

Um Metamodelo para Adaptação de Processos de Desenvolvimento de Software

Eliana B. Pereira¹, Ricardo M. Bastos¹, Rafael T. Brito¹, Marcelo H. Yamaguti¹

¹Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681 - Prédio 30, Bloco C, CEP: 90619-900 - Porto Alegre - RS

{epereira,bastos,ad107126,yamaguti}@inf.pucrs.br

***Abstract** – The main objective of this paper is to present a metamodel about the main elements that constitute a standard process, as well as which are the main existing relationships between them. Based on this, a rules set for tailoring process are presented with main objective of aiding software development organizations in tailoring process, aiming to keep conformity with the standard process.*

***Resumo** – O principal objetivo deste artigo é apresentar um metamodelo sobre os principais elementos que constituem um processo padrão, bem como quais os principais relacionamentos existentes entre eles. A partir disto, um conjunto de regras para adaptação de processos é apresentado com objetivo principal de apoiar organizações de desenvolvimento de software na adaptação de processos de software, visando manter conformidade com o processo padrão.*

1. Introdução

Atualmente, as organizações de desenvolvimento software estão cada vez mais interessadas em utilizar processos de software bem definidos para o desenvolvimento de seus produtos. A implantação de um processo padrão de desenvolvimento de software nas organizações tem sido fortemente demandado pela indústria, e vários processos têm sido sugeridos por organizações e companhias internacionais [Yoon 2001].

Contudo, a criação de um processo padrão de desenvolvimento de software não garante melhoria na qualidade, pois na maioria das vezes a utilização do mesmo processo para todos os tipos de projeto não é possível, isto porque, diferentes projetos possuem diferentes necessidades.

A maneira de fornecer flexibilidade aos projetos sem deixar de padronizar é através da adaptação do processo padrão de desenvolvimento de software. Adaptar um processo padrão consiste em adicionar, excluir ou modificar alguns de seus elementos e relacionamentos, sendo o processo resultante mais apropriado para o alcance das metas do projeto [Yoon 2001] [Jalote 2002]. Assim, durante um processo de adaptação a organização deve permitir um conjunto de derivações a partir de seu processo padrão, tornando possível seu ajuste a necessidades específicas de projetos ou negócios em particular.

Entretanto, é importante considerar que permitir estas derivações a partir do processo padrão sem controle, efetivamente implica que nenhum padrão existe [Jalote 2002]. Por isto, as organizações devem estar preparadas para aplicar algumas regras durante o processo de adaptação. Estas regras devem garantir que a conformidade do processo padrão de desenvolvimento de software seja mantida, o que exige um esforço considerável por parte das organizações. Este esforço diz respeito principalmente a conhecer todos os elementos que constituem o processo padrão e os principais

relacionamentos existentes entre eles, já que em um procedimento de adaptação tais elementos e relacionamentos irão sofrer alterações.

Neste sentido, o principal objetivo deste artigo é identificar como devem ser tratados os elementos de um processo padrão de desenvolvimento de software durante sua adaptação. Para tornar isto possível, inicialmente um estudo de caso foi realizado em uma grande organização de desenvolvimento software. A organização estudada utiliza um processo padrão baseado no RUP [Kruchten 2000] para execução de seus projetos e é certificada com CMM nível 2. Este estudo de caso teve como principal objetivo o entendimento sobre como os elementos de um processo padrão estão estruturados, além da identificação das várias dependências entre estes elementos. Também foram identificados os procedimentos adotados pela organização para a adaptação de seu processo padrão para projetos de manutenção e desenvolvimento de novos sistemas.

A partir disto, como forma de generalizar os resultados obtidos, realizou-se um estudo no metamodelo do processo RUP, sendo proposta uma extensão ao mesmo com a inclusão de novos elementos para suporte ao processo de adaptação. Com base no metamodelo proposto, resultados do estudo de caso e estudos realizados na literatura definiu-se um conjunto de regras para adaptação de processos. A contribuição principal destas regras é apoiar organizações de desenvolvimento de software na adaptação de processos de software, visando manter conformidade com o processo padrão.

