

Vitor Ricardo Pires Santos

**AVALIAÇÃO DE PADRÕES DE MODELAGEM DE PROCESSOS PARA A
GESTÃO DE RECURSOS HUMANOS EM PROCESSOS DE SOFTWARE**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para conclusão do curso de MBA em Tecnologia de Software.

São Paulo
2017

Vitor Ricardo Pires Santos

**AVALIAÇÃO DE PADRÕES DE MODELAGEM DE PROCESSOS PARA A
GESTÃO DE RECURSOS HUMANOS EM PROCESSOS DE SOFTWARE**

Monografia apresentada ao PECE – Programa de Educação Continuada em Engenharia da Escola Politécnica da Universidade de São Paulo como parte dos requisitos para a conclusão do curso de MBA em Tecnologia de Software.

Área de Concentração: Tecnologia de Software

Orientador: Prof. Dr. Rogério Rossi

São Paulo
2017

Catálogo-na-publicação

Santos, Vitor Ricardo Pires

Avaliação de Padrões de Modelagem de Processos para a Gestão de Recursos Humanos em Processos de Software / V. R. P. Santos -- São Paulo, 2017.

81 p.

Monografia (MBA em Tecnologia de Software) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1.Modelagem de processos 2.Gestão de recursos humanos 3.Padrões de modelagem de processos 4.Gestão de recursos humanos em processos de software I.Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

DEDICATÓRIA

*Dedico este trabalho a minha família
e a minha noiva pelo apoio e
paciência durante a realização deste
trabalho.*

AGRADECIMENTOS

Ao PECE – Programa de Educação Continuada em Engenharia que foi prestativo e proporcionou as condições necessárias para realização deste trabalho.

Ao professor Rogério Rossi que demonstrou dedicação e desempenhou um papel muito importante para a realização deste trabalho.

RESUMO

Em projetos de desenvolvimento de software os recursos representam um fator crítico para o sucesso do projeto. A conclusão de produtos de software com qualidade, dentro do prazo esperado e com o orçamento estimado, devem ser os objetivos do desenvolvimento de software. A modelagem de processos em geral, preocupa-se mais com o fluxo de execução do processo, negligenciando por vezes a devida alocação dos recursos humanos. Um processo de software pode ser modelado segundo padrões próprios de uma organização ou utilizando-se padrões estabelecidos e reconhecidos, como por exemplo, o modelo *workflow*. A modelagem de processos neste caso utiliza-se de uma linguagem denominada Linguagem de Modelagem de Processos (*Process Modeling Language*). Cada processo deve possuir um nível de abstração e uma perspectiva que sejam compatíveis com os objetivos da modelagem. Neste trabalho a perspectiva abordada para modelagem de processos se volta a devida alocação dos recursos humanos, ou seja, os elementos que representam os recursos humanos devem estar presentes no processo de software. Para modelar os processos na perspectiva de recursos humanos se apresentam os padrões de modelagem do modelo *workflow*. Esses padrões devem ser utilizados com o objetivo de realizar a gestão de recursos humanos de forma eficiente em processos de software. Para observar a aplicação dos padrões, pode-se utilizar uma ferramenta de modelagem e gestão de processos. Utilizou-se a ferramenta Bonita BPM que é viável e específica para este tipo de modelagem, apresentando os resultados conforme processos modelados de acordo com determinados padrões do modelo *workflow*.

Palavras-chave: Modelagem de processos. Gestão de recursos humanos. Padrões de modelagem de processos. Gestão de recursos humanos em processos de software.

ABSTRACT

In software development projects, resources represent a critical factor for the success of the project. The completion of quality software products, within the expected timeframe and with the estimated budget, be the goals of software development. Process modeling in general is concerned with the flow of process execution, sometimes neglecting a proper allocation of human resources. A software process can be modeled according to an organization's standards or using established and recognized standards, such as the workflow model. The modeling of processes in case of using a language called Process modeling language. Each process must have a level of abstraction and a perspective that are compatible with the objectives of the modeling. In this paper, an approach to process modeling is focused on the allocation of human resources, that is, the elements that represent the human resources are present not the software process. For human resource perspective process models, refer to the template workflow model patterns. These patterns should be used for the purpose of efficiently managing human resources in software processes. To observe an application of the patterns, a modeling and process management tool can be used. Use as a Bonita BPM tool that is feasible and specific to this model, presenting the appropriate results models according to the patterns of the workflow model.

Keywords: Process modeling. Human resource Management. Patterns of process modeling. Management of human resources in software processes.

LISTA DE ILUSTRAÇÕES

Pág.

Figura 1 – Níveis de maturidade do People-CMM	25
Figura 2 – Interação entre os elementos modelados	32
Figura 3 – Ciclo de vida de um item de trabalho.....	44
Figura 4 – Processo de desenvolvimento do sistema para incentivo da Xmarketing	63
Figura 5 – Alocação de recursos humanos às raias do processo BPMN	65
Figura 6 – Seleciona um usuário específico para cada raia	66
Figura 7 – Lista de processos para acompanhamento do gestor.....	67
Figura 8 – Lista de tarefas para serem alocadas	68
Figura 9 – Lista de tarefas de um usuário	69
Figura 10 – Alocando uma tarefa à um recurso humano	69
Figura 11 – A tarefa pode ser alocada ou desalocada	70
Figura 12 – Como criar variáveis para simulação do processo	71
Figura 13 – resultado da simulação do processo.....	72

LISTA DE TABELAS

Pág.

Tabela 1 – Comparação entre linguagens de modelagem de processos	
<i>Business Process Model and Notation</i> (BPMN)	36
Tabela 2 – Resultado da avaliação da Bonita BPM.....	73

LISTA DE ABREVIATURAS E SIGLAS

PML	<i>Process Modeling Language</i>
SEI	<i>Software Engineering Institute</i>
People-CMM	<i>People Capability and Maturity Model</i>
UML	<i>Unified Modeling Language</i>
BPMN	<i>Business Process Model and Notation</i>

SUMÁRIO

	Pág.
1. INTRODUÇÃO	12
1.1 Motivações	13
1.2 Objetivo	14
1.3 Justificativas	15
1.4 Estrutura do Trabalho	16
2. RECURSOS	18
2.1 O Que São Recursos	18
2.2 Gestão de Recursos	19
2.3 Gestão de Projetos de Desenvolvimento de Software	21
2.4 Gestão de Recursos em Projetos de Desenvolvimento de Software	23
2.5 Considerações Finais	25
3. MODELAGEM DE PROCESSOS	26
3.1 O Que é Processo?	26
3.2 Processos de Software	27
3.3 Modelagem de Processos de Software	29
3.4 Linguagens de Modelagem de Processos	33
3.5 Multivisão da Modelagem de Processos de Software	37
3.6 Simulação de Processos	38
3.7 Considerações Finais	41
4. PADRÕES PARA MODELAGEM DE PROCESSOS DE SOFTWARE PARA A GESTÃO DE RECURSOS HUMANOS	42
4.1 <i>Workflow</i> com ênfase em recursos humanos	43
4.2 Padrões para Modelar Processos de Software com Foco em Recursos Humanos	46
4.2.1 <i>Pattern R-DA Direct Allocation</i> (alocação direta)	47
4.2.2 <i>Pattern R-RBA Role-Based Allocation</i> (alocação baseada no papel/função).	47
4.2.3 <i>Pattern R-RF Retain Familiar</i> (alocação por familiaridade)	48

4.2.4 <i>Pattern R-CBA Capability-based Allocation</i> (alocação baseada na habilidade)	49
4.2.5 <i>Pattern R-HBA History-based Allocation</i> (alocação baseado no histórico)	50
4.2.6 Análise dos padrões de criação	50
4.3 Padrões que influenciam na gestão de recursos humanos	51
4.3.1 <i>Push Patterns</i> (padrões de empurrar)	51
4.3.1.1 <i>Pattern R-DBOS Distribution by Offer – Single Resource</i> (distribuição por oferta a um recurso)	51
4.3.1.2 <i>R-DBOM Distribution by Offer – Multiple Resources</i> (distribuição por oferta a um grupo de recursos)	52
4.3.1.3 <i>Pattern R-DBAS Distribution by Allocation – Single Resource</i> (distribuição por alocação de um recurso)	52
4.3.1.4 <i>Pattern R-RRR Round Robin Allocation</i> (alocação por rodízio)	53
4.3.1.5 <i>R-SHQ Shortest Queue</i> (fila mais curta)	53
4.3.1.6 Análise dos padrões da categoria push patterns	54
4.3.2 <i>Pull Patterns</i> (padrões para puxar)	54
4.3.2.1 <i>R-RIA Resource-Initiated Allocation</i> (alocação iniciada por recursos)	54
4.3.2.2 Análise dos padrões da categoria Pull Patterns	55
4.3.3 <i>Detour Patterns</i> (padrões de desvio)	55
4.3.3.1 - <i>Pattern R-E Escalation</i> (escalação)	56
4.4 Modelagem de processos de software e a gestão de recursos, além dos padrões	56
4.5 Considerações Finais	58

5 AVALIAÇÃO E USO DE FERRAMENTA DE GESTÃO DE PROCESSOS COM BASE NOS PADRÕES DE MODELAGEM DE PROCESSOS

5.1 Bonita BPM	60
5.2 Estudo de Caso	61
5.3 Resultados Obtidos	64
5.3.1 Padrões em tempo de modelagem	64
5.3.2 Padrões durante a execução do processo	67
5.3.3 Bonita BPM além dos recursos	71
5.4 Considerações Finais	73

6. CONSIDERAÇÕES FINAIS.....	75
6.1 Contribuições do Trabalho	76
6.2 Trabalhos Futuros	77
 7. REFERÊNCIAS	 78

1. INTRODUÇÃO

Em projetos de desenvolvimento de software os recursos humanos são um fator crítico para o sucesso do projeto (ACUÑA; JURISTO, 2004). No entanto, os recursos humanos, normalmente, não são representados nos processos de software.

Para Russell *et al.* (2005, p. 219), recursos são entidades capazes de realizar um determinado trabalho que podem ser utilizados em tarefas de um processo. Recursos podem ser entendido também, como ativos de uma empresa que são capazes de realizar tarefas de um processo (CESTARI FILHO, 2011, p. 10). Os recursos podem ser divididos em dois grupos: recursos humanos e recursos não-humanos. Este trabalho ficará restrito aos recursos humanos.

Gerir recursos humanos pode ser entendido como a realização de processos com o objetivo de organizar, gerenciar e guiar a equipe de projetos (*PROJECT MANAGEMENT INSTITUTE*, 2013, p. 255). A equipe de projetos é formada por pessoas que possuem funções, com responsabilidades de desempenhar tarefas, que contribuem com o objetivo do processo.

Na área de engenharia de software os recursos humanos devem ser alocados de forma eficiente, já que pode influenciar na qualidade do serviço entregue (CESTARI FILHO, 2011, p. 1). Com isso a gestão de recursos humanos tem grande impacto na qualidade do software entregue.

O processo de software é um processo criativo, que depende do julgamento e das capacidades humanas. Trata-se de um conjunto de atividades que tem como objetivo a produção de software (SOMMERVILLE, 2007, P. 42; LORSZ, 2001, p. 1).

Um processo de software deve ser modelado de acordo com padrões de uma organização ou utilizando padrões estabelecidos e conhecido, como por exemplo o modelo *workflow*, que pode ser entendido como um modelo de representação de processos de negócio na forma de um grafo ordenado (RUSSELL *et al.*, 2005, p. 218).

Um processo é modelado usando uma linguagem de modelagem de processos (*Process Modeling Language, PML*). Para que uma PML seja escolhida deve-se analisá-la de modo que todos os elementos do processo a ser modelado sejam representados. Com isso a PML deve possuir elementos que sejam capazes de expressar as características desejadas para um determinado processo.

Um processo é uma abstração da realidade, ou seja, é uma descrição simplificada desta realidade e nem todas as características são representadas (LORSZ, 2001, p. 4). Normalmente, um processo pode ser modelado com diversas perspectivas diferentes, cada uma com um conjunto de características diferentes, porém os processos devem ser equivalentes entre si. Neste trabalho será abordado os processos de software na perspectiva de recursos humanos.

Para Russell *et al.* (2005, p. 217), os processos devem ter mais detalhes sobre os recursos humanos, com isso pode-se ter mais controle sobre os mesmos. Tal objetivo pode ser alcançado com o uso dos padrões de modelagem de processos do modelo *workflow*.

1.1 Motivações

Muitas organizações de software não desenvolvem seus processos, pois aplicam processos de softwares prontos, criados como soluções para problemas de todas as organizações de softwares que os aplicam. Cada organização possui seu ambiente e sua cultura, os profissionais de uma organização podem se adaptar a um processo de maneira rápida e eficiente, enquanto os de outra podem ter dificuldades na aplicação do mesmo processo.

Mesmo que uma organização desenvolva seus próprios processos, de acordo com o seu ambiente e cultura, normalmente, quando um processo é modelado, preocupa-se com o fluxo de execução do processo, com isso elemento que representam os

recursos de uma organização são negligenciados, apesar da grande importância dos recursos na execução dos processos (RUSSELL *et al.*, 2005).

No ambiente de desenvolvimento de software, as pessoas que executam o processo são um fator de sucesso dos softwares desenvolvidos. Porém a alocação de recursos é feita de maneira intuitiva. Geralmente, os recursos são alocados para as tarefas sem que tenha um método ou um processo definido. Os processos de software são focados em tarefas e não nos requisitos que os recursos humanos devem ter para executá-las.

Para Otero *et al.* (2010), a conclusão de produtos de software com qualidade, dentro do prazo esperado e com o orçamento estimado representam um objetivo a ser buscado entre as organizações desenvolvedoras de software. Para alcançar tal objetivo pode ser crítico a alocação correta dos recursos humanos.

Pode-se observar que, assim como os processos devem ser adaptados e melhorados para cada organização, a alocação de recursos humanos também deve ser analisada de acordo com a cultura de cada organização. Com isso, pode ser analisado que um processo modelado por uma organização, deve representar os recursos humano da mesma.

1.2 Objetivo

Inicialmente, as pesquisas para a inicialização deste trabalho se deram em torno da modelagem de processo e simulação de processo para a gestão de recursos humanos. Após pesquisas bibliográficas, constatou-se a necessidade de que o escopo do trabalho fosse mais restrito. Com isso a pesquisa foi realizada com base na análise conceitual da modelagem de processos para a gestão de recursos humanos.

O objetivo principal deste trabalho é apresentar os padrões de modelagem de processos do modelo *workflow* e analisar como podem contribuir para a gestão de recursos humanos em processos de software.

Notou-se, através da pesquisa para cumprir o objetivo principal, a necessidades de aplicação dos padrões de modelagem de processo. Para isso pode-se utilizar uma ferramenta de gestão e modelagem de processos, fazendo uma avaliação da mesma, com o objetivo de analisar sua adequação aos padrões apresentados.

Como objetivo secundário, pretende-se a aplicar os padrões de modelagem de processos do modelo *workflow* a um processo que se encontra em execução, com o auxílio da ferramenta de modelagem e gestão de processos denominada Bonita BPM, além de avaliar se a ferramenta se adequa aos padrões apresentados.

1.3 Justificativas

Os recursos humanos são importantes para o desenvolvimento de software, que é uma atividade intelectual e depende do julgamento humano. Com isso gerir os recursos de forma eficiente pode ser crítico para o sucesso do software desenvolvido em uma organização.

Os padrões de modelagem de processos do modelo *workflow* podem ser uma maneira eficiente de modelar recursos humanos para que possam ser geridos em processos de software, facilitando a alocação de tarefas e a obtenção de melhor uso da equipe de desenvolvimento de software.

Os elementos que representam os recursos humanos podem ser aplicados nos processos de uma organização de software de modo que durante a execução do processo os recursos humanos sejam alocados e geridos de forma eficiente.

Pode-se observar que padrões que estabelecem os elementos de recursos humanos são aplicados aos processos; e como são alocados os recursos humanos durante a execução de um determinado processo; podem ser relevantes para a produção de software com maior qualidade.

