto de Computação (IComp), Universidade Federal do Amazonas – Manaus – AM

²PESC/COPPE, Universidade Federal do Rio de Janeiro – Rio de Janeiro – RJ

³Departamento de Informática, Universidade Estadual de Maringá – Maringá – PR

{nayane.maia,arilo}@icomp.ufam.edu.br, {rps,werner}@cos.ufrj.br,
emhuzita@din.uem.br

Resumo. *A organização de Ecosystemas de Software (ECOSs) na indústria vem interferindo nos projetos com Desenvolvimento Distribuído de Software (DDS), uma vez que demanda novos modelos abertos de negócio, que geram novos papéis e padrões para colaboração, inovação e proposição de valor. Este cenário requer atenção aos testes, tidos como a segunda maior atividade terceirizada, dado que seu planejamento e controle visam assegurar o sucesso de projetos de DDS. Este artigo tem por objetivo identificar e classificar elementos que impactam o planejamento de testes em ambientes de DDS a partir de uma agenda de pesquisa, assim como discutir seus desdobramentos no contexto dos ECOSs, contribuindo para pesquisas envolvendo esses temas.*

1. Introdução

O Desenvolvimento Distribuído de Software (DDS) tem sido adotado pelas empresas em busca de redução de custos por meio do melhor aproveitamento de recursos humanos, distribuídos ao redor do mundo. Este cenário impacta diretamente a pesquisa e prática da Engenharia de Software (ES), considerando que os desafios do DDS se relacionam com o tratamento das questões econômicas e sociais da ES (Boehm, 2006). No entanto, o DDS não vem se limitando ao desenvolvimento de projetos ou produtos de software únicos, mas incluindo o contexto de arquiteturas e/ou plataformas comuns e/ou relacionadas, com a participação de desenvolvedores externos e usuários, conhecidos ou não (Santos & Werner, 2012).

Conhecidas como Ecosystemas de Software (ECOSs), essas redes de atores e de artefatos formadas sobre uma tecnologia de software central passam a interferir nos projetos de DDS, pois requerem novos modelos abertos de negócio que geram papéis e padrões para colaboração, inovação e proposição de valor. Especificamente entre os processos de ES, um estudo revela que o teste é a segunda maior atividade terceirizada, após a codificação (Shah *et al.*, 2011), de modo que planejamento e controle são necessários para assegurar o sucesso de projetos de DDS (Collins *et al.*, 2012). Assim, torna-se importante compreender que elementos impactam o planejamento de testes nas combinações de cenários de DDS em geral e que são afetados pelo contexto dos ECOSs.

Este artigo tem por objetivo identificar e classificar elementos que impactam o planejamento de testes em ambientes de DDS a partir de uma agenda de pesquisa,

proposta por Huzita *et al.* (2012), assim como discutir seus desdobramentos no contexto dos ECOSs (Santos & Werner, 2012). O artigo está dividido da seguinte forma: a Seção 2 discute os ECOSs e o DDS; a Seção 3 apresenta uma classificação dos elementos que impactam o planejamento de testes em ambientes de DDS; a Seção 4 discorre sobre os desdobramentos dos ECOSs; e a Seção 5 conclui o artigo, enunciando trabalhos futuros.

2. Ecossistemas de Software e o Cenário de DDS

O DDS tem sido cada vez mais adotado pelas empresas, de forma que desenvolver software se torna uma tarefa quase ininterrupta. No entanto, devido às diferenças culturais, políticas e organizacionais, bem como às distâncias temporal e espacial, emergem também desafios relacionados a cooperação, coordenação e comunicação. A fim de reduzir estas diferenças e distâncias e assegurar que indivíduos geograficamente distribuídos estão colaborando nos projetos de software, é essencial uma infraestrutura que garanta a troca de informação e de conhecimento entre os envolvidos (Chaves *et al.*, 2010). Nesse sentido, Huzita *et al.* (2012) discutem uma agenda de pesquisa envolvendo gerenciamento de informação contextual e de conhecimento em DDS (Tabela 1).