Este artigo está organizado da seguinte forma: na seção 2 é realizada uma breve apresentação do RUP. A seção 3 apresenta o metamodelo proposto. Ainda, apresenta o conjunto de regras para adaptação de processos. Na seção 5, são apresentados os trabalhos relacionados. E por fim, a seção 6 apresenta as conclusões.

2. Rational Unified Process (RUP)

O RUP é um processo de desenvolvimento de software que apresenta uma abordagem disciplinada para organizações. É considerado como um *framework* composto por vários tipos de elementos que, em conjunto, formam um processo completo de desenvolvimento de software. Segundo [Jacobson 2001], este *framework* pode ser adaptado e estendido de acordo com as necessidades da organização podendo ser aplicado a diferentes áreas, tipos de organização, níveis de competência e tamanho de projeto.

Conforme [Kruchten 2000], o RUP é formado basicamente por quatro elementos primários de modelagem: papel, atividade, artefato e fluxo, sendo que o elemento fluxo pode ser central ou detalhado. Os fluxos centrais são as disciplinas do processo e são formados por um conjunto de fluxos detalhados. Fluxos detalhados são agrupamentos de atividades relacionadas e geralmente são usados para detalhar fluxos de informação. Na seção 3 deste trabalho, uma descrição mais detalhada sobre conceitos de elementos do RUP será realizada.

O RUP apresenta um conjunto de nove disciplinas, sendo seis delas de engenharia (modelagem de negócio, requisitos, análise e projeto, implementação, teste e implantação) e três de suporte (gerência de projeto, configuração e gerência de mudanças e ambiente).

Embora papéis, atividades, artefatos e fluxos sejam os principais elementos do RUP, deve-se considerar que seu desenvolvimento é realizado em fases, sendo estes elementos importantes do processo. Uma fase representa uma parte do ciclo de vida e é constituída de iterações que são intervalos de tempo onde o sistema é construído.

2.1 Metamodelo RUP

Para entender como os elementos são estruturados no processo RUP e quais são as relações válidas entre estes elementos, em [Bencomo 2005] um metamodelo é apresentado conforme pode ser visto na figura 1.

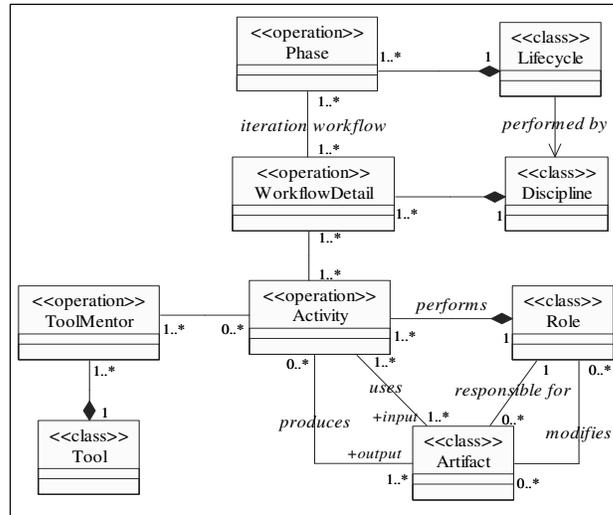


Figura 1 – Metamodelo RUP [6]

O metamodelo é representado por um diagrama de classes UML [UML 2005], sendo que os elementos do processo são representados no diagrama pelas classes. No metamodelo dois tipos de elementos são definidos: os elementos do tipo classe representados pelo estereótipo `<<class>>` e os elementos operação representados pelo estereótipo `<<operation>>`. Esta divisão entre os elementos é criada para que elementos do tipo classe descrevam os “objetos” do processo e os elementos do tipo operação descrevam o comportamento destes “objetos”. Ainda, as associações entre as classes do metamodelo são usadas para indicar quais são as relações válidas entre os elementos do processo.

