

Melhoria de Processos de Desenvolvimento de Software Aplicando *Process Patterns*

PAULO AUGUSTO OYAMADA TAMAKI
KECHI HIRAMA

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia de Computação e Sistemas Digitais
Laboratório de Tecnologia de Software
Av. Prof. Luciano Gualberto, nº 158 - Cidade Universitária - São Paulo, SP - CEP 05508-900
{ paulo.tamaki , kechi.hirama } @poli.usp.br

Resumo: Nos últimos anos muitas empresas e organizações voltadas para o desenvolvimento de softwares investem na melhoria de seus processos de desenvolvimento. Existem diversos padrões, referências, ou modelos reconhecidos que podem ser aplicados para atingir este fim. De modo geral, esses modelos apresentam apenas metas ou estruturas necessárias para que um processo de desenvolvimento apresente excelência na qualidade de seus produtos, mas não determinam como projetar ou implantar as melhorias necessárias no processo de desenvolvimento. Para preencher esta lacuna, este artigo pretende apresentar uma contribuição para essa área, propondo um método para melhoria de processos de software aplicando *process patterns*.

Palavras-chave: Engenharia de Software, *Rational Unified Process*, *Project Management Body of Knowledge*, *Process Patterns*.

Software Processes Improvement Applying Process Patterns

Abstract: *In the past few years a wide range of software development companies and organizations have invested in software process improvement. There are different standards or reference models that can be applied to achieve this aim. Generally, these models only define goals or structures that a software development process must have in order to excel in the quality of their products, but they don't explain how to design or implement such improvements. In order to fill this gap, the present study attempts to contribute in this area, defining a method for software process improvement using process patterns.*

Keywords: *Software Engineering, Rational Unified Process, Project Management Body of Knowledge, Process Patterns.*

(Received December 19, 2006 / Accepted March 08, 2007)

1 Introdução

Muitas empresas e organizações voltadas para o desenvolvimento de softwares investem na melhoria de seus processos de desenvolvimento. A melhoria do processo traz grandes benefícios para a empresa ou organização, aumentando a qualidade de seus produtos e diminuindo os esforços para produzi-los e mantê-los [10]. Entretanto, a aplicação de um programa de melhoria de processos não é simples, pois não existe um método padronizado para a sua execução.

Visando a melhoria da qualidade dos processos de desenvolvimento de software, existem diversos padrões, referências, ou modelos reconhecidos que podem ser aplicados para uma empresa ou organização, como o ISO/IEC 15504 [12], o *Capability Maturity Model Integration* (CMMI) [6], o *Project Management Body of Knowledge* (PMBok) [16], e o *Software Engineering Body of Knowledge* (SWEBoK) [11].

De maneira geral, esses modelos apresentam apenas metas ou estruturas necessárias para que um processo de

desenvolvimento apresente excelência na qualidade de seus produtos, mas não determinam como projetar ou implantar as melhorias necessárias no processo de desenvolvimento.

Para apoiar estas melhorias, este artigo propõe a aplicação de *process patterns*. São apresentados inicialmente os seus conceitos e definições. Em seguida, é apresentada uma proposta de um modelo de melhoria de processo utilizando *process patterns*. E ao final, é apresentada uma aplicação de exemplo do modelo proposto, utilizando o *Rational Unified Process* como processo de desenvolvimento de software para ser melhorado.

2 Process Patterns

Para entender o que são *process patterns*, primeiro é preciso entender o que são processos e o que são *patterns*.

Um processo pode ser definido como uma série de ações na qual uma ou mais entradas são utilizadas para produzir uma ou mais saídas [2] [3] [4].

Processos atuam sobre domínios bastante distintos, como projetos de engenharia, produção industrial, recrutamento de funcionários, entre outros. Cada um desses domínios apresenta necessidades e regras específicas. Por esta razão, os processos apresentam diferentes características em cada domínio em que são aplicados. Entretanto, é possível notar que existem similaridades entre processos de domínios distintos.

Por exemplo, a maioria dos projetos de engenharia apresenta uma natureza flexível e dinâmica, não sendo possível encontrar um único modelo de processo capaz de atender as necessidades de todos os tipos de projeto de engenharia. Entretanto, reconhecidamente existem muitas similaridades entre diferentes tipos de projeto de engenharia. A maioria dos projetos de engenharia passa pelas seguintes grandes etapas: levantamento de requisitos, especificação, e projeto. E os engenheiros, de tipos de engenharia diferentes, apresentam uma forma de trabalho similar quando confrontados com cada uma dessas etapas [14].

Essa natureza dinâmica e flexível dos processos pode ser encontrada no *Rational Unified Process* (RUP), que promove a customização de seu *framework* de processo de desenvolvimento de software para atender as

necessidades específicas de cada projeto ou organização [13] [17]. E, da mesma maneira, as similaridades entre os processos de diversos projetos são consolidadas nas atividades, papéis e artefatos presentes em seu *framework*.

