

Catálogo de Padrões de Definição de Processos de Software

Luiz Carlos M. Ribeiro, Kamilla H. Crozara, Hilmer R. Neri
André Luiz P. M. Lanna, Cristiane S. Ramos, Rejane M. C. Figueiredo

¹Faculdade Gama, Campus Gama - Universidade de Brasília (UnB).

{lcarlos, hilmer, andrelanna, cristianesramos, rejanecosta}@unb.br,
holanda.kamilla@gmail.com

Abstract. *This work presents the specification of a catalog of patterns for software processes definition, which consists on a set of generic and reusable solutions that could be applied on classes of known and recurrent problems. The catalog was defined on a context of an architecture of reusable software processes components based on Software & Systems Process Engineering Metamodel specification (SPEM). Additionally, examples of its use are presented.*

Resumo. *Neste trabalho é apresentada a especificação de um catálogo de padrões de definição de processos de software (software process definition patterns), que consiste em um conjunto de soluções genéricas e reutilizáveis que podem ser aplicadas em classes de problemas conhecidos e recorrentes. O catálogo foi definido no contexto de uma arquitetura de componentes de processos de software reutilizáveis, baseada no meta-modelo Software & Systems Process Engineering Meta-Model Specification (SPEM). Adicionalmente, exemplos de uso do catálogo são apresentados.*

1. Introdução

Uma Arquitetura de Processos pode ser conceituada como um conjunto de visões representadas por modelos de processos. Esses modelos representam uma estrutura de processos e dos seus componentes em termos da sua composição, comportamento e relacionamentos, de acordo com um domínio e um contexto [Borsoi 2008]. O conjunto de visões determina os níveis de processos de software definidos, sendo que, desde o processo de referência até o nível de instanciação, os componentes são descritos em diferentes níveis de abstração. Sendo assim, em uma abordagem *top down*, quanto mais baixo o nível, mais específico será o seu conteúdo. Percebe-se então que a organização em níveis da Arquitetura de Processos permite o reúso de todo o conteúdo de um processo padrão ou de referência.

Algumas iniciativas vêm sendo conduzidas a fim de contribuir com o avanço da reutilização de processos de software na indústria [Barreto et al. 2011, Carvalho et al. 2011, Aleixo et al. 2010, Lanna e Pietrobon 2010]. A maioria desses trabalhos utilizam técnicas e processos de reutilização de software comuns a fim de derivar métodos para reutilização de processos de software [Ribeiro et al. 2011b]. No entanto, para Fiorini (2001) uma arquitetura de processos de software facilita o acesso e o reaproveitamento dos processos. Nesse contexto, o grupo de Engenharia de Software da Faculdade Gama da Universidade de Brasília vem trabalhando em uma solução de arquitetura de processos proposta em [Ribeiro et al. 2011b] e [Ribeiro et al. 2011a]. A Arquitetura é composta por três componentes (figura 1): (i) Tipos de Processos, que consistem em três camadas: Modelos

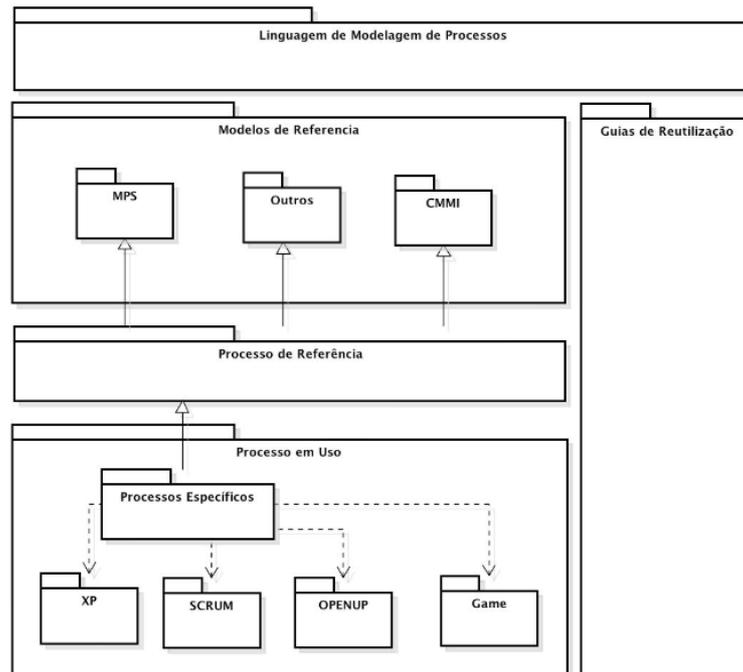


Figura 1. Arquitetura de Processos de Software

de Referência, Processos de Referência e, Processos em Uso; (ii) Linguagem de Modelagem, uma notação gráfica baseada em UML, definida pelo meta-modelo SPEM; e (iii) Guias de reutilização, que são diretrizes de como desenvolver processos **para** reúso e como desenvolver processos **com** reúso.

Os Modelos de Referência agregam solução aos problemas do ponto de vista do domínio e carregam um conjunto de conhecimento bastante amadurecido. O Processo de Referência deve carregar os elementos mínimos de um processo, tais como: tarefas; entradas; saídas; papéis; responsabilidades; habilidades necessárias e medidas de qualidade de processo e de produto, em conformidade com os modelos de referência adotados. Um processo de referência pode também conter um *framework* de processos já consolidado e testado, pronto para ser reutilizado e instanciado quase que integralmente.