3. Metamodelo Proposto

Neste trabalho está sendo proposta uma extensão ao metamodelo do RUP com a inclusão de novas classes e associações. Tal extensão tem como propósito descrever o conjunto de elementos e relacionamentos necessários para adaptação de processos de desenvolvimento de software compatíveis com o RUP, evitando assim possíveis inconsistências no processo adaptado.

Na figura 2, um diagrama de classes UML [UML 2005] foi utilizado para representar o metamodelo proposto. Na seqüência, uma descrição de suas classes e associações é apresentada fazendo a devida relação com o metamodelo RUP.

3.1 Classes do Metamodelo

As classes apresentadas no metamodelo representam os elementos que constituem um processo de software. O conjunto de atributos definidos para cada uma delas é baseado em estudos realizados em [RUP 2005] e representam as informações que necessitam ser mantidas sobre tais classes. Os principais atributos comuns às diversas classes são: *id* que mantém informações sobre a identificação das classes, *nome*, *descrição* que pode manter informações resumidas sobre as classes e o atributo *obrigatório* usado para estabelecer que as instâncias de uma classe não podem ser eliminadas no processo de adaptação.

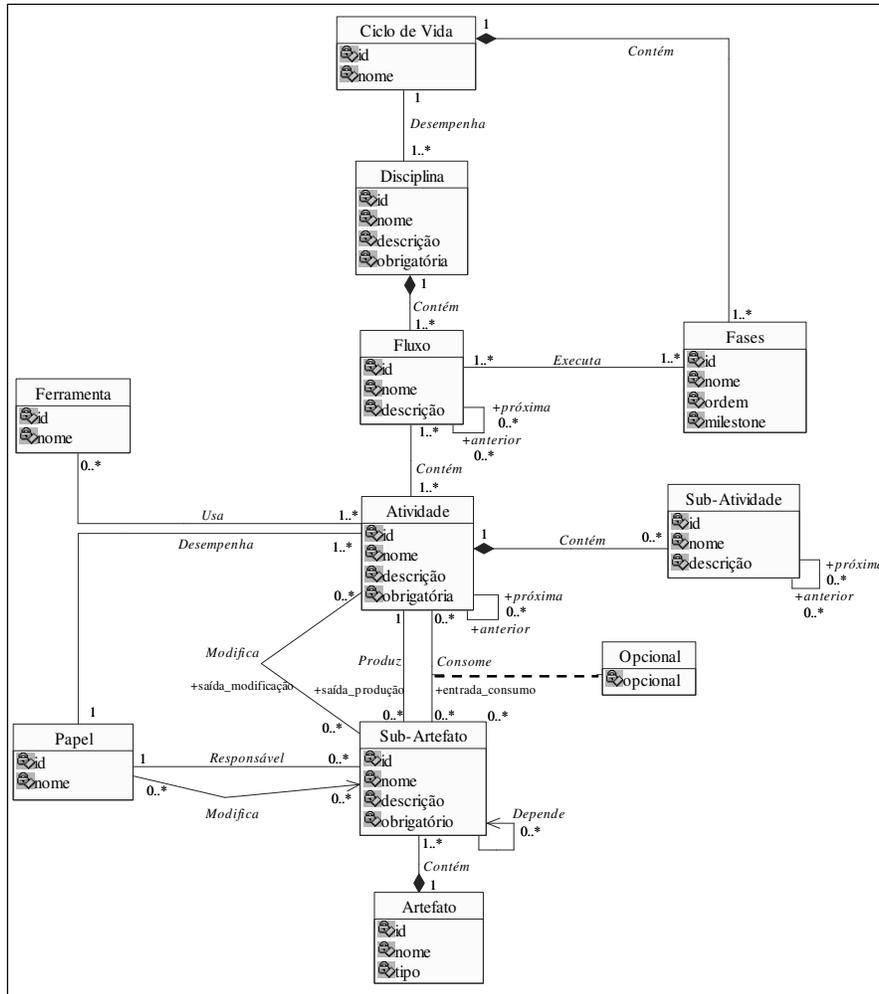


Figura 2 - Metamodelo para Adaptação de Processos

Na tabela 1 é apresentado um resumo das classes e atributos específicos que compõem o metamodelo.