Os padrões de modelagem de processos do modelo *workflow* podem ser aplicados com o auxílio de uma ferramenta de modelagem e gestão de processos, de modo

que a ferramenta auxilie também a gestão de recursos humanos de uma organização durante a execução do processo.

1.4 Estrutura do Trabalho

Este trabalho trata do uso da modelagem de processos, por meio da utilização dos padrões de modelagem do modelo *workflow* como uma forma de representar e gerir os recursos humanos.

Os recursos são definidos no capítulo 2. Aborda os recursos em projetos de desenvolvimento de software, que é um ambiente onde os recursos humanos representam um fator de sucesso e qualidade para o software produzido. Apresenta ainda, como são classificados os recursos e a importância da gestão dos mesmos.

O Capítulo 3 que aborda a modelagem de processos, apresenta o conceito de processos no ambiente de desenvolvimento de software. O capítulo aborda também a modelagem de processos e a abstração que um processo pode ter, dependendo dos objetivos ao modelá-lo; como a linguagem de modelagem que um processo deve ser modelado pode ser escolhida; e como um processo pode ser simulado para que seu comportamento possa ser entendido em diversas situações, sem precisar aplicá-lo na prática.

Os padrões de modelagem de processo do modelo *workflow* são apresentados no capítulo 4. Os padrões apresentados são analisados de que forma podem influenciar na gestão de recursos humanos, de modo que a organização que os aplicam possam ter os recursos humanos representados em seus processos. O capítulo aborda também, a importância das características humanas nos processos de software e como a organização pode realizar a alocação dos seus recursos humanos durante o fluxo de execução dos processos.

No capítulo 5 é feita uma avaliação de uma ferramenta de gestão e modelagem de processos. Para isto aplica-se os padrões de modelagem de processos apresentados no capítulo 4. Após testes em 3 ferramentas, uma denominada Bonita

BPM foi escolhida, que foi analisada e avaliada. Foi apresentado também, o resultado da simulação do processo modelado, onde o tempo de execução do processo foi avaliado.

As considerações finais deste trabalho são apresentadas no capítulo 6. O capítulo conclui que os padrões de modelagem de processos do modelo *workflow* podem ser utilizados para modelar processos com o objetivo de que se tornem úteis na gestão de recursos humanos, além da importância de aplicar os padrões em um processo com o auxílio da Bonita BPM.

2. RECURSOS

Este capítulo trata dos recursos em projetos de desenvolvimento de software. Na primeira seção é abordado o conceito de recursos, enquanto que a segunda relata como os recursos podem ser geridos. A terceira seção discute sobre projetos de desenvolvimento de software, já a quarta apresenta como a gestão de recursos é importante para a gestão dos projetos de software. Na quinta seção do capítulo as considerações finais são feitas.

2.1 O Que São Recursos?

De acordo com Russell *et al.* (2005, p. 219) recursos são entidades capazes de realizar um determinado trabalho que pode ser invocado em determinadas tarefas para que os objetivos das mesmas sejam alcançados. Recursos podem ser classificados em dois grandes grupos: recursos humanos e recursos não-humanos. Recursos não-humanos podem ser, por exemplo, os servidores necessários para o desenvolvimento de um determinado software ou até mesmo o dinheiro necessário para realizar o projeto.

Já em Cestari Filho (2011, p. 10) recursos são ativos de uma empresa que são capazes de realizar tarefas para que os objetivos de um projeto sejam alcançados.

Este capítulo tem como objetivo abordar o conceito de recursos no ambiente de desenvolvimento de software, então, neste ambiente os recursos podem ter classificações específicas, como a classificação feita por Pressman (2011, p. 603), em que os recursos são divididos em três grupos:

1. Recursos humanos: são as pessoas com suas habilidades e localizações;
2. Recursos de software reutilizáveis: são softwares que podem realizar parte ou toda uma funcionalidade do produto do desenvolvimento;
3. Recursos de ambiente: hardwares e softwares que fazem parte de um ambiente capaz de suportar um projeto de software ou um ambiente de engenharia de software.

De acordo com o que foi apresentado, recursos é um conceito bem amplo, no entanto, este trabalho está restrito aos recursos humanos e a gestão destes recursos. Neste trabalho a definição utilizada como base é a utilizada em Russell *et al.* (2005, p. 219).

Para Russell *et al.* (2005, p. 220), recurso humano pode ser definido como um membro de uma organização e que tem uma posição hierárquica, como, por exemplo, gerente de projetos. Ressaltando que, conforme Russell *et al.* (2005, p. 220) uma organização corresponde a um grupo de pessoas que executam tarefas com objetivos em comum, como, por exemplo, uma organização de desenvolvimento de software.

Existem algumas características dos recursos humanos que devem ser observadas no planejamento e gestão de projetos de desenvolvimento de software, que são as características que variam de pessoa para pessoa. Para Russell *et al.* (2005, p. 220) os recursos devem ser alocados para as tarefas de um determinado projeto de acordo com seu papel na organização, ou seja, um programador deve fazer tarefas relacionadas a programação de computadores. Outros pontos devem ser considerados, como a experiência do recurso humano com trabalhos executados anteriormente, habilidades e competências.

Nas próximas seções e capítulos quando o termo recurso for utilizado deve ser entendido como um recurso humano. Isto não significa que os conceitos não possam ser aplicados a recursos não-humanos, pois, podem haver similaridades.

2.2 Gestão de Recursos

Para o *Project Management Institute* (2013, p. 255), gerir recursos humanos implica em realizar alguns processos com o objetivo de organizar, gerenciar e guiar a equipe de projeto. A equipe de projeto é formada por pessoas que possuem papéis, por exemplo, desenvolvedores e analista, além de responsabilidades com as tarefas designadas, que contribuem para alcançar o objetivo do projeto. Cada membro de

uma equipe tem habilidades, únicas, para realizar o trabalho que lhe é proposto. Outro ponto que deve ser observado é o tempo que pode ser dedicado pelos membros da equipe, alguns podem se dedicar em período integral ou parcial ao projeto, ou ainda, podem ser retirados do projeto dependendo da necessidade. Vale ressaltar que projetos podem estar ocorrendo em paralelo e concorrendo pelos mesmos recursos humanos.

De acordo com o *Project Management Institute* (2013, p. 255) os membros da equipe de projeto devem participar do planejamento do projeto, pois diversos tipos de conhecimento podem ser considerados na hora de tomar decisões. Fazer com que os membros participem do planejamento pode proporcionar maior engajamento por parte da equipe. Alguma decisão tomada em tempo de planejamento pode impactar na qualidade do trabalho realizado, então a opinião técnica pode ser importante.

Como apresentado anteriormente, de acordo com *Project Management Institute* (2013, p. 255) o gerenciamento de recursos pode conter diversos processos:

- **Desenvolver o plano dos recursos humanos** – Este processo tem como objetivo identificar os papéis da equipe, as responsabilidades de cada membro, habilidades necessárias para desenvolver as tarefas do projeto, bem como a hierarquia e o plano de gerenciamento pessoal.
- **Mobilizar a equipe do projeto** – Verificar as disponibilidades dos recursos humanos para realizar as atividades do projeto, bem como a obtenção de equipe, visto que determinados recursos podem não estar disponíveis.
- **Desenvolver a equipe do projeto** – O objetivo é promover a melhoria das competências dos membros, a interação entre as pessoas e cuidar para que o ambiente seja próprio para que o projeto tenha o desempenho esperado.
- **Gerenciar a equipe do projeto** – Este processo tem como objetivo acompanhar o desempenho da equipe e fornecer *feedback*. Se problemas são detectados deve-se resolvê-los e promover melhorias para que o desempenho seja o esperado.

2.3 Gestão de Projetos de Desenvolvimento de Software

Apesar de o capítulo tratar sobre recursos, é importante discutir sobre a gestão de projetos, que tem como parte relevante a gestão de recursos.

Para o *Project Management Institute* (2013, p. 255), gerenciamento de projetos é “atender aos requisitos de um projeto por meio da aplicação de conhecimento, habilidades, ferramentas e técnicas às suas atividades”. Um gerente de projetos normalmente tem entre suas atribuições:

- Identificar requisitos;
- Cuidar para que as partes interessadas tenham suas expectativas atendidas, por meio de diversas abordagens, com relação ao planejamento e a execução do projeto;
- Definir uma forma de comunicação entre as partes interessadas do projeto;
- Gerenciar o atendimento dos requisitos do projeto e entregas;
- Resolver conflitos e inconsistências em relação a escopo, qualidade, cronograma, orçamento, recursos e riscos.

Segundo Sommerville (2007, p. 61), uma gestão eficiente não garante o sucesso do projeto de software, mas uma ineficiente geralmente ocasiona atraso, gasto além do previsto no orçamento, e ainda compromete a qualidade do produto.

Para Sommerville (2007, p. 61), um projeto de desenvolvimento de software pode ter as mesmas atividades que um projeto de outras áreas, porém tem particularidades, como as listadas a seguir:

- Software é intangível – trata-se de um produto em que é necessário consultar a documentação técnica para se ter noção do tamanho e progresso do desenvolvimento, não se pode tocar no software;
- Diversidade de processos – não há um processo de desenvolvimento de software que seja unânime. Mesmo com a escolha de um processo pode haver variações, dependendo da organização em que é aplicado e do software que está sendo desenvolvido, além de não estar imune a problemas que podem ocorrer durante

o desenvolvimento, como, por exemplo, uma tecnologia específica exigida pelo cliente;

- Cada projeto pode ser único – dois softwares podem ser desenvolvidos com o mesmo processo e ter as mesmas funcionalidades, porém podem ser únicos.

Gerenciar um projeto de desenvolvimento de software, de acordo com Sommerville (2007, p. 62), deve conter algumas atividades, são elas:

- Elaborar a proposta: a proposta deve relatar os objetivos do projeto, o modo como será desenvolvido, estimativas de custo e cronograma. Deve justificar junto ao cliente o motivo pelo qual a referida equipe deve ser escolhida para o desenvolvimento do software;
- Planejar e elaborar o cronograma: tem como objetivo identificar atividades, marcos e produtos gerados. O cronograma guia o desenvolvimento do software para que os objetivos sejam alcançados;
- Planejar o custo do projeto: é uma estimativa dos recursos necessários para alcançar os objetivos do projeto. Neste caso recursos são mencionados no conceito mais amplo, podendo ser recursos humanos e não-humanos;
- Monitorar e revisar o projeto: monitorar pode ser uma estratégia para evitar ou amenizar possíveis problemas que podem acontecer enquanto o projeto é executado ou atrasos de cronograma. A revisão tem como objetivo, principalmente em projetos de longa duração, alinhar o projeto com os objetivos da organização que o executa, podendo interromper o desenvolvimento;
- Selecionar e avaliar o pessoal: a equipe do projeto deve ser selecionada de acordo com a experiência e habilidades requeridas, porém nem sempre o gerente pode contar com a equipe ideal, pode ser por problemas de orçamento ou política da organização. Uma atividade importante é avaliar o trabalho dos membros da equipe.
- Elaborar relatórios e apresentações: algumas informações devem ser apresentadas ao cliente, para que o mesmo fique ciente sobre o progresso do desenvolvimento.

Para Pressman (2011, p. 567) a gestão de projetos de software deve ser focada em 4 Ps, referenciando-se a pessoas, produto, processo e projeto. A ordem dos Ps é

consistente com a importância do foco. Não se pode negar a importância dos outros três para o sucesso do projeto, no entanto, pessoas deve ser considerado prioridade, visto a importância do fator humano para o desenvolvimento de software. Sendo assim, na próxima seção serão abordados conceitos que consideram os recursos humanos em projetos de desenvolvimento de software.

Fazendo um paralelo entre os processos de gestão de recursos humanos e as atividades da gestão de projeto de software, pode-se notar que o processo de desenvolver o plano dos recursos humanos pode ser realizado na atividade de elaboração da proposta do projeto.

O processo de mobilizar a equipe do projeto pode ser executado na atividade de planejar e elaborar o cronograma. Já o processo de desenvolver a equipe do projeto e o processo de gerenciar a equipe do projeto podem ser executados na atividade de monitorar e revisar o projeto, bem como na de selecionar e avaliar o 'pessoal'.

2.4 Gestão de Recursos em Projetos de Desenvolvimento de Software

Na área de engenharia de software os recursos devem ser alocados de forma eficiente de maneira que a qualidade do serviço entregue seja percebida pelo cliente (CESTARI FILHO, 2011, p. 1). Dessa forma, o gerenciamento de recursos humanos tem grande importância e impacto sobre o produto final.

Segundo Santos *et al.* (2014, p. 30) os processos de desenvolvimento de software são altamente dependentes da equipe do projeto, que pode ser responsável por seu sucesso e a qualidade do software entregue. Com isso, pode-se afirmar que os recursos humanos de uma organização de engenharia de software devem ser gerenciados considerando-se as características únicas de cada pessoa da equipe, formando um ambiente complexo.

De acordo com Otero *et al.* (2010, p. 137) uma falha na gestão dos recursos humanos pode prejudicar um projeto de desenvolvimento de software permitindo que

somente uma pequena parte dos projetos sejam entregues conforme o prazo e o orçamento. Este fato pode ser atribuído à ineficiência na alocação de recursos, o que pode resultar em falta de qualidade do software entregue, insatisfação do cliente e uma reputação negativa para a organização responsável pelo desenvolvimento.

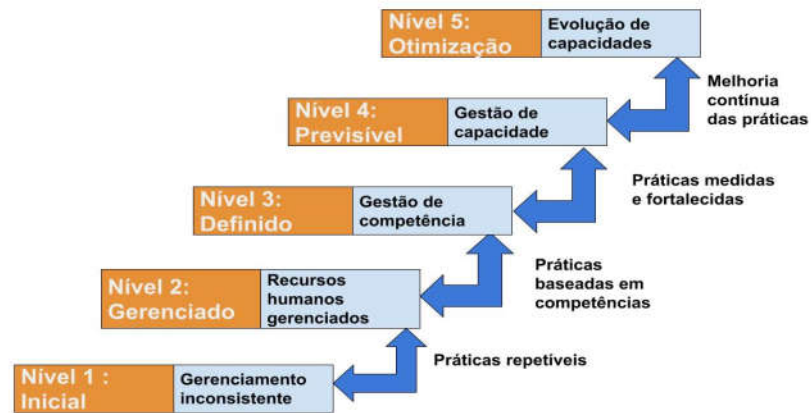
Os recursos humanos são importantes para o desenvolvimento de software, com isso o *Software Engineering Institute* (SEI) desenvolveu um modelo de maturidade e capacidade para recursos humanos em projetos de software denominado *People Capability and Maturity Model (People-CMM)* (PRESSMAN, 2011, p. 567).

People-CMM é um conjunto de práticas de gestão de recursos humanos e são divididas em cinco níveis de maturidade. Os níveis são evolutivos, ou seja, cada nível representa capacidade maior de gerir e desenvolver os recursos humanos em relação ao nível anterior, o que deixa a organização cada vez sofisticada em relação a gestão de recursos humanos. Um conjunto de práticas é incorporado à organização. As práticas são processos organizacionais que podem ser evoluídos e melhorados (CURTIS; HEFLEY; MILLER, 2009, p. 4). A figura 1 mostra os níveis evolutivos onde as práticas são inseridas, adaptadas à organização, medidas e fortalecidas, além de continuamente melhoradas de acordo com que a organização sobe de nível.

O *People-CMM* permite que a organização que o aplique ganhe cada vez mais capacidade de reter funcionários eficientes. A organização pode oferecer produtos com alto nível de qualidade e desempenho da equipe de trabalho (CURTIS; HEFLEY; MILLER, 2009, p. 9). Os benefícios podem ser acentuados a cada nível de maturidade que a organização vai alcançando.

Os arranjos entre os processos da gestão de recursos com as atividades dos projetos de desenvolvimento de software, além dos argumentos mencionados anteriormente mostram que o sucesso na gestão de recursos está ligado ao sucesso na gestão do projeto.

