

Petri-SW: Uma Metodologia baseada em Redes de Petri para Avaliação do Desempenho de Processos de Software

Lívia Maria R. de Vasconcelos¹, César Augusto Lins de Oliveira¹, Ricardo Massa F. Lima¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

{lmrv, calo, rmfl}@cin.ufpe.br

***Abstract.** This paper presents a methodology based on Petri nets with main focus on performance evaluation of software processes. The methodology aims to support strategic decision making and facilitate the alignment of software development processes to business goals of Information Technology organizations and it is based on software process improvement approaches, such as IDEAL Model, Cycle of Improvement ISO/IEC 15504 and DMAIC.*

***Resumo.** Este artigo apresenta uma metodologia baseada em Redes de Petri com foco principal na avaliação do desempenho de processos de software. A metodologia visa apoiar a tomada de decisões estratégicas e viabilizar o alinhamento dos processos de desenvolvimento de software aos objetivos de negócio de organizações de Tecnologia da Informação, sendo baseada em práticas de abordagens de melhoria de processos existentes, tais como Modelo IDEAL, Ciclo de Melhoria da norma ISO/IEC 15504 e DMAIC.*

1. Introdução

Diante dos avanços tecnológicos e do aumento da complexidade dos problemas que precisam ser resolvidos, o desenvolvimento de software tem exigido ferramentas e métodos cada vez mais ágeis e adaptáveis às distintas realidades de projetos. Por sua vez, a inexistência de dados (base histórica) impossibilita que planejamentos realistas e bem sucedidos de desenvolvimento de software sejam feitos, o que implica em prazos estourados, custos extrapolados e a qualidade dos produtos finais acaba por ser deixada em segundo plano. A falta de padronização das atividades realizadas por projetos, especialmente por projetos de naturezas distintas, dificulta predições de estimativas.

Visando diminuir o problema da imprevisibilidade de resultados, empresas têm investido cada vez mais na implantação de processos de desenvolvimento de software que orientem projetos na realização de boas práticas que agreguem qualidade aos produtos intermediários gerados durante o desenvolvimento e, conseqüentemente, agreguem qualidade aos produtos finais gerados. Neste sentido diversos modelos de qualidade têm sido usados como referência, tais como o CMMI (CMMI, 2011) e o MPS.BR (SOFTEX, 2011).

O uso de processos tem se tornado uma prática bem estabelecida em empresas de desenvolvimento de software. No entanto, observa-se que, no contexto de empresas de desenvolvimento de software, a análise dos resultados reais obtidos (como produtividade, satisfação do cliente) com a implantação dos processos é bastante subjetiva, resultantes de opiniões e expectativas pessoais.

Diversos problemas associados com a implantação de processos são conhecidos. Entre os problemas mais comuns, pode-se citar a resistência a mudanças, a cultura organizacional, a falta de patrocínio gerencial, entre outros. Para tratar destes problemas, empresas optam geralmente por realizar treinamentos, *workshops*, aplicar o processo em projetos pilotos e obter *feedback* da equipe para evolução contínua do processo (von Wangenheim *et al*, 2009).

Sendo assim, para que as avaliações de implantações de processos sejam objetivas, faz-se necessária a apresentação de medidas quantitativas que comprovem os benefícios reais alcançados com a implantação de processos. Especialmente em empresas onde há resistência à adoção de processos, ter informações objetivas é imprescindível para justificar o uso do processo pelos projetos e apoiar a tomada de decisões estratégicas nas organizações. Incluir métricas de software em um sistema de melhoria contínua permite validar quantitativamente as mudanças realizadas no processo. Desta forma, é possível avaliar se uma mudança implica ou não em uma melhoria real na qualidade do processo.

Um grande número de pesquisas propõe o uso de Redes de Petri como método de análise qualitativa e quantitativa de *workflow*, com o fim de oferecer apoio a projetos de melhoria. Este formalismo é amplamente usado para verificação e análise de desempenho de sistemas variados e apresenta características que fazem a sua aplicação em *workflow* bastante natural (van der Aalst *et al*, 1994), (Dehnert *et al*, 2000), (Li *et al*, 2008). Através de redes de Petri é possível modelar um processo específico, de forma

que seja possível realizar análises do desempenho e confiabilidade deste processo. De maneira mais detalhada, redes de Petri podem ser utilizadas para representar modelos de simulação utilizados para analisar processos e avaliar o desempenho de diferentes alternativas de projeto (Rozinati *et al*, 2009).

Este artigo pretende apresentar uma metodologia baseada em redes de Petri para prover avaliação do desempenho de processos de desenvolvimento de software. A técnica de redes de Petri foi escolhida por possuir uma fundamentação matemática e prover representação gráfica compreensível, além da possibilidade de simulações e verificações automáticas. O artigo está organizado da seguinte forma: a Seção 2 aborda conceitualmente as Redes de Petri; a Seção 3 apresenta a Proposta Metodológica deste trabalho, a Seção 4 apresenta a metodologia Petri-SW; a seção 5 apresenta um estudo de caso que exemplifica a aplicação da metodologia; e, finalmente, a seção 6 apresenta as conclusões.

