

# Gerência de Projetos e Processos de Desenvolvimento de Software: uma proposta de integração

Maurício Covolan Rosito<sup>1</sup>, Daniel Antonio Callegari<sup>1</sup>, Ricardo Melo Bastos<sup>1</sup>

<sup>1</sup> Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul  
Avenida Ipiranga, 6681 – 90619-900 – Porto Alegre – RS – Brasil  
{mrosito, danielc, bastos}@inf.pucrs.br

**Abstract.** *Software organizations are constantly looking for better solutions when designing and using well-defined software processes for the development of their products and services. However, many existing software development processes lack for more project management skills in their models. This research proposes the definition of a model that integrates the concepts of project management and those available on the software development processes, helping not only process integration but also assisting managers in the decision making process during project planning. We developed a prototype in order to illustrate the proposed concepts and we performed a qualitative exploratory evaluation of this prototype with project managers from nine companies.*

**Resumo.** *As organizações de software constantemente buscam por processos de software bem definidos para o desenvolvimento de seus produtos e serviços. Entretanto, muitos processos de desenvolvimento de software existentes apresentam carências no quesito de gerência de projetos. Esta pesquisa propõe a definição de um modelo que integra os conceitos de gerência de projetos e de processos de desenvolvimento de software, contribuindo não somente para a integração destes processos, mas também auxiliando os gestores na tomada de decisões durante o planejamento de projetos. De forma complementar, desenvolveu-se um protótipo para demonstrar os conceitos propostos e realizou-se uma avaliação qualitativa deste protótipo com gerentes de projetos de nove empresas.*

## 1. Introdução

A crescente preocupação relativa ao desenvolvimento de software pode ser observada devido à adoção de práticas de engenharia de software pelas empresas [1]. O desenvolvimento de produtos de software requer o planejamento e a execução de atividades, definidas de acordo com o escopo do projeto, onde é necessário lidar tanto com assuntos gerenciais quanto técnicos. Em qualquer contexto deve-se considerar o fato de que projetos sempre são esforços únicos e temporários, além de envolverem um elevado nível de incerteza [2], [3].

A gerência de projetos (GP) em um ambiente de desenvolvimento de software é definida como a gerência das pessoas e de outros recursos por um gerente de projetos a fim de planejar, analisar, projetar, construir, testar e manter um sistema de informação [2], [3]. Para cumprir estes objetivos, um gerente de projetos precisa de algum tipo de suporte, geralmente baseado em uma metodologia de gerência de projetos, que permita

lidar com diferentes variáveis de projeto, responsabilidades e tarefas. Para este fim, existem diversas propostas na literatura ou práticas já realizadas nas empresas. Contudo, conforme observado em [3], a maioria dos modelos ou guias voltados para a gerência de projetos, tal como o Project Management Body of Knowledge Guide (PMBOK) [4], não se dirigem especificamente a processos de desenvolvimento de software.

Os processos de desenvolvimento de software (PDSs), embora identifiquem a importância das atividades relacionadas à gestão, não tratam de forma adequada os aspectos relacionados à gerência de projetos. Tal fato é reforçado em [3] e [5], onde é destacado que dois dos mais importantes PDSs, respectivamente o Rational Unified Process (RUP) [6], por sua absorção no mercado, e o Object-oriented Process, Environment and Notation (OPEN) [7], por sua contribuição no meio acadêmico, necessitam de maior suporte no quesito de gerência de projetos. Tanto o RUP quanto o OPEN auxiliam na execução das melhores práticas para o desenvolvimento de software. No entanto, ambos os modelos mostraram-se incompletos em áreas essenciais de conhecimento da gerência de projetos, por exemplo, a gerência de aquisição, de comunicação e de pessoas. Cabe lembrar, também, que os PDSs atuais em geral não abordam de maneira adequada o tratamento das questões relacionadas a recursos humanos e outros tipos de recursos, tais como equipamentos e materiais envolvidos [3].

A fim de se obter um processo mais detalhado para o gerenciamento de projetos software, é necessário aplicar os conhecimentos de gestão de projetos aos processos de desenvolvimento do software. Portanto, se por um lado o PMBOK Guide pode fornecer uma perspectiva gerencial da solução, a visão sobre a produção deve ser obtida a partir de um modelo de processo de desenvolvimento de software, tal como os processos RUP e OPEN.

De acordo com [8], o processo de desenvolvimento de software é um dos principais mecanismos responsáveis por gerenciar e controlar os projetos e produtos de software. Durante o planejamento de atividades, o gerente de projetos pode necessitar interagir com outros departamentos da organização a fim de obter informações relevantes para o projeto (contatar o setor de recursos humanos sobre a necessidade de contratação de pessoal, por exemplo). Neste instante, há uma dissociação entre o fluxo de atividades de um projeto de software e os demais fluxos de atividades de suporte ao projeto da organização (aqui denominados **Fluxos Empresariais**). Conforme ilustrado na figura 1, ambos os fluxos de trabalho são executados de forma distinta e podem influenciar nos prazos das atividades e custos dos projetos. Pode também haver uma relação de dependência entre as atividades pertencentes a estes dois tipos de fluxos de trabalho. Conseqüentemente, o gerente de projetos precisa de algum tipo de suporte que permita acompanhar o andamento das atividades destes fluxos de trabalho.

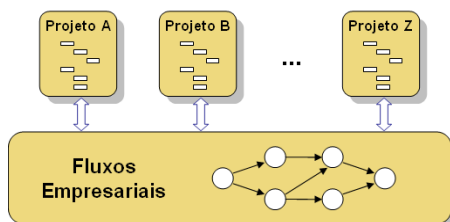


Figura 1. Relação entre os projetos e os Fluxos Empresariais

Este artigo apresenta uma proposta de modelo de integração entre conceitos de gerência de projetos e de processos de desenvolvimento de software, denominado **Software Planning Integrated Model (SPIM)**. Esse modelo inclui um conjunto de regras (Tabela 1) que permitem o planejamento integrado de atividades gerenciais e produtivas no contexto de desenvolvimento de software. No SPIM são desenvolvidos os conceitos advindos da integração do PMBOK com o RUP em extensão à proposta apresentada em [3]. Com base no SPIM, torna-se possível o desenvolvimento de ferramentas para o apoio ao processo de planejamento e acompanhamento das atividades gerenciais e produtivas para projetos de desenvolvimento de software. Neste sentido, foi desenvolvida a ferramenta denominada **Software Planning Integrated Tool (SPIT)** que atua como um *Add-in* para uma ferramenta comercial de gerenciamento de projetos já existente e muito utilizada.

Pesquisas anteriores apresentaram resultados interessantes, mas uma íntima integração da gerência de projetos e dos processos de software com resultados práticos ainda é uma questão em aberto [2], [5], [14], [15]. Conseqüentemente, faz-se necessário mais estudo para uma solução que permita um melhor nível de integração entre os conceitos e modelos para estas duas áreas de conhecimento.

Este artigo está organizado da seguinte forma: a Seção 2 discorre sobre o referencial teórico; a Seção 3 apresenta alguns trabalhos encontrados na literatura que estão relacionados à análise da gestão de projetos nos processos de desenvolvimento de software. Na Seção 4 são descritos os metamodelos base de integração oriundos da integração do PMBOK com o RUP e da integração do PMBOK com o OPEN. A partir da análise destes metamodelos é apresentado um modelo de integração entre a GP e o PDS na Seção 5. A Seção 6 contém os resultados da avaliação do modelo proposto em empresas de desenvolvimento de software. Finalmente, na Seção 7, são apresentadas as conclusões e as próximas etapas.

## **2. Referencial Teórico**

### **2.1. Project Management Body of Knowledge – PMBOK**

Reconhecido internacionalmente pelo seu esforço em definir normas e dar suporte aos profissionais de gerência de projetos, o Project Management Institute (PMI) [9] publicou um guia geral de gerência de projetos, o PMBOK Guide. O Project Management Body of Knowledge reúne as melhores práticas aplicáveis à maioria dos projetos e sobre as quais há um amplo consenso sobre o seu valor e utilidade.

De acordo com [4] e [9], o principal objetivo do PMBOK Guide é identificar um subconjunto dos conhecimentos sobre gerência de projetos que seja reconhecido, genericamente, como sendo uma coleção de boas práticas. O PMBOK, porém, não é um processo sem seu sentido estrito, pois não determina quais são as ações e nem indica como estas devem ser executadas para o correto desenvolvimento de um projeto. Além disso, o PMBOK é dito mais compatível com atividades industriais e, portanto, não aborda especificamente processos de desenvolvimento de software [3].

Para efeito da discussão proposta por este artigo, observa-se que os conceitos do PMBOK são, em sua maioria, representados por textos descritivos. Entretanto,

objetivando comparar dois modelos e, posteriormente, realizar a integração entre eles, deve-se representá-los sob estruturas compatíveis – neste artigo será utilizado o metamodelo previamente desenvolvido em [3], usando notação UML (figura 2).

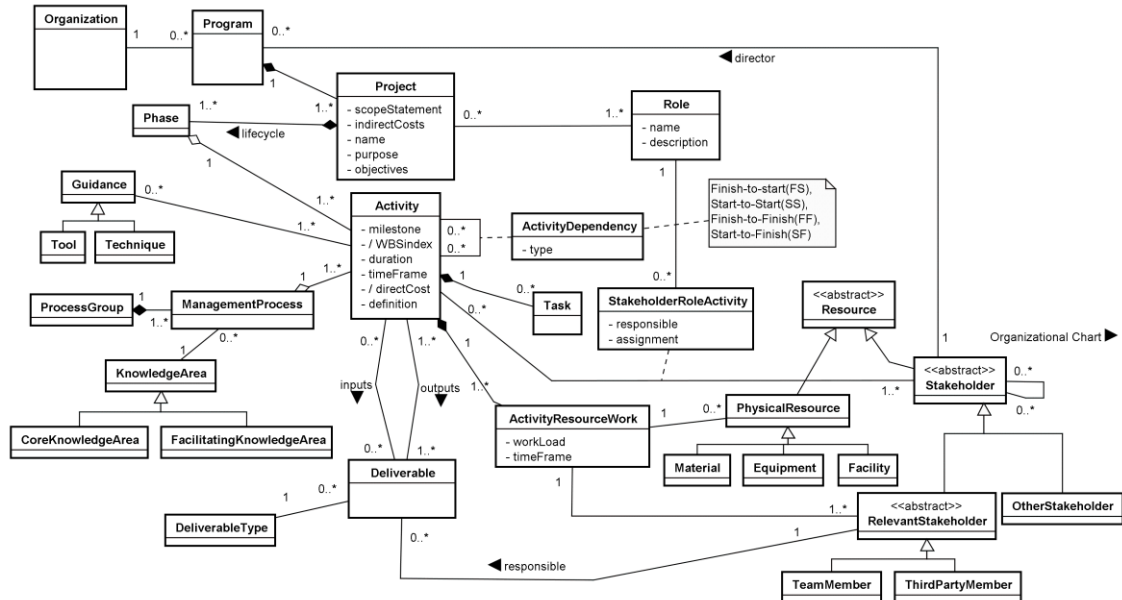


Figura 2. Metamodelo de gerência de projetos baseado no PMBOK Guide [3]

## 2.2. Rational Unified Process - RUP

O RUP é um processo iterativo de desenvolvimento de software desenvolvido pela empresa IBM Rational Software, originado a partir do metamodelo SPEM [10], [11]. O RUP atua como um *framework* que pode ser adaptado e estendido de acordo com as características do processo de desenvolvimento de software da organização [12]. Segundo [6], o RUP apresenta um modelo semântico, ilustrado na figura 3, contendo seus principais elementos e relacionamentos.