Tabela 1 – Agenda de pesquisa para DDS. Fonte: (Huzita *et al.*, 2012)

ID	Desafio	Descrição
A1	<i>Explorar gestão de conhecimento em projetos de DDS</i>	Identificar, extrair, ordenar, evoluir e analisar a importância e impacto de conhecimentos relacionados a domínio, processo de desenvolvimento, recursos técnicos, recursos não técnicos e tecnologias.
A2	<i>Definir mecanismos para apoiar a captura de informação contextual que possa vir de fontes heterogêneas</i>	Como o cenário considerado é o de DDS, os dados poderão ser provenientes de vários locais. Além disso, é importante também estabelecer a frequência e formato que estes dados serão capturados.
A3	<i>Explorar computação em nuvem sob os aspectos de: software como serviço, plataforma como serviço e infraestrutura como serviço</i>	Pelo volume de dados gerados, é importante que sejam usadas tecnologias que garantam confiabilidade e disponibilidade de informação. O foco seria explorar a definição de estratégias adequadas para armazenar, nas nuvens, informações contextuais provenientes de diferentes locais.
A4	<i>Explorar meios adequados para a representação, processamento e disseminação de conhecimento e informação contextual</i>	Uma ideia é fazer uso de ontologia (Chaves <i>et al.</i> , 2010). Possíveis benefícios: definição de um vocabulário de consenso; possibilidade de desenvolver software mais complexo ao associar técnicas de inferência; e reutilização do conhecimento modelado em novas aplicações.
A5	<i>Explorar meios para apresentação, visualização e disseminação de informação contextual de acordo com o objetivo, perfil e formato</i>	A ideia é estudar técnicas adequadas que permitam a visualização, disseminação e apresentação de informações de acordo com parâmetros e características que sejam de interesse no cenário de DDS.
A6	<i>Mapear os aspectos socioculturais capturados em projetos DDS para cada uma das fases do modelo de conversão de conhecimento</i>	Estudar como as informações socioculturais capturadas durante o desenvolvimento de software podem ser mapeadas ou alinhadas com as diferentes fases do modelo de conversão de conhecimento: socialização, externalização, combinação e internalização.
A7	<i>Definir indicadores e o subsequente gerenciamento de desempenho em DDS</i>	Definir as dimensões de desempenho para o processo de tomada de decisão e, também, para a contínua melhoria de processo e produto em DDS, a fim de projetar mecanismos para apoiar o desenvolvimento e a reutilização de conhecimento.
A8	<i>Prover o tomador de decisão com ferramentas para gestão de informação de contexto e de conhecimento</i>	Desenvolver ferramentas que ofereçam o apoio necessário para capturar, recuperar, reutilizar e integrar o conhecimento e as informações de contexto geradas em DDS.

Paralelamente, o desenvolvimento de software requer pensar cuidadosamente as plataformas que vão apoiá-lo e as redes de artefatos e envolvidos, i.e., relacionamentos de conectividade e dependência entre produtos e organizações, que afetam (e são afetados por) esse cenário (Jansen *et al.*, 2009). Isso corrobora com o fato de que as inovações não mais se originam de uma organização, mas da chamada “co-inovação” a partir de diferentes agentes da indústria de software, além dos processos denominados de “co-evolução”, possíveis somente a partir da reunião de organizações colaborativamente focadas em dar suporte a novos produtos, a satisfazer as necessidades dos clientes e a incorporar novas rodadas de inovação (Arndt & Dibbern, 2006).

Neste caso, o sistema de software representa uma combinação de software,

hardware e “peopleware”, constituída sobre um ambiente comum, i.e., uma plataforma. Pode-se identificar, então, uma tendência na ES: o nascimento, desenvolvimento, amadurecimento e eventual “morte” ou transformação de plataformas levam à “desconstrução” da visão tradicional da ES. Privilegia-se assim uma nova visão, que contempla a colaboração e a interoperabilidade entre os atores (Campbell & Ahmed, 2010). Na década de 2000, essas estruturas foram alvo de pesquisa na ES, chamadas de Ecossistemas de Software ou ECOSs (Messerschmitt & Szyperski, 2003). Segundo Hanssen & Dyba (2012), um ECOS consiste em uma comunidade de atores e de organizações em rede, que apoia as relações entre eles sobre um interesse comum no desenvolvimento e no uso de uma tecnologia de software central.

