

M3DS: um modelo de dinâmica de desenvolvimento distribuído de software

Alexandre L'Erario¹

¹Coordenação de Informática – Universidade Tecnológica Federal do Paraná (UTFPR)
Cornélio Procópio, Paraná CEP 86300-000, Brasil

alerario@utfpr.edu.br

Abstract. *This work presents a dynamic model of distributed development of software, whose objective is to represent the reality and the aspects of DDS environments. With this model it is possible to understand the operation of a distributed project, aiding in the effectiveness of the manager of the network production and people can obtain a precise positioning in network. The results presented in this work answer the question: how the development software organizations produce software in a distributed way. The originality of the research is the construction of a model of dynamics of the distributed development elaborated from data of six cases studies.*

Resumo. *Este trabalho apresenta um modelo de dinâmica de desenvolvimento distribuído de software, cujo objetivo é representar a realidade e os aspectos de ambientes de DDS (Desenvolvimento distribuído de software). Com este modelo, é possível compreender o funcionamento de um projeto distribuído e auxiliar na eficácia da gestão da rede de produção, além de auxiliar as demais entidades e pessoas envolvidas a obterem um posicionamento na rede mais preciso. Os resultados apresentados neste trabalho respondem a questão de como as organizações desenvolvedoras de software produzem software de maneira distribuída. A originalidade da pesquisa centra-se na construção de um modelo de dinâmica do desenvolvimento distribuído elaborado com os dados levantados a partir de seis estudos de casos.*

1. Introdução

Uma maneira de estabelecer parceria entre as empresas que produzem software é constituir redes de produção. Segundo Alstyne (1997), uma rede pode ser definida a partir dos elementos envolvidos na sua estrutura, pelo seu processo e seu propósito.

As parcerias nem sempre são locais. São compostas por várias organizações distantes fisicamente umas das outras. Esse fato ocorre porque forças econômicas atualmente tendem a migrar para mercados globais, ultrapassando barreiras e desenvolvendo novas formas de competição e cooperação. O resultado, segundo Herbsleb e Grinter (1999), é a criação de organizações virtuais que podem operar com sucesso sobre as barreiras geográficas e culturais.

O desenvolvimento global de software, abordado neste trabalho com um tipo de desenvolvimento distribuído de software, tem tomado grandes proporções, pois as organizações das mais diversas categorias descobrem constantemente que o desenvolvimento distribuído de software pode aumentar a produtividade e reduzir o

custo. Por isso, segundo Carmel e Agarwal (2001), o número de entidades que se engajaram no GSD (*Global Software Development*) é crescente. O desenvolvimento distribuído de software agrega várias vantagens sobre o desenvolvimento centralizado, como ensinam Battin et al (2001), Carmel e Argawall (2001), L’Erario et al(2004) em seus estudos de caso.

O objetivo deste trabalho é elaborar um modelo da dinâmica de funcionamento de uma rede de produção de software. Este auxilia a compreensão de como a produção distribuída de software ocorre e identifica os principais pontos de possíveis impasses. Tal modelo é denominado neste trabalho como M3DS: modelo de dinâmica do desenvolvimento distribuído de software.

Este artigo está organizado da seguinte maneira: na seção 2 é apresentada uma revisão bibliográfica sobre DDS; na seção 3 é apresentada a metodologia; a seção 4 apresenta o M3DS; a seção 5 apresenta as conclusões e finalmente as referências bibliográficas.

2. Desenvolvimento Distribuído de Software

Segundo Audy e Prikladnicki (2007), o desenvolvimento distribuído de software é caracterizado quando um dos atores no envolvidos projeto estiverem fisicamente distante dos demais. Neste cenário, a complexidade do processo de desenvolvimento se amplia.

Dividir a produção de um software entre diversos nós envolve uma série de problemas. Alguns problemas que, localmente, são insignificantes ganham proporção quando uma tarefa é dividida entre organizações fisicamente distantes. Esses problemas são relacionados a atividades do desenvolvimento do software, como é o caso do levantamento de requisitos. Vários autores, como Edwards e Sridhar (2002) e Campbell e Van de Walle (2003) citam em seus trabalhos os problemas inerentes ao desenvolvimento distribuído.

