

Organizando Processos de Requisitos

Soeli T. Fiorini

Julio Cesar Sampaio do Prado Leite

Carlos José Pereira de Lucena

soeli,julio,lucena@inf.puc-rio.br

Pontifícia Universidade Católica do Rio de Janeiro — PUC-Rio

Rua Marquês de São Vicente 255, 22451-041 Gávea-RJ

Rio de Janeiro, Brasil

Resumo

Considerando que a definição e a documentação de processos é um ponto importante para a gerência de requisitos o presente artigo propõe um esquema básico para organização de processos de requisitos, cuja ênfase está na descrição de padrões. Organizando os processos em três tipos (padrão de processo, processo padrão e processo solução) procuramos fornecer informações sob três pontos de vista, de modo a facilitar a aplicação e adaptação de processos. Como damos ênfase nas descrições de padrões de processo, os apresentamos em diferentes níveis de abstração (padrão indivíduo, padrão em família e padrão em comunidade). Nas descrições de processos utiliza-se a idéia do LAL; a idéia de hipertexto; e estrutura-se mais a linguagem natural que descreve os padrões.

Palavras-chave: *padrões de processo, processo, requisitos, padrões.*

1. Introdução

Embora não seja fácil, determinar requisitos é uma atividade extremamente importante, pois os requisitos formam a base para o planejamento, o acompanhamento do desenvolvimento, e a aceitação dos resultados do projeto de software.

O processo de determinar requisitos é chamado de *processo de definição de requisitos*. Tê-lo bem definido auxilia, em muito, o entendimento do que precisa ser realizado ao desenvolver ou evoluir um determinado sistema ou componente, bem como auxilia na gerência dos requisitos. Com um processo bem definido pode-se obter (*elicitar*), entender e validar as necessidades e expectativas do cliente e, posteriormente, documentá-las (documento *Especificação de Requisitos*).

O principal objetivo do processo de definição de requisitos é concluir com êxito um acordo entre quem solicita e

quem desenvolve, estabelecendo clara e rigorosamente o que deverá ser produzido. O processo de definição de requisitos compreende basicamente três subprocessos: elicitação, modelagem e análise, que na sua utilização, são adaptados para cada caso. Um exemplo das atividades de um processo de definição de requisitos pode ser encontrado em [2] e [3].

Além de ser ter um processo de definição de requisitos é importante salientar a questão da gerência de requisitos. Esta preocupação está clara no CMM [4], um exemplo de modelo de melhoria do processo de desenvolvimento de software. Este modelo define, em seu segundo nível de maturidade, a necessidade da gerência de requisitos e estabelece as seguintes metas a serem alcançadas pela organização:

- Os requisitos alocados devem ser documentados e controlados para estabelecer uma *baseline* para engenharia de software e para uso gerencial.
- Os planos, artefatos e atividades de software devem ser mantidos consistentes com os requisitos alocados.

Quando se aborda a gerência de requisitos alguns conceitos devem ser lembrados: a evolução dos requisitos, *baseline* de requisitos e processos de requisitos.

Os requisitos possuem uma forte característica: podem mudar, evoluir durante todo o ciclo de vida de um sistema. Ou seja, desde a elicitação até a desativação do sistema os requisitos podem ser alterados, novos requisitos podem surgir e requisitos definidos podem ser excluídos devido a motivos, como por exemplo, restrições de implementação. E a proposta de manter planos, artefatos e atividades de software consistentes com os requisitos alocados torna-se uma tarefa complexa. Dessa forma, para auxiliar na gerência dos requisitos deve-se estabelecer uma *baseline* de requisitos.

Uma *baseline* é um conjunto de artefatos aceitos e controlados, e que serão utilizados em atividades posteriores à sua aceitação. A evolução da *baseline* se dá através da evolução dos artefatos existentes, exclusão de artefatos ou adição de novos artefatos e documentos, e é um indicador do progresso realizado no desenvolvimento dos trabalhos. Uma vez estabelecida uma *baseline*, a alteração da mesma (dos artefatos aceitos e aprovados) só pode ser realizada via procedimento formal. Assim, a partir do momento que alguns requisitos forem definidos pode-se estabelecer uma *baseline* de requisitos e desta forma, tem-se um controle maior sobre as alterações e as versões dos requisitos. Para maiores detalhes veja a proposta de *baseline* de requisitos em [5].

