



engenharia  
de software

magazine

DevMedia group Ano I - Edição 04

Processo

Avaliação e Melhoria do Processo  
Organizacional Alinhada ao  
MPS.BR e PGQP

Processo

Conheça algumas Notações de Apoio à  
Modelagem de Processos de Negócio

# Metodologias Ágeis

*Uma reflexão sobre o Estado Atual*

## Planejamento

Conheça algumas Tendências na Área de Gestão de Riscos em Ambientes de Desenvolvimento de Software

## Requisitos

Desenvolvimento de Software Dirigido por Casos de Uso – Parte 3

## Metodologias Ágeis

Conheça os Principais Conceitos que Fundamentam o Scrum

## Aulas desta edição:

- Introdução à Engenharia de Requisitos - Parte 10
- Introdução à Engenharia de Requisitos - Parte 11
- Introdução à Engenharia de Requisitos - Parte 12
- Introdução ao MS Project - Parte 03
- Implementando Padrões de Projeto na Prática – Parte 1: Singleton e DAO
- Implementando Padrões de Projeto na Prática – Parte 2: MVC
- Implementando Padrões de Projeto na Prática – Parte 3: Facade e Command
- Implementando Padrões de Projeto na Prática – Parte 4: Factory Method

## Verificação, Validação & Teste

Introdução à Automação de Testes

## Ferramentas

Eclipse Process Framework: Um Ambiente de Engenharia de Software Livre para Publicar e Manter Métodos e Processos



# Já pensou em hospedar todos os seus sites em um único plano de hospedagem?

## PLANO PROFISSIONAL 1

3 domínios independentes  
50 GB de transferência  
30 GB de e-mails com SSL  
1 GB de espaço  
Painel de controle  
ASP, ASP.NET 3.5 E PHP 5  
Access ilimitado  
3 bancos MYSQL 5  
1 banco SQL Server 2005 Express

Tudo isso por apenas R\$ **25,90** por mês

**30 DIAS GRÁTIS**

Ao assinar, digite o código de desconto exclusivo para leitores desta revista e ganhe!

**RVSTMEDIA**

**Na Hospedix você ainda ganha registro de domínio (.com .net .org ou .info) grátis e isento de taxas anuais enquanto for cliente.**

Acesse agora mesmo e descubra por que Hospedix é a hospedagem de sites preferida entre os profissionais.

**[hospedix.com.br](http://hospedix.com.br)**



**HOSPEDIX**  
HOSPEDAGEM PROFISSIONAL

# engenharia de software

magazine

Ano 1 - 4ª Edição 2008 - Impresso no Brasil

## Corpo Editorial

### Colaboradores

Rodrigo Oliveira Spínola  
rodrigo@sqlmagazine.com.br

Marco Antônio Pereira Araújo  
Eduardo Oliveira Spínola

### Editor de Arte

Vinicius O. Andrade  
viniciusoandrade@gmail.com

### Diagramação

Compublix – Cia. de Publicações Especiais

### Revisão

Gregory Monteiro  
gregory@clubedelphi.net

### Na Web

www.devmedia.com.br/esmag



Apoio



## PARCEIROS:



### Rodrigo Oliveira Spínola

rodrigo@sqlmagazine.com.br

Doutorando em Engenharia de Sistemas e Computação (COPPE/UFRJ). Mestre em Engenharia de Software (COPPE/UFRJ, 2004). Bacharel em Ciências da Computação (UNIFACS, 2001). Colaborador da Kali Software (www.kalisoftware.com), tendo ministrado cursos na área de Qualidade de Produtos e Processos de Software, Requisitos e Desenvolvimento Orientado a Objetos. Consultor para implementação do MPS.BR. Atua como Gerente de Projeto e Analista de Requisitos em projetos de consultoria na COPPE/UFRJ. É Colaborador da Engenharia de Software Magazine.



### Marco Antônio Pereira Araújo

(maraujo@devmedia.com.br)

É Doutorando e Mestre em Engenharia de Sistemas e Computação pela COPPE/UFRJ – Linha de Pesquisa em Engenharia de Software, Especialista em Métodos Estatísticos Computacionais e Bacharel em Matemática com Habilitação em Informática pela UFJF, Professor dos Cursos de Bacharelado em Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora e da Faculdade Metodista Granbery, Analista de Sistemas da Prefeitura de Juiz de Fora. É editor da Engenharia de Software Magazine.



### Eduardo Oliveira Spínola

(eduspínola@gmail.com)

É Colaborador das revistas Engenharia de Software Magazine, Java Magazine e SQL Magazine. É bacharel em Ciências da Computação pela Universidade Salvador (UNIFACS) onde atualmente cursa o mestrado em Sistemas e Computação na linha de Engenharia de Software, sendo membro do GESA (Grupo de Engenharia de Software e Aplicações).

## EDITORIAL

“As metodologias ágeis como a Extreme Programming (XP) e o Scrum, entre outras, têm despertado atenção crescente do mercado. Esse movimento, baseado no ciclo de desenvolvimento incremental e iterativo, está focado na colaboração do cliente, no valor dos indivíduos e na adaptação às mudanças, tendo mostrado ganhos de produtividade nos mais diversos tipos de projetos de desenvolvimento de software.”

“A escolha da metodologia mais adequada para o desenvolvimento de software em uma organização não é uma tarefa trivial. As metodologias ágeis têm despertado o interesse do mercado, apresentando evidências de melhoria na produtividade, mas, para que possam ser efetivamente usadas em larga escala, precisam provar alguns de seus pontos de vista.”

Neste contexto, a Engenharia de Software Magazine destaca nesta edição uma matéria cujo propósito é verificar quantitativamente alguns pontos de interesse, identificando a presença das metodologias ágeis no mercado, o estado atual do movimento ágil e a pouca disponibilidade de estudos de caso que possam ser usados como fonte sistemática de resultados para comparação. Uma leitura muito interessante!

Em complemento a esta discussão sobre abordagens ágeis para apoiar o desenvolvimento de software, gostaríamos de destacar também uma matéria introdutória sobre o Scrum. O Scrum é um framework de processo ágil utilizado para gerenciar e controlar o desenvolvimento de um produto de software através de práticas iterativas e incrementais. Nesta edição você conhecerá em detalhes os principais conceitos que o norteiam.

Além destas duas matérias, esta quarta edição traz mais seis artigos:

- Modelagem de Processos de Negócio;
- Tendências na área de Gestão de Riscos em Ambientes de Desenvolvimento de Software;
- Desenvolvimento de Software Dirigido por Caso de Uso Parte III: Caso de Uso de Negócio;
- Avaliação e Melhoria do Processo Organizacional (AMP) Alinhada ao MPS.BR e PGQP;
- Introdução à Automação de Testes;
- Eclipse Process Framework: Um ambiente de Engenharia de Software livre para publicar e manter métodos e processos.

Desejamos uma ótima leitura!

Equipe Editorial Engenharia de Software Magazine

## Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



Caro Leitor,  
Para esta quarta edição, temos um conjunto de 8 vídeo aulas. Estas vídeo aulas estão disponíveis para download no Portal da Engenharia

de Software Magazine e certamente trarão uma significativa contribuição para seu aprendizado. A lista de aulas publicadas pode ser vista abaixo:

#### 01 – Engenharia de Requisitos

**Título:** Introdução à Engenharia de Requisitos - Parte 10

**Autor:** Rodrigo Oliveira Spínola

**Mini-Resumo:** Esta vídeo aula é parte de um curso de introdução à engenharia de requisitos. Nesta décima parte. São apresentados conceitos relacionados aos requisitos de domínio.

#### 02 – Engenharia de Requisitos

**Título:** Introdução à Engenharia de Requisitos - Parte 11

**Autor:** Rodrigo Oliveira Spínola

**Mini-Resumo:** Esta vídeo aula é parte de um curso de introdução à engenharia de requisitos. Nesta décima primeira parte iremos entender os principais conceitos relacionados a diagramas de casos de uso.

#### 03 – Engenharia de Requisitos

**Título:** Introdução à Engenharia de Requisitos - Parte 12

**Autor:** Rodrigo Oliveira Spínola

**Mini-Resumo:** Esta vídeo aula é parte de um curso de introdução à engenharia de requisitos. Nesta décima segunda parte daremos continuidade à discussão sobre os principais conceitos relacionados a diagramas de casos de uso.

#### 04 – Planejamento e Gerência de Projetos

**Título:** Introdução ao MS Project - Parte 03

**Autor:** Rodrigo Oliveira Spínola

**Mini-Resumo:** Esta vídeo aula apresenta as funcionalidades básicas do MS Project. Nesta terceira parte veremos como trabalhar com o conceito de baseline e caminho crítico.

#### 05 – Projeto

**Título:** Implementando Padrões de Projeto na Prática – Parte 1

**Autor:** Marco Antônio Pereira Araújo

**Mini-Resumo:** Esta vídeo aula apresenta a implementação prática dos padrões de projeto Singleton e DAO (Data Access Object) numa aplicação para a Web.

#### 06 – Projeto

**Título:** Implementando Padrões de Projeto na Prática – Parte 2

**Autor:** Marco Antônio Pereira Araújo

**Mini-Resumo:** Esta vídeo aula apresenta a implementação prática do padrão de projeto MVC (Model – View – Controller) numa aplicação para a Web.

#### 07 – Projeto

**Título:** Implementando Padrões de Projeto na Prática – Parte 3

**Autor:** Marco Antônio Pereira Araújo

**Mini-Resumo:** Esta vídeo aula apresenta a implementação prática dos padrões de projeto Facade e Command numa aplicação para a Web.

#### 08 – Projeto

**Título:** Implementando Padrões de Projeto na Prática – Parte 4

**Autor:** Marco Antônio Pereira Araújo

**Mini-Resumo:** Esta vídeo aula apresenta a implementação prática do padrão de projeto Factory Method numa aplicação para a Web.

#### Atendimento ao Leitor

A DevMedia conta com um departamento exclusivo para o atendimento ao leitor. Se você tiver algum problema no recebimento do seu exemplar ou precisar de algum esclarecimento sobre assinaturas, exemplares anteriores, endereço de bancas de jornal, entre outros, entre em contato com:

**Carmelita Mulin** – Atendimento ao Leitor  
www.devmedia.com.br/central/default.asp  
(21) 2220-5375

**Kaline Dolabella**  
Gerente de Marketing e Atendimento  
kalined@terra.com.br  
(21) 2220-5375

#### Publicidade

Para informações sobre veiculação de anúncio na revista ou no site entre em contato com:

**Kaline Dolabella**  
publicidade@devmedia.com.br

#### Fale com o Editor!

É muito importante para a equipe saber o que você está achando da revista: que tipo de artigo você gostaria de ler, que artigo você mais gostou e qual artigo você menos gostou. Fique a vontade para entrar em contato com os editores e dar a sua sugestão!

Se você estiver interessado em publicar um artigo na revista ou no site SQL Magazine, entre em contato com os editores, informando o título e mini-resumo do tema que você gostaria de publicar:

**Rodrigo Oliveira Spínola - Colaborador**  
editor@sqlmagazine.com.br

## ÍNDICE

06 - Avaliação e Melhoria do Processo Organizacional (AMP)

*Isabel Albertuni e Josiane Brietzke Porto*

12 - Modelagem de Processos de Negócio

*André Luiz de Castro Leal e José Luis Braga*

22 - Metodologias Ágeis

*André Luiz Banki e Sérgio Akio Tanaka*

30 - Por que SCRUM?

*Isabella Fonseca e Alberto Campos*

36 - Tendências na área de Gestão de Riscos em Ambientes de Desenvolvimento de Software

*Cristine Gusmão*

44 - Desenvolvimento de Software Dirigido por Caso de Uso

*Vinicius Lourenço de Sousa*

50 - Eclipse Process Framework

*Gustavo Serafim*

60 - Introdução à Automação de Testes

*Cristiano Caetano*

# Cursos Online



A revista **Java Magazine** oferece para seus assinantes uma série de **Cursos Online** de alto padrão de qualidade.

**Conheça abaixo o curso já disponível.**

Curso em destaque

## Introdução ao desenvolvimento para celulares com J2ME

Confira neste curso os principais recursos do J2ME. Aprenda também com o passo a passo para criar sua primeira aplicação J2ME. Neste curso você irá aprender diversas funcionalidades desta tecnologia para desenvolvimento de dispositivos móveis.

Confira o plano de aula completo:  
[www.devmedia.com.br/celularesj2me](http://www.devmedia.com.br/celularesj2me)

Assine a **Java Magazine** e comece já seu treinamento!  
[www.devmedia.com.br/assine](http://www.devmedia.com.br/assine)

A sua melhor opção de aprendizagem!

Outros cursos disponíveis: [www.devmedia.com.br/curso](http://www.devmedia.com.br/curso)



**DevMedia**  
GRP

Mais Informações: [www.devmedia.com.br/central](http://www.devmedia.com.br/central) - Tel.: 2220-5375 / 2220-5435

# Avaliação e Melhoria do Processo Organizacional (AMP)

Alinhada ao MPS.BR e PGQP



**Isabel Albertuni**

[isabel.albertuni@qualita.inf.br](mailto:isabel.albertuni@qualita.inf.br)

Graduada em Análise de Sistemas pela FARGS em 2008. Autora de artigos na área de qualidade de software (SBQS 2007 e WE-MPS.BR). Experiência de mais de 6 anos na área de TI atuando em desenvolvimento de software e de 3 anos na área de qualidade e melhoria de processos. Integrante desde 2008 do Comitê Setorial de Informática - Programa Qualidade RS. Atua desde 2006 na Qualidade Informática em projetos de melhoria de processos baseados em MPS.BR e PGQP.



**Josiane Brietzke Porto**

[josiane\\_brietzke@hotmail.com](mailto:josiane_brietzke@hotmail.com)

Pós-graduanda em Melhoria de Processos de Software pela UFLA desde 2007. Bacharel em Ciência da Computação pelo Unilasalle em 2005. Autora de artigos na área de qualidade de software (ASSE 2005, WIS 2005, CLEI Electronic Journal, 2W-MPSBR, SBQS 2007 e WE-MPS.BR). Experiência de mais de 5 anos na área de TI atuando em desenvolvimento de software e de mais de 4 anos na área de qualidade e melhoria de processos. Implementadora MR-MPS desde 2004, Certified Quality Improvement Associate (CQIA) desde 2006 e integrante desde 2008 do Comitê Setorial de Informática - Programa Qualidade RS. Atua desde 2005 na Qualidade Informática em projetos de melhoria de processos baseados em MPS.BR e PGQP.

A partir da aplicação dos processos padrão da organização e outros ativos de processo organizacional na definição, planejamento e estimativa de processos para os projetos e da execução dos processos definidos, podem ser identificadas oportunidades de melhoria nos processos padrão para identificar pontos de ajustes nos processos de acordo com as necessidades de negócio da organização. A realização sistemática de revisões nos processos, planejamento e implementação de melhorias identificadas, a partir dessas revisões e da experiência em utilizar os processos padrão da organização, é o objetivo do processo Avaliação e Melhoria do Processo Organizacional (SOFTEX, 2007b).

Diante deste contexto, as organizações têm se preocupado em alocar recursos e definir um mecanismo de definir, manter, disseminar e aprimorar seus processos principais de negócio e de apoio, normalmente, tendo como base modelos e normas nacionais e/ou internacionais de qualidade.

## **De que se trata o artigo:**

Processo de Avaliação e Melhoria do Processo Organizacional (AMP) alinhado ao MPS.BR e PGQP.

Neste artigo apresenta-se o alinhamento e a possibilidade de implementar este processo de forma integrada com estes modelos de qualidade.

## **Para que serve:**

Fornecer uma visão geral do processo de AMP sob o ponto de vista do MPS.BR e do PGQP e orientações que podem servir como referência para organizações, que possuem iniciativas de melhoria de processo e desejam implementar este processo.

## **Em que situação o tema é útil:**

Na definição e avaliação de melhorias dos processos organizacionais, para fins de melhoria contínua e alinhamento com os objetivos de negócio da organização.

Este artigo está organizado da seguinte forma: a seção 2 apresenta uma visão geral sobre o programa de Melhoria de Processo do Software Brasileiro (MPS.BR); a seção 3 descreve o Programa Gaúcho

de Qualidade e Produtividade (PGQP); a seção 4 apresenta o processo de Avaliação e Melhoria do Processo Organizacional (AMP) sob o ponto de vista do MPS.BR e PGQP; a seção 5 trata da implementação da AMP numa organização; e por fim, a seção 6 trata das considerações finais.

**MPS.BR**

Criado em dezembro de 2003 com base em lições aprendidas de outros programas mobilizadores, como o Programa de Desenvolvimento Estratégico em Informática no Brasil (DESI-BR), o programa Melhoria de Processo do Software Brasileiro (MPS.BR) visa a melhoria de desenvolvimento de *software* em micros, pequenas e médias empresas em todas as regiões do País. Além disso, busca o reconhecimento nacional e internacional como um modelo aplicável à indústria de *software*.

Conforme SOFTEX (2007a), o MPS.BR é composto pelo Modelo de Referência (MR-MPS.BR) destinado à melhoria de processo e o Método de Avaliação (MA-MPS.BR), para a avaliação de melhoria de processo de *software*. Para isso, possui como base técnica as normas ISO 12207, ISO 15504 em conformidade com *Capacity Maturity Model Integration (CMMI)*, como ilustra a **Figura 1**.

O MR-MPS é composto pelo: (i) Guia Geral que contém a descrição geral do programa MPS.BR, detalha o MR-MPS e apresenta as definições comuns necessárias para seu entendimento e aplicação; (ii) Guia de Aquisição que contém as melhores práticas de aquisição de *software* e serviços e descreve o processo de aquisição destinado a empresas que queiram adquirir ou subcontratar *software* e serviços de terceiros; (iii) Guia de Implementação que se divide em sete partes e possui orientações de como os requisitos do MR-MPS podem ser implementados pelas organizações.

Já o MA-MPS possui um Guia de Avaliação que contém o método e o processo de avaliação do programa MPS.BR e características de qualificação dos avaliadores, destinando-se às Instituições Avaliadoras (IA), avaliadores líderes e adjuntos.

O programa MPS.BR também conta com o Modelo de Negócio (MN-MPS), o qual descreve regras de negócio para: (i) tipos de implementação do MR-MPS que as instituições implementadoras podem conduzir, (ii) as avaliações com base no MA-MPS, (iii) organização de grupos de empresas para im-

plementação do MR-MPS, (iv) avaliação do MA-MPS através de Instituições Organizadoras de Grupos de Empresas, (v) certificação de consultores de Aquisição (de acordo com o Guia de Aquisição) e (vi) da realização de curso, provas, *workshops* do MPS.

Os tipos de contratação atuais são: Modelo de Negócio Cooperado (MNC) para grupos de micro, pequenas e médias empresas que visam compartilhar custos na implementação do MR-MPS e a avaliação no MA-MPS; Modelo de Negócio Específico (MNE), que corresponde ao modelo de negócio personalizado a uma única empresa que visa a implementação do Modelo MPS.