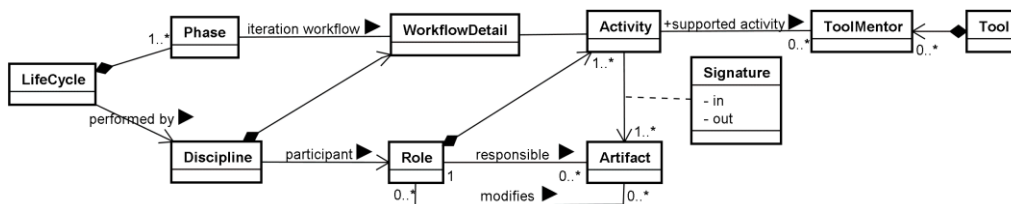


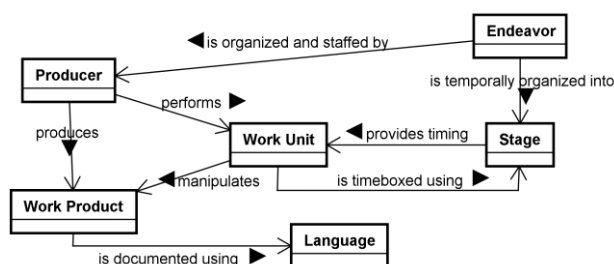
Figura 3. Modelo semântico do RUP [6]

Na figura 3, a classe **LifeCycle** representa o ciclo de vida de desenvolvimento de um software. Este conceito é particionado em um conjunto de quatro fases (classe **Phase**). A classe **Discipline** divide os elementos de processo em áreas de interesse. Um papel (classe **Role**) representa o elemento responsável por desempenhar atividades (**Activity**) para produzir ou modificar os artefatos (**Artifact**) do processo. As informações de como os papéis devem colaborar entre si através de suas atividades são definidos pela classe **Workflow Detail**. A classe **Artifact** descreve os tipos de produtos de trabalho que são produzidos ou consumidos no desempenho de atividades. Assim, a classe associativa **Signature** indica que um artefato é utilizado como entrada ou saída

de uma atividade. A classe **Tool** descreve as ferramentas que podem ser utilizadas auxiliando a produção ou modificação de um artefato. Finalmente, a classe **ToolMentor** descreve o uso de ferramentas no contexto de algumas atividades.

### 2.3. Object-oriented Process, Environment and Notation - OPEN

O OPEN é uma metodologia de desenvolvimento de software orientado a objetos mantido pelo OPEN Consortium Group [7], [13]. Ele pode ser definido como um framework (OPEN Process Framework, OPF) que fornece um metamodelo extensível e que pode ser configurado para PDSs distintos [7]. O OPF reconhece os elementos da figura 4 como sendo os componentes centrais em seu framework.



**Figura 4. Componentes centrais do OPEN Process Framework [13]**

Um produtor (classe **Producer**) é o elemento responsável por produzir ou modificar – direta ou indiretamente – os artefatos do processo. Os produtores podem ser ferramentas ou pessoas definidas através de papéis. A classe **Work Unit** consiste de um conjunto de operações coesas executadas pelo produtor no desenvolvimento de seu trabalho, e podem ser classificadas como tarefas, técnicas, fluxos de trabalho e atividades. Finalmente, a classe **Stage** determina as divisões de intervalos de tempo do processo, sendo dividida em estágios com duração (fases) e instantâneos (*milestones*).

### 3. Trabalhos Relacionados

Atualmente, alguns trabalhos vêm sendo desenvolvidos pela comunidade de engenharia de software salientando a importância de aplicar os conceitos de gestão de projetos nos processos de software. Nesse capítulo serão descritos os trabalhos propostos por [5], [14] e [15], os quais serviram como referência para esta pesquisa.

Em [5], dois importantes processos de desenvolvimento de software, respectivamente o RUP e o OPEN, são analisados a partir um ponto de vista da gestão de projetos. Os autores concluem que ambos os processos são deficientes em certas áreas de conhecimento padrão da gestão de projetos, como a gerência de aquisição, de comunicação e de pessoas aquisição de gestão, gestão da comunicação e gestão de pessoal.

Uma avaliação qualitativa dos elementos componentes do RUP e do OPEN foi realizada em [14]. Os autores centralizam sua comparação sobre os aspectos do processo e da arquitetura do metamodelo, conceitos e terminologia utilizada, e o suporte a gestão de projetos. Segundo os autores, o ciclo de vida do RUP, por exemplo, baseia-se em quatro fases, as quais são compostas por uma ou mais iterações. Cada iteração segue o padrão cascata (*waterfall*) contendo atividades básicas de especificação de

requisitos, análise, projeto, implementação, teste e implantação, que resultam em um *release* de um produto que evolui gradativamente a partir de cada iteração até se tornar o sistema final. Entretanto, os autores acreditam que a arquitetura do metamodelo do RUP não permite o apoio a um verdadeiro desenvolvimento iterativo. Em contrapartida, os autores defendem que o metamodelo do OPEN apresenta um melhor suporte para desenvolver um processo que atenda as necessidades específicas de um determinado domínio de aplicação ou para adaptar um processo a projetos específicos.

Em [15], quatro importantes metodologias/frameworks de gerência de projetos – Projects In Controlled Environments (PRINCE2) [16], RUP, Agile Development Methods e o Microsoft Solutions Framework (MSF) [23] – foram comparados com o PMBOK. No caso do PMBOK e do PRINCE2, suas documentações não informam como utilizar alguma das técnicas ou ferramentas descritas. Estas metodologias apenas estabelecem os processos e o conjunto de ferramentas e técnicas que podem ser adotadas em projetos. Além disso, metodologias como RUP, PRINCE2 e MSF não fornecem informações relacionadas a algumas das áreas de conhecimento de gestão de projetos. Os autores concluem que todas as metodologias/frameworks discutidas possuem ferramentas e procedimentos comuns. Eles também propõem uma abordagem que combina os conceitos destas metodologias objetivando obter os melhores resultados de cada abordagem.

Como apresentado acima, diferentes são as abordagens propostas para dar apoio gerencial aos projetos de desenvolvimento de software, cada qual com suas contribuições e limitações. Analisando o suporte oferecido pelos métodos descritos nesta seção, pode ser visto que, embora exista a preocupação acerca da carência de conceitos de gestão de projetos dos processos de desenvolvimento de software atuais, estes estudos não abordaram sobre como realizar planejamento de projetos considerando os elementos que compõem a GP e os processos de desenvolvimento de software.

#### **4. Metamodelos Base de Integração**

O estudo detalhado dos modelos do PMBOK e RUP (reforçado posteriormente com a inclusão do OPEN para seu refinamento) permitiu identificar como são organizados e quais as relações válidas entre os elementos de cada modelo. Através da integração entre a gerência de projetos e os processos de desenvolvimento de software foi possível identificar as principais características e discrepâncias entre os elementos de tais modelos. Conforme citado anteriormente, o metamodelo de referência do PMBOK [3] inclui os elementos necessários para a gerência de projetos, enquanto que os conceitos de processos de desenvolvimento de software são obtidos pelo RUP ou pelo OPEN.

A proposta de integração entre gerência de projetos e processos de desenvolvimento de software apresentada neste trabalho é constituída de três pacotes: um para os conceitos de GP, outro para os relacionados aos PDSs e, finalmente, um pacote comum (“*Common*”) que une os conceitos que ocorrem em ambos os modelos. O critério de integração entre os modelos seguiu o conjunto de regras identificado por [3], que afirma que ao se realizar uma integração entre dois modelos, as seguintes situações podem ocorrer:

- (a) Uma sobreposição de conceitos (duas classes com o mesmo conceito em cada

modelo): neste caso, pode-se transformar e unir estas duas classes em um único conceito dentro de um pacote comum;

(b) Uma Relação entre conceitos (uma classe de um dos modelos se relaciona com alguma outra classe de outro modelo, mas estas classes não representam exatamente o mesmo conceito): deve-se manter as classes em seus modelos originais e criar uma associação entre elas; e

(c) Conceitos independentes (classes com conceitos independentes e distintos): deve-se deixar cada classe em seu próprio pacote.

De acordo com critérios acima, foram desenvolvidos dois metamodelos base de integração durante esta pesquisa: PMBOK+RUP e PMBOK+OPEN (ver seções a seguir). Esses metamodelos apresentam uma estrutura de classes similar, substituindo apenas o pacote referente ao processo de desenvolvimento de software. Assim, tanto as classes dos pacotes PMBOK e *Common* como os relacionamentos entre as classes pertencentes a estes dois pacotes permanecem inalterados. Porém, como os dois processos de desenvolvimento de software possuem características particulares, estas diferenças são refletidas em classes distintas e em diferentes relacionamentos com os pacotes PMBOK e *Common*.

Cabe salientar que, apesar de o modelo SPIM implementar exclusivamente os conceitos advindos da integração do PMBOK com o RUP, foi também realizada a integração do PMBOK com o OPEN objetivando-se mostrar que a estrutura constituída de três pacotes (um para os conceitos de gerência de projetos, outro para os relacionados aos processos de desenvolvimento de software e um pacote comum que une os conceitos que ocorrem em ambos os modelos) funciona com diferentes processos de desenvolvimento de software. Dessa forma, a integração do PMBOK com o RUP e o OPEN permitiu tanto a confirmação quanto a adequação dos conceitos propostos no modelo final.