Inspirado nos ecossistemas biológicos, sociais e de negócios, o tema ECOSs favorece o uso de analogias para aplicar novas visões e abstrações sobre os desafios da indústria de software (Jansen *et al.*, 2013). Por exemplo, a influência da visão de negócio pode ser notada no fato de que as equipes de desenvolvimento têm trabalhado geograficamente dispersas. Isso permite a expansão da indústria para novos mercados, embora necessite de um olhar para questões socioculturais (Pinto *et al.*, 2009). Como um dos desafios, Santos *et al.* (2012) apontam a decisão sobre qual é a estratégia mais inovadora no desenvolvimento de software globalizado. É importante mapear os pontos fortes e fracos das plataformas de ECOSs para definir e automatizar um *framework* para o seu gerenciamento e monitoração, local e globalmente.

Emergem três tendências que aceleram a complexidade do novo contexto da ES (Bosch & Bosch-Sijtsema, 2010): (1) ampla adoção de *Linhas de Produtos de Software* (LPSs): como expor os ativos reutilizáveis e abrir a arquitetura da plataforma; (2) *Globalização do Desenvolvimento de Software* (GDSs): como gerenciar a dependência socio-técnica em suas redes de produção; e (3) *organização de ECOSs* em comunidades de desenvolvedores externos sobre produto de sucesso: como entender as relações entre as partes sob a perspectiva da reutilização mais ampla. Nesse sentido, Santos & Werner (2012) propõem um *framework* para classificar a iniciativas de pesquisa em ECOSs em quatro dimensões de análise, chamado *ReuseECOS*, três destas inicialmente exploradas por Campbell & Ahmed (2010). As duas primeiras (transacional e técnica) são normalmente alvo de desejo de controle pelo centralizador, ao passo que a terceira (social) é dinâmica e a “pedra angular” para o sucesso do ECOS, ao impactar as fases do seu ciclo de vida, estruturado pela quarta (engenharia e gerenciamento):

- *técnica*: dimensão voltada ao entendimento da abertura das plataformas e do envolvimento de atores externos. É dirigida por três fatores: (1) engenharia da plataforma (sistema de software); (2) arquitetura (estrutura dos componentes, seus relacionamentos e princípios e diretrizes que norteiam a sua evolução); (3) gerenciamento de requisitos (funcionalidades comuns ou específicas);
- *transacional*: dimensão voltada ao entendimento da colaboração e/ou competitividade, e nas expectativas dos envolvidos e o impacto na produtividade no ambiente destes. É dirigida por três fatores: (1) visão do ambiente; (2) inovação; e (3) planejamento estratégico;
- *social*: dimensão voltada ao entendimento da abertura da empresa como um todo, nas comunidades originadas ao redor das plataformas e no desenvolvimento colaborativo. É dirigida por três fatores: (1) utilidade (compensações e recompensas, esperadas e percebidas pelos membros); (2) promoção

- (reconhecimento implícito/explicito de capacidades/habilidades do colaborador); e (3) ganho de conhecimento (armazenamento e manutenção de experiências, e oportunidades para as partes interessadas se envolverem com novas tecnologias);
- *engenharia e gerenciamento*: dimensão focada em intercambiar e integrar as três dimensões apresentadas a fim de viabilizar uma infraestrutura de apoio. Utiliza as relações que correspondem a: (1) motivar o desenvolvimento e evolução da plataforma (**técnica-transacional**); (2) contribuir para o estabelecimento da plataforma (**técnica-social**); e (3) mapear as proposições e percepções de valor (**transacional-social**).

