Relatório Técnico

Operações de Instanciação: Mecanismo de rastreamento entre Processos de Software e Projetos de Software

Renata Mesquita da Silva Santos
(renatames@cos.ufrj.br)

Toacy Cavalcante Oliveira
(toacy@cos.ufrj.br)

Rio de Janeiro
Maio de 2019

Operações de Instanciação: Mecanismo de rastreamento entre Processos de Software e Projetos de Software

Renata M. S. Santos[1], Toacy C. Oliveira[1]
[1] COPPE, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro - Brazil
renatames@cos.ufrj.br toacy@cos.ufrj.br

Resumo

As organizações de desenvolvimento de software buscam continuamente melhorar seus processos de desenvolvimento e manutenção de software, já que estes estão diretamente relacionados à qualidade dos produtos de software resultantes. Processos de software são considerados importantes para o setor de desenvolvimento de software, pois orquestram atividades, pessoas e informações envolvidas no desenvolvimento de software. O uso de processos de software para apoiar o desenvolvimento de software envolve a instanciação de elementos do domínio do processo de software em elementos do domínio do projeto de software. Nesse sentido, a instanciação preenche a lacuna entre processos de software e projetos de software e, se não executada com a devida assistência, pode aumentar a distância entre eles, dificultando sua reconciliação. Este relatório técnico apresenta um conjunto de operações de instanciação, como *link*, *split* e *merge*, que permitem mapear explicitamente os elementos de projeto e processo. As operações de instanciação pretendem promover uma transição suave entre processo e projeto, por meio de um mecanismo de rastreamento que permite a reconciliação de ambas as perspectivas. O mecanismo de trace cria vínculos entre as perspectivas de processo e projeto, por meio de uma estrutura de mapeamento entre os elementos utilizados para modelar o processo de software e os elementos usados para executar os projetos de software, reduzindo o distanciamento entre eles. Desta forma, este mapeamento apoia a extração de conhecimento sobre os processos a partir dos registros de execução gerados ao longo dos projetos de desenvolvimento de software.

Abstract

Software development organizations continually seek to improve their software development and maintenance processes, since the latter are directly related to the quality of the resulting software products. Software processes are also considered important players for the software development industry, as they orchestrate activities, people and information involved in software development. Using software processes to support software development, involves instantiating concepts from the software process domain into concepts of the software project domain. In this sense, process instantiation bridges the gap between software process and software project, and, if not executed with proper assistance, may increase the distance between them, making it difficult to reconcile them. This technical report introduces a set of instantiation operations such as link, split, and merge, that allow to explicitly map software project and process elements. Instantiation operations aim to foster a smooth transition between process and project, by means of a tracing mechanism that allows the reconciliation of both perspectives. The tracing mechanism creates links between the process and project perspectives, through a structure mapping structure between the elements used to model the software process and the elements used to execute the software projects, reducing the distance between them. In this way, this mapping supports the extraction of knowledge about the processes from the event logs generated during the software development projects.

Sumário

1. Introdução

Organizações de desenvolvimento de software buscam constantemente o desenvolvimento de produtos de software com qualidade. Para isso, buscam a melhoria de seus processos, uma vez que a qualidade do produto de software está diretamente relacionada à qualidade do processo que é utilizado em seu desenvolvimento e manutenção (OSTERWEIL, 1987; CUGOLA & GHEZZI, 1998; FUGGETTA, 2000; VALLE, SANTOS & LOURES, 2017).

Processos de software são considerados importantes no desenvolvimento de software, pois orquestram atividades, pessoas e informações envolvidas no desenvolvimento de software. Processo de software pode ser definido como um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que são necessários para conceber, desenvolver, implementar e manter um produto de software (FUGETTA, 2000). Além disso, é requerido que# um processo de software tenha uma organização lógica das diversas atividades técnicas e gerenciais que envolvem agentes, métodos, ferramentas e artefatos, e restrições que possibilitem disciplinar, sistematizar e organizar o desenvolvimento e manutenção de produtos de software (PRESSMAN, 2014). Muitos esforços têm sido dedicados para o aumento da produtividade, eficiência e efetividade do processo de desenvolvimento e manutenção dos produtos de software, com destaque para a criação de padrões de qualidade de processo, tais como a norma ISO/IEC 12207:2008 (ISO/IEC, 2008) e os modelos de maturidade CMMI (CMMI Product Team, 2010) e MPS.BR (SOFTEX, 2016a).

Para disseminar a utilização de um processo em uma organização, é primordial que este processo esteja bem documentado, e de acordo com Campos e Oliveira (2011), a modelagem de processos, na última década, tornou-se um importante mecanismo para apoiar a compreensão do comportamento dinâmico das organizações. Existem várias notações ou métodos disponíveis para a modelagem de processos, e ferramentas que suportam estas notações, tais como SPEM (S*oftware Process Engineering Metamodel Specification*) (OMG, 2008) e BPMN (Business Process Modeling and Notation) (OMG, 2014). Estas notações são definidas por meio de um metamodelo e consideram conceitos como atividades,

artefatos, papéis e relacionamentos.

O uso de processos de software para apoiar o desenvolvimento de software envolve a instanciação de elementos do domínio do processo de software em elementos do domínio do projeto de software (OMG, 2008 e Derniame et al.,1999). Como destacado no CMMI (CMMI Product Team, 2010), a implementação e gerenciamento do processo definido para o projeto são tipicamente descritos em um plano de projeto. De acordo com Reis (2003), a instanciação do processo modifica a especificação do processo, acrescentando informações detalhadas sobre os prazos, agentes e recursos utilizados para cada atividade definida no processo. Assim, as atividades definidas no processo de software geralmente são instanciadas, executadas e registradas na forma de tarefas em ferramentas de gerenciamento de projetos. Essas ferramentas são capazes de registrar informações importantes sobre a execução de processos, como registros de data e hora, identificação de tarefas e as partes interessadas envolvidas na execução de uma atividade. O registro destas informações em ferramentas de gerenciamento de projetos é classificado como sendo o registro de execução (LEMOS et.al., 2011).

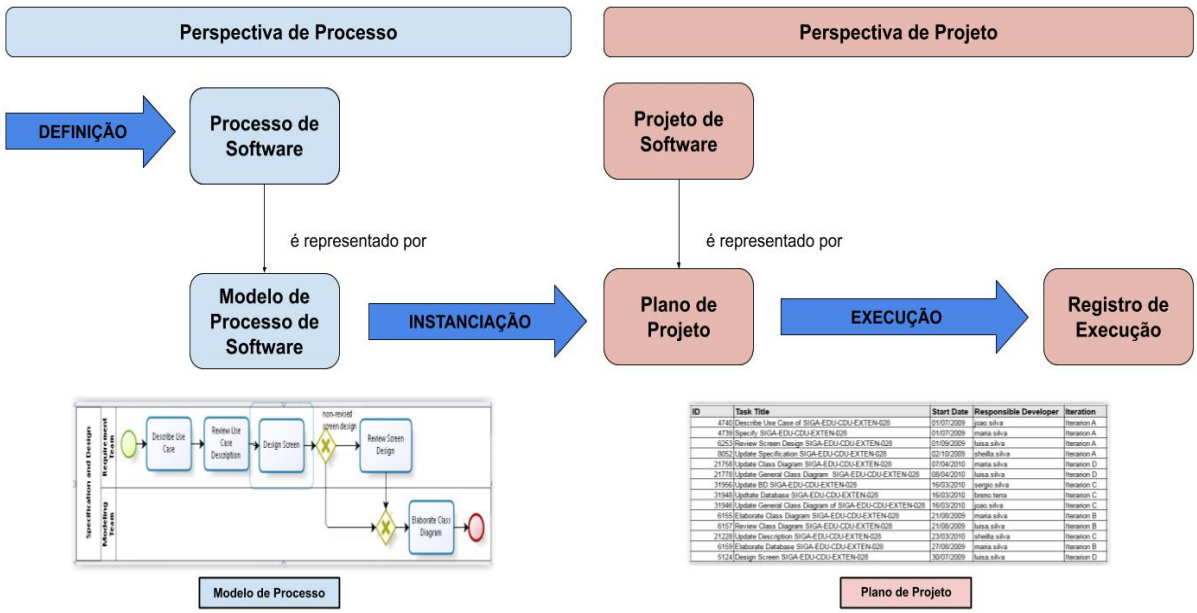


Figura 1. Perspectivas de Processo e Projeto

A partir do exposto, é possível identificar a existência de duas perspectivas: a Perspectiva de Processo e a Perspectiva de Projeto, apresentada na Figura 1. A Perspectiva de Processo está relacionada à definição do processo de software e

geralmente é representada na forma de um modelo de processo ou em linguagem natural, esta trata da representação e organização de elementos de processo, bem como a definição de tecnologias e melhores práticas de desenvolvimento de software requeridas. A Perspectiva do Projeto descreve um plano de projeto, na qual a mesma está relacionada à instanciação e a execução do processo, onde as atividades previstas na definição do processo, em geral, são registradas na forma de tarefas em ferramentas que são utilizadas na condução de projetos de software.

Embora os modelos de processo e os planos de projeto possam refletir práticas prescritivas ou ágeis, um plano de projeto deve inevitavelmente materializar alguns conceitos, se não todos, prescritos pelos processos de software associados (Derniame et al., 1999). No entanto, os gerentes têm dificuldade em entender e visualizar as atividades de desenvolvimento de software, definidas do processo de software, em um plano de projeto de software (Wu e Simmons (2000), Bastarrica et al (2017)). Pode não ser fácil mapear os conceitos encontrados nos domínios de processo e projeto, impedindo assim a reconciliação entre o modelo de processo e o plano do projeto. Por exemplo, uma atividade presente no modelo de processo pode ser considerada complexa demais para ser realizada por um único desenvolvedor, ao considerar a experiência da equipe, impondo assim que várias tarefas sejam criadas no plano de projetos, e vários desenvolvedores sejam alocados a essas.

Conforme observado por Ferreira, Szimanski & Ralha (2013) na prática, muitas vezes acontece dos registros de execução serem criados sem uma conexão explícita com a atividade de alto nível que está sendo executada e que cada atividade do processo pode gerar mais de um registro. De acordo com De Leoni, & Marrella (2017), às vezes, as atividades registradas (tarefas) não podem ser correspondidas a nenhuma das atividades definidas no modelo (atividades do processo). De acordo com Baier et al. (2018) o mecanismo de registro dos sistemas capturam etapas bem granulares em um nível técnico. Conforme observado por Ferreira, Szimanski & Ralha (2013) na prática, muitas vezes acontece que os registros de execução que podem ser capturados pelos sistemas de suporte são de baixa granularidade, como, por exemplo, "*O agente X enviou uma mensagem M ao agente Y*".

De acordo Valle, Santos e Loures (2017), atualmente, devido à crescente

utilização de sistemas de informação para apoiar a execução do processo, informações detalhadas sobre a implementação de processos são registradas como registros de eventos, logs de transações, etc. Porém a extração de conhecimento sobre os processos a partir dos registros de execução gerados ao longo dos projetos de desenvolvimento de software não é uma atividade trivial. Identificar uma correspondência entre as atividades do processo, nos planos e nos registros de execução não é uma tarefa simples (BASTARRICA et.al., 2017).