As seções a seguir apresentam uma discussão mais elaborada, que envolve o desenvolvimento dos dois metamodelos de integração elaborados nesta pesquisa.

#### **4.1. Metamodelo Integrado entre o PMBOK e o RUP**

De modo a explicar o metamodelo de integração PMBOK+RUP (figura 5), seus principais elementos serão descritos. Neste artigo, serão apresentadas especificamente as principais extensões realizadas para o desenvolvimento do SPIM ao metamodelo integrado entre o PMBOK e o RUP proposto em [3].

É importante ressaltar que alguns conceitos relacionados à gerência de projetos que estão contidos nos processos de desenvolvimento de software (neste caso, no RUP e no OPEN) foram propositadamente deslocados para o pacote de classes gerenciais (pacote PMBOK) com o objetivo de deixar mais explícita a classificação dos conceitos de GP e do processo de desenvolvimento de software. Também optou-se por manter os conceitos na língua inglesa para facilitar a comparação com os modelos originais.

Neste modelo, a classe **Organization** representa uma empresa que se organiza por programas (classe **Program**). Os programas são grupos de projetos (classe **Project**)

designados a alcançar um objetivo estratégico. As organizações geralmente dividem os projetos em várias fases (classe **Phase**) visando um melhor controle gerencial.

Os recursos necessários para um projeto são explicitamente descritos no sub-pacote **Resources**. Sendo assim, pessoas, equipamentos e locais são representados pela classe **Resource**. Estes recursos são divididos em recursos ativos (classe **Stakeholder**) e não-ativos (classe **PhysicalResource**). Stakeholders correspondem às pessoas e organizações cujos interesses são afetados pelo projeto [4]. A classe associativa **ProjectStakeholder** permite definir se um stakeholder é “chave” para o projeto [4]. O sub-pacote **Resources** também contém informações sobre a disponibilidade de cada recurso ao atribuí-lo às atividades, mesmo que realizado de forma manual ou automática, através da classe **ResourceAvailability** [3]. Esta classe permite automatizar os processos de alocação de recursos em projetos de software. Paralelamente, a definição da carga de trabalho (atributo *workload*) dos recursos físicos (ao associar-se a diferentes atividades) é representada pela classe **ActivityPhysicalResourceWork**.



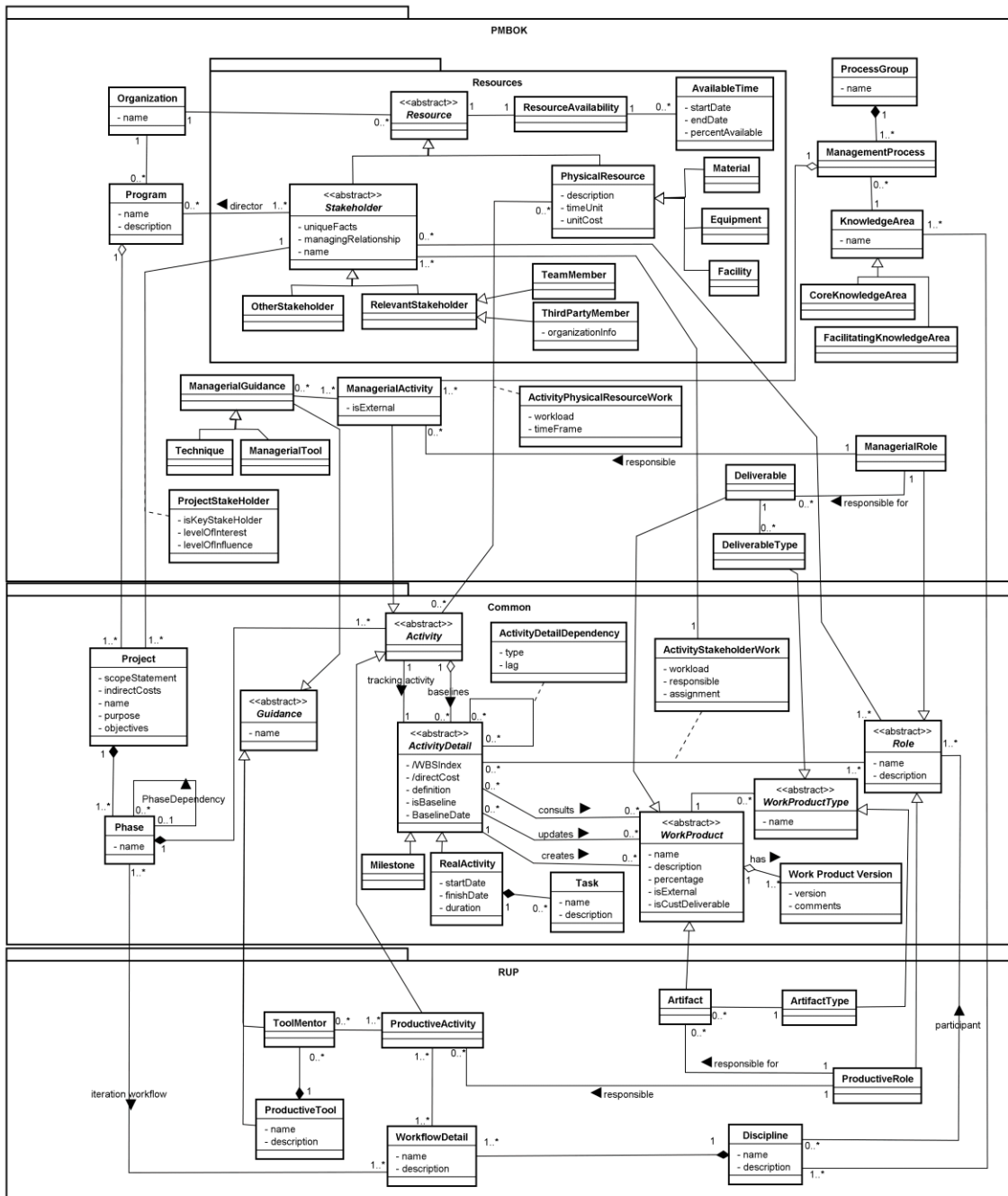


Figura 5. Metamodelo de integração PMBOK+RUP

Uma atividade (classe **Activity**) pode ser classificada como atividade produtiva (**ProductiveActivity**) ou atividade gerencial (**ManagerialActivity**). As atividades produtivas são diretamente relacionadas com a construção do software. Entretanto, as atividades gerenciais podem pertencer ao fluxo de desenvolvimento de software (*isExternal=false*) ou aos fluxos empresariais da organização (*isExternal=true*). Logo, identificam-se três tipos de atividades: **produtivas**, **gerenciais** e **gerenciais de apoio**. Cada atividade pode pertencer a um ou mais *baselines*. Em cada geração da *baseline*, uma atividade deve manter os relacionamentos com os papéis e produtos de trabalho.

Assim, objetivando-se manter estes relacionamentos das atividades com outras entidades do modelo durante a geração de *baselines*, decidiu-se assumir que a classe **Activity** conterá uma agregação de uma ou mais classes **ActivityDetail**.

Os stakeholders podem desempenhar diversos papéis (classe **Role**) durante a execução das atividades do projeto. Assim, para cada associação entre um papel e uma atividade (representado pela classe associativa **ActivityStakeholderWork**) deve haver também uma associação dessa atividade com um stakeholder capaz de desempenhar aquele papel. Além disso, como o conceito de papéis (classe **Role**) aparece em ambos os modelos, estes foram divididos em papéis gerenciais (classe **ManagerialRole**) e papéis produtivos (classe **ProductiveRole**), tal como ocorreu com as atividades.

Um produto de trabalho (classe **WorkProduct**), por sua vez, pode ser classificado como um produto gerencial (classe **Deliverable**) ou produtivo (classe **Artifact**), o qual deve estar associado a um tipo de produto (classes **DeliverableType** e **ArtifactType**, respectivamente). Cabe salientar que o modelo faz distinção das possíveis relações entre uma atividade e um artefato (criar/atualizar/consultar). Isto é importante para garantir a consistência do modelo a partir de regras, como a de número 16 na tabela apresentada mais adiante na seção 5.

Em relação à gerência de projetos, o PMBOK Guide representa suas práticas em duas dimensões lógicas. Uma dimensão descreve as áreas de conhecimento (classe **KnowledgeArea**) enquanto que a outra dimensão descreve os processos gerenciais de um projeto (classe **ManagementProcess**), os quais estão contidos em cinco grupos de processo (classe **ProcessGroup**). As áreas de conhecimento são responsáveis por descrever as principais competências que os gerentes de projeto devem desenvolver e derivam as áreas centrais (classe **CoreKnowledgeArea**) e as de apoio (classe **FacilitatingKnowledgeArea**). Assim, cada atividade gerencial pertence a um processo gerencial, sendo também relacionada a uma área de conhecimento.

Em contrapartida, o RUP define uma disciplina (classe **Discipline**) como sendo a divisão de elementos de processo em áreas de interesse. Cada disciplina é composta por um ou mais fluxos de trabalho (classe **WorkflowDetail**). Os fluxos de trabalho definem como os papéis produtivos devem colaborar entre si através de suas atividades.

## 4.2. Metamodelo Integrado entre o PMBOK e o OPEN

O metamodelo de integração PMBOK+OPEN (figura 6) possui uma estrutura similar ao apresentado na seção anterior, substituindo-se apenas o pacote referente ao processo de desenvolvimento de software. Assim, as classes dos pacotes PMBOK e *Common* são as mesmas que as apresentadas no metamodelo de integração PMBOK+RUP. Também permanecem inalterados os relacionamentos entre as classes pertencentes a estes dois pacotes. Os dois PDSs, porém, possuem características particulares que são refletidas em classes distintas e em diferentes relacionamentos com os pacotes PMBOK e *Common*.

Cabe salientar que a Figura 4 apresenta apenas os componentes centrais (classes **Producer**, **WorkUnit**, **WorkProduct**, **Stage** e **Language**) do framework do OPEN. Estes componentes representam classes abstratas e derivam num conjunto maior de sub-

classes (algumas destas sub-classes são ilustradas no pacote OPEN [13]).