3. Práticas em Ambientes de DDS Aplicáveis no Planejamento de Testes

Segundo Juristo *et al.* (2004), o teste de software é considerado uma das práticas mais custosa do processo de desenvolvimento e, por isso, requer recursos adequados, cuja utilização efetiva necessita de um bom planejamento e controle (McGregor & Sykes, 2001). O planejamento garante que o critério de teste seja especificado e o conjunto de casos e procedimentos de testes seja desenvolvido antes do fim da fase de implementação do produto, evitando testes tendenciosos. Por sua vez, a atividade de controle assegura que os testes planejados sejam monitorados constantemente e que seus resultados sejam registrados. Alguns trabalhos discutidos a seguir tratam da atividade de teste de software no processo de DDS, destacando o planejamento e controle como fatores de impacto no sucesso dos projetos.

No estudo de Collins *et al.* (2012), foram apresentados os desafios e as lições aprendidas no contexto de um projeto de desenvolvimento ágil utilizando testes distribuídos. Constatou-se neste estudo que algumas práticas de planejamento e controle asseguram o sucesso do projeto, tais como: (i) comunicação e coordenação são fatores essenciais para o sucesso de testes distribuídos; (ii) as informações do projeto devem estar disponíveis com detalhes a todos os envolvidos; (iii) a automação reduz as necessidades de presença física no processo de teste; e (iv) as ferramentas de apoio são sempre importantes, mas a organização da equipe de teste é mais significativa. Por sua vez, em Causevic *et al.* (2010), constatou-se, entre outras coisas, que a ausência de infraestrutura é considerada um fator importante para retardar a implantação de software em projetos de desenvolvimento distribuído.

Maia *et al.* (2012) apresentam um relato de experiência no contexto de um projeto que executou testes ágeis com equipes distribuídas, composto por uma equipe de teste local e outra remota, mostrando que a atividade de teste requer colaboração e cooperação entre os envolvidos. No entanto, esta colaboração e cooperação se tornam mais difíceis quando os membros da equipe de teste estão separados geograficamente da equipe de desenvolvimento. Para contornar a distância de parte da equipe de testes, foi necessário fornecer uma infraestrutura de testes, que permitisse o acesso às informações do projeto por todos os envolvidos de forma eficiente, constituída por: ferramenta de gestão de tarefas Scrum (*FireScrum*), ferramenta de gestão de teste (*TestLink*), ferramenta de gestão de defeitos (*Mantis*), ferramenta de versionamento (*Subversion*), servidor de aplicação para testes e ferramentas de comunicação (*e-mail* e *chats*).

A fim de compreender os temas afins que impactam os testes em DDS, a Tabela 2 reúne os desafios encontrados durante esta fase, a partir dos estudos de Jiménez *et al.* (2009) e Rocha *et al.* (2010), e mostra o seu cruzamento com as práticas a serem

aplicadas em DDS, extraídas de (Huzita *et al.*, 2008): (P1) oferecer mecanismos para facilitar a comunicação; (P2) desenvolver um modelo de produtos; (P3) gerenciar processo/projeto; (P4) praticar a co-alocação temporária; (P5) estabelecer critérios para constituição de equipes e estabelecer senso de equipe; (P6) disponibilizar e compartilhar informações de projeto; (P7) lidar com heterogeneidade; (P8) distribuir responsabilidades; (P9) apoiar a colaboração por meio de *awareness* e *group awareness*; (P10) distribuir atividades; (P11) definir métricas; e (P12) estabelecer senso de confiança. Essas práticas resultaram de uma análise considerando os desafios em DDS, tais como aqueles relativos às distâncias observadas (geográfica, cultural e temporal) e, também, aqueles relacionados a coordenação, colaboração e controle.