Os Processos em Uso, por sua vez, são as instâncias de processos que são compostas por processos específicos, que podem possuir dependência com outros processos ou práticas, tais como: SCRUM [Schwaber e Sutherland 2011] e XP [Wells 2011].

No entanto, para que seja possível a implementação do conceito da Arquitetura de Processos, é necessária a utilização de um mecanismo que permita a descrição dos processos e a variação, extensão e reutilização do conteúdo do Processo de Referência para a criação das instâncias de processos, ou seja, o Processo em Uso para cada situação específica.

Visando esse objetivo, a Arquitetura de Processos foi baseada no meta-modelo Software & Systems Process Engineering Meta-Model Specification (SPEM) [OMG 2008], que define os conceitos necessários para modelar, documentar, apresentar, gerenciar e representar métodos e processos de desenvolvimento. O SPEM descreve um mecanismo

denominado *Variability Element* que é uma classe abstrata que provê a capacidade de variação e extensão de uma lista específica de classes do SPEM 2.0 [OMG 2008]. Aplicando-se tais conceitos à Arquitetura de Processos definida, pode-se verificar a possibilidade de variação, extensão e reutilização dos vários elementos descritos no Processo Padrão para a criação do Processo em Uso.

Entretanto, tais mecanismos possuem comportamentos distintos que dependem do tipo de elemento de processo no qual são aplicados (ex. Tarefa, Papel, Produto de Trabalho, Guias, entre outros). Nesse contexto, é necessário mapear tais comportamentos para que o reuso seja realmente efetivo e evitar o procedimento “copia-cola” ao gerar as instâncias de Processo em Uso. Ao identificar situações comuns de reuso de elementos de processos, foi possível estabelecer classes de problemas genéricos e, conseqüentemente, soluções genéricas e reutilizáveis denominadas padrões de definição de processos de software. Um exemplo de classe de problema seria reutilizar uma Tarefa do processo padrão, manter sua descrição, produtos de entrada e saída, recursos e ferramentas associados, porém alterar ou substituir o papel executor da referida tarefa.

Dado o contexto, apresenta-se, neste trabalho, um catálogo de Padrões de Definição de Processos, que descrevem os problemas comuns do uso da variabilidade aos elementos do Processo em Uso e uma solução, definindo assim um método consistente para a resolução dos principais problemas identificados durante a descrição de cada novo Processo em Uso.

O trabalho está organizado em seções. Na Seção 2 são apresentados conceitos de padrões de software e a motivação para o desenvolvimento do trabalho em questão. Na Seção 3 apresenta-se um catálogo com a proposta de padrões de definição de processos de software. Finalizando, nas Seções 4 e 5 são apresentados exemplos de uso do catálogo e a conclusão, respectivamente.

2. Padrões de Software

Durante a descrição de um Processo em Uso, é possível aplicar conceitos de variabilidade nos elementos que serão reutilizados a partir do Processo Padrão. Cada tipo de variabilidade altera os elementos do processo de forma diferente: se por um lado a variabilidade do tipo <<extends>> funciona como uma herança das características de outros elementos, por outro lado a variabilidade do tipo <<extends and replaces>> afeta ambos os elementos envolvidos. Por exemplo: dados o elemento Base e o elemento EmUso, o elemento EmUso aplica a variabilidade <<extends and replaces>> em relação ao elemento Base; assim, o elemento EmUso herda as características do elemento Base e este, por sua vez, tem as suas características substituídas pelas características criadas/alteradas no elemento EmUso, o que difere do conceito de herança tradicional do paradigma Orientado a Objetos. Existem ainda os tipos de variabilidade <<replaces>> e <<contributes>>, nos quais o primeiro substitui completamente, enquanto o segundo complementa as características de um elemento.

Para reutilizar os elementos do processo de referência é preciso aplicar os mecanismos de variabilidade. Contudo, tais mecanismos podem apresentar-se de forma confusa devido aos vários tipos de alterações que o seu uso implica, como por exemplo a diversidade de alterações de características (descrição, passos, produtos e papéis associados, e outros) que um elemento pode sofrer apesar de ser usado o mesmo tipo de

variabilidade.

Devido ao comportamento distinto da relação tipo de variabilidade e elemento de processo, a definição de processos de software com reúso gera um conjunto de problemas que se repetem durante a descrição de cada novo Processo em Uso. Tais problemas geralmente possuem o mesmo tipo de solução que deve ser aplicada aos diversos contextos de descrição do Processo em Uso.

Nesse sentido, padrões de software vêm sendo definidos ao longo do tempo e utilizados para resolver os mais variados problemas recorrentes durante o desenvolvimento de software. Tal conceito foi adaptado das ideias de Alexander et. al (1977) que define um padrão como sendo a descrição de um problema e o cerne de sua solução, de modo que é possível usar essa solução diversas vezes, sem nunca fazê-lo da mesma maneira. Portanto, padrões de software não são soluções fixas e estáveis, mas descrições de problemas e suas soluções, adaptáveis a cada caso e aplicáveis em diferentes níveis de abstração.