Ferreira, Szimanski & Ralha (2013) destacam que existe claramente um gap entre como os processos são definidos e como os registros são gerados durante a execução do processo. Este gap dificulta a identificação do processo do software que efetivamente está sendo realizado durante projetos de desenvolvimento de software. Desta forma, a identificação de não conformidades é impactada diretamente, uma vez que se faz necessário ter conhecimento do processo de software executado.

Este relatório apresenta um conjunto de operações de instanciação, que pretende apoiar adequadamente a transição entre as perspectivas de processo e projeto, por meio de um mecanismo de rastreamento que permita a reconciliação de ambas as perspectivas. A partir de uma estrutura de mapeamento entre os elementos utilizados para modelar o processo de software e os elementos usados para executar os projetos de software, objetiva-se minimizar o distanciamento entre processo e projeto, por meio da criação de vínculos entre eles.

2. Operações de Instanciação

Este relatório apresenta um conjunto de operações de instanciação para representar o mapeamento entre os conceitos encontrados na perspectiva do processo, para conceitos encontrados na perspectiva do projeto, permitindo assim a reconciliação de processos e projetos. De acordo Magdaleno, Werner & De Araujo (2012) reconciliação é definida como o ato de restabelecer as relações normais entre os beligerantes, e que no desenvolvimento de software, é uma tentativa de aproximar os diferentes modelos de desenvolvimento para estabelecer processos de software mais efetivos. Neste trabalho, pretende-se apresentar um mecanismo para

reconciliar modelos de processos e planos de projetos com base em operações explícitas de instanciação que capturam a lógica de mapeamento de processo para projeto.

As mudanças ocorridas na instanciação introduzem novos conceitos, como iteração, causando o distanciamento entre a definição e a execução dos processos de software. Além disso, a instanciação leva em consideração outros aspectos, como tempo, orçamento e alocação de equipe, que podem influenciar se ou quando as tarefas devem ser executadas. Neste sentido, é desejável implementar mecanismos que permitam que elementos do processo de software definido estejam explícitos durante a instanciação e execução. As operações de instanciação podem ser usadas ao criar o plano do projeto e podem capturar mapeamentos simples, como a ligação (*link*) entre dois conceitos, ou mapeamentos complexos envolvendo vários conceitos, como agrupamento (*merge*) ou divisão (*split*).

A Figura 1 apresenta o modelo conceitual resultante da aplicação das Operações de Instanciação. A figura 1 (lado esquerdo) apresenta um trecho de um modelo conceitual de processo, que está de acordo com os metamodelos SPEM (OMG, 2008) e BPMN (OMG, 2014), composto de elementos de processo de software como *Process*, *Activity*, *SequenceFlow*, *Artifact* e *Role*. Projetos de software são instâncias de modelos de processos e podem ser representados por meio de planos de projeto. A Figura 1 (lado direito) apresenta o modelo conceitual do projeto, que é resultante da instanciação do processo de software, foi desenvolvido com base nos conceitos do PMBOK (*Project Management Body of Knowledge*) (PMI, 2017) e na análise dos dados das tarefas que são registrados das ferramentas utilizadas durante os projetos de desenvolvimento de software. incluindo os seguintes elementos: *Project*, *Task*, *TaskArtifact*, *TaskActor*, *Instance* e *Iteration*.

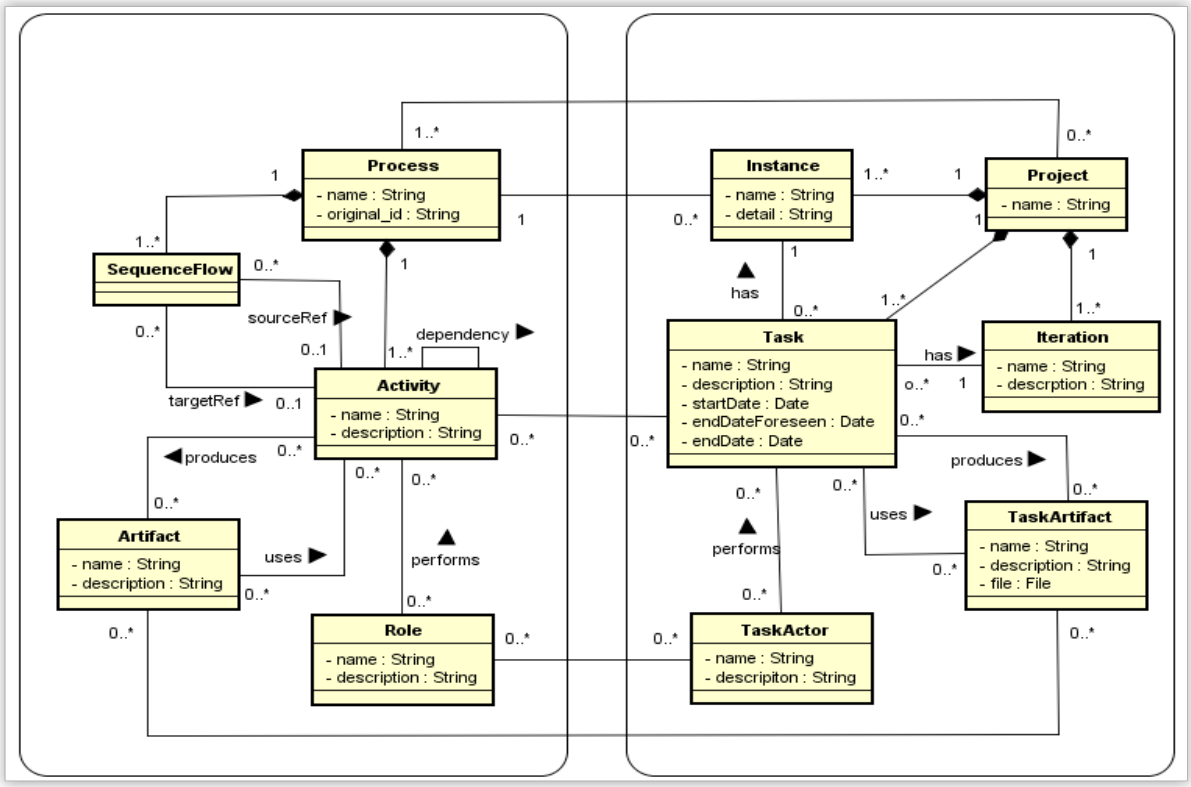


Figura 2. Mapeamento entre a atividade do processo e a tarefa do projeto

A Tabela 1 apresenta o nome e descrição de cada operação proposta. Elas foram definidas a partir da análise de planos de projeto de projetos de desenvolvimento de software, com base na experiência dos autores em gerenciamento de projetos de desenvolvimento de software, e considerando o estado da arte da literatura (BASTARRICA et al., 2017, MARTINS & da SILVA, 2016 e ADRIANSYAH and BUIJS, 2012). Nos planos de projeto analisados, identificamos manualmente, para cada tarefa do plano do projeto, a atividade correspondente do processo de software definido, com o apoio dos profissionais envolvidos no desenvolvimento do projeto. Por meio dessa análise, foi possível identificar três tipos de relacionamentos: *um para um*, *um para muitos* e *muitos para um*. Por meio das operações de instanciação é possível definir a associação entre *Process* e *Project*, *Process* e *Instance*, *Task* e *Activity*, *Role* e *TaskActor*, e *Artifact* e *TaskArtifact*. Nesse sentido, as operações explicitam a relação entre as perspectivas.

Tabela 1 Operações de Instanciação

Nome	**Descrição**
Link	Estabelece uma ligação entre uma atividade do processo e uma única tarefa do plano de projeto. A tarefa deve ter o mesmo nome da atividade do processo associada.
Detail	Estabelece uma ligação entre uma atividade do processo e uma única tarefa do plano de projeto. O nome da tarefa deve ser mais detalhado que o nome da atividade definida no processo, mas com o mesmo significado.
Rename	Estabelece uma ligação entre uma atividade do processo e uma única tarefa do plano de projeto. O nome da tarefa deve ser diferente do nome da atividade de processo associada.
Case	Estabelece uma ligação entre cada atividade prevista para uma instância do processo e as tarefas do plano de projeto. Cada nome de tarefa pode ser mais detalhado que o nome da atividade, de acordo com a necessidade do contexto do projeto de software. Todas as tarefas serão associadas a uma instância do processo.
Split	Estabelece uma ligação entre uma atividade do processo e duas ou mais tarefas do plano de projeto, em uma iteração. O nome de cada tarefa deve ser diferente do nome da atividade de processo associada.
Split Iteration	Estabelece uma ligação entre uma atividade do processo e duas ou mais tarefas do plano de projeto e as distribui em diferentes iterações. O nome de cada tarefa deve ser diferente do nome da atividade de processo associada.
Merge	Estabelece uma ligação entre duas ou mais atividade do processo e uma única tarefa do plano de projeto.
Add	Cria uma tarefa no plano de projeto que não possui uma atividade prevista no processo de software.
Remove	Uma atividade do processo de software não tem uma tarefa correspondente no plano de projeto.
Link Artifact	Estabelece uma ligação entre um artefato associado a uma atividade de processo e um artefato produzido por uma tarefa do plano do projeto. O artefato produzido por uma tarefa deve ter o mesmo tipo que a atividade de processo associada ao artefato.
Link Role	Estabelece uma ligação entre o papel associado a uma atividade de processo e o ator responsável por uma tarefa do plano do projeto.

O mapeamento entre as perspectivas é introduzido utilizando as operações, apresentadas na Tabela 1, que relacionam as tarefas (projeto) às atividades (processo). Também são relacionados o ator (projeto) e o papel (processo) e artefato produzido (projeto) e o artefato esperado (processo). A atividade do processo de referência é registrada em cada tarefa, exceto para as operações *Add* (a tarefa criada não possui atividade correspondente associada) e *Remove* (nenhuma tarefa é

criada a partir da atividade). O mapeamento entre *Role* e *Actor* e *Artifact* e *TaskArtifact* são introduzidos pelas operações *Link Artifac*t e *Link Role*, respectivamente.

Conforme descrito anteriormente, a aplicação de operações de instanciação estabelece um vínculo entre um plano de projeto (projeto de software) e modelos de processo (processo de software). Então, para usar as operações de instanciação, é necessário que a organização tenha pelo menos um processo de software definido. Engenheiros de processos responsáveis por projetar e implementar processos, gerentes ou *coaches* que os auxiliem e os orientem, geralmente possuem o conhecimento necessário para garantir a precisão das operações de instanciação, uma vez que devem conhecer o processo de software definido usado como referência para um determinado projeto.

Considera-se que ao usar as operações de instanciação propostas, o resultado do mapeamento estabelecido possui informações que permitem verificar a conformidade entre as perspectivas do processo e do projeto. Portanto, a reconciliação de projetos de software com processos de software permite a identificação de não conformidades, tais como atividades previstas não realizadas, alteração na ordem prevista, bem como atividades não previstas e realizadas, que podem ser confirmadas como não conformidades após uma análise minuciosa.