Tabela 2 – Desafios introduzidos pelo DDS na atividade de testes

Temas Afins	Desafios dos Testes em DDS	Práticas
<i>Comunicação</i>	Demora para resposta via <i>e-mails</i> . Idioma utilizado. Ausência de contato visual.	P1, P4, P9, P12
<i>Stakeholders</i>	Definição de responsabilidades. Dificil determinação da fronteira entre as atividades dos membros.	P5, P8, P9, P10
<i>Plano de Testes</i>	Definir planos de testes e executá-los, de forma que todos os envolvidos participem efetivamente dos testes.	P2, P3, P6
<i>Ambiente Técnico</i>	Prover uma infraestrutura computacional adequada para apoiar os testes e ter um nível de conhecimento técnico em todos os sítios (locais) envolvidos.	P1, P7, P11
<i>Complexidade</i>	Decidir como dividir o trabalho em diversas equipes ou centros distribuídos geograficamente, considerando os diferentes níveis técnicos das equipes.	P5, P6, P8, P9, P10,
<i>Equipes</i>	Dependência de equipes externas.	P1, P4, P9, P12
<i>Gestão de Conhecimento</i>	Compartilhar informação entre os membros do projeto e manter a documentação constantemente atualizada.	P3, P6, P8, P9
<i>Gestão do Projeto/Processo</i>	Processos diferentes aplicados nos diferentes sítios.	P2, P3, P5, P12
<i>Questões Culturais</i>	Unificar as culturas existentes nos diferentes sítios onde os testes são aplicados.	P1, P9, P12

A título de ilustração, ao considerar os temas *stakeholders*, *equipes* e *questões culturais* presentes na Tabela 2, verifica-se que uma forma de mitigá-los seria aplicar as práticas P1, P4, P5, P8, P9, P10, P12, o que consiste na união das práticas associadas a estes três temas. Pode ser observado ainda que, na junção destes temas, o interesse é que se consiga uma equipe de teste com os membros distribuídos geograficamente, que possa gerar informações contextuais que devam ser adequadamente capturadas e usadas. Isto pode motivar pesquisas que vão de encontro ao item A2, presente na Tabela 1.

A partir da análise dos trabalhos relacionados, é possível perceber que um dos desafios em DDS é a comunicação. Para se estabelecer a comunicação, é necessária uma infraestrutura adequada, quer seja de redes como também de ferramentas de apoio. No entanto, apenas a disponibilidade de infraestrutura não seria suficiente pois as pessoas envolvidas nas equipes necessitam deter um conhecimento técnico específico, particularmente em testes, em nível adequado e suficiente para que se estabeleça a colaboração. Neste contexto, Shah *et al.* (2011) descrevem um estudo em que o teste distribuído de software é estruturado em torno de três fases: preparação dos testes, execução dos testes e gestão de clientes, surgindo a necessidade de se criar mais um papel (gestor de cliente), que normalmente não existe no processo tradicional de desenvolvimento. Este, por sua vez, irá funcionar como uma “ponte” entre a equipe de testes e o cliente. Por se tratar de DDS, estas pessoas podem estar em locais distintos e atribuir tarefas a pessoas com quem muitas vezes não se teve ainda a oportunidade para um encontro face a face – bem diferente de atribuir tarefas quando se conhece a pessoa.

O estudo de Jain *et al.* (2011) revela que em DDS o time de teste passa a ter um papel mais estratégico (planejar e controlar) do que meramente técnico (executar e reportar defeitos). Portanto alocar pessoas para testes e acompanhar o trabalho delas

requer um cuidado maior. Com isso, pode-se dizer que que todas as fases do desenvolvimento sofrem algum impacto quando o contexto é DDS. Contudo, a fase de testes possui suas especificidades, pois depende diretamente dos resultados providos em outras fases do processo de desenvolvimento para finalizar a atividade de validação.

4. Desdobramentos do Contexto de Ecossistemas de Software

Partindo dos desafios introduzidos pelo DDS nas atividades de teste, os temas afins identificados podem ser inicialmente classificados nas quatro dimensões do *framework ReuseECOS*, conforme ilustra a Figura 1. Na dimensão técnica, o tema *ambiente técnico* é considerado por tratar da concepção de uma infraestrutura de apoio aos testes em DDS, que se relaciona diretamente com engenharia da plataforma e com o suporte ao envolvimento de atores externos frente a certificação e testes de suas contribuições na arquitetura da plataforma. Na dimensão transacional, ressalta-se o tema *comunicação*, por representar um elemento chave para viabilizar a colaboração entre equipes de teste distribuídas, tanto dos componentes, serviços e aplicações desenvolvidos por terceiros como na sua integração e/ou execução sobre a plataforma. Na dimensão social, o tema *questões culturais* demanda atenção especial, sobretudo pelo tratamento dos diversos desenvolvedores e usuários da plataforma, conhecidos ou não, a fim de mantê-los ativos.