Além destes pontos é necessário conhecer os processos que tratam dos requisitos. Não se pode esperar um gerência efetiva se não se sabe realmente como as atividades são realizadas e como deveriam ser realizadas. No entanto muitas pessoas executam atividades relacionadas a requisitos sem sequer conhecer ou definir um processo de requisitos. Aprenderam um dia com alguém, que o cliente define os requisitos; deve-se realizar uma análise; os requisitos gerados serão a base para o desenvolvimento e, quando possível, devem ser documentados. Percebe-se então, que não existe um controle sobre as alterações e versões dos requisitos e imaginar um rastreamento dos requisitos é impossível. Sem contar com fato que a "análise de requisitos" (assim denominado todo e qualquer processo relacionado a requisitos; não existe claramente o conceito de processo) é um conhecimento das pessoas e não da organização. Na medida que um analista sai da organização e o processo não está documentado o conhecimento o acompanha. A definição e documentação de processos fornece dados para uma gerência efetiva na medida que medições podem ser realizadas, na medida que o acompanhamento do processo pode revelar áreas de melhoria e na medida em que é um guia para a execução das atividades.

Considerando a importância da definição e documentação de processos, propomos um esquema básico para registro e organização dos processos de requisitos. O CMM também dá esta ênfase em processos, no seu nível 3, e apresenta uma proposta de se ter um processo de software padrão da organização, onde um de seus elementos seria o processo de requisitos. Este processo padrão é então adaptado, segundo critérios e diretrizes, para definir um processo específico para cada projeto. No entanto, o CMM não define quais são estes critérios e diretrizes e não descreve como elaborá-los. Fowler [6] também apresenta uma forma de manter e disponibilizar "um conhecimento sobre requisitos", ou seja, apresenta procedimentos e outros documentos relativos a requisitos e suas práticas. Este conhecimento auxilia na definição de processos.

Como a nossa proposta envolve padrões (*patterns*), faremos, a seguir, uma breve introdução ao conceito, que está mais detalhado em [7].

1.1 Padrões (Patterns)

A idéia de um padrão é capturar práticas estabelecidas que permanecem obscuras nas práticas gerais de um dado domínio [8]. O uso deste conceito na Engenharia de Software é recente, e foi inspirado na arquitetura e planejamento urbano ([9], [10] e [11]). Algumas bibliografias tem se tornado a base de conceitos e estudos referentes a aplicação de padrões na Engenharia de Software: Gamma et al. [12], Coplien/Vlissides [13 e 14] e

Buschmann [15]. Um site dedicado ao assunto também fornece, entre outros, conceitos e terminologias, templates, bibliografias e listas de discussão sobre padrões [16].

Em Coplien [17] é abordada a questão de padrões de organização e de processo (*process and organization patterns*). Coplien comenta que o estudo de processos e organizações é um domínio maduro. Entretanto, a forma de padrões para a descrição desses processos ainda é um tema de pesquisa.

2.Proposta

Como descrito, propomos um esquema básico para registro dos processos de requisitos. Para descrevermos nossa proposta, alguns conceitos devem ser entendidos [7]:

- **Processo Padrão** - (definido a partir de [4]): descreve processos básicos (elementos fundamentais do processo) que geram o estabelecimento de um processo comum a um domínio.
- **Padrão de Processo** (definido a partir de [8] e [13]): descreve processos básicos, que associados a recomendações, definem práticas bem sucedidas e relacionadas a processos. Não é a solução, no entanto, gera uma solução através do entendimento de dos processos básicos e das recomendações.
- **Processo Solução**: é o processo que está em um nível de detalhamento que o particulariza. Ele é uma adaptação do processo padrão e foi gerado com base nas recomendações do padrão de processo. É uma solução para um determinado problema.