2. Redes de Petri

Segundo Hermanns *et al* (2011), a análise de desempenho é o tipo mais empregado de avaliação quantitativa de sistemas e inclui a descrição, análise e a otimização do comportamento dinâmico e dependente de tempo dos sistemas. Alguns dos problemas que podem ser tratados com avaliação de desempenho são: comparação de sistemas, identificação de gargalos, caracterização de cargas de trabalho, configuração de sistemas, análise do comportamento de processos e previsão de desempenho.

O trabalho desenvolvido e apresentado neste artigo utiliza dos benefícios da avaliação de desempenho para buscar analisar o comportamento de processos de desenvolvimento de software e realizar previsões do desempenho. Neste sentido, as redes de Petri vão ao encontro destes objetivos, visto que podem ajudar a avaliar o desempenho de processos e estimar o benefício de um redesenho deles ou prever resultados através da variação dos recursos responsáveis por uma determinada tarefa.

Em linhas gerais, redes de Petri é uma ferramenta gráfica para a descrição formal de sistemas cuja dinâmica é caracterizada por concorrência, sincronização, exclusão mútua e conflitos, que são características típicas de ambientes distribuídos. Redes de Petri foram criadas em 1962 por Carl Adam Petri (Petri *et al*, 1962) e incorporam uma noção de estados e regras para mudança de estados que permitem capturar as características estáticas e dinâmicas de um sistema real.

Uma rede de Petri é formada por lugares, transições e arcos direcionados. Arcos interligam lugares e transições, não podendo conectar lugares a lugares ou transições a transições. Os lugares de entrada de uma transição são aqueles aos quais um arco que sai da transição se destina. Os lugares de saída são aqueles dos quais um arco sai em direção a uma transição e podem conter um ou mais tokens, que são representados por círculos pretos dentro do lugar. O número de tokens pode mudar durante a execução da rede (Marsan *et al*, 1995).

Os lugares correspondem às variáveis de estado e as transições às ações (eventos) realizadas pelo sistema. Para que uma ação seja realizada, algumas pré-condições precisam ser atendidas (condição de algumas variáveis de estado). Isto quer

dizer que os lugares e transições se relacionam, o que possibilita a realização de uma ação (Maciel *et al*, 1996).

Um lugar p é chamado de lugar de entrada de uma transição t se existe um arco dirigido de p para t . O lugar p é chamado de lugar de saída de t se existe um arco direcionado a partir de t para p . O estado de uma rede de Petri é definido por sua marcação. Uma marcação indica o número de *tokens* presente em cada lugar da rede (Marsan *et al*, 1995). A Figura 1 apresenta um exemplo simples de uma rede de Petri.

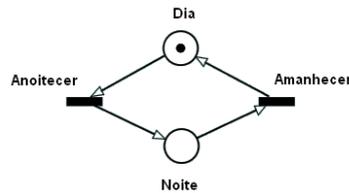


Figura 1: Exemplo de uma rede de Petri (Oliveira, 2008).

O exemplo da Figura 1 modela o ciclo dia e noite. Quando um *token* é colocado no lugar Dia, significa que o sistema está no estado Dia. Isto habilita a transição “Anotecer” a ocorrer. Quando esta transição dispara, o *token* é removido do lugar Dia e um *token* é colocado no lugar Noite, indicando que o sistema está neste estado. Isto, por sua vez, desabilita a transição “Anotecer” e habilita a transição “Amanhecer”. De forma análoga, o disparo da transição “Amanhecer” faz com que o *token* seja removido do lugar Noite e um *token* seja colocado no lugar Dia.

O comportamento dos sistemas pode ser descrito em função dos seus estados e suas alterações. A marcação da rede de Petri é modificada a cada ação realizada (transição disparada), respeitando regras de execução. Desta forma é possível simular o comportamento dinâmico dos sistemas (Maciel *et al*, 1996). Existem diversas extensões às redes de Petri, tais como redes de Petri Coloridas (Maciel *et al*, 1996), redes de Petri com Arco Inibidor (Maciel *et al*, 1996) e Redes de Petri Estocásticas Generalizadas (Oliveira *et al*, 2011). Para desenvolvimento deste trabalho foram utilizadas as Redes de Petri Estocásticas Generalizadas, definidas da seguinte maneira.

Uma Rede de Petri Estocástica Generalizada (GSPN) é uma 7-tupla definida como $GSPN = (P, T, \Pi, I, O, H, M_0, W)$, onde:

- P é o conjunto de lugares;
- T é o conjunto de transições temporizadas e imediatas, $P \cap T = \emptyset$;
- $\Pi : T \rightarrow \mathbb{N}$ é uma função prioridade, onde:

$$\Pi(t) = \begin{cases} \geq 1, & \text{se } t \text{ é uma transição imediata;} \\ 0, & \text{caso contrário.} \end{cases}$$

- $I, O, H: (T \times P) \rightarrow \mathbb{N}$ são as funções de entrada, de saída e inibidora, respectivamente;
- $M_0: P \rightarrow \mathbb{N}$ é a função de marcação inicial;

- $W: (T \times (P \rightarrow \mathbb{N})) \rightarrow \mathbb{R}_+^*$ é a função peso, a qual representa o peso (w_i) das transições imediatas e a taxa (λ_i) de transições temporizadas. Estes valores podem ser dependentes da marcação M_i de uma rede.