Através da observação de diferentes domínios pode-se perceber a existência de padrões e características comuns que são recorrentes entre estes domínios. Entretanto, essas características podem ser consideradas comuns apenas sob um ponto de vista mais abrangente, se cada característica for observada detalhadamente ela apresenta propriedades únicas [1].

A existência de características comuns ou padrões foi base para a criação de *design patterns* de engenharia de software [8]. Cada *design pattern* apresenta uma solução comum para um determinado tipo de problema. A descrição desta solução é feita de uma maneira abstrata, e cabe a cada arquiteto de software determinar os detalhes específicos para a aplicação de um *design pattern*.

Da mesma maneira que a engenharia de software denotou padrões nas suas estruturas, criando *design patterns*, a engenharia de processos, criou *process patterns*.

O uso de *design patterns* diminui a curva de aprendizado de um novo desenvolvedor ao ser integrado a um novo projeto [19]. Além disso, *patterns* são utilizados para captar e refinar o conhecimento de uma maneira reutilizável [18].

A aplicação de *process patterns* também apresenta benefícios. Ela tende a diminuir o tempo de aprendizado de um novo engenheiro de processo, e permite que ele se torne mais eficiente em menos tempo [5].

O uso de *process patterns* serve como mecanismo de comunicação de soluções bem sucedidas ou eficientes ao serem aplicadas. Além disso, eles servem como blocos reutilizáveis que podem ser aplicados na customização do processo de desenvolvimento de um determinado projeto ou organização [3].

Por essas razões, a aplicação de *process patterns* em conjunto com o *Rational Unified Process* se torna uma possibilidade. Pode-se aplicar *process patterns* para melhorar a customização do processo de desenvolvimento no início do projeto.

2.1 Definição de *Process Patterns*

Existem diversos estudos voltados para a área de reconhecimento de soluções padrões para determinados tipos de problemas na engenharia de processos [2] [3] [4] [5] [7] [20] [21]. Infelizmente, não existe uma proposta única para a definição e levantamento de *process patterns*, alguns dos estudos referenciados apresentam correlação entre si, mas nem todos são integrados.

Para este trabalho, *process pattern* é uma descrição de uma solução consolidada de como organizar um conjunto de atividades para resolver um problema comum na área de engenharia de processos.

Existem duas grandes vertentes que servem de base para este trabalho: a primeira apresenta *process patterns* voltados para a engenharia de software [7], a segunda apresenta um estudo abrangente com propostas mais genéricas [20].

A primeira vertente apresenta estudos voltados para o levantamento de *process patterns* específicos para a engenharia de software. Estes estudos apresentam uma estrutura para a definição do *pattern*, mas não apresentam uma notação formal para a representação esquemática dos *process patterns*, eles aceitam qualquer notação desde diagramas seguindo a especificação UML até fluxogramas [2] [3] [4] [7].

A segunda vertente apresenta um estudo que abrange *process patterns* em todos os domínios [5] [20] [21]. Ela não possui uma estrutura padronizada para a definição de um *process pattern*, mas utiliza uma notação formal para a representação dos *process patterns*: o *Business Process Modeling Notation* (BPMN). O BPMN é um padrão para a representação de processos de negócios que foi desenvolvido inicialmente pelo *Business Process Management Initiative* (BPMI) e posteriormente absorvido pelo *Object Management Group* (OMG) [15]. O BPMI é uma iniciativa que tem como objetivo definir padrões para o gerenciamento de processos de negócios.

Apesar de não haver uma unificação no estudo dos *process patterns*, as duas vertentes apresentadas seguem a mesma direção tendo conclusões similares e trabalhos complementares.

2.2 Estrutura para Documentação de um *Process Pattern*

Atualmente, não existe uma estrutura padronizada e consolidada para a documentação de um *process pattern*, a qual cada autor utiliza uma notação própria. Entretanto, todos os trabalhos relacionados aos *process patterns* apresentam uma estrutura similar.

A documentação dos *process patterns* utilizada neste trabalho é baseada nas estruturas encontradas nos principais trabalhos sobre *process patterns* [2] [3] [4] [5] [7] [20] [21] e segue o seguinte formato:

- **Nome:** Nome do *process pattern*.
- **Objetivo:** Descreve o *process pattern* em um ou dois parágrafos, fornecendo, se necessário, uma representação gráfica do *process pattern*.
- **Tipo:** Indica a classificação do *process pattern*.
- **Contexto Inicial:** Indica a situação em que o *process pattern* é aplicável. E se aplicável, quais são as pré-condições para que o processo possa ser iniciado.
- **Solução:** Descreve detalhadamente como realizar as atividades do *process pattern*, fornecendo, se necessário, uma representação gráfica do mesmo.
- **Contexto Resultante:** Indica a situação resultante após a aplicação do *process pattern*. E, se aplicável, quais são as condições para considerar que o processo foi concluído.