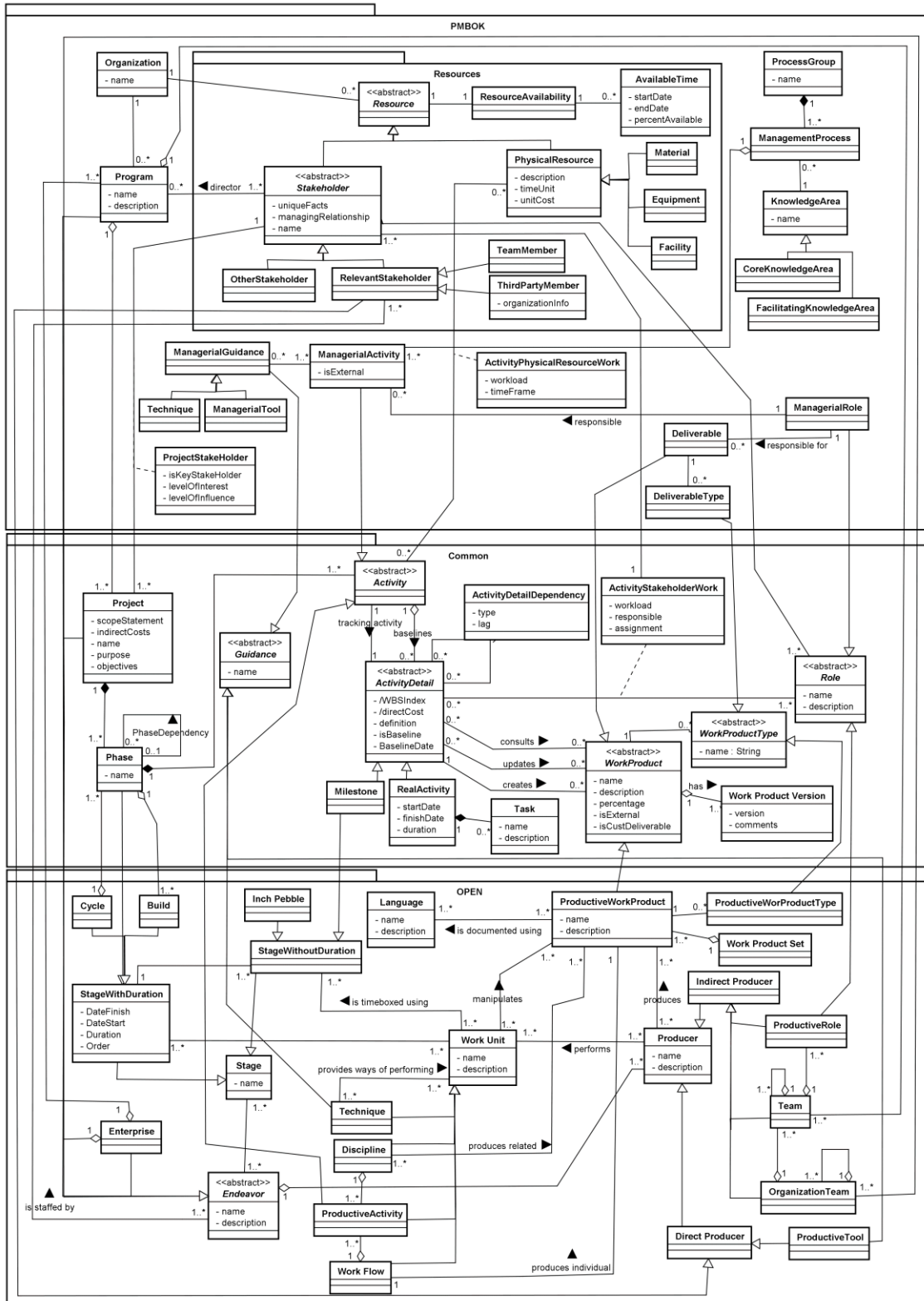


Figura 6. Metamodelo de integração PMBOK+OPEN

Neste modelo, a classe **Endeavor** representa o esforço empreendido pelos

produtores durante o desenvolvimento do projeto. Esta classe possui como responsabilidade desenvolver e/ou manter um ou mais produtos e serviços relacionados ao esforço empreendido. O OPEN define que a classe **Enterprise** representa o nível mais elevado de esforço, consistindo em uma coleção de programas relacionados que são controlados como uma única unidade. Possui as seguintes subclasses: **Enterprise**, **Program** e **Project**.

A classe **ProductiveWorkProduct** (originalmente denominada **WorkProduct** no metamodelo OPF) representa um produto do trabalho que é produzido, consumido ou modificado durante a execução de atividades produtivas por papéis produtivos. O conjunto de produtos produzidos pelas tarefas de uma ou mais atividades é representado pela classe **Work Product Set**, enquanto que a classe **Work Product Version** corresponde a uma versão específica do produto obtido através do processo de desenvolvimento incremental e iterativo.

A classe **Producer** é responsável por produzir ou modificar, diretamente ou indiretamente, os produtos de trabalho, além de realizar um ou mais serviços. É subdividida em produtores diretos (pessoas e ferramentas) e produtores indiretos (organização, equipe e papel).

Segundo [13], as unidades de trabalho (classe **WorkUnit**) modelam as operações coesas que são executadas pelos produtores durante o processo de entrega do projeto. Estas são classificadas como tarefas, técnicas, fluxos de trabalho e atividades.

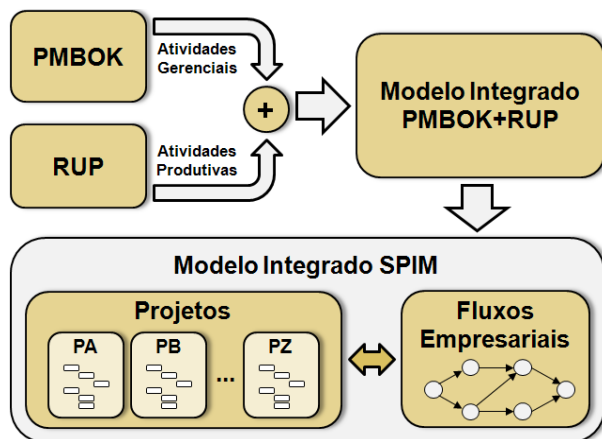
A classe **Stage** representa os intervalos de tempo que fornecem uma organização macro às unidades de trabalho, sendo subdividida em estágios com duração (ciclos, fases, fluxos, projeto, desenvolvimento, versões de entregas e entregas) e *milestones*.

Para demonstrar a viabilidade dos conceitos propostos, foi desenvolvido o modelo integrado SPIM, que auxilia o planejamento de projetos considerando os elementos que compõem o processo de desenvolvimento de software. Novamente, a atual versão do modelo implementa exclusivamente os conceitos advindos da integração do PMBOK com o RUP. Uma implementação que aborda também o OPEN está prevista futuramente.

## **5. Modelo de Integração entre a Gerência de Projetos e o Processo de Desenvolvimento de Software**

O estudo da integração com os dois processos distintos de desenvolvimento de software (RUP e OPEN) permitiu elaborar uma visão mais ampla de como a gerência de projetos pode positivamente contribuir no desenvolvimento de um produto de software. Assim, os metamodelos definidos durante esta pesquisa (PMBOK+RUP e PMBOK+OPEN) fornecem a estrutura conceitual necessária para o desenvolvimento de um modelo que auxilie o planejamento de projetos considerando os elementos que compõem o PDS.

A figura 7 representa as etapas que constituíram o desenvolvimento deste modelo de integração.



O estudo detalhado dos modelos do PMBOK e RUP permitiu identificar características relevantes da GP e dos PDSs. Este estudo resultou no desenvolvimento de um modelo de integração PMBOK+RUP. Posteriormente, o modelo SPIM foi desenvolvido implementando os conceitos advindos desta integração.

Também se identifica a dissociação entre o fluxo de atividades produtivas de um projeto e os demais fluxos empresariais da organização. Assim, o modelo SPIM tem como objetivo apoiar o processo decisório de uma organização permitindo o planejamento integrado de atividades gerenciais e produtivas.

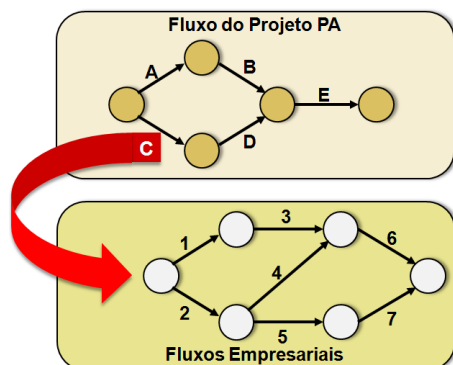
**Figura 7. Etapas do desenvolvimento do modelo de integração SPIM**

Conforme mencionado anteriormente, como o gerente de projetos pode não possuir todas as informações relevantes para o projeto, poderão ocorrer interações com outros departamentos da organização durante o planejamento de atividades de um projeto de software. Assim, o fluxo de atividades de um projeto de software pode interagir com os demais fluxos de atividades da organização (**Fluxos Empresariais**). É importante salientar que ambos os fluxos de trabalho são executados em paralelo, possuem recursos próprios e podem influenciar nos prazos das atividades e custos do projeto de software. As ferramentas atuais mais utilizadas, porém, não apresentam uma solução que permita o planejamento integrado de atividades gerenciais e de produção – e esta carência pode resultar em distorções no planejamento de projetos pela desconsideração das dependências entre as atividades dos diferentes fluxos de trabalho.

Durante o planejamento de atividades de um novo projeto de software, por exemplo, o gerente de projetos informa ao setor de recursos humanos sobre a necessidade de contratação de um administrador de banco de dados. Neste caso, constata-se a existência de uma relação de dependência entre as atividades do projeto de software (como a modelagem do banco de dados) com as atividades pertencentes ao fluxo de trabalho do setor de recursos humanos que são referentes à contratação do profissional solicitado para executar a atividade de produção. A dificuldade para visualizar esta interdependência dos fluxos de trabalho durante o planejamento de atividades gerenciais e de produção pode afetar negativamente o projeto, resultando, por exemplo, no aumento dos custos e em atrasos nos prazos do projeto. A preparação técnica e a liberação de uma sala ou equipamento de testes são outros exemplos cujas atividades não são exclusivas de um projeto em especial, mas de um fluxo comum da empresa, compartilhado pelos projetos em andamento e que utiliza recursos não alocados diretamente ao projeto de software.

A solução proposta nesta pesquisa permite que cada instância de um **fluxo empresarial** seja registrada como correspondente a uma atividade gerencial de apoio ao projeto de desenvolvimento de software (figura 8). Desta forma, cada atividade gerencial de apoio pode ter claramente definidas as suas relações de dependência com outras atividades do projeto. Isto permite a integração com uma ferramenta de *workflow*, de maneira que este último seja responsável por atualizar os prazos das atividades de apoio ao projeto (um fluxo empresarial pode atender múltiplos projetos

simultaneamente; daí a necessidade de um mecanismo tal como um *workflow*.)