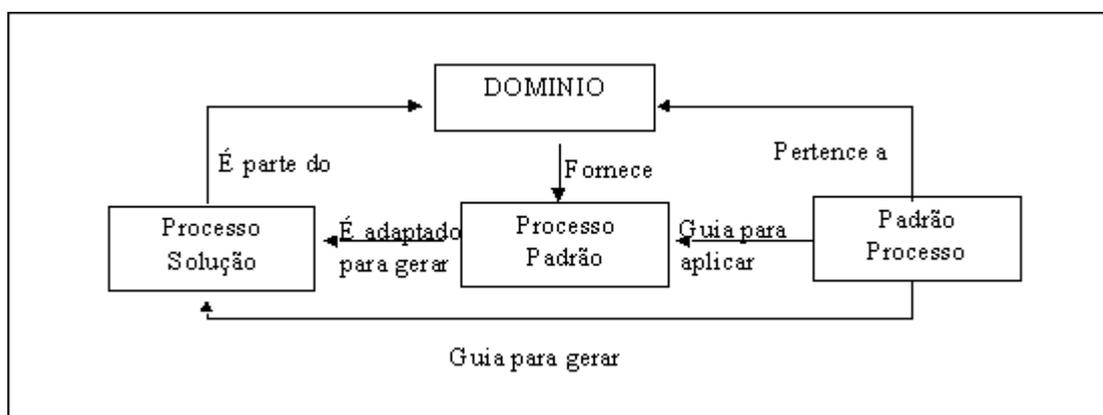


Figura 1 — Relações entre os tipos de processo [7].

Um processo padrão é um modelo que é aplicado como descrito ou é adaptado para situações específicas. Este processo padrão tem descrição (passos a serem realizados), critérios de entrada, entradas, critérios de saída, saídas e restrições. Mesmo que este processo tenha outros atributos, dificilmente se referem a, por exemplo, explicações de como adaptá-lo para uma situação *x*, qual o problema que está se tentando solucionar e as causas que levam a usar aquele processo; características de um padrão de processo. Dessa forma inserimos no nosso esquema (veja a Figura 1) as três modalidades de processos, tentando fornecer uma base de informações significativas para um completo entendimento e adaptação de processos.

2.1Proposta para Descrição de Padrões de Processo

Utilizamos três níveis de abstração através dos quais o padrão pode ser observado. Cada nível mostra o padrão inserido em um determinado contexto: o padrão indivíduo, o padrão na sua família e o padrão na comunidade de padrões [7]. O Padrão em Comunidade refere-se ao esquema de classificação das famílias de padrões [11], o Padrão em Família refere-se ao padrão relacionado com outros padrões, sendo que todos estão em um mesmo domínio, possuem objetivos comuns, e um sumário e o Padrão Indivíduo refere-se a descrição de um único padrão — o padrão em si.

2.1.1 Forma para Padrões de Processos

Descrevemos abaixo as formas propostas em [7] para padrões em família e padrão indivíduo.

2.1.1.1 Padrão em Família

Mantendo os três itens básicos de descrição de processos, de Christopher Alexander – problema, contexto e solução – e utilizando a proposta 5W1H [18], chega-se ao seguinte:

The screenshot shows a window titled "Descrição da Família" with the following fields and content:

- Identificação:** RAPPeL: A Requirements-Analysis-Process Pattern Language for Object-Oriented Development.
- Origem:** Estudos de caso na Knowledge Systems Corporation (KSC).
- Classificação:** Padrão de processo, desenv. de sw e análise de requisitos.
- Finalidade:** Guiar analistas e desenvolvedores de produto na aplicação apropriada de um conjunto de técnicas e métodos, tal que entendam o problema e produzam um análise completa...
- Problema: [O que]** Existe um problema sério e crônico em realizar análise suficiente de requisitos.
- Contexto:**
 - [Onde]** No desenvolvimento de software bem sucedido.
 - [Por quê]** Uma boa análise de requisitos, para um domínio de problema complexo, é difícil e requer muito trabalho, bem como a aplicação especial de uma variedade de técnicas.
 - [Quando]** Clientes tipicamente pensam que seus requisitos estão completos e seus problemas bem entendidos.
 - [Quem]** Os envolvidos são analistas, desenvolvedores, gerentes de projetos e clientes.

Buttons at the bottom: Mapa de Rel., Notação, OK, Cancela.

Figura 2 – Exemplo da Descrição da Família [7]

2.1.1.2 Padrão Indivíduo

Baseando-se nas quatro formas mais populares de descrições de padrões, bem como em alguns trabalhos relacionados a processos [19] [20], [21] e [22], definimos [7] uma forma na qual a Solução do padrão está muito mais detalhada do que as propostas anteriores:

| IDENTIFICAÇÃO | |
|---------------|---|
| Nome | nome que identifica o padrão de processo. |
| Problema | a questão que se quer resolver. |