As técnicas de análise de *workflow* se caracterizam em qualitativas e quantitativas. As técnicas qualitativas visam à correção lógica de um determinado processo, tratando de problemas como, por exemplo, tarefas mortas. As técnicas de análise quantitativas têm foco no desempenho de um dado processo, buscando indicadores de desempenho, como média de tempo de execução e capacidade de utilização. Existem dois tipos de análises quantitativas:

- **Transiente** – Avalia o comportamento do sistema durante um período de tempo a partir da saída do estado inicial.
- **Estacionária** – Avalia o comportamento do sistema no estado de equilíbrio. Será utilizada no estudo de caso deste artigo.

3. Proposta Metodológica

Para alcance dos principais objetivos deste trabalho, foram utilizados os passos da proposta metodológica apresentada na Figura 2.

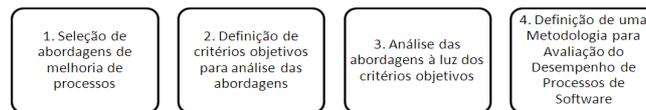


Figura 2: Proposta metodológica.

O ponto de partida para a proposição da metodologia Petri-SW foi a análise de abordagens e metodologias existentes para melhoria de processos de desenvolvimento de software e referentes a negócios genéricos. Para seleção das abordagens de melhoria de processos, uma das bases utilizadas foi o artigo (Sharma et al, 2008), cujo objetivo é apresentar uma análise comparativa entre diferentes modelos de melhoria de processos de software existentes na literatura. Assim, as abordagens selecionadas foram: modelo IDEAL, o Ciclo de Melhoria da ISO/IEC 15504, o modelo DMAIC e a Metodologia de Modelos Estocásticos de Oliveira (2008).

O segundo passo foi a definição de critérios objetivos para realização de uma análise crítica das abordagens de melhoria de processos selecionadas. Os principais critérios foram extraídos dos artigos (Dyba, 2005), (Khokhar, 2010), (Rocha *et al*, 2005), (Bertollo *et al*, 2006) e (Wan *et al*, 2011).

A Tabela 1 resume os resultados alcançados com a comparação realizada entre as abordagens (passo 3 da proposta metodológica). Os critérios e resultados dessa análise foram imprescindíveis para construção da metodologia Petri-SW, que busca contemplar em suas características a maioria desses critérios, devido à importância deles em uma metodologia de melhoria de processos.

Tabela 1: Comparativo entre IDEAL, ISO/IEC 15504, DMAIC e Metodologia de Oliveira (2008).

Característica	IDEAL	ISO 15504	DMAIC	Metodologia (Oliveira, 2008)
----------------	-------	-----------	-------	------------------------------

C1. Possui passos bem descritos em termos de entrada e saída.	Sim	Sim	Sim	Sim
C2. Baseada em valores de alinhamento estratégico.	Sim	Sim	Sim	Sim
C3. Baseada em valores de envolvimento da gerência.	Sim	Sim	Sim	Não
C4. Orienta à realização de um Gap Analysis para conhecer as práticas atuais da Organização.	Sim	Sim	Sim	Não
C5. Orienta a definição de estratégias de implementação prática das melhorias, com aplicação de pilotos, por exemplo.	Sim	Sim	Sim	Não
C6. Reconhecida internacionalmente ou utiliza explicitamente práticas de modelos reconhecidos internacionalmente.	Sim	Sim	Sim	Sim
C7. Orientada a avaliações para “certificação” dos processos.	Sim	Sim	Não	Não
C8. Explicita boas práticas para implantação do processo em projetos, tais como a possibilidade de “ <i>tailoring</i> ” ou adaptações do processo para diferentes naturezas de projeto.	Não	Não	Não	Não
C9. Explicita a necessidade de disseminação do conhecimento das mudanças efetuadas, seja através de treinamento, mentoring e/ou acompanhamento periódico.	Sim	Não	Não	Não
C10. Orienta a utilização de ferramentas que automatizem atividades do processo e/ou forneçam apoio à melhoria de processos.	Não	Não	Sim	Sim
C11. Apresenta formas de medir quantitativamente o desempenho dos processos.	Não	Não	Sim	Sim
C12. Possibilita avaliar a correteza do processo antes de sua utilização (análise qualitativa).	Não	Não	Não	Sim
C13. Sugere técnicas para cálculo de tempo, recursos e carga de trabalho.	Não	Não	Não	Sim
C14. Orienta à aprendizagem, através da exploração do conhecimento.	Sim	Sim	Sim	Sim
C15. Prevê a reafirmação do comprometimento da gerência e/ou a apresentação da evolução do desempenho dos processos para a Gerência Sênior.	Sim	Sim	Sim	Não

Como observado, todas as abordagens atendem os seguintes critérios: base em valores de alinhamento estratégico (C3); utilização de práticas de modelos reconhecidos internacionalmente (C6) e orientação à aprendizagem, através da exploração do conhecimento (C14).