O MPS.BR caracteriza a maturidade de processos através da combinação de processos e capacidade, de acordo com a maturidade é necessário obter determinado nível de capacidade para a execução de um processo. O processo possui propósitos e resultados. Então, é necessário capacidade para atingir o propósito e resultados esperados do processo. Quanto mais complexo o processo, maior deve ser a capacidade para sua realização.

Além da capacidade e processos, o MPS.BR está distribuído em sete níveis de maturidade: G – Parcialmente Gerenciado, F – Gerenciado, E – Parcialmente Definido, D – Largamente Definido, C – Definido, B – Gerenciado Quantitativamente e A – Em Otimização.

**PGQP**

Com o intuito de melhorar produtos e serviços, economizar tempo e aperfeiçoar recursos no estado do Rio Grande do Sul (RS), em 1992 surgiu o Programa Gaúcho de Qualidade e Produtividade (PGQP), oriundo do Programa Brasileiro de Qualidade e Produtividade. Esta iniciativa alavancou um avanço significativo no desenvolvimento e crescimento nesta região e que vem ganhando espaço cada vez mais.

Através da parceria formada entre o setor público e iniciativas privadas, o PGQP permitiu a divulgação de forma democrática da filosofia e dos princípios da qualidade no estado e também a identificação e aprimoramento dos produtos e serviços das empresas gaúchas.

A competitividade e a qualificação nos serviços públicos e privados no RS demonstram o quanto o PGQP contribuiu para sua melhoria. O aprimoramento cada vez maior dos sistemas de gestão se dá também com o comprometimento do governo, empresários, trabalhadores e consumidores. Isto pode ser observado pelo reconhecimento que o RS tem em todo o Brasil como o estado que mais avançou na disseminação dos conceitos e na aplicação permanente das técnicas e ferramentas de qualidade, melhorando os resultados das organizações gaúchas (PGQP, 2008a).

O PGQP adota como referência o Modelo de Excelência da Gestão (MEG) e participa através de representantes da Rede Nacional da Gestão Rumo a Excelência da Fundação Nacional da Qualidade (FNQ). Além disto, o Sistema de Avaliação (SA) do PGQP está alinhado ao Prêmio Nacional da Qualidade (PNQ) e consiste num instrumento de diagnóstico organizacional que verifica o estágio de desenvolvimento gerencial das organizações, identifica lacunas e possibilita a elaboração do Plano de Ação do Sistema Gerencial – PASG (PGQP, 2008b).

O MEG possui níveis de maturidade, conforme **Figura 2** e constitui-se por oito critérios: 1 Liderança, 2 Estratégias e Planos, 3 Clientes, 4 Sociedade, 5 Informações e Conhecimento, 6 Pessoas, 7 Processos e 8 Resultados.

Segundo a FNQ (2008), organizações no nível Compromisso com a Excelência são consideradas iniciantes e ao adotaram o MEG conseguem mapear com clareza o

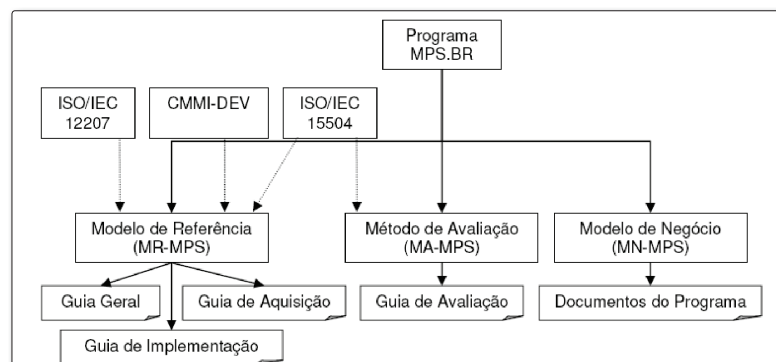


Figura 1. Componentes do MPS.BR (SOFTEX, 2007a)



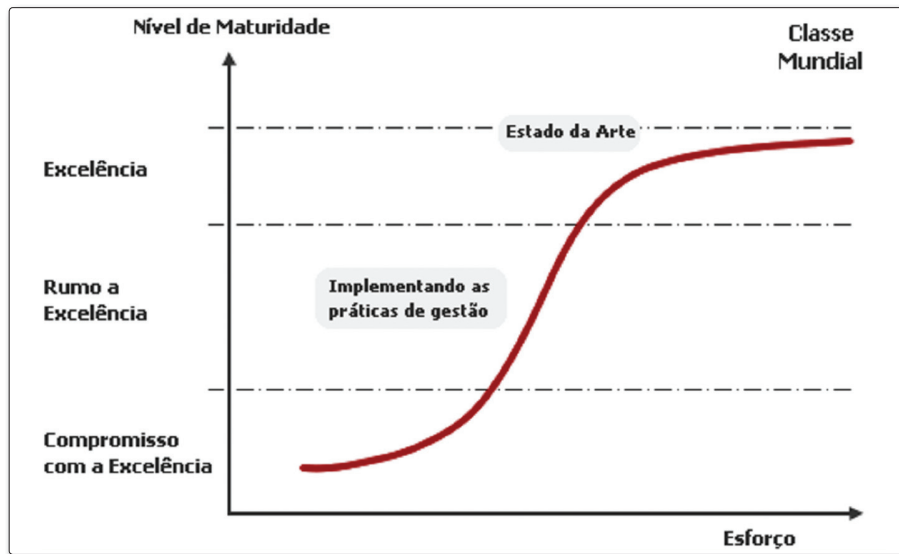


Figura 2. Evolução e estágios de maturidade da gestão da FNQ

<b>AMP 1</b> - A descrição das necessidades e os objetivos dos processos da organização são estabelecidos e mantidos;
<b>AMP 2</b> - As informações e os dados relacionados ao uso dos processos padrão para projetos específicos existem e são mantidos;
<b>AMP 3</b> - Avaliações dos processos padrão da organização são realizadas para identificar seus pontos fortes, pontos fracos e oportunidades de melhoria;
<b>AMP 4</b> - Registros das avaliações realizadas são mantidos acessíveis;
<b>AMP 5</b> - Os objetivos de melhoria dos processos são identificados e priorizados;
<b>AMP 6</b> - Um plano de implementação de melhorias nos processos é definido e executado, e os efeitos desta implementação são monitorados e confirmados com base nos objetivos de melhoria;
<b>AMP 7</b> - Ativos de processo organizacional são implantados na organização;
<b>AMP 8</b> - Os processos padrão da organização são utilizados em projetos a serem iniciados e, se pertinente, em projetos em andamento;
<b>AMP 9</b> - A implementação dos processos padrão da organização e o uso dos ativos de processo organizacional nos projetos são monitorados;
<b>AMP 10</b> - Experiências relacionadas aos processos são incorporadas aos ativos de processo organizacional.

Quadro 1. Resultados Esperados de AMP

negócio, todos conseguem compreender melhor seu papel e a direção para qual a organização caminha. Já no nível Rumo a Excelência, as organizações estão em estágio intermediário e possuem os processos delineados, postura proativa e começam a atender de forma consistente aos requisitos das partes interessadas, podendo atingir níveis de desempenho em algumas áreas superiores aos seus concorrentes. Por fim, as organizações no nível Excelência estão em estágios avançados no caminho da excelência. Possuem um sistema de gestão delineado e implementado, avaliam e melhoram de forma rotineira os seus resultados e as suas práticas de gestão, além de apresentar resultados acima dos concorrentes em várias áreas, mas por outro lado têm dificuldade em alcançar os referenciais de excelência.

Este artigo apresenta o alinhamento do processo de Avaliação e Melhoria do Processo Organizacional do MPS.BR Nível E com o nível de maturidade Compromisso com a Excelência.

### Avaliação e Melhoria do Processo Organizacional

Avaliação e Melhoria do Processo Organizacional (AMP) é o processo do MPS.BR Nível E que tem por objetivo a avaliação periódica dos processos organizacionais, apoiando a melhoria contínua com base nos seus pontos fortes e pontos fracos. Além disso, assegura que apenas serão executados os processos que realmente são necessários para a obtenção dos objetivos de negócio da organização. Para isso é muito importante entender os principais objetivos de melhoria de processos, conforme (SOFTEX, 2007b):

1. Compreender as características do processo para poder avaliar os fatores que influenciam a obtenção da capacidade esperada;
2. Planejar as ações para atender as melhorias do processo;
3. Avaliar os impactos, bem como os benefícios obtidos com relação à mudança implementada no processo.

Dentro deste contexto, o processo de AMP possui 10 resultados esperados alinhados com os objetivos de melhoria de processo, que estão listados no Quadro 1.

É importante salientar que alguns destes resultados estão ligados a outras áreas de processo do MPS.BR, como por exemplo, a Gerência de Recursos Humanos, pois ao avaliar um processo, o fator de influência que está impedindo que o processo atinja sua capacidade esperada pode estar relacionado a falta de treinamento e não oportunidades de melhoria ao processo. Nestes casos, o tratamento deve ser encaminhado de acordo com esta gerência.

Para identificar melhorias em processos, práticas como auditorias, realização de avaliações formais ou informais baseadas em MPS.BR ou CMMI colaboram para o atingimento deste resultado esperado de AMP.

Com o apoio da aplicação de avaliações, oportunidades de melhoria, pontos fracos e pontos fortes também podem ser identificados, sendo possível observar em que etapas do processo há necessidade de alterações, bem como a avaliação da exclusão do processo quando não agrega valor aos resultados de negócio.

Outra forma de identificar melhorias é a partir de reuniões de *post mortem*, realizadas ao finalizar o projeto, na qual não fica restrita somente a avaliação das atividades e do andamento do projeto encerrado, mas também dá oportunidade da equipe apontar melhorias (pontos fracos e pontos fortes) no processo aplicado no projeto.

O parecer da equipe pode e deve ser apontado não somente após concluir o projeto. Sempre que identificadas melhorias nos processos a equipe pode apontá-las quando sua resolução tenha influência para o bom andamento do projeto.

Mas nada vale a identificação da melhoria sem sua implementação. Após a identificação, planos de ação e prioridades devem ser estabelecidos para que as alterações sejam atendidas e aplicadas.

Aqui está outro relacionamento da AMP com o processo *Organizational Process Focus*



(OPF): “a implementação bem sucedida de melhorias requer participação em planejamento de ação de processo e implementação do responsável pelo processo” (SEI, 2006).

Percebe-se então que é necessário estabelecer mecanismos para identificar as melhorias, como identificá-las e que permitam que as melhorias devem ser implementadas. Mas a AMP não encerra por aqui, mecanismos de acompanhamento devem ser aplicados, pois como avaliar se as melhorias estão atingindo o resultado desejado? Novamente a OPF colabora com AMP e diz que “a monitoração garante que o conjunto de processos organizacionais está adequadamente implantado em todos os projetos” (SEI, 2006). Esta etapa fará com que se avalie criticamente a implementação das melhorias, podendo verificar se os pontos fracos foram realmente sanados.

Assim como o processo de AMP do MPS. BR, o critério 7 Processos do PGQP, também visa assegurar a melhoria, implementação e monitoramento de melhorias nos processos da organização, bem como o projeto de um novo processo. De acordo com o nível Compromisso com a Excelência (Nível 1), um dos objetivos do critério 7 é examinar como a organização identifica, gerencia, analisa e melhora os processos principais do negócio e os processos de apoio (FNQ, 2008).

No Nível 1, este critério é composto por 6 marcadores (que podem ser considerados como resultados esperados) representados pelas letras de A a F. Os marcadores que dizem respeito à melhoria de processos deste critério são (FNQ, 2008):

**a:** Como os processos principais do negócio e os processos de apoio são projetados ou modificados, visando ao cumprimento dos requisitos aplicáveis?

**b:** Como os processos principais do negócio e os processos de apoio são controlados, visando assegurar o atendimento dos requisitos aplicáveis?

**c:** Como os processos principais do negócio e os processos de apoio são analisados e melhorados?

No marcador **a** podemos observar que ao definir um novo processo ou modificar um existente, é necessário ter padrões para projetar, ou seja, necessário planejamento para a realização de um novo processo.

A definição de um processo requer avaliação prévia de recursos necessários, viabilidade de implementá-lo, entendimento do negócio e identificar em que etapa está

a organização em relação ao processo a ser desenvolvido/melhorado. Isto não quer dizer que todos os pontos serão atendidos de imediato, mas é necessário que o básico para atender o resultado esperado seja desenvolvido. Para que isso ocorra, sem deixar de implementar futuramente melhorias, é necessário ter um ciclo de melhoria de processo que contempla desde seu planejamento até a captação de melhorias.

Uma forma de realizar este ciclo é utilizando o ciclo PDCA, que surgiu na década de 1930, na escola do Controle da Qualidade Total por Walter A. Shewhart nos Estados Unidos (Skora, 2006). O PDCA ganhou forças no final da II Guerra Mundial no Japão, com apoio de Deming, estatístico americano.

Deming dizia que todo gerenciamento do processo consta em estabelecer a manutenção nas melhorias dos padrões montados na organização, que servem como referências para o seu gerenciamento, essencial para a melhoria contínua. E isto é possível com o ciclo PDCA, conforme ilustra a **Figura 3**.

O ciclo de PDCA é composto pelo conjunto de ações em seqüência dada pela ordem estabelecida pelas letras que formam a sigla: P (*plan*: planejar), D (*do*: fazer, executar), C (*check*: verificar, controlar) e o A (*act*: agir, atuar corretivamente):

- *Plan* (planejamento): nesta etapa devem ser identificados os objetivos da organização, definição do escopo do trabalho que será desenvolvido, identificação e alocação dos recursos necessários (humanos, infraestrutura, financeiros), definição de cronograma (ou somente distribuição de atividades/ações) e estabelecimento de metas;

- *Do* (execução): este é o momento de realizar o que foi planejado. Os recursos são envolvidos nas atividades para o alcance das metas e objetivos estabelecidos. Quando necessário, os envolvidos devem passar por capacitação/treinamentos para a execução das atividades;

- *Check* (controle): etapa de acompanhar e controlar o que foi desenvolvido, bem como verificar se o que foi planejado está sendo realizado, desta forma, é possível definir ações caso o realizado não esteja de acordo com o planejado;

- *Act* (agir/atuar corretivamente): última etapa do ciclo, que consiste em aplicar as ações corretivas quando forem identificados desvios na realização das

atividades, aplicação de melhorias, identificação das causas dos desvios para a definição de ações de prevenção.

Assim como o ciclo PDCA, para a melhoria de processos e definição de um novo processo, também pode ser utilizado o modelo IDEAL. Suas etapas correspondem a *Initiating* (Início), *Diagnosing* (Diagnóstico), *Establishing* (Estabelecimento), *Acting* (Ação), *Learning* (Aprendizado). Este modelo foi desenvolvido e aprimorado pelo *Software Engineering Institute* (SEI) para a melhoria de processo de *software*, que vem sendo ampliado para aplicação em diversas áreas. Outros modelos que podem ser aplicados para a melhoria de processos são o ciclo de Melhoria de processo da ISO/IEC 15504, PRO2PI-Cycle, entre outros. Veja a **Figura 4** que representa os estágios de cada ciclo.

Além de projetar um novo processo, é necessário definir mecanismos de controle para acompanhar e monitorar a aplicação dos processos. De acordo com o marcador **b**, verificar se a aplicação dos processos está sendo realizado como esperado, sendo possível a identificação de não-conformidades, ou seja, quando o processo não é executado, caracteriza-se uma não-conformidade, pois a realização deste processo atenderia uma necessidade do cliente.

Quando um processo deixa de ser executado corretamente, também se caracteriza uma não-conformidade. A não-conformidade é composta por causa e efeito. A causa é o motivo pelo qual não foi executado corretamente. O efeito é o resultado gerado pela falta da execução (Albertuni, 2008).

Quando ocorre uma não-conformidade, é necessário não só tratá-la, mas também identificar as causas pelo desvio. É necessário definir ações para prevenção do problema, não somente reativo ao problema; são formas de melhorias para o processo. A avaliação periódica permite identificar nas não-conformidades grandes oportunidades de melhoria no processo.

É importante salientar que o efeito causado por uma não-conformidade pode resultar em desvios em mais de um processo, então ao avaliar uma melhoria



**Figura 3.** Ciclo de PDCA



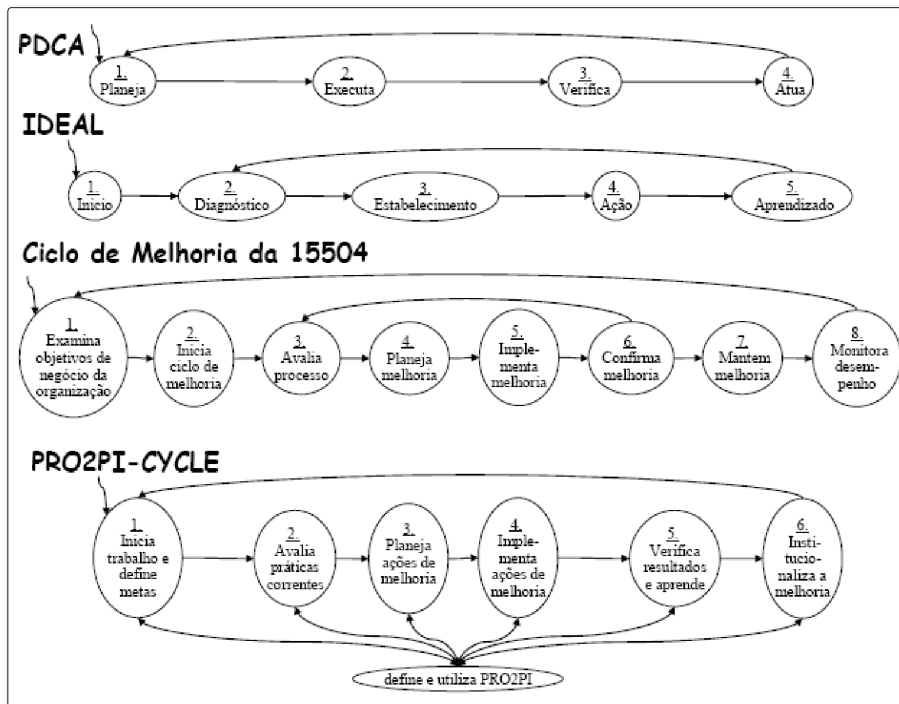


Figura 4. Fases de PDCA, IDEAL, 15504 e PRO2PI-CYCLE (SALVIANO, 2006)

para determinado processo com base na prevenção de não-conformidade, deve-se avaliar se não irá afetar outro processo que tenha algum tipo de integração.

Uso de auditorias e indicadores são exemplos de formas de identificar e controlar se a aplicação do processo está sendo feita de maneira correta, podendo visualizar os desvios.

Além de ter mecanismos para o projeto de um novo processo, o acompanhamento e melhorias não são atividades somente para os novos processos, é necessário estabelecer formas de identificação e implantação de melhorias em processos existentes. Este é o objetivo do marcador c.