Pesquisas em padrões de software vêm sendo realizadas desde 1980, mas foi na década de 1990 que o tema começou a ser difundido a partir da publicação de *Design Patterns*, um catálogo de 23 padrões voltados para o projeto de software [Gamma et al. 1995]. O catálogo apresenta, para cada padrão, o seu nome, o problema em que ele pode ser utilizado, sua solução e uma análise de sua utilização de modo a facilitar a tomada de decisões do desenvolvedor. A partir da publicação desse catálogo vários outros padrões de software foram propostos, tais como padrões de análise e padrões de requisitos.

O termo Padrão de Processo foi definido por Coplien (1995) ao afirmar que padrões de atividades dentro de uma organização e seus projetos são chamados de processos. Em um contexto mais específico, Ambler (1998) define Padrões de Processos de software como “*uma coleção de técnicas gerais, ações e/ou tarefas (atividades) para o desenvolvimento de software orientado a objetos*”. Segundo a abordagem de Ambler, um padrão de processo descreve o que deve ser feito, mas não detalha como devem ser feitas as atividades do processo, sendo assim os padrões podem ser usados como blocos que serão utilizados para definir um processo de software que atenda necessidades específicas de determinada organização.

Neste ponto é importante ressaltar a diferença existente entre Padrões de Processo e Padrões de Definição de Processos (tema central deste trabalho). Enquanto Ambler (1998) trata da definição de cada um dos elementos do processo de desenvolvimento de software (como atividades, tarefas, ferramentas, produtos de trabalho e papéis), os Padrões de Definição de Processos descrevem soluções para problemas recorrentes que ocorrem no desenvolvimento de processos de software **com** reúso.

Para a descrição dos Padrões de Definição de Processos foi usado como base o modelo de descrição de Padrões de Processos utilizado por Ambler (1998) (Tabela 1).

3. Padrões de Definição de Processos de Software

A Tabela 2 apresenta o catálogo dos Padrões de Definição de Processos. A coluna Variabilidade apresenta os tipos de variabilidade aplicados a cada padrão. Os padrões são categorizados por tipos: **Descrição**: padrões de reúso de elementos de processo com alterações na descrição de elementos; **Passos**: padrões de reúso de elementos de processo do tipo Tarefa com alterações em passos; **Recursos e Ferramentas**: padrões de reúso de

Tabela 1. Modelo para a descrição dos Padrões de Definição de Processos. Adaptado de [Ambler 1998].

Nome	Ao padrão é dado um nome significativo que descreva a sua função dentro do processo de software.
Intenção	O objetivo do padrão é descrito brevemente.
Tipo	O tipo de padrão é especificado.
Variabilidade	Os tipos de variabilidades utilizados no padrão são especificados.
Contexto Inicial	As condições sobre as quais o padrão se aplica são descritas.
Problema	O problema a ser resolvido pelo padrão é descrito.
Solução	A implementação do padrão é descrita, passo a passo, de forma a apresentar a solução proposta.
Estrutura	A estrutura do padrão é apresentada.
Contexto Resultante	As condições que resultarão quando o padrão tiver sido implementado com sucesso são descritas.

elementos de processo do tipo Tarefa com alterações de papéis, produtos de trabalho de entrada e saída e guias; **Geral**: padrões de reuso aplicáveis a qualquer tipo de elemento de processo.

Tabela 2. Catálogo de Padrões de Definição de Processos.

N.	Padrão	Variabilidade	Tipo
01	Complementar a descrição	Contributes / Extends	Descrição
02	Substituir a descrição	Extends and replaces / Extends	Descrição
03	Herdar características	Extends	Descrição / Recursos e Ferramentas
04	Substituir recursos e ferramentas	Extends and replaces / Extends	Recursos e Ferramentas
05	Adicionar passos ao início	Extends	Passos
06	Adicionar passos ao final	Contributes / Extends	Passos
07	Substituir passos	Extends and replaces / Extends	Passos
08	Substituir elementos	Replaces	Geral
09	Combinar elementos	Contributes	Geral

As próximas subseções apresentam o detalhamento de cada padrão de definição de processos do catálogo.

3.1. Complementar a descrição

Intenção: Complementar informações nos campos da aba “Descrição” de elementos do Processo em Uso quando esse reutiliza um elemento do Processo de Referência.

Tipo: Descrição.

Variabilidade: *Contributes / Extends*.

Contexto Inicial: um elemento do Processo de Referência está definido e será reutilizado pelo Processo em Uso, mas é necessário que campos da aba “Descrição” do elemento do Processo de Referência sejam complementados.

Problema: a descrição do elemento do Processo de Referência a ser reutilizada não deve sofrer alterações.

Solução: (1) Deve ser criado, no Processo em Uso, um meta elemento com o complemento desejado para os campos da aba “Descrição”; (2) O meta elemento deve aplicar a variabilidade <<contributes>> em relação ao elemento do Processo de Referência; (3) Criar o elemento em uso definitivo, mantendo em branco os campos da Descrição que devem ser complementados e (4) O novo elemento do Processo em Uso deve aplicar a variabilidade <<extends>> em relação ao elemento do Processo de Referência.

Estrutura:

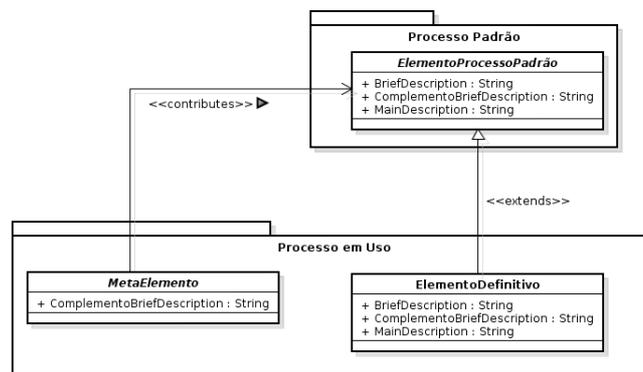


Figura 2. Complementar a descrição.