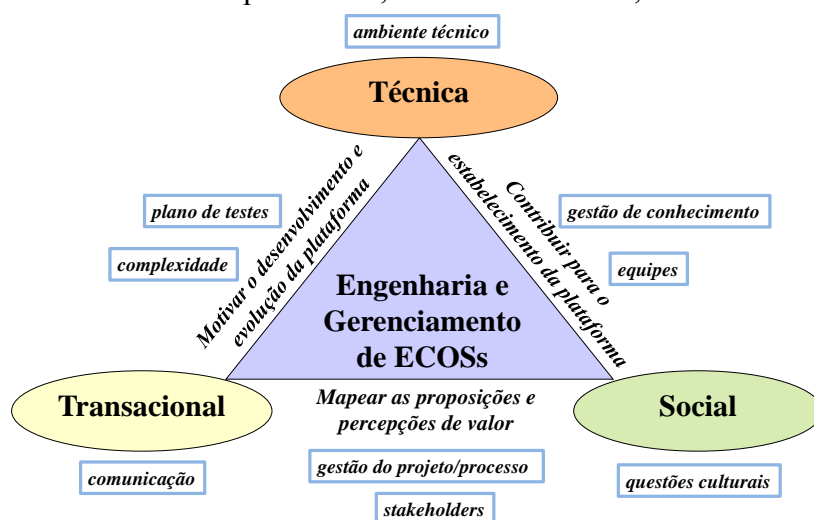


Figura 1 – Temas afins dos desafios introduzidos pelo DDS na atividade de teste e sua classificação nas dimensões de ECOSs

A dimensão engenharia e gerenciamento permite a compreensão dos elementos que existem nas relações entre as outras dimensões. Na relação técnica-transacional, dois temas são discutidos: *plano de testes* e *complexidade*. Ambos os temas aparecem nesta relação por dependerem diretamente do modelo de negócio de DDS adotado pelo centralizador da plataforma de um ECOS, i.e., empresa que lidera o ECOS (e.g., Apple, Microsoft, Nokia). Por exemplo, segundo Angeren *et al.* (2011), caso o modelo seja de parceria, os desenvolvedores externos atuam somente na plataforma do ECOS, o que permite gerir de maneira mais padronizada o planejamento e execução dos testes. Por outro lado, caso o modelo seja de membro, os desenvolvedores podem atuar em diversas plataformas, o que pode impactar as atividades de teste pelas experiências vividas pelas equipes na atuação em diferentes ECOSs. A complexidade pode ser gerida por meio de políticas e *roadmaps* em estratégias de governança (Albert *et al.*, 2012), podendo produzir desdobramentos sobre a comunicação e, conseqüentemente, sobre a interação

humana (complexa por natureza), atingindo, assim, a dimensão social.

Na relação técnica-social, os temas *gestão de conhecimento e equipes* aparecem devido ao fato de que recursos, informações e artefatos disponíveis e/ou compartilhados na plataforma de um ECOS devem ser gerenciados e mantidos, não somente pelo centralizador da plataforma como também pela comunidade subjacente. Novamente, a dimensão social pode ser alcançada pois pensar a gestão de conhecimento no ECOS envolve pensar a forma com as transações acontecem. Com a existência de equipes de desenvolvimento distribuídas, as equipes de teste, também distribuídas, devem ser capazes de realizar um planejamento dos testes, que possa permitir e certificar as contribuições dos diversos *stakeholders*, como o caso da AppleStore. As redes que se formam entre atores e artefatos devem ser identificadas e analisadas com base na rede sócio-técnica (Lima *et al.*, 2013), a fim de verificar quais são as equipes com maior potencial de impacto sobre a plataforma por desenvolverem o maior percentual de contribuições e/ou por requererem maior demanda de testes. Isso pode mostrar também os principais benefícios e fraquezas em um ambiente de DDS no contexto de ECOSs.