A execução de processos na organização deve agregar valor aos resultados esperados destes processos. Processo é a transformação de entradas em resultados, decorrentes na saída. Se a entrada tem mais valor que a saída, o processo não é necessário para organização. Deve-se avaliar a retirada deste processo ou então sua alteração.

Quais são as maneiras de identificar melhorias? Assim como já vimos no AMP, o mesmo pode ser aplicado neste marcador, definir mecanismos para monitoramento como auditorias, avaliação informal (no caso do PGQP, auto-avaliação), retorno da equipe em relação a dificuldades na aplicação do processo, uso de indicadores são formas de identificar melhorias.

### Implementando o Processo de Avaliação e Melhoria do Processo Organizacional

O propósito do processo AMP do MPS.BR é determinar o quanto os processos padrão da organização contribuem para alcançar os objetivos de negócio da organização e para apoiar a organização a planejar, realizar e implantar melhorias contínuas nos processos com base no entendimento de seus pontos fortes e fracos (SOFTEX, 2007a). Já o propósito do critério 7 Processos do PGQP, no que diz respeito aos processos gerenciais, examina como a organização identifica, gerencia, analisa e melhora os processos principais do negócio e os processos de apoio (FNQ, 2008).

Sendo assim, percebe-se um alinhamento nos objetivos deste processo do MPS.BR e deste critério do PGQP, além de possuírem práticas complementares, quando seus resultados esperados e marcadores, respectivamente, são analisados com maior detalhe.

O Quadro 2 pretende demonstrar esta relação através de um mapeamento e também apoiar uma organização na implementação de seu processo de AMP de forma alinhada ao MPS.BR e ao PGQP. A primeira coluna mostra os resultados esperados do processo AMP do MPS.BR Nível E e nas próximas três colunas, os marcadores

do critério 7 Processos e os cruzamentos com os respectivos resultados esperados desta primeira coluna. Os cruzamentos são assinalados com "X" onde o mapeamento é objetivo e com "O", quando subjetivo, isto é, pode haver alguma discordância.

Cabe salientar que estes cruzamentos foram definidos com base na experiência, conhecimento e interpretação das autoras e não possuem a pretensão de verdade absoluta, podendo haver alguma discordância e/ou interpretação diferente do que está apresentado no Quadro 2.

A partir do Quadro 2 e das informações apresentadas na seção anterior conclui-se que ambas as referências técnicas recomendam a melhoria contínua dos processos de forma alinhada com as necessidades de negócio da organização. Para tanto exigem das organizações mecanismos de controle constantes para assegurar a agregação de valor de seus processos e a criação de uma cultura de excelência.

A seguir, um exemplo de processo de AMP alinhado ao MPS.BR e PGQP é apresentado na Figura 5. Todavia, este exemplo visa fornecer orientações de como este processo pode ser implementado numa organização e recomenda-se a sua adaptação ao contexto, características e realidade de cada organização. Este processo está representado na notação ETVX (*Entry criteria, Task, Verification, and eXit criteria*), conforme (Radice, 1988).

Um ponto importante a ser considerado em avaliação e melhoria de processos é a alocação de recursos humanos na equipe responsável pela elaboração de um novo processo ou melhoria de um processo existente. Recomenda-se que esta alocação deva ser previamente negociada entre o superior imediato do integrante da equipe e o patrocinador da iniciativa de qualidade para assegurar a sua efetiva participação.

Conforme relatado em Brietzke (2007), para a seleção dos participantes desta equipe recomenda-se que sejam selecionados de acordo com o conhecimento e experiência no processo que será desenvolvido ou melhorado (especialistas do processo). Além disso, sugere-se a participação de pelo menos um representante do Grupo de Melhoria de Processos e um integrante da alta direção para manter o alinhamento com os objetivos estratégicos da empresa e o acompanhamento das atividades.

AMP – Resultados Esperados	Processos - Marcadores		
	a	b	c
AMP 1	X		X
AMP 2		0	0
AMP 3		0	X
AMP 4			X
AMP 5	X		X
AMP 6	X	0	X
AMP 7	X		X
AMP 8	0		0
AMP 9		X	
AMP 10			X

Quadro 2. Mapeamento entre resultados esperados e marcadores

### Conclusão

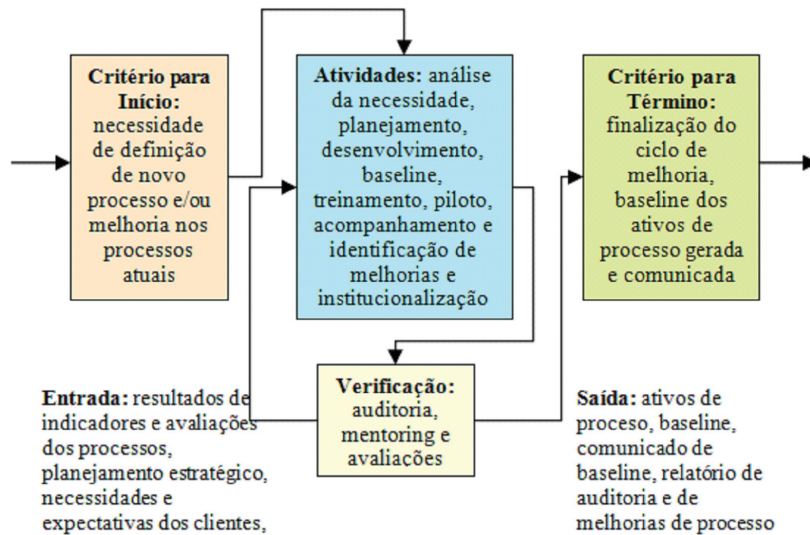
Observa-se através da implementação do processo de AMP a possibilidade de visualizar a organização como um sistema que funciona como um conjunto de processos (de negócio e de apoio) inter-relacionados e que interagem entre si. As saídas destes processos constituem entradas para um ou mais processos com o objetivo de buscar o atendimento de necessidades e expectativas dos clientes (internos ou externos).

Também se percebe que o processo de AMP de acordo com o MPS.BR está mais voltado para os processos de *software* (de negócio). E, por outro lado, a sua implementação de forma alinhada com o critério 7 Processos permite a ampliação de seu escopo para outros processos da organização, que dão sustentação aos processos principais de negócio e a si mesmos como, por exemplo, processos administrativos, de infra-estrutura, comerciais, entre outros.

Organizações que atuam ou atuaram em iniciativas de melhoria de processos possuem uma maior facilidade para implementar este processo uma vez que já possuem uma forma mesmo que informal de executar o processo de AMP, sendo necessária apenas uma análise para identificação de lacunas em relação às práticas requeridas pelo modelo e/ou norma adotado como referência pela organização (MPS.BR, PGQP, etc.), que são simples de serem resolvidas, normalmente.

Além disso, o mapeamento e o conhecimento dos processos da organização permitem a compreensão e a melhoria contínua, gerando um ambiente de cooperação, compartilhamento de informações e trabalho em equipe. ●

**Propósito:** Avaliar periodicamente os processos organizacionais a partir de seus resultados, pontos fortes e fracos, oportunidades de melhoria e implementar as melhorias identificadas de forma alinhada aos objetivos de negócio, acompanhando-as até a sua efetiva institucionalização na organização.



**Atores:** Gerentes, Equipes, Grupo de Melhoria de Processos e Alta Direção

Figura 5. Exemplo de processo de AMP

### Referências

Albertuni, Isabel. Implantação da Avaliação e Melhoria do Processo Organizacional Alinhado com Critério Processos (PGQP). Relatório de Estágio Supervisionado. Graduação em Administração com Habilitação em Análise de Sistemas, Faculdades Rio-Grandenses (FARGS), Porto Alegre, 2008.

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO (SOFTEX). MPS.BR – Guia Geral (Versão 1.2), [http://www.softex.br/mpsbr/\\_guias/MPS.BR\\_Guia\\_Geral\\_V1.2.pdf](http://www.softex.br/mpsbr/_guias/MPS.BR_Guia_Geral_V1.2.pdf), 2007a.

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO (SOFTEX). MPS.BR – Guia de Implementação (Versão 1.1), [http://www.softex.br/mpsbr/\\_guias/MPS.BR\\_Guia\\_de\\_Implementacao\\_Parte\\_3\\_v1.1.pdf](http://www.softex.br/mpsbr/_guias/MPS.BR_Guia_de_Implementacao_Parte_3_v1.1.pdf), 2007b.

Brietzke, Josiane; López, Pablo A. do Prado; Albertuni, Isabel; Richter, Luis A. A Conquista do MPS.BR Nível F na Qualidade Informática: Um Caso de Sucesso. VI Simpósio Brasileiro de Qualidade de Software, Pernambuco, 2007.

FUNDAÇÃO NACIONAL DA QUALIDADE (FNQ). Critérios Compromisso com a Excelência e Rumo a Excelência. São Paulo: Fundação Nacional da Qualidade, 2008.

PROGRAMA GAÚCHO DE QUALIDADE E PRODUTIVIDADE (PGQP). O PGQP, [http://www.mbc.org.br/mbc/pgqp/index.php?option=com\\_content&task=view&id=50&Itemid=151](http://www.mbc.org.br/mbc/pgqp/index.php?option=com_content&task=view&id=50&Itemid=151), 2008a.

PROGRAMA GAÚCHO DE QUALIDADE E PRODUTIVIDADE (PGQP). Sistema de Avaliação – Informações Gerais, [http://www.mbc.org.br/mbc/pgqp/hot\\_sites/sa2008/index.php?pagina=info](http://www.mbc.org.br/mbc/pgqp/hot_sites/sa2008/index.php?pagina=info), 2008b.

Radice, Ronald A.; Phillips, Richard W. Software Engineering, An Industrial Approach. Englewood Cliffs, New Jersey: Prentice Hall, 1988.

Salviano, Clênio Figueiredo. Melhoria e Avaliação de Processo de Software com o Modelo ISO/IEC 15504-5:2006. Curso de Pós-Graduação “Lato-Sensu” (Especialização) a Distância: Melhoria de Processo de Software. UFLA, 2006.

SEI. SOFTWARE ENGINEERING INSTITUTE. CMMI for Development (CMMI-DEV), Version 1.2, Technical report CMU/SEI-2006-TR-008. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006.

Skora, Cláudio Marlus. PDCA: o ciclo mágico, <http://www.administradores.com.br/artigos/995/>, 2006.

### Links

- MPS.BR: Site Oficial**  
[www.softex.br/mpsbr/\\_home/default.asp](http://www.softex.br/mpsbr/_home/default.asp)
- PGQP: Site Oficial**  
[www.mbc.org.br/mbc/pgqp](http://www.mbc.org.br/mbc/pgqp)
- FNQ: Site Oficial**  
[www.fpnq.org.br/site/292/default.aspx](http://www.fpnq.org.br/site/292/default.aspx)

### Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:  
[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



# Modelagem de Processos de Negócio

Uma abordagem baseada em Business Process Management Notation (BPMN), Business Process Execution Language (BPEL) e XML Process Definition Language (XPDL)



**André Luiz de Castro Leal**

*andreastr@gmail.com*

É mestrando e especialista em Ciência da Computação pela Universidade Federal de Viçosa – UFV, especialista em Gestão das Tecnologias da Informação pela Faculdade Machado Sobrinho, atua a mais de 14 anos no mercado de informática com projetos de software e atualmente coordena a equipe de desenvolvimento de sistemas computacionais do Grupo Mult unidade de Juiz de Fora. É também professor de disciplinas de Gerência de Projetos e Banco de Dados da Faculdade Estácio de Sá - JF. Áreas de Interesse: Engenharia de Software, Qualidade de Software, Processos de Desenvolvimento de Software.



**José Luis Braga**

*zeluis@dpi.ufv.br*

Pós-doutoramento em Tecnologias da Informação na University of Florida (1998-1999). Doutor em Informática - Departamento de Informática da PUC-Rio (1990). Mestre em Ciências da Computação - Departamento de Ciência da Computação da UFMG (1980). Atualmente é Professor Titular do Departamento de Informática do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa-MG. Atua na área de Ciência da Computação, com ênfase em Engenharia de Software e Sistemas de Informação. Áreas de Interesse: Qualidade de Software com Foco em Processos, Engenharia de Software Experimental, Engenharia de Software Apoiada por Ontologias, Engenharia de Software Baseada em Agentes, Sistemas de Apoio à Decisão.

Diante de um mercado competitivo e vulnerável à entrada de produtos internacionais, onde a qualidade desses produtos e o preço final estão sendo impostos por uma demanda de mercado e de uma concorrência acirrada, cada empresa necessita entender claramente cada etapa dos processos administrativos e produtivos, a fim de atender a uma demanda cada vez mais exigente de consumo.

Além do exposto, à medida que aumenta o número de fornecedores e soluções de TI no mercado corporativo, surge a necessidade de interoperabilidade e comunicação entre diferentes softwares envolvidos nos processos empresariais. A partir dessa necessidade, o mercado busca descrever um processo de negócio em um formato padronizado e inteligível tanto por analistas quanto por sistemas.

Nesse contexto, a administração empresarial está frente a novos e grandes desafios. Os empresários necessitam

## **De que se trata o artigo:**

O artigo é uma revisão bibliográfica a respeito de modelagem de processos de negócios baseada na notação BPMN e linguagens BPEL e XPDL.

## **Para que serve:**

Com o advento da reestruturação das empresas com foco nos processos de negócio, se faz premente a utilização e evolução de métodos e ferramentas que dêem suporte a modelagem dos processos empresariais, sua simulação e execução.

## **Em que situação o tema é útil:**

Para suportar a modelagem dos processos empresariais tem sido amplamente utilizado no mercado a notação BPMN e a tradução para linguagens executáveis como o BPEL e XPDL.

empreender de forma a reduzir o tempo dos processos entre a gestão e o consumidor, aumentar a transparência nos relacionamentos entre organizações e clientes através de tecnologias adequadas, reduzir a burocracia e introduzir na gestão do negócio princípios que agilizem as decisões.

Para competir nesse novo mercado é fundamental que os executivos obtenham uma visão mais completa da empresa. Entender o funcionamento de suas atividades, da interação de seus profissionais, da relação com outras organizações, da utilização e funcionamento dos recursos físicos e sistemas computacionais através de uma abordagem de toda a empresa. Assim, é possível criar uma imagem congelada e organizada da interação de cada um dos elementos envolvidos nos elos de funcionamento empresarial. Uma forma de se produzir essa imagem é criar modelos de empresa [VERN, 1996]. Segundo VERNADAT [VERN, 1996]:

- **modelo:** pode ser definido como uma representação, com maior ou menor grau de formalidade, da abstração de uma realidade expressa em algum tipo específico de formalismo;
- **modelo de empresa:** é um tipo específico de modelo formado por um conjunto de modelos que procuram representar as diferentes visões da empresa. É formado por um conjunto consistente e complementar de modelos que descrevem os aspectos de uma organização e que tem por objetivo auxiliar um ou mais usuários de uma empresa em algum propósito.

Nesse modelo de empresa é possível representar alguns elementos que irão contribuir fortemente para o entendimento empresarial, tais como:

- a funcionalidade e comportamento da empresa em termos de processos, atividades, operações básicas e eventos envolvidos;
- os sistemas computacionais e os recursos físicos necessários para seu funcionamento;
- os produtos, seus ciclos produtivos até os processos de distribuição;
- os componentes físicos ou recursos, como máquinas, ferramentas, dispositivos de armazenagem e movimentação, podendo apresentar seus *layouts*, capacidades para armazenamento ou alocação de pessoas;
- processo, fluxo e pontos das decisões que têm que ser tomadas;
- os dados e informações, seus fluxos na forma de ordens, documentos, dados discretos, arquivos de dados ou bases de dados complexas;
- conhecimento e know-how da empresa, regras específicas de decisão, políticas de gerenciamento interno, regulamentação, entre outros;

- indivíduos, especialmente suas qualificações, habilidades, regras, papéis e disponibilidades;
- responsabilidade e distribuição de autoridade sobre cada um dos elementos aqui descritos, ou seja, sobre as pessoas, materiais e funções;
- os tempos envolvidos em cada processo, uma vez que a empresa é um sistema dinâmico.

VERNADAT [VERN, 1996] cita a lista abaixo como sendo os principais benefícios da implantação de uma gestão por processos:

- construção de uma cultura, visão e linguagem compartilhadas;
- formalização do *know-how* e memória dos conhecimentos e práticas da empresa;
- suporte a decisões para melhoria e controle das operações da empresa, onde inclui-se a introdução dos recursos da tecnologia de informática como um dos principais habilitadores para esta melhoria.

Para atender a essa demanda de construção de um modelo da empresa, a *Object Management Group* (OMG) criou em novembro de 2002 o primeiro rascunho de uma notação para modelagem de processo de negócios e o denominaram de *Business Process Modeling Notation* (BPMN 0.9 Draft).

Para o melhor entendimento das abordagens técnicas que serão expostas, cabe o esclarecimento sobre o que vem a ser modelagem de processos de negócio. Modelagem de Processos de Negócios significa desenvolver diagramas (Diagramas de Processos) que mostram as atividades de uma área de negócios ou da empresa como um todo, e a sua seqüência de execução, permitindo assim o entendimento de seu funcionamento. A partir do funcionamento dos processos empresariais apresentados nos diagramas é possível verificar onde há ineficiência / ineficácia, complexidade, redundâncias e não conformidades nas atividades e remodelar o seu funcionamento a fim de se propor uma otimização no processo atual, fazendo assim um desenho completamente novo.

Para auxiliar as discussões sobre a modelagem de processos e propor uma alternativa para execução do modelo, as empresas Microsoft, IBM, Siebel, BEA e SAP resolveram se unir e desenvolver uma linguagem completa e universal denominada *Business Process Execution Language* (BPEL).

O BPEL surgiu da união de outros padrões, mais especificamente o *Web Services for Business Process Design* (XLANG) da Microsoft e o *Web Services Flow Language* (WSFL) da IBM. Após a conclusão do meta-modelo e da especificação inicial, o padrão passou para o controle da organização internacional OASIS. A *Organization for the Advancement of Structured Information Standards* (<http://www.oasis-open.org>) é um consórcio internacional, sem fins lucrativos, que está responsável pelo desenvolvimento e difusão da adoção de padrões abertos para a sociedade global da informação. Fundado em 1993, a OASIS tem 5.000 participantes representando mais de 600 organizações em 100 países.

### BPMN - Business Process Modeling Notation

O BPMN é uma notação gráfica simples e poderosa para especificação visual de processos de negócio. A abordagem foi desenvolvida para ser utilizada para a comunicação entre analistas de negócios e analistas de sistemas para o entendimento dos requisitos de negócio a serem implementados em sistemas computacionais. Na prática, o modelo se tornou tão simples que pode ser utilizado por usuários finais na modelagem de suas atividades e, conseqüentemente, a explicação formal do que se deve ser implementado.