Figura 1 – Níveis de maturidade do People-CMM.



Fonte: Baseada em Curtis; Hefley e Miller (2009, p. 18).

2.5 Considerações Finais

Este capítulo apresentou os conceitos relacionados a recursos no ambiente de desenvolvimento de software bem como os de gestão de projetos de software, apresentando a importância da gestão de recursos para que o software produzido tenha qualidade e o projeto tenha sucesso.

3. MODELAGEM DE PROCESSOS

Este capítulo aborda a modelagem de processos. Na seção 3.1 é apresentado o conceito de processo, na seção 3.2 se conceitua o ambiente de desenvolvimento de software.

Na seção 3.3 a modelagem de processos é abordada, em seguida, na seção 3.4 é apresentado o conceito de linguagens de modelagem de processos. A abstração e o foco da modelagem de processos são abordados na seção 3.5, já a seção 3.6 apresenta uma maneira de simular a implementação de processos de software.

3.1 O Que é Processo?

De acordo com Cestari Filho (2011, p. 3), processo é um “conjunto de atividades definidas que combinam recursos e capacidades para realizar um objetivo específico, e que direta ou indiretamente cria valor para o cliente. Um processo possui uma ou mais entradas e as transforma em saídas definidas”.

Os processos são mensuráveis, ou seja, é possível medir, por exemplo, seus custos, orçamento, prazos e qualidade. Esta característica permite que o responsável por um determinado processo possa obter parâmetros para evoluí-lo e melhorá-lo, além de permitir a comparações com outros processos (CESTARI FILHO, 2011, p. 3).

Para Pressman (2011, p. 40), processo pode ser definido como um conjunto de atividades, ações e tarefas realizadas para criar um produto de trabalho para o cliente do processo. Para o autor os elementos do processo são:

- Atividade: esforço para atingir um objetivo que independe da natureza do processo, do tamanho, da complexidade. Pode ser usado como exemplo a comunicação entre os interessados do processo;
- Ação: corresponde a um conjunto de tarefas que tem como resultado um artefato fundamental para alcançar o objetivo do processo;
- Tarefa: esforço para atingir um objetivo tangível e bem definido, como por exemplo, fazer um teste de unidades de um determinado software.

O conceito de processo, conforme se apresentou, pode ser utilizado em diversas áreas, como processos de construção civil e processos administrativos, porém o foco deste trabalho são os processos de software, abordado na seção seguinte. Neste trabalho o conceito de processos que será utilizado é o apresentado em Pressman (2011, p. 40).

3.2 Processos de Software

Para Sommerville (2007, p. 42) e Lorsch (2001, p. 1) um processo de software é um conjunto de atividades que tem como objetivo a produção de um software. Processo de software é um processo intelectual e criativo, ou seja, depende do julgamento humano.

Como pode-se notar o conceito de processo de software é parecido ao de processo, a diferença está no produto final, que é um software.

O processo de software deve ser realizado pela equipe do projeto com uma estrutura organizacional e com o apoio de ferramentas técnicas e conceituais (LORSZ, 2001, p. 4).

Em ambientes de desenvolvimento de software o processo não tem uma forma fixa e imutável, ou seja, a equipe tem a liberdade de adaptar o processo de modo que as ações e tarefas possam ser adaptadas de acordo com a natureza do software ou da organização que executa o processo, por exemplo. Modificar o processo deve ter o objetivo de garantir a entrega dentro do prazo e qualidade exigida pelo cliente e usuários (PRESSMAN, 2011, p. 40).

Para Pressman (2011, p. 40), um processo de software deve ter cinco atividades básicas. São elas:

1. Comunicação: a comunicação é importante para manter os membros da equipe informados de todos os passos do desenvolvimento. A comunicação com o cliente e usuários do software desenvolvido é importante para que a equipe

entenda as funções que foram solicitadas de forma correta, enquanto que o cliente e os usuários podem garantir que a equipe está desenvolvendo o que foi pedido;

2. Planejamento: o objetivo do planejamento é fornecer à equipe do projeto um guia, que descreve: o cronograma, as tarefas, os artefatos entregues em cada fase do projeto, entre outras coisas;
3. Modelagem: para que a equipe de projetos tenha um entendimento mais consistente sobre o software que é desenvolvido é necessário utilizar modelos que especifiquem melhor a organização ou a estrutura do software, ou seja, pode ser feita uma modelagem da arquitetura do software ou dos elementos que compõem o software e até mesmo modelagens mais técnicas. Vale ressaltar que a modelagem apresentada é do software produzido e não do processo;
4. Construção: a atividade de construção consiste na geração de código fonte do software que é desenvolvido, seja ela manual ou automatizada por ferramentas. Faz parte da atividade de construção também os testes feitos pela equipe com o objetivo de encontrar erros de código, bem como possíveis falhas que possam ocorrer no software desenvolvido;
5. Entrega: entrega do software para que seja utilizado no ambiente proposto pelo cliente. Assim o cliente e usuários podem avaliar o software entregue e fornecer *feedbacks* para a equipe de projeto.

O processo de software pode conter mais atividades do que as listadas anteriormente, dependendo na natureza, complexidade ou tecnologias utilizadas no software que é desenvolvido, características da organização que executa o processo também são relevantes. Algumas atividades de apoio podem ser utilizadas, como as seguintes: controle e acompanhamento do projeto; administração de riscos; garantia de qualidade do software; revisões técnicas; gerenciamento de configuração do software (PRESSMAN, 2011, p. 40); dentre outras.

Cada organização de software pode executar o processo de software que mais se adapta a sua realidade, não existe um processo "ideal". A organização é livre para escolher as melhores práticas de engenharia de software e adaptá-las ao seu processo. Com o intuito de obter eficiência uma organização deve buscar melhorias no processo (SOMMERVILLE, 2007, p. 43).

O conjunto de processos de software de uma organização desenvolvedora de software pode conter dois conjuntos de subprocessos que se inter-relacionam: o de produção e o de gestão. O subprocesso de produção está ligado ao desenvolvimento do software, enquanto que o subprocesso de gestão está relacionado ao planejamento das atividades, alocação de recursos, entre outras coisas, ou seja, controla o subprocesso de produção. Este controle pode gerar informações sobre o subprocesso de produção. As informações podem ser analisadas e ajudar na melhoria do processo de software, o que aumenta a qualidade do software desenvolvido (LORSZ, 2001, p. 5).

3.3 Modelagem de Processos de Software

Criar modelos de processos de desenvolvimento de software é o objetivo da modelagem de processo de software. Um modelo do processo de software pode ser definido como uma representação abstrata da arquitetura, design ou definição do processo de software (LORSZ, 2001, p. 1). Cada representação descreve uma organização de elementos de um processo e fornece uma definição do mesmo para ser utilizada na avaliação e melhoria do processo. A representação pode ter diversos níveis de detalhes dos elementos de um processo.

Um processo pode ser analisado, validado e simulado. Numa organização, um modelo normalmente é utilizado para controle do processo de software, mas pode ser usado para prever possíveis problemas e promover automatizações com o uso de simulação de processos (LORSZ, 2001, p. 1).

De acordo com Lorsz (2001, p. 4), os modelos de processos de softwares podem ser utilizados para introduzir novos processos em uma organização, bem como treinamento e motivação para o 'pessoal' da organização que executa o processo.

A modelagem de processos pode auxiliar uma organização a atingir seus objetivos e obter outros benefícios, tais como: a melhoria da comunicação entre os membros da equipe e a melhoria da gestão de processo. Conforme apresentado em Curtis;

Kellner & Over (1992, p. 76), os cinco principais benefícios que uma organização pode alcançar com a modelagem de processos são os seguintes:

1. Facilitar o entendimento e a comunicação humana dado que a modelagem torna o processo formal para que as pessoas que o executam possam entendê-lo, assim a comunicação é facilitada;
2. Apoiar melhoria de processos pois ajuda na identificação de componentes necessários para que o processo tenha eficiência. A modelagem permite que processos diferentes sejam comparados;
3. Apoiar a gestão de processos dado que um processo pode ser modelado para que um projeto com características específicas seja desenvolvido. A modelagem permite que os processos de uma organização sejam monitorados, gerenciados e coordenados;
4. Automatizar a orientação do progresso do processo pois um ambiente de software eficaz deve ser definido. A modelagem deve fornecer orientação, sugestões e material de referência para facilitar o desempenho desejado e permite que o processo seja reutilizado; e
5. Automatizar o suporte a execução do processo dado que algumas partes de um determinado processo podem ser automatizadas, assim como a coleta de dados de medição de um processo. A modelagem apoia o trabalho cooperado entre os membros de uma equipe e a aplicação de regras para garantir a sua integridade.

De acordo com Lorsz (2001, p. 5), diferentes elementos do processo de software devem ser modelados. Os possíveis elementos a serem modelados são os seguintes:

- Agente ou ator: é uma entidade que executa o processo. Os atores podem ser classificados em dois grupos:
 - Atores humanos: são pessoas que desenvolvem o software ou estão envolvidos no processo de desenvolvimento de software;
 - Atores não-humanos: são sistemas, ferramentas ou *hardware* que estão ligados ao processo de software, como sistemas automatizam partes do processo, ferramentas que ajudam na produção de algum artefato ou computadores que podem ser usados para realizar alguma atividade.

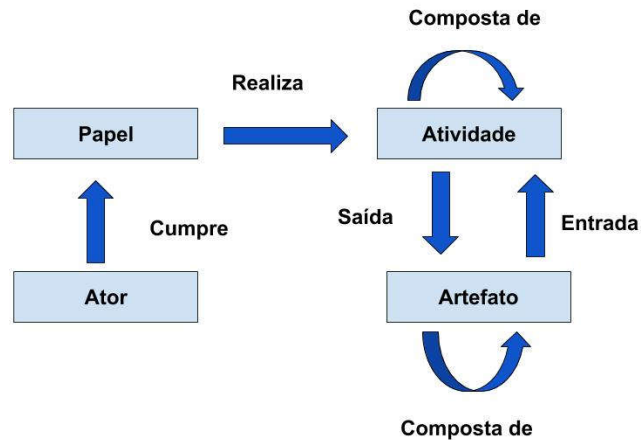
- Papel: corresponde a um conjunto de atores ou um grupo de responsabilidades, conhecimentos ou habilidades que são necessárias para desempenhar atividades específicas no processo de software;
- Atividade: é um elemento do processo que produz uma mudança no software que é produzido. Uma atividade deve ter uma entrada, uma saída e resultados intermediários. Atores devem desempenhar atividades, que podem estar associadas a artefatos ou a outras atividades;
- Artefato: é um subproduto do processo, que deve ser utilizado por outros processos ou pelo mesmo para ajudar a produzir outros artefatos ou alcançar os objetivos. Um artefato deve ser desenvolvido e mantido, pode ficar disponível para a organização por um tempo longo e deve ter diversas versões, à medida que é evoluído. Atividades podem criar, acessar ou modificar artefatos.

Lorsz (2001, p. 5) apresenta uma classificação dos elementos de um processo diferente da apresentada na seção 3.1, porém a diferença não invalida nenhuma das classificações, pois depende da abordagem do autor.

Já para Lonchamp (1993, p. 44), os elementos de um processo a serem modelados apresentam elementos dos dois autores já apresentados: atividade, tarefa, agente, artefato e papel. Além destes elementos, deve ser modelado também os recursos utilizados no processo. Vale ressaltar que os agentes correspondem aos recursos humanos do processo.

Como pode-se notar, os elementos de processos podem se relacionar com elementos que tem a mesma ou outra classificação, por exemplo, uma atividade pode estar relacionada a um agente ou à outra atividade, a figura 2 apresenta como os elementos interagem entre si. A modelagem tem como objetivo produzir um modelo de processo com a organização dos elementos e suas interações. Um ator cumpre um papel dentro de uma organização; cada papel tem a responsabilidade de realizar um conjunto de atividades; cada atividade pode ser dividida em outras atividades, além de utilizar artefatos como entrada, bem como produzir novos artefatos; e cada artefato pode ser dividido em mais de um artefato e é manipulado ou produzido em atividades.

Figura 2 – Interação entre os elementos modelados.



Fonte: Baseado em Lorsz (2001, p. 6).

Um processo pode ser modelado de acordo com os padrões da empresa ou utilizando um padrão conhecido, como o modelo de *workflow* (fluxo de trabalho). Um *workflow* ou modelo de *workflow* é representação de um processo de negócios, de modo que possa ser executado por um sistema gerenciador de *workflow* (fluxo de trabalho) Russell *et al.* (2005, p. 218). Pode ser utilizado um software capaz de seguir a sequência das tarefas, facilitando a gestão do processo.

Um processo que está representado no modelo de *workflow* deve ser representado por uma série de tarefas que estão conectadas na forma de um grafo dirigido. Cada tarefa executada no modelo *workflow* é denominada item de trabalho.

Para Lorsz (2001, p. 6) os processos podem ser modelados com diversos níveis de abstração, por exemplo, um gerente de uma organização necessita saber das atividades de um processo, sem detalhes de informações sobre as mesmas, enquanto que o gerente do projeto deve saber por quem, como e quando uma atividade irá ser realizada. Pode haver um modelo genérico ou um modelo detalhado com informações adicionais em relação aos elementos do processo. O nível de abstração depende do objetivo que o modelo quer alcançar, por exemplo, o modelo pode ser apenas um guia de arquitetura para os processos. Com isso, pode-se

perceber que as informações tratadas através dos modelos de processos podem ter diversos pontos de visão.

É importante estabelecer o objetivo para a modelagem do processo. A seção 3.4 aborda como as linguagens de modelagem de processo podem ser usadas para produzir um modelo de processo, já na seção 3.5 é apresentado como a modelagem pode representar os pontos de vista de um processo.

A organização pode ter a necessidade de fazer vários processos equivalentes com pontos de visões e abstrações diferentes, para que atendam a diversidade de interesses dentro da organização.

3.4 Linguagens de Modelagem de Processos

Informações devem ser integradas ao modelo de processo de software para que a organização de software possa entendê-lo, aplicá-lo e analisá-lo. Para isso existem as linguagens de modelagem de processos (*Process Modeling Languages - PML*) (CURTIS; KELLNER; OVER, 1992, p. 77; ZAMLI; LEE, 2001, p. 1).

Os processos utilizados no desenvolvimento de software impactam diretamente na qualidade do produto de software, com isso é importante gerenciar a criação, manutenção e melhoria dos processos de software. Os processos definidos por meio de uma PML podem orientar os atores de uma organização, impor regras e regulamentos, além de permitir a automatização de tarefas (ZAMLI; LEE, 2001, p. 1).

De acordo com Zamli e Lee (2001, p. 1), as PMLs devem ter determinadas características para que possam definir modelos de processos. As cinco principais características são as seguintes:

1. Suporte à modelagem: uma PML deve estar apta a representar, no mínimo, os elementos de processos como ator, papel, atividade e artefato. Deve também suportar abstração e modularização de elementos para realizar a construção de modelos;

2. Suporte à implementação: uma organização pode ter seus membros fisicamente distribuídos, então a PML usada na definição do modelo de processo deve permitir que cada membro possa receber partes do processo que são pertinentes ao seu papel na organização. Isso também torna possível o desenvolvimento incremental de um modelo de processo, de modo que a organização possa aplicá-lo de forma incremental;
3. Suporte à avaliação: um processo de software orienta de alguma forma o desenvolvimento do produto de software, então é necessário medir e avaliar o andamento e o desempenho do desenvolvimento;
4. Suporte à evolução: processos de software devem ser melhorados e evoluídos a medida que opções mais eficientes vão sendo propostas com base na avaliação do processo;
5. Suporte ao entendimento humano: as PMLs devem oferecer uma visualização que facilite a visualização do processo para que um agente humano de uma organização possa entender e executar o processo.

Existem diversos tipos de PMLs, com maior ou menor formalismo e diferentes tipos de focos. Linguagens muito formais podem oferecer complexidade para o uso, porém podem ser facilmente executadas por softwares de simulação de processos. Outras linguagens oferecem menor formalismo, porém oferecem uma capacidade semântica maior, como por exemplo as linguagens baseadas na *Unified Modeling Language* (UML), conforme Bendraou *et al.* (2010, p. 662).