A metodologia de Oliveira (2008) possui destaque em relação aos critérios C11, C12 e C13, possibilitando a realização de avaliações de correteza do processo antes de sua utilização (análise qualitativa) e de avaliação de desempenho de processos (análise quantitativa). Tais características foram mantidas na Metodologia Petri-SW.

O último passo da proposta metodológica representada na Figura 2 se refere à definição de uma Metodologia para avaliação do desempenho de processos de software. A execução deste passo resultou na Metodologia Petri-SW, descrita na Seção 4 a seguir.

4. Metodologia Petri-SW

A Metodologia Petri-SW é composta por cinco fases: Iniciação, Definição, Implantação, Monitoração e Aprendizagem (ver Figura 3). Cada fase será descrita em termos de atividades e passos. Alguns dos papéis envolvidos na execução das atividades são: Gerente Sênior, Gerente do Programa de Melhoria, Grupo de Processos, Gerente do Grupo de Processos, Gestor de Projeto, Analista de Medição, Analista de Qualidade e Equipe de Projeto.



Figura 3: Fases da Metodologia.

Fase 1: Iniciação

A fase Iniciação, ilustrada na Figura 4, é composta por quatro atividades e tem como objetivo formalizar o início do programa de melhoria dentro da organização. As atividades da fase Iniciação são descritas a seguir.

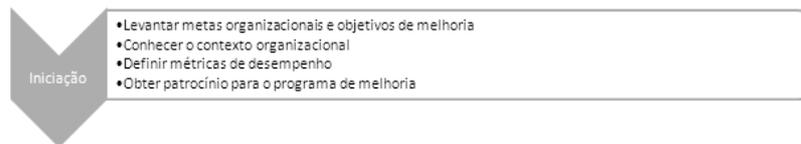


Figura 4: Atividades da Fase de Iniciação.

Levantar metas organizacionais e objetivos de melhoria - Esta atividade visa conhecer os objetivos de negócio da organização que busca realizar melhorias em seus processos. Os passos para executar esta atividade são: coletar metas organizacionais, derivar objetivos de melhoria, definir equipe do programa de melhoria e elaborar versão inicial do Plano de Melhoria.

Conhecer o contexto organizacional - O objetivo desta atividade é conhecer os processos de trabalho e relacionamentos entre as áreas, conhecer as boas práticas adotadas pelas equipes de desenvolvimento e também conhecer os pontos fracos que afetam o programa de melhoria. Os passos para executar esta atividade são: realizar entrevistas, selecionar processo a ser melhorado e elaborar Plano de Ação para o processo selecionado.

Definir métricas de desempenho - As métricas de desempenho são fatores que a gerência sênior, que está patrocinando o programa de melhoria, tem interesse em acompanhar como forma de garantir que o programa está trazendo os resultados esperados como objetivos de negócio. Os passos para executar esta atividade são: levantar métricas de desempenho, validar métricas de desempenho e atualizar Plano de Melhoria.

Obter patrocínio para o programa de melhoria - O propósito desta atividade é registrar o comprometimento do Gerente Sênior com o Programa de Melhoria e reforçar a importância da participação dos colaboradores para o sucesso do programa na organização. Os passos para executar esta atividade são: assinar Plano de Melhoria e realizar apresentação de abertura do programa de melhoria para toda a organização.

Fase 2: Modelagem

Nesta fase é construído o modelo em GSPN para ser utilizado durante as análises de corretude e de desempenho que fornecerão insumos para serem trabalhados no ciclo de melhoria. As atividades da fase Modelagem podem ser vistas na Figura 5 e serão descritas a seguir.



Figura 5: Atividades da fase de Modelagem.

Definir processo - Iniciar a execução do Plano de Ação, definindo um novo processo ou alterando um processo existente. Os passos para executar esta atividade são os seguintes: executar Plano de Ação e documentar processo.

Construir modelo em GSPN - Este processo deverá ser modelado em uma Rede de Petri Estocástica Generalizada (Oliveira *et al*, 2011), para que análises sejam feitas posteriormente. Existem ferramentas que realizam a transformação de um processo em um modelo GSPN automaticamente (como o BPEL2Net). Os passos para executar esta atividade são: transformar processo em GSPN, selecionar ferramenta de modelagem e modelar processo em GSPN.

Validar Qualitativamente o Modelo GSPN - Garantir que o modelo GSPN está realizando a execução do processo de forma semelhante à realidade.

Verificar corretude do processo - Aplicar técnicas qualitativas para verificar propriedades lógicas do modelo (*workflow*), tais como ausência de tarefas desnecessárias, execução completa do processo e sem tokens remanescentes. Atendendo esta expectativa mínima, o processo é chamado *sound*. Os passos para executar esta atividade são: verificar se o processo atende a propriedade *soundness* e adequar modelo.

Fase 3: Implantação

A fase Implantação é composta por três atividades, ilustradas na Figura 6, e visa definir estratégias de implantação do processo para executá-lo em um projeto. Outro objetivo dessa fase é garantir uma infraestrutura de monitoração adequada para coleta dos dados

de execução do processo, que serão monitorados e analisados na fase seguinte. As atividades da fase Implantação são detalhadas a seguir.