Contexto Resultante: os campos do elemento do Processo de Referência serão complementados com a descrição dos campos do meta elemento. Os campos do elemento definitivo do Processo em Uso apresentarão as descrições do meta elemento do Processo em Uso e do elemento do Processo de Referência.

3.2. Substituir a descrição

Intenção: substituição da descrição de elementos do Processo em Uso ao reutilizar elementos do Processo de Referência.

Tipo: Descrição.

Variabilidade: *Extends and Replaces/Extends*.

Contexto Inicial: um elemento do Processo de Referência está totalmente definido e será reutilizado, mas é necessário que a descrição deste elemento seja substituída no elemento do Processo em Uso.

Problema: o elemento do Processo de Referência não pode ser alterado ou substituído, mas o elemento resultante no Processo em Uso deve conter uma descrição diferente do elemento reutilizado.

Solução: (1) Deve ser criado no processo em uso um meta elemento com a descrição que deverá substituir a descrição do elemento reutilizável; (2) O meta elemento criado deve aplicar a variabilidade <<extends and replaces>> em relação ao elemento do processo de referência; e (3) O novo elemento do processo em uso deve aplicar a variabilidade <<extends>> em relação ao elemento reutilizável do processo de referência.

Estrutura:

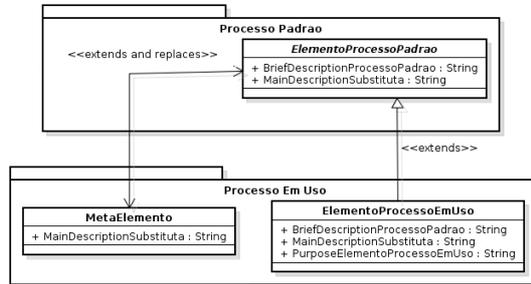


Figura 3. Substituir a descrição de elementos.

Contexto Resultante: a descrição do elemento do Processo em Uso será substituída em relação ao elemento do Processo de Referência reutilizado. No entanto, as demais características, recursos e ferramentas não substituídas no Elemento reutilizado pelo meta elemento serão mantidas no Elemento do Processo em Uso.

3.3. Herdar características

Intenção: Herdar descrição, recursos e ferramentas de elementos do Processo de Referência.

Tipo: Descrição / Recursos e Ferramentas

Variabilidade: *Extends*.

Contexto Inicial: é necessário que alguns dos campos da Descrição de um elemento reutilizável sejam adicionados à descrição de um novo elemento do Processo em Uso, ou que alguns dos recursos e ferramentas associados a elementos do Processo de Referência sejam herdados por elementos do Processo em Uso.

Problema: ao reutilizar um elemento do Processo de Referência é preciso que as suas características sejam herdadas sem que sejam feitas cópias de tais características.

Solução: a variabilidade <<extends>> deve ser aplicada no elemento do processo em uso em relação ao elemento do processo padrão. (1) Criar um Elemento no processo em uso; e (2) aplicar a variabilidade <<extends>> no elemento do Processo em Uso em relação ao elemento do Processo de Referência.

Estrutura:

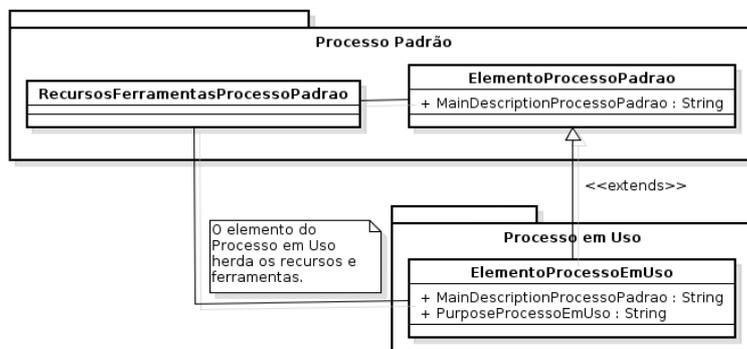


Figura 4. Herdar características do Processo de Referência.

Contexto Resultante: os campos da Descrição do elemento do Processo em Uso que não foram definidos herdarão a descrição dos campos da Descrição do elemento reutilizado do processo de referência. Os elementos do Processo em Uso possuirão os mesmos recursos e ferramentas do elemento do Processo de Referência. Os recursos e ferramentas herdados aparecerão em conjunto com os recursos e ferramentas que já haviam sido associados ao elemento do Processo em Uso.

3.4. Substituir recursos e ferramentas

Intenção: substituir recursos e ferramentas associadas a uma tarefa do Processo em Uso ao reutilizar uma tarefa do Processo de Referência.

Tipo: Recursos e Ferramentas.

Variabilidade: *Extends and Replaces / Extends*.

Contexto Inicial: uma tarefa do Processo de Referência está totalmente definida e será reutilizada, mas é necessário que recursos e ferramentas associadas a esta tarefa sejam substituídas na tarefa do processo em uso.