Tabela 1 –Classes do Metamodelo

Ciclo de Vida (<i>Lifecycle</i> no RUP)	Define o comportamento completo de um processo em um determinado projeto.
Fase (<i>Phase</i> no RUP)	Definido como parte do ciclo de vida do processo que possui uma meta específica. A classe fase possui alguns atributos específicos que são <i>ordem</i> para definir sua seqüência de execução em um ciclo de vida e <i>milestone</i> que mantém informações sobre sua meta específica.
Disciplina (<i>Discipline</i> no RUP)	Divisão de elementos de processo em áreas de interesse.
Fluxo (<i>WorkflowDetail</i> no RUP)	Agrupamento de atividades relacionadas a uma determinada disciplina.
Atividade (<i>Activity</i> no RUP)	Unidade de trabalho que produz um resultado significativo no contexto de um projeto. Uma atividade possui um propósito claro expresso em termos de produzir ou modificar sub-artefatos, sendo que toda atividade deve ser atribuída a um papel.
Sub-Atividade (não modelado no RUP)	É parte do elemento atividade e é considerada a menor ação a ser desempenhada, ou seja, é um elemento atômico no processo.
Ferramenta (<i>Tool</i> no RUP)	Elemento que pode ser usado como auxílio para execução de algumas atividades.
Papel (<i>Role</i> no RUP)	Responsável por desempenhar atividades para produzir e/ou modificar os sub-artefatos do processo.
Sub-Artefato (não modelado no RUP)	Informação produzida, consumida ou modificada em um processo de software. Os <u>sub-artefatos</u> são definidos como partes de artefato e podem ser, por exemplo, uma seção de um documento, partes de um modelo (por

	exemplo, os atores em um modelo de casos de uso), entre outros.
Artefato (<i>Artifact</i> no RUP)	Definido com um conjunto de sub-artefatos. Pode ser dos seguintes tipos: modelo, documento, relatório, executável e código-fonte. Para armazenar informações de qual tipo é o artefato possui um atributo denominado <i>tipo</i> .
Opcional (não modelado no RUP)	Não representa elemento de processo e também não é previsto em [6]. Esta classe esta relacionada com a associação <i>Consume</i> entre Atividade e Sub-Artefato e mantém informações se um determinado sub-artefato é opcional ou não em uma atividade.

Os conceitos apresentados na tabela 1 são os mesmos apresentados em [RUP 2005], sendo que os seguintes elementos foram incorporados ao metamodelo: Sub-Atividade e Sub-Artefato.

Sub-Atividade é um elemento de processo previsto no processo RUP embora não seja modelado em seu metamodelo. Em [Kruchten 2000], tem-se que uma atividade deve ser dividida em etapas que representam suas partes. Desta forma, no metamodelo proposto criou-se a classe Sub-Atividade para que seja possível decompor uma atividade em partes.

A criação do elemento Sub-Artefato é a principal mudança incorporada ao metamodelo do RUP em termos de elementos de processo. Através deste elemento permite-se que qualquer artefato seja dividido em partes e que as atividades sejam modeladas em termos de quais destas partes específicas irão produzir, consumir e/ou modificar.

O uso de sub-artefatos traz uma importante contribuição ao processo de adaptação que é o principal interesse deste trabalho. Isto porque, torna possível que durante a exclusão de uma atividade sejam identificadas quais sub-artefatos não serão mais produzidos. Isto não é possível se o processo modela suas atividades em termos de consumo e produção de artefatos como é o caso do metamodelo RUP.

3.2 Associações do Metamodelo

Esta seção descreve as associações existentes entre as classes do metamodelo proposto (Tabela 2). Cada uma das associações é descrita devido a sua importância em processos de adaptação. Isto porque, um processo de adaptação permite que elementos sejam adicionados, excluídos ou ainda modificados a partir de um processo de software [Yoon 2001], [Jalote 2002], [Ginsberg 1995] e [Fitzgerald 2003]. Assim, é preciso respeitar as relações entre estes elementos para garantir a conformidade com o processo padrão durante a sua adaptação.