3 O Método de Melhoria de Processos Aplicando *Process Patterns*

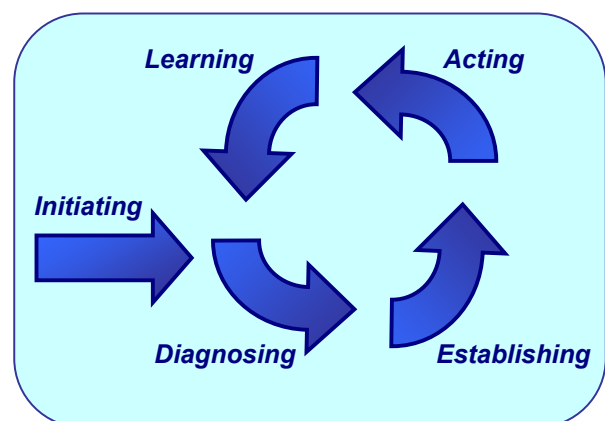


Figura 1 – Fases do Modelo IDEAL.

Conforme discutido previamente, os principais padrões, referências ou modelos aplicados em processos de

desenvolvimento de software não apresentam detalhes ou sugestões do processo necessário para implantar a melhoria de processos. Como são os casos do ISO/IEC 15504 [12], do CMMI [6], do PMBoK [16], e do SWEBoK [11].

No entanto, existem modelos, tal como o modelo IDEAL, que podem ser usados para orientar a implementação de melhorias em processos de desenvolvimento. O modelo IDEAL, proposto pelo SEI (*Software Engineering Institute*) da *Carnegie Mellon University* dos EUA, apresenta as atividades a serem realizadas para ações de melhoria [9]. Entretanto, as atividades propostas estão em um alto nível de abstração.

O modelo IDEAL é dividido em cinco fases (ilustradas na Figura 1):

- **Initiating:** Preparação das bases necessárias para que o esforço de melhoria de processos seja bem sucedido.
- **Diagnosing:** Levantamento do estado atual e definição do estado desejado.
- **Establishing:** Planejamento detalhado de como alcançar as melhorias desejadas.
- **Acting:** Execução do planejamento.
- **Learning:** Aprendizado adquirido durante o processo de melhoria.

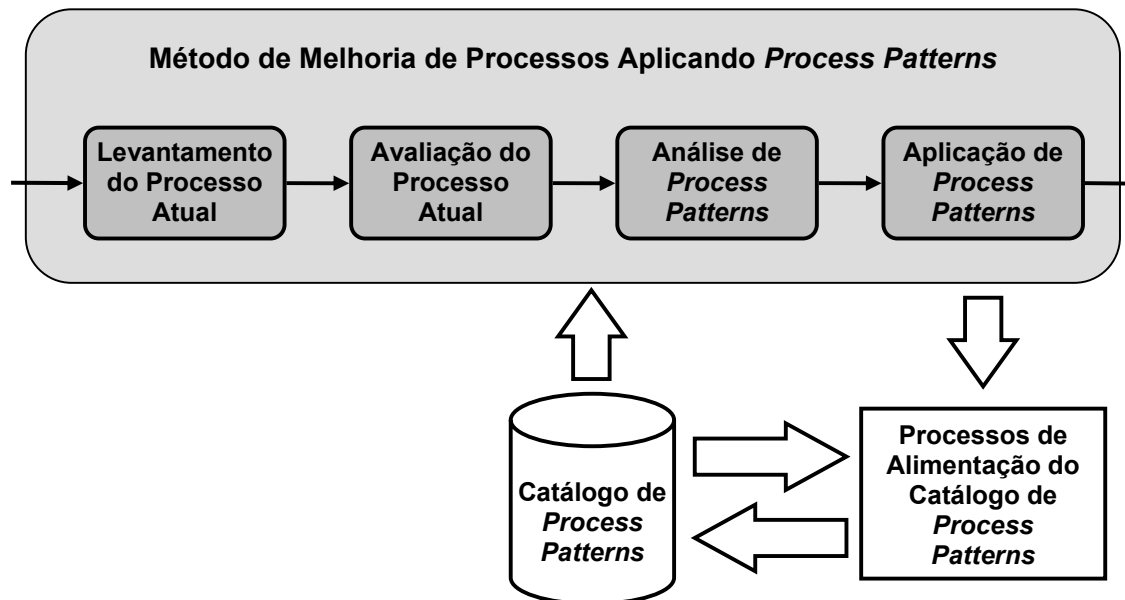


Figura 2 – Método para Melhoria de Processos Aplicando Process Patterns.

Neste contexto, os *process patterns* podem servir como ferramenta de suporte para a melhoria de processos [18] da fase de *Diagnosing* do modelo IDEAL. Esta fase possui a etapa “*Characterize Current and Desired States*” que é responsável por entender o estado atual do processo e determinar o estado futuro desejado. Durante esta etapa é possível utilizar *process patterns* para suportar a melhoria de processos.