Um ambiente de DDS agrega um conjunto de variáveis multidisciplinares que abordam desde a visão técnica de desenvolvimento até mesmo a visão de organização em rede. Por isso, para este trabalho foram consideradas variáveis como: **a dispersão física**, explicados por Audy e Prikladnicki (2007); **o modelo de negócio de DDS**, explicado por Robinson et. al. (2004); **o modelo de coordenação**, abordado sobre o ponto de vista gerencial - Mintzberg(2003) e Burton(1998) - e sobre o ponto de vista técnico - Cataldo(2007), Borden(2007) e Sinha (2007).

3. Metodologia

A metodologia empregada neste trabalho é o estudo de múltiplos casos – delineamento explanado por Yin (2005) - que consiste na aplicação do protocolo do delineamento estudo de caso em várias organizações que praticam o desenvolvimento distribuído de software. Segundo Yin (2005), o estudo de caso trabalha com o como e o porquê dos fenômenos, sem exigir controle sobre os eventos comportamentais. Avalia eventos contemporâneos e possibilita a replicação literal do estudo de caso, sobre o aspecto de resultados semelhantes. Os casos abordados neste projeto de pesquisa englobam grandes empresas e/ou grandes projetos distribuídos de software. Separar o contexto do projeto de software sem compreender a razão pela qual a empresa desenvolve de forma

distribuída pode fazer com que a veracidade do modelo final seja reduzida. Por isso, o estudo de caso foi o método de pesquisa que mais se alinhou com o problema deste trabalho.

Um protocolo de pesquisa foi elaborado, baseando-se nas teorias citadas na seção 2 deste trabalho. Este protocolo de pesquisa foi aplicado em vários casos, sendo que 3 destes são empresas com certificação CMMI nível 5; 2 destes casos são empresas com certificação CMMI nível 3 e último caso foi o desenvolvimento de um aplicativo *open source*.

A execução desta pesquisa consistiu em uma visita sistemática aos casos. Neste sentido, o protocolo foi investigado na organização, sem influência do pesquisador. Tal protocolo é composto em três partes. A primeira parte do protocolo investiga a organização da rede, por isso, explora os projetos distribuídos e a relação entre os sites. A segunda parte do protocolo explora o gerenciamento da produção que consiste em investigar e identificar os processos distribuídos e o distribuidor de tarefas. A última parte do protocolo investiga a produção, explorando a fabricação de artefatos e a interação/comunicação entre os sites.

Cada parte do protocolo foi composta por dois blocos. Cada bloco tem seu constructo teórico e um conjunto de item que foram investigados na organização. A aplicação do protocolo gerou conclusões, que conseqüentemente alimentaram a construção do modelo M3DS.

4. M3DS

A dinâmica de um ambiente de DDS revela o funcionamento da rede, de sua concepção inicial até seu encerramento. Porém, abordar individualmente os sites na rede pode não revelar a influência do projeto tem sobre sua existência na rede. Portanto, a dinâmica da rede, representada na figura 1, aborda conjuntamente dois componentes fundamentais em um ambiente de DDS.

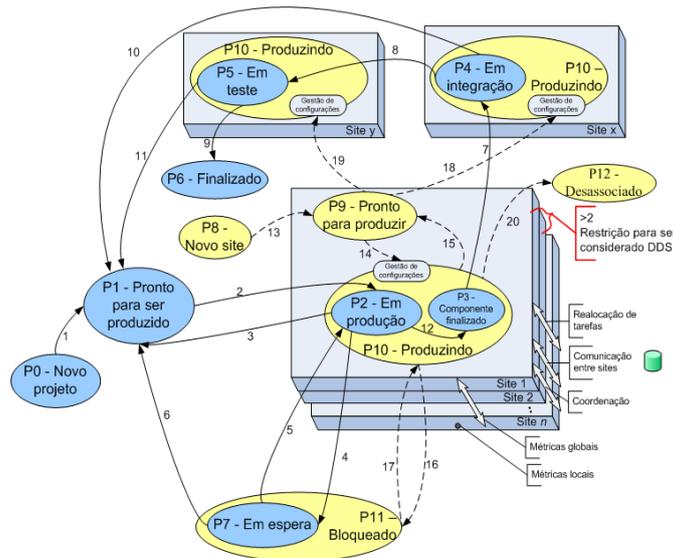


Figura 1 - M3DS: modelo de dinâmica de DDS

O primeiro indica a dinâmica de uma unidade de produção de software. O segundo é atrelado ao projeto desenvolvido em uma rede de produção de software. Sobre essa dinâmica, há uma intersecção, que ocorre quando o site precisa desenvolver um subproduto de rede, e uma instância do projeto para vários sites. A seguir, serão explicados brevemente todos os estados do modelo e identificados quais estados existentes e quais condições necessárias para a mudança de estado. Toda seta (transição) da figura 1 é disparada quando as atividades atribuídas ao estado que a antecede sejam concluídas.