A especificação BPMN foi desenvolvida pela *Business Modeling Integration* (BMI – [www.bpmi.org](http://www.bpmi.org)), integrada à OMG no ano de 2005. A principal missão do grupo é o desenvolvimento de especificações de modelos integrados para dar suporte a processos empresariais. Essas especificações devem promover a integração de processos internos da empresa, bem como processos externos que devem se acoplar às atividades da corporação. Assim, deve objetivar a integração e colaboração de pessoas, sistemas, processos e informações da empresa, incluindo parceiros de negócios e clientes [BPMN, 2006].

Algumas razões importantes têm feito o BPMN se sobrepor a outros padrões para modelar processos e que, na essência, não tinham de fato esse objetivo, como é o caso dos diagramas de atividades da UML, redes petri, IDEF0 e outros.







Evento	Representado por objetos de círculo e são utilizados para representar o que ocorre no curso de um processo de negócio. Os eventos representados a partir desses objetos geralmente afetam o fluxo do processo e têm uma causa, também chamado de gatilho, ou um impacto, resultado da ação do processo. Os três tipos de eventos são: Inicial, Intermediário e Final. O evento inicial é utilizado para dar início a um processo, assim todo fluxo deve apenas sair desse evento em direção a um outro elemento do BPD. O evento intermediário está entre os eventos inicial e final, e afeta o fluxo do processo, mas não inicia ou termina o processo. O evento final é o evento que finaliza o fluxo do processo e, dessa forma, não deve haver fluxo saindo desse evento.	<p>inicial</p>  <p>intermediário</p>  <p>final</p> 
Atividade	Uma atividade de um processo pode ser uma tarefa ou um sub-processo do processo principal. É representada por um retângulo com as bordas arredondadas e é um objeto que irá representar uma ação executada dentro da empresa. As figuras ao lado sugerem os dois tipos de atividades que podem ser desenhadas no BPD. Como exemplos das atividades que podem ser representadas por esses elementos podem ser citados: a emissão de faturas, a solicitação de um pedido, o fechamento de caixa, entre outros.	<p>tarefas</p>  <p>processo ou sub-processo</p> 
Gateway	Um gateway é representado por um losango e é usado para controlar a convergência e divergência de um Fluxo de Sequência. Assim, ele determinará as decisões tradicionais, bem como a quebra e junção de caminhos. Os Gateways podem representar tipos de comportamento como: desvio (branching), bifurcação (forking), fusão (merging) e junção (joining).	

Tabela 1. Objetos de Fluxo. Fonte: [BPMN, 2006]




Fluxo Sequencial	O Fluxo Sequencial é representado por uma linha e uma seta sólida e é usada para mostrar a ordem (seqüência) que as atividades serão executadas em um processo.	
Fluxo de Mensagem	O Fluxo de Mensagem é representado por uma linha tracejada com uma seta aberta e é usada para mostrar o fluxo de mensagens entre dois participantes do processo (entidades de negócio ou regras de negócio) que as enviam e as recebem. Em BPMN, dois Pools separados no diagrama representarão os dois participantes.	
Associação	Uma Associação é representada por uma linha pontilhada com uma seta aberta e é usada para associar dados, texto e outros artefatos com objetos do fluxo. Associações são usadas para mostrar as entradas e saídas das atividades.	

Tabela 2. Objetos de Conexão. Fonte: [BPMN, 2006]


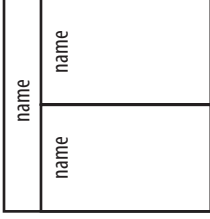
Pool	Um Pool representa o agrupamento de atividades de um participante em um determinado processo. Um exemplo seria o agrupamento das atividades de um diretor da empresa envolvido no processo que está sendo modelado. Na representação gráfica o nome do diretor ou sua função na empresa entrariam como identificador do Pool onde se encontra a descrição Name.	
Lane	Um Lane é uma sub-partição dentro de um Pool, dividindo o Pool inteiramente, horizontal ou verticalmente. São usados para organizar e categorizar atividades. Por exemplo, podem-se indicar os representantes de uma determinada área ou o processo da empresa nas raias, identificando cada participante no cabeçalho da Lane e o departamento ou o processo principal, no cabeçalho geral e único. A área seria a de suprimentos e, nas raias, seriam representadas as atividades do gerente e de um determinado fornecedor em um processo de compra.	

Tabela 3. Swimlanes. Fonte: [BPMN, 2006]

A primeira razão é a facilidade do uso de uma linguagem visual clara e de fácil entendimento. Outra razão que merece destaque é a disponibilidade de recursos gráficos que podem representar dos mais simples aos mais complexos processos de negócio. Dessa forma, permite um mapeamento dos processos agregando informações técnicas e a consequente migração para modelos de execução, como é o caso do BPEL.

Generalizando, a notação BPMN nos permite representar os seguintes conceitos:

- Processos, sub-processos e atividades;
- Loops, instanciação múltipla de atividades e transações de compensação;
- Eventos de início, de fim e intermediários no processo (por exemplo, um processo pode iniciar a partir do evento “Email vindo do cliente”);
- Decisões, paralelismo e sincronização de processos;
- Organizações, departamentos e papéis que participam do processo;
- Trocas de mensagens entre organizações participantes do processo (essencial para representar cenários *Business to Business* (B2B) onde, por exemplo, é comum a integração de diferentes plataformas de sistemas legados;
- Objetos de dados que tramitam ao longo do processo [BPMN, 2007].

Por ser uma ferramenta simples e poderosa, o BPMN tem se tornado um padrão para a modelagem de processos. Para contribuir com o cenário de consolidação dessa ferramenta no mercado, a fusão entre a criadora do BPMN, a BPMI, com a OMG (mantenedora de padrões como CORBA e UML) vem com uma proposta de incorporação do BPMN à UML. Assim, pretende-se preencher uma deficiência do modelo UML que é modelagem de processos de negócio.

Para representar o processo de negócio que está sendo modelado, BPMN define o *Business Process Diagram* (BPD), que se baseia na técnica de fluxograma. Com isso, disponibiliza uma série de figuras gráficas que representam os elementos envolvidos no processo modelado. Um modelo de processo de negócio é, então, uma rede de objetos gráficos que representam controle de fluxo que definem a ordem de execução de cada atividade do modelo. Os principais elementos compõem um BPD são descritos como [BPMN, 2006]:




- **Objetos de fluxo:** são utilizados para representar o comportamento dos processos de negócio, formando a estrutura central do BPD. Os objetos de fluxos são representados por: Eventos, Atividades e Gateways (ver Tabela 1);

- **Objetos de conexão:** são utilizados para determinar a ordem dos fluxos de mensagens, a conexão entre artefatos aos objetos do fluxo ou entre atividades, ou podem ser utilizados também para representar a troca de mensagens entre elementos do modelo. Têm três representações: Fluxos de Seqüência, de Mensagens e de Associação. A Tabela 2 contém a explicação detalhada desses objetos;

- **Swimlanes:** são utilizadas para organizar os participantes do processo ou organizar e agrupar as categorias de objetos do fluxo. O participante é chamado de pool e a raia onde serão classificados os objetos é chamada de lane. Assim, ficam representadas nos Swimlanes diferentes funcionalidades e responsabilidades, agrupadas por categorias de atividades de acordo com cada participante. A Tabela 3 contém a discriminação desses elementos;

- **Artefatos:** São utilizados para representar as informações complementares do processo. Podem ser dados, anotações ou grupos de atividades. Os artefatos estão representados na Tabela 4.

A Figura 1 apresenta um exemplo de processo de registro de reserva de viagem contendo as atividades de reserva de vôo, hotel, carro, a verificação de crédito até a confirmação final das solicitações das reservas. O processo detalhado da reserva de viagem segue os seguintes pontos:

- inicia com a recepção do pedido de reserva de viagem;
- efetua uma verificação do cartão de crédito;
- a falha do cartão, representada pela figura  na atividade "Verificação cartão de crédito", deverá emitir uma resposta para a falha de manuseio do cartão;
- as reservas são feitas para o vôo, o hotel e o carro;
- a reserva do carro pode ter que ser feita com mais de uma tentativa;
- após das três reservas serem efetuadas, é enviada uma resposta de confirmação.

Vários detalhes podem ser incorporados ao modelo podendo assim enriquecê-lo a fim de auxiliar as decisões a serem

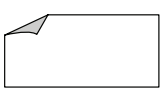
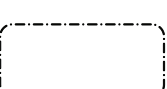
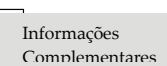
Objeto de Dados	Objetos de Dados são mecanismos que mostram como os dados são requeridos ou produzidos pelas atividades. Eles são conectados às atividades através das associações. Nessa representação podem ser representados valores totais do custo da viagem, como: o valor da reserva do hotel, o valor do aluguel do carro, taxas, parcelas, entre outros.	
Grupo	Um Grupo é representado por um retângulo com bordas arredondadas, desenhado com linhas tracejadas. Agrupamento pode ser usado com fins de documentação ou para facilitar a análise de determinado conjunto de atividades, não afetando o Fluxo Seqüencial. As atividades de registro de reserva podem ser agrupadas para facilitar a visualização dentro do modelo.	
Anotação	Anotações são mecanismos que fornecem informação textual adicional para o leitor de um diagrama BPMN. Por exemplo, para o pagamento das reservas do registro de reserva de viagem, podem ser incluídas no modelo anotações com informações complementares dos fluxos de tipo de pagamento a ser efetuado, tais como: cartão de crédito, cartão de débito ou depósito em conta.	

Tabela 4. Artefatos. Fonte: [BPMN, 2006]

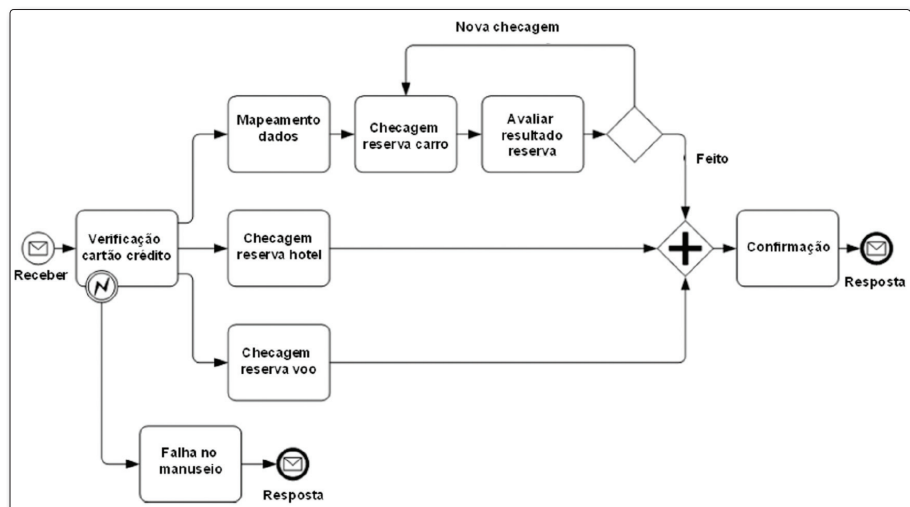


Figura 1. Processo de registro de reserva de viagem com BPMN (adaptado de [WHIT, 2005]).

tomadas e uma maior proximidade com o problema real. Como representado na Figura 2, existe um trecho de um processo com a configuração de tempo que este deverá ser executado, assim pode-se enriquecer as decisões do modelo apresentado anteriormente.

Na Figura 2 é apresentada uma pequena parte da riqueza de detalhes que pode ser incorporada ao BPMN. Dessa forma, com a caracterização de cada elemento do modelo deve ser feita criteriosamente para que todos os detalhes dos requisitos dos processos sejam configurados e possam servir como base para uma exportação para uma linguagem de execução.

Além da diagramação do modelo do processo representado no BPD, as ferramentas computacionais para modelagem

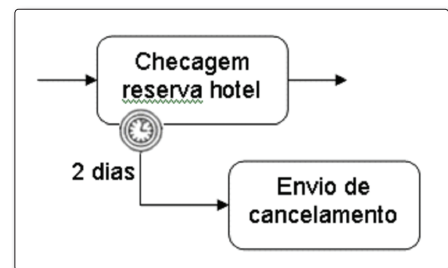
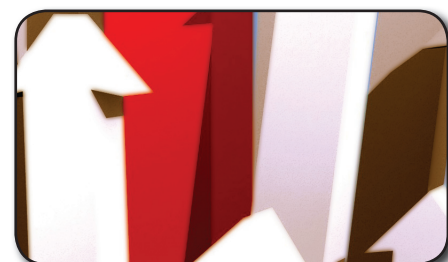


Figura 2. Configuração do tempo de resposta a um evento no processo.



permitted that they be configured textually as the properties of each element of the diagram. You can configure, for example, a programming language to which the model will be generated or its compatibility with a language, the name of the main process, or the name for each rule of the participants of the process, as well as the type of implementation to which it will be linked to that process.

Figure 3 presents the textual configuration of the model of the travel reservation process presented previously.

With the configurations implemented, it is possible to transform the model of the process into a language. Thus, it is possible to simulate the execution of the diagrammed process, or even generate XML code, which will serve for integration or communication with a computer system.

As the languages used for the execution of the process model are well-known, such as *Business Process Execution Language (BPEL)* or *XML PROCESS DEFINITION LANGUAGE (XPDL)*, they will be presented in the following sessions.

### BPEL - Business Process Execution Language

BPEL is a language based on *eXtensible Markup Language (XML)* that seeks a standard of development so that processes of different systems interact and communicate, making these activities more flexible, distributed, and independent of the platform. In addition to these objectives, BPEL is a tool to promote sequencing, coordination, and management of conversations between *Web Services* within an application of business [BOLI, 2006].

Currently, BPEL has been considered as the base for projects based on the *Service-Oriented Architecture (SOA)*. This has been a factor of motivation for companies to adopt and use *Web Services* in their computational applications.

The development of BPEL began with a consortium formed by BEA Systems, IBM, and Microsoft, which initially aimed to replace standards such as IBM's *WSFL* and Microsoft's *XLANG* [JURI, 2006]. At the moment that its meta-model and its initial specification were defined, the model was passed to the control of the international organization OASIS, which is currently responsible for its evolution.

Despite the definitions found in the literature regarding the language, there is still divergence on the part of some players regarding its concept. Oracle and IBM consider BPEL as a communication and import/export language for process execution, while Microsoft considers it as an interface for communication and import/export of process rules. Microsoft uses this foundation in its application *Biz Talk Server*, where its objective is the integration of systems and processes that can export their business rules to the BPEL model.

BPEL operation is basically tied to communication activities. With technical bases, it is possible to describe these communications starting from activities such as [JURI, 2006]:

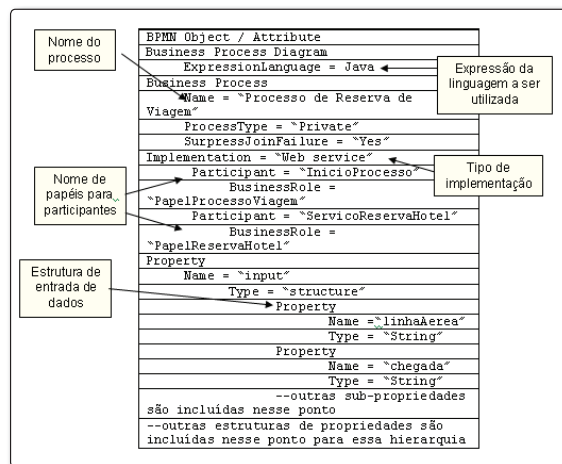


Figure 3. Textual configurations of the travel reservation process (adapted from [WHIT, 2005]).

#### Listagem 1. Inicialização e configuração do Link ServicoReservaHotel, parte do PapelProcessoViagem (adaptado de [WHIT, 2005])

```
<partnerLinks>
  <partnerLink myRole="PapelProcessoViagem" name="InicioProcesso"
    partnerLinkType="wsdl5:ProcessoViagem"/>
  <partnerLink name="ServicoReservaHotel" partnerLinkType="wsdl5:ReservaHotelPartnerPLT"
    partnerRole="PapelReservaHotel"/>
  <!-- Another 3 partnerLinks are defined -->
</partnerLinks>
```

#### Listagem 2. Definição das mensagens referentes às transações de reserva aérea (adaptado de [WHIT, 2005])

```
<message name="input">
  <part name="linhaAerea" type="xsd:string"/>
  <part name="chegada" type="xsd:string"/>
  <part name="partida" type="xsd:string"/>
  <!-- Another 10 parts are defined -->
</message>

<message name="facaSolicitacaoReservaVoo">
  <part name="linhaAerea" type="xsd:string"/>
  <part name="chegada" type="xsd:string"/>
  <part name="partida" type="xsd:string"/>
  <!-- Another four parts are defined -->
</message>

<!-- Another nine messages are defined -->
```





**Listagem 3.** Definição das variáveis correspondentes às transações com cartão de crédito (adaptado de [WHIT, 2005])

```
<variables>
  <variable messageType="wsdl0:input" name="input"/>
  <variable messageType="wsdl4:façaVerificacaoSolicitacaoCartaoCredito" name="VerificaSolicitacaoCartaoCredito"/>
  <variable messageType="wsdl4:façaVerificacaoRespostaCartaoCredito" name="VerificaRespostaCartaoCredito"/>
  <variable messageType="wsdl4:Exception" name="FalhaCartaoCredito"/>
  <variable messageType="wsdl1:façaSolicitacaoReservaCarro" name="RespostaReservaCarro"/>
  <!-- Another 6 variables are defined -->
</variables>
```

- enviar mensagens XML para serviços remotos;
- manipular estrutura de dados XML;
- receber mensagens XML assíncronas de serviços remotos;
- gerenciar eventos e exceções;
- definir seqüências paralelas de execução e retornar partes do processo quando as exceções ocorrem.

Os trechos de código das **Listagens 1, 2 e 3** representam a semântica do BPEL, a partir do exemplo do modelo de processos de registro de reserva de viagem apresentado anteriormente.

Na **Listagem 1** é representada a definição do processo a partir da configuração do *PartnerLink*. Em *PartnerLinkType* é definido o tipo de implementação para o processo, no caso um *Web Service* que terá a definição *wsdl5:ReservaHotelPartnerPLT*.

Na **Listagem 2** é apresentado o código de configuração das mensagens que serão inseridas no código BPEL para a representação do modelo do registro de reserva de viagens. Cada mensagem terá seus componentes como no caso da mensagem *Input* com os elementos de linha aérea, destino da viagem e origem da viagem, todas do tipo *string*.

Na **Listagem 3** é apresentada a utilização da mensagem *Input* definida no padrão XML acima. No caso, a mensagem fará parte das variáveis definidas para representar as transações com cartão de crédito utilizadas no modelo do registro de reserva de viagens.

Dessa forma, através de códigos XML, o BPEL consegue descrever um processo de negócio para interagir com Web Services, sejam eles internos ou externos. Isso significa definir e criar uma série de regras de fluxogramas, como seqüências, paralelismos, condicionais, loops, dentre outros, para a execução de diversos Web Services em seqüência [BOLI, 2006].