Para Bendraou *et al.* (2010, p. 662), as PMLs baseadas em UML devem ser avaliadas de acordo com a sua capacidade de serem entendidas pelas pessoas ou ainda, se podem ser executadas por ferramentas de simulação. Com isso um conjunto de parâmetros de comparação de linguagens de modelagem foi estabelecido. Os parâmetros de avaliação de PMLs são os seguintes:

- Riqueza semântica: a capacidade que a PML tem de expressar o que é realizado durante a execução do processo de software modelado;
- Conformidade com o padrão UML: o quanto uma linguagem se baseia no padrão UML. Este tipo de avaliação com o padrão permite a construção de ferramentas para a utilização da PML;

- Representação gráfica e suporte a múltiplas visões: na avaliação por representação gráfica deve-se observar o quanto uma PML pode expressar com clareza o processo que foi modelado. O suporte a múltiplas visões corresponde a avaliar se uma linguagem de modelagem de processos pode oferecer diversas visões de um mesmo processo e se todas elas são equivalentes entre si.
- Executável: corresponde a avaliar se um PML gera modelos de processos de software capazes de serem simulados ou testados, além de permitir que ferramentas utilizem os modelos para apoiar a organização que os executa;
- Modularidade: entende-se como capacidade da PML de construir modelos de processos que podem ser unidos formando um processo maior. Um exemplo seria os processos de testes, codificação e análise formando um processo de maior de desenvolvimento de software;
- Formalismo: permite que uma PML seja validada, verificada e analisada de acordo com padrões definidos;
- Suporte de ferramentas: uma PML deve ter ferramentas que a apoie e a apliquem na modelagem de processos.

Os parâmetros apresentados servem, entre outras coisas, para comparar linguagens de modelagens de processos. Em Bendraou *et al.* (2010, p. 662), sete PMLs são comparadas de acordo com os sete parâmetros abordados, são as seguintes: SPEM1.1, SPEM2.0, Di Nitto's *et al.* (2002), Promede, *Chou's Approach* e UML4SPM. Cada linguagem recebe notas de acordo com o suporte que oferece para cada parâmetro. Se a linguagem não oferece suporte ao parâmetro a linguagem recebe nota zero, caso o suporte seja parcial a nota dada é um, já se o suporte for total a PML recebe nota dois. A tabela 1 apresenta os resultados da comparação das seis linguagens, a partir dela pode ser escolhido uma PML.

Deve-se analisar as características de uma PML antes de escolhê-la para realizar uma modelagem de processo, já que pode-se tentar modelar uma característica do processo que a PML não possui. O modo de comparação entre PMLs baseadas em UML apresentado por Bendraou *et al.* (2010) pode ser um modo prático e eficiente de escolher uma PML para modelar um processo.

Tabela1 – Comparação entre linguagens de modelagem de processos

Linguagens vs Parâmetros	SPEM1.1	SPEM2.0	Di Nitto's et al. (2012)	Promenade	Chou's Approach	UML4SPM
riqueza semântica	1	1	1	1	1	1
conformidade com o padrão UML	2	2	2	2	2	2
representação gráfica e suporte a múltiplas visões	2	1	2	2	2	1
executável	0	0	2	0	1	2
modularidade	1	2	0	1	0	1
formalismo	1	1	0	0	0	1
suporte de ferramentas	2	2	1	0	1	1

Fonte: baseada em Bendraou *et al.* (2010, p. 672).

Pode-se escolher uma PML baseada na utilização, ou seja, se muitas empresas de desenvolvimento de software utilizam uma determinada PML para modelar os seus processos, então pode ser uma escolha eficiente esta mesma PML. Uma notação que é muito utilizada para fazer a modelagem de processos de negócio, e é também utilizada em processos de software, trata-se da *Business Process Model and Notation* (BPMN) (DUMAS; PFAHL, 2016).

Para White (2004, p. 1) e o *Project Management Institute* (2011, p. 1), o principal objetivo BPMN é fornecer uma notação que seja facilmente compreensível por todos os interessados no processo modelado, com isso os usuários do negócio, os analistas e gerentes podem entender o processo. A notação é composta de um conjunto de elementos gráficos. Estes elementos permitem que um diagrama, representando um processo, seja facilmente desenvolvido e que seja compreendido pelos analistas de negócios. Com isso, a BPMN cria um canal comum de comunicação entre quem desenvolve e quem executa o processo.

A modelagem pode ter diversos tipos de visões de um mesmo processo, como será abordado na próxima seção, destacando-se a importância de uma PML suportar a criação de mais de uma visão, além de permitir a modelagem de elementos que são necessários a cada visão que se pretende modelar.

3.5 Multivisão da Modelagem de Processos de Software

Um modelo de processo de software pode estar focado nas relações que existem entre as atividades, enquanto outro focará nos atores que estarão envolvidos em cada atividade, ou seja, um modelo de processos de software tem diferentes pontos de vista, que devem ser equivalentes entre si (LORSZ, 2001, p. 4).

Como uma abstração da realidade, um modelo de processos representa uma descrição simplificada desta realidade, ou seja, nem todas as características são representadas. Geralmente, um modelo de processos pode ter diversos outros modelos que são equivalentes, porém tem perspectiva e pontos de vista diferentes (LORSZ, 2001, p. 4).

Para Curtis; Kellner & Over (1992, p. 77), um processo de software deve conter diversos tipos de informações para descrever adequadamente o processo. As informações extraídas de modelo de processo devem responder a cinco perguntas principais:

1. O que vai ser feito?
2. Quem vai fazê-lo?
3. Como e por que será feito?
4. Quando e onde será feito?
5. Quem é afetado?

De acordo com Curtis; Kellner & Over (1992, p. 77), para responder às perguntas apresentadas, um modelo de processos de software deve representar uma determinada perspectiva. As perspectivas que normalmente são representadas são as seguintes:

1. **Funcional:** representa os elementos de um determinado processo que estão sendo executados, bem como os dados, artefatos que estão sendo utilizado pelos elementos;
2. **Comportamental:** corresponde ao momento em que os elementos de processo são executados, assim como as situações em que ocorrem repetições de elementos, iterações, condições de tomadas de decisão, condições de entrada e saída, etc;

3. **Organizacional:** representa o lugar e por quais agentes os elementos de processo são executados. Representa também as políticas de comunicações entre os elementos, bem como a política de configuração;
4. **Informativo:** representa as entidades de informações que são manipuladas ou produzidas pelos elementos de processo. As entidades de informações podem ser representadas por dados, artefatos, produtos e objetos. Essa perspectiva tem como objetivo tanto as entidades de informações como as relações entre elas.

O que se pode notar é que uma organização desenvolvedora de software deve definir um processo em diferentes perspectivas de modo que as informações em cada perspectiva sejam relevantes para cada grupo de agentes. A organização deve garantir que as diversas perspectivas devem ser equivalentes. Vale ressaltar que um modelo de processo pode conter mais de uma perspectiva (LORSZ, 2001, p. 4).

3.6 Simulação de Processos

A simulação de processos pode ser usada para facilitar diversos tipos de atividades de uma organização, como por exemplo, o gerenciamento estratégico de desenvolvimento de software, melhoria de processos e gestão de projetos (KELLNER; MADACHY; RAFFO, 1999, p. 92).

É uma atividade que também pode ser utilizada como estratégia para prever o impacto de implantações e modificações de processos complexos ou que sejam críticos para o funcionamento da organização que os executa (KELLNER; MADACHY; RAFFO, 1999, p. 92).

Processos complexos e críticos têm como característica a incerteza, ou seja, não é trivial estabelecer o comportamento destes tipos de processos sob determinadas circunstâncias, algumas variáveis podem ser analisadas por meio de simulações. Com isso pode-se conhecer o comportamento antes da organização aplicá-lo em ambiente produtivo.

Por meio da simulação uma organização pode prever o comportamento que os processos executados podem ter. Caso alguma situação incomum aconteça, por exemplo, a mudança no quadro de funcionário de uma organização, pode ser simulada e analisada de modo que a organização possa se planejar e minimizar o impacto em seus processos.

Além das razões já abordadas, podem haver diversos motivos para que uma organização utilize a simulação de processos, dentre os quais se destacam:

- **Gestão estratégica:** a simulação pode ser útil para responder algumas dúvidas de gestão estratégica (WILLIFORD; CHANG, 1999, p. 203). Os gestores de uma determinada organização podem simular os processos com as diversas abordagens para responder questões como:
 - A organização deve realocar recursos humanos para um projeto novo ou contratar mais profissionais?
 - É melhor desenvolver um software que supra as necessidades da organização ou comprar um pronto?
- **Planejamento:** quando um processo é simulado a organização pode obter como resultado a estimativa de custo e tempo para realizar um projeto. O gestor pode simular também situações como a restrição e alocação de recursos e uma maneira eficiente de gestão de riscos. A organização pode simular modificações em seus processos de modo que se adaptem a situações específicas (POWELL; MANDER; BROWN, 1999, p. 151; RUS; COLLOFELLO; LAKEY, 1999, p. 174; PFAHL; LEBSANFT, 1999, p. 140);
- **Controle e gestão operacional:** durante a simulação de um processo podem ser extraídos valores para variáveis que representem o controle e a gestão operacional, então, quando o processo é executado pela organização, pode-se saber quais os resultados esperados e, se pode estar ocorrendo algo de indevido com a execução do processo;
- **Melhoria de processo e adoção de tecnologia:** muitas modificações dos processos podem ser sugeridas, porém uma organização nem sempre pode testar essas alterações, então a simulação de processos pode ser uma ferramenta útil para que a organização possa entender o impacto que as modificações podem causar em seus processos. As diversas variáveis de um determinado processo podem ser analisadas durante a simulação de processos

proporcionando assim mecanismos objetivos de melhoria de processos para que o mesmo se torne mais eficiente;

- Compreensão: uma organização pode entender seus processos por meio da simulação. Por exemplo, numa organização desenvolvedora de software, o gerente de projetos e os desenvolvedores podem entender melhor o fluxo do processo de desenvolvimento de software;
- Formação e aprendizagem: a simulação de processos pode ser útil para novos funcionários da organização que podem aprender mais sobre os processos em um ambiente controlado e simulado.

É possível que simular um determinado processo pode ser vantajoso para uma organização, permitindo-se que o processo se torne mais eficiente e a organização possa obter vantagens competitivas.

Para que a simulação de processos seja efetiva deve-se escolher com cuidado o que será simulado, por exemplo, deve-se escolher as variáveis que devem ser medidas, quais as situações serão simuladas, ou seja, quais serão os dados de entrada.

Foi abordado na seção 3.4 deste capítulo as PMLs (*Process Modeling Languages*) e como uma linguagem pode ser escolhida de acordo com os recursos que oferece. Para fazer a simulação de processos é necessário que a linguagem escolhida seja executável e tenha suporte de um software que possa simular o processo. A PML escolhida também deve oferecer recursos suficientes para que o processo seja modelado com a abstração e os detalhes necessários para a simulação. O software utilizado deve fornecer funcionalidades para que a simulação possa executar as situações desejadas, por exemplo, com a simulação de Monte Carlo pode-se simular as probabilidades de valores para as variáveis analisadas em um determinado processo.

3.7 Considerações Finais

Neste capítulo foi abordado o conceito de processos e como são os processos de software. Foi abordado as formas de modelagem de processos com suas diversas visões e abstrações utilizando uma PML. Outro ponto apresentado foi a simulação de processo como uma técnica para entender o comportamento do processo para então poder tomar decisões sobre ele.

4. PADRÕES PARA MODELAGEM DE PROCESSOS DE SOFTWARE PARA A GESTÃO DE RECURSOS HUMANOS

O objetivo da modelagem de processos é definir, controlar e gerenciar as atividades do processo de software. Os recursos humanos podem ser os fatores menos formalizados num processo de software, então, pode-se perceber que o processo onde os recursos humanos não são formalizados a gestão desses recursos fica comprometida. Contudo, a importância dos recursos humanos no processo de software pode ser observada (ACUÑA; JURISTO, 2004, p. 675; SOMMERVILLE; RODDEN, 1996).

Para Russell *et al.* (2005, p. 217), os processos devem oferecer mais detalhes sobre os recursos humanos, ou seja, um determinado processo que tenha sido modelado com foco no controle de fluxo, porém não detalha os recursos humanos que a executam não tem controle sobre os mesmos.

Neste capítulo se aborda a gestão dos recursos humanos como foco da modelagem de processos, bem como na gestão de recursos humanos utilizando o processo modelado, por meio da utilização de padrões.

Na seção 4.1 se apresenta o modelo *workflow* com foco em recursos humanos, ou seja, de que modo um item de trabalho pode ser alocado ou oferecido aos recursos humanos. Já na seção 4.2 são apresentados os padrões de modelagem de processos que tem como objetivo garantir a representação dos recursos humanos no processo. Enquanto que a seção 4.3 os padrões apresentados descrevem a gestão dos recursos humanos durante a execução do processo.

Os padrões apresentados podem ser utilizados como o auxílio de um software que tem como finalidade a modelagem e o controle de execução de processos de software, assim como é apresentado no capítulo 6. Aplicar os padrões pode proporcionar maior controle dos recursos humanos durante a execução do processo.

4.1 *Workflow* com ênfase em recursos humanos

Geralmente, para se executar um item de trabalho é necessário que um ou mais recursos humanos sejam alocados. Vale lembrar que item de trabalho pode ser entendido como uma tarefa em execução e os recursos humanos são membros de uma organização que são capazes de realizar um determinado trabalho.

Cada recurso humano dentro de uma organização tem uma posição específica e tem privilégios associados a ela. Os recursos humanos também podem estar agrupados em unidades organizacionais que realizam itens de trabalho relacionados a um conjunto de funções dentro das regras de negócio da organização, por exemplo, os desenvolvedores de uma organização desenvolvedora de software têm a função dentro da organização de desenvolvedor e estão alocados em um grupo de desenvolvimento, onde realizam itens de trabalho que estão relacionados à programação de software (RUSSELL, 2005, p. 220; RUSSELL, 2004, p. 3).

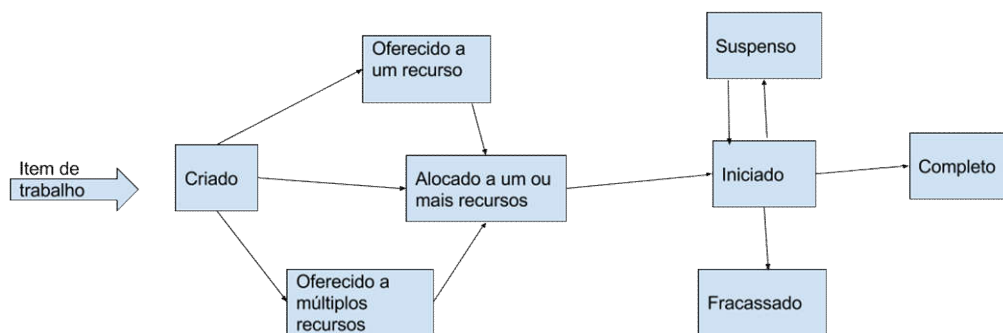
Cada recurso humano pode possuir habilidades e atributos pessoais que o qualificam para realizar itens de trabalhos específicos. As habilidades e atributos podem ser adquiridos em experiências anteriores, formação acadêmica ou até mesmo segundo suas próprias características pessoais (RUSSELL, 2005, p. 220; RUSSELL, 2004, p. 3).

Na perspectiva de recursos, os itens de trabalhos são vinculados a um recurso humano, de maneira que este recurso deve ser o responsável pela execução. A figura 3 ilustra desde o momento da criação de um item de trabalho até o momento da execução por um ou mais recursos humanos. Assim que o item de trabalho é criado ele tem três possibilidades:

1. É alocado a um recurso;
2. É oferecido a um recurso; ou
3. É oferecido a múltiplos recursos.