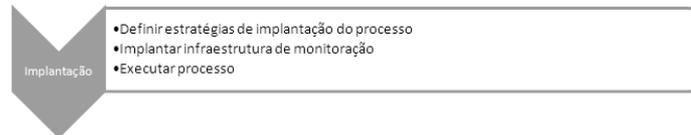


Figura 6: Atividades da Fase de Implantação.

Definir estratégias de implantação do processo - O resultado da execução desta atividade é um Plano de Implantação, que irá conter informações de papéis, recursos, estratégias de implantação (planejamento de treinamentos e mentorizações) e especialmente o cronograma de implantação do processo. Os passos para executar esta atividade são: definir equipe envolvida na execução do processo, definir treinamentos e mentorizações, definir cronograma e elaborar versão inicial do Plano de Implantação.

Implantar infraestrutura de monitoração - Esta atividade é imprescindível para garantir que os dados necessários para as análises de desempenho (fase seguinte) estejam corretos e disponíveis. A infraestrutura deverá fornecer ferramentas adequadas para coleta dos dados relativos ao tempo gasto com a execução de cada tarefa do processo. Os passos para executar esta atividade são: definir infraestrutura de monitoração, finalizar elaboração do Plano de Implantação e implantar infraestrutura de monitoração.

Executar processo - Esta atividade visa a execução do processo trabalhado no ciclo de melhoria em um projeto, como forma de levantamento de dados de desempenho e informações gerenciais que comprovem que o processo está atendendo as expectativas do Gerente Sênior e agregando valor às metas de negócio da organização. Os passos para executar esta atividade são: realizar reunião de apresentação do processo ao projeto, realizar treinamentos, executar processo, aplicar mentorização aos participantes do piloto e realizar avaliações de aderência ao processo.

Fase 4: Monitoração

A fase Monitoração é composta por três atividades, ilustradas na Figura 7, e tem como objetivo principal obter subsídios para a análise posterior do comportamento do processo em relação ao desempenho. As atividades da fase Monitoração são descritas a seguir.

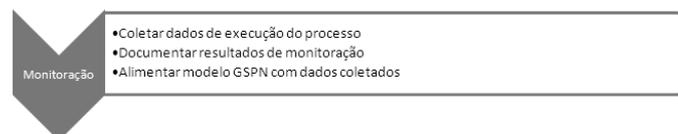


Figura 7: Atividades da fase de Monitoração.

Coletar dados de execução do processo - Realizar coletas da execução do processo, buscando levantar informações tais como tempo gasto com execução de tarefas. Esta atividade é realizada em paralelo à execução do piloto (última atividade da fase de Implantação).

Documentar resultados de monitoração - Consolidar os dados de execução do processo inseridos na ferramenta de forma a subsidiar com dados a posterior construção do Modelo em GSPN. Nesta atividade são consolidados os dados de execução das atividades e documentadas as probabilidades de decisão relativas a atividades que podem ser executadas de mais de uma forma (*gateway*).

Alimentar Modelo GSPN com dados coletados - Antes de iniciar a validação quantitativa do modelo GSPN, para posterior avaliação do desempenho do workflow, é necessário que o Modelo GSPN seja alimentado com informações históricas acerca da execução prática do processo. O objetivo dessa atividade é alimentar o Modelo com esses dados. A execução desta atividade prevê a realização do passo inserir dados de execução para o Modelo GSPN.

Fase 5: Aprendizagem

A fase de Aprendizagem é composta por quatro atividades, ilustradas na Figura 8, e tem como objetivo principal analisar quantitativamente o Modelo GSPN definido na fase anterior, com o objetivo de obter estimativas para o comportamento futuro do processo, através de simulações.

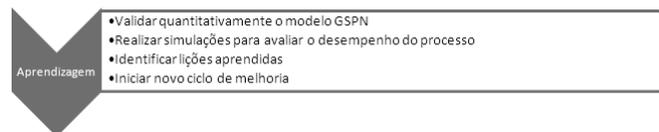


Figura 8: Atividades da fase de Aprendizagem.

Validar quantitativamente o modelo GSPN - O objetivo dessa atividade é realizar diversos experimentos através da variação de cenários (variando a quantidade de recursos, por exemplo), para confirmar que o modelo GSPN está se apresentando resultados próximos à realidade. Esta atividade prevê a execução de um passo: realizar experimentos variando os cenários do Modelo GSPN.

Realizar simulações para avaliar o desempenho do processo - Conhecer o comportamento do processo através da análise dos dados de sua execução. Recomenda-se que a avaliação de desempenho seja feita utilizando uma ferramenta como o Time Net (Zimmerman *et al*, 2007), que terá como entradas as métricas selecionadas; o tipo de análise (transiente ou estacionária) e os parâmetros para simulação. Os passos para executar esta atividade são: monitorar Plano de Melhoria, simular comportamento do processo, adequar modelo e realizar novas simulações do comportamento do processo com o modelo adequado.

Identificar lições aprendidas - Levantar as lições aprendidas do ciclo de melhoria através de uma reunião envolvendo todos os participantes do ciclo de melhoria finalizado. Os passos para executar esta atividade são: realizar reunião de lições aprendidas e elaborar relatório de lições aprendidas.