Uma possível abordagem para a execução da etapa *Characterize Current and Desired States* aplicando *process patterns* é apresentada na Figura 2. A abordagem proposta é chamada Método de Melhoria de Processos Aplicando *Process Patterns*. Essa abordagem utiliza *process patterns* como um meio de detectar e

aplicar melhorias no processo em avaliação, e apresenta as seguintes etapas:

- **Levantamento do processo atual:** Fase de levantamento do estado atual do processo de desenvolvimento do projeto ou da organização alvo.
- **Avaliação do processo atual:** Fase de análise do processo atual, identificando deficiências e possíveis melhorias. Nessa etapa já é possível propor alterações no processo atual. Além disso, essa etapa pode detectar soluções bem sucedidas para problemas no processo de desenvolvimento, e informar essas soluções para os processos de alimentação do catálogo de *process patterns*.

- **Análise de *process patterns*:** Fase de pesquisa de possíveis *process patterns* que sejam capazes de resolver deficiências ou de trazer melhorias para o processo de desenvolvimento. Essa etapa utiliza um catálogo de *process patterns* como fonte para a procura dos *patterns* candidatos.
- **Aplicação de *process patterns*:** Fase responsável pela elaboração do novo processo aplicando os *process patterns* identificados na etapa anterior.

4 Aplicação do Método para Extensão do RUP

Para exemplificar a aplicação do método, este trabalho utiliza o *Rational Unified Process* como processo a ser melhorado através da aplicação de *process patterns*. O RUP foi selecionado, pois é um processo de desenvolvimento de software amplamente estudado e utilizado. Esses fatos simplificam a descrição do processo a ser melhorado, visto que existem diversas referências para detalhá-lo melhor, e os resultados deste estudo poderão ser incorporados, se desejado, por aqueles que utilizam o RUP.

O exemplo apresentado restringe-se apenas às atividades apresentadas na Figura 2, não discutindo todas as etapas existentes no modelo IDEAL. Este exemplo de aplicação do método é apresentado a seguir.

4.1 Levantamento do Processo Atual: O RUP

O *Rational Unified Process* (RUP) é um *framework* de processo de engenharia de software num formato que pode ser adaptado para uma grande variedade de projetos ou organizações. Ele fornece uma abordagem disciplinada para delegar tarefas e responsabilidades em uma organização de desenvolvimento de software [13] [17]. Sua meta é garantir a produção de software de alta qualidade que atenda às necessidades dos usuários dentro de um cronograma e de um orçamento previsíveis.

No RUP o projeto de software é dividido ao longo do tempo em quatro fases (iniciação, elaboração, construção, e transição). Cada uma das fases é dividida em uma ou mais iterações. Uma iteração é um ciclo de desenvolvimento completo, resultando em uma versão, um subconjunto do produto final, que cresce incrementalmente de iteração a iteração para se transformar no produto final.

Para organizar a estrutura do seu *framework* de processo de desenvolvimento, o RUP representa seu processo utilizando os seguintes elementos:

- **Papéis:** um conjunto de comportamentos ou responsabilidades que um indivíduo pode ter durante o processo de desenvolvimento.
- **Atividades:** Cada atividade é algo que um papel pode realizar e que produz um resultado significativo para o projeto.
- **Artefatos:** Um artefato é um produto de trabalho do processo. Um papel utiliza e produz artefatos ao realizar uma atividade.
- **Fluxos de Trabalho:** O fluxo de trabalho no RUP é uma seqüência de atividades que produz um resultado de valor observável.
- **Disciplinas:** Uma disciplina é um conjunto de atividades relacionadas a uma mesma área de interesse.

4.2 Avaliação do Processo Atual: Os Processos de Revisão no RUP

O RUP apresenta diversas atividades na disciplina Gerenciamento de Projeto que tem como finalidade a revisão de um conjunto de artefatos críticos para o projeto antes de sua utilização. Dentre essas atividades podem-se citar a “Revisão do Planejamento do Projeto”, “Revisão do Plano de Iteração”, “Revisão da Aceitação da Iteração”, “Revisão da Aceitação do Projeto”, dentre outras.

Cada uma dessas atividades apresenta um detalhamento específico, indicando o que deve ser realizado em cada passo, além de sugestões de como realizar determinados passos.

Entretanto, através da observação dos passos necessários para realizar cada uma dessas atividades de revisão, é possível notar um comportamento comum em seus passos. Todas as atividades de revisão citadas seguem um mesmo padrão de fluxo de trabalho. Este fluxo comum está ilustrado na Figura 3, e seus passos são os seguintes:

- **Programar revisão:** Identificar os participantes da reunião, e agendar um local e data para sua realização.

- **Distribuir materiais de revisão:** Enviar artefatos necessários para permitir que os participantes da revisão estejam preparados para a revisão.
- **Realizar revisão:** Os participantes revisam os artefatos alvo da revisão, avaliam suas deficiências e

indicam possíveis melhorias ou correções necessárias.

- **Registrar decisões da revisão:** Ao final da revisão, um documento é gerado indicando os resultados da revisão e as decisões tomadas.