| | |
|-----------------------------|--|
| CONTEXTO | |
| <i>Causas</i> | forças que levam ao problema. |
| <i>Usos conhecidos</i> | usos conhecidos do padrão. |
| CONTROLES. | |
| <i>Indicadores</i> | uma forma de medir o grau de sucesso da aplicação do padrão. Podem ser indicadores de qualidade, produtividade,... |
| SOLUÇÃO | |
| <i>componentes</i> | processos básicos que formam o padrão. Podem ser outros padrões. |
| Eles possuem: | |
| <i>Identificação</i> | Palavra/frase descrevendo o processo ou padrão. |
| <i>Motivo</i> | Por que voce está usando aquele componente. Pode fazer referência a outros padrões. |
| <i>Recomendação</i> | Dicas que devem ser observadas quando da aplicação daquele componente. Podem fazer referências a outros padrões |
| <i>Pré-condições</i> | Restrições antes dos componentes serem aplicados. |
| <i>Restrições</i> | Restrições durante a aplicação dos componentes. |
| <i>Pós-condições</i> | Restrições depois da aplicação dos componentes. |
| <i>Entradas</i> | Entradas que o componente recebe |
| <i>Saídas</i> | Saídas que o componente fornece |
| <i>Recursos</i> | todos os recursos (humanos, técnicas, métodos e ferramentas) que podem apoiar a aplicação do componente |
| PADRÕES RELACIONADOS | |
| <i>Padrões Similares</i> | menciona padrões similares ao em questão |

2.1.1.2.1 Exemplo

Como exemplo da descrição de um componente do processo de requisitos (ver na tabela acima o item solução) e do padrão indivíduo reutilizamos parcialmente o exemplo apresentado em [7] - um padrão da Família RAPPeL. No exemplo as palavras sublinhadas representam a linguagem da aplicação LAL [23] e as palavras entre "< >" representam padrões referenciados. Os itens abordados, segundo a tabela anterior, são *identificação*, *contexto* e *solução*.

IDENTIFICAÇÃO:

Nome do Padrão: Construindo as coisas certas.

Problema: Como você captura, comunica e valida requisitos de software, tal que você possa construir sistemas de sucesso que "fazem as coisas certas"?

CONTEXTO:

Causas:

- clientes freqüentemente não expressam adequadamente seus requisitos;
- desenvolvedores tem dificuldade de entender o que os clientes precisam;
- requisitos mudam, ou eles são incompletos, não bem entendidos ...

....

SOLUÇÃO:

5. Identificação: USAR MÉTODOS E ATIVIDADES DE DEFINIÇÃO DE REQUISITOS

Motivo: determinar qual o melhor uso e ligar, ou ordenar, as três atividades chaves — análise de domínio, especificação de requisitos e avaliação de requisitos — para produzir a especificação que melhor comunicará e modelará os requisitos.

Recomendação: isto é extremamente importante para incluir o uso de <protótipos> como parte chave da análise de requisitos.

Pré-condição: estabelecer e manter um acordo com o cliente; a análise de requisitos só inicia quando a investigação estiver completa e todos os envolvidos forem identificados.

Entradas: Requisitos.

Saídas: Especificação de requisitos validados; uma ou mais simulações de produtos.

3. Conclusão

Neste artigo argüimos que os processos de requisitos necessários para que uma organização possa defini-los e gerenciá-los precisam estar devidamente organizados. É sabido [24] que uma organização para ser considerada madura no que diz respeito a produção de software, deve no mínimo estar no nível 2 do modelo de maturidade (CMM), proposto pelo *Software Engineering Institute* [4]. Fundamental para atingir este patamar é a presença de uma gerência de requisitos. Por isso é fundamental que os processos de requisitos estejam organizados de maneira clara e fácil de utilizar. Acreditamos que a proposta originalmente exposta [7] e aqui adaptada é uma contribuição importante para os processos de requisitos.

Vale ressaltar que a forma de documentar processos segundo três pontos de vista (processo padrão, padrão de processo e processo solução) e segundo três níveis de detalhamento (processo indivíduo, processo em família e processo em comunidade), permite uma melhor organização e entendimento dos processos, possibilitando a adaptação dos mesmos às características específicas, por exemplo, de cada projeto.

A organização proposta é uma forma sistemática para especialistas em Engenharia de Requisitos registrarem e manterem seus conhecimentos relativos a processos de requisitos. Também é um guia para os sucessores destes especialistas e gerentes, no que se refere a execução e gerência de processos relativos a requisitos, mantendo o conhecimento na organização.

Também é importante salientar a clareza da organização na medida em que a mesma utiliza a descrição de padrões de processos em linguagem natural com base em um léxico ampliado da linguagem (LAL) [23]. Este léxico possibilita uma descrição mais precisa dos termos utilizados e facilita a possibilidade de rastreamento das informações utilizadas no processo [5].