Neste exemplo, a atividade “C” do projeto PA representa a execução total de um fluxo de trabalho empresarial (seus atributos, tal como o tempo, dependem diretamente dos atributos das atividades do fluxo empresarial). Deste modo, esta atividade do projeto só é dada como finalizada depois que todas as atividades do fluxo empresarial tiverem sido executadas. Assim, tão logo as atividades “6” e “7” do *workflow* terminem, a atividade “C” do projeto é dada como finalizada, liberando a execução para a atividade “D”. Cabe salientar que os exemplos estão na notação de redes PERT [17].

**Figura 8. Inter-relacionamento dos fluxos de trabalho**

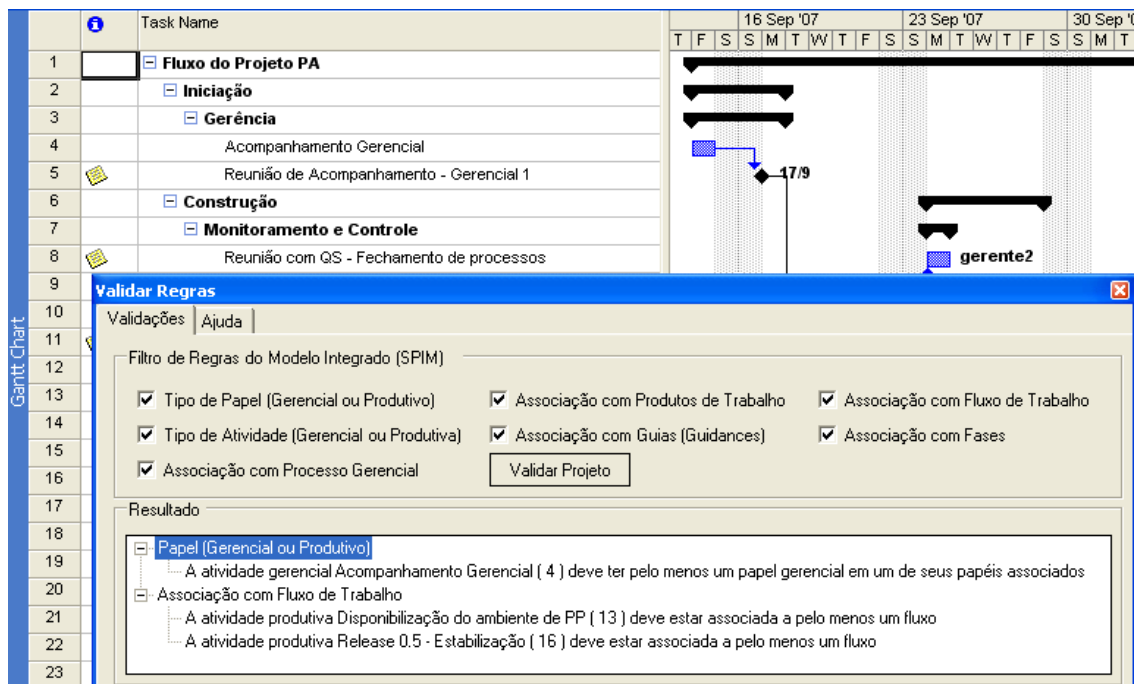
A integração dos modelos também permitiu que um conjunto de regras (ou restrições) pudesse ser derivado (Tabela 1). Tais características, em sua maioria para garantir a consistência do modelo, não puderam ser expressas no diagrama devido à falta de expressividade do diagrama de classes da UML para este fim. Estas regras podem, contudo, ser definidas através de uma linguagem para especificação formal de restrições, tal como a Object Constraint Language (OCL) [18]. Neste artigo optou-se por grafá-las em linguagem natural, para facilitar a compreensão.

**Tabela 1. Conjunto de Restrições Adicionais para o Modelo Integrado**

<b>Restrições Adicionais ao Modelo Integrado</b>	
1.	Um programa deve possuir um diretor. Logo, um stakeholder que é diretor de um programa deve possuir um papel gerencial;
2.	Um projeto deve ter apenas um stakeholder-chave, ou seja, um recurso responsável por direcionar e fundamentar o projeto (atributo <i>isKeyStakeholder</i> de <b>ProjectStakeholder</b> );
3.	Uma fase não pode ter ela mesma como predecessora ou antecessora;
4.	As fases do projeto não podem ocorrer em paralelo, de maneira que uma fase deve ser totalmente finalizada antes de iniciar outra;
5.	Uma atividade não pode ter ela mesma como predecessora ou antecessora;
6.	O fluxo de atividades não pode resultar em um ciclo; por exemplo, a atividade A é pré-requisito para a atividade B e a atividade B é pré-requisito para a atividade A;
7.	Uma mesma atividade não pode criar, modificar ou consultar um mesmo artefato. Para realizar tais operações devem ser criadas atividades distintas;
8.	Uma atividade gerencial deve ter pelo menos um papel gerencial como um de seus papéis;
9.	Uma atividade produtiva deve ter pelo menos um papel produtivo como um de seus papéis;
10.	O stakeholder responsável por uma atividade gerencial deve possuir um papel gerencial;
11.	O stakeholder responsável por uma atividade produtiva deve possuir um papel produtivo;
12.	Cada atividade do projeto deve ser de responsabilidade de apenas um indivíduo, mesmo que muitas pessoas venham a trabalhar naquela atividade;
13.	Uma atividade gerencial não pode produzir ou modificar um produto de trabalho produtivo, somente um produto de trabalho gerencial. Porém, esta atividade pode consultar um produto de trabalho produtivo;
14.	Uma atividade produtiva não pode produzir ou modificar um produto de trabalho gerencial, somente um produto de trabalho produtivo. Porém, esta atividade pode consultar um produto de trabalho gerencial;
15.	Para cada associação entre um papel e uma atividade (representado pela classe <b>ActivityStakeholderWork</b> ) deve haver também uma associação dessa atividade com um

- 
- stakeholder capaz de desempenhar aquele papel;
- 
16. Uma atividade somente pode atualizar ou consultar um produto de trabalho que já tenha sido criado por uma atividade antecessora;
- 
17. Uma atividade pode ou não ter *baselines*. Se tiver, a atividade original deve ter o atributo *IsBaseline=false* e todas as outras atividades (relacionadas via a associação *Baselines*) devem manter *IsBaseline* como *true*;
- 
18. Na classe associativa **ActivityStakeholderWork**, o stakeholder associado deve possuir o papel que se está associando à atividade;
- 
19. O papel do stakeholder envolvido deve ser compatível com o tipo de atividade (gerencial ou produtiva).
- 

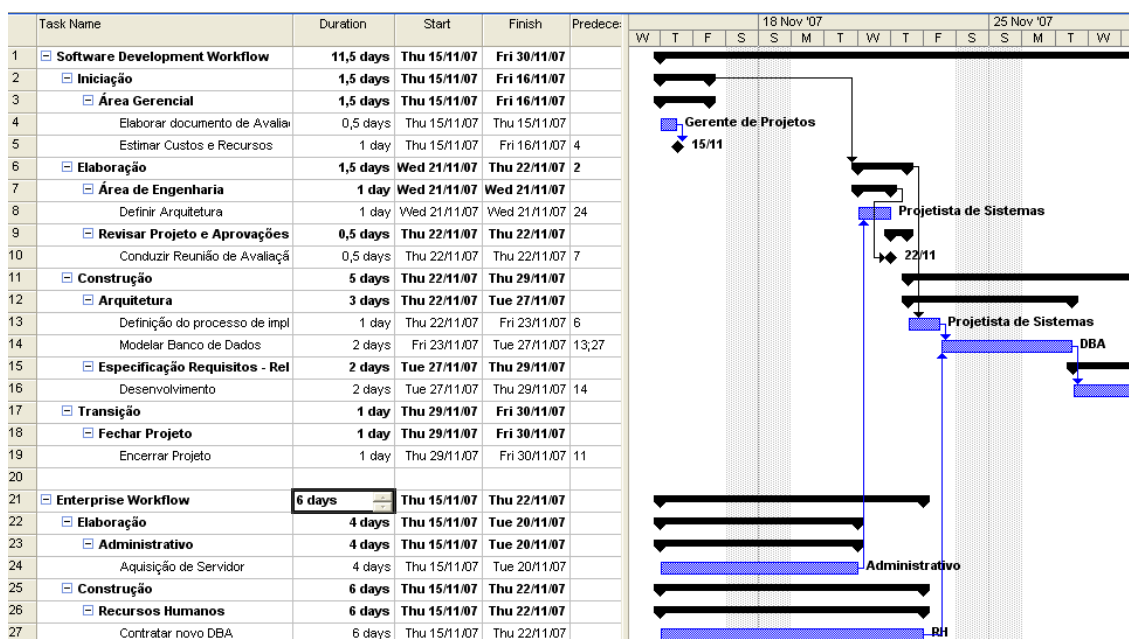
De todas as regras definidas para o modelo SPIM, oito foram avaliadas neste trabalho (Tabela 3). Assim, foi desenvolvido o protótipo **SPIT** para demonstrar os conceitos propostos pelo modelo integrado SPIM e pelo conjunto de regras descritas acima, fornecendo o suporte necessário para trabalhar com diferentes fluxos de trabalho, variáveis de projeto, responsabilidades e tarefas. O SPIT (figura 9), foi desenvolvido na linguagem C#.Net e atua como um *Add-in* para a ferramenta de gerenciamento de projetos Microsoft Office Project [19]. Esta escolha permite tomar proveito das funcionalidades que a ferramenta já implementa e que estão de acordo com o modelo de integração proposto. As informações adicionais necessárias para as validações dos conceitos propostos pelo modelo integrado SPIM estão contidas em campos customizados (*custom fields*) da ferramenta comercial.



**Figura 9. Protótipo Software Planning Integrated Tool (SPIT)**

Para realizar a operação de validação de uma restrição através do *Add-in*, deve-se primeiramente abrir um projeto cujos campos customizados já tenham sido previamente configurados. Após, deve-se selecionar a(s) regra(s) que se deseja avaliar e clicar no botão “Validar Projeto”. O sistema irá realizar o processamento e retornará os resultados em uma lista. Os parágrafos seguintes exemplificam o seu uso.