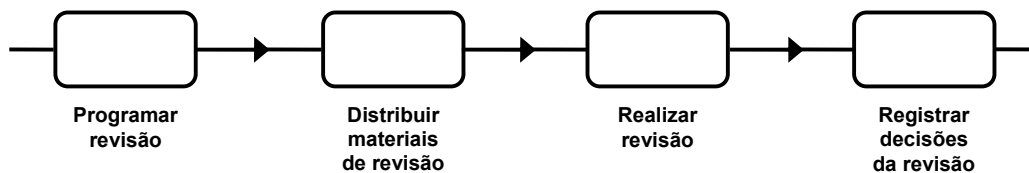


Figura 3 – As características comuns dos processos de revisão do RUP.

4.3 Análise dos *Process Patterns* Existentes

Através do estudo dos *process patterns* existentes, é possível encontrar *patterns* relacionados ao processo de revisão de documentos críticos de um projeto, como o Revisão [7], Validação em Grupo [7], e o Revisão Técnica [2] [3].

Um dos possíveis *process patterns* de revisão de documentos é apresentado a seguir, de acordo com a estrutura de documentação de um *process pattern* apresentada previamente. Este *process pattern* foi escolhido, pois foi elaborado a partir dos outros *process patterns* citados.

O *Process Pattern* Revisão Técnica [3]

Objetivo

As entregas criadas pelo projeto precisam ser validadas para garantir que elas atendem as necessidades dos usuários e seguem os padrões de qualidade da sua organização.

Tipo

Pattern de atividade.

Contexto Inicial

Existe uma ou mais entregas para serem revisadas, as entregas estão prontas para serem revisadas, e a equipe está pronta para ter as entregas revisadas.

Solução

A Figura 4 apresenta um diagrama com o processo proposto pelo *pattern* revisão técnica.

O *pattern* possui seis passos básicos que são os seguintes:

- **Preparar para revisão:** a equipe de projeto que gerou os artefatos reúne todos os itens a serem revisados de modo que eles possam ser apresentados aos revisores.
- **Indicar prontidão para a revisão:** A equipe de projeto deve informar ao gerente de revisão quando eles estarão prontos para a revisão e quais são os alvos da revisão.
- **Realizar revisão preliminar:** O gerente de revisão determina se a equipe de projeto já reuniu os itens necessários para a revisão. A principal meta é determinar se o trabalho produzido é bom o suficiente para reunir a equipe de revisão.
- **Organizar revisão:** O gerente de revisão reserva o local e quaisquer equipamentos necessários para a revisão, convoca os envolvidos, e envia quaisquer materiais que devem ser estudados antes da revisão.
- **Realizar revisão:** Os revisores avaliam os artefatos, e esclarecem quaisquer dúvidas com a equipe.
- **Atuar sobre os resultados da revisão:** Um documento é produzido durante a revisão indicando as qualidades e os defeitos dos artefatos revisados. Este documento indica o que precisa ser corrigido e sugestões de como corrigir os defeitos encontrados. Em seguida, este documento será entregue à equipe de projeto de modo que ela possa atuar sobre os artefatos e será utilizado pelo gerente de revisão para

garantir que, em futuras revisões destes artefatos, os defeitos encontrados estejam corrigidos.

Contexto Resultante

A alta gerência tem a garantia de que a qualidade dos artefatos atende as necessidades dos usuários da entrega.

Os revisores e a equipe de projeto têm um melhor entendimento dos artefatos construídos e como eles se encaixam no processo como um todo. Os membros da equipe de projeto e revisores tendem a aprender novas técnicas durante a revisão, como técnicas aplicadas nos artefatos, técnicas de revisão, ou técnicas propostas durante a revisão para a melhoria dos artefatos.

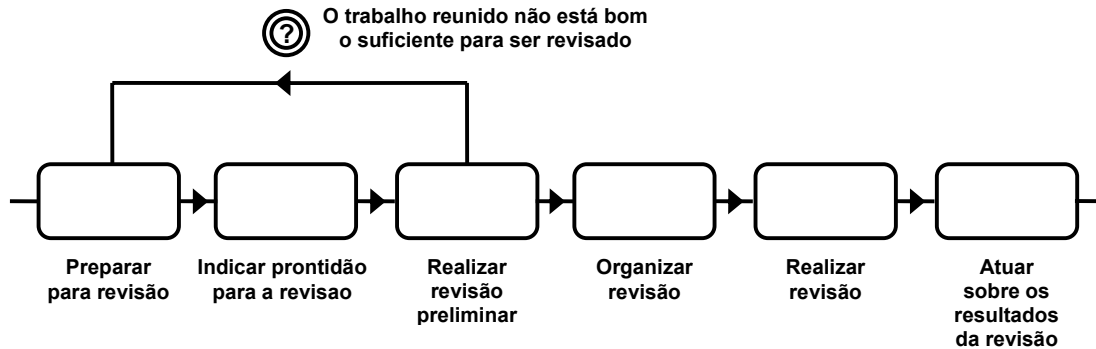


Figura 4 – O *Process Pattern* Revisão Técnica.

4.4 Aplicação de *Process Patterns*: *Process Pattern* Revisão Técnica

Através da análise do processo de revisão do *Rational Unified Process* é possível constatar que ele é consistente com o *process pattern* “Revisão Técnica”. Ambos apresentam o mesmo objetivo: revisar um documento para garantir que ele possui qualidade suficiente para ser utilizado como referência oficial para o restante do projeto ou entregue para o cliente.