Ferramentas para implementação de soluções BPM, como o *framework* da JBoss, o BPELPM da Oracle, assim como a solução BPM Suíte da nova versão do NetBeans, são capazes de criar o modelo do fluxograma em BPEL e também de executá-lo, isso é, ler as regras definidas, executar as regras de negócio e invocar os Web Services. Alguns fornecedores como a Microsoft, por exemplo, utilizam o BPEL somente para importação e exportação de regras de fluxos para seus sistemas, similar ao XPDL.

A principal crítica ao BPEL é a falta da comunicação entre agentes participantes do processo, ou seja, as pessoas. Por exemplo, o tratamento de *workflow* para comunicação entre os participantes do processo é inexistente no modelo básico, não existe, por exemplo, um elemento ou regra que descreva uma tarefa que necessita a aprovação do diretor autorizando o fechamento do pacote de viagens para o interessado. Essa solução é resolvida pela maioria dos fornecedores com a criação de serviços (*Web Services*) de “notificação de pessoas”, que podem ser acoplados ao BPM. Nesse ponto, o modelo cai em um

antigo problema da utilização de soluções proprietárias, dificultando a portabilidade entre ferramentas.

Assim, talvez a maior questão envolvendo o BPEL, hoje, esteja relacionada a *Web Services*. O BPEL trabalha somente com *Web Services*. Há, portanto, uma incompatibilidade com arquivos texto, tabelas de banco de dados, *File Transfer Protocol* (FTP), a não ser via *Web Services* programados. Nessa situação, voltamos às soluções proprietárias de fornecedores que implementam serviços para a interoperabilidade com esses tipos de interfaces.

### XPDL - Process Definition Language

A linguagem XPDL foi proposta pela *Workflow Management Coalition* (WfMC), entidade que define padrões para a comunidade que trabalha com workflow. Essa linguagem constitui em um modelo para intercâmbio de processos de negócio entre ferramentas de modelagem diferentes [WfMC, 2005].

Esta especificação utiliza XML como o mecanismo para definir o processo de intercâmbio ou comunicação entre produtos. XPDL forma uma normatização para intercâmbio comum, que permite que os produtos troquem suas informações internas a partir das representações e definições de um processo padrão [WfMC, 2005].

Uma variedade de diferentes mecanismos pode ser utilizada para o processo de transferência de dados

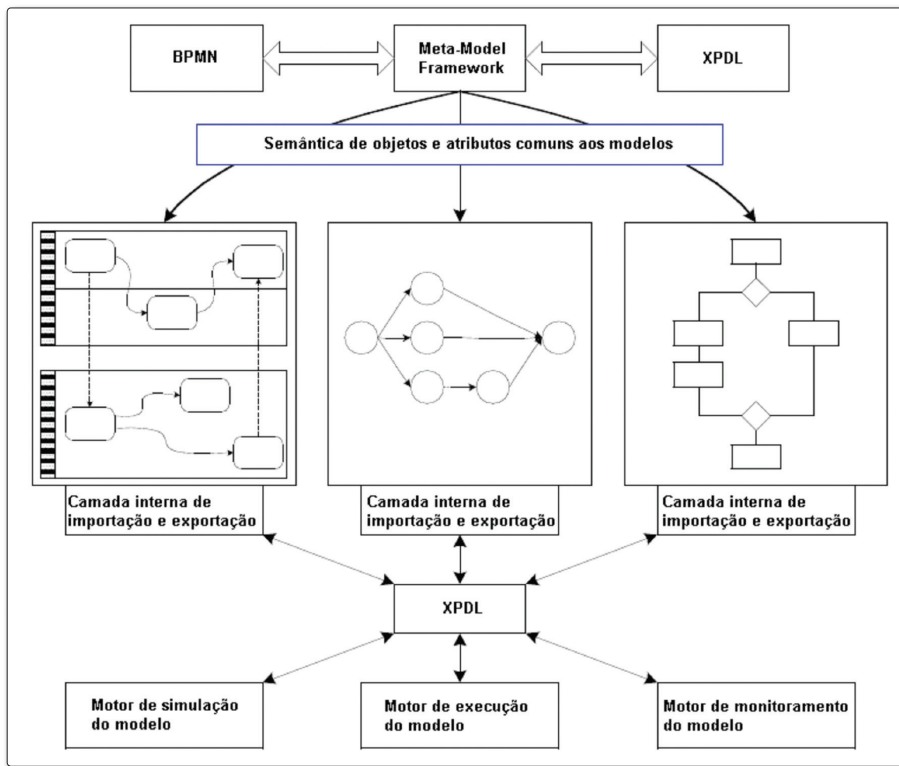


Figura 4. O conceito do processo de definição de intercâmbio da linguagem (adaptado de [WfMC, 2005]).



```

<xsd:element name="DataObject">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="xpdl:DataFields"
        minOccurs="0"/>
      <xsd:any namespace="##other"
        processContents="lax"
        minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id"
      type="xsd:NMTOKEN"
      use="required"/>
    <xsd:attribute name="Name"
      type="xsd:string"
      use="optional"/>
    <xsd:attribute name="State"
      type="xsd:string"
      use="optional"/>
    <xsd:attribute name="RequiredForStart"
      type="xsd:boolean"
      use="required"/>
    <xsd:attribute
      name="ProducedAtCompletion"
      type="xsd:boolean"
      use="required"/>
    <xsd:anyAttribute namespace="##other"
      processContents="lax"/>
  </xsd:complexType>
</xsd:element>
  
```

Figura 5. Código de implementação de DataObject. Fonte: [WfMC, 2005]

Elementos BPMN	Elementos BPEL
<b>BPMN Object/Attribute</b>	<b>BPEL Element/Attribute</b>
<b>Business Process Diagram</b>	See next row... mapped to attributes of a <code>process</code> element
<b>ExpressionLanguage = "Java"</b>	<code>expressionLanguage="Java"</code>
<b>Business Process</b>	The <code>process</code> element
<b>Name = "Processo Reserva Viagem"</b>	<code>name="ProcessoReservaViagem"</code>
<b>ProcessType = "Private"</b>	<code>abstractProcess="no"</code> or not included
<b>SupressJoinFailure = "Yes"</b>	<code>suppressJoinFailure="yes"</code>

Figura 6. Mapeamento dos atributos básicos do processo de negócio. Fonte: [WfMC, 2005].

entre sistemas, de acordo com as características dos diferentes cenários empresariais. Em todos os casos, o processo de intercâmbio deve ser expresso de forma consistente a fim de se obter um conjunto comum de objetos, relacionamentos e atributos expressando seus conceitos subjacentes.

Dessa forma, é possível que diferentes sistemas computacionais com diferentes modelos de processos de negócio interajam com base em uma linguagem padrão baseada em XML.

A Figura 4 apresenta um contexto da interação de modelos de negócio e sua transformação na linguagem comum baseada na normatização XPDL. O meta-modelo representado descreve as entidades do nível mais alto que estão na definição do processo, seus relacionamentos e atributos e, apesar de serem utilizadas várias notações para modelagem dos diversos processos, é utilizada apenas a definição padrão para a interface de dados, no caso é utilizada o XPDL. Portanto, a execução, a simulação e o monitoramento dos processos são concebidos sob uma única interface padrão de dados.

As entidades do BPMN – *Swimlanes*, *process activit*, *transition information*, *participant declaration*, *artifact*, *message flow*, *association* entre outros [WfMC, 2005] – podem ser encontradas nas sintaxes do XPDL e, para cada uma notação, existe um código apropriado. Por exemplo, a seguir pode ser observado o código XML para a notação *Swimlane* e a definição de *Data Object*, gerados com base no padrão XPDL.

Outras representações podem ser verificadas como, por exemplo, na Figura 5 onde é descrito o código de geração para o elemento *DataObject* da notação BPMN.

### Transformação do processo de reserva de viagem

Essa sessão apresenta nas Figuras 6, 7 e 8 a transformação de alguns elementos do modelo de processo de reserva de viagem apresentado para o respectivo código BPEL. De um lado estão os elementos do modelo BPM com a representação da configuração de cada um dos seus elementos. Do outro, estão os respectivos códigos em BPEL após a transformação do modelo em uma linguagem executável.

A Figura 6 apresenta a configuração básica do BPM com o nome principal do processo, o tipo de linguagem de implementação, entre outras configurações. Do lado BPEL é encontrado o código em linguagem executável relativo à configuração do modelo do processo.

Os *WebServices* gerados a partir de parte do modelo são demonstrados na Figura 7. A inicialização do processo bem como a atividade de Serviço de Reserva do Hotel são mapeados de forma a gerarem serviços para o *PartnerLink* que serão transformados em *WebServices* em tempo de execução do modelo do processo de reserva de viagens.

A Figura 8 apresenta os tipos de objetos de inputs gerados a partir do BPMN e seu respectivo código executável em BPEL. São demonstrados na figura as mensagens e tipos de objetos de entrada de dados como companhia aérea e a chegada dos vôos mapeados.

A Figura 9 apresenta uma ilustração do ambiente da transformação do modelo em BMPN e sua forte aderência aos processos de negócio da empresa, em que analistas, projetistas e consultores de estratégia de negócio estão com seu esforço voltado a elaborar e construir um modelo diagramático dos processos empresarias. Enquanto que, ao longo da transformação do modelo de negócios em produto executável, temos engenheiros de software, arquitetos de sistemas e projetistas preocupados em transformar o modelo desenhado para uma linguagem que possa ser executada, onde se verifica a presença de implementação tecnológica baseada em BPEL ou XPDL.

É importante destacar que quanto mais alto o nível, ou seja, voltado ao desenvolvimento do negócio, mais

BPMN Object/Attribute	BPEL Element/Attribute
<b>Implementation</b> = "Web service"	Invoke, but some properties, below, will map to a <b>partnerLink</b>
<b>Participant</b> = "InicioProcesso"	<b>name</b> ="InicioProcesso" <b>partnerLinkType</b> ="InicioProcessoPLT"
<b>BusinessRole</b> ="PapelProcessoViagem"	<b>myRole</b> ="PapelProcessoViagem"
<b>Participant</b> = "ServicoReservaHotel"	<b>name</b> =" ServicoReservaHotel " <b>partnerLinkType</b> ="ServicoReservaHotelPLT"
<b>BusinessRole</b> = "PapelReservaHotel"	<b>myRole</b> ="PapelReservaHotel"

Figura 7. Mapeamento das propriedades *WebServices*. Fonte: [WfMC, 2005].

BPMN Object/Attribute	BPEL Element/Attribute
<b>Property</b>	BPEL variable and WSDL message
<b>Name</b> = "input"	For BPEL variable: <b>name</b> ="input" <b>messageType</b> ="input" For WSDL message: <b>name</b> ="input"
<b>Type</b> = "structure"	The sub-Properties of the structure will map to the WSDL message elements
<b>Property</b>	For WSDL message, in the <b>part</b> element:
<b>Name</b> = "linhaAerea"	<b>name</b> ="linhaAerea "
<b>Type</b> = "string"	<b>type</b> ="xsd:string"
<b>Property</b>	For WSDL message, in the <b>part</b> element:
<b>Name</b> = "chegada"	<b>name</b> ="chegada" ←
<b>Type</b> = "string"	<b>type</b> ="xsd:string"
Eleven more sub-Properties are included	Eleven more <b>part</b> elements are included
Ten more structure Properties are included	Ten more BPEL variable elements and WSDL message elements will be included

Figura 8. Mapeamento das propriedades e atributos dos objetos. Fonte: [WfMC, 2005].

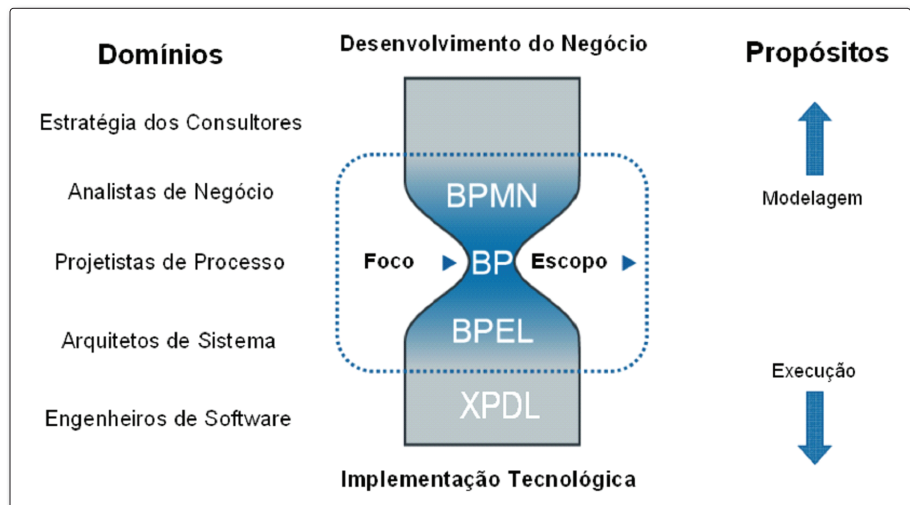


Figura 9. Contextualização de transformação de modelo BPMN em linguagem de execução BPEL e XPDL. Fonte: [BPMN, 2005].



aderentes são os domínios relacionados às estratégias do negócio e os propósitos de modelagem, nesse contexto estão fortemente envolvidos os profissionais que mais conhecem de todos os processos da empresa. De outro lado, quanto mais voltado à implementação tecnológica, mais estão envolvidos os profissionais voltados à tecnologia das linguagens trabalhando para que os modelos de negócios possam ser executados.

## O estado da arte

Basicamente, as técnicas de modelagem de processos de negócio são suportadas por representações diagramáticas de fluxo de dados ou controle das atividades de um determinado processo. Isso facilita a documentação, o entendimento e a manutenção dos modelos. Alguns elementos são fundamentais nesse contexto, como:

- atividades;
- ligações entre atividades que podem representar pontos de decisão;
- padrões de coordenação, tais como o fluxo seqüencial e a execução paralela.

A facilidade do uso da notação, o crescimento do paradigma SOA e a crescente implantação nas empresas,

impulsiona a utilização de abordagens de modelagens de processos. Para Natí [NATI, 2006]:

*“SOA será utilizada parcialmente em mais de 50% das novas aplicações de missão crítica e processos de negócio criados em 2007, e mais de 80% até 2010 (70% de probabilidade)”.*

*“Organizações que comecem sua transformação para BPM durante 2006 e 2007 serão recompensadas pelo domínio de suas indústrias por volta de 2010. Ter uma arquitetura de processos (parte de uma arquitetura de negócios) e alinhar as iniciativas de BPM com as iniciativas de SOA são atividades chaves a serem realizadas em 2007”.*

O relatório do Gartner Group de novembro de 2006 aponta o crescimento dos investimentos e pesquisas nas chamadas BMP Suítes. Os métodos baseados nessa *suíte* estarão no topo dos investimentos nos próximos 2 a 5 anos e ficará entre as mais altas no gráfico de evolução de ferramentas, técnicas e tecnologias do grupo. A **Figura 10** apresenta os resultados do relatório onde pode ser visualizada a curva de crescimento das tecnologias e ferramentas computacionais se aproximam-

do ou afastando da visibilidade de investimentos ao longo do tempo.

Outro indicador pesquisado pelo Gartner Group sobre o avanço em soluções para BPM é o investimento dos grandes *players* do mercado. A **Figura 11** apresenta uma relação desses *players* e os seus principais focos de investimento para soluções de BPM.

## Conclusões

As considerações feitas sobre a modelagem de processos de negócio e a geração de código XML a partir de um modelo diagramático, muda sensivelmente a visão de como modelar os requisitos de negócio da empresa. A visão do analista de sistemas, o envolvimento de analistas de negócios e outros setores estratégicos fazem com que haja uma aproximação entre os setores tecnológicos da empresa e os setores de negócio.

Aparentemente estamos em um novo limiar de visão da TI na empresa. E os modelos de negócio sendo executados pela equipe técnica fazem com que uma visão empresarial seja mais difundida. Dessa forma, entende-se que produtos possam ser mais bem elaborados e mais próximos da realidade do usuário.

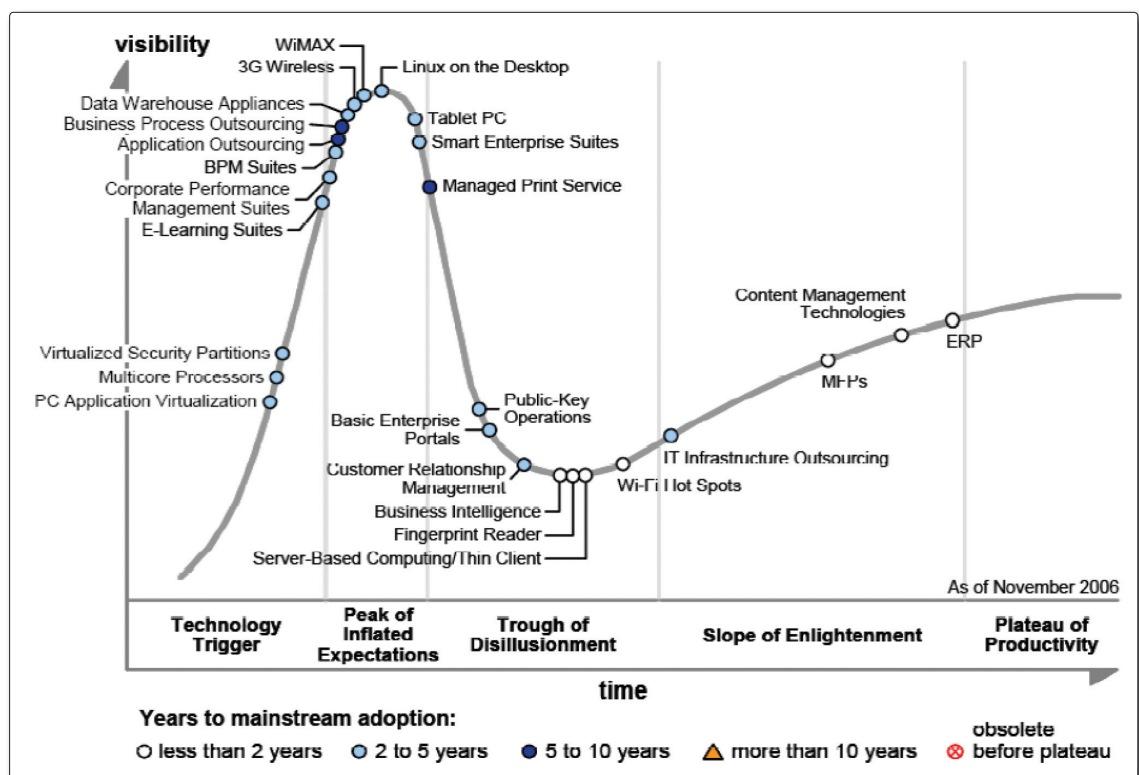


Figura 10. Gráfico de evolução de tecnologias, ferramentas, aplicativos entre outros. Fonte: Gartner Group, novembro, 2006 [NATI, 2006].