Uma característica muito importante do modelo SPIM, implementada no protótipo SPIT, é a capacidade de auxiliar o gerente de projetos na antecipação das necessidades dos projetos de software advindas das integrações com os diferentes setores da organização. Objetivando explicar esta funcionalidade, a Figura 10 apresenta um cronograma de projeto de software. Pode-se observar que as atividades (produtivas e gerenciais) do projeto de software estão definidas no agrupamento **Software Development Workflow** e as atividades dos fluxos empresariais ao agrupamento **Enterprise Workflow**. Neste exemplo, o início da atividade produtiva “8 - Definir Arquitetura” depende do término da atividade gerencial de apoio “24 - Aquisição de Servidor”. Este mesmo tipo de dependência acontece entre a atividade produtiva “14 - Modelar Banco de Dados” e a atividade gerencial de apoio “27 - Contratar novo DBA”.



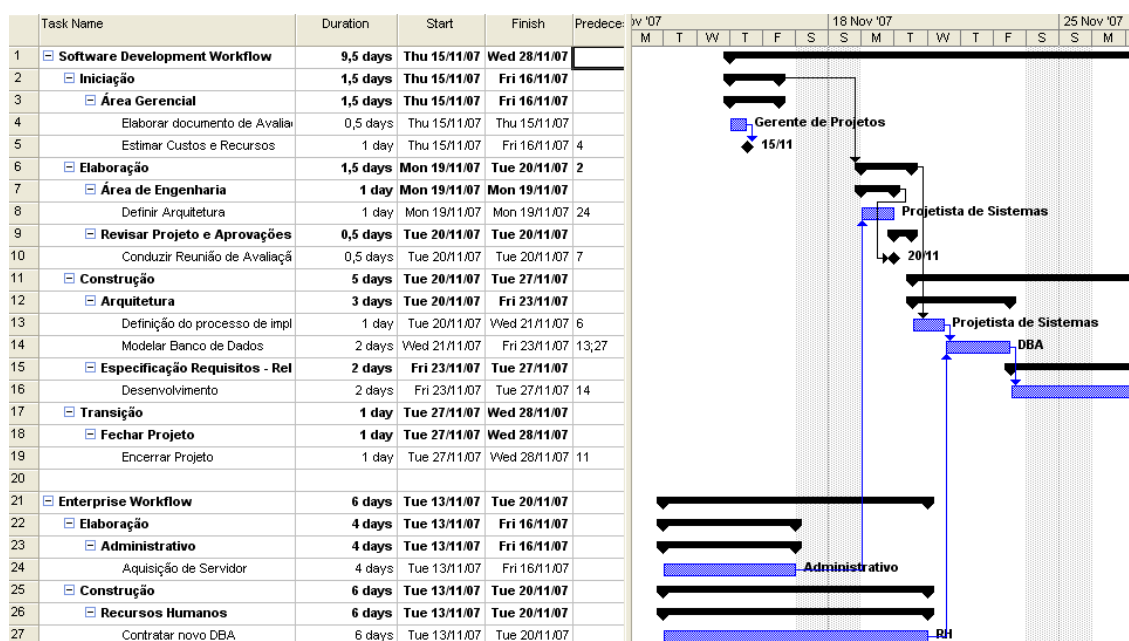
**Figura 10. Cronograma original (sem o re-planejando das atividades pelas regras do modelo SPIM)**

Ao realizar o planejamento deste projeto, o gerente de projetos identificou a necessidade de aquisição de um servidor e a contratação de pessoal (ambas são atividades pertencentes ao fluxo empresarial da empresa). Além disso, o gerente de projetos definiu que ambas essas atividades gerenciais de apoio seriam iniciadas na data definida para o início do projeto, ou seja, dia 17/11/2007. Esta definição, porém, ocasiona o atraso no início de algumas atividades do projeto de software. De acordo com o fluxo de atividades deste projeto de software, por exemplo, a atividade produtiva “8 - Definir Arquitetura” depende do término das atividades da fase de “Iniciação”, que ocorre no dia 16/11/2007 (sexta-feira). Logo, esta atividade poderia iniciar no dia 19/11/2007 (segunda-feira). Entretanto, devido à relação de dependência desta atividade do projeto com o fluxo empresarial responsável pela aquisição de uma máquina servidora (que tem seu término previsto para a terça-feira dia 20/11/2007), ela tem seu início postergado para o dia 21/11/2007 (quarta-feira). Dessa forma, pode-se observar que a dificuldade de identificar a distinção entre estes dois tipos de fluxos de trabalho está resultando em distorções no planejamento deste projeto (ambas as atividades



gerenciais de apoio estão afetando negativamente no prazo das atividades do projeto de software).

Entretanto, o modelo SPIM possui um conjunto de regras que auxilia o gerente de projetos na antecipação das atividades gerenciais de apoio e, conseqüentemente, no re-planejando do projeto de software devido à existência de relações de dependência destas atividades com as atividades do projeto. A Figura 11 apresenta o cronograma utilizado no exemplo anterior após o re-planejando das atividades afetadas por estas relações de dependência. Dessa forma, as atividades gerenciais de apoio “24 - Aquisição de Servidor” e “27 - Contratar novo DBA” foram antecipadas para o dia 13/11/2007 (quinta-feira). Conseqüentemente, as atividades pertencentes ao fluxo deste projeto de software também tiveram seus prazos modificados: por exemplo, a atividade “8 - Definir Arquitetura” teve sua data de início antecipada para o dia 19/11/2007 (segunda-feira).



**Figura 11. Cronograma modificado (após o re-planejando das atividades pelas regras do modelo SPIM)**

Além disso, a distinção explícita entre as atividades pertencentes aos fluxos empresariais e de um projeto de software específico permite a identificação e a mensuração dos custos indiretos do projeto de software (advindos das atividades de apoio da organização) pelo gerente de projetos.

Percebe-se, dessa forma, que alguns tipos de atividades gerenciais são inerentes ao processo e não aparecem no momento do planejamento do projeto. São justamente estas atividades (ou suas dependências) que na maioria das vezes causam um atraso considerável no cronograma e não são consideradas na definição dos riscos do projeto. Conseqüentemente, o modelo SPIM pretende ajudar os gerentes a resolver problemas relacionados à definição inadequada e a inter-relação das atividades em um projeto de software. A análise de como o conhecimento sobre a gerência de projetos permite aperfeiçoar os processos de desenvolvimento de software atuais torna possível o desenvolvimento de novas ferramentas capazes de suportar diferentes níveis de

automatização no planejamento e na execução de atividades num projeto de software.

## 6. Avaliação do modelo

Depois de implementado o modelo, conforme descrito detalhadamente em [20], elaborou-se um *Survey* qualitativo envolvendo um grupo de doze gerentes de projetos que atuam em nove empresas de desenvolvimento de software. Esta análise teve como objetivo a avaliação dos conceitos advindos do modelo integrado SPIM no que diz respeito à sua aceitação e aplicabilidade. Nesse artigo apresentam-se as principais observações obtidas da análise do modelo SPIM realizada em [20].

A aquisição dos dados foi realizada por meio de um questionário (Anexo A) elaborado em conformidade com [21]. A elaboração do questionário foi realizada de acordo com os seguintes passos:

- coleta de dados preliminares a respeito do tema e da população alvo da pesquisa;
- discussão em grupo sobre as questões;
- elaboração do rascunho do questionário;
- realização de um pré-teste;
- revisão do instrumento baseado nas observações obtidas no pré-teste; e
- delineamento do questionário final.

Na primeira etapa realizou-se o levantamento bibliográfico e o estudo do referencial teórico que permitiu aprofundar os conhecimentos sobre os processos de desenvolvimento de software e os modelos de gerência de projetos. Além disso, foi definida a população alvo desta pesquisa (gerentes de projetos de software, neste caso). Em seguida, foram realizadas reuniões entre os pesquisadores e o professor orientador para o levantamento das questões e a estruturação do roteiro de entrevistas. Estas reuniões resultaram na elaboração de um rascunho do questionário necessário para a aquisição dos dados desta pesquisa. Posteriormente, foi realizada a validação de face e conteúdo do protocolo de avaliação da pesquisa por um pesquisador sênior. Com base nesta validação o protocolo sofreu pequenas correções. A seguir foi realizado o pré-teste com um gerente de projetos, onde foi possível realizar os últimos ajustes no roteiro de entrevistas. A versão final do protocolo de pesquisa contendo o roteiro de entrevistas e descrevendo os procedimentos seguidos encontra-se no Anexo A.

O questionário desenvolvido nesta pesquisa é constituído por um conjunto de 33 questões: as 8 primeiras capturam o perfil das empresas enquanto que outras 10 perguntas foram dirigidas especificamente aos gerentes de projeto participantes. As questões restantes abordaram a percepção das contribuições do modelo SPIM do ponto de vista dos gerentes de projeto.

A aquisição dos dados deste questionário foi realizada durante o mês de novembro de 2007. Cada indivíduo participante da pesquisa respondeu separadamente o questionário. Todos os participantes receberam um breve treinamento no modelo SPIM e, posteriormente, foi dada a oportunidade para que fizessem perguntas e dirimissem dúvidas sobre o trabalho proposto. Em seguida, foram apresentados a uma mesma

descrição sobre um projeto predefinido e foram convidados a realizar o planejamento do projeto correspondente usando o SPIT. Depois disso, responderam ao questionário, cujos resultados foram utilizados para refinar o modelo. Objetivando-se evitar possíveis distorções nos resultados obtidos, não houve qualquer interação com o entrevistador tanto na fase de experimentação do SPIT quanto na fase de resolução do questionário. Os resultados são apresentados a seguir.

As empresas envolvidas estão distribuídas em diferentes segmentos da tecnologia da informação (três fábricas de software, duas companhias de desenvolvimento de tecnologia móvel, duas companhias de desenvolvimento web, uma instituição governamental e um centro de pesquisa e desenvolvimento). Juntas estas empresas somam mais de 1620 profissionais relacionados à área de tecnologia da informação (TI), dos quais 90 são gerentes de projeto e 823 são desenvolvedores de software.

A análise das perguntas relacionadas ao perfil das empresas revela que quase 67% destas empresas adotam algum tipo de processo de desenvolvimento de software baseado no RUP. Além disso, todas as empresas adotam os conceitos encontrados no PMBOK Guide nos seus projetos. Estes dados são importantes, pois indicam que os gestores entrevistados trabalham em ambientes que permitem a utilização das idéias propostas pelo modelo SPIM.