No processo do RUP faltam os três primeiros passos descritos no *process pattern*. Entretanto, os passos apresentados no RUP refletem os passos finais do *process pattern*.

O correlacionamento dos passos do *Rational Unified Process* para os passos do *process pattern* Revisão Técnica é apresentado na Figura 5. Ela indica quais passos do RUP são condizentes com os passos do *process pattern*.

Para aplicar o *process pattern* “Revisão Técnica”, todas as atividades de revisão do RUP precisam ser alteradas para apresentarem os seguintes passos iniciais (conforme ilustrado na Figura 5):

- **Preparar para revisão:** a equipe de projeto que produziu os artefatos reúne todos os itens a serem revisados de modo que eles possam ser apresentados aos revisores.
- **Indicar prontidão para a revisão:** A equipe deve informar ao “Revisor do Projeto” quando eles estarão prontos para a revisão e quais são os alvos da revisão.
- **Realizar revisão preliminar:** O “Revisor do Projeto” determina se a equipe de projeto já reuniu os itens necessários para a revisão. A principal meta é determinar se o trabalho produzido é bom o suficiente para reunir a equipe de revisão.

Os outros passos das atividades de revisão permanecem iguais, visto que são condizentes com o *process pattern*.

A alteração no RUP para utilizar o *process pattern* Revisão Técnica afeta as macro-atividades assinaladas na Figura 6. A Tabela 1 apresenta todas as atividades modificadas dentro de uma macro-atividade do fluxo de trabalho da disciplina de gerenciamento de projetos do RUP.

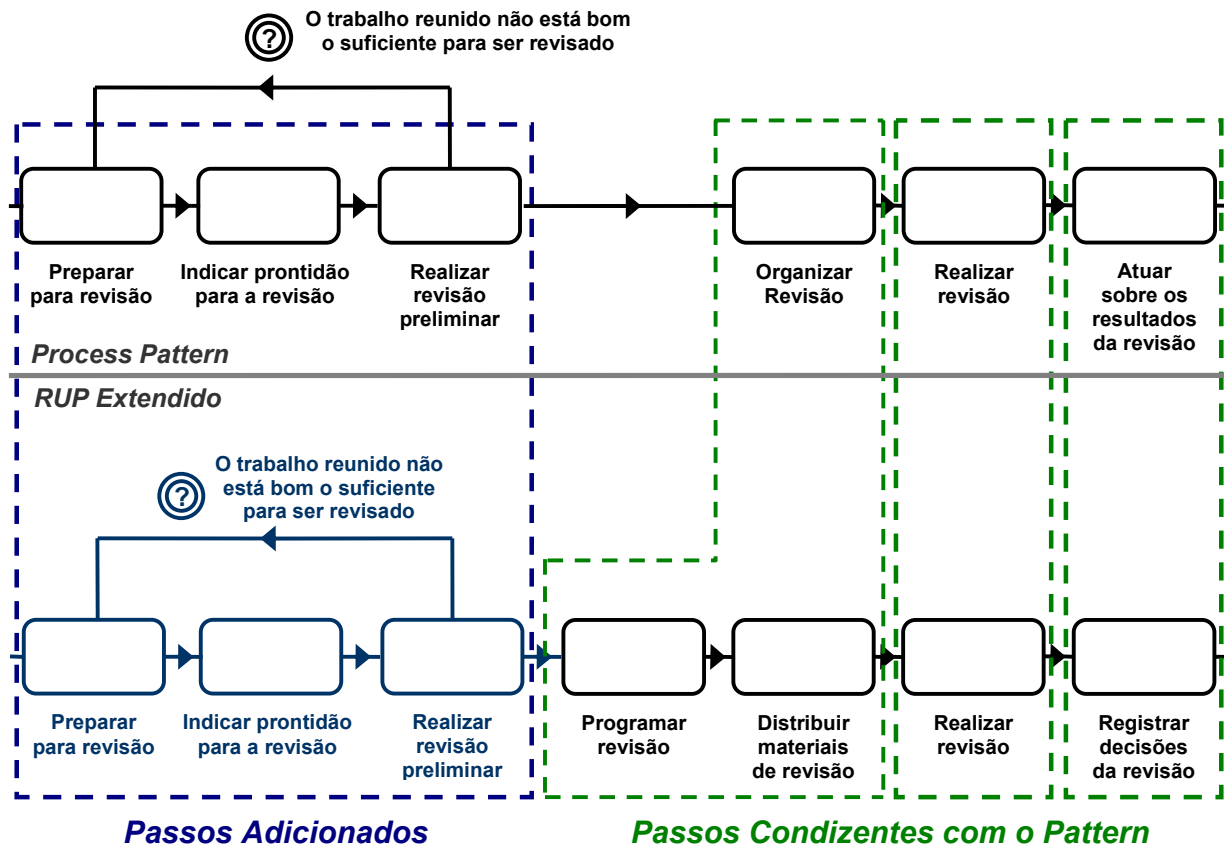


Figura 5 – Mapeamento do novo processo de revisão do RUP para o *process pattern* Revisão Técnica.