Por fim, na relação transacional-social, os temas *gestão do projeto/processo e stakeholders* emergem como cruciais para o sucesso de projetos com DDS em ECOSs, pois isso requer uma análise e manutenção do mapeamento entre as proposições e percepções de valor dos *stakeholders*, visando maximizar esta relação (Santos & Werner, 2012). No caso, a gestão deve considerar os diferentes níveis em que os ECOSs podem ser observados (Jansen *et al.*, 2009), i.e., o nível da organização (aquela que centraliza a plataforma), o nível da rede de produção de software (as organizações que fornecem e que consomem da plataforma) e o nível dos ECOS (os concorrentes e demais ECOSs que coexistem na indústria de software globalizada). Para isso, um *framework* para monitoramento e controle dos testes deve ser estabelecido, tanto considerando o nível local (desenvolvedores externos e usuários) como global (organização centralizadora), a fim de manter a qualidade dos produtos e serviços construídos a partir da plataforma. Estabelece-se assim a ligação entre as questões culturais e a comunicação, dado que elas impactam a forma das pessoas de comunicarem.

5. Conclusão

Os desafios do DDS ressaltam o tratamento das questões econômicas e sociais da ES e ainda incluem o contexto das arquiteturas e/ou plataformas comuns e/ou relacionadas com a formação de redes de desenvolvedores externos e usuários, conhecidos ou não, gerando os ECOSs. Neste contexto, o sucesso de projetos com DDS depende do planejamento e controle dos testes, de modo que se torna necessário compreender que elementos impactam o planejamento de testes nas diferentes combinações de cenários de DDS em geral e que também são afetados pelo contexto dos ECOSs. Este artigo identificou e classificou elementos que impactam o planejamento de testes em ambientes de DDS a partir de uma agenda de pesquisa, e também discutiu seus desdobramentos no contexto dos ECOSs. Considerando a carência de pesquisas sobre testes e DDS em ECOSs, este trabalho contribui no sentido de integrar grupos de pesquisa em torno desses temas a fim de construir soluções nessa direção. Como trabalhos futuros, pretende-se adaptar um processo de teste para contemplar os elementos discutidos nas Seções 3 e 4, indicando artefatos, papéis, atividades e ferramentas que possam apoiar o tratamento de questões econômicas e sociais no planejamento de teste com DDS.