Com relação às perguntas relacionadas ao perfil dos indivíduos entrevistados, estes apresentam uma média de 12,25 anos de experiência profissional (mín.=7; máx.=20). Além da referida experiência profissional dos entrevistados, a média de experiência em gestão de projetos foi de 5,04 anos, variando de 1 a 12 anos (3,43 foi o desvio padrão e a média da amostra foi superior a 7 anos). Estes dados traduzem a vasta gama de experiência dos entrevistados relativamente à gestão de projetos.

Sobre a análise das questões do modelo SPIM, inicialmente os entrevistados foram questionados sobre os benefícios diretos identificados em realizar o planejamento integrado de atividades gerenciais e produtivas em um projeto de software. Nesta questão, conforme ilustrado na Tabela 2, destaca-se que todos os gestores responderam que o SPIM permite evitar as distorções no planejamento de projetos de software ocasionadas pela desconsideração de que as atividades gerenciais de apoio utilizam recursos não alocados diretamente ao projeto de software. Além disso, alguns entrevistados mencionaram a possibilidade de manter conformidade com os processos organizacionais e de desenvolvimento, bem como o auxílio na identificação prévia dos tempos das demandas de algumas atividades (evitando impactos nas atividades produtivas dependentes).

**Tabela 2: Benefícios percebidos em fazer o planejamento integrado de atividades gerenciais e produtivas.**

<b>Questão</b>	<b>Média</b>
Redução do tempo durante o processo elaboração do projeto.	58%
Identificação das dependências entre as atividades gerenciais de apoio e as atividades do projeto de software.	100%
Identificação e mensuração dos custos indiretos do projeto, devido as atividades gerenciais de apoio.	67%

Capacidade de ter acesso a informações dos fluxos empresariais	40%
A capacidade de evitar distorções ao planejar o projeto quando atividades gerenciais de apoio são envolvidas.	100%

Um subconjunto de oito de todas as regras também foi avaliado pelos entrevistados (Tabela 3). Os gestores avaliaram cada regra de acordo com a seguinte escala: 1-nenhum, 2-baixo, 3-moderado, 4-alto e 5-muito alto.

**Tabela 3: Conjunto de regras de validação do modelo SPIM e a sua avaliação pelos gestores.**

<b>Restrições Verificadas pelo Modelo SPIM</b>	<b>Média</b>	<b>Desvio Padrão</b>
1 Uma atividade não pode criar, modificar ou consultar um mesmo artefato num dado momento. Estas operações devem ser efetuadas por atividades distintas.	3,75	0,75
2 As atividades gerencial e gerencial de apoio não podem produzir ou modificar produtos de trabalho produtivos, mas somente produtos de trabalho gerenciais. Entretanto, elas ainda podem consultar produtos de trabalho produtivos.	4,42	0,79
3 A atividade produtiva não pode produzir ou modificar produtos de trabalho gerenciais, somente produtos de trabalho produtivos. Entretanto, ela ainda pode consultar produtos de trabalho gerenciais.	4,42	0,79
4 Uma atividade somente pode atualizar ou consultar um produto de trabalho que já tenha sido criado por uma atividade antecessora.	4,08	0,90
5 O papel do stakeholder envolvido deve ser compatível com o tipo de atividade (gerencial ou produtivo).	4,25	0,87
6 O benefício de informar o tipo de guidance relacionado, se produtivo ou administrativo, para cada atividade.	3,83	1,03
7 As atividades gerencial e gerencial de apoio devem ter pelo menos um papel gerencial como um de seus papéis.	4,50	0,67
8 A atividade produtiva deve ter pelo menos um papel produtivo como um de seus papéis.	4,42	0,67

A média geral para as regras selecionadas foi de 4,20, o que indica um ótimo nível de aceitação das restrições propostas no modelo SPIM. Apesar da alta pontuação, as regras #1 e #6 foram consideradas como as que menos trouxeram benefícios entre todas as regras (respectivamente, 3,75 e 3,83, valores próximos da pontuação considerada como sendo alto grau de benefício). Em contrapartida, as regras de número #2, #3, #7 e #8 obtiveram a maior média dentre as restrições analisadas, próximo de 4,5 pontos, sendo consideradas como as regras favoritas entre os entrevistados. Estes números revelam uma relativa importância de todos os assuntos que devem ser considerados ao se realizar a verificação do plano do projeto de acordo com o modelo SPIM. Além disso, estas informações indicam quais itens exigem preocupação mais específica segundo a opinião dos entrevistados.

Sobre a questão relativa à identificação dos pontos fortes observados no modelo integrado SPIM (Anexo A), todos os gestores concordaram ser adequada a distinção proposta nesta pesquisa entre as atividades produtivas e gerenciais de um projeto de software e as atividades gerenciais de apoio dos demais departamentos da organização.

A visibilidade das atividades gerenciais de apoio em conjunto com as atividades do projeto de software também foi identificada como sendo um ponto forte do modelo SPIM.

Relativamente à questão 6 do questionário, cerca de 60% dos entrevistados salientaram que muitas vezes o gerente de projetos somente percebe a necessidade de ter solicitado anteriormente uma informação de outro departamento da empresa no momento de executar uma determinada atividade do projeto que depende deste outro departamento (por exemplo: a aquisição de equipamento ou a contratação de um novo desenvolvedor). Estas observações corroboram com o ponto de vista desta pesquisa para se desenvolver o modelo integrado.

Como último ponto a ser observado neste artigo, todos os gerentes de projetos constataram que o modelo SPIM colaborou na identificação das dependências entre as atividades pertencentes ao fluxo de um projeto de software e aquelas pertencentes aos fluxos empresariais (questão 8). Dessa forma, há evidências que o SPIM permite antecipar as necessidades advindas das áreas de apoio durante o planejamento do projeto de software.

Os resultados observados reafirmam os benefícios que o modelo SPIM contribui para solucionar os problemas relacionados à definição inadequada de tarefas (por exemplo, o aumento dos custos e atrasos nos prazos do projeto) ocasionadas pela a dificuldade de visualizar a interdependência entre os fluxos de trabalho da empresa e de um projeto de software específico.

## **7. Considerações Finais**

Este trabalho apresentou uma proposta para a integração dos principais conceitos sobre gerência de projetos e processos de desenvolvimento de software. Inicialmente, identificou-se a importância das atividades de gerência de projetos durante um projeto de desenvolvimento de software. Em seguida, observou-se a carência existente no quesito de GP nos PDSs atuais. Após uma análise individual de cada metamodelo base, foram apresentadas as principais colaborações ao metamodelo proposto em [3]. Depois, foi apresentado o protótipo que implementa o modelo de integração SPIM e o seu conjunto de restrições. Finalmente, foram analisados os principais resultados obtidos em uma avaliação com gerentes de projetos de software.

A análise de como o conhecimento sobre gerência de projetos permite aperfeiçoar os processos de desenvolvimento de software atuais torna possível o desenvolvimento de novas ferramentas capazes de suportar diferentes níveis de automatização no planejamento e na execução de atividades num projeto de software. Por essa razão, o modelo integrado foi desenvolvido com os seguintes objetivos:

- (a) permitir o planejamento de projetos considerando os elementos que compõem a GP e o PDS;
- (b) fazer a distinção dos tipos de atividades (gerencial ou produtiva) e produtos;
- (c) adicionar a noção de disponibilidade a um recurso, de modo que permita automatizar os processos de alocação de recursos em projetos de software;

- (d) prever a informação de carga de trabalho (*workload*) para as associações de um papel, atividade e stakeholder, preparando para a automatização; e
- (e) distinguir as possíveis relações entre uma atividade e um artefato (criar/atualizar/consultar);

Durante o desenvolvimento desta pesquisa foram identificadas algumas questões que necessitam de maior desenvolvimento:

- (a) formalização das restrições do metamodelo através da linguagem OCL;
- (b) ampliação do estudo sobre a integração do processo de gerência com outros processos de desenvolvimento de software, tais como Extreme Programming (XP) [22] e MSF; e
- (c) avaliação do modelo proposto com empresas de software, utilizando o protótipo em projetos reais de longo prazo.

A principal limitação deste estudo está associada à estratégia adotada para a avaliação do modelo integrado. O estudo visou levantar dados iniciais que orientassem desenvolvimentos futuros e a amostra foi escolhida de acordo com a acessibilidade e conveniência. Esta abordagem resultou em um número reduzido de empresas e respondentes na pesquisa realizada. Além disso, seria interessante se a pesquisa envolvesse projetos reais de desenvolvimento de software. Entretanto, não foi possível realizar a avaliação nesses moldes devido à limitação imposta pelo cronograma desta pesquisa. Outra limitação importante deve-se ao fato que o modelo proposto SPIM implementa exclusivamente os conceitos advindos da integração do PMBOK com o RUP, restringindo sua aplicação a empresas que adotam processos baseados nos seus respectivos fundamentos.

Não obstante, acredita-se ser possível estender os metamodelos de integração para outros PDSs já que, de acordo com [24], os diferentes modelos de desenvolvimento de software compartilham atividades fundamentais, tais como a especificação, projeto, implementação, validação e evolução do software. Atualmente, existem diversos artigos na literatura sobre diferentes integrações de GP com PDSs, tais como [3] e [25]. Entretanto, parece não haver estudos suficientes para suprir a carência no quesito de gerência de projetos destes processos. Dessa forma, a continuidade desta pesquisa indica novas contribuições para a Engenharia de Software. Em um trabalho paralelo, um modelo multi-critérios para alocação de recursos já está sendo integrado ao protótipo.

## Referências

- [1] R. Pressman, “Software Engineering : a practitioner's approach”, 5ª Ed., McGraw-Hill, 2001.
- [2] K. Schwalbe, “Information Technology Project Management”, 2ª Ed., Thomson Learning, Canada, 2002.
- [3] D. Callegari, e R. Bastos, “Project Management and Software Development Processes: Integrating RUP and PMBOK”, ICSEM - International Conference on Systems Engineering and Modeling, Haifa, Israel, 2007.
- [4] PMBOK – “A Guide to the Project Management Body of Knowledge”, 2000.