Tabela 1 – Atividades alteradas pela aplicação do *process pattern* Revisão Técnica.

Macro-atividade	Atividades Alteradas
Conceber novo projeto	Revisão da aprovação do projeto
Elaborar plano de desenvolvimento de software	Revisão do planejamento do projeto
Planejar próxima iteração	Revisão do plano de iteração
Monitorar e controlar projeto	Revisão do projeto pelo gerente de revisão do projeto
Gerenciar iteração	Revisão dos critérios de avaliação da iteração, e Revisão de aceitação da iteração
Finalizar fase	Revisão do marco do ciclo de vida
Finalizar projeto	Revisão da aceitação do projeto

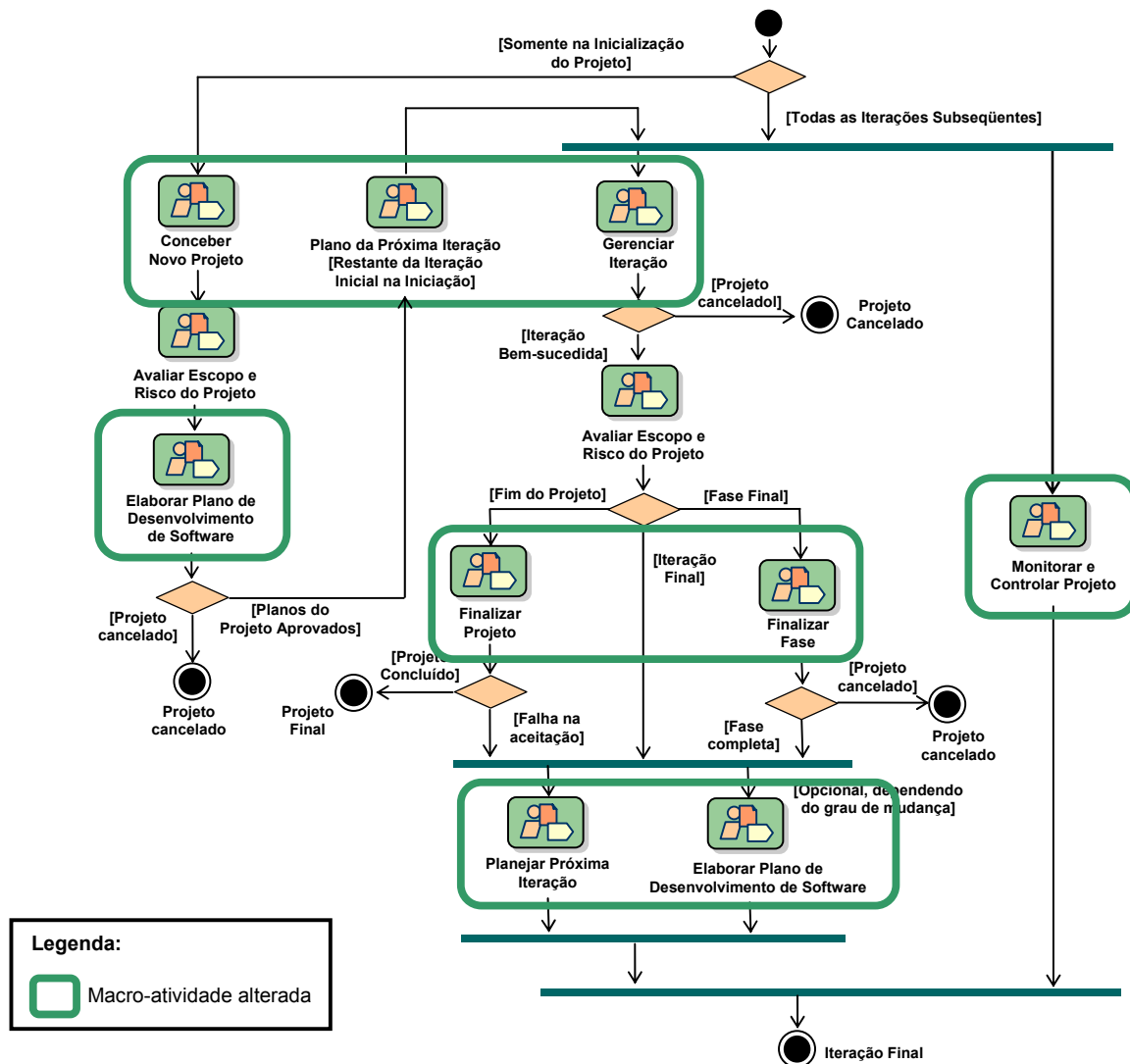


Figura 6 – Alterações no Fluxo de Trabalho do Gerenciamento de Projeto.

5 Conclusões e Perspectivas Futuras

A aplicação de *process patterns* no processo de melhoria de processos facilita o trabalho do engenheiro de processos. Apresentando soluções consolidadas para problemas no processo atual ou apresentando possíveis melhorias para o processo atual.