Problema: a tarefa do processo de referência não pode sofrer alterações em seus recursos e ferramentas. No entanto, a tarefa resultante no processo em uso deve conter recursos e ferramentas diferentes dos contidos no processo de referência.

Solução: (1) Deve ser criado, no Processo em Uso, o recurso ou ferramenta que deve substituir recursos ou ferramentas do Processo Padrão; (2) O novo elemento do Processo em Uso deve aplicar a variabilidade *<<extends and replaces>>* em relação ao elemento do Processo de Referência que se deseja substituir; e (3) A nova tarefa do Processo em Uso deve aplicar a variabilidade *<<extends>>* em relação à tarefa Processo de Referência.

Estrutura:

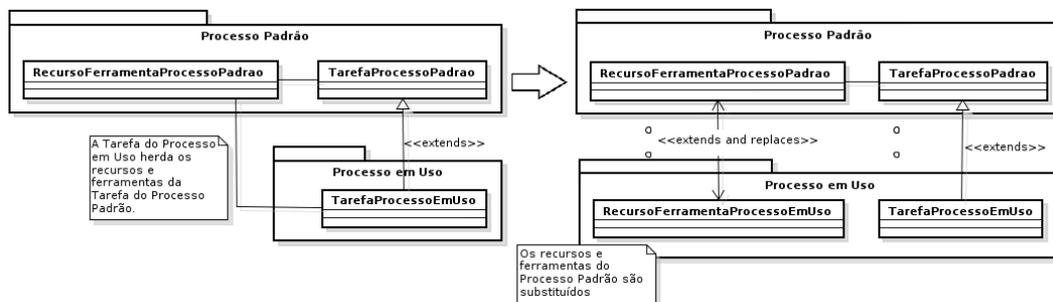


Figura 5. Substituir recursos e ferramentas.

Contexto Resultante: os recursos e ferramentas associados à tarefa do Processo em Uso são substituídos em relação à tarefa reutilizada. No entanto, as demais características, recursos e ferramentas não estendidos são mantidos na tarefa do Processo em Uso.

3.5. Adicionar passos ao início

Intenção: adição de passos iniciais à tarefa do processo em uso.

Tipo: Passos.

Variabilidade: *Extends*.

Contexto Inicial: os passos da tarefa do Processo de Referência estão definidos, no entanto, no processo em uso, é necessário que a tarefa contenha novos passos herdados da

tarefa reutilizada.

Problema: é preciso adicionar os passos da tarefa do processo padrão ao processo em uso e evitar a simples cópia de tais passos na tarefa do processo em uso.

Solução: (1) Criar uma tarefa no processo em uso e inserir os passos adicionais; e (2) aplicar a variabilidade <<extends>> na tarefa do processo em uso em relação à tarefa do processo de referência.

Estrutura:

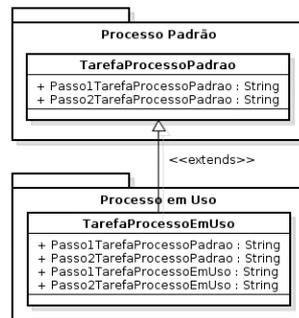


Figura 6. Adição de passos a uma tarefa do Processo em Uso.

Contexto Resultante: os passos da tarefa reutilizada são adicionados pela nova tarefa. A nova tarefa conterá os passos da tarefa reutilizada no início da lista, e os passos originais da tarefa em uso serão mantidos ao final da lista de passos.

3.6. Adicionar passos ao final

Intenção: adicionar passos finais a uma tarefa do Processo em Uso ao reutilizar uma tarefa do Processo de Referência.

Tipo: Passos.

Variabilidade: *Contributes / Extends*.

Contexto Inicial: uma tarefa do Processo de Referência está definida e será reutilizada, mas é necessário que sejam adicionados passos à tarefa do Processo em Uso.

Problema: a tarefa do Processo de Referência não pode sofrer alterações em seus passos. No entanto, a tarefa resultante no Processo em Uso deve conter passos adicionais ao final do conjunto de passos originais.

Solução: (1) Deve ser criado, no processo em uso, uma meta tarefa com os novos passos a serem adicionados; (2) A meta tarefa deve aplicar a variabilidade <<contributes>> em relação à meta tarefa do processo de referência que será reutilizada; (3) Criar a tarefa em uso definitiva; e (4) A nova tarefa do processo em uso deve aplicar a variabilidade <<extends>> em relação à tarefa Processo de Referência.

Contexto Resultante: os passos da tarefa do Processo de Referência serão acrescidos dos passos da meta tarefa, sem, no entanto, alterar a configuração da tarefa do Processo de Referência. A tarefa do Processo em Uso apresentará os passos da meta tarefa, da tarefa do Processo de Referência e também os passos que foram definidos na própria tarefa definitiva.

Estrutura:

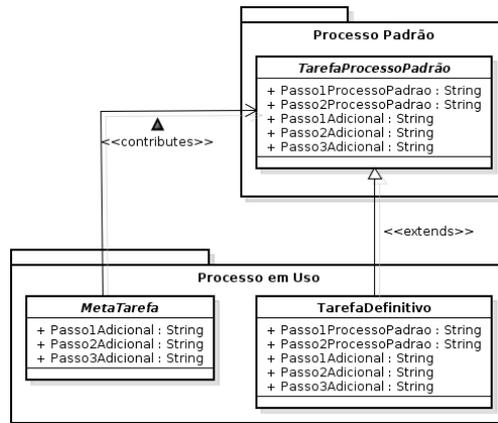


Figura 7. Adição de passos ao final de uma tarefa do Processo em Uso.