Quando um item de trabalho é oferecido a um ou mais recursos humanos, não há por parte dos recursos o compromisso de executá-lo. Por outro lado, quando um item de trabalho é alocado, há um compromisso por parte do recurso de executar a tarefa.

Figura 3 - Ciclo de vida de um item de trabalho



Fonte: (RUSSELL, 2005, p. 220; RUSSELL, 2004, p. 3).

Um software ou um recurso humano pode ser o responsável pelo controle do fluxo de execução de processo. Com isso o gestor do processo que tem a responsabilidade de oferecer ou alocar os itens de trabalho aos recursos humanos.

Nem sempre um item de trabalho termina com sucesso, sua execução pode terminar por motivo de falha ou ainda pode ser suspenso. No caso de o item ser suspenso, o mesmo pode voltar a ser executado.

O ciclo de vida de um item de trabalho apresentado na figura 3 é apenas um exemplo, podendo sofrer alterações de acordo com a exigência de cada situação. Quando os padrões de modelagem de processos são aplicados, o ciclo de vida apresentado pode sofrer alterações.

Em Workflow Patterns Initiative (2017), pode-se encontrar uma lista de diversos padrões para modelagem de processos, porém neste trabalho serão abordados apenas os relacionados aos recursos humanos. Os padrões de recursos humanos podem ser divididos em diversas categorias:

- *Creation patterns* (padrões de criação),
- *Push patterns* (padrões de empurrar) - são padrões onde o software ou gestor da execução do processo estabelece o recurso que realizará o item de trabalho,
- *Pull patterns* (padrões de puxar) - são padrões onde os recursos humanos podem assumir a responsabilidade de realizar um item de trabalho,
- *Detour patterns* (padrões de desvio),
- *Auto-start patterns* (padrões de início automático),
- *Visibility patterns* (padrões de visibilidade), e
- *Multiple resource patterns* (padrões de múltiplos recursos).

Os padrões abordados seguem o modelo *workflow*, mas não tem uma linguagem específica de modo que podem ser implementados por ferramentas que utilizam qualquer linguagem, desde que a linguagem de modelagem de processos (PML - *Process Modeling Language*) tenha a capacidade de expressar o padrão, ou seja, desde que seja possível especificar os elementos exigidos pelo padrão na PML escolhida.

Os padrões para modelagem de processos relacionados aos recursos humanos, neste trabalho, podem ter duas classificações:

1. Padrões em tempo de modelagem: são padrões que devem ser adotados no momento em que o processo é modelado. Detalhados na seção 4.2;
2. Padrões que influenciam a gestão dos recursos: são padrões que impactam diretamente na gestão de recursos, ou seja, são aplicados no momento em que o processo está sendo aplicado e instanciado. Serão abordados na seção 4.3.

Cabe ressaltar que os padrões apresentados seguem o modelo *workflow* e foram utilizados em Russell *et al.* (2004) e Russell *et al.* (2005) para realizar uma comparação entre ferramentas de modelagem e controle de processos.

Os padrões de modelagem de processos apresentados a seguir são apresentados por Russell *et al.* (2004, p. 19) e Workflow Patterns Initiative (2017), onde são abordadas todas as categorias e padrões relacionados com os recursos. Neste trabalho foram selecionadas algumas categorias e dentro dessas categorias alguns padrões. Os padrões selecionados são os mais simples dentro de cada categoria, ou seja, os padrões não apresentados são derivações dos apresentados ou apresentam muita similaridade com os apresentados.

4.2 Padrões para Modelar Processos de Software com Foco em Recursos Humanos

Para os padrões que são adotados no momento em que o processo é modelado se verifica apenas uma categoria que se refere a *creation patterns* (padrões de criação). Como o próprio nome sugere, os padrões que são classificados como padrões de criação têm como objetivo estabelecer regras para modelar processo com foco em recursos humanos, ou seja, como criar processos com elementos que representam os recursos humanos, tornando-o capaz de auxiliar a gestão de recursos humanos. Considerando-se o ciclo de vida do item de trabalho, os padrões da categoria *creations patterns* são aplicados no momento em que o item de trabalho é criado.

Os padrões de criação (*creations patterns*) abordam as limitações e exigências em relação a execução de um item de trabalho. As restrições podem ser as seguintes:

- Recursos humanos que devem executar uma determinada tarefa;
- O intervalo de tempo em que uma tarefa deve ser executada (quando);
- O local onde uma tarefa deve ser executada (onde).

Os padrões que são classificados como padrão de criação (*creations patterns*), estão definidos e detalhados nas subseções seguintes. O detalhamento do padrão é apresentado e, em seguida, são apresentadas suas contribuições à gestão de recursos humanos.

4.2.1 Pattern R-DA Direct Allocation (alocação direta)

O padrão *direct allocation* estabelece que o desenvolvedor do processo pode alocar um item de trabalho a um recurso específico, ou seja, um item de trabalho que tem como objetivo o desenvolvimento da comunicação com o banco de dados será realizado pelo “João”.

O padrão de alocação direta, no melhor dos casos, torna trivial a alocação de recursos humanos para os itens de trabalho que o aplicam. Entretanto, estabelecer um recurso humano específico para uma tarefa pode causar problemas:

- O recurso humano pode não estar disponível para realizá-lo;
- O recurso pode estar com muitas tarefas para realizar;
- Outros recursos humanos disponíveis e com igual capacidade de realizar a tarefa pode ficar ocioso;

Conforme apresentada, a abordagem de alocação direta deixa o processo estático, ou seja, o software ou gestor do processo não tem a opção de modificar em tempo de execução do processo o recurso estabelecido para realizar o item de trabalho.

Apesar de ser uma alocação de recursos humanos de fácil aplicação, a alocação direta pode causar problemas de gestão de recursos humanos. Com isso pode-se concluir que itens de trabalho que aplicam o padrão de alocação direta devem ser usados em tarefas que são críticas ao funcionamento do processo e tem um grande risco de falha.

4.2.2 Pattern R-RBA Role-Based Allocation (alocação baseada no papel/função)

A alocação de recursos humanos baseada no papel (*role-based allocation*) estabelece que um item de trabalho pode ser alocado para um papel específico dentro da organização, por exemplo, itens de trabalho que tem como objetivo a programação de alguma função de um determinado software só podem ser executados por programadores.

O padrão *role-based allocation* é comum em processos de software. A decisão de qual recurso humano será alocado para realizar o item de trabalho será tomada em tempo de execução do processo.

O gestor (software ou humano) responsável pela execução do processo, que deve alocar um recurso para um item de trabalho que aplica o padrão *role-based allocation*, tem à disposição todos os recursos humanos que tem o papel estabelecido.

É necessária uma análise dos recursos humanos disponíveis, já que pode ser que tenha diversas instâncias de diversos processos em execução. Então, o gestor do processo deve observar o andamento dos processos concorrentes e a disposição dos diversos recursos humanos possíveis, de maneira, que recursos humanos não fiquem sobrecarregados, enquanto outros ficam ociosos.

Ao contrário do padrão de alocação direta, o padrão de alocação baseada no papel dá ao processo mais liberdade ao gestor durante a execução, porém a escolha na alocação do recurso humano é mais complexa.

4.2.3 Pattern R-RF Retain Familiar (alocação por familiaridade)

O padrão *retain familiar* estabelece que um item de trabalho deve ser realizado pelo recurso humano que executou o item anterior no processo. Esta abordagem pode ser vantajosa, do ponto de vista do andamento do processo, já que o recurso que está familiarizado com o um item de trabalho pode ter mais facilidade ao executar o seguinte, desde que tenha capacidade e habilidade para realizá-lo, a comunicação é facilitada.

A gestão dos recursos humanos, no caso em que um item de trabalho aplica o padrão *retain familiar*, é facilitada para a alocação, já que é o mesmo do item anterior. A dificuldade pode surgir quando o recurso humano responsável, por algum motivo externo, fica indisponível.

4.2.4 Pattern R-CBA *Capability-based Allocation* (alocação baseada na habilidade)

A alocação baseada na habilidade (*capability-based allocation*) é o padrão que estabelece que uma tarefa deve ser executada por um recurso humano que tem uma habilidade específica, por exemplo, uma organização tem uma tarefa do seu processo de software para modelar um banco de dados, porém existe a restrição que o analista de banco de dados deve ter a habilidade de modelar o banco de dados no sistema gerenciador de banco de dados MySQL.

As habilidades dos recursos humanos são avaliadas e a alocação é decidida durante a execução do processo, com isso torna-se necessário mapear as habilidades de todos os recursos humanos disponíveis. Contudo, há duas abordagens possíveis para aplicação do padrão *capability-based allocation*: a primeira é a abordagem onde o gestor da execução do processo pode decidir dentre os recursos disponíveis qual será alocado para o item de trabalho em questão; na segunda o recurso humano pode perceber que tem a habilidade de realizar o item de trabalho a ser alocado.

Independente da abordagem adotada, o gestor deve atentar a execução de itens de trabalho que exijam habilidades que não são frequentes na organização. Pode ser que no momento necessário para a realização de um item de trabalho não haja recursos humanos disponíveis com a habilidade requerida ou nenhum recurso humano se disponha a realizar o item de trabalho. Esta situação deve ser analisada pelo gestor, que com o objetivo de resolver possíveis atrasos no processo, pode avaliar se a solução é contratar novos recursos para a organização, ou remanejar recursos de outras equipes dentro da organização, ou capacitar os recursos que já existem na equipe que aplica o processo. Cada uma das situações deve ser analisada com relação a custo, prazo e qualidade ao aplicá-la.

4.2.5 Pattern R-HBA History-based Allocation (alocação baseado no histórico)

O padrão *history-based allocation* (alocação baseada no histórico) estabelece que os itens de trabalho sejam alocados com base no histórico de trabalho dos recursos humanos. Com isso pretende-se que a tarefa seja executada por quem tem mais experiência com itens de trabalho similares ou até mesmos na mesma tarefa de outras instâncias do processo.

Para aplicar o padrão de alocação baseado no histórico, a organização deve manter os dados referentes à realização de tarefas para cada recurso, de modo que em tempo de execução o gestor possa tomar a decisão de qual recurso alocar. Dentre os dados armazenados sobre as tarefas executadas por um dado recurso, podem ser citados o tempo de execução da tarefa e se houve sucesso ao executá-la.

Pode-se perceber que a gestão de recursos humanos fica mais complexa com aplicação do padrão de alocação baseado no histórico, já que todos os dados referentes à realização dos itens de trabalho devem ser armazenados para todos os recursos, bem como devem ser analisados durante a alocação dos recursos humanos aos itens de trabalho.

4.2.6 Análise dos padrões de criação

Os padrões de criação têm grande importância para a gestão de recursos humanos, já que podem influenciar a alocação dos recursos humanos durante a execução do processo. Todas as restrições e exigências inseridas nos processos devem ter impacto na gestão de recursos humanos.

Alguns padrões exigem maturidade da organização na gestão de recursos humanos, por exemplo, os padrões de alocação baseada nas habilidades e alocação baseada no histórico dos recursos humanos. Estes dois padrões exigem que a organização mantenha registros sobre as habilidades dos recursos humanos, bem como coleta de dados para controlar o histórico de realização dos itens de trabalho durante a execução dos processos.

4.3 Padrões que influenciam na gestão de recursos humanos

Durante a execução de um processo, podem ser usados alguns padrões que influenciam na gestão de recursos humanos. As subseções abordadas a seguir abordam as categorias de padrões que tem como objetivo estabelecer maneiras de como pode ser feita a gestão de recursos humanos em tempo de execução de um determinado processo.

4.3.1 *Push Patterns* (padrões de empurrar)

A categoria de padrões *push patterns* tem como característica a ação de alocar ou oferecer itens de trabalho para os recursos humanos. A alocação ou oferecimento é feita com base nas exigências e restrições estabelecidas para cada item de trabalho.

4.3.1.1 *Pattern R-DBOS Distribution by Offer – Single Resource* (distribuição por oferta a um recurso)

Ofertar um item de trabalho a um único recurso é o que estabelece o padrão *Distribution by Offer – Single Resource* (distribuição por oferta a um recurso). Vale ressaltar que é feito apenas o oferecimento do item de trabalho ao recurso humano, não há o compromisso, ainda, do recurso com a realização do item de trabalho.

Os recursos devem ser notificados em relação ao item de trabalho e podem optar por aceitar ou não sua alocação ao determinado item de trabalho. O gestor do processo deve ficar atento para possíveis atrasos na resposta do recurso já que pode também atrasar a realização do item de trabalho e por consequência o processo. Caso o recurso rejeite o item de trabalho o mesmo deverá ser oferecido a outro recurso. Esta situação deve ser observada, já que se diversos recursos rejeitarem os itens de trabalho, o andamento do processo pode ser prejudicado.

4.3.1.2 *R-DBOM Distribution by Offer – Multiple Resources* (distribuição por oferta a um grupo de recursos)

De acordo com o padrão *Distribution by Offer – Multiple Resources* (distribuição por oferta a um grupo de recursos), determinada tarefa pode ser distribuída a um grupo de recursos humanos. Todos os recursos humanos que pertencem ao grupo saberão que a tarefa deve ser executada, então qualquer recurso do grupo pode se candidatar para realizar a tarefa. Um exemplo deste padrão é quando uma tarefa de desenvolvimento de uma determinada classe de um software é alocada a todos os desenvolvedores, então um deles pode se candidatar para desenvolver a classe.

Os membros do grupo podem ficar na indecisão em relação a quem deve realizar o item de trabalho, então o gestor do processo deve estar atento a isto, já que pode ocasionar atrasos na execução do processo. Uma vantagem ao oferecer para um grupo e não a um único recurso é que a tarefa não corre o risco de ser rejeitada diversas vezes.

4.3.1.3 *Pattern R-DBAS Distribution by Allocation – Single Resource* (distribuição por alocação de um recurso)

O padrão *distribution by allocation - single resource* (distribuição por alocação de um recurso) é parecido com os padrões de distribuição por oferta. Porém em vez de oferecer, como o nome do padrão sugere, o padrão de distribuição por alocação aloca o recurso a um item de trabalho.

Como o padrão de distribuição por alocação estabelece um recurso para realizar um item de trabalho, fica sob a responsabilidade do gestor do processo observar e analisar a disponibilidade dos recursos de modo que estes não fiquem sobrecarregados, enquanto outros fiquem ansiosos.

4.3.1.4 *Pattern R-RRA Round Robin Allocation* (alocação por rodízio)

O padrão *round robin allocation* (alocação por rodízio), estabelece que ocorra um rodízio entre os recursos humanos disponíveis para executar o processo, de modo que os recursos tenham a mesma quantidade de itens de trabalho. Pode-se perceber que o padrão de alocação por rodízio pode ser muito vantajoso para a gestão de recursos humanos, principalmente se a alocação for feita por software, já que a alocação de itens de trabalho fica restrita a manter o equilíbrio entre os recursos humanos.

O gestor do processo deve ficar atento às restrições e exigências dos itens de trabalho, o que pode impedir que a distribuição dos itens seja uniforme. Outra situação a se observar é que tarefas diferentes tem complexidade de execuções diferentes, com isso mesmo que recursos humanos tenha o mesmo número de itens de trabalhos alocados, pode haver sobrecarga devido à complexidade das tarefas a serem executadas.

4.3.1.5 *R-SHQ Shortest Queue* (fila mais curta)

Normalmente, em uma organização desenvolvedora de software as equipes de desenvolvimento trabalham em mais de um projeto, então um desenvolvedor pode ter mais de uma tarefa sob sua responsabilidade. O padrão de fila mais curta é o padrão que permite que o gestor do fluxo do processo possa alocar uma tarefa para o recurso humano que tenha menos tarefas alocadas.

Uma variável que pode ser relevante para a escolha do recurso a ser alocado é o tempo que levaria para o item de trabalho ser realizado, porém essa abordagem de escolha do recurso exige uma análise dos dados sobre o desempenho dos recursos, bem como a complexidade da tarefa a ser executada. Assim como o padrão apresentado anteriormente, o gestor do processo deve ficar atento às exigências e restrições das tarefas.