Cabe ressaltar que as operações de instanciação apresentadas podem ser confundidas com operações de adaptação, porém se diferenciam pelo tempo em que ocorrem e também pela estrutura de dados resultante. Adaptação de Processo é a ação de adaptar um processo de software definido para encontrar as necessidades específicas de uma organização ou projeto (PEDREIRA et al., 2007), e consiste em excluir, modificar ou adicionar novos elementos e/ou relacionamentos a um processo de software (PEREIRA et al., 2008). Nesse sentido, adaptação de processo é o ato de ajustar as definições e/ou de particularizar os termos de uma descrição de processo geral para derivar um novo processo aplicável em um ambiente alternativo (MARTÍNEZ-RUIZ et al.,2012), ou seja, as operações de adaptação tem como resultado um novo processo que se aplica às necessidades do projeto, e são definidas na perspectiva de definição. Enquanto que as operações de instanciação, propostas neste trabalho, ocorrem na perspectiva de execução,

considerando informações do contexto do projeto de desenvolvimento, tais como prazo e disponibilidade de recursos.

As operações definidas foram sistematizadas/implementadas em uma ferramenta, visando apoiar, de forma semi-automatizada, o mapeamento entre as perspectivas de processo e projeto, bem como apoiar a identificação de não-conformidades.

Cada operação será apresentada em detalhes a seguir.

2.1 Descrição detalhada das Operações de Instanciação

Cada Operação de Instanciação está descrita utilizando a estrutura apresentada na Tabela 2. Este formato de descrição foi baseado na forma como os Padrões de Projetos são descritos em Gamma et al. (1995).

Tabela 2 Formato padrão para descrição de cada Operação de Instanciação

Nome	*[Nome da Operação]*
Operação	*[Representação da operação]*
Descrição	*[O que a operação faz* *Que tipo de problema resolve.]*
Motivação	*[Um cenário que ilustre o problema e como o comportamento de cada operação resolvem o problema do cenário de exemplo.]*
Aplicabilidade	*[Pretende-se descrever por meio de respostas para as seguintes questões:* *Quais são as situações em que a operação pode ser aplicada?* *Quais são os exemplos de situações que a operação pode tratar?* *Como você pode reconhecer as situações de aplicabilidade?]*
Participantes	*[Explicação de cada elemento do modelo conceitual envolvidos na aplicação da operação.]*
Relacionamento	*[Tipo de relacionamento entre os participantes envolvidos]* *[um para um, um para muitos e muitos para um]*
Pseudocódigo	*[Pseudocódigo do funcionamento da operação.]*
Exemplo Projeto SIGA-EPCT	*[Figura que exemplifica a aplicação da operação.]*

Cada operação esta descrita em detalhes nas subseções seguintes,

utilizando a estrutura apresentada na Tabela 2.

2.1.1 Operação *Link*

Nome

Link

Operação

task.createLink(iteration: Iteration, instance: Instance, project: Project, processActivity: Activity)

Descrição

Estabelece uma ligação entre uma atividade do processo e uma única tarefa do plano de projeto. A tarefa deve ter o mesmo nome da atividade do processo associada.

Motivação

Atividades do processo de software definido para o projeto são instanciadas, porém a falta de vínculo entre o processo e o plano de projeto dificulta a extração de informações do processo ao longo do projeto de software. A criação de tarefas com o mesmo nome da atividade e a representação da atividade do processo associada à tarefa do plano de projeto ajuda na redução do distanciamento entre processo e projeto.

Aplicabilidade

Esta operação se aplica à situação na qual se pretende criar uma tarefa no plano de projeto igual à atividade prevista no processo de software definido. Nesta situação a atividade prevista atende às necessidades em sua descrição e propósito. Esta operação cria uma tarefa no plano de projeto com o mesmo nome de atividade de processo e registra a atividade do processo definido.

Participantes

Activity: Atividade do processo de software definido para o projeto

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

Um para Um entre *Activity* e *Task*.

Pseudocódigo

```
function task.createLink (iteration: Iteration, instance: Instance,
project: Project, processActivity: Activity)
begin
   self.iteration = iteration
   self.instance = instance
   self.project = project
   self.processActivity = processActivity
   self.name = processActivity.name
   self.defineTask(task_description, task_status,  task_startDateForeseen,
   task_startDate, task_endtDateForeseen, task_endDate)

   return task
end
```

Exemplo

Projeto SIGA-EPCT-EPCT

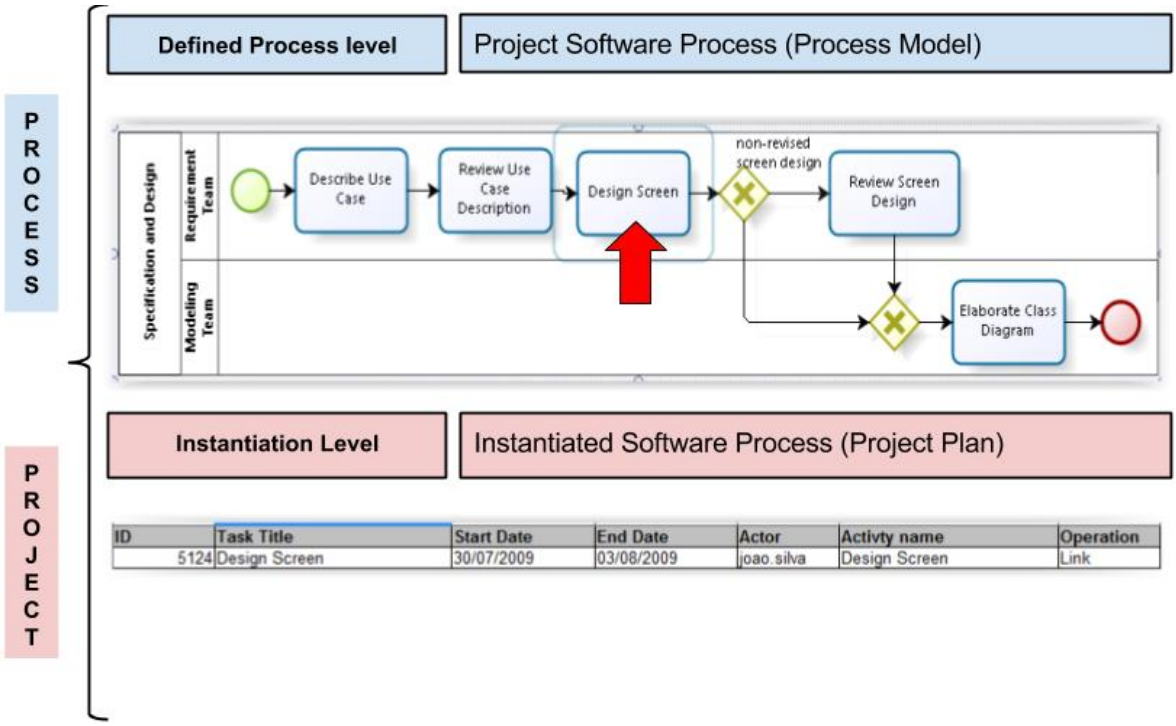


Figura 3. Exemplo da operação *Link*

2.1.2 Operação *Detail*

Nome

Detail

Operação

task.createDetail(iteration: Iteration, instance: Instance, project: Project, processActivity: Activity, task_name_detail: String)

Descrição

Estabelece uma ligação entre uma atividade do processo e uma única tarefa do plano de projeto. O nome da tarefa deve ser mais detalhado que o nome da atividade, mas com o mesmo significado.

Motivação

As atividades previstas no processo de software podem ser definidas em alto nível, sendo necessário um maior detalhamento em sua instanciação no projeto de software. A criação de tarefas com o nome da atividade mais detalhado e a representação da atividade do processo associada à tarefa do plano de projeto ajuda na redução do distanciamento entre processo e projeto.

Aplicabilidade

Esta operação se aplica à situação na qual se pretende instanciar uma atividade do processo de software, porém no contexto do projeto de software, a tarefa a ser criada deve ter o nome da atividade prevista mais detalhado. Nesta situação, a atividade prevista atende às necessidades em sua descrição e propósito, sendo necessário um melhor detalhamento. Esta operação cria uma tarefa no plano de projeto com o nome mais detalhado que o nome da atividade de processo, mas com o mesmo significado, e registra a atividade do processo de referência.

Participantes

Activity: Atividade do processo de software definido para o projeto

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

Um para Um entre *Actitivy* e *Task*.

Pseudocódigo

```
function task.createLink (iteration: Iteration, instance: Instance,
project: Project, processActivity: Activity, task_name_detail: String)
begin
   self.iteration = iteration
   self.instance = instance
   self.project = project
   self.processActivity = processActivity
   self.name = processActivity.name + task_name_detail
   self.defineTask(task_description, task_status, task_startDateForeseen,
   task_startDate, task_endtDateForeseen, task_endDate)

   return task
end
```

Exemplo

Projeto SIGA-EPCT

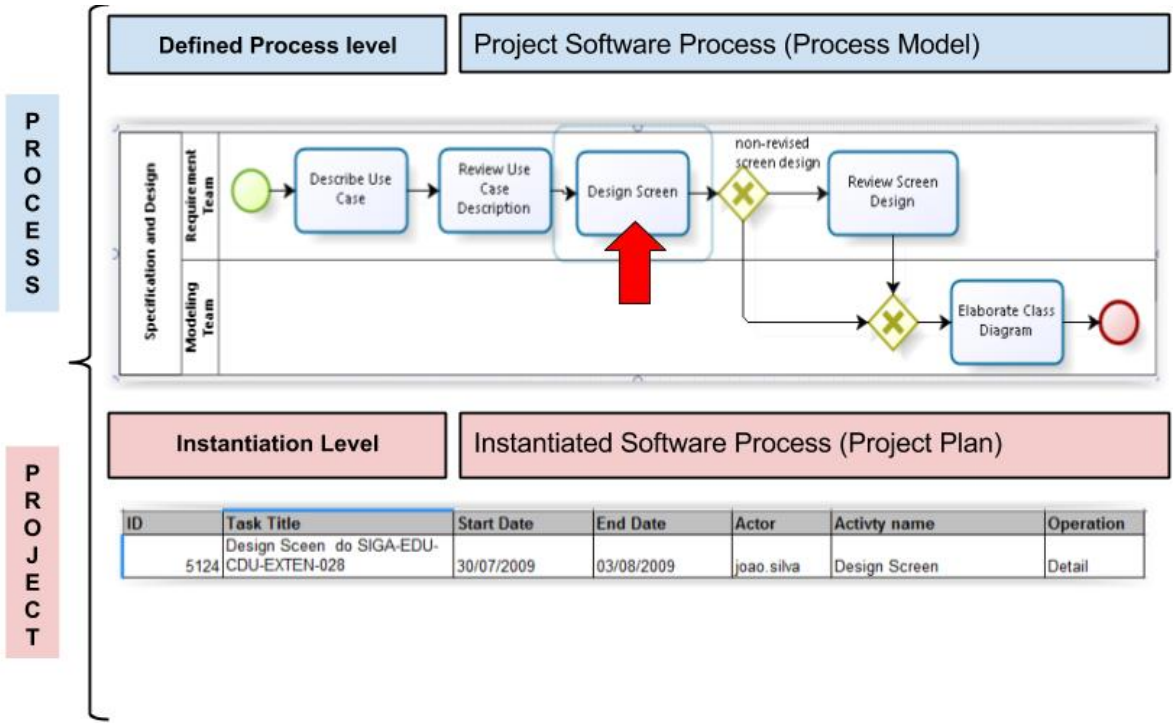


Figura 4. Exemplo da operação *Detail*

2.1.3 Operação *Rename*

Nome

Rename

Operação

task.createRename(iteration: Iteration, instance: Instance, project: Project, processActivity: Activity, task_name: String)

Descrição

Estabelece uma ligação entre uma atividade do processo e uma única tarefa do plano de projeto. O nome da tarefa deve ser diferente do nome da atividade de processo associada.