Tabela 2 – Associações do Metamodelo

Ciclo de Vida / Disciplina	As disciplinas são sempre desempenhadas durante o ciclo de vida do processo. Desta forma, tem-se uma relação de associação entre ciclo de vida e disciplina. Nesta relação, uma disciplina está sempre associada a um ciclo de vida, sendo que um ciclo de vida pode ser associado a uma ou várias disciplinas.
Ciclo de Vida / Fase	O ciclo de vida define a partição do tempo num conjunto de fases. A relação de associação entre estes elementos é do tipo composição e indica que a existência de uma fase depende da existência do ciclo de vida a que pertence.
Fase / Fluxo	A relação de associação entre fase e fluxo é estabelecida visto que os fluxos devem ser executados nas fases. Em uma fase temos a execução de pelo menos um ou mais fluxos, enquanto que um fluxo deve ser executado em pelo menos uma ou mais fases.
Disciplina / Fluxo	Uma disciplina é um conjunto de um ou vários fluxos. Desta forma, é estabelecida uma associação do tipo composição entre disciplina e fluxo. Esta associação indica que a existência de um fluxo depende da existência da disciplina a que pertence.
Fluxo / Fluxo	A relação de associação criada entre fluxos é estabelecida visto que estes são elementos sequenciais ou paralelos em uma disciplina e sendo assim apresentam relações de precedência.
Fluxo / Atividade	A relação de associação entre fluxo e atividade é estabelecida visto que as

	atividades são agrupadas em fluxos do processo. Em um fluxo devemos associar pelo menos uma ou mais atividades, enquanto que uma atividade deve ser associada em pelo menos um ou mais fluxos.
Atividade / Atividade	A relação de associação criada entre atividades é estabelecida visto que estes são elementos seqüenciais ou paralelos em um fluxo e sendo assim apresentam relações de precedência.
Atividade / Sub-atividade	Tem-se uma associação do tipo composição entre atividade e sub-atividade, pois uma atividade é um conjunto de sub-atividades. Esta associação indica que a existência de uma sub-atividade depende da existência da atividade a que pertence.
Sub-Atividade / Sub-Atividade	A relação de associação entre sub-atividades é estabelecida visto que estes são elementos seqüenciais ou paralelos em uma atividade e sendo assim apresentam relações de precedência.
Atividade / Ferramenta	A relação de associação entre atividade e ferramenta é estabelecida pois atividades usam ferramentas durante sua execução. Nesta relação, uma ferramenta deve ser associada a uma ou mais atividades, enquanto que uma atividade pode ser associada a nenhuma ou várias ferramentas.
Atividade / Papel	Atividades e papéis relacionam-se pois papéis executam atividades. No metamodelo, tem-se uma associação do tipo composição entre papel e atividade, pois um papel é um conjunto de atividades. Esta associação indica que a existência de uma atividade depende da existência do papel a que pertence.
Sub-Artefato / Atividade	Na relação de associação chamada <i>Consome</i> tem-se que uma atividade pode estar associada a nenhum ou vários sub-artefatos, enquanto que um sub-artefato associa-se a nenhuma ou várias atividades. Na relação de associação chamada <i>Produz</i> tem-se que uma atividade pode estar associada a nenhum ou vários sub-artefatos, enquanto que um sub-artefato associa-se a exatamente uma atividade. Na relação de associação chamada <i>Modifica</i> tem-se que uma atividade pode estar associada a nenhum ou vários sub-artefatos, enquanto que um sub-artefato associa-se a exatamente uma atividade.
Sub-Artefato / Papel	Na associação de responsabilidade chamada <i>Responsável</i> tem-se que um sub-artefato deve estar associado a exatamente um papel, enquanto que um papel pode ser responsável por nenhum ou vários sub-artefatos. A associação de modificação chamada <i>Modifica</i> entre sub-artefato e papel indica que um sub-artefato pode ser modificado por nenhum ou vários papéis, enquanto que um papel pode ser modificador de nenhum ou vários sub-artefatos.
Sub-Artefato / Sub-Artefato	Sub-artefatos de saída de uma atividade podem depender dos sub-artefatos de entrada desta atividade. Assim, pode-se criar uma relação de associação de um sub-artefato para outro(s). Esta relação pode ser estabelecida de sub-artefatos de um artefato para outros sub-artefatos do mesmo artefato, ou ainda, com sub-artefatos de outros artefatos.
Artefato / Sub-Artefato	Um artefato é um conjunto de um ou vários sub-artefatos. Desta forma, é estabelecida uma associação do tipo composição entre artefato e sub-artefato. Esta associação indica que a existência de um sub-artefato depende da existência do artefato a que pertence.