4.3.1.3 Análise dos padrões da categoria *push patterns*

Os padrões da categoria *push patterns* são os mais simples, já que empiricamente, é natural que haja um gestor que delegue as tarefas de um processo aos recursos humanos de uma organização, apesar de não ser imperativo em padrões onde definem que um item de trabalho deve ser oferecido a um ou mais recursos humanos.

Os padrões que definem o oferecimento de itens de trabalho a um ou mais recursos exigem vigilância do gestor do processo, já que a demora ou a recusa dos itens de trabalho por parte dos recursos humanos podem ocasionar atrasos na execução do processo.

Os padrões de fila mais curta e alocação por rodízio exigem da organização que os aplica controle com relação as atividades dos seus recursos humanos, porém estes padrões buscam que os recursos humanos tenham cargas de trabalho parecidas.

4.3.2 *Pull Patterns* (padrões para puxar)

Os padrões que pertencem à categoria *pull patterns* tem como característica o foco nos recursos humanos. Um determinado recurso assume a responsabilidade de realizar um item de trabalho que está disponível para ser realizado, ou seja, quem tem a atitude de alocar é o recurso humano. Um padrão é abordado a seguir como exemplo da categoria, os demais padrões podem se derivações deste.

4.3.2.1 *R-RIA Resource-Initiated Allocation* (alocação iniciada por recursos)

Dada uma lista de itens de trabalho que estão disponíveis para serem realizados, um recurso humano pode assumir o compromisso de realizar um ou mais itens, mesmo

que não seja imediatamente a alocação. Isto é o que estabelece o padrão *resource-initiated allocation* (alocação iniciada por recursos).

O padrão exige vigilância por conta do gestor de execução do processo em execução, já que se os recursos ficam responsáveis pelas escolhas dos itens de trabalho, itens de trabalho podem ficar sem escolha. Esse padrão pode estar associado ao padrão *Distribution by Offer – Multiple Resources*, onde os itens de trabalho são oferecidos a um grupo de recursos humanos.

4.3.2.2 Análise dos padrões da categoria *Pull Patterns*

Os padrões da categoria *pull patterns* tem como característica principal a proatividade dos recursos humanos, já que são estes que são responsáveis pela alocação dos itens de trabalho.

A categoria *pull patterns*, por contar com a colaboração dos recursos humanos na alocação dos itens de trabalho, exige maturidade dos recursos, já que podem ficar ociosos sem se comprometerem com tarefas, causando atrasos na execução dos processos.

4.3.3 *Detour Patterns* (padrões de desvio)

Padrões que pertencem a categoria *detour patterns* (padrões de desvio), tem como objetivo deixar o processo mais dinâmico. Permite se realizar ajustes nas alocações dos recursos, além da interrupção de itens de trabalho. O ajuste na alocação pode ser realizado pelo recurso, que por algum motivo não consegue dar prosseguimento à execução do item de trabalho ou pelo gestor da execução do processo.

4.3.3.1 - *Pattern R-E Escalation* (escalação)

O padrão *escalation* (escalação) estabelece que o gestor tem a possibilidade de retirar a alocação um item de tarefa que estava sob responsabilidade de um recurso humano, para que seja alocado a outro recurso.

A mudança na alocação do recurso pode ocorrer para que o item de trabalho seja executado com maior agilidade ou a tarefa pode estar com o prazo excedido, então deve ser alocado a um recurso humano que tenha disponibilidade imediata. Outra possibilidade é a do gestor tentar equilibrar o esforço entre os recursos humanos, de modo que todos fiquem com carga de trabalho parecidas.

Pode-se observar que o padrão *escalation* permite que a gestão de recursos humanos durante a execução seja dinâmica, sempre com o objetivo de melhorar e agilizar a execução das tarefas.

4.3.3.1 Análise dos padrões da categoria *Detour Patterns*

Os padrões da categoria *detour patterns* são importantes para corrigir problemas na alocação dos itens de trabalho, porém devem ser usados com cautela já que mudanças na alocação durante a execução dos itens de trabalho podem levar a atrasos na execução do processo.

Pode-se observar quando um item de trabalho é executado por um recurso humano e deve ser alocado a outro, dados podem ser perdidos, ocasionando retrabalho durante a execução do item.

4.4 Modelagem de processos de software e a gestão de recursos, além dos padrões

Os padrões oferecem uma maneira de modelar e controlar recursos humanos em um processo de software. Os padrões supracitados abordam habilidades técnicas,

porém não tratam das características humanas ou da personalidade de cada recurso humano de uma organização.

As organizações desenvolvedoras de software apresentam uma alta dependência das habilidades pessoais dos seus engenheiros de software e da sua equipe de desenvolvimento, então, caso o processo tenha elementos que representam estas habilidades, a alocação de tarefas pode ser mais eficiente, podendo aumentar a qualidade do software desenvolvido (WANG, 2006, p. 204).

De acordo com Acuña; Gómez & Juristo (2009, p. 627), os recursos humanos são fatores fundamentais e críticos para o sucesso do processo de desenvolvimento de software. Com isso as 'pessoas' devem ser incorporadas ao processo de software. Alguns estudos como Acuña & Juristo (2004), Moore (1991), Turley & Bieman (1995) e Wynekoop & Walz (2000) apresentam análises para poder estabelecer um relacionamento entre as tarefas de processos e os recursos humanos, através das características pessoais de cada recurso. É natural que as pessoas se comportem de maneira não determinística e subjetiva, porém essa característica tem impacto no software produzido, já que é uma atividade intelectual e social.

A modelagem de processo pode não especificar as habilidades e aptidões de cada membro de uma organização. Os membros podem realizar tarefas que estão aquém ou além das suas características profissionais e pessoais. Com isso pode haver dificuldade em se realizar a gestão dos recursos humanos com base nas suas características pessoais, ou seja, as tarefas estabelecidas para um determinado membro de uma organização podem não estar de acordo com suas aptidões (HUMPHREY, 1996) (SLOMP; MOLLEMAN, 2002). É necessário que a modelagem de processos de software consiga representar não só a parte técnica, mas também as características humanas.

Neste capítulo se abordou modelagem de recurso com base em grupos e papéis pertencentes a uma organização desenvolvedora de software, porém foi assumido que os recursos humanos que fazem parte destas divisões foram assim agrupados de acordo com características pessoais.

Em Acuña & Juristo (2004), é apresentado uma relação das habilidades individuais de cada recurso humano com o papel que ele exerce em uma organização desenvolvedora de software. Com isso pode-se notar que se cada recurso humano exerce o papel que lhe é mais conveniente, a organização pode obter maior eficiência na execução de seus processos. Já em André; Baldoquín & Acuña (2011) é proposto um modelo, baseado em um questionário psicológico, que tem como objetivo alocar recursos humanos a papéis do processo de desenvolvimento de software e então pode-se formar uma equipe completa.

4.5 Considerações Finais

Os padrões do modelo *workflow* podem ser uma alternativa viável para a gestão de recursos humanos, pois permitem que os elementos que representam esses recursos sejam inseridos na modelagem dos processos, bem como soluções para a gestão dos recursos humanos durante a execução dos processos.

Os padrões partem da premissa de que os recursos humanos ocupam papéis ou desempenham funções dentro de uma organização de acordo com suas características e habilidades. Com isso pode-se perceber que a alocação de recursos humanos aos itens de trabalhos dos processos executados pode ganhar em qualidade e desempenho na sua realização.

No capítulo 6 os padrões apresentados neste capítulo são aplicados em um processo utilizando uma ferramenta de modelagem e execução de processos. É feita a observação de quais padrões são suportados pelo software utilizado.

5. AVALIAÇÃO E USO DE FERRAMENTA DE GESTÃO DE PROCESSOS COM BASE NOS PADRÕES DE MODELAGEM DE PROCESSOS

Os padrões de modelagem de processos foram apresentados no capítulo 4 com a análise de como podem influenciar na gestão de recursos humanos. Com isso faz-se necessário a aplicação dos padrões em um processo, com o auxílio de uma ferramenta de modelagem de processo. Aplicar os padrões tem como objetivo entender como os mesmos podem influenciar na modelagem de um processo e como é importante a escolha da ferramenta de modelagem e gestão de processos.

Pode ser encontrado em Russell *et al.* (2004), Russell *et al.* (2005) e Workflow Patterns Initiative (2017) diversas comparações de ferramentas de modelagem e gestão de processos. As comparações são realizadas com o uso dos padrões de modelagem de processos do modelo de *workflow*. Em *Workflow Patterns Initiative* (2017) pode ser encontrado diversas comparações entre ferramentas. A ferramenta utilizada neste trabalho foi avaliada com relação os padrões de modelagem de processos que estão relacionados aos recursos.

Para escolher a ferramenta que auxilia na aplicação dos padrões de modelagem de processo, fez-se necessário observar as seguintes premissas: 1) se a ferramenta pode modelar processos de acordo com o modelo *workflow*; 2) se a ferramenta possui uma interface intuitiva; e 3) se a ferramenta possui controle de fluxo da execução do processo modelado.

Neste capítulo se apresenta um estudo de caso utilizando-se a ferramenta de modelagem e gestão de fluxo de processos denominado Bonita BPM (BONITASOFT, 2017). A ferramenta é utilizada para que sejam aplicados padrões de modelagem de processos do modelo *workflow*, apresentados no capítulo 4.

Além da ferramenta Bonita BPM, outras duas ferramentas foram analisadas na possibilidade de serem utilizadas, são elas a CPN *Tools* (CPN GROUP, 2017) e a jBPM (RED HAT, 2017).

A CPN *Tools* é uma ferramenta que edita, simula e analisa Redes de Petri Coloridas (*Coloured Petri Net*). Por ser uma ferramenta que permite a modelagem de processos por meio de um grafo, então se pode aplicar o modelo *workflow* nos processos modelados (CPN SAMPLES, 2017).

Apesar de ser uma ferramenta que oferece muitas possibilidades de uso, a CPN *Tools* apresenta alto grau de dificuldade para utilização, bem como utiliza-se uma linguagem que não é específica para modelagem de processos. Dentre seus pontos negativos, destaca-se o fato de que a ferramenta não apresenta mecanismos de acompanhamento do fluxo de execução do processo.

A jBPM é uma ferramenta desenvolvida utilizando a linguagem de programação Java. A ferramenta pode ser utilizada para modelar e executar processos de negócio. A jBPM não foi utilizado pela dificuldade em utilizar o software. A jBPM apresentou dificuldades durante sua instalação, foi necessário a instalação de softwares auxiliares já que esta funciona como uma extensão de uma ferramenta de desenvolvimento de software, ou seja, é necessário instalar um software que tem como objetivo a codificação de programas de computador. Então, usuários que não sejam desenvolvedores de software podem ter dificuldade ao modelar os processos utilizando esta ferramenta. Outra dificuldade encontrada foi a utilização, a interface mostrou-se pouco intuitiva e com grande dificuldade na modelagem dos processos.

5.1 Bonita BPM

A Bonita BPM é a ferramenta que foi escolhida para aplicar os padrões de modelagem de processos apresentados no capítulo 4. Trata-se de uma ferramenta que permite a modelagem de processos na notação BPMN (*Business Process Model and Notation*) criada em 2001. Atualmente a ferramenta utiliza a versão 2.0 do BPMN, e encontra-se na versão 7.2. Outra função da Bonita BPM é que oferece a

possibilidade de acompanhar o fluxo de execução do processo na ordem em que as tarefas são executadas, ou seja, assim que uma tarefa do processo é executada a próxima fica disponível para ser alocada pelo gestor do processo (BONITASOFT, 2017).

A Bonita BPM é intuitiva e de fácil instalação, ou seja, não é necessário de conhecimentos técnicos para a modelagem dos processos, que é realizada através de elementos gráficos. Uma característica importante da Bonita BPM, é que a ferramenta tem o código aberto, ou seja, pode ser modificada e adaptada de acordo com a necessidade.

O acompanhamento do fluxo de execução do processo se dá de forma igualmente intuitiva e no modelo *workflow*. Após a modelagem do processo e o preenchimento dos atores, a ferramenta apresenta um painel de controle, para que o gestor do processo possa fazer a alocação dos itens de trabalho.

Assim que o processo é modelado ou sofre alterações a aplicação que controla o fluxo do processo apresenta os resultados. Com isso a melhoria dos processos pode ser realizada com maior rapidez.

5.2 Estudo de Caso

O estudo de caso apresentado refere-se a uma empresa pequena que não tem o desenvolvimento de software como ramo de atuação principal. Trata-se de uma empresa que organiza e controla campanhas de marketing de incentivo, que doravante será denominada XMarketing.

Normalmente, os clientes da XMarketing querem oferecer um programa de recompensa para sua equipe de vendas, incentivando assim, as vendas da empresa contratante.

A XMarketing oferece como parte do serviço oferecido, um sistema para controle e interação com o programa de incentivo contratado. O processo de construção deste

sistema é analisado para o estudo de caso, ou seja, os demais processos que estão envolvidos na construção de um programa de incentivo não são abordados. Na figura 4 pode-se observar o processo de desenvolvimento do sistema de incentivo na XMarketing. O processo foi modelado utilizando-se a ferramenta Bonita BPM.

Na modelagem do processo destacado, representado na figura 4, foram utilizados cinco papéis no processo de desenvolvimento do sistema de incentivo. Como pode ser observado, cada função ficou representada nas raias do diagrama BPMN. Os papéis são os seguintes:

1. Atendimento: os profissionais de atendimento são responsáveis pela comunicação com o cliente e com a equipe de desenvolvimento de modo que as duas partes possam se entender. O atendente também fica responsável de gerenciar o programa de incentivo junto ao cliente, bem como criar demandas junto à equipe de desenvolvimento. Na XMarketing, normalmente um profissional de atendimento é alocado para cada programa de incentivo;
2. Design gráfico: compete ao profissional da equipe de design gráfico fazer a logomarca da campanha de incentivo, bem como a identidade visual da campanha e do sistema desenvolvido. A XMarketing conta com dois profissionais na equipe de design gráfico;
3. Desenvolvedor *Frontend*: na XMarketing apenas um profissional é responsável pelo desenvolvimento *frontend*. Este profissional é responsável por transformar a identidade visual, produzida pela equipe de design gráfica, em formato *web*, para que a equipe de desenvolvimento *backend* possa desenvolver o sistema;
4. Desenvolvedor de Banco de Dados: o profissional que faz o desenvolvimento de banco de dados recebe os requisitos e regras junto ao atendimento da campanha e então modela o banco de dados do sistema utilizado. Nesta função um profissional fica alocado;
5. Desenvolvedor *Backend*: um desenvolvedor *backend* fica disponível na XMarketing. Este profissional desenvolve em linguagem de programação, com tecnologia *web*, todas as regras e requisitos da campanha de incentivo solicitada.

Neste trabalho não será analisado se o processo da XMarketing é eficiente ou pode ser melhorado, ou seja, o foco é utilizar a ferramenta como forma de gestão de recursos humanos. A seguir é analisado o que foi observado.

5.3 Resultados Obtidos

Todos os padrões que foram apresentados no capítulo 4 foram analisados na ferramenta Bonita BPM. A ferramenta não oferece suporte a todos os padrões, mas os que puderam ser usados foram aplicados ao processo apresentado na figura 4, já os padrões que não são suportados foram destacados na análise feita nesta seção.

Quando foram apresentados no capítulo 4, os padrões se dividiram em duas categorias: padrões que são aplicados durante a modelagem do processo e padrões que são aplicados durante a execução do processo. Desta forma, também, foi dividida a avaliação da Bonita BPM.

5.3.1 Padrões em tempo de modelagem

Nesta seção são analisados padrões que são aplicados durante a modelagem do processo, ou seja, o *designer* do processo utiliza os padrões apresentados para modelar o processo de modo que os elementos de recursos humanos sejam representados.