P0: Novo Projeto

Este estado indica a concepção de um novo produto de software por alguma organização, seja ela em rede ou não. Nesta fase, além da concepção do projeto há a elaboração de uma visão compartilhada para que todos os nós envolvidos na produção possam compreender a razão do produto final.

P1: Pronto para ser produzido

Para se alcançar o estado P1 é necessário que seja disparada a seta 1. Este disparo indica que o plano gerencial do projeto em termos de componentização e divisão de tarefas foram realizados. Após esta seta, todos, ou parte dos artefatos estarão prontos para serem disponibilizados e distribuídos para que a rede e os *sites* possam desenvolvê-los. Este estado indica que um papel de distribuidor de tarefas entra em ação para gerenciar a distribuição das tarefas do projeto na rede. Representa a segmentação e alocação das tarefas nos diversos sites de produção. É neste estado que se inicia o processo de ruptura, ou seja, o projeto é segmentado e alocam-se a produção dos segmentos para sites distintos.

P2: Em produção

Este estado representa uma tarefa que tem como resultado um subproduto de trabalho. Para almejar o estado P2, a seta 2 precisa ser disparada. Esta seta indica que o subproduto foi alocado em produção. Esta etapa implica em enviar para o site a tarefa e os dados sobre ela. Podem ser enviados, por exemplo, códigos mais atualizados, scripts de banco de dados, requisitos do software, etc.

P3: Componente finalizado

O componente é finalizado quando sua construção é concluída. Neste caso, indica o fim do processo de codificação de um componente ou módulo de software. Este estado sinaliza que o componente está apto a ser integrado com outro. As operações de *check-out* são efetuadas após a conclusão do componente. A gestão de configurações está presente explicitamente nesta etapa.

P4: Em integração

Após a codificação de dois ou mais componentes, o processo de integração é iniciado. Neste caso apenas um site é responsável em integrar dois ou mais componente, gerando outro produto de trabalho. A seta 7 indica o início deste estado.

P5: Em teste

Quando a seta 8 é disparada inicia-se o procedimento de teste. Neste caso o teste em um ambiente de DDS é de integração do software. Após esta etapa, o projeto

pode ser finalizado, mudando seu estado para P6, ou o teste pode gerar outro produto de trabalho, realimentando o ciclo por meio do estado P1.

P6: Finalizado

O projeto é concluído. Este estado representa que o projeto foi totalmente montado e concluído.

P7: Em espera

As dependências funcionais entre artefatos gerados pelos sites podem ocasionar em bloqueio em uma determinada tarefa. Este estado representa o bloqueio da tarefa em relação a um projeto. As setas 4 e 16 ocorrerem paralelamente quando há uma dependência de produtos de trabalho. Se isso ocorrer, para o projeto, o subproduto fica bloqueado. Este estado ocorre juntamente com o estado P11.

P8: Novo site

Indica que um novo nó da rede está pronto para associar-se a uma rede de produção de software. Para ser considerado um novo site, um mecanismo de coordenação, agrupamento e processo de software devem ser estipulados. Este estado indica que o site está apto a trabalhar em equipe com os demais sites. Um site pode ser criado a partir de um modelo já existente, importando processos e modelos de desenvolvimento. Os procedimentos de replicação são detalhados por Fabri (2004), Fabri (2007) e Trindade (2004).

P9: Pronto para produzir

Para que o estado P9 seja alcançado é necessário que a transação indicada pela seta 13 seja disparada. A seta 13 representa os procedimentos iniciais de produção.

P10: Produzindo

Este estado representa que o site está executando uma tarefa e retornará como resultado um subproduto. Pode ser alcançado após o disparo da seta 14. Neste caso, a produção inicia-se, ou seja, as atividades referentes ao desenvolvimento do produto que foram atribuídas para o site são executadas. Esta etapa ocorre paralelamente a seta 2.