Em razão de o metamodelo proposto ser usado exclusivamente para adaptação de processos algumas associações precisaram ser incluídas e modificadas a partir do metamodelo RUP para expressar as devidas relações de dependência existente entre os elementos do processo. A seguir tais associações serão resumidamente descritas.

Os auto relacionamentos criados para fluxo, atividade e sub-atividade não existem no metamodelo RUP e foram definidos no metamodelo proposto pois estes elementos possuem uma determinada ordem de execução em processos de software. Esta ordem de execução precisa ser devidamente verificada durante um processo de adaptação visto que possivelmente sofrerá alterações caso um destes elementos seja excluído, adicionado ou modificado.

O auto relacionamento criado para sub-artefato foi definido visto que os sub-artefatos podem depender de outros sub-artefatos para poderem ser produzidos no processo. Em [Yoon 2001], [RUP 2005], [Kellner 1996] e [Coelho 2003] este tipo de dependência é prevista embora seja importante considerar que é tratada como uma

dependência entre artefatos do processo. A motivação para que neste trabalho tal relacionamento tenha sido criado para sub-artefato é em virtude das atividades agora serem modeladas em termos de quais sub-artefatos produzem, consomem ou modificam como já explicado na seção anterior deste trabalho.

Outras associações modificadas no metamodelo RUP foram entre as classes atividade e artefato. No RUP, artefatos são produzidos e usados por atividades e desta forma, duas associações são estabelecidas entre estas classes: as associações *uses* e *produces*. No metamodelo proposto estas associações são representadas entre as classes atividade e sub-artefato, isto porque, como explicado na seção anterior deste trabalho, o metamodelo proposto modela as atividades em termos de quais sub-artefatos devem ser produzidos, consumidos ou modificados. Ainda, uma nova associação foi estabelecida para modificação de sub-artefatos chamada *Modifica*.

A motivação para criar uma nova associação de modificação é para que durante o processo de adaptação seja possível diferenciar quando um sub-artefato esta sendo produzido ou modificado no processo.

3.3 Adaptação do Processo de Desenvolvimento de Software

Para tornar possível a adaptação de processos de desenvolvimento de software, um conjunto de regras foi desenvolvido com base em estudos realizados na literatura, resultados do estudo de caso e o metamodelo proposto. Estas regras estabelecem como os elementos e relacionamentos do processo padrão devem ser tratados durante sua adaptação, de forma a garantir consistência no processo resultante.

Considera-se neste trabalho que a adaptação do processo padrão ocorre pela exclusão e inclusão de atividades durante sua adaptação, sendo esta a prática corrente na organização estudada. Neste sentido, o conjunto de regras para tais operações é apresentado na Tabela 3.