Iniciar novo ciclo de melhoria - O objetivo desta atividade é consolidar todas as melhorias e iniciar um novo ciclo de melhoria, através de uma nova execução desta metodologia retomando para a fase de Implantação. Um novo Plano de Ação é elaborado contendo todas as melhorias identificadas na simulação do modelo GSPN, que deverão ser implementadas no processo quando o novo ciclo for iniciado.

5. Estudo de Caso

Esta seção apresenta um estudo de caso que exemplifica a adoção da Metodologia Petri-SW. O estudo de caso descrito neste capítulo considera um processo real adotado em uma célula de manutenção de uma empresa pública de desenvolvimento de software. No entanto, a execução deste processo ainda não é automatizada. Por esta razão, dados históricos precisos não estão disponíveis. Logo, este estudo de caso irá apresentar alguns dados hipotéticos para ilustrar a aplicação da Metodologia através de todas as suas fases. Apesar da não existência de base histórica com dados reais de execução do processo, os dados propostos se aproximam da realidade, pois foram obtidos através de entrevistas e acompanhamento do trabalho da equipe de desenvolvimento.

Para aplicação da fase Iniciação, foi utilizado como insumo o Plano Estratégico da empresa de TI para derivar o seguinte objetivo para o programa de melhoria: *produzir processos simples, onde a execução das atividades seja otimizada, permitindo que os produtos sejam entregues dentro dos prazos.*

A principal ação prevista no Plano de Ação foi a formalização de um diagrama de atividades referente ao processo de Correção de Defeitos, descrito em termos de atividades e papéis responsáveis pelas atividades. O processo resultante é apresentado na Figura 9.

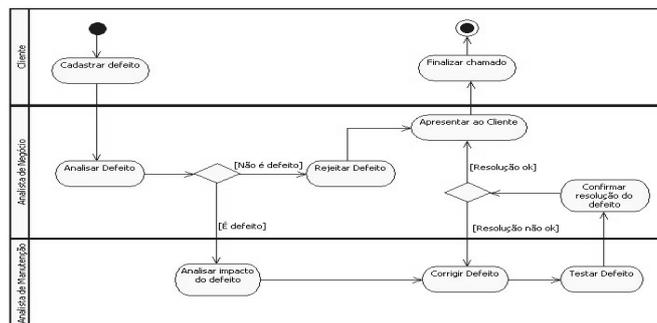


Figura 9: Processo de Correção de Defeitos.

Para atender o objetivo de melhoria e conseqüentemente as metas de negócio da organização, foi definida a métrica **Tempo de Resposta**, que corresponde ao tempo médio que se passa entre o momento que o cliente solicita a correção de um defeito até a homologação deste pedido ao cliente. Considerando a necessidade de que este tempo de atendimento seja otimizado, a meta de processo será a redução do tempo de resposta em 20% (parâmetro quantitativo para definir a melhoria a ser realizada).

Como forma de definir estratégias de implantação dos processos, no contexto da célula de manutenção da empresa de TI foram planejados (Plano de Implantação) e executados treinamentos sobre a execução das atividades do processo e sobre a utilização das ferramentas de testes que apoiarão a execução de uma das atividades (Testar Defeito).

Durante o estudo de caso também foi definida a infraestrutura de monitoração durante a confecção do Plano de Implantação. Considerando a métrica Tempo de Resposta, foi definido que a ferramenta de coleta utilizada será o Timesheet, que é uma

funcionalidade disponível na ferramenta de Gerência de Projetos adotada pela empresa e funciona como uma espécie de planilha de horas que resume todo o tempo gasto por um recurso durante a execução das atividades. Segundo procedimento de coleta definido no Plano de Implantação, o responsável pela execução de cada tarefa irá preencher semanalmente o timesheet com a quantidade de horas gastas na execução das tarefas.

Os tempos médios gastos com a execução das atividades foram estimados por participantes do processo. O valor da taxa de chegada foi calculado a partir de estimativas disponíveis na ferramenta adotada pela empresa para registro de solicitações do Cliente. As decisões do processo também foram retiradas de estimativas dos Analistas de Negócio, que afirmam que 20% das solicitações do cliente não costumam ser defeitos e 20% dos defeitos não costumam ter sua resolução confirmada.

Tabela 2: Histórico do estudo de caso (estimado).

Nome do Processo	Taxa de Chegada
Correção de Defeitos	12 defeitos / dia = 0,5 Cases/hora

ID	Lista de Atividades	Tempo de execução
S1	Cadastrar defeito	10 min
S2	Analisar defeito	20 min
S3	Rejeitar defeito	5 min
S4	Analisar impacto do defeito	1 hora
S5	Corrigir defeito	6 horas
S6	Testar defeito	30 min
S7	Apresentar ao Cliente	1 hora
S8	Confirmar resolução do defeito	30 min
S9	Finalizar chamado	5 min

Decisões do fluxo	Probabilidade
É defeito?	20% não
Resolução ok?	20% não

As informações resultantes da aplicação da metodologia foram utilizadas como insumo para a construção do Modelo GSPN, que pode ser visto na Figura 10.