Motivação

As atividades previstas no processo de software podem ser definidas em alto nível e não descrevem especificamente o que se pretende realizar na tarefa do projeto, sendo necessário renomeá-las em sua instanciação no projeto de software. A criação de tarefas com o nome diferente da atividade associada, para se adequar ao contexto do projeto, e a representação da atividade do processo associada à tarefa do plano de projeto ajuda na redução do distanciamento entre processo e projeto.

Aplicabilidade

Esta operação se aplica à situação na qual se pretende instanciar uma atividade do processo de software, porém no contexto do projeto de software, a tarefa a ser criada deve ter o nome diferente. Nesta situação a atividade prevista atende às necessidades em seu propósito, porém o nome da tarefa deve refletir mais especificamente o que será realizado, sendo necessário definir um nome diferente da atividade do processo associada. Esta operação cria uma tarefa no plano de projeto com o nome da tarefa diferente do nome da atividade de processo e registra a atividade do processo definido.

Participantes

Activity: Atividade do processo de software definido para o projeto

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

Um para Um entre *Activity* e *Task*.

Pseudocódigo

```
function task.createLink (iteration: Iteration, instance: Instance,
project: Project, processActivity: Activity, task_name: String)
begin
   self.iteration = iteration
   self.instance = instance
   self.project = project
   self.processActivity = processActivity
   self.name = task_name
   self.defineTask(task_description, task_status, task_startDateForeseen,
   task_startDate, task_endtDateForeseen, task_endDate)

   return task
end
```

Exemplo

Projeto SIGA-EPCT

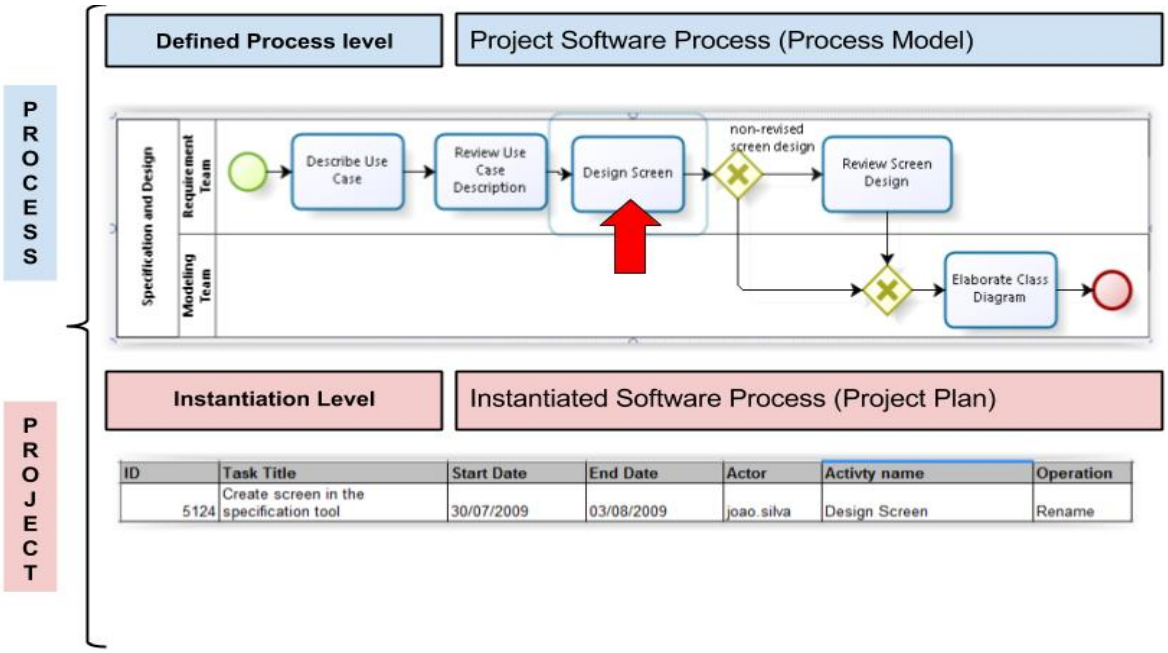


Figura 5. Exemplo da operação *Rename*

2.1.4 Operação *Case*

Nome

Case

Operação

task.createCase(instance: Instance, project: Project, process_activities: Array)

Descrição

Estabelece uma ligação entre cada atividade prevista para uma instância do processo e as tarefas do plano de projeto. Cada nome de tarefa pode ser mais detalhado que o nome da atividade, de acordo com a necessidade do contexto do projeto de software. Todas as tarefas serão associadas a uma instância do processo.

Motivação

O processo de software prescreve um conjunto de atividades que devem ser instanciadas no contexto de um projeto de software. A criação de tarefas, com o nome da atividade mais detalhado, para cada atividades previstas em uma determinada instância, e a representação da atividade do processo associada à tarefa do plano de projeto ajuda na redução do distanciamento entre processo e projeto.

Aplicabilidade

Esta operação se aplica à situação na qual se pretende instanciar todas as atividades previstas para uma determinada instância do processo de software Nesta situação as atividades previstas atendem às necessidades em seus propósitos. Esta operação cria uma tarefa no plano de projeto, para cada atividade prevista na instância do processo, e registra a atividade do processo definido.

Participantes

Activity: Atividade do processo de software definido para o projeto

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

Um para Um entre cada *Activity* e *Task*.

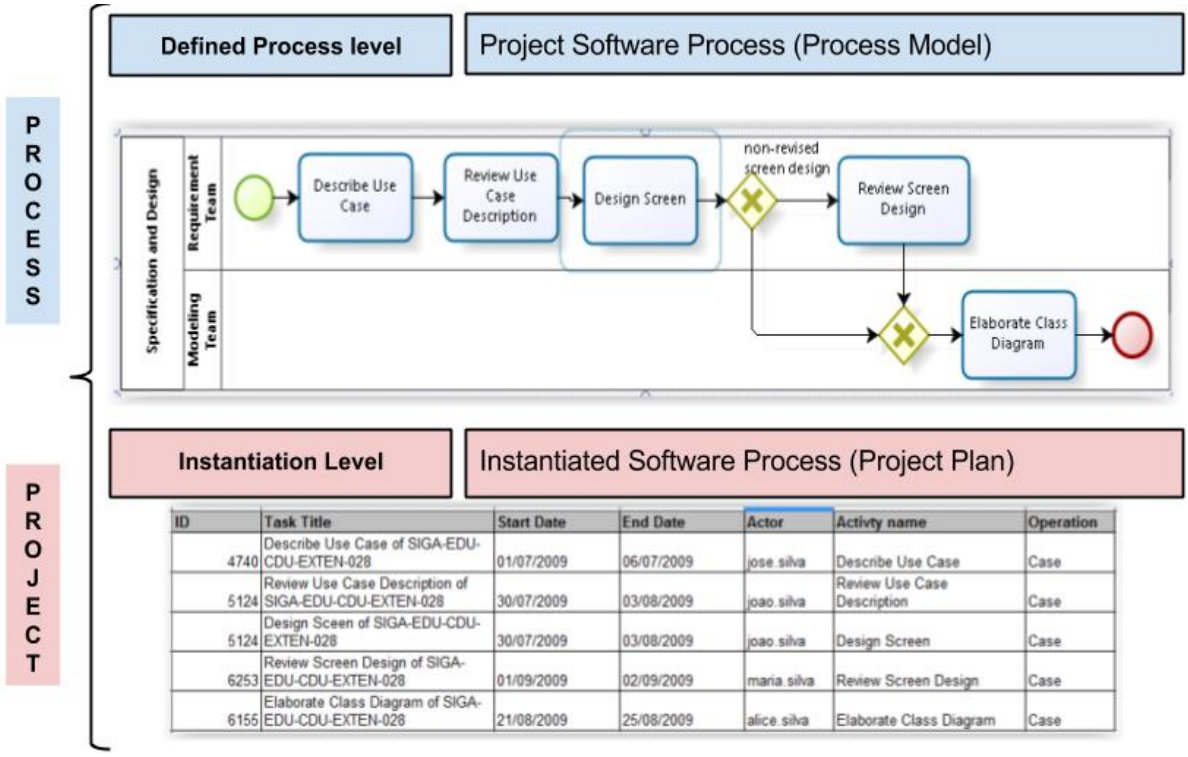
Pseudocódigo

```
function create_case (instance: Instance, project: Project, process_activities: Array)
begin
    for each activity in process_activities
    begin
      self.instance = instance
      self.project = project
      self.process_activity = process_activity
      self.iteration = read_iteration()
      self.name = process_activity.name + read_task_name_detail()
      self.defineTask(task_description, task_status, task_startDateForeseen,
        task_startDate, task_endtDateForeseen, task_endDate)
      return task

end
```

Exemplo

Projeto SIGA-EPCT



ID	Task Title	Start Date	End Date	Actor	Activty name	Operation
4740	Describe Use Case of SIGA-EDU-CDU-EXTEN-028	01/07/2009	06/07/2009	jose.silva	Describe Use Case	Case
5124	Review Use Case Description of SIGA-EDU-CDU-EXTEN-028	30/07/2009	03/08/2009	joao.silva	Review Use Case Description	Case
5124	Design Sceen of SIGA-EDU-CDU-EXTEN-028	30/07/2009	03/08/2009	joao.silva	Design Screen	Case
6253	Review Screen Design of SIGA-EDU-CDU-EXTEN-028	01/09/2009	02/09/2009	maria.silva	Review Screen Design	Case
6155	Elaborate Class Diagram of SIGA-EDU-CDU-EXTEN-028	21/08/2009	25/08/2009	alice.silva	Elaborate Class Diagram	Case

Figura 6. Exemplo da operação *Case*

2.1.5 Operação *Split*

Nome

Split

Operação

task.createSplit(iteration: Iteration, instance: Instance, project: Project, processActivity: Activity, task_name: String)

Descrição

Estabelece uma ligação entre uma atividade do processo e duas ou mais tarefa do plano de projeto, em uma iteração. O nome de cada tarefa deve ser diferente do nome da atividade de processo associada.

Motivação

As atividades previstas no processo de software podem ser definidas em alto nível, sendo necessário na instanciação para o projeto de software criar duas ou mais tarefas no plano de projeto, em uma mesma iteração. A criação de tarefas, como o nome adequado ao contexto do projeto de software, e a representação da atividade do processo associada a cada tarefa do plano de projeto ajuda na redução do distanciamento entre processo e projeto.