Tabela 3 – Regras para Exclusão e Inclusão de Atividades

	Exclusão	Inclusão
Artefato	- Sub-artefatos produzidos pela atividade devem ser excluídos do artefato a que pertencem.	-
Atividade	- Atividades relacionadas (anterior e próxima) devem ser conectadas; - Atividades que consomem sub-artefatos (somente os não opcionais) excluídos pela atividade devem ser eliminadas.	- Definir atividades relacionadas (anterior e próxima).
Ciclo de Vida	-	-
Disciplina	-	-
Fase	-	-
Ferramenta	- Atividade deve ter suas relações de associação com ferramentas eliminadas.	- Se necessário, associar uma ou mais ferramentas a atividade.
Fluxo	- Atividade deve ter suas relações de associação com fluxos eliminadas.	- Associar a atividade a um ou mais fluxos.
Papel	- Atividade deve ter sua relação de associação com o papel eliminada.	- Associar a atividade a um papel.
Sub-artefato	- Sub-artefatos produzidos pela atividade devem ser excluídos juntamente com seus sub-artefatos dependentes (isto pode implicar em exclusão de outras atividades); - Sub-artefatos consumidos pela atividade devem ter sua relação de associação eliminada; - Sub-artefatos modificados pela atividade devem ter sua relação de associação eliminada.	- Se necessário, associar sub-artefatos a serem consumidos, produzidos ou modificados pela atividade;
Sub-atividade	- Sub-atividades pertencentes a atividade devem ser excluídas.	- Se necessário, sub-atividades devem ser incluídas.

Conforme se pode observar nas regras da Tabela 3, a exclusão e inclusão de

atividades impactam em muitos outros elementos de um processo durante sua adaptação. Embora todos estes impactos estejam sendo previstos, é importante considerar que a exclusão de uma atividade só pode ser realizada se a mesma não é obrigatória no processo padrão. Ainda, se algum dos sub-artefatos excluídos juntamente com a atividade é obrigatório, a exclusão da atividade é cancelada.

Atualmente, no contexto desta pesquisa está se desenvolvendo uma ferramenta para apoio automatizado a adaptação de processos. A ferramenta permite o registro de um processo padrão com base no metamodelo proposto para posterior adaptação. As operações de inclusão e exclusão de atividades estão sendo implementadas com base nas regras apresentadas na Tabela 3. Ainda, funcionalidades de publicação de processos em *websites* também serão oferecidas pela ferramenta.

4. Trabalhos Relacionados

Atualmente, vários trabalhos vêm sendo desenvolvidos pela comunidade de engenharia de software para apoiar a construção de processos de adaptação. O trabalho desenvolvido por [Yoon 2001] propõe várias operações para adaptação do processo padrão como adicionar, remover, dividir e unir atividades e artefatos, sendo que para isto algumas regras devem ser seguidas, na maioria delas dizendo respeito a novas propriedades que artefatos e atividades devem receber. De acordo com [Yoon 2001], é importante que o novo processo esteja em conformidade com o processo padrão, e é por esta razão que as regras sobre as propriedades devem ser seguidas. Contudo, ainda que [Yoon 2001] considere importantes relacionamentos para o processo de adaptação, algumas limitações são encontradas como a ausência de referência aos papéis, ferramentas, fases e outros elementos do processo padrão e falta de suporte a alguns tipos de relações, como por exemplo, relação entre partes de artefatos, atividades, ferramentas e etc.

Em [Coelho 2003], é apresentada uma proposta de configuração de processos de software a partir de processos padrão para projetos específicos chamada *PConfig*. O trabalho baseia-se na escolha dos artefatos do processo padrão (RUP) que farão parte do processo adaptado de acordo com as características do projeto. A partir desses artefatos, os papéis e atividades necessários são derivados. O que pode se constatar neste trabalho, é que o autor limita-se apenas a exclusão de artefatos do processo e não considera outros importantes elementos como fluxos, ferramentas, fases, disciplinas, entre outros. Ainda, considera somente a relação de dependência entre artefatos.