Na situação estudada, o principal benefício da utilização do *process pattern* no processo de revisão existente no RUP é evitar a realização de revisões onde os artefatos para revisão não têm qualidade suficiente para serem revisados, eliminando esforços desnecessários sobre a equipe do projeto e sobre a equipe de revisão do projeto.

Para permitir o sucesso da aplicação do método de melhoria de processos em geral, não somente na área de desenvolvimento de software, utilizando *process patterns* é fundamental a existência de um catálogo de *patterns* confiável. A alimentação desse catálogo é inicialmente baseada nos estudos existentes sobre *process patterns*, portanto, quanto melhor a qualidade desses estudos, melhor será a aplicação do método. O método de melhoria proposto ainda poderá ser mais detalhado através de pesquisas para a definição e estruturação do catálogo de *process patterns*.

Neste trabalho é apresentado um exemplo para ilustrar a aplicação do método e evidenciar os benefícios encontrados nesse exemplo, portanto, muitas conclusões

ainda não podem ser generalizadas a respeito do método. Pesquisas futuras com a utilização do método de melhoria proposto poderão fornecer mais evidências e detalhes, consolidando seus benefícios e restrições.

Referências Bibliográficas

- [1] Alexander, C. *The Timeless Way of Building*, New York: Oxford University Press, 1979.
- [2] Ambler, S. W. *Process Patterns: Building Large-Scale Systems Using Object Technology*. Cambridge: Cambridge University Press, 1998.
- [3] Ambler, S. W. *An Introduction to Process Patterns*. 1998. Disponível em: <<http://www.ambysoft.com/downloads/processPatterns.pdf>>. Acesso em: 17 ago. 2006.
- [4] Ambler, S. W. *More Process Patterns: Delivering Large-Scale Systems Using Object Technology*. Cambridge: Cambridge University Press, 1999.
- [5] Atwood, D. *BPM Process Patterns: Repeatable Design for BPM Process Models*. 2000. Disponível em: <<http://www.bptrends.com/publicationfiles/05-06-WP-BPMProcessPatterns-Atwood1.pdf>>. Acesso em: 16 dez. 2006.
- [6] Chrissis, M. B. Konrad, M., Shrum, S. *CMMI Guidelines for Process Integration and Product Improvement*. Addison-Wesley, 2003.
- [7] Coplien, J. A. *Development Process Generative Pattern Language in Pattern Languages of Program Design*. Addison-Wesley, 1995. Bibliografia: p 183-238.
- [8] Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1995.
- [9] Gremba, J., Myers, C. *The IDEAL Model: A practical guide for improvement*. Software Engineering Institute (SEI), 1997. Disponível em: <<http://www.sei.cmu.edu/ideal/ideal.bridge.html>>. Acesso em: 11 mar. 2007.
- [10] Harjumaa, L., Tervonen, I., Vuorio, P. *Improving Software Inspection Processes with Patterns*. In: IEEE QSIC'04 - Volume 00, 2004. Washington, DC, USA, 2004. p. 118-125.
- [11] Institute of Electrical and Electronic Engineering (IEEE). *SWEBoK – Guide to the Software Engineering Body of Knowledge*, 2004.
- [12] International Standards Organization (ISO). *ISO/IEC 15504 Standard*, 2003.
- [13] Kruchten, P. *The Rational Unified Process - An Introduction*, Addison-Wesley-Longman, 2000.
- [14] Moore, J., Inder, R., Chung, P., Macintosh, A., Stader, J. *Combining and adapting process patterns for flexible workflow*. In: *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, Washington, DC, USA: IEEE Computer Society, 2000. p. 797.
- [15] Object Management Group (OMG). *Business Process Modeling Notation Specification*, OMG, 2006.
- [16] Project Management Institute (PMI). *A Guide to the Project Management Body of Knowledge*, 2004.
- [17] Rational Software Corporation, *Rational Unified Process (Versão 2003)*, Cupertino, California: Rational Software, 2003, CD-ROM.
- [18] Rising, L. *Patterns: a way to reuse expertise*. *Communications Magazine*, IEEE, v. 37, n. 4, p. 34-36, 1999.
- [19] Schmidt, D. *Using Design Patterns to Develop Reusable Object-Oriented Communication Software*. *Communications of the ACM*, New York, NY, USA, v. 38, n. 10, p. 65-74, 1995.
- [20] van der Aalst, W. M. P., Barros, A. P., ter Hofstede, A. H. M., Kiepuszewski, B. *Advanced Workflow Patterns*. *Lecture Notes in Computer Science*, Springer-Verlag, London, UK, v. 1901, p. 18-29, 2000.
- [21] White, S. A. *Process Modeling Notation and Workflow Patterns*, IBM Corp, USA, 2004. Disponível em: <<http://www.bptrends.com/publicationfiles/03-04%20WP%20Notations%20and%20Workflow%20Patterns%20-%20White.pdf>>. Acesso em: 16 dez. 2006.