3.7. Substituir passos

Intenção: substituir os passos de uma tarefa do Processo em Uso ao reutilizar uma tarefa do Processo de Referência.

Tipo: Passos.

Variabilidade: *Extends and Replaces/Extends*.

Contexto Inicial: uma tarefa do Processo de Referência está totalmente definida e será reutilizada, mas é necessário que os passos desta tarefa sejam substituídas na tarefa do Processo em Uso.

Problema: a tarefa do Processo de Referência não pode sofrer alterações em seus passos. No entanto, a tarefa resultante no Processo em Uso deve conter passos diferentes dos contidos no Processo de Referência.

Solução: (1) deve ser criado, no Processo em Uso, uma meta tarefa com os passos que deverão substituir os passos da tarefa do processo de referência; (2) a meta tarefa criada deve aplicar a variabilidade <<extends and replaces>> em relação à tarefa do processo de referência que deve ter seus passos substituídos; e (3) a nova tarefa do Processo em Uso deve aplicar a variabilidade <<extends>> em relação à tarefa processo de referência.

Estrutura:

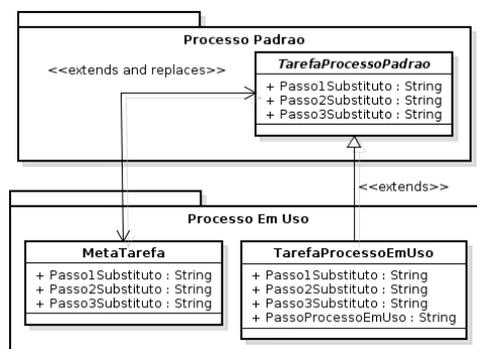


Figura 8. Substituir passos de tarefas no Processo em Uso.

Contexto Resultante: os passos da tarefa do Processo em Uso são substituídos em

relação à tarefa do Processo de Referência reutilizada. No entanto, as demais características, recursos e ferramentas não substituídas na Tarefa reutilizada pela meta tarefa, são mantidas na tarefa do Processo em Uso.

3.8. Substituir elementos

Intenção: substituir, de maneira temporária, totalmente um elemento do Processo em Uso.

Tipo: Geral.

Variabilidade: *Replaces*.

Contexto Inicial: um elemento do Processo em Uso está definido, mas é necessário substituir, temporariamente, um de seus elementos.

Problema: por ser uma substituição temporária, o elemento do Processo em Uso a ser substituído não pode ser excluído ou mesmo alterado.

Solução: (1) criar um novo elemento, de mesmo tipo, que irá realizar a substituição temporária; e (2) aplicar a variabilidade *<<replaces>>*, ao elemento que foi criado, em relação ao elemento do Processo em Uso que se deseja substituir.

Estrutura:

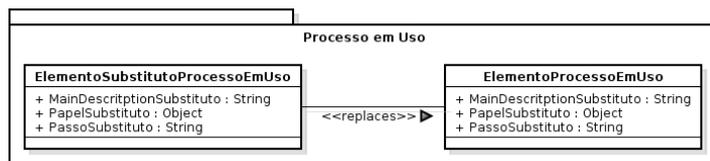


Figura 9. Substituir temporariamente os elementos.

Contexto Resultante: todos os campos da descrição (exceto o *Presentation Name*, que será mantido o original), passos, recursos e ferramentas serão substituídos pelos campos do novo elemento.

3.9. Combinar elementos

Intenção: combinar dois ou mais elementos do Processo em Uso em somente um novo elemento do Processo em Uso.

Tipo: Geral.

Variabilidade: *Contributes*.

Contexto Inicial: o Processo em Uso está definido, mas é necessário que um novo elemento seja criado e possua características de dois ou mais elementos do Processo em Uso.

Problema: deve ser criado um novo elemento com a combinação de várias características de outros elementos do Processo em Uso e deve-se evitar a simples cópia de tais características.

Solução: (1) Criar o novo elemento que irá possuir a combinação das características; (2) Aplicar a variabilidade *<<contributes>>*, aos elementos do Processo em Uso - que possuem as características que devem ser combinadas - em relação ao novo elemento do Processo em Uso.

Contexto Resultante: as características de todos os elementos do Processo em Uso serão combinadas no novo elemento do Processo em Uso e inclusive as características do novo elemento serão mantidas.

Estrutura:

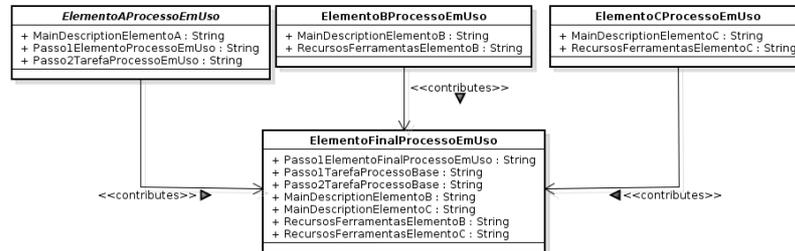


Figura 10. Combinar elementos no Processo em Uso.