5. Conclusões

A diversidade nas características dos projetos de desenvolvimento e manutenção de software, por vezes, requer a adaptação do processo padrão de uma empresa, de modo a obter melhores resultados em termos de custo, prazo e qualidade em projetos específicos. Neste sentido, ao realizar-se a adaptação de um processo de software, faz-se necessário garantir a consistência do processo resultante, em termos da integração entre seus vários elementos (atividades, artefatos, etc.). Neste sentido, a principal contribuição deste trabalho envolve a definição de um metamodelo descrevendo o conjunto de elementos e relacionamentos necessários para adaptação de processos de desenvolvimento de software compatíveis com o RUP. Com base no metamodelo, torna-se possível a identificação das relações de dependências entre os elementos do processo, permitindo a definição de regras para garantir a consistência entre os elementos do processo adaptado.

Através das regras definidas para inclusão e exclusão de atividades torna-se possível adaptar um processo padrão mantendo a integridade do processo resultante. Isto garante a consistência entre os elementos e relacionamentos do processo resultante, garantindo a conformidade com o processo padrão, bem como evitando as falhas de execução pelas inconformidades geradas durante a adaptação do processo, conforme identificado em [Jalote 2002].

O metamodelo e regras propostas já foram avaliados utilizando-se dois estudos de caso apresentados em [Kroll 2003] e [Pollice 2003]. Os estudos sugerem atividades e artefatos para projetos de desenvolvimento com menor porte. Os resultados obtidos com os estudos se mostraram satisfatórios, já que a partir da exclusão de algumas atividades, utilizando-se o metamodelo e as regras definidas neste trabalho, obteve-se o mesmo conjunto de atividades e artefatos sugeridos em [Kroll 2003] e [Pollice 2003].

Atualmente está em desenvolvimento uma ferramenta para adaptação de processos baseada no metamodelo e nas regras para inclusão e exclusão de atividades propostas neste trabalho. Trabalhos futuros incluem a avaliação do metamodelo e regras em projetos de desenvolvimento da organização estudada. Ainda, pretende-se estender este trabalho definindo regras para outras operações de adaptação de processos, tais como inclusão e exclusão de disciplinas, fluxos, papéis, sub-artefatos, artefatos e ferramentas.

6. Referências

- Bencomo, A. (2005) "Extending the RUP, Part 1: Process Modeling", Capturado em: http://www-128.ibm.com/developerworks/rational/library/05/323_extrup1/, Janeiro 2005.
- Coelho, C. (2003) "MAPS: Um Modelo para Adaptação de Processos de Software", Dissertação de Mestrado. Universidade Federal de Pernambuco.
- Fitzgerald B., Russo N. L., O'kane T. (2003) "Software Development Method Tailoring at Motorola", In: Communications of the ACM, April, pp.65-70.
- Ginsberg, M. P., Quinn, L. H. (1995) "Process Tailoring and the Software Capability Maturity Model." Technical Report, November.
- Jacobson, I., Booch G., Rumbaugh J. (2001) "The Unified Software Development Process", Upper Saddle River, Addison Wesley.
- Jalote, P. (2002) "CMM in Practice. Processes for Executing Software Projects at Infosys", The SEI Series in Software Engineering.
- Kellner, M. I. (1996) "Connecting reusable software process elements and components", In Proc. International Software Process Workshop, pp.8-11.
- Kroll, P., Kruchten, P. (2003) "The Rational Unified Process Made Easy: A Practitioner's Guide to the Rup", Addison-Wesley, Boston.
- Kruchten, P. (2000) "The Rational Unified Process: An Introduction", Upper Saddle River, NJ: Addison-Wesley.
- Pollice G. (2003) "Using o RUP for small projects: Expanding upon Extreme Programming", capturado em: <http://www-128.ibm.com/developerworks/rational/library/409.html>, Agosto.
- RUP (2005) "Rational Unified Process Evaluation Assembly V2003.06.13 for Windows", capturado em: http://www14.software.ibm.com/webapp/download/product.jsp?id=JGRM-5R2KER&s=z&cat=&S_TACT=104AH+W42&S_CMP, Julho.
- UML (2005) "Unified Modeling Language Specification", Capturado em: <http://www.uml.org/>, Julho.
- Yoon, I., Min, S., Bae, D. (2001) "Tailoring and Verifying Software Process", In: Institute of Electrical and Electronic Engineers - IEEE, pp.202-209.