Entendemos que nossa proposta de aplicação de um esquema baseado em padrões para organizar o processo de requisitos abre oportunidade para trabalhos futuros, tanto no aspecto de aquisição de conhecimento, através da elaboração de padrões de requisitos como também na possível integração da organização proposta com o conceito de base de requisitos (*requirements baseline*) [5], com ênfase no aspecto evolutivo. Também está claro que são necessários mais estudos de caso para validar os conceitos aqui apresentados e os futuros desenvolvimentos a partir desses.

Bibliografia

- [1] Leite, J.C.S.P.; "Elicitação de Requisitos"; *Anais do XXIII Congresso Nacional de Informática*; SUCESU – RJ, 1993.
- [2] Brackett, J.W.; Software Requirements; SEI-CM-19-1.2, *Software Engineering Institute*, January 1990.
- [3] Whitenack, B., RAPPeL: A Requirements-Analysis-Process Pattern Language for Object-Oriented Development, in *PLoP*, 1995.
- [4] Paulk, C.M., Curtis, B., Chrissis, M.B., Weber, V.C., Capability Maturity Model for Software, Ver. 1.1, *Software Engineering Institute*, Carnegie Mellon University, CMU/SEI-93-TR-24, ESC-TR-93-177, Feb 1993.
- [5] Leite J.C.S.P. et al, Enhancing a Requirements Baseline with Scenarios, *Requirements Engineering*, pp. 184-198, Springer-Verlag London, 1997.
- [6] Fowler, P., Patrick, M., Carleton, A. and Merrin, B., Transition Packages: An Experiment in Expediting the Introduction of Requirements Management, *Proceedings of ICRE - Third International Conference on Requirements Engineering*, IEEE Computer Society, pp. 138-145, Califórnia 1998
- [7] Fiorini, S.T., Leite, J.C.S.P., Lucena, C.J.P., *Descrição de Padrões de Processo em um Ambiente Automatizado para Modelagem de Processos*, Monografias em Ciência da Computação no. 16/97, ISSN 0103-9741, PUC-Rio, 1997.
- [8] Coplien, J. O., *Software Patterns*, SIGS Books & Multimedia, USA, 1996.
- [9] Alexander, C., *Notes on the Synthesis of Form*, Harvard University Press, 1964.
- [10] Alexander, C., *The Timeless Way of Building*, New York: Oxford University Press, 1979.
- [11] Buschmann, F., Meunier, R., A System of Patterns, in *PLoP*, 1995.
- [12] Gamma, E., Helm, R., Johnson, R., e Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Design*, Addison-Wesley, 1995.
- [13] Coplien, J. O., Schmidt, D.C., *Pattern Languages of Program Design*, (Primeira Conferência PLoP), Addison-Wesley, 1995.
- [14] Vlissides, J. M., Coplien, J. O., Kerth, N. L., *Pattern Languages of Program Design 2*, (Segunda Conferência PLoP), Addison-Wesley, 1996.
- [15] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., e Stal, M., *Pattern-Oriented Software Architecture: A System of Patterns*,
- [16] <http://st-ww.cs.uiuc.edu/users/patterns>
- [17] <http://c2.com/ppr/>
- [18] BR, Petrobras Distribuidora S. A., *Curso de Gerência da Qualidade Total*, 1993.
- [19] Fiorini, S.T., Leite, J.C.S.P, Macedo-Soares, T.D.L., *Integrando Processos de Negócio à Elicitação de Requisitos*, IX Simpósio Brasileiro de Engenharia de software, Recife - 1995.
- [20] Scacchi, W., *Modeling, Simulating, and Enacting Complex Organizational Process: A Life Cycle Approach*, Information and Operation Management Department, University of Southern California, 1996.
- [21] Vasconcelos, F. , Werner, C. M. L., *Software Development Process Reuse Based on Patterns*, 9th International Conference on Software Engineering and Knowledge Engineering — SEKE'97, Madrid, Espanha.
- [22] Fontoura, M. F. M. C., *Um Ambiente para Modelagem e Execução de Processos*, Dissertação de Mestrado, PUC/RJ, 1997.
- [23] Leite, J.C.S.P., Franco, A.P.M., O uso do Hipertexto na Elicitação de Linguagens da Aplicação, 4º Simpósio Brasileiro de Engenharia de Software, Águas de São Pedro, 1990.
- [24] Fiorini, S.T., Staa, A.v., Baptista, R.M., *Engenharia de Software com CMM*, Brasport, 1998.