4. Exemplos de Uso do Catálogo

Os exemplos de aplicação de Padrões de Definição de Processos em instâncias de Processos em Uso apresentados neste trabalho utilizam-se do *plug-in* do SCRUM [Eclipse 2012]. Este *plug-in* é representado, à luz da visão de Arquitetura de Processos [Ribeiro et al. 2011b], como a instância da camada de Processo de Padrão, que por sua vez, referencia os resultados esperados da área de processo de Gerência de Projetos (GPR), do nível G, descritas no MR-MPS [SOFTEX 2009]. Este, por sua vez, ilustra a instância da camada de Modelos de Referência.

A seguir são apresentados exemplos de instanciação de elementos de um Processo em Uso por meio do desenvolvimento com reúso de um Processo de Referência baseado no processo ágil de gestão de projetos SCRUM.

4.1. Aplicação do Padrão “Adicionar passos ao final”

No *plug-in* do SCRUM existe uma tarefa definida, “Estimar o Escopo do Projeto”, referenciada no exemplo, na camada do Processo de Referência. Para a utilização desta tarefa no Processo em Uso, fez-se necessário complementá-la com os demais elementos de processo necessários, como por exemplo, passos que complementam os passos definidos no Processo de Referência. Caso não existam passos definidos na tarefa do Processo de Referência, então o conjunto de passos será definido pelo domínio da tarefa do Processo em Uso.

A técnica utilizada para estimar projetos que utilizam o SCRUM e o XP é conhecida como Pontos por Histórias (story point) de usuários. Neste exemplo foi utilizada a dinâmica do jogo do planejamento para obterem-se as estimativas. Sendo assim, foi criada a meta tarefa “Meta-tarefa Estimando o Product Backlog” e a ela foram adicionados os seguintes passos:

- o baralho do Planning Poker é numerado;
- o moderador lê a descrição para cada história ou tema;
- a equipe deve selecionar a história que julgar mais simples e atribuir a estimativa de menor valor;
- o dono do produto deve responder a todas as questões dos jogadores;

- cada jogador deve escolher uma carta que represente sua estimativa;
- o grupo deve discutir a história por alguns minutos,
- após a discussão, cada jogador deve escolher uma carta que representa outra estimativa.

No Processo em Uso foi criada a tarefa “Estimar Product Backlog”, aplicando-se o padrão “Adicionar passos ao final”. A tarefa reutilizada não sofre alterações em quaisquer de seus elementos. A aplicação deste padrão é ilustrada na Figura 11.

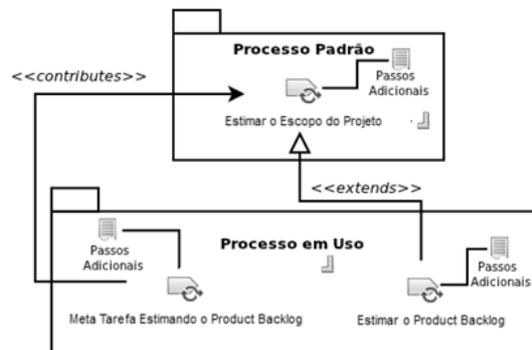


Figura 11. Aplicação do padrão “Adicionar passos ao final”

4.2. Aplicação do Padrão “Substituir Recursos e Ferramentas”

No Processo de Referência existe a tarefa “Estabelecer Estimativas de Atributos de Produto de Trabalho e Tarefas”, e a esta, existe um guia associado que descreve como se realizar a estimativa de esforço, derivada a partir do tamanho funcional, em pontos de função. Considerando que o Processo em Uso foi definido baseado no SCRUM, fez-se necessário promover a alteração do guia associado a tarefa do Processo de Referência. Para tanto, foi criada a meta tarefa “Meta tarefa Adicionar Guidance”, e a esta, foi adicionado um novo guia que descreve como é realizado o cálculo da capacidade produtiva da equipe, de forma a se identificar o velocity.

A Figura 12 apresenta a aplicação do padrão “Substituir Recursos e Ferramentas”, a tarefa “Estimar Product Backlog”, destacando o reúso das características da tarefa do Processo de Referência, exceto o guia a ela associado, que foi substituído no Processo em Uso.

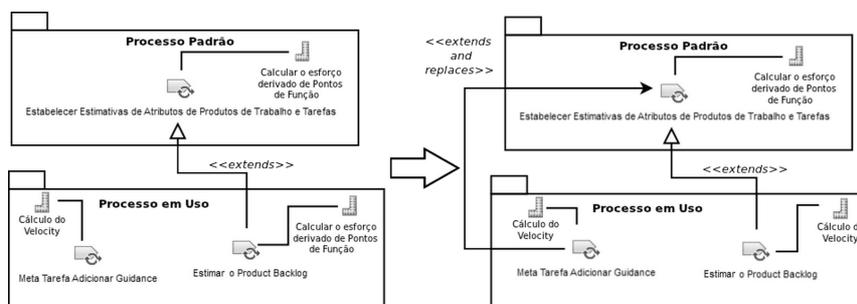


Figura 12. Aplicação do Padrão Substituir Recursos e Ferramentas

4.3. Aplicação do Padrão “Combinar Elementos”

Para definição da tarefa “Estimar o Product Backlog” no Processo em Uso, é necessário combinar características de três tarefas diferentes representadas na camada do Processo de Referência. São elas: Determinar Estimativas de Esforço e Custo, Estimar o Escopo do Projeto e Estabelecer Estimativas de Atributos de Produto de Trabalho e Tarefas.