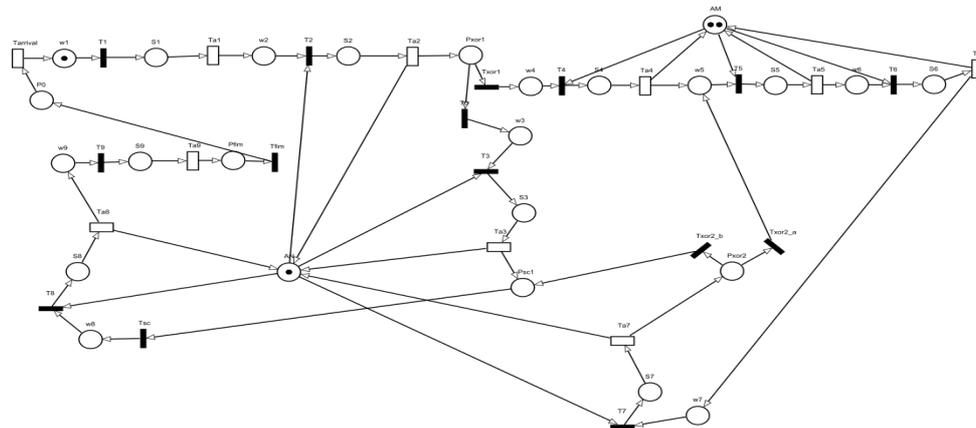


Figura 10: Modelo GSPN do Processo de Correção de Defeitos.

O modelo GSPN foi construído através da transformação manual dos elementos de processo em um modelo GSPN. Oliveira (2008) define o modelo em GSPN para cada elemento de um *workflow*, tais como atividades, papéis, operadores de paralelismo, sequenciamento, caminho alternativo, entre outros. Dada a não trivialidade desta transformação manual, sugere-se que ela seja feita automaticamente através de

ferramentas, como o BPEL2Net (Oliveira *et al*, 2011), que traduz descrições de processos executáveis descritos em BPEL (OASIS, 2007) em modelos GSPN.

Tendo modelado o processo em GSPN, o próximo passo foi a realização da Análise de Corretude do modelo, realizada através da ferramenta Time Net (Zimmerman *et al*, 2007) que demonstrou que o Modelo é *Sound*, pois não contém nenhuma tarefa desnecessária e todo case do processo é finalizado sem *tokens* remanescentes. A mesma ferramenta foi utilizada para simulações do processo, que foi realizada através do estado estacionário com 95% de confiança e tolerância a erro de 10%.

Durante a simulação foi considerado que a quantidade de recursos disponíveis para execução do processo era: um Cliente, um Analista de Negócio e dois Analistas de Manutenção. Em seguida foi calculado o tempo médio de execução por atividade com esta quantidade de recursos, levando ao total de 32,5 instâncias do processo (*cases*) em média. A partir deste valor, alcançamos o Tempo de Resposta total da execução do processo, que foi de 65 horas (equivalente a cerca de 8 dias úteis numa jornada de 8 horas). Este resultado indica que cada solicitação de correção leva em torno de 8 dias pra ser finalizada.

A meta de processo é diminuir em 20% o tempo de resposta de uma solicitação de correção, o que levaria o tempo de resposta para 52 horas (em torno de 6 dias e meio). Para que essa meta fosse alcançada, novas simulações foram realizadas, aumentando a quantidade de recursos do papel Analista de Manutenção para 5 e do papel Analista de Negócio para 3. Os resultados obtidos com a nova simulação foram: tempo de resposta de 51 horas e 25 instâncias do processo em média. Assim, foi possível confirmar que a meta de melhoria do processo almejada foi alcançada com o aumento da quantidade de recursos do sistema. Sugere-se que um novo ciclo de melhoria seja iniciado para colocar as alterações propostas ao processo em prática, através de um novo Plano de Ação.

6. Conclusões

O trabalho apresentado neste artigo teve como principal objetivo atender uma necessidade constante das empresas de desenvolvimento de software, que é entregar produtos de qualidade ao cliente dentro dos prazos estabelecidos. Este objetivo depende de diversos fatores, tais como a produtividade da equipe, a quantidade de recursos disponíveis, a existência de processos de apoio simples e eficientes, dentre outros.

Diante da subjetividade existente na implantação de processos, onde mudanças são feitas sem a garantia de que trarão benefícios em termos práticos à produtividade e prazos de entrega, buscou-se a utilização de Redes de Petri Estocásticas Generalizadas como técnica quantitativa para avaliação do desempenho dos processos de desenvolvimento.

Além da ênfase em utilizar técnicas quantitativas, a metodologia Petri-SW possui diversos pontos que merecem destaque no contexto da Qualidade de Software, como a possibilidade de realizar análises qualitativas para verificar a corretude da estrutura do modelo GSPN; a preocupação com o alinhamento estratégico e com os objetivos de negócio de uma organização; o uso de boas práticas de abordagens de melhoria de processos existentes e reconhecidas; a base em valores de envolvimento da

gerência com o programa de melhoria; a orientação à execução de pilotos e existência de atividades específicas para disseminação do conhecimento dos processos através de treinamentos e mentorizações.

Novas simulações com o detalhamento do contexto em que o processo está sendo executado podem ser realizadas como trabalhos futuros. Através deste detalhamento, será possível realizar análises de causas e fatores que influenciam na qualidade dos produtos entregues e no tempo de execução do processo.