P11: Bloqueado

Podem ocorrer bloqueios na produção por dependências funcionais. Por exemplo, um site precisa de um produto de trabalho de outro site para prosseguir. Este estado representa que o site está bloqueado para uma determinada tarefa. Para tanto, as transições 4 e 16 ocorrem quando há uma dependência de produtos de trabalho. Se isso ocorrer, para o projeto, o subproduto fica bloqueado, pois, não pode ser desenvolvido pelo fato de sofrer dependência funcional.

P12: Desassociado

A ação indicada pela seta 20 indica a desassociação do *site* à rede que é removido da mesma e deixa de fazer parte do ambiente de DDS. Esta transição ocorre quando o site deixa de atuar neste projeto/grupo. Há uma desassociação do site com relação aos demais sites e projeto.

5. Conclusões

As contribuições deste trabalho centra-se na construção um modelo de dinâmica de desenvolvimento distribuído de software. Para tanto, diversas teorias, como organização do trabalho (MINTZBERG, 2003), foram analisadas com o intuito de agregar valor ao modelo M3DS. Dessa maneira, o M3DS incorpora conceitos de organização do trabalho em produção de software. Assim, foi possível estabelecer uma posição mais precisa da rede de produção, mapeando-a em estados que podem ser analisados qualitativa e/ou quantitativamente. O modelo M3DS é um modelo geral de dinâmica com aplicação generalizada em outros casos, mas com restrições impostas pela metodologia de pesquisa empregada. Seu objetivo é demonstrar o funcionamento do desenvolvimento distribuído.

Nos casos aplicados em empresas, uma configuração rigorosa e específica era aplicada sobre os projetos e os sites. Nesses casos, os projetos foram preparados para serem distribuídos. A elaboração da solução escolhida pelo arquiteto, junto com analistas de negócio, influencia altamente a distribuição de tarefas. Uma configuração de ambiente é determinada junto com o projeto e a distribuição de tarefas. Foi detectado que os sites eram homogêneos porque todos seguiam os mesmos processos. No caso do software livre, como não havia divisão sistemática do projeto e homogeneização dos sites, houve muitos impasses como dependência funcional e repúdio de tarefa.

A interação instantânea por meio de conferência, chat e VOIP entre os desenvolvedores, durante todo o projeto mostrou-se necessário em todos os casos. Embora os casos com processos mais sistemáticos, isto é, empresas do setor, tenham um processo de divisão de tarefas – processo de ruptura – formalizado, todos os desenvolvedores julgaram necessária a interação entre os elementos dos times. Além disso, a coordenação, seja esta técnica ou gerencial, também se mostrou essencial por todo o ciclo de vida do produto.

Os casos apresentaram diferentes aspectos culturais que influenciaram na produção de software. Estes aspectos, que compreenderam a pronúncia de um mesmo idioma, o calendário referente a feriados, a religião e outros que compõem a cultura organizacional – explicado por Hofstede (2005) – diretamente impactaram na interação entre todos os indivíduos envolvidos em um projeto e foram acentuados pela distância física.

No desenvolvimento distribuído de software alguns problemas que localmente são sutis tornam-se mais acentuados em função da distância física. Entretanto, estes problemas não representam um empecilho para que as organizações desenvolvam software de maneira distribuída. Seja qual for o motivo e o modelo de negócios que a tornam viável – por aquisição, instalação de filial, parcerias, etc. – a distribuição da produção de software agrega muitos benefícios para a organização e para a região na qual o site é instalado.

Referencias

- ALSTYNE, Marshall Van. **The state of network organization: a survey in three frameworks.** in *Journal of Organizational Computing & Electronic Commerce*, 7(3); pp.83-151, 1997.