Além dessa melhoria, a modelagem de processos faz com que os profissionais saiam do campo operacional e entrem em visões mais estratégicas da empresa. Dessa forma, os investimentos passam a focar em um outro âmbito, não só o de treinamento em ferramentas ou linguagens de programação, mas em formação multidisciplinar de profissionais de TI, que passam agora a ter como demanda uma formação mais generalista e sistêmica. ●

**Dê seu feedback sobre esta edição!**

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

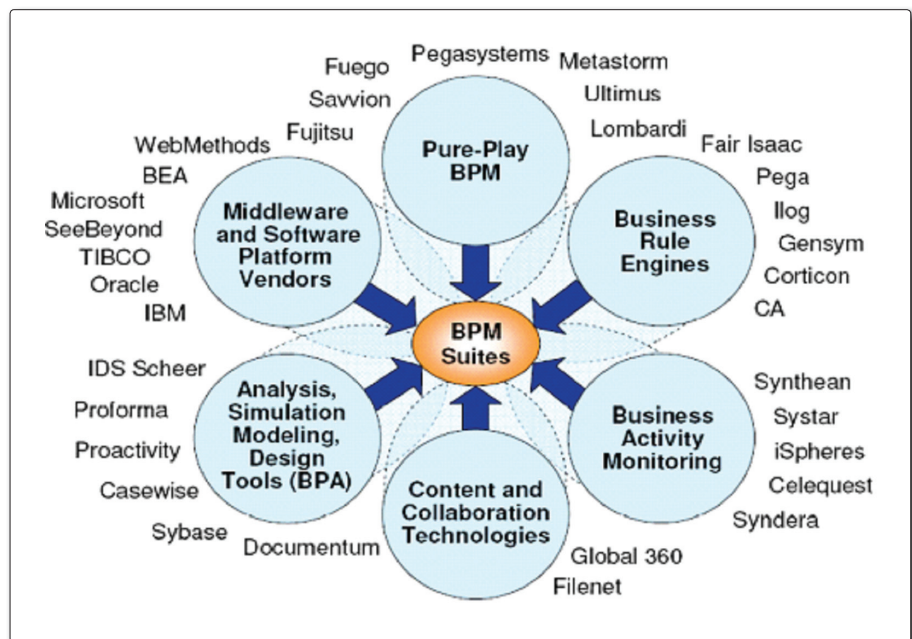


Figura 11. Ilustração contendo fornecedores de soluções BPM. Fonte: Gartner Group, novembro, 2006 [NATI, 2006].

**Referências**

[AKIY, 2004] AKIYAMA, E. (2004) Gestão de Processos: BPM (Business Process Management). SAP. Basf Case.

[BOLI, 2006] BOLIE, J. et al. (2006) BPEL Cookbook: Best Practices for SOA-based integration and composite applications development. 188p. Packt Publishing Ltd.

[BPMN, 2005] BPMN Fundamentals (2005). OMG Beidtf Meeting. Atlanta. September.

[BPMN, 2006] Business Process Modeling Notation Specification (2006). OMG Final Adopted Specification. February.

[BPMN, 2007] BPMN: o Modelo E-R dos Processos. <Disponível em: <http://www.iprocess.com.br/artigos/4.htm>> <Consultado em: 22/07/2008>

[JURI, 2006] JURIC, M. B. (2006) Business Process Execution Language for Web Services BPEL and BPEL4WS. 2nd Edition. 372p. Packt Publishing Ltd.

[NATI, 2006] NATI, Y. (2006). Predicts 2007: SOA Advances. EUA: Gartner, id: G00144445, Novembro.

[VERN, 1996] VERNADAT, F. B. (1996) Enterprise modelling and integration: principles and applications. London: Chapman & Hall.

[WfMC, 2005] Process Definition Interface - XML Process Definition Language (2005). Workflow Management Coalition Workflow Standard. October.

[WHIT, 2005] WHITE, A. S. (2005) Using BPMN to Model a BPEL Process. BPTrends. 2005. <Disponível em [www.bptrends.com.br](http://www.bptrends.com.br). > <Consultado em: 22/07/2008>



# Metodologias Ágeis

## Uma Visão Prática



### André Luiz Banki

[banki@altoqi.com.br](mailto:banki@altoqi.com.br)

Engenheiro Civil pela Universidade Federal de Santa Catarina (UFSC), Mestre em Engenharia Civil (na área de Estruturas), também pela UFSC, e Especialista em Engenharia de Software pelo SENAI/Florianópolis, trabalha como Gerente de Desenvolvimento na empresa AltoQi Tecnologia em Informática Ltda., tendo sido responsável pela concepção e coordenação do desenvolvimento de diversos sistemas de renome nacional na área de Engenharia.



### Sérgio Akio Tanaka

[sergio.tanaka@audare.com.br](mailto:sergio.tanaka@audare.com.br)

Especialista em Gestão Empresarial pelo Instituto Superior de Ensino (ISE), em convênio com o IESE de Barcelona; Mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul; Pós-graduado pela Universidade Estadual de Londrina nas áreas de Redes de Computadores e Banco de Dados e em Análise de Sistemas pela UniFil. É Diretor e consultor certificado pela IBM Rational e Websphere na AUDARE Engenharia de Software. Na área acadêmica, atua como Professor e Coordenador de Pós-Graduação da Universidade Filadélfia (UNIFIL) e da área de Engenharia de Software do SENAI/SC.

#### De que se trata o artigo:

Avaliação da presença das metodologias ágeis no mercado de desenvolvimento de software. Neste artigo, o autor procura verificar quantitativamente alguns pontos de interesse, identificando o estado atual do movimento ágil e a pouca disponibilidade de estudos de caso que possam ser usados como fonte sistemática de resultados para comparação.

#### Para que serve:

Fornecer subsídios para o processo de escolha de uma metodologia de desenvolvimento a ser adotada por uma empresa de software através de uma avalia-

ção crítica das informações disponíveis sobre os resultados da aplicação das metodologias ágeis em projetos reais de desenvolvimento.

#### Em que situação o tema é útil:

As metodologias ágeis, como a Extreme Programming (XP) e o Scrum, entre outras, têm despertado atenção crescente do mercado. Esse movimento, baseado no ciclo de desenvolvimento incremental e iterativo, está focado na colaboração do cliente, no valor dos indivíduos e na adaptação às mudanças, tendo mostrado ganhos de produtividade nos mais diversos tipos de projetos de desenvolvimento de software.

A escolha da metodologia mais adequada para o desenvolvimento de software em uma organização não é uma tarefa trivial. As metodologias ágeis têm despertado o interesse do mercado, apresentando evidências de melhoria na produtividade, mas, para que possam ser efetivamente usadas em larga escala, precisam provar alguns de seus pontos de vista. Neste artigo, procura-se verificar quantitativamente alguns pon-

tos de interesse, identificando a presença das metodologias ágeis no mercado, o estado atual do movimento ágil e a pouca disponibilidade de estudos de caso que possam ser usados como fonte sistemática de resultados para comparação.

### Introdução

A maior parte dos projetos de desenvolvimento de software pode ser descrita simplesmente como “programar

e corrigir”, sendo desenvolvidos sem planejamento ou uma fase organizada de design do sistema. Disso usualmente decorre uma grande quantidade de erros, os quais precisam ser resolvidos, em uma longa etapa que sempre estende o prazo inicialmente proposto. O movimento original de melhoria no setor foi o que introduziu a noção de metodologia, ou seja, uma abordagem disciplinada para o desenvolvimento de software com o objetivo de tornar o processo mais previsível e eficiente (FOWLER, 2005).

Em 2001, movidos pela observação de que equipes de desenvolvimento de software nas mais diversas organizações estavam presas por processos cada vez mais burocráticos, um grupo de profissionais reuniu-se para delinear os valores e princípios que permitiriam às equipes de desenvolvimento produzir rapidamente e responder às mudanças. Eles chamaram a si mesmos de Aliança Ágil. Trabalharam por dois dias para criar um conjunto de valores. O resultado foi o Manifesto da Aliança Ágil (MARTIN, 2002). Embora as metodologias que compõem o movimento ágil já estivessem no mercado há alguns anos, com a denominação de “metodologias leves”, o Manifesto Ágil é considerado oficialmente como o início do movimento ágil.

Segundo Abrahamsson (2002), uma metodologia pode ser dita ágil quando efetua o desenvolvimento de software de forma incremental (liberação de pequenas versões, em iterações de curta duração), colaborativa (cliente e de-

envolvedores trabalhando juntos, em constante comunicação), direta (o método em si é simples de aprender e modificar) e adaptativa (capaz de responder às mudanças até o último instante). Nesse conceito, inclui como metodologias ágeis: Extreme Programming (XP), Scrum, Crystal, Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), Open Source Software Development e, com certa ressalva, o Rational Unified Process (RUP).

**Importância dos estudos para o mercado**

As idéias relativas ao movimento ágil têm sido rapidamente disseminadas pela comunidade de desenvolvimento. Todavia, mesmo que os desenvolvedores avaliem, de forma favorável, técnicas como o desenvolvimento incremental e a programação orientada a testes, sugerindo a adoção de uma metodologia ágil, essa decisão ainda tem que ser tomada pela organização na qual estão inseridos. Para isso, se faz necessária uma argumentação quantitativa.

Na Curva de Adoção de Tecnologia, apresentada na **Figura 1**, Rogers (2003, apud GRINYER, 2007) introduz o conceito de “decisão sobre a inovação”, o qual indica que um grupo (ou indivíduo) procura determinar as vantagens e desvantagens de uma inovação, com o objetivo de reduzir a incerteza antes de sua adoção. A partir disso, descreve cinco categorias sociais, determinadas com base no seu grau de adoção das inovações. Os “inovadores” são os que

mais facilmente adotam as inovações e, para a direita do gráfico, cada grupo apresenta uma resistência maior, terminando com os “tardios”, menos afeitos à adoção de qualquer inovação.

Segundo Ambler (2006), existe uma grande mudança na percepção do mercado, dos dois lados do “abismo” apontado nessa curva. As empresas à direita do abismo possuem uma expectativa significativamente diferente daquelas à esquerda na curva. Segundo esse autor, as empresas no lado esquerdo do abismo estão mais interessadas em novas idéias e mais propensas a aceitar riscos, enquanto que as empresas no lado direito são mais conservadoras e preferem aguardar por uma prova de que a idéia realmente funciona.

Grinyer (2007) afirma que as metodologias ágeis como o XP e o Scrum são consideradas inovações, pois ainda são relativamente recentes no mercado. Todavia, nem todas as técnicas ágeis são inovadoras. Por exemplo, o desenvolvimento iterativo já se encontra bem estabelecido no mercado. Ambler (2006) complementa, afirmando que, como um conceito, o movimento ágil já cruzou o abismo, visto que a maior parte das empresas que se enquadram na categoria de primeira ou segunda maioria está ao menos demonstrando interesse nas técnicas ágeis. A metodologia Scrum e técnicas como a refatoração e o desenvolvimento orientado a testes certamente cruzaram o abismo, mas o XP como um todo ainda encontra resistência nesse sentido.

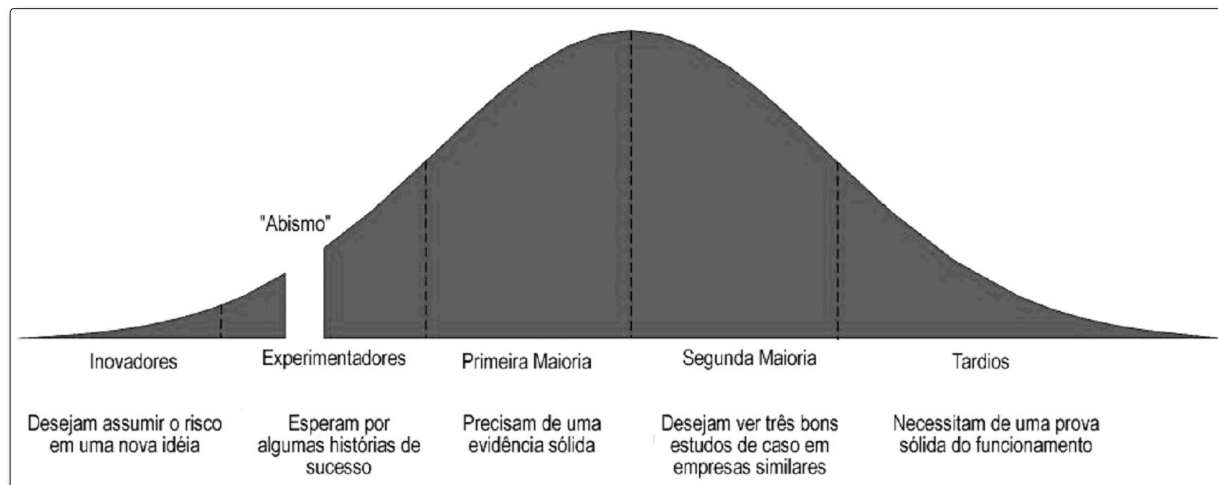


Figura 1. Curva de Adoção de Tecnologia. Fonte: adaptado de Ambler (2006)



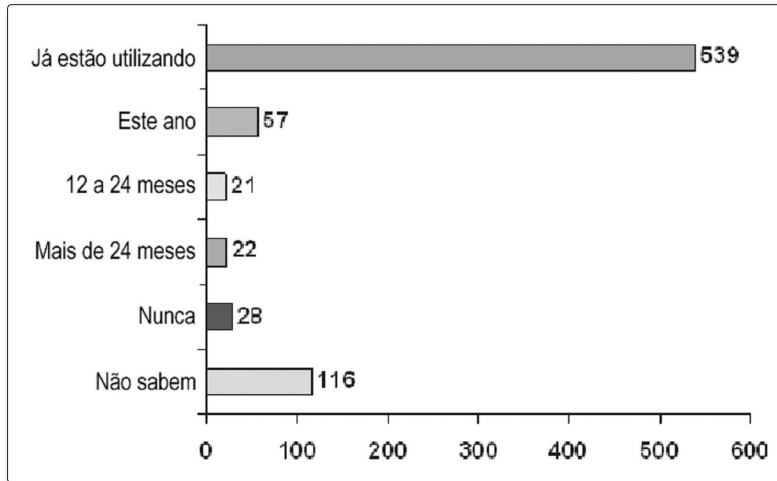


Figura 2. Pesquisa: prazo para adoção do desenvolvimento ágil. Fonte: adaptado de Ambler (2007)

Item da pesquisa	Resultado
Percentual de empresas que adotam metodologias ágeis em todos os projetos	46%, nas empresas de porte médio, e 12%, nas empresas de grande porte
Percentual de empresas que adotam metodologias ágeis em algum projeto	44%
Principal barreira para a adoção do desenvolvimento ágil	Falta de conhecimento (51% entre os desenvolvedores, 56% entre os executivos)
Principal motivação para a adoção do desenvolvimento ágil (para os executivos)	Melhorar a produtividade e previsibilidade no desenvolvimento de software (51%)
Principal motivação para a adoção do desenvolvimento ágil (para os desenvolvedores)	Auxiliar no gerenciamento do escopo dos projetos (47%)

Tabela 1. Pesquisa sobre a aceitação das metodologias ágeis. Fonte: Projects@Work (2006)

Para que o desenvolvimento ágil possa ser efetivamente aceito em todos os segmentos do mercado, é necessário satisfazer às necessidades dos diversos grupos, os quais são muito diferentes dos “inovadores”. Ao apresentar uma nova metodologia a uma comunidade bastante técnica como a dos desenvolvedores de software, pode-se conseguir atenção apontando as suas qualidades e idéias inovadoras. Por outro lado, para cruzar o “abismo” e ganhar efetivamente a atenção das organizações situadas no seu lado direito, é necessário reunir informações e apresentá-las de forma a atingir a esfera gerencial, responsável pela efetiva tomada de decisões. No decorrer deste artigo, são fornecidos alguns dados recentes, com respeito ao efetivo posicionamento das metodologias na Curva de Adoção de Tecnologia.

### Dificuldades envolvidas

Um tipo de estudo que pode ser utilizado para a avaliação de uma metodologia de desenvolvimento envolve uma com-

paração da performance alcançada em dois grupos de interesse, sendo que um usa uma dada metodologia e outro não. A cada grupo, é atribuída uma tarefa; ao final, avaliam-se itens como produtividade e taxa de erros. São exemplos desse tipo de estudo os trabalhos de Bona (2002) e Nonemacher (2003). Harrison (2005) critica essa abordagem, afirmando que esses estudos seriam realmente válidos se envolvessem dúzias de programadores, ao invés de três ou quatro, pois, em grupos pequenos, existe uma grande influência de fatores aleatórios.

Uma alternativa aos estudos em laboratório é a pesquisa de campo, na qual os dados são obtidos através de medições no desenvolvimento de projetos reais. Todavia, para que seja obtido um número suficiente de dados, isso é feito agrupando projetos desenvolvidos por equipes muito diferentes em áreas também distintas, dificultando qualquer comparação. Outra alternativa é a da observação individual, efetuada sobre um único profissional, equipe ou projeto. Nesse caso, é feita uma medição inicial de

performance, aplicada a alteração de metodologia e efetuada nova medição. Normalmente, os estudos de caso encaixam-se nesta última categoria, com a diferença de que não são feitos de forma controlada, estabelecendo suas hipóteses apenas depois da implantação da metodologia e não como objetivo inicial do estudo.

Segundo Grinyer (2007), apesar do volume crescente de informação formada pelas diversas pesquisas empíricas já publicadas, há um consenso de que as organizações ainda tomam pouco conhecimento delas.

Nossa maior descoberta é a de que há uma aparente contradição, entre a afirmação dos desenvolvedores de que estes precisam de evidências de melhorias no processo de desenvolvimento de software e o que os mesmos efetivamente aceitam como evidências. Isto apresenta um problema de pesquisa bastante sério: mesmo que os pesquisadores possam demonstrar uma forte e confiável correlação entre a melhoria no processo de desenvolvimento e a melhoria para a organização como um todo, ainda resta o problema de convencer os desenvolvedores de que as evidências encontradas aplicam-se à sua situação em particular (RAINER et al, 2003, *apud* GRINYER, 2007, p.45).

Com respeito às metodologias ágeis, Jeffries (2005, *apud* GRINYER, 2007) afirma que os profissionais formam a maior parte das suas idéias pelo contato direto com outros profissionais, pelo que possam ler rapidamente e, principalmente, pela prática. Nesse sentido, os estudos desenvolvidos na indústria do software, em cenários similares aos da situação da empresa na qual se trabalha, mostram-se valiosos na formação de opiniões.

### Presença dos métodos ágeis no mercado

Uma questão muito importante, a qual será sempre levantada por qualquer organização que estude a adoção de uma nova metodologia, é a quantidade de empresas que já a adotam com sucesso. Infelizmente, na comunidade de desenvolvimento de software, não existe praticamente nada determinado estatisticamente. É necessário valer-se de alguns poucos estudos, conduzidos informalmente, para verificar as tendências do mercado.