Para tanto, foi aplicada a variabilidade <<contributes>>, a partir das três tarefas do Processo Padrão, em relação à tarefa Estimar o Product Backlog, sendo que, dessa forma esta tarefa concreta foi composta a partir da combinação das características das outras três tarefas.

A Figura 13 apresenta o exemplo da utilização do padrão Combinar Elementos.

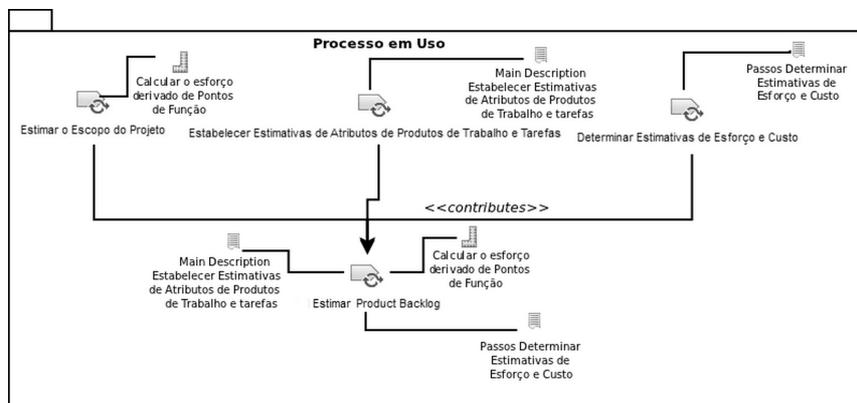


Figura 13. Aplicação do Padrão Combinar Elementos

5. Conclusões

Neste artigo foi apresentada a proposta de um catálogo de Padrões de Definição de Processos de Software como parte integrante de uma Arquitetura de Processos de Software.

Recomenda-se a utilização do catálogo como guia de desenvolvimento de processos **com** reúso de componentes previamente definidos em um Processo de Software de Referência no contexto da Arquitetura de Processos. O objetivo foi mapear os principais comportamentos resultantes da aplicação dos tipos de variabilidade fornecidos pelo meta-modelo SPEM [OMG 2008] e estabelecer soluções prévias e elegantes para promover o reúso efetivo de elementos de processos de software.

Como próximos passos, além da evolução do próprio catálogo apresentado, serão desenvolvidas diretrizes para orientação de desenvolvimento de processos **para** reúso, bem como mecanismos de busca de elementos do Processo de Referência, como forma de completar a implementação da Arquitetura de Processos proposta.

Referências

Aleixo, F. A., Freire, M. A., dos Santos, W. C., e Kulesza, U. (2010). An approach to manage and customize variability in software processes. *Simpósio Brasileiro de Engenharia de Software*.

- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., e Angel, S. (1977). *A pattern language. Towns, buildings, construction*. Oxford University Press.
- Ambler, S. W. (1998). *Process patterns: building large-scale systems using object technology*. Cambridge University Press, New York, NY, USA.
- Barreto, A. S., Murta, L. G. P., e da Rocha, A. R. C. (2011). Software process definition: a reuse-based approach. *Journal of Universal Computer Science*, 17:1765–1799.
- Borsoi, B. T. (2008). *Arquitetura de processo aplicada na integração de fábricas de software*. PhD thesis, Escola Politécnica USP, São Paulo.
- Carvalho, D. D., da Costa, A. J. S., de Oliveira Sales, E., Lima, A. M., e Reis, R. Q. (2011). Apoio à reutilização de processos de software em um ambiente de engenharia de software centrado em processo. In *Simpósio Brasileiro de Qualidade de Software*, Curitiba.
- Coplien, J. O. (1995). Pattern languages of program design. chapter A generative development-process pattern language. ACM Press/Addison-Wesley, New York, USA.
- Eclipse (2012). *Eclipse Process Framework Project (EPF): SCRUM*. Eclipse Foundation. <http://epf.eclipse.org/wikis/scrum/>.
- Fiorini, S. T. (2001). *Arquitetura para Reutilização de Processos de Software*. PhD thesis, PUC-RJ.
- Gamma, E., Helm, R., Johnson, R., e Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, Boston, USA.
- Lanna, A. L. P. M. e Pietrobon, C. A. M. (2010). Reuso de processos de software baseado na componentização de processos e conhecimento. In *Concurso de Teses e Dissertações em Qualidade de Software*, Belem.
- OMG (2008). Software & systems process engineering meta-model specification. *Process Engineering*, (April):236.
- Ribeiro, L. C. M., Ramos, C. S., Brito, M. F., e Figueiredo, R. M. C. (2011a). Definição de um processo de engenharia de requisitos para software embarcado na indústria automotiva baseada em uma arquitetura de processos de software. In *Workshop Anual do MPS*, Campinas.
- Ribeiro, L. C. M., Ramos, C. S., Crozara, K. H., Neri, H. R., e Figueiredo, R. M. C. (2011b). Definição de processos de software baseada em uma arquitetura de componentes de processo. In *X Simpósio Brasileiro de Qualidade de Software (SBQS)*.
- Schwaber, K. e Sutherland, J. (2011). The scrum guide. http://www.scrum.org/storage/scrumguides/Scrum_Guide.pdf.
- SOFTEX (2009). *MPS-BR: Guia Geral*. Associação para Promoção da Excelência do Software Brasileiro. <http://www.softex.br>.
- Wells, D. (2011). Extreme programming: A gentle introduction. <http://www.extremeprogramming.org/>.