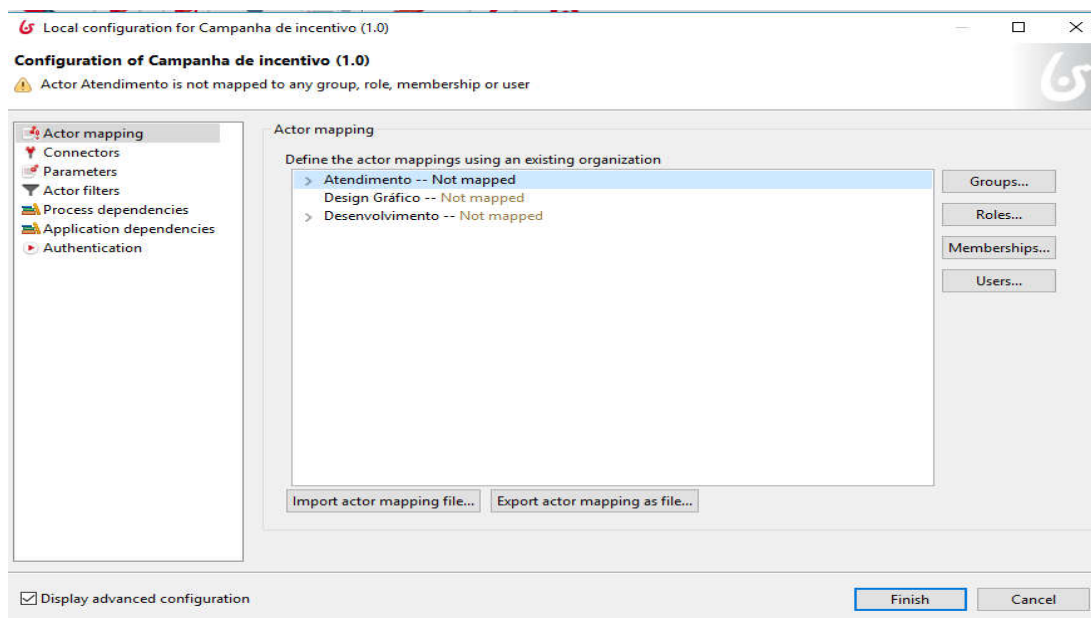
Na figura 5 é apresentada uma interface onde os atores, papéis e grupos podem ser alocados para cada função do processo, ou seja, é possível alocar uma ou mais pessoas para cada função representada nas raias do processo, na figura 4.

A Bonita BPM apresenta uma interface de usuário na língua inglesa, então as funções e mensagens apresentados nas imagens serão em inglês também. Pode-se observar na figura 5, o menu na lateral direita, há os seguintes botões:

- *Groups* (grupos): permite fazer a alocação das raias do processo por grupo de profissionais.
- *Roles* (papéis/funções): faz a seleção de um ou mais papéis da organização para cada raia do processo.
- *Memberships* (associações): é possível fazer a combinação de papéis e grupos, de modo que pode ser alocado um ou mais papéis por grupo.

- *Users* (usuários): a alocação na raia do processo pode ser realizada de forma individual também, cada ator fica responsável por uma raia do processo.

Figura 5 – Alocação de recursos humanos às raias do processo BPMN.



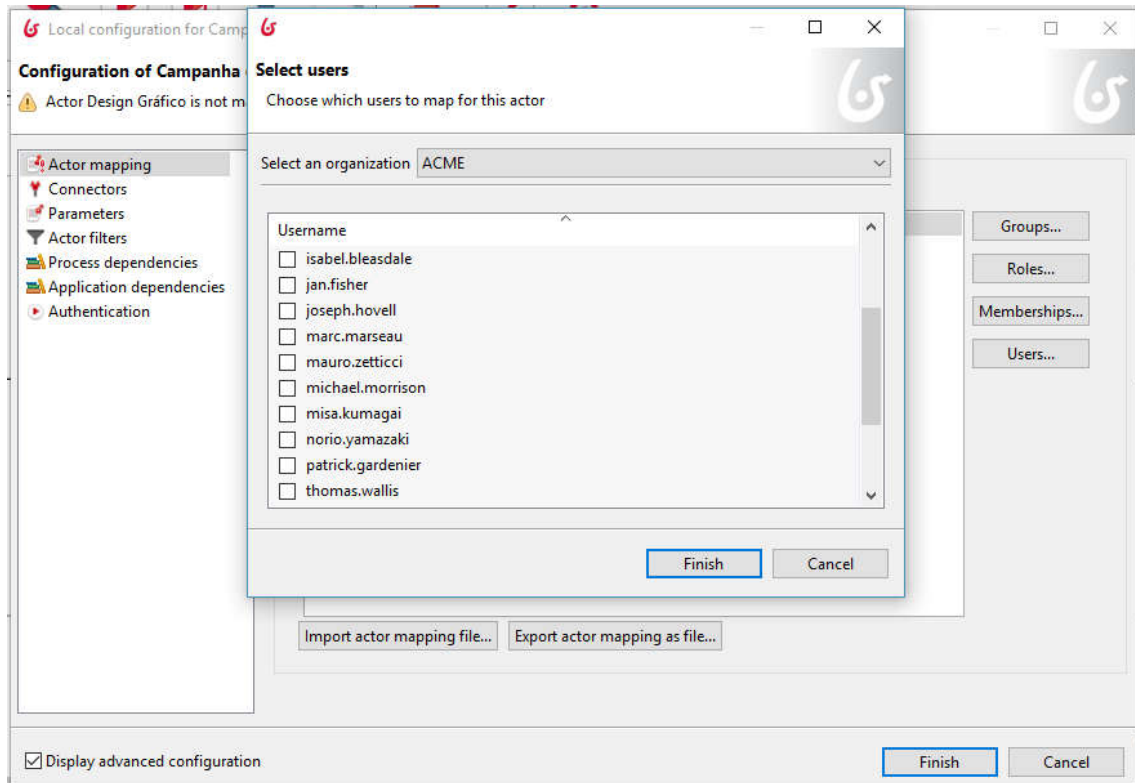
Fonte: próprio autor.

Conforme apresentado foi possível separar as tarefas por papel e de forma individual conforme apresentado na figura 6, onde pode ser observado que a Bonita BPM tem como funcionalidade atribuir em cada raia um processo um recurso humano.

De acordo com o que foi apresentado, os padrões que possuem suporte na Bonita BPM são os seguintes:

- *Pattern R-DA Direct Allocation* (alocação direta);
- *Pattern R-RBA Role-Based Allocation* (alocação baseada no papel/função).

Figura 6 – Seleciona um usuário específico para cada raia.



Fonte: próprio autor.

Alguns padrões não encontraram suporte na Bonita BPM, pois não foram encontradas funcionalidades para que possam ser aplicados, são os seguintes:

- Pattern R-RF Retain Familiar (alocação por familiaridade);
- Pattern R-HBA History-based Allocation (alocação baseado no histórico).
- *Pattern R-CBA Capability-based Allocation* (alocação baseada na habilidade)

Pode-se notar que a Bonita BPM consegue suportar padrões básicos como os padrões de alocação direta e alocação por papel, porém alguns padrões que exigem mapeamento de habilidades ou do histórico de execuções anteriores do processo não são suportados.

5.3.2 Padrões durante a execução do processo

Os padrões apresentados anteriormente são padrões que devem ser observados durante a modelagem do processo. Os padrões analisados a seguir visam a gestão de recursos durante a execução do processo.

Após a modelagem do processo é possível executá-lo. A Bonita BPM permite que seja acompanhado o fluxo do processo tanto pelo gestor quando pelos usuários do processo.

A figura 7 apresenta uma lista de processos em execução, de modo que o gestor pode acompanhar as estatísticas de cada um, além de gerenciá-los. Na figura 8, é apresentada a lista de tarefas para ser alocada do processo da figura 4, como pode-se notar só fica disponível a tarefa de “coleta com o cliente os requisitos da campanha”, é a primeira tarefa do processo da XMarketing, como o processo não prevê a execução de nenhuma outra tarefa em paralelo, apenas após essa estar completa é que a próxima tarefa fica disponível para alocação.

Figura 7 – Lista de processos para acompanhamento do gestor.

The screenshot displays the Bonita BPM interface. At the top, there is a 'DISABLE' button. Below it is a table with columns for 'Name', 'Version', and 'Installed on'. The table lists three processes: 'Campanha' (checked), 'Campanha de incentivo', and 'Pool'. To the right of the table, there is a detailed view for 'Campanha (1.0)', including its description, installation details, statistics, and configuration.

Name	Version	Installed on
☒ Campanha No description.	1.0 ✓	12/12/2016
☐ Campanha de incentivo No description.	1.0 ✓	12/12/2016
☐ Pool No description.	1.0 ⚠	12/11/2016

1 - 3 of 3

Campanha (1.0)
No description.
Installed on: 12/12/2016 7:32 PM
Installed by: System

Statistics
Number of cases: 0 ongoing / 0 archived
Tasks in failed status: 0

Configuration
Connectors: Resolved
Actors: Resolved

Fonte: próprio autor.

O usuário que recebe as tarefas alocadas consegue visualizá-las e alimentar as informações sobre a mesma, como comentários e a conclusão. A figura 9 mostra a lista de tarefas de um usuário.

Figura 8 – Lista de tarefas para serem alocadas.



Fonte: próprio autor.

O gestor do processo, conforme apresentado na figura 10, consegue alocar uma determinada tarefa a um recurso humano da organização. Consegue também desfazer uma alocação, como apresentado na figura 11, onde após a seleção da tarefa os botões ASSING e UNASSIGN, ficam disponíveis para alocar e retirar a alocação, respectivamente, uma tarefa a um recurso humano.

Conforme o que foi abordado anteriormente, pode-se concluir que os padrões que são suportados durante a execução do processo são os seguintes:

- Pattern R-DBAS Distribution by Allocation Single Resource (distribuição por alocação de um recurso);
- Pattern R-E Escalation (realocação).

Figura 9 – Lista de tarefas de um usuário.

Task list 		
		1 - 1 / 1 
 Task name ^	Process name	Due date
 Coleta com o cliente os requisitos da campanha	Campanha de incentivo	Dec 12 10:05 PM
		1 - 1 / 1

Fonte: próprio autor.

Figura 10 – Alocando uma tarefa a um recurso humano.

Select user to assign task

Select a user*

US

Usuario1 Silva

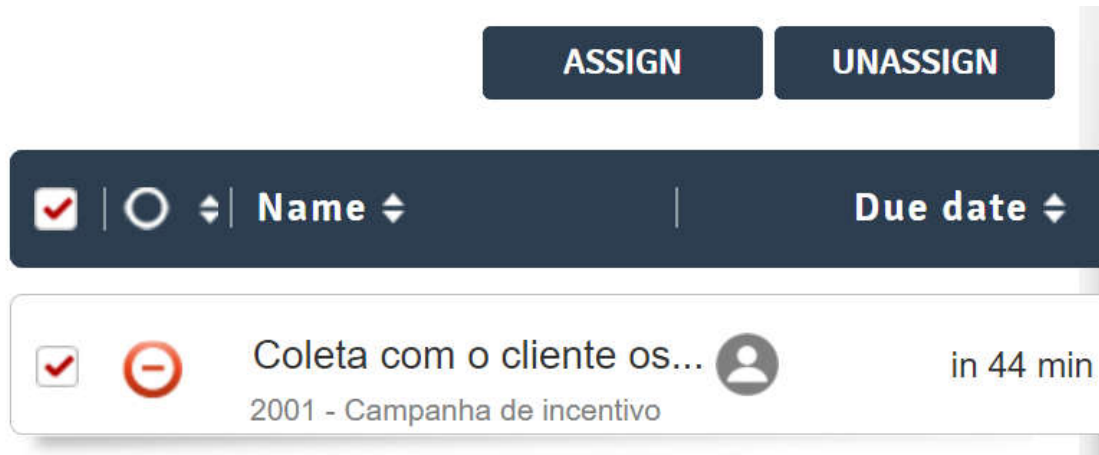
Usuario2 Silva

Usuario3 Silva

Usuario4 Silva

Fonte: próprio autor.

Figura 11 – A tarefa pode ser alocada ou desalocada.



Fonte: próprio autor.

A Bonita BPM não oferece suporte a recursos que tem como princípio o oferecimento de itens de trabalho, ou seja, quando uma tarefa é alocada a um recurso humano o mesmo não tem a opção de não assumir a responsabilidade de realizar a tarefa. Um recurso não tem a opção de se alocar para uma tarefa também. Com isso, os padrões que não possuem suporte na Bonita BPM são os seguintes:

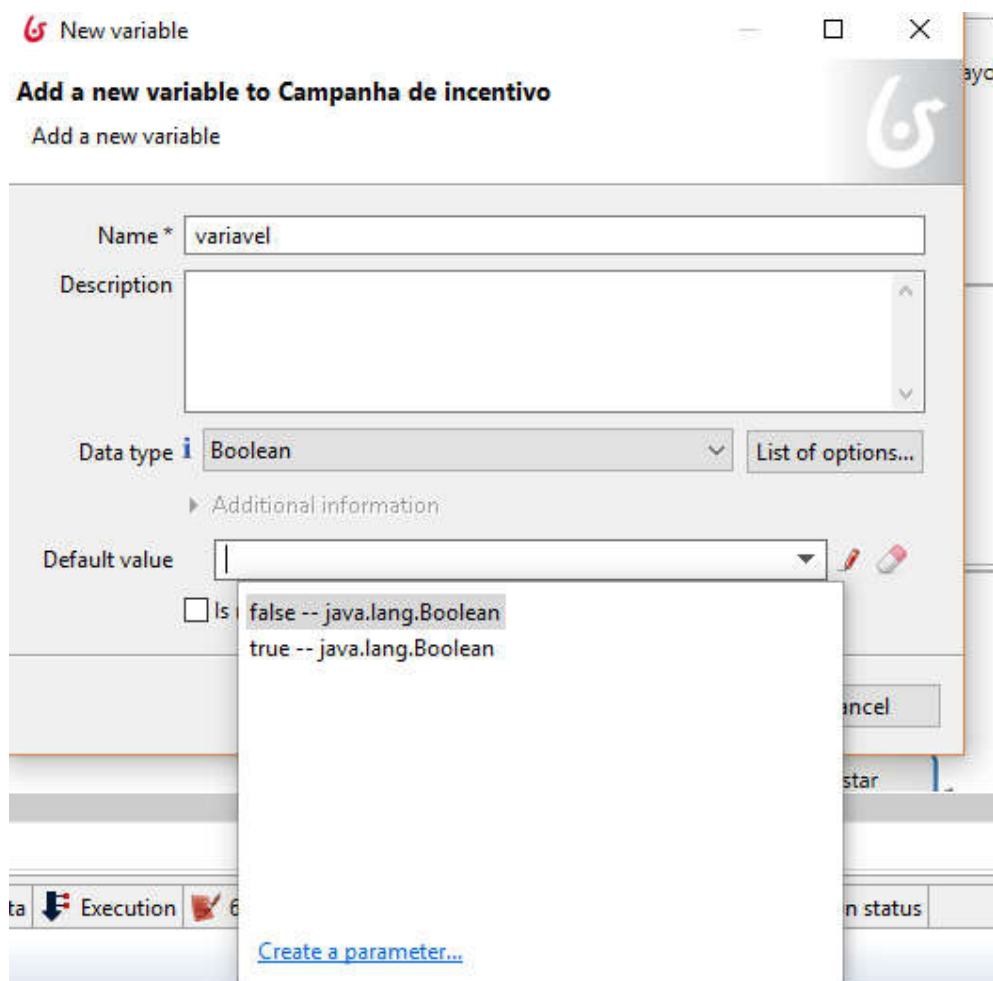
- Pattern R-RRR Round Robin Allocation (alocação por rodízio);
- *R-SHQ Shortest Queue* (fila mais curta).
- *Pattern R-DBOS Distribution by Offer – Single Resource* (distribuição por oferta a um recurso);
- *R-DBOM Distribution by Offer – Multiple Resources* (distribuição por oferta a um grupo de recursos);
- *R-RIA Resource-Initiated Allocation* (alocação iniciada por recursos).