Segundo uma pesquisa realizada pela empresa DigitalFocus ([www.digitalfocus.com](http://www.digitalfocus.com)) e apresentada em um dos eventos mais importantes da comunidade ágil, a Agile 2006, o interesse nas metodologias de desenvolvimento ágil está crescendo, com 81% das empresas adotando uma metodologia ágil ou procurando por uma oportunidade para fazê-lo (PROJECTS@WORK, 2006). Essa pesquisa, a qual coletou informações de 136 profissionais, provindos de 128 empresas de diferentes portes, teve como objetivo identificar os fatores chave envolvidos na adoção das metodologias ágeis, sob o ponto de vista técnico e gerencial. As principais colocações estão resumidas na **Tabela 1**.

Outra pesquisa, bastante recente, foi conduzida por Ambler (2007), através de uma consulta on-line, executada em maio de 2007, a qual alcançou 781 profissionais distribuídos nos mais diversos papéis e em organizações de porte diverso. Nessa pesquisa, o autor afirma que o desenvolvimento ágil certamente já cruzou o “abismo” citado anteriormente. A **Figura 2** apresenta o grau de adoção das técnicas ágeis nas empresas consultadas,

indicando que 69% delas já utilizavam alguma metodologia ágil e mais 7% pretendiam fazê-lo no mesmo ano.

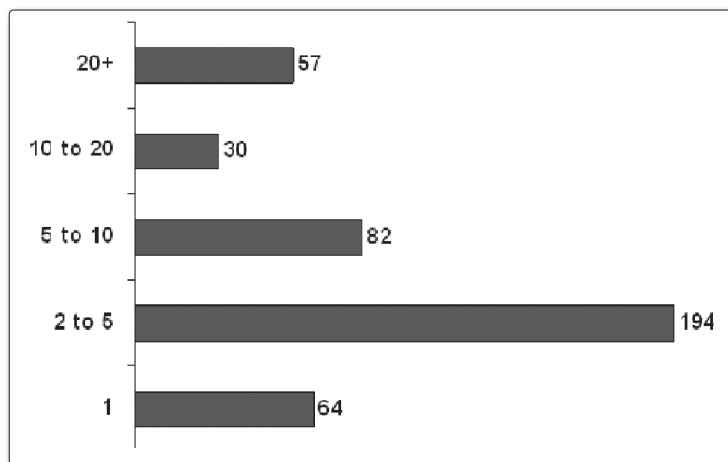
Outro dado importante, apresentado na **Figura 3**, é o de que o desenvolvimento ágil já passou a fase do “projeto piloto”, com a maior parte das companhias tendo reportado a execução bem-sucedida de um número relevante de projetos com o uso de metodologias ágeis.

Esse tipo de pesquisa possui como importante limitação o fato de que sua amostragem não se baseia em nenhum método científico, ficando limitada ao perfil de “profissionais que se interessam em responder questionários on-line”. É razoável supor que os profissionais que estão descontentes com uma dada metodologia não freqüentem, da mesma forma como os demais, o tipo de comunidade on-line consultada. Apesar disso, são importantes para apontar as tendências do mercado.

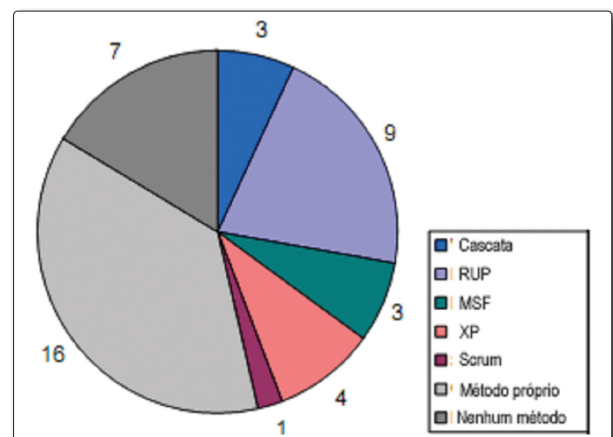
Um estudo mais dirigido foi feito na Suécia, por Fransson e Klercker (2005), com o objetivo de mapear as pequenas e médias empresas de desenvolvimento de software do país e levantar o grau de satisfação de cada uma delas, com

relação à metodologia de desenvolvimento que estão adotando. Partindo de um universo de 1016 empresas, os autores estipularam uma amostra relevante de 100 empresas e, destas, chegaram a 57 relevantes. Na **Figura 4**, é indicado o número de empresas que adotam cada metodologia, descartados os valores menos representativos.

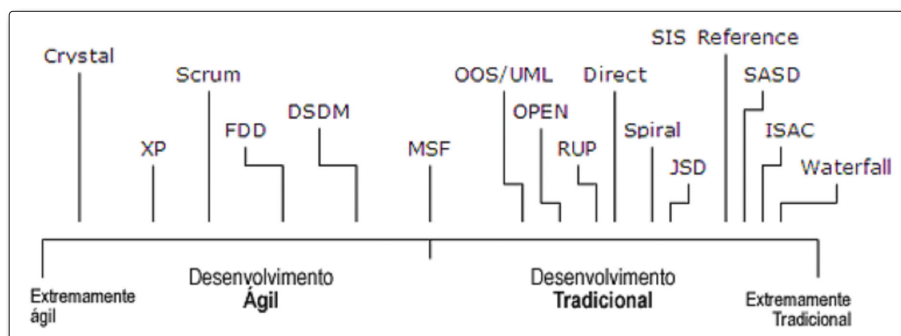
Pode-se observar que mais da metade dessas empresas reportou a utilização de um método próprio, ou nenhum método definido, para o desenvolvimento de software. No restante, esse levantamento imparcial aponta uma expressiva presença do RUP e o destaque para o XP, o Scrum e o Microsoft Solutions Framework (MSF). É importante comentar que esse trabalho classifica o MSF como uma metodologia “neutra”, consistindo tanto de técnicas ágeis como tradicionais, das quais o desenvolvedor pode adotar as que desejar, e posiciona o RUP no lado das metodologias tradicionais. A **Figura 5** apresenta o Eixo de Agilidade usado pelos autores, no qual se observam diversas metodologias encontradas na bibliografia.



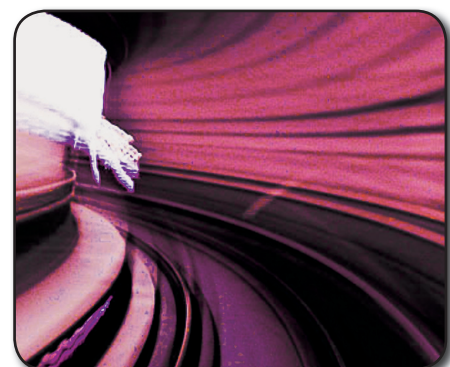
**Figura 3.** Pesquisa: número de projetos ágeis já executados. Fonte: adaptado de Ambler (2007)



**Figura 4.** Metodologia de desenvolvimento adotada (Suécia). Fonte: adaptado de Fransson e Klercker (2005, p.51)

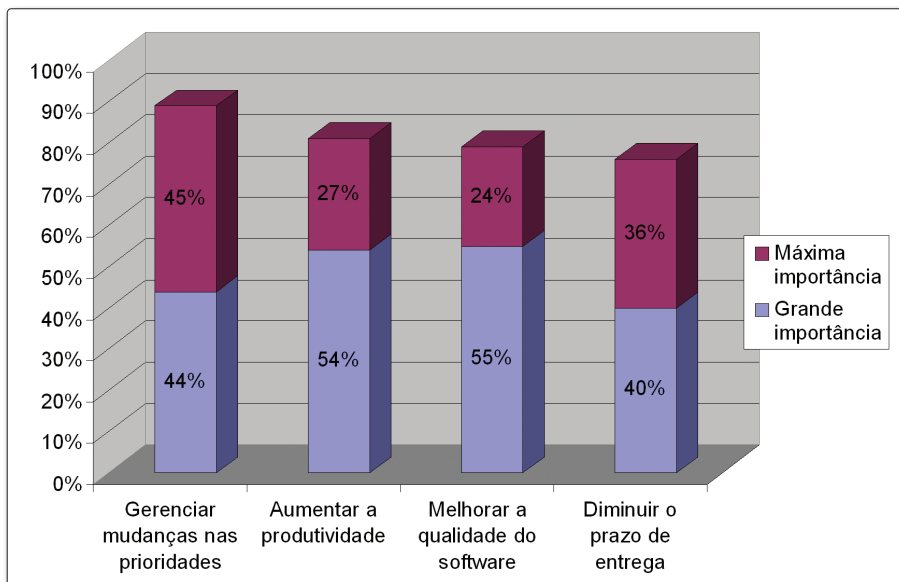


**Figura 5.** Eixo de Agilidade. Fonte: adaptado de Fransson e Klercker (2005, p.35)



Categoria	Nº. de fatores
1 – Melhoria no processo de desenvolvimento, como resposta a repetidos fracassos em projetos	15
2 – Influências internas na organização (gerente, desenvolvedor sênior ou arquiteto)	11
3 – Fator competitivo (prazo para colocação do produto no mercado)	10
4 – Influências externas à organização (treinamento ou consultoria externa)	8
5 – Resposta a requisitos em constante variação	5
6 – Acompanhamento de tendências tecnológicas e do mercado	2
7 – Downsizing da equipe e processo de desenvolvimento	2
<b>Total</b>	<b>53</b>

**Tabela 2.** Fatores de adoção de uma metodologia ágil. Fonte: adaptado de Grinyer (2007)



**Figura 6.** Fatores de adoção de uma metodologia ágil. Fonte: adaptado de Barnett (2006)

A partir disso, contam-se 11 empresas adotando uma abordagem tradicional, contra 6 adotando a abordagem ágil. Embora diversos autores, como Kruchten (2001), Pollice (2001), Kohrell e Wonch (2005), entre outros, defendam a ideia de que o RUP é um processo configurável, contendo os mesmos valores das metodologias ágeis, deve-se lembrar que, para fins de pesquisa, deve-se levar em conta a forma como este está sendo efetivamente aplicado. Nesse estudo, o RUP foi percebido por seus participantes como uma metodologia mais tradicional.

### Casos de implantação das metodologias

Um ponto interessante a se avaliar, destacado por Grinyer (2007), é a forma pela qual as organizações têm se decidido pela adoção dos métodos ágeis, visto que, conforme já mencionado, não existem indicações claras na literatura ou no meio acadêmico. Em um estudo baseado na análise da literatura disponível e em

uma consulta feita a três listas de discussão populares na comunidade ágil, esse autor coletou 41 relatos, relacionando 53 fatores que levaram à adoção de uma metodologia ágil. Esses fatores foram organizados em 7 categorias, apresentadas na **Tabela 2**.

Grinyer (2007) menciona outro fato relevante observado nesse levantamento: nenhum dos estudos de caso reportou a aplicação de uma metodologia exatamente como descrita, mas sempre com alguma personalização ao contexto no qual foi aplicada. Deve-se apontar, também, que nenhum dos estudos indicou os motivos pelos quais as empresas que adotaram uma nova metodologia acreditaram que essa mudança efetivamente levaria à solução dos problemas apontados na **Tabela 2**.

Além desse levantamento documental, deve-se destacar a pesquisa conduzida pela VersionOne ([www.versionone.net](http://www.versionone.net)) e apoiada pela Agile Alliance ([\[alliance.org\]\(http://alliance.org\)\), a qual atingiu mais de 700 profissionais, com a mesma intenção de determinar como os processos ágeis têm sido implementados nas diversas organizações. Dos 722 profissionais pesquisados, mais de 25% vinham de empresas com mais de 250 pessoas, mas apenas 18% deles trabalhavam em equipes de mais de 10 pessoas. Esses números comprovam que, embora o desenvolvimento ágil possa ser adaptado para empresas de maior porte, a grande maioria dos projetos é desenvolvida por pequenas equipes \(BARNETT, 2006\).](http://www.agile-</a></p>
</div>
<div data-bbox=)

Essa pesquisa aponta que o desenvolvimento ágil tem garantido um significativo retorno sobre o investimento para as organizações que o adotam. Analisando os fatores que levaram à adoção de uma metodologia ágil, identificou os fatores presentes na **Figura 6**.

Comparando esses resultados com os da **Tabela 2**, pode-se observar que o aumento da produtividade e da qualidade do software, aliado à capacidade de gerenciar requisitos em constante variação, é o grande motivador da adoção de uma metodologia ágil de desenvolvimento, seja essa qualquer uma das abordadas no presente trabalho.

Com relação às barreiras para a adoção do desenvolvimento ágil, Barnett (2006) comenta que, no início do Movimento Ágil, o principal motivo citado era a falta de apoio por parte das gerências e da organização como um todo. Agora, isso não é mais tão importante. A falta de profissionais qualificados para o desenvolvimento ágil e a resistência dos próprios desenvolvedores à mudança são agora apontados como os principais fatores, conforme observado na **Figura 7**.

Esse resultado, somando 42% nos fatores relativos à formação da própria equipe de desenvolvimento, é consistente com as conclusões apresentadas pela Projects@Work (2006), na **Tabela 1**, e com as colocações feitas por Ambler (2007), segundo o qual o desenvolvimento ágil já cruzou o “abismo” citado na Curva de Adoção de Tecnologia. Isso se confirma pela observação de que já existe grande apoio, por parte das gerências, na adoção do movimento ágil. A iniciativa pela adoção de uma metodologia ágil não está mais partindo unicamente dos “inovadores” (normalmente, apenas o

pessoal estritamente técnico), mas dos líderes. Segundo Grinyer (2007), esse é o segundo fator mais importante, com 27% de participação. Barnett (2006) apontou, em sua pesquisa, que em 11% dos casos, a iniciativa pela adoção do desenvolvimento ágil veio diretamente da presidência da empresa e, em outros 33% dos casos, dos gerentes de desenvolvimento.

## Medidas objetivas de sucesso

Conforme já comentado no início deste artigo, um fator que é levado em conta pelos profissionais ao decidir pela adoção de uma nova metodologia é a existência de casos de sucesso documentados por outras empresas, principalmente do mesmo porte e área de atuação. Podem-se obter diversos estudos de caso em sites conceituados, como o da Agile Alliance ([www.agilealliance.org](http://www.agilealliance.org)) ou da Scrum Alliance ([www.scrumalliance.org](http://www.scrumalliance.org)). Todavia, esse conjunto de informações está disperso e carece de medidas objetivas. Com relação à avaliação dos resultados obtidos em estudos de caso comparativos, a principal conclusão estabelecida foi a de que não existem estudos que possam ser usados, cientificamente, como referência. Todos os casos referem-se aos resultados obtidos na adoção de uma nova metodologia, mas comparados a uma situação anterior, na qual não existia nenhum processo formal ou era adotada uma metodologia estritamente tradicional, em cascata. Além disso, os resultados apresentados são sempre bastante subjetivos, apontando dificuldades ou melhorias sem quantificá-las especificamente.

Uma das poucas exceções nesse sentido é o caso publicado pela Sabre Airline Solutions (2004), o qual relata a adoção do XP por sua equipe de 300 desenvolvedores, como resposta aos problemas de atraso no cronograma e elevado número de defeitos enfrentados antes disso. Embora a empresa tenha precisado acrescentar técnicas não previstas no XP, para lidar com projetos de maior escala, as práticas do desenvolvimento iterativo, estórias de usuário, testes automatizados e programação em pares, por exemplo, foram levadas a cabo com sucesso. Segundo esse relato, a adoção de outras metodologias ágeis, como o RUP, o Scrum e o FDD também foram avaliadas, mas com a opção pelo XP.

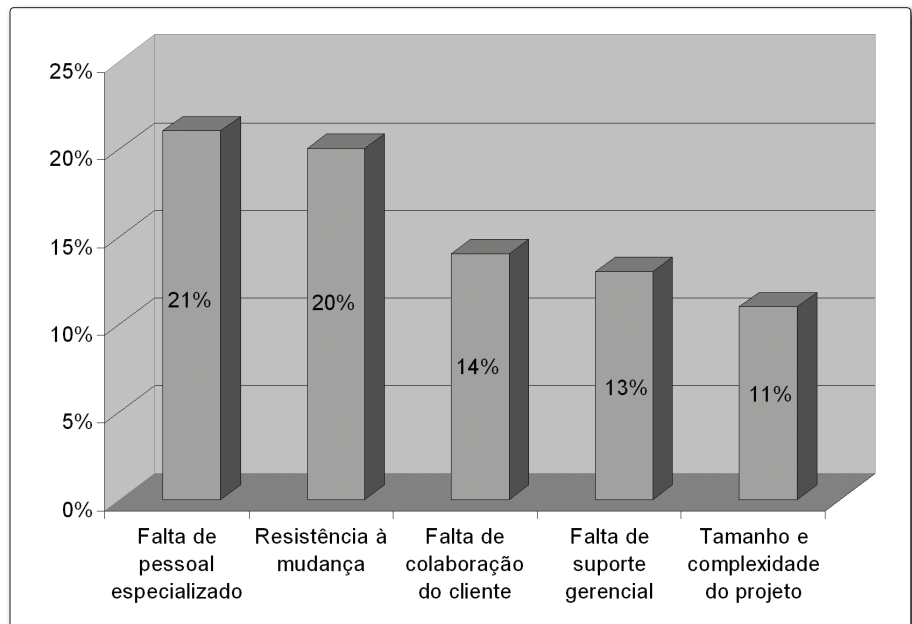


Figura 7. Barreiras para a adoção do desenvolvimento ágil. Fonte: adaptado de Barnett (2006)

	Antes do XP (2001)	Depois do XP (2002)
Programas	87	91
Horas de trabalho	23,531	17,726
Horas / Programa	270	190

Tabela 3. Aumento de produtividade na Sabre. Fonte: Sabre Airline Solutions (2004)



O aumento de produtividade na Sabre foi medido em um projeto de conversão de uma aplicação C++ para Java, com a mudança da plataforma destino para a Web, o qual obrigou o re-desenvolvimento de mais de 100 programas, na camada de interface com o usuário. O resultado da adoção do XP foi uma melhoria de 42%, conforme os dados apresentados na Tabela 3.

Além disso, esse mesmo estudo reporta uma redução no número de defeitos encontrados com relação às suas versões anteriores. Em um projeto, ProfitManager, com cerca de 500.000 SLOCs, apenas 10 erros foram reportados nos dois primeiros meses que sucederam o lançamento da sua versão desenvolvida com a metodologia XP, contra mais de 500 em sua versão anterior. Possivelmente outros fatores podem ter influenciado este resultado como, maior conhecimento do domínio e experiência da equipe de desenvolvimento. Aliado a isto, este resultado pode também indicar um ganho com o uso da metodologia XP.

Outro estudo relevante foi reportado por Marchesi et al (2002), comparando o desenvolvimento de duas versões de um produto para comércio eletrônico, feito

pela Escrow.com, uma empresa de pequeno porte. Uma delas, chamada "V2", foi desenvolvida com o uso de uma metodologia dita "tradicional" e a outra, chamada "V3", com o XP. Os resultados apresentados na Figura 8, embora também se baseiem em critérios subjetivos, apontam benefícios importantes obtidos com a adoção dessa metodologia. Neste cenário, o XP foi adotado sem a necessidade de adaptações.