Aplicabilidade

Esta operação se aplica à situação na qual se pretende criar duas ou mais tarefas, para uma iteração, oriundas de uma única atividade prevista no processo de software definido. Nesta situação a atividade prevista atende às necessidades em sua descrição e propósito, mas o alto nível da definição da mesma e o contexto do projeto, implicam na criação de tarefas mais específicas. Esta operação cria duas ou mais tarefa no plano de projeto, em uma iteração, para uma única atividade do processo, e registra a atividade do processo definido.

Participantes

Activity: Atividade do processo de software definido para o projeto

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

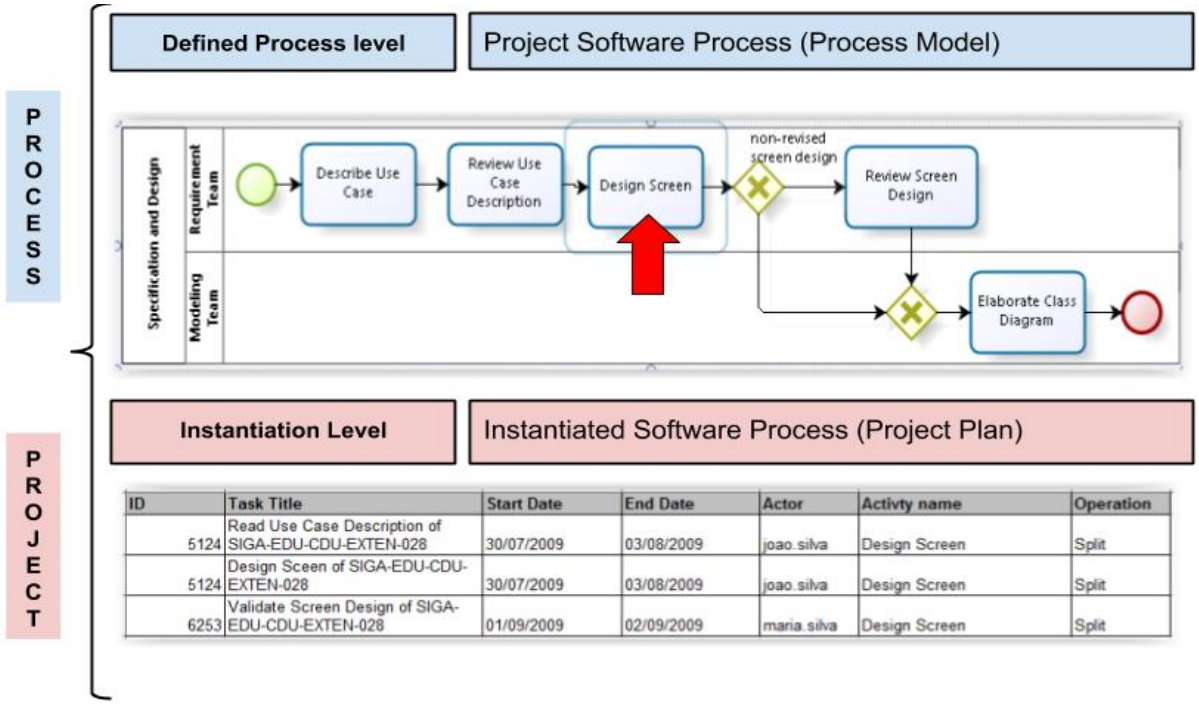
Um para Muitos entre *Activity* e *Task*.

Pseudocódigo

```
function create_split (instance: Instance, iteration: Iteration, project: Project, process_activity:
Activity, task_name: String)
begin
    while
    begin
        self.instance = instance
        self.iteration = iteration
        self.project = project
        self.process_activity = process_activity
        self.name = task_name
        self.defineTask(task_description, task_status,  task_startDateForeseen,
          task_startDate, task_endtDateForeseen, task_endDate)
        return task
    end
end
```

Exemplo

Projeto SIGA-EPCT



ID	Task Title	Start Date	End Date	Actor	Activty name	Operation
5124	Read Use Case Description of SIGA-EDU-CDU-EXTEN-028	30/07/2009	03/08/2009	joao.silva	Design Screen	Split
5124	Design Sceen of SIGA-EDU-CDU-EXTEN-028	30/07/2009	03/08/2009	joao.silva	Design Screen	Split
6253	Validate Screen Design of SIGA-EDU-CDU-EXTEN-028	01/09/2009	02/09/2009	maria.silva	Design Screen	Split

Figura 7. Exemplo da operação *Split*

2.1.6 Operação *Split Iteration*

Nome

Split Iteration

Operação

task.createSplit(iteration: Iteration, instance: Instance, project: Project, processActivity: Activity, task_name: String)

Descrição

Estabelece uma ligação entre uma atividade do processo e duas ou mais tarefa do plano de projeto e as distribui em diferentes iterações. O nome de cada tarefa deve ser diferente do nome da atividade de processo associada.

Motivação

As atividades previstas no processo de software podem ser definidas em alto nível, sendo necessário na instanciação para o projeto de software criar duas ou mais tarefas no plano de projeto, em diferentes iterações. A criação de tarefas como o nome adequado ao contexto do projeto de software e a representação da atividade do processo associada a cada tarefa do plano de projeto ajuda na redução do distanciamento entre processo e projeto.

Aplicabilidade

Esta operação se aplica à situação na qual se pretende criar duas ou mais tarefas, para iterações distintas, oriundas de uma única atividade prevista no processo de software definido. Nesta situação, a atividade prevista atende às necessidades em sua descrição e propósito, mas o alto nível da definição da mesma e o contexto do projeto, implicam na criação de tarefas mais específicas. Esta operação cria duas ou mais tarefa no plano de projeto, em iterações distintas, para uma única atividade do processo, e registra a atividade do processo definido.

Participantes

Activity: Atividade do processo de software definido para o projeto

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

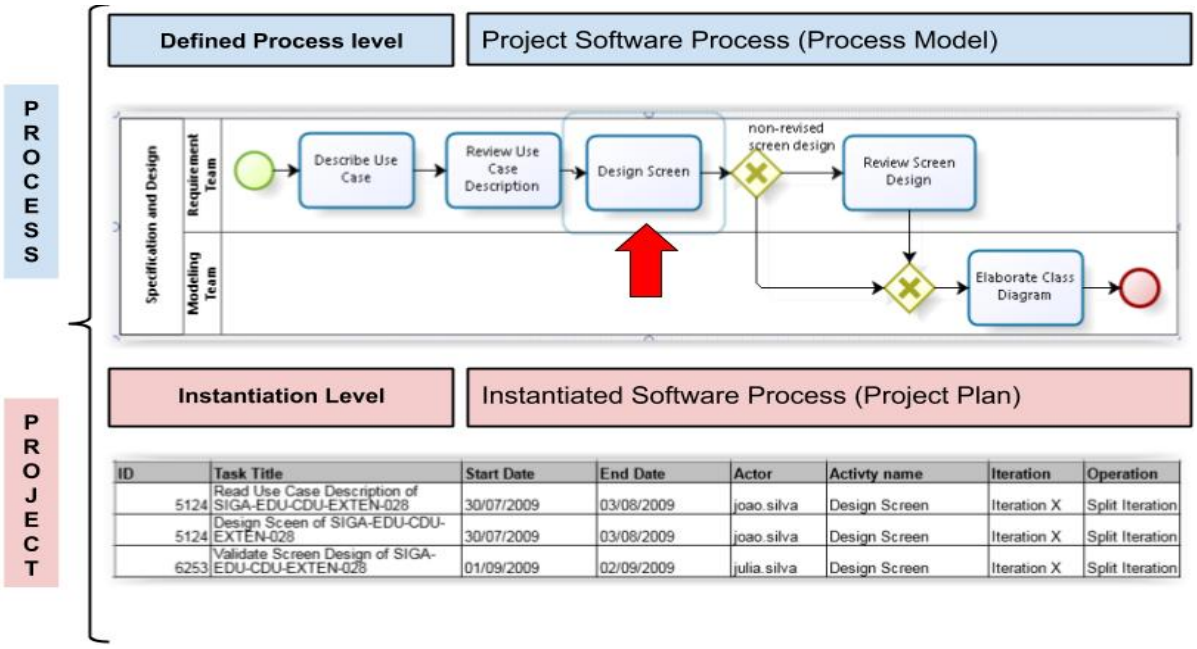
Um para Muitos entre *Activity* e *Task*.

Pseudocódigo

```
function createSplitIteration (instance: Instance, iteration: Iteration, project: Project
process_activity: Activity, task_name: String)
begin
    while
    begin
        self.iteration = read_iteration()
        self.instance = instance
        self.project = project
        self.process_activity = process_activity
        self.name = task_name
        self.defineTask(task_description, task_status,  task_startDateForeseen,
  task_startDate, task_endtDateForeseen, , task_endDate)
        return task
    end
end
```

Exemplo

Projeto SIGA-EPCT



ID	Task Title	Start Date	End Date	Actor	Activty name	Iteration	Operation
5124	Read Use Case Description of SIGA-EDU-CDU-EXTEN-028	30/07/2009	03/08/2009	joao.silva	Design Screen	Iteration X	Split Iteration
5124	Design Sceen of SIGA-EDU-CDU-EXTEN-028	30/07/2009	03/08/2009	joao.silva	Design Screen	Iteration X	Split Iteration
6253	Validate Screen Design of SIGA-EDU-CDU-EXTEN-028	01/09/2009	02/09/2009	julia.silva	Design Screen	Iteration X	Split Iteration

Figura 8. Exemplo da operação *Split Iteration*

2.1.7 Operação *Merge*

Nome

Merge

Operação

task.createMerge(iteration: Iteration, instance: Instance, project: Project, process_activities: Array, task_name: String)

Descrição

Estabelece uma ligação entre duas ou mais atividade do processo e uma única tarefa do plano de projeto.

Motivação

As atividades previstas no processo de software podem ser definidas um baixo nível de detalhamento, sendo necessário, em sua instanciação no projeto de software, agrupá-las em uma tarefa no plano de projeto. A criação da tarefa com o nome diferente das atividades associadas, para se adequar ao contexto do projeto, e a representação das atividades do processo associadas à tarefa do plano de projeto ajuda na redução do distanciamento entre processo e projeto.

Aplicabilidade

Esta operação se aplica à situação na qual se pretende instanciar duas ou mais atividades do processo de software em uma única tarefa no plano de projeto. Nesta situação, as atividades previstas associadas atendem às necessidades em seu propósito e descrição, porém para o contexto do projeto de software uma única tarefa deve ser criada. Esta operação cria uma tarefa no plano de projeto com o nome diferente das atividades associadas e registra as atividades do processo definido.

Participantes

Activity: Atividade do processo de software definido para o projeto

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

Muitos para Um entre *Activity* e *Task*.