5.3.3 Bonita BPM além dos recursos

Além de todas as funcionalidades apresentadas, a Bonita PBM pode fazer simulações para que o gestor entenda como o processo pode se comportar em determinadas situações.

O processo apresentado na figura 4 foi simulado para apresentação neste trabalho. Na figura 12 pode-se observar a criação de variáveis na Bonita BPM. As variáveis podem ser atribuídas para pontos de decisão como “a aprovação do sistema feita pelo cliente”.

Figura 12 – Como criar variáveis para simulação do processo.



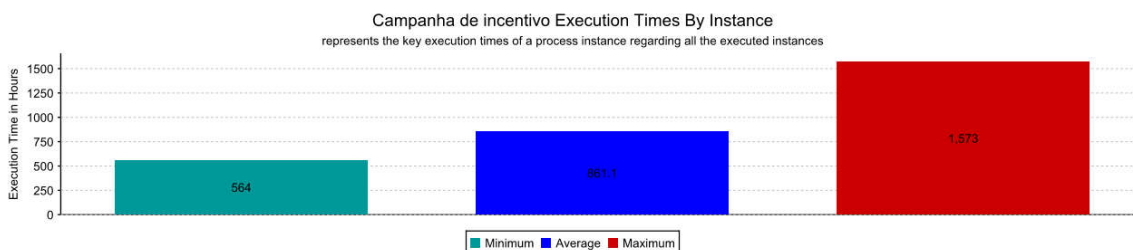
Fonte: próprio autor.

Existe uma variável booleana em cada ponto de decisão com cinquenta por cento da variável ser verdadeira, caso o cliente aprove o sistema desenvolvido, e a mesma porcentagem de ser falsa. Com isso algumas tarefas podem ser executadas mais de uma vez, fazendo com o sistema gaste mais tempo para ser desenvolvido.

Foram feitas estimativas de tempo para cada tarefa do processo, porém há a estimativa de que a execução do processo deve ser feita em 720 horas, ou seja, o sistema que é o produto do processo apresentado na figura 4, deve ser desenvolvido em 720 horas. Com isso pode-se fazer uma simulação para analisar as possíveis estimativas feitas pela Bonita BPM, de acordo com as variáveis definidas.

Na figura 13 pode-se observar o resultado da simulação. Em cada ponto e decisão do processo foram definidas variáveis com cinquenta por cento de chance de a resposta ser 'sim' ou 'não'. Foram feitas cem repetições, ou seja, o processo foi executado 100 vezes. A partir dessas execuções foram extraídos os tempos de execução de cada instância do processo.

Figura 13 – Resultado da simulação do processo.



Fonte: próprio autor.

Conforme, observado o tempo mínimo que o processo levou para ser executado foi de 564 horas, a média foi de 861,1 horas, já o máximo foi de 1573 horas. Com isso pode-se observar que a média é maior do que a estimativa, então de acordo com a simulação feita, é provável que o projeto atrase.

Com o passar do tempo e com as execuções do processo, o gestor do processo pode ir mudando as porcentagens das variáveis, ou até criando outras, para que a simulação possa ser mais precisa. Com isso uma estimativa de tempo mais aproximada da realidade pode ser dada aos clientes da XMarketing.

5.4 Considerações finais

A Bonita BPM mostrou-se uma ferramenta intuitiva e fácil de se usar, possui funcionalidades de modelagem de processos e gestão do fluxo de execução de processos. Com a aplicação dos padrões apresentados no capítulo 4, pode-se fazer a gestão de recursos humanos de forma mais eficiente. A tabela 2 mostra todos os padrões avaliados e quais possuem suporte pela ferramenta Bonita BPM.

Tabela 2 – Resultado da avaliação da Bonita BPM.

Padrão	Suportado pela Bonita BPM
<i>Pattern R-DA Direct Allocation</i> (alocação direta)	Sim
<i>Pattern R-RBA Role-Based Allocation</i> (alocação baseada no papel/função)	Sim
<i>Pattern R-CBA Capability-based Allocation</i> (alocação baseada na habilidade)	Não
<i>Pattern R-RF Retain Familiar</i> (alocação por familiaridade)	Não
<i>Pattern R-HBA History-based Allocation</i> (alocação baseado no histórico)	Não
<i>Pattern R-DBAS Distribution by Allocation Single Resource</i> (distribuição por alocação de um recurso)	Sim
<i>Pattern R-E Escalation</i> (realocação)	Sim
<i>Pattern R-RRA Round Robin Allocation</i> (alocação por rodízio)	Não
<i>R-SHQ Shortest Queue</i> (fila mais curta)	Não
<i>Pattern R-DBOS Distribution by Offer – Single Resource</i> (distribuição por oferta a	Não

um recurso)	
<i>R-DBOM Distribution by Offer – Multiple Resources</i> (distribuição por oferta a um grupo de recursos)	Não
<i>R-RIA Resource-Initiated Allocation</i> (alocação iniciada por recursos)	Não

Fonte: próprio autor.

Apesar de alguns padrões não serem suportados pela ferramenta Bonita BPM, conforme apresentado na tabela 2, a ferramenta tem o código aberto, ou seja, pode ser modificada para atender às necessidades da organização que a utiliza.

6. CONSIDERAÇÕES FINAIS

Os conceitos de recursos humanos são apresentados, principalmente no ambiente de desenvolvimento de software, onde pode-se perceber que os recursos humanos são importantes e devem ser geridos com eficiência, já que podem impactar na qualidade e no sucesso do software produzido.

O conceito de processos de software foi definido, ou seja, como a produção de software é realizada por organizações de software. Um processo de software pode ser modelado com uma linguagem de modelagem de processos. Cada processo tem diversos níveis de abstrações e focos, dependendo dos objetivos ao modelá-lo. Para diminuir os impactos da implantação ou melhoria de um processo, pode-se utilizar simulação de processos, para que possam ser previstos os comportamentos do processo em diversas situações.

Os padrões do modelo *workflow* podem ser uma possibilidade de modelar processos de software de modo que os elementos de recursos humanos sejam expressos no processo. Com isso a gestão de recursos pode ser beneficiada, já que os padrões tratam da alocação de tarefas do processo.

Mesmo que os processos estejam modelados de acordos com os padrões do modelo *workflow*, pode-se observar que outros elementos podem ser modelados, com a finalidade de desempenhar com maior eficiência uma tarefa, como as características pessoais dos recursos humanos. Uma premissa para a alocação de recursos humanos de modo eficiente pode ser a de que as pessoas dentro de uma organização desenvolvedora de software ocupem os papéis de acordo com suas características pessoais e habilidades.

Os padrões apresentados podem ser utilizados para modelar e gerir um processo e os recursos humanos. Apesar da ferramenta Bonita BPM não dar suporte a todos os padrões do modelo *workflow*, ela mostrou-se ser uma ferramenta de fácil instalação e utilização. Uma vantagem desta ferramenta escolhida é que pode-se melhorá-la e

adaptá-la de modo que possa se adequar as necessidades da organização que a utiliza, já que se trata de uma ferramenta de código aberto.

6.1 Contribuições do Trabalho

Os padrões de modelagem de processos do modelo *workflow* podem ser uma maneira de que a modelagem de processo se torne uma ferramenta importante na gestão de recursos humanos no ambiente de desenvolvimento de software. Os padrões estabelecem uma forma de representar os recursos humanos no processo, além de definir possibilidades de alocação de tarefas enquanto o processo está em execução.

Pode-se perceber que este trabalho cumpre o objetivo de utilizar padrões de modelagem de processos do modelo *workflow* para analisar os impactos na gestão de recursos humanos em processos de software, por meio de análise conceitual.

Aplicar os padrões permite ao gestor da execução do processo controle os recursos humanos durante a execução do processo, por meio dos elementos que representam as restrições e limitações dos recursos humanos que foram aplicados ao processo. Contudo enquanto o processo está sendo executado pode-se perceber que há padrões, também, que estabelecem uma forma de gerir os recursos humanos.

O objetivo de aplicar os padrões de modelagem de processo também foi cumprido, já que foi avaliada e testada a ferramenta Bonita BPM para aplicar em um processo os padrões apresentados no capítulo 4. Pode-se perceber também que a escolha da ferramenta deve ser feita de acordo com a adequação ao modelo *workflow*, ou seja, a ferramenta deve permitir a modelagem de processos na forma de um grafo direcional, além de permitir o acompanhamento do controle de fluxo de execução do processo.

6.2 Trabalhos Futuros

Como sugestão de trabalho futuro, pode-se destacar a investigação de como as características pessoais dos recursos humanos de uma organização, se modeladas no processo, podem afetar a gestão de recursos humanos e a qualidade do software produzido.

Pode-se propor como trabalho futuro a implementação dos padrões de modelagem de processos do modelo *workflow* em um processo que seja aplicado e utilizado em uma organização desenvolvedora de software, de modo que possa verificar os possíveis efeitos na gestão de recursos humanos.

Outro trabalho futuro possível a partir deste trabalho de pesquisa refere-se ao desenvolvimento de uma ferramenta de modelagem e gestão de processos que tenha como foco a utilização dos padrões de modelagem de processos do modelo *workflow*.

7. REFERÊNCIAS

ACUÑA, S. T.; GÓMEZ, M. & JURISTO, N. How do personality, team processes and task characteristics relate to job satisfaction and software quality?. **Information and Software Technology**. v. 51. 2009. p. 627-639.

ACUÑA, S. T. & JURISTO, N. Assigning people to roles in software projects. **Software: Practice and Experience**. v. 34. 2004. p. 675-696.

ANDRÉ, M.; BALDOQUÍN, M. G. & ACUÑA, S. T. Formal model for assigning human resources to teams in software projects. **Information and Software Technology**. v. 53. 2011. p. 259-275.

BENDRAOU, R.; JEZEQUEL, J. M.; GERVAIS, M. P. & Blanc, X. A comparison of six uml-based languages for software process modeling. **IEEE Transactions on Software Engineering**. v. 36. 2010. p. 662-675.

BONITASOFT. Bonita BPM 7.2. Disponível em: <<http://www.bonitasoft.com/>>. Acesso em 12 jan. 2017.

BUSINESS PROCESS MODEL NOTATION. **Notation (BPMN) version 2.0**. OMG Specification, Object Management Group. 2011.

CESTARI FILHO, Felício. **ITIL V3 Fundamentos**. Rio de Janeiro: RNP/ESR. 2011.

CPN GROUP. CPN Tools Homepage . Disponível em: <<http://cpntools.org/>>. Acesso em 15 jan. 2017.

CPN SAMPLES. CPN Tools, CPN Samples. Disponível em: <http://di.uern.br/cpnsamples/?page_id=4>. Acesso em 10 fev. 2017.

CURTIS, B.; HEFLEY, B. & MILLER, S. **People capability maturity model (P-CMM) version 2.0**. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst. 2009.

CURTIS, B.; KELLNER, M. I. & OVER, J. Process modeling. **Communications of the ACM**. v. 35. 1992. p. 75-90.

DI NITTO, E.; LAVAZZA, L.; SCHIAVONI, M.; TRACANELLA, E. & TROMBETTA, M. Deriving executable process descriptions from UML. In: **Proceedings of the 24th International Conference on Software Engineering**. ACM. 2002. p. 155-165.

DUMAS, M. & PFAHL, D. Modeling Software Processes Using BPMN: When and When Not?. In: **Managing Software Process Evolution**. Springer International Publishing. 2016. p. 165-183.

HUMPHREY, W. S. **Managing technical people: innovation, teamwork, and the software process**. Addison-Wesley Longman Publishing Co. 1996.

KELLNER, M. I.; MADACHY, R. J. & RAFFO, D. M. Software process simulation modeling: Why? what? how?. **Journal of Systems and Software**. v. 46. 1999. p. 91-105.

LONCHAMP, J. A structured conceptual and terminological framework for software process engineering. In: **Software Process, 1993. Continuous Software Process Improvement, Second International Conference on the**. IEEE. 1993. p. 41-53.

LORSZ, M. The software process: Modeling, evaluation and improvement. **Handbook of Software Engineering and Knowledge Engineering**. v. 1. 2001. p. 193.

MOORE, J. E. Personality characteristics of information systems professionals. In: **Proceedings of the 1991 conference on SIGCPR**. ACM. 1991. p. 140-155.

OTERO, C. E.; OTERO, L. D.; WEISSBERGER, I. & QURESHI, A. A multi-criteria decision making approach for resource allocation in software engineering. In: **Computer Modelling and Simulation (UKSim)**, 2010 12th International Conference on. IEEE. 2010. p. 137-141.

PFAHL, D. & LEBSANFT, K. Integration of system dynamics modelling with descriptive process modelling and goal-oriented measurement. **Journal of Systems and software**. v. 46. 1999. p. 135-150.

PROJECT MANAGEMENT INSTITUTE. **Um guia do conjunto de conhecimentos em gerenciamento de projetos (Guia PMBOK)**. Pennsylvania: Project Management Institute, Inc. 2004.

POWELL, A.; MANDER, K. & BROWN, D. Strategies for lifecycle concurrency and iteration—A system dynamics approach. **Journal of Systems and Software**. v. 46. 1999. p. 151-161.

PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. 7ª Edição. Porto Alegre : AMGH. 2011.

RED HAT. jBPM - Open Source Business Process Management - Process engine. Disponível em: <<http://www.jbpm.org/>>. Acesso em 20 jan. 2017.

RUS, I. ; COLLOFELLO, J. & LAKEY, P. Software process simulation for reliability management. **Journal of Systems and Software**. v. 46. 1999. p. 173-182.

RUSSELL, N.; TER HOFSTEDE, A. H.; EDMOND, D. & VAN DER AALST, W. M. **Workflow resource patterns** (BETA Working Paper Series, WP 127). The Netherlands: University of Technology, Eindhoven. 2004.

RUSSELL, N.; VAN DER AALST, W. M.; TER HOSPEDE, A. H. & EDMOND, D. Workflow resource patterns: Identification, representation and tool support. In: **International Conference on Advanced Information Systems Engineering**. Springer Berlin Heidelberg. 2005. p. 216-232.

SANTOS, T. J. A.; LIMA, A. M.; REIS, C. A. L.; & REIS, R. Q. Automated support for human resource allocation in software process by cluster analysis. In: **Proceedings of the 4th International Workshop on Recommendation Systems for Software Engineering**. ACM, 2014. p. 30-31.

SLOMP, J. & MOLLEMAN, E. Cross-training policies and team performance. **International Journal of Production Research**. v. 40. 2002. p. 1193-1219.

SOMMERVILLE, I. ; RODDEN, T. Human, social and organisational influences on the software process. **Software Process**. v. 4. 1996. p. 89-100.

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson Addison-Wesley. 8ª edição. 2007.

TURLEY, R. T. & BIEMAN, J. M. Competencies of exceptional and nonexceptional software engineers. **Journal of Systems and Software**. v. 28. 1995. p. 19-38.

WANG, Q. ; XIAO, J.; LI, M.; NISAR, M. W.; YUAN, R.; & ZHANG, L. A process-agent construction method for software process modeling in SoftPM. In: **Software Process Workshop**. Springer Berlin Heidelberg. 2006. p. 204-213.

WHITE, S. A. Introduction to BPMN. **IBM Cooperation**. v. 2. 2004.

WILLIFORD, J. & CHANG, A. Modeling the FedEx IT division: a system dynamics approach to strategic IT planning. **Journal of Systems and Software**. v. 46. 1999. p. 203-211.

WORKFLOW PATTERNS INITIATIVE. Workflow Patterns home page. Disponível em: <<http://www.workflowpatterns.com/>>. Acesso em 10 jan. 2017.

WYNEKOOP, J. L. & WALZ, D. B. Investigating traits of top performing software developers. **Information Technology & People**. v. 13. 2000. p. 186-195.

ZAMLI, K. Z. & LEE, P. A. Taxonomy of process modeling languages. In: **Computer Systems and Applications, ACS/IEEE International Conference on**. 2001. IEEE. 2001. p. 435-437.