Segundo Barnett (2006), a comunidade técnica precisa ser capaz de demonstrar quantitativamente os benefícios atingidos pela adoção do desenvolvimento ágil a fim de convencer as empresas mais tradicionais a fazer a mesma mudança. O estudo citado no item anterior tentou fazer com que os profissionais pesquisados apontassem os valores que foram efetivamente obtidos pela adoção de uma metodologia ágil. Os resultados, apresentados na Figura 9 mostram que os desenvolvedores que adotam as metodologias ágeis estão bastante satisfeitos com os resultados que têm encontrado.

Finalmente, pode-se concluir que o desenvolvimento ágil está em um momento de inflexão, passando a ser adotado por

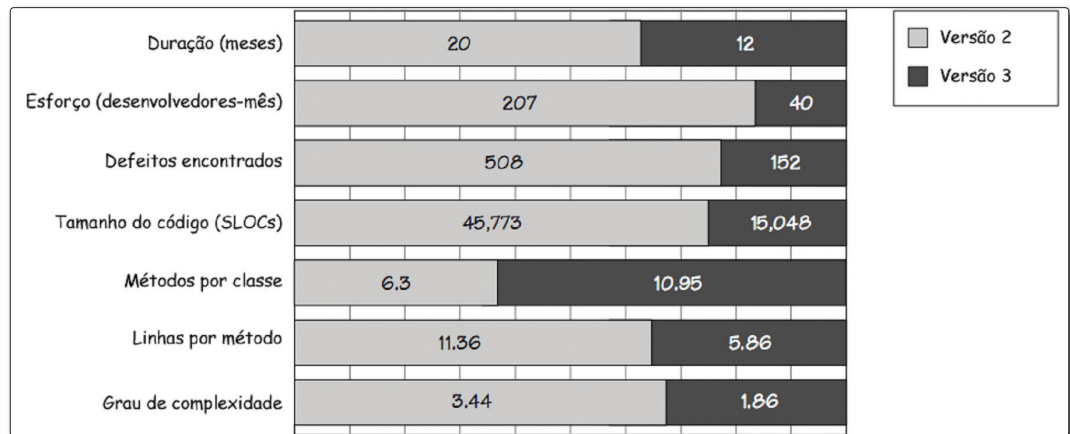
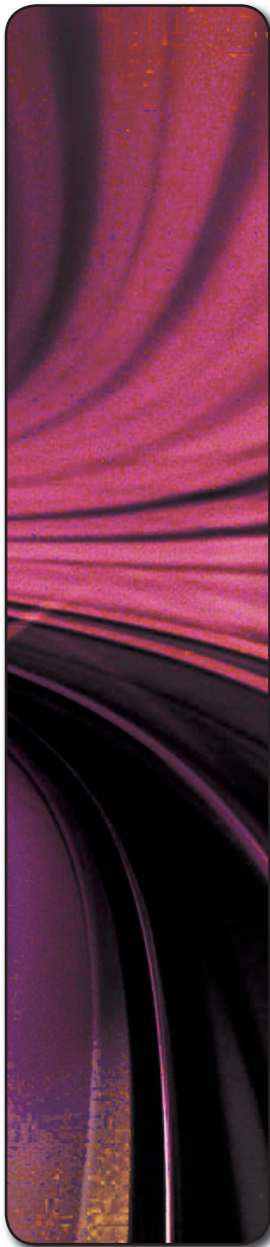


Figura 8. Comparação dos resultados obtidos com o XP na Escrow.com. Fonte: Marchesi *et al* (2002, p. 358)

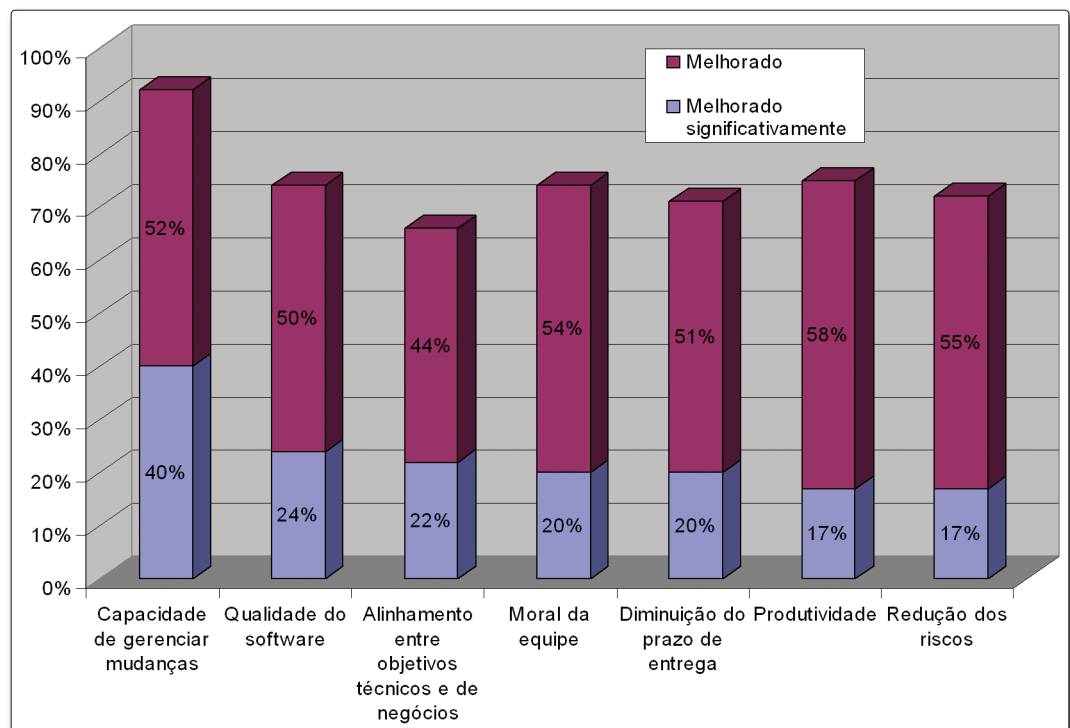


Figura 9. Valores obtidos pela adoção de uma metodologia ágil. Fonte: adaptado de Barnett (2006)

um número crescente de empresas. Com isso, um número maior de equipes pode passar a quantificar seus resultados. Espera-se, para um futuro próximo, que informações mais consistentes, relativas à adoção das metodologias estudadas neste trabalho, sejam levantadas e apresentadas à comunidade de desenvolvimento.

### Conclusões

A escolha da metodologia mais adequada para o desenvolvimento de software em uma organização, levando em consideração os inúmeros fatores envolvidos, não é uma tarefa trivial. Por outro lado, é um fator preponderante para o sucesso da organização. Embora um bom proces-

so não possa garantir o sucesso de um projeto, certamente a adoção de um processo inadequado pode compromê-lo.

Neste trabalho, foi mostrada a importância da publicação de estudos de caso, como ferramenta indispensável para a adoção de metodologias inovadoras nas empresas. O movimento ágil ainda deve ser classificado como uma inovação, embora alguns dados já apontem para um novo cenário, no qual o desenvolvimento ágil está em um momento de inflexão, passando a ser defendido também na esfera gerencial das organizações.

Uma deficiência importante, apontada por este trabalho, foi a ausência de dados que possam ser usados para avaliar,

quantitativamente, a presença das diversas metodologias ágeis no mercado de desenvolvimento de software, bem como as adaptações que se fazem necessárias e o grau de satisfação de seus usuários. Essa ausência é ainda mais sentida quando se procuram por dados referentes ao mercado nacional. ●

#### Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

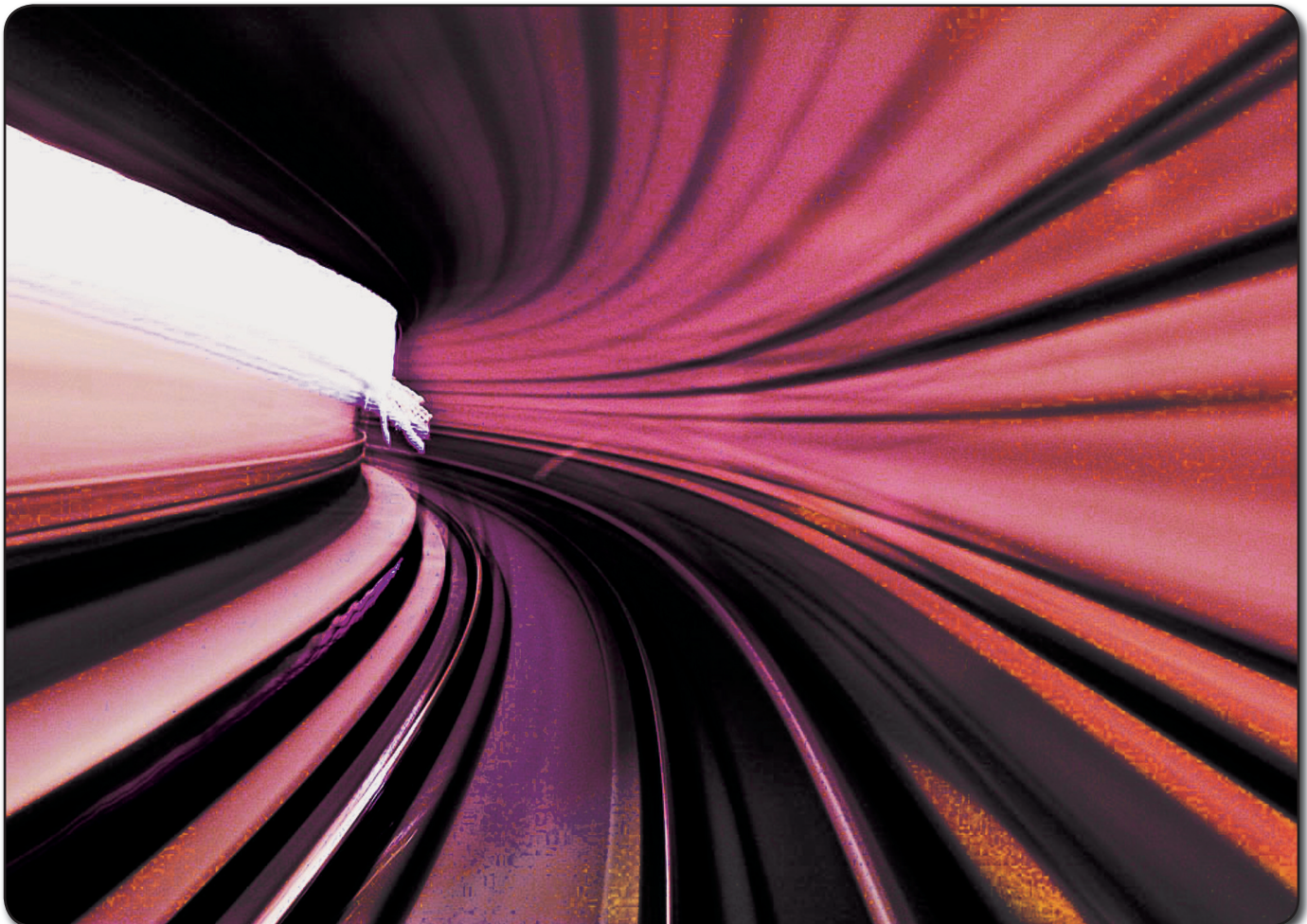
Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



## Referências

- ABRAHAMSSON, Pekka et al. Agile software development methods: Review and analysis. VTT Publications 478. Oulu, Finland: VTT Publications, 2002.
- AMBLER, Scott W. Imperfectly Agile: You Too Can Be Agile. Dr. Dobb's Portal. Setembro, 2006. Disponível em <[www.ddj.com/architect/192700252](http://www.ddj.com/architect/192700252)>. Acesso em 17/01/2008.
- \_\_\_\_\_. Survey Says... Agile Has Crossed The Chasm. Dr. Dobb's Portal. Julho, 2007. Disponível em <[www.ddj.com/architect/200001986](http://www.ddj.com/architect/200001986)>. Acesso em 12/02/2008.
- BARNETT, Liz. Agile Survey Results: Solid Experience And Real Results. Agile Journal. Setembro, 2006. Disponível em <[www.agilejournal.com/articles/from-the-editor/agile-survey-results%3a-solid-experience-and-real-results.html](http://www.agilejournal.com/articles/from-the-editor/agile-survey-results%3a-solid-experience-and-real-results.html)>. Acesso em 16/02/2008.
- BONA, Cristina. Avaliação de Processos de Software: Um Estudo de Caso em XP e ICONIX. Dissertação de Mestrado em Ciências da Computação. Universidade Federal de Santa Catarina, 2002.
- FOWLER, Martin. The New Methodology. 2005. Disponível em: <[www.martinfowler.com/articles/newMethodology.html](http://www.martinfowler.com/articles/newMethodology.html)>. Acesso em 18/08/2007.
- FRANSSON, Oskar; af KLERCKER, Patrick. Agile Software Development in Sweden – A quantitative study of developers' satisfaction and their attitude towards agile thinking. Dissertação de mestrado em Informática. Jönköping University: Sweden, 2005.
- KRUCHTEN, Philippe. Agility with the RUP. Cutter IT Journal. v. 14, n. 12, Dezembro de 2001. Disponível em: <[www.emory.edu/BUSINESS/readings/MethodologyDebate.pdf](http://www.emory.edu/BUSINESS/readings/MethodologyDebate.pdf)>. Acesso em 22/05/2007.
- GRINYER, Antony R. Investigating the Adoption of Agile Software Development Methodologies in Organizations. Technical Report nº 2007/11. Faculty of Mathematica and Computing. The Open University, United Kingdom: Julho, 2007.
- HARRISON, Warner. Skinner Wasn't a Software Engineer. IEEE Software. Maio / Junho 2005. Disponível em <[www.computer.org/portal/cms\\_docs\\_software/software/content/skinner.pdf](http://www.computer.org/portal/cms_docs_software/software/content/skinner.pdf)>. Acesso em 05/06/2007.
- KOHRELL, David; WONCH, Bill. Using RUP to manage small projects and teams. The Rational Edge. Julho, 2005. Disponível em <[www.ibm.com/developerworks/rational/library/jul05/kohrell/index.html](http://www.ibm.com/developerworks/rational/library/jul05/kohrell/index.html)>. Acesso em 18/01/2008.
- MARCHESI, Michele et al. eXtreme Adoption eXperiences of a B2B Start Up. In: Extreme Programming Perspectives. Pearson Education, 2002.
- MARTIN, Robert C. Agile Processes. 2002. Disponível em <[www.objectmentor.com/resources/articles/agileProcess.pdf](http://www.objectmentor.com/resources/articles/agileProcess.pdf)>. Acesso em 15/08/2007.
- NONEMACHER, Marcos L. Comparação e Avaliação entre o Processo RUP de Desenvolvimento de Software e a Metodologia Extreme Programming. Dissertação de Mestrado em Ciências da Computação. Universidade Federal de Santa Catarina, 2003.
- POLLICE, Gary. Using the IBM Rational Unified Process for Small Projects: Expanding Upon eXtreme Programming. Rational Software White Paper, 2001. Disponível em <<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/tp183.pdf>>. Acesso em 22/05/2007.
- PROJECTS@WORK. Agile 2006 Roundup. Agosto, 2006. Disponível em <[www.projectsatwork.com/article.cfm?ID=232426](http://www.projectsatwork.com/article.cfm?ID=232426)>. Acesso em 16/02/2008.
- SABRE AIRLINE SOLUTIONS. Sabre takes extreme measures. Computerworld Inc. Março, 2004. Disponível em <[www.computerworld.com/developmenttopics/development/story/0,10801,91646,00.html](http://www.computerworld.com/developmenttopics/development/story/0,10801,91646,00.html)>. Acesso em 16/02/2008.





## Por que SCRUM?



**Isabella Fonseca**

*isabella@powerlogic.com.br*

Atua desde 1999 como consultora em e-Business para grandes empresas utilizando Java EE. É Certified ScrumMaster e gerente da equipe de desenvolvimento e de projetos do eCompany Portal - primeira solução de EIP (Enterprise Information Portal) do país, e do eCompany Process - solução de definição e controle de Processos Corporativos integrada ao Gerenciamento de Projetos (EPM e APM), de Requisitos e de Produtos (Application Lifecycle Management). Ambos são gerenciados com SCRUM e certificados MPS.Br nível F.



**Alberto Campos**

*alberto.campos@localiza.com.br*

Está na área de TI desde 1984 atuando com gerenciamento de projetos ágeis desde 2002 para projetos de grande porte multidisciplinares incluindo Engenharia, Automação Industrial e Desenvolvimento de Software. Fora do Brasil, já implantou sistemas nos Estados Unidos, Europa e Japão.

A indústria de desenvolvimento de software evoluiu para se tornar uma das mais importantes instituições de nosso tempo criando produtos essenciais para o nosso dia a dia. Inúmeros exemplos podem ser citados. Neste ambiente intensamente competitivo, diferenciais devem ser criados para a sobrevivência. A capacidade de criar e de entregar mais rapidamente produtos de software que melhor satisfaçam as necessidades reais do cliente é um deles.

Segundo PMI Brasil 2006, os problemas mais frequentes em gerenciamento de projetos levantados são:

- Não cumprimento de prazos (72%)
- Problemas de comunicação (71%)
- Mudança de escopo (69%)
- Estimativa errada de prazo (66%)

Analisando os problemas levantados, pode-se constatar que o cerne da questão reside principalmente na comunicação – refletindo nos demais. Quando presentes, mecanismos de comunicação

ineficientes são utilizados contribuindo para a falta de compreensão e colaboração por parte dos envolvidos.

A **Figura 1** promove uma comparação entre a efetividade de comunicação e a “riqueza” do canal de comunicação. Dois mecanismos são evidenciados: os que não possuem perguntas e respostas e os que possuem. Este primeiro tipo propõe pouca interação e não permite a colaboração e troca necessária para o alcance do objetivo proposto. A utilização de papel para descrever e comunicar o problema tem importante cunho “documentacional”, além de poder ser compartilhado entre diversas pessoas.

Por outro lado, mecanismos que promovem e facilitam a comunicação, os mecanismos interativos, são compostos por perguntas e respostas - como duas pessoas falando ao telefone ou discutindo via email, são notoriamente mais eficientes. E ainda torna-se mais eficiente se tivermos a presença física na mesma sala dos diversos envolvidos. Para completar,

um quadro branco auxiliando na dinâmica das discussões, pode expandir em muito as possibilidades de entender e se fazer entendido.

Comunicação “face-to-face” é uma das formas de se resolver grande parte dos problemas citados e, metodologias ágeis, como o SCRUM, pregam incessantemente esta prática. O Manifesto Ágil, criado em 2001, instituiu valores e princípios da escola que corroboram com o descrito acima. Abaixo os quatro valores principais:

*“Nós estamos descobrindo melhores formas de se desenvolver software, fazendo software e ajudando os outros a fazê-lo.*

*Através deste trabalho nós viemos valorizar:