Referências

- Albert, B., Santos, R., Werner, C. (2012) “A Study on Software Components Governance Based on SOA Governance Elements”. In: VI SBCARS, Natal, Brasil, 120-129.
- Angeren, J., Kabbedjik, J., Jansen, S., Popp, K.M. (2011) “A Survey of Associate Models used within Large Software Ecosystems”. In: 3rd IWSECO, IC SOB Workshops, Brussels, Belgium, 27-39.
- Arndt, J., Dibbern, J. (2006) “Co-Innovation in a Service Oriented Strategic Network”. In: IEEE International Conference on Services Computing, Chicago, USA, 285-288.
- Bosch, J., Bosch-Sijtsema, P. (2010) “From Integration to Composition: On the Impact of Software Product Lines, Global Development and Ecosystems”, *Journal of Systems and Software* 83, 1, 67-76.
- Causevic, A., Sundmark, D., Punnekkat, S. (2010) “An Industrial Survey on Contemporary Aspects of Software Testing”. In: 3rd International Conference on Software Testing, Verification and Validation, Paris, France, 393-401.
- Campbell, P., Ahmed, F. (2010) “A Three-dimensional View of Software Ecosystems”. In: 2nd IWSECO, ECSA Workshops, Copenhagen, Denmark, 81-84.
- Chaves, A., Huzita, E., Vieira, V., Steinhacher, I. (2010) “A Context Conceptual Model for a Distributed Software Development Environment”. In: 22nd SEKE, Redwood City, USA, 437-442.
- Collins, E., Macedo, G., Maia, N., Dias-Neto, A. (2012) “An Industrial Experience on the Application of Distributed Testing in an Agile Software Development Environment”. In: 7th IEEE ICGSE, Porto Alegre, Brasil, 190-194.
- Hanssen, G., Dyba, T. (2012) “Theoretical Foundations of Software Ecosystems”. In: 4th IWSECO, IC SOB Workshops, Boston, USA, 6-17.
- Huzita, E., Silva, C., Wiese, I., Tait, T., Quinaia, M., Schiavoni, F. (2008) “Um Conjunto de Soluções para Apoiar o Desenvolvimento Distribuído de Software”. In: II WDDS, Campinas, Brasil, 101-110.
- Huzita, E., Leal, G., Balancieri, R., Tait, T., Cardoza, E., Pentead, R., Vivian, R. (2012) “Knowledge and Contextual Information Management in Global Software Development: Challenges and Perspectives”. In: VI WDDS, ICGSE Workshops, Porto Alegre, Brasil, 43-48.
- Jain, R.; Poston, R. & Simon, J. (2011) “An Empirical Investigation of Client Managers’ Responsibilities in Managing Offshore Outsourcing of Software-Testing Projects”, *IEEE Transactions on Engineering Management*, v. 58, n. 4 (November), 743-757.
- Jansen, S., Cusumano, M., Brinkkemper, S. (eds.) (2013) “Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry”. Edward Elgar Publishing.
- Jansen, S., Finkelstein, A., Brinkkemper, S. (2009) “A Sense of Community: A Research Agenda for Software Ecosystems”. In: 31st ICSE, Vancouver, Canada, 187-190.
- Jiménez, M., Piattini, M., Vizcaíno, A. (2009) “Challenges and Improvements in Distributed Software Development: A Systematic Review”, *Advances in Software Engineering*, 3 (Jan), 1-16.
- Juristo, N., Moreno, A., Vegas, S. (2004) “Reviewing 25 Years of Testing Technique Experiments”, *Empirical Software Engineering* 9, 1-2 (Mar), 7-44.
- Lima, T., Santos, R., Werner, C. (2013) “Apoio à Compreensão das Redes Socio-técnicas em Ecosystemas de Software”. In: II BraSNAM, XXXIII CSBC Workshops, Maceió, Brasil.
- McGregor, J., Sykes, D. (2001) “A Practical Guide to Testing Object-oriented Software”.
- Maia, N., Dias Neto, A., Collins, E., Macedo, G. (2012) “Aplicando Testes Ágeis com Equipes Distribuídas: Um Relato de Experiência”. In: XI SBQS, Fortaleza, Brasil, 365-372.
- Messerschmitt, D., Szyperki, C. (2003) “Software Ecosystem: Understanding an Indispensable Technology and Industry”. The MIT Press.
- Pinto, A., Almeida, A., Morais, E. (2009) “Collaborative Software Development Process for Geographically Distributed Teams”. In: III WDDS, Fortaleza, Brasil, 89-97.
- Rocha, R., Costa, C., Prikladnicki, R., Azevedo, R., Farias Jr, I., Meira, S. (2010) “Desafios nas Fases do Ciclo de Vida de Projetos Distribuídos”. In: IV WDDS, CBSOft Workshops, Salvador, Brasil, 34-41.
- Santos, R., Werner, C. (2012) “ReuseECOS: An Approach to Support Global Software Development through Software Ecosystems”. In: VI WDDS, ICGSE Workshops, Porto Alegre, Brasil, 60-65.
- Santos, R., Werner, C., Barbosa, O., Alves, C. (2012) “Software Ecosystems: Trends and Impacts on Software Engineering”. In: XXVI SBES/ST, Natal, Brasil, 206-210.
- Shah, H., Sinha, S., Harrold, M. (2011) “Outsourced, Offshored Software-Testing Practice: Vendor-Side Experiences”. In: 6th IEEE ICGSE, Helsinki, Finland, 131-140.