Pseudocódigo

```
function createMerge(iteration: Iteration, instance: Instance, project: Project,
process_activities: Array, task_name: String)
begin
    self.instance = instance
    self.iteration = iteration
    self.project = project
    self.process_activities = process_activities
    self.name = task_name
    self.defineTask(task_description, task_status,  task_startDateForeseen,
  task_startDate, task_endtDateForeseen, , task_endDate)
    return task
end
```

Exemplo

Projeto SIGA-EPCT

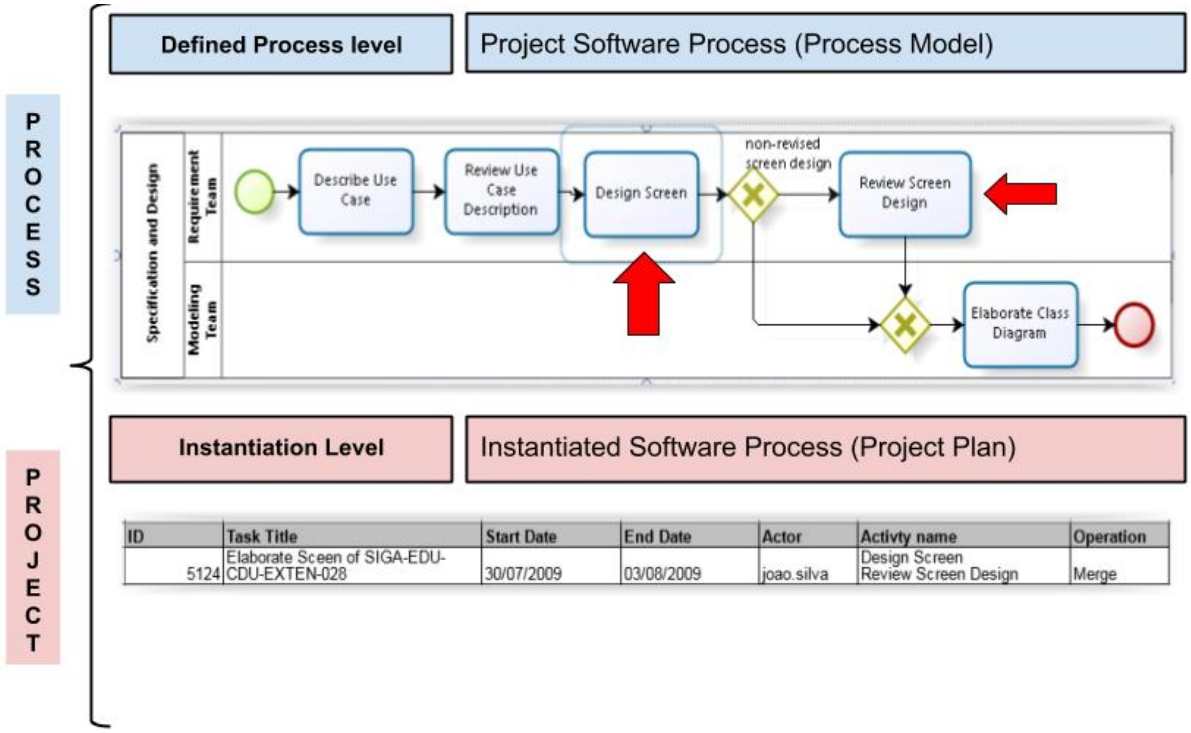


Figura 9. *Exemplo da operação Merge*

2.1.8 Operação *Add*

Nome

Add

Operação

task.createAdd(iteration: Iteration, instance: Instance, project: Project, task_name: String)

Descrição

Cria uma tarefa no plano de projeto que não possui uma atividade prevista no processo de software.

Motivação

As atividades previstas no processo de software não atendem às necessidades de uma tarefa do plano de projeto. Uma tarefa é criada e não possui a representação da atividade do processo associada.

Aplicabilidade

Esta operação se aplica à situação na qual nenhuma atividade do processo de software atende à tarefa no plano de projeto. Nesta situação, nenhuma atividade prevista atende às necessidades em sua descrição e propósito, sendo necessário criar uma tarefa no plano de projeto sem nenhuma atividade do processo associada. Esta operação cria uma tarefa no plano de projeto e não é registrada uma atividade do processo definido.

Participantes

Task: Tarefa do projeto presente no plano de projeto

Relacionamento

Não se aplica.

Pseudocódigo

```
function task.createAdd(iteration: Iteration, instance: Instance,
project: Project,, task_name: String)
begin
     self.iteration = iteration
     self.instance = instance
     self.project = project
     self.process_activity = NULL
     self.name = task_name
     self.defineTask(task_description, task_status,  task_startDateForeseen,
   task_startDate, task_endtDateForeseen, , task_endDate)
     return task
end
```

Exemplo

Projeto SIGA-EPCT

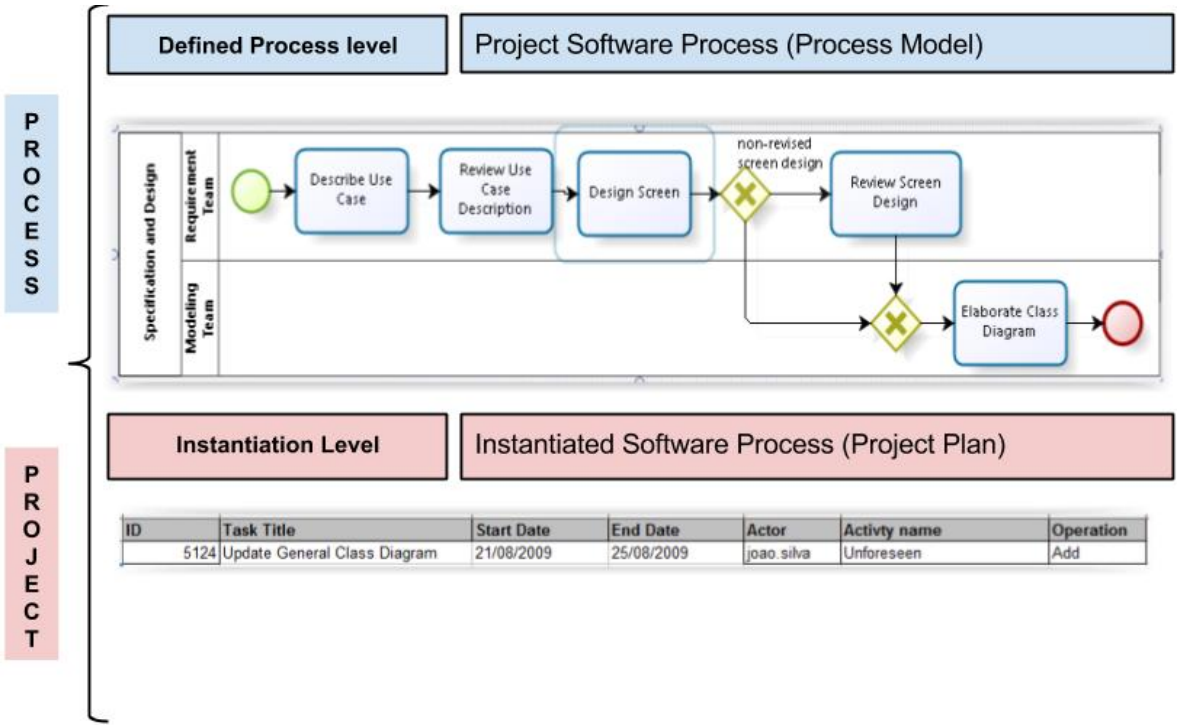


Figura 10. *Exemplo da operação Add*

2.1.9 Operação *Remove*

Nome

Remove

Operação

task.remove(processActivity)

Descrição

Uma atividade do processo de software não tem uma tarefa correspondente no plano de projeto.

Motivação

As atividades previstas no processo de software podem não ser instanciadas no projeto de software. Nenhuma tarefa é associada às atividades previstas.

Aplicabilidade

Esta operação se aplica à situação na qual não se pretende criar tarefas para uma determinada atividade prevista no processo de software. Nesta situação esta atividade prevista não atende às necessidades em sua descrição e propósito, não sendo associada a tarefas no plano de projeto.

Participantes

Activity: Atividade do processo de software definido para o projeto

Relacionamento

Não se aplica.

Pseudocódigo

Não se aplica.

Exemplo

Projeto SIGA-EPCT

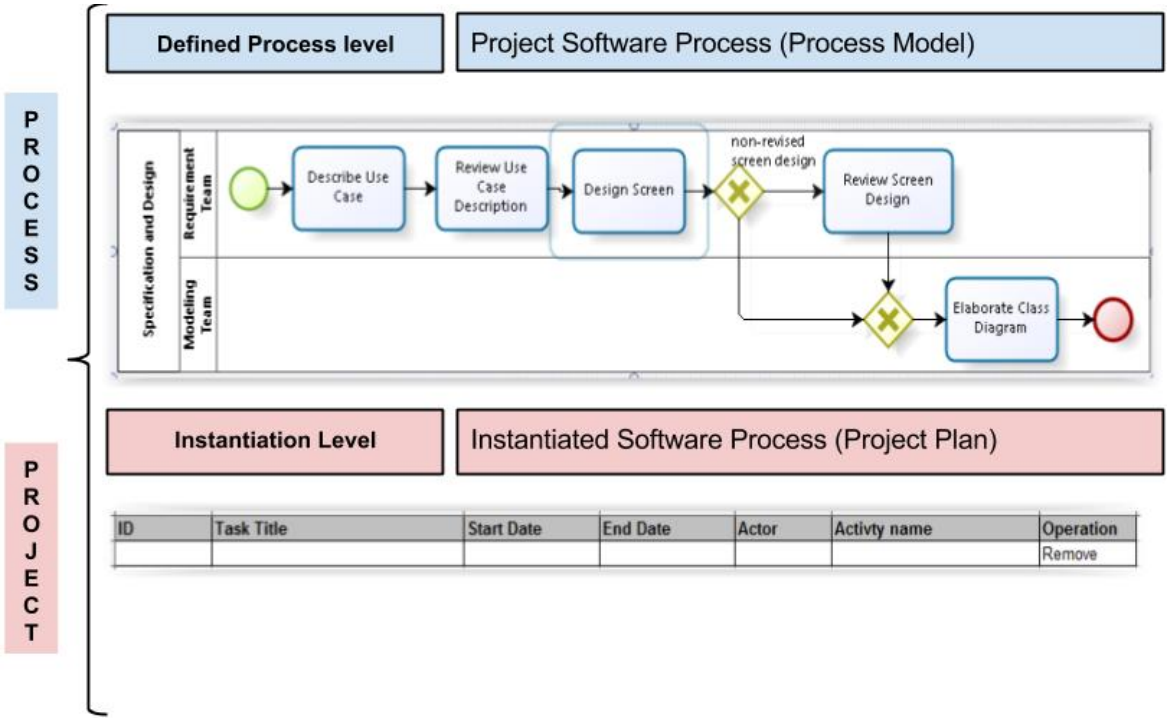


Figura 11. *Exemplo da operação Remove*

2.1.10 Operação *Link Artifact*

Nome

Link Artifact

Operação

task.createLinkArtifact(artifact: Artifact, project: Project, processActivity: Activity)

Descrição

Estabelece uma ligação entre um artefato associado a uma atividade de processo e um artefato produzido por uma tarefa do plano do projeto. O artefato produzido por uma tarefa deve ter o mesmo tipo que a atividade de processo associada ao artefato.