- [5] B. Henderson-Sellers, R. Dué, I. Graham, e G. Collins, “Third generation OO processes: a critique of RUP and OPEN from a project management perspective”, ASPEC - Seventh Asia-Pacific Software Engineering Conference, 2000.
- [6] PEP – “A RUP Adoption Process. Rational Process Engineering Process”, Rational Unified Process, 2003.
- [7] I. Graham, B. Henderson-Sellers, e H. Younessi, “The OPEN Process Specification”, Addison Wesley, 1997.
- [8] W. S. Humphrey, T.R. Snyder, and R.R. Willis, “Software Process Improvement at Hughes Aircraft”, Institute of Electrical and Electronic Engineers, pp. 11-23, 1991.
- [9] PMI Web Site, <http://www.pmi.org/info/default.asp>, Novembro, 2006.
- [10] Jacobson, I., Booch G., Rumbaugh J., “The Unified Software Development Process”, Upper Saddle River, Addison Wesley, 2001.
- [11] A. Bencomo, “Extending the RUP, Part 1: Process Modeling”, <http://www-128.ibm.com/developerworks/rational/library/4721.htm>, Novembro, 2006.
- [12] Rational Software Corporation Web Site, “Rational Unified Process: Best Practices for Software Development Teams”, <http://www.rational.com/products/rup/whitepapers.jsp>, Novembro, 2006.
- [13] OPFRO Web Site, “OPEN Process Framework”, <http://www.opfro.org/index.html>, Fevereiro 2007.
- [14] B. Henderson-Sellers, G. Collins, R. Dué e I.M. Graham. “A Qualitative Comparison of Two Processes for Object-Oriented Software Development”. Information and Software Technology. United Kingdom: Elsevier Science, vol. 43, November 2001, pp. 705-724.
- [15] A. Rehman e R. Hussain. “Software Project Management Methodologies/ Frameworks Dynamics ‘A Comparative Approach’ ”. 7th International Conference on Information and Emerging Technologies, Karachi, Pakistan, July 2007, pp. 1-5.
- [16] Office of Government Commerce (OGC) Web Site, “Introduction to PRINCE2”, <http://www.ogc.gov.uk>, Março 2007.
- [17] R. Burke, “Project Management: Planning and Control Techniques”. 3ª Ed., John Wiley & Sons, 2001.
- [18] J. Warmer, A. Kleppe, “The Object Constraint Language - Precise Modeling with UML”, Addison Wesley, 1999.
- [19] Microsoft Project Web Site, [www.microsoft.com/Project/](http://www.microsoft.com/Project/), Março, 2007.
- [20] D. Callegari, M. Rosito, M. Blois e R. Bastos. “An Integrated Model for Managerial and Productive Activities in Software Development”. ICEIS 2008 - 10th International Conference on Enterprise Information Systems, 2008, Barcelona.
- [21] L. M. Rea e R. Parker, “Designing and Conducting Survey Research: A Comprehensive Guide”. San Francisco: Jossey-Bass, 2005, 304 p.
- [22] K. Beck, “Extreme Programming Explained: Embrace Change”, Ad. Wesley, 1999.
- [23] R. Hundhausen, “Working with Microsoft Visual Studio 2005 Team System”, White Paper, 2006.
- [24] I. Sommerville, “Software engineering”, 5ª Ed., Addison-Wesley, 1995.
- [25] S. S. Alhir, “Integrating the Project Management Body of Knowledge (PMBOK)

Guide and the Unified Process (UP)”, White Paper, 2003.



## Anexo A

### Questionário de Avaliação do Perfil dos Entrevistados

#### Perfil da Empresa

1. Razão Social:
2. Ramo de Atividade:
3. Tempo de atuação no mercado de Tecnologia da Informação:
4. Número aproximado de funcionários:
5. Número aproximado de gerentes de projeto:
6. Número aproximado de desenvolvedores de software:
7. A empresa adota algum processo de desenvolvimento de software baseado no RUP para os seus projetos?  
 Sim       Não
8. A empresa adota os conceitos gerenciais advindos do PMBOK Guide em seus projetos?  
 Sim       Apenas parcialmente       Não

#### Perfil do Gerente de Projetos

1. Nome:
2. Email:
3. Idade:
4. Experiência profissional no ramo da indústria de software (em anos):
5. Experiência profissional como gerente de projetos (em anos):
6. Número médio de projetos que você costuma gerenciar ao mesmo tempo?
7. Já realizou algum treinamento referente à Gerência de Projetos?  
 Sim      Nome do Curso: \_\_\_\_\_ Duração: \_\_\_\_\_  
 Não
8. Como você qualificaria seu conhecimento relacionado à Gerência de Projetos?  
 Nenhum     Pouco     Moderado     Avançado
9. Como você qualificaria seu conhecimento relacionado aos conceitos advindos do PMBOK Guide?  
 Nenhum     Pouco     Moderado     Avançado
10. Como você qualificaria seu conhecimento relacionado aos conceitos advindos do RUP?  
 Nenhum     Pouco     Moderado     Avançado

## Questionário de Avaliação do Modelo Integrado SPIM

### Glossário

- *Atividade Produtiva:* representa uma unidade de trabalho, desempenhada por um papel produtivo, que produz um resultado significativo no contexto de um projeto de software.
  - *Exemplo:* modelagem do banco de dados da aplicação.
- *Atividade Gerencial:* descreve as atividades gerenciais, desempenhadas por papéis gerenciais, do projeto. Estas atividades gerenciais pertencem exclusivamente ao projeto de desenvolvimento de software.
  - *Exemplo:* organizar e conduzir uma reunião de acompanhamento do projeto.
- *Atividade Gerencial de Apoio:* São atividades gerenciais que pertencem exclusivamente aos demais fluxos de atividades de suporte ao projeto da organização (aqui denominados Fluxos Empresariais).
  - *Exemplo:* contratar um administrador de banco de dados (atividade realizada pelo setor de recursos humanos da organização).

### Avaliação do Modelo Integrado SPIM

1. Ao realizar planejamento de um projeto, o gerente de projetos pode necessitar interagir com outros departamentos da organização a fim de obter informações relevantes sobre as atividades gerenciais de apoio necessárias para a realização das atividades produtivas do projeto. Por exemplo, durante o planejamento de atividades de projeto de software, o gerente de projetos informa ao setor de recursos humanos sobre a necessidade de contratação de um administrador de banco de dados. Neste caso, constata-se a existência de uma relação de dependência entre as atividades do projeto de software (tais como, a modelagem do banco de dados) com as atividades pertencentes ao fluxo de trabalho do setor de recursos humanos referentes à contratação do profissional requerido para executar a atividade de produção. Assim, percebe-se que o fluxo de atividades de um projeto de software pode interagir com os demais fluxos de atividades da organização (fluxos empresariais). Dessa forma, marque abaixo os benefícios que você percebe em fazer o planejamento integrado de atividades gerenciais e produtivas para projetos de desenvolvimento de software?

- Redução do tempo no processo de elaboração do planejamento do projeto;
- Identificação das dependências entre as atividades gerenciais de apoio e de produção;
- Identificação e mensuração dos custos indiretos do projeto advindos das atividades gerenciais de apoio;
- Capacidade de ter acesso a informações dos fluxos empresariais (pertencentes aos outros departamentos da organização);
- Capacidade de evitar distorções no planejamento de projetos (tais como, o aumento dos custos e atrasos nos prazos do projeto) pela desconsideração de que as atividades gerenciais de apoio utilizam recursos não alocados diretamente ao projeto de software;
- Outros:

2. Em função da integração dos modelos da gerência de projetos e dos processos de desenvolvimento

de software, pode-se identificar um conjunto de regras (ou restrições). Tais regras visam, em sua maioria, garantir a consistência do modelo de integração SPIM. Dessa forma, para cada questão, marque com um “X” o grau de benefício observado dessas restrições para o planejamento do projeto de acordo com a classificação apresentada abaixo:

- 1 – Nulo
- 2 – Baixo
- 3 – Moderado
- 4 – Alto
- 5 – Muito alto

Restrições Verificadas pelo Modelo Integrado SPIM	Resposta				
	1	2	3	4	5
2.1 Uma mesma atividade (produtiva, gerencial ou gerencial de apoio) não pode criar, modificar ou consultar um mesmo artefato. Para realizar tais operações devem ser criadas atividades distintas;					
2.2 As atividades gerencial e gerencial de apoio não podem produzir ou modificar produtos de trabalho produtivos, somente produtos de trabalho gerenciais. Porém, estas atividades podem consultar produtos de trabalho produtivos;					
2.3 A atividade produtiva não pode produzir ou modificar produtos de trabalho gerenciais, somente produtos de trabalho produtivos. Porém, esta atividade pode consultar produtos de trabalho gerenciais;					
2.4 Uma atividade (produtiva, gerencial ou gerencial de apoio) somente pode atualizar ou consultar um produto de trabalho que já tenha sido criado por uma atividade antecessora;					
2.5 O papel do <i>stakeholder</i> envolvido deve ser compatível com o tipo de atividade (gerencial ou produtivo);					
2.6 A adição das informações relativas ao relacionamento das atividades com os tipos de <i>guidances</i> (gerencial ou produtivo);					
2.7 As atividades gerencial e gerencial de apoio devem ter pelo menos um papel gerencial como um de seus papéis;					
2.8 A atividade produtiva deve ter pelo menos um papel produtivo como um de seus papéis;					

3. Quais foram os pontos fortes observados no modelo integrado SPIM durante a sua utilização neste projeto?
4. Quais foram os pontos fracos observados no modelo integrado SPIM durante a sua utilização neste projeto?
5. Objetivando realizar o planejamento integrado dos conceitos advindos da gerência de projetos e dos processos de desenvolvimento de software, o modelo SPIM propõe a distinção explícita entre as atividades produtivas e gerenciais de um projeto de software com as atividades gerenciais de apoio dos demais departamentos da organização. Comente se você achou adequada esta distinção proposta nesta pesquisa.
6. O modelo SPIM foi concebido considerando que a dificuldade para visualizar a interdependência dos fluxos de trabalho da empresa e do projeto de software durante o planejamento do projeto pode resultar, por exemplo, no aumento dos custos e em atrasos nos prazos do projeto. Você

concorda que a dificuldade de identificar as atividades gerenciais de apoio, durante o planejamento do projeto, pode afetar negativamente o projeto? Justifique sua resposta.

7. As atividades gerenciais de apoio não são exclusivas de um projeto em especial, mas de um fluxo comum da empresa, compartilhado pelos projetos em andamento e utiliza recursos não alocados diretamente ao projeto de software. Você concorda se esta lógica é benéfica na definição do planejamento do projeto? Discorra sobre os motivos que influenciaram sua resposta.
  
8. Considerando que o fluxo de atividades de um projeto de software e os demais fluxos de atividades de apoio ao projeto da organização são executados de forma distinta, comente se modelo SPIM colaborou para a identificar a relação de dependência entre as atividades pertencentes a estes dois tipos de fluxos de trabalho, permitindo antecipar as necessidades advindas das áreas de apoio da organização durante o planejamento do projeto.