- AUDY, Jorge Luis Nicolas; PRIKLADNICKI, Rafael . **Desenvolvimento Distribuído de Software: Desenvolvimento de Software com Equipes Distribuídas**. 1. ed. Rio de Janeiro: Campus/Elsevier, 2007. v. 1. 211 p.
- BATTIN, Robert D. et al. **Leveraging resources in Global Software Development**. IEEE transactions on Software Engineering, v. 18, n. 2. Março/Abril 2001. p. 70-77.
- BORDEN, Alexander, NETT, Bernhard e WULF Volker. Coordination Practices in Distributed Software Development of Small Enterprises, icgse,pp.235-246, INTERNATIONAL CONFERENCE ON GLOBAL SOFTWARE ENGINEERING (ICGSE 2007), 2007
- BURTON R.M.; OBEL B. **Strategic Organizational Diagnosis and Design**. Kluwer Academic Publishers, 1998.
- CAMPBELL, Catherine Lowry; VAN DE WALLE, Bartel. Asynchronous Requirements Engineering: Enhancing Distributed Software Development. In: INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY: RESEARCH AND EDUCATION, 2003, Newark. **ITRE 2003**. Newark: ACM, 2003. p. 133 - 136.
- CARMEL, Erran; AGARWAL, Ritu. Tactical Approaches for Alleviating Distance in Global Software Development. **IEEE Software**, v. 18, n. 2. março/abril 2001. p. 22-29.
- CATALDO, et al. On Coordination Mechanisms in Global Software Development. IN PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON GLOBAL SOFTWARE ENGINEERING, **ICGSE**. IEEE Computer Society, 2007
- EDWARDS, H. Keith; SRIDHAR, Varadharajam. Analysis of the Effectiveness of Gloval Virtual Teams in Software Engineering Projects. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 36., 2003, Hawaii. **Proceedings Hawaii: IEEE**, 2003.
- FABRI, J. A. et al. Tutorial: **Desenvolvimento e Replicação de uma Fábrica de Software**. In: IV Simpósio Internacional de Melhoria de Processo de Software, 2004, São Paulo. Anais do IV Simpósio Internacional de Melhoria de Processo de Software - Tutorial, 2004. v. 001. p. 3-1-3-50.
- FABRI, J. A.; TRINDADE, A. L. ; A. L'ERARIO, A. ; PESSOA, M. S. P. . **A Organização de uma Máquina de Processo e a Melhoria do Processo de Produção de Software em um Ambiente de Fábrica**. In: VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, 2007, Lima - Peru. Anais das VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, 2007. v. 1. p. 229-236.
- HERBSLEB, James D.; GRINTER, Rebecca E.. Architectures, Coordination, and Distance: Conway. **IEEE Software**, v. 16, n. 1, p.63-70, 1999.

- HOFSTEDE, Geert; HOFSTEDE, Gert-Jan. **Cultures and Organizations: Software of the Mind**. New York: McGraw-Hill U.S.A., 2004.
- L'ERARIO, Alexandre; et al. Desenvolvimento distribuído de software para sistemas pervasivos: um estudo de caso. In: SIMPÓSIO BRASILEIRO DE SISTEMA DE INFORMAÇÃO, 1., 2004, Porto Alegre. Simpósio. Porto Alegre: SBSI, 2004. p. 163 - 170.
- MINTZBERG, Henry. **Criando organizações eficazes: Estruturas em cinco configurações**. 2. ed. São Paulo: Atlas, 2003. 334 p.
- PRIKLADNICKI, Rafael; AUDY, Jorge Luis Nicolas. Towards a Model of Software Development Process for a Physically Distributed Environment - Minimizing communication difficulties and adding planning and evaluation view to the software development life cycle. In: VIII CONGRESSO ARGENTINO DE CIENCIAS DE LA COMPUTACION, 2002, Buenos Aires: CACIC 2002 - Anais del VIII Congreso Argentino de Ciencias de la Computacion. 2002. v. I, p. 798-809.
- ROBINSON, M., Kalakota, **Offshore Outsourcing: Business Models, ROI and Best Practices**. EUA: Mivar Press.2004
- SINHA, V.S.; SENGUPTA, B.; GHOSAL, S., An Adaptive Tool Integration Framework to Enable Coordination in Distributed Software Development, GLOBAL SOFTWARE ENGINEERING, 2007. *ICGSE 2007. Second IEEE International Conference on* , vol., no., pp.151-155, 27-30 Aug. 2007
- TRINDADE, André L P ; et al . Desenvolvimento e Replicação de uma Fábrica de Software. In: IV JORNADAS IBEROAMERICANAS DE INGENIERÍA DEL SOFTWARE E INGENIERÍA DEL CONOCIMIENTO, Madri. **Anais del IV Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento**, 2004
- WOMACK, J. P., JONES, D. T., ROSS, D. **A máquina que mudou o mundo**. São Paulo, 1991.
- YIN, Robert K.. **Estudo de caso: planejamento e métodos**. 3. ed. Porto Alegre: Bookman, 2005. 212 p.