Participantes

Artifact: Artefato associado a uma atividade do processo de software definido para o projeto

TaskArtifact: Artefato associado a uma tarefa do projeto presente no plano de projeto

Relacionamento

Um para Um entre *Artifact* e *TaskArtifact*.

Pseudocódigo

Não se aplica.

2.1.11 Operação *Link Role*

Nome

Link Role

Operação

task.createLinkRole(role: Role, project: Project, processActivity: Activity)

Descrição

Estabelece uma ligação entre o papel associado a uma atividade de processo e o ator responsável por uma tarefa do plano do projeto.

Participantes

Role: Papel responsável por atividade do processo de software definido para o projeto.

TaskActor: Ator responsável por uma tarefa do projeto presente no plano de projeto

Relacionamento

Um para Um entre *Role* e *TaskActor*.

Pseudocódigo

Não se aplica.

2.2 Estudos conduzidos

Estudos foram conduzidos para apoiar a definição e estruturação das operações de instanciação propostas.

Após a definição das operações de instanciação descritas acima, foi realizado um estudo de observação em uma organização de desenvolvimento de software de médio porte, a fim de avaliar as operações de instanciação propostas. A técnica de estudo de observação foi escolhida para entender como o planejamento de um novo projeto é feito. O estudo foi realizado em uma reunião de planejamento de uma nova versão de software (sprint), na qual foi possível identificar a ocorrência das operações propostas. Também foram realizadas entrevistas com os gerentes de projeto com o objetivo de caracterizar o processo de criação de tarefas e, assim, mapear as operações utilizadas. Em geral, com base no estudo de observação e nas entrevistas realizadas, foi possível confirmar que a relação proposta por cada operação é observada durante a criação das tarefas.

Além disso, a fim de avaliar a eficácia das operações de instanciação, foi realizada uma validação usando dados reais do processo de desenvolvimento de um sistema integrado de gerenciamento acadêmico. A validação foi realizada da seguinte forma: (i) obtenção das tarefas realizadas no projeto (logs de eventos); (ii) recriação de tarefas por meio das operações, estabelecendo o mapeamento entre cada tarefa e sua atividade correspondente no processo de software de referência, quando possível; (iii) registro da operação usada para recriar cada tarefa; e (iv) registro da instância de processo de cada tarefa. O mapeamento estabelecido reduziu a distância entre o processo de software e o projeto de software e permite uma melhor compreensão da execução dos processos.

3. Mapeamento Sistemático

A fim de identificar os trabalhos, que relatem sobre operações de instanciação aplicadas à criação de tarefas, um mapeamento sistemático da literatura foi

realizado, o qual consiste em uma estratégia de revisão da literatura e será descrita na seção a seguir.

3.1 Planejamento

3.1.1 Objetivo da Busca Estruturada

Seguindo a abordagem GQM (*Goal, Question, Metric*) (BASILI, CALDIERA e ROMBACH, 1994), a busca estruturada desenvolvida neste trabalho possui como objetivo, *a análise de publicações que tratam sobre operações de instanciação*, com **o propósito de** *caracterizar* **com respeito a** *identificação abordagens (ações, operações, padrões, políticas ou regras) de instanciação de processo relacionadas à utilização de operações de instanciação para criação do plano de projeto*, **do ponto de vista dos** *pesquisadores envolvidos na busca,* **no contexto** acadêmico e da indústria.

3.1.2 Questão de Pesquisa

Questão Principal
Quais abordagens (ações, operações, padrões, políticas ou regras) de instanciação de processo são utilizadas para a criação do plano de projeto (criação de tarefas)?

3.1.3 Estratégia de Busca e Artigos de Controle

Nesta busca estruturada, a abordagem PICO – *Population of interest, evaluated Intervention, Intervention Comparison e expected Outcome* (PAI et al., 2004), foi utilizada para organizar e estruturar a busca a ser realizada. A estruturação adotada está descrita a seguir:

(P) População (*Population*): Processo de Software**.**
(I) Intervenção (*Intevention*): Pesquisas relacionadas à instanciação de processo de software**.**

(C) Comparação (*Comparison*): Não há, pois não se tem como objetivo comparar abordagens, e sim, caracterizá-las.
(O) Resultados (*Output*): Ações, Operações, Regras, Padrões ou Políticas de apoio a criação do plano de projeto.

Baseado na estrutura PICO, os seguintes termos e seus sinônimos foram definidos:

(P) População: *software process, software project, process of software, software development project, software development process, project construction, software development*
(I) Intervenção: *instantiation, creating task, creating activity, creation of task, creation of activity, creating project plan, creation of project plan, creating plan, creation of plan, deployment, creating schedule, creation of schedule, creating WBS, "creation of WBS*
(C) Comparação: não aplicável
(O) Resultados: *action, operating, pattern, policy, rule*

A partir desta lista de sinônimos a *string* de busca foi definida e é apresentada no Quadro 1.

Quadro 1 String de busca

(TITLE-ABS-KEY (("software process" OR "software project" OR "process of software" OR "software development project" OR "software development process" OR "reference model" OR "standard process" OR "defined process" OR "process description" OR "description of process" OR "modelled process" OR "modelled software process")) AND TITLE-ABS-KEY (("instantiation" OR "creating task" OR "creating activity" OR "creation of task" OR "creation of activity" OR "creating project plan" OR "creation of project plan" OR "creating plan" OR "creation of plan" OR "deployment" OR "creating schedule" OR "creation of schedule" OR "creating WBS" OR "creation of WBS")) AND TITLE-ABS-KEY (("action" OR "operation" OR "pattern" OR "policy" OR "rule")))

Base de Busca: Scopus (www.scopus.com)

String de Busca: (TITLE-ABS-KEY (("software process*" OR "software project*" OR "process of software" OR "software development project*" OR "software development process*" OR "reference model" OR "standard process*" OR "defined process*" OR "process description*" OR "description of process*" OR "modelled process" OR "modelled software process*")) AND TITLE-ABS-KEY (("instantiation" OR "creating task" OR "creating activit*" OR "creation of task*" OR "creation of activit*" OR "creating project plan" OR "creation of project plan" OR "creating plan" OR "creation of plan" OR "deployment" OR "creating schedule" OR "creation of schedule" OR "creating WBS" OR "creation of WBS")) AND TITLE-ABS-KEY (("action" OR "operation" OR "pattern" OR "polic*" OR "rule")))

Artigos de Controle

Não identificado antes da execução da busca

3.1.4 Critérios de Inclusão e Exclusão dos Artigos

A seguir são especificados os Critérios de Inclusão (CI) e Critérios Exclusão (CE) que foram adotados para apoiar a seleção dos artigos.

Critérios de Inclusão dos Artigos:

(CI1) Trabalhos que abordam sobre instanciação de processo de software, apresentando como esta deve ser realizada; OU
(CI2) Trabalhos que abordam a forma como é realizada a criação do plano de projeto; OU
(CI3) Trabalhos que apresentam ferramental de apoio a criação do plano de projeto.

Critérios de Exclusão de Artigos:

(CE1) Trabalhos fora da área de computação; OU

(CE2) Trabalhos cujo foco de pesquisa encontra-se em outras áreas da computação que não seja a de Engenharia de Software; OU
(CE3) Propostas que não sejam aplicadas a processos de software, especificamente; OU
(CE4) Trabalhos que não sejam destinados a instanciação do processo de software; OU
(CE5) Trabalhos que não apresentam ações, operações, padrões, políticas ou regras utilizadas para a criação do plano de projeto, ou seja, criação de tarefas;
(CE6) Trabalhos indisponíveis; OU
(CE7) Trabalhos não escritos em inglês.

3.1.5 Procedimentos para seleção dos estudos

A seleção dos estudos foi realizada em três etapas:

Etapa 1

Seleção preliminar das publicações: A seleção preliminar das publicações foi realizada por meio da execução da *string* de busca na base pesquisa *Scopus*, selecionada para este estudo.

Etapa 2

Seleção das publicações relevantes - 1º filtro: Para uma seleção inicial das publicações relevantes, os resumos de cada artigo retornado foram lidos e avaliados de acordo com os critérios de inclusão e exclusão definidos no planejamento da busca. Em alguns artigos, apenas a leitura dos resumos, pode não ser suficiente, gerando dúvidas se deveriam ou não ser incluídos. Neste caso os artigos devem selecionados para leitura completa.

Etapa 3

Seleção das publicações relevantes - 2º filtro: A seleção final das publicações relevantes ocorreu por meio da leitura completa dos textos das publicações

selecionadas no 1º filtro. Estas publicações também foram avaliadas de acordo com os critérios de inclusão e exclusão previamente definidos.

Por questão de tempo e disponibilidade de pesquisadores para participar deste estudo, a avaliação da inclusão ou não dos artigos retornados pela busca estruturada foi realizada apenas pela autora deste texto, o que representa uma ameaça a validade

3.1.6 Campos de Extração e Avaliação da Qualidade dos Artigos

Os campos de extração de dados dos trabalhos selecionados foram definidos visando capturar dados que auxiliem na resposta à questão de pesquisa estabelecida para esta busca estruturada. Estes são listados a seguir:
(C1) Descrição de como é realizada a instanciação do processo de software;
(C2) Descrição das ações, operações, padrões, políticas ou regras utilizadas para a criação do plano de projeto, ou seja, criação de tarefas; e
(C3) Descrição de como são aplicadas as ações, operações, padrões, políticas ou regras utilizadas para a criação do plano de projeto, ou seja, criação de tarefas.

3.2 Execução da Busca Estruturada

De acordo com os procedimentos de seleção dos artigos definidos no planejamento desta busca estruturada, a primeira etapa consistiu da execução da *string* de busca na fonte selecionada. Nesta etapa foram retornados 192 estudos distintos. A Tabela 3 exibe os resultados desta 1ª etapa.

Tabela 3 Retorno da String de busca

Base de Busca	**Número de Publicações retornadas**
Scopus	199

Na etapa seguinte, de seleção dos artigos relevantes, os resumos de cada um dos 192 artigos retornados na etapa anterior foram lidos. Nesta etapa, considerado os critérios de inclusão e exclusão, 14 artigos foram selecionados..

A lista de artigos resultantes após aplicação dos 1º e 2º filtros é apresentada na Tabela 4.