Referências

- Associação para a Promoção da Excelência do Software Brasileiro - SOFTEX. (2011) “MPS.BR, Guia Geral 2011”. Disponível em <<http://www.softex.br>>.
- Bertollo, G. S., Falbo, R. B. (2006) “Definição de processos de software em um Ambiente de Desenvolvimento de Software Baseado em Ontologias”. In: V Simpósio Brasileiro de Qualidade de Software, Anais, 2006.
- Braghetto, K. R. (2011) “Técnicas de modelagem para a análise de desempenho de processos de negócio”. Tese de Doutorado, Universidade de São Paulo, 2011.
- CMMI Product Team. (2011) “CMMI for Development, version 1.3”. Disponível em <<http://www.sei.cmu.edu/cmmi/tools/dev/download.cfm>>.
- Dehnert, J., Freiheit, J. and Zimmermann, A. (2000) “Modeling and performance evaluation of workflow systems”, 4th World Multiconference on Systemics Cybernetics and Informatics SCI'2000.
- Dyba, T. (2005) “An Empirical Investigation of the Key Factors for Success in Software Process Improvement”. IEEE Transactions on Software Engineering, 2005.
- Hermanns, H., Herzog, U., e Katoen, J. Process algebra for performance evaluation. Theoretical Computer Science, 2002.
- IDEAL. (2008) “The IDEAL Model”. Software Engineering Institute. Disponível em: www.sei.cmu.edu/ideal. 2008.
- ISO/IEC 15504. (2003) “ISO/IEC 15504 - Information Technology - Process Assessment, International Standard (IS)”. 2003.
- Khokhar, M. N., Zeshan, K., Aamir, J. (2010) “Literature Review on the Software Process Improvement Factors in the Small Organizations”. New Trends in Information Science and Service Science, 2010 4th International Conference on.
- Li, S. and Zhu, H. (2008) “Generalized stochastic workflow net-based quantitative analysis of business process performance”. In ICIA'08: Proceedings of the IEEE International Conference on Information and Automation, pages 1040–1044, Washington, DC, USA, 2008. IEEE Computer Society.
- Maciel, P. R. M., Lins, R. D. e Cunha, P. R. F. (1996) “Introdução às Redes de Petri e Aplicações”. Departamento de Informática. Universidade Federal de Pernambuco. Escola de Computação, Campinas – SP, Julho de 1996.
- Marsan, M. A., Balbo, G., Conte, G. Donatelli, S., Franceschinis, G. Modelling with Generalized Stochastic Petri Nets. Università degli studi di Torino, 1995.

- OASIS. (2007) “Web service business process execution language (WS-BPEL) version 2.0”, [Online]. Available: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- Oliveira, C. A. L. (2008) “Uma Abordagem para Melhoria de Workflow Baseada em Redes de Petri Estocásticas Generalizadas”. Dissertação de Mestrado. Centro de Informática - Universidade Federal de Pernambuco, 2008.
- Oliveira, C. A. L., Lima, R. M. F., Reijers, H. A. and Ribeiro, J. T. S. (2011) “Quantitative Analysis of Resource-Constrained Business Processes”. IEEE Transactions on Systems, Man, and Cybernetics, 2011.
- Petri, C. A. (1962) “Kommunikation mit Automaten”. PhD thesis, Schriftedes IIM Nr. 2, Bonn, 1962.
- Rocha, A. R., Montoni, M., Santos, G., et al. (2005) "Dificuldades e Fatores de Sucesso na Implementação de Processos de Software Utilizando o MR-MPS e o CMMI", PROQUALITY (I Encontro de Implementadores de MPS.BR), 2005.
- Rozinat, A., Mans, R. S., Song, M. and van der Aalst, W. M. P. (2009) “Discovering simulation models”. Information Systems, 2009.
- Sharma, B., Sharma, Neeraj. e Sharma, Neeshu. (2009) “Software Process Improvement: A Comparative Analysis of SPI models”. Second International Conference on Emerging Trends in Engineering and Technology, ICETET 2009.
- van der Aalst, W. and van Hee, K. (2002) “Workflow Management. Models, methods and systems”. Eindhoven University of Technology, 2002.
- van der Aalst, W.M.P., Hee, K. M. V. and Houben, G. J. (1994) “Modelling and Analysing Workflow using a Petri-net based approach”. Proceedings of Second Workshop on Computer Supported Cooperative Work, Petri nets related formalisms.
- von Wangenheim, C., Hauck, J. C. R. e von Wangenheim, A. (2009) “Um modelo de Treinamento para Programas de Melhoria de Processo de Software”. Simpósio Brasileiro de Qualidade de Software, 2009.
- Wan, J., Luo, W., Wan, Xiaoyao. (2011) “Case study on Critical Success Factors of agile software process improvement”. Business Management and Electronic Information (BMEI), 2011 International Conference on.
- Zimmermann, A. and Knoke, M. (2007) “A Software Tool for the Performability Evaluation with Stochastic and Colored Petri Nets”. Technische Universität Berlin, Real-Time Systems and Robotics Group, 2007.