Tabela 4 Lista de Artigos Selecionados

Ano	Referência
2015	VO, Tan Thuan et al. **Defining and using collaboration patterns for software process development**. In: Model-Driven Engineering and Software Development (MODELSWARD), 2015 3rd International Conference on. IEEE, 2015. p. 557-564.
2015	VO, Thuan Tan et al. **An Approach to Define and Apply Collaboration Process Patterns for Software Development**. In: International Conference on Model-Driven Engineering and Software Development. Springer International Publishing, 2015. p. 248-262.
2015	Khalil, R., Stockton, D., Alkaabi, M. S., & Manyonge, L. M. (2015). **Investigating the effect of variability in product development life cycle.** International Journal of Product Development, 20(6), 495-513.
2014	Khaiyum, S., & Kumaraswamy, Y. S. (2014, February). **Pattern of transformation of failures based on severity in real-time embedded software projects.** In Electronics and Communication Systems (ICECS), 2014 International Conference on (pp. 1-5). IEEE.
2012	Gaffo, F. H., & Barros, R. M. (2012). **GAIA Risks: A risk management framework**. In Proceedings of the 25th International Conference on Computer Applications in Industry and Engineering (Vol. 1, No. 2, pp. 57-62).
2010	Lamersdorf, A., Munch, J., Torre, A. F. D., Sanchez, C. R., Heinz, M., & Rombach, D. (2010, August). **A rule-based model for customized risk identification in distributed software development projects.** In Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on (pp. 209-218). IEEE.
2010	KILLISPERGER, Peter et al. **A framework for the flexible instantiation of large scale software process tailoring**. In: International Conference on Software Process. Springer Berlin Heidelberg, 2010. p. 100-111.
2009	LIGGESMEYER, Peter et al. **Visualization of software and systems as support mechanism for integrated software project control**. In: International Conference on Human-Computer Interaction. Springer Berlin Heidelberg, 2009. p. 846-855.
2009	Lee, S. W., Gandhi, R. A., & Wagle, S. J. (2009**). Ontology-Guided Service-Oriented Architecture Composition to Support Complex and Tailorable Process Definitions**. International Journal of Software Engineering and Knowledge Engineering, 19(06), 791-821.
2007	FONTOURA, Lisandra M.; PRICE, Roberto Tom. **A Framework for Tailoring Software Process**. In: SEKE. 2007. p. 63-66.

2002	Lima Reis, C. A., Reis, R. Q., Schlebbe, H., & Nunes, D. J. (2002, July)**. A policy-based resource instantiation mechanism to automate software process management.** In Proceedings of the 14th international conference on Software engineering and knowledge engineering (pp. 795-802). ACM.
2002	Reis, C. A. L., Reis, R. Q., Schlebbe, H., & Nunes, D. J. (2002). **Resource instantiation policies for software process environments**. In Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International (pp. 53-58). IEEE.
1996	McGuire, E. G. (1996). **Factors affecting the quality of software project management: An empirical study based on the Capability Maturity Model**. Software Quality Journal, 5(4), 305-317.
1991	Madhavji, N. H., & Schafer, W. (1991). **Prism-methodology and process-oriented environment.** IEEE Transactions on Software Engineering, 17(12), 1270-1283.

Na terceira e última etapa, foi realizada a leitura do texto completo dos 14artigos. A seleção dos 3 estudos relevantes (Reis et al. (2002a), Reis et al. (2002b) e Killisperger et al. (2010)) também foi realizada de acordo com os critérios de inclusão e exclusão.

A busca foi executada em maio de 2017 e reexecutada em novembro de 2018. Na reexecução da busca não houve alteração nos trabalhos selecionados. A partir da leitura completa de todos os artigos apresentados na Tabela 4 foi realizada a análise dos dados extraídos para responder as questões de pesquisas definidas no planejamento deste estudo.

3.3 Avaliação dos Resultados

Baseado nas informações extraídas dos artigos selecionados (Reis et al. (2002a), Reis et al. (2002b) e Killisperger et al. (2010)), não foi possível identificar trabalhos que abordem ações, operações, padrões, políticas ou regras de instanciação relacionados à criação do plano de projeto, ou seja, à criação de tarefas. Muitos trabalhos eram relacionados a padrões de projeto e instanciação de *framework*. Em Reis et al. (2002a) e Reis et al. (2002b) destacam-se as políticas de instanciação de recursos, que apoiam a alocação de recurso às tarefas criadas durante a Instanciação. Em Killisperger et al. (2010) são apresentados operadores

de instanciação, que permitem instanciação antes e depois do projeto, porém eles tratam de operações aplicadas na perspectiva de processo (modelo de processo), e não na transição entre as perspectivas atuando na criação do plano de projeto (lista de tarefas) estabelecendo vínculo entre elas.

A busca estruturada permitiu-nos identificar que não existem abordagens e tecnologias disponíveis na literatura que possa apoiar aos engenheiros de software a mapear o processo definido para as tarefas do projeto. Desse modo, entende-se que as operações de instanciação propostas neste trabalho tem o potencial para apoiar de forma eficiente e eficaz o mapeamento entre essas duas perspectivas, bem como apoiar a identificação de não-conformidades.

4. Considerações Finais

As operações de instanciação proposta neste trabalho implementam mecanismos que permitem que elementos do processo de software definido estejam explícitos durante a instanciação e execução. A aplicação de operações de instanciação estabelece um vínculo entre um plano de projeto (projeto de software) e modelos de processo (processo de software). O resultado do mapeamento estabelecido possui informações que permitem verificar a conformidade entre as perspectivas do processo e do projeto.

Os próximos passos são a condução de estudos experimentais para avaliar a viabilidade das operações propostas.

REFERÊNCIAS

ADRIANSYAH, Arya; BUIJS, Joos CAM. Mining process performance from event logs. In: International Conference on Business Process Management. Springer, Berlin, Heidelberg, 2012. p. 217-218.

BAIER, T., DI CICCIO, C., MENDLING, J., & WESKE, M. (2018). Matching events and activities by integrating behavioral aspects and label analysis. Software & Systems Modeling, 17(2), 573-598.

BASILI, V.R., CALDIERA, G., ROMBACH, D. (1994) The Experience Factory. Encyclopaedia of Software Engineering, pp. 469-476.

BASTARRICA, M. C., PEROVICH, D., MARÍN, J., & RIOSECO, L. (2017, July). Process-based project management and SPI. In Proceedings of the 2017 International Conference on Software and SystemProcess (pp. 124-133). ACM.

CAMPOS, Andre LN; DE OLIVEIRA, Toacy Cavalcante. Modeling Work Processes and Software Development-Notation and Tool. In: ICEIS (3). 2011. p. 337-343.

CMMI PRODUCT TEAM, 2010, CMMI® for Development (CMMI-DEV), Version 1.3. Disponível em: http://cmmiinstitute.com/cmmi-models. Acesso em: julho/2017.

CUGOLA, Gianpaolo; GHEZZI, Carlo. Software Processes: a Retrospective and a Path to the Future. Software Process: Improvement and Practice, v. 4, n. 3, p. 101-123, 1998.

DE LEONI, Massimiliano; MARRELLA, Andrea. Aligning real process executions and prescriptive process models through automated planning. Expert Systems with Applications, v. 82, p. 162-183, 2017.

DERNIAME, Jean-Claude; KABA, Badara A.; WASTELL, David (Ed.). Software Process: Principles, Methodology, and Technology. Springer Science & Business Media, 1999.

FERREIRA, Diogo R.; SZIMANSKI, Fernando; RALHA, Célia Ghedini. Mining the low-level behaviour of agents in high-level business processes. International Journal of Business Process Integration and Management, v. 6, n. 2, p. 146-166, 2013.

FUGGETTA, Alfonso. Software process: a roadmap. In: Proceedings of the Conference on The Future of Software Engineering. 2000.

GAMMA, Erich. Design patterns: elements of reusable object-oriented software. Pearson Education India, 1995.

ISO/IEC-12207, 2008. "Systems and Software Engineering - Software Life Cycle Process", The International Organization for Standardization and the International Electrotechnical SEI. 2010. CMMI for Development, version 1.3. Improving processes for developing better products and services. Carnegie Mellon University.

Killisperger, P., Stumptner, M., Peters, G., Grossmann, G., & Stückl, T. (2010, July). A framework for the flexible instantiation of large scale software process tailoring. In *International Conference on Software Process* (pp. 100-111). Springer, Berlin, Heidelberg.

LEMOS, A. M., SABINO, C. C., LIMA, R. M., & OLIVEIRA, C. A. (2011). Using process mining in software development process management: A case study. In Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on (pp. 1181-1186). IEEE.

MAGDALENO, Andréa Magalhães; WERNER, Cláudia Maria Lima; DE ARAUJO, Renata Mendes. Reconciling software development models: A quasi-systematic review. Journal of Systems and Software, v. 85, n. 2, p. 351-369, 2012.

MARTÍNEZ-RUIZ, T., MÜNCH, J., GARCÍA, F., & PIATTINI, M. (2012). Requirements and constructors for tailoring software processes: a systematic literature review. *Software Quality Journal*, *20*(1), 229-260.

MARTINS, Paula Ventura; DA SILVA, Alberto Rodrigues. Process and project alignment methodology: A case study based analysis. Computer Science and Information Systems, v. 13, n. 3, p. 901-925, 2016.

OMG "Business Process Model and Notation (BPMN)," 2014. http://www.omg.org/spec/BPMN/2.0.2/

OMG, "Software & Systems Process Engineering Meta-Mo- del Specification," 2008. http://www.omg.org/spec/SPEM/2.0/PDF

OSTERWEIL, Leon. Software processes are software too. In: Engineering of Software. Springer, Berlin, Heidelberg, 2011. p. 323-344.

PAI, M., MCCULLOCH, M., GORMAN, J. D., PAI, N., ENANORIA, W., KENNEDY, G., ... & COLFORD, J. J. (2004). Systematic reviews and meta-analyses: an illustrated, step-by-step guide. *The National medical journal of India*, *17*(2), 86-95.

PEDREIRA, O., PIATTINI, M., LUACES, M. R., & BRISABOA, N. R. (2007). A systematic review of software process tailoring. *ACM SIGSOFT Software Engineering Notes*, *32*(3), 1-6.

PEREIRA, Eliana Beatriz; BASTOS, Ricardo Melo; DE OLIVEIRA, Toacy Cavalcante. Process tailoring based on well-formedness rules. In: SEKE. 2008. p. 185-190.

PMI, A Guide to the Project Management Body of Knowledge (PMBOK) Sixth Edition. Project Management Institute, 2017.

REIS, C. A. 2003. Uma abordagem flexível para execução de processos de software evolutivos. 2003. 267 f. PhD dissertation in Computer Science (in Portuguese), Universidade Federal do Rio Grande do Sul, Porto Alegre

REIS, C. A. L., REIS, R. Q., SCHLEBBE, H., & NUNES, D. J. (2002, July). A policy-based resource instantiation mechanism to automate software process management. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering* (pp. 795-802). ACM.

REIS, C. A. L., REIS, R. Q., SCHLEBBE, H., & NUNES, D. J. (2002). Resource instantiation policies for software process environments. In *Proceedings 26th Annual International Computer Software and Applications* (pp. 53-58). IEEE.

SOFTEX, 2016a, "MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral MPS de Software". Disponível em: http://www.softex.br/mpsbr. Acesso em: junho/2016. (Accessed: 15 June 2016)

VALLE, Arthur M.; SANTOS, Eduardo AP; LOURES, Eduardo R. Applying process mining techniques in software process appraisals. Information and software technology, v. 87, p. 19-31, 2017.

WU, Ching-Seh; SIMMONS, Dick B. Software Project Planning Associate (SPPA): a knowledge-based approach for dynamic software project planning and tracking. In: Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000. IEEE, 2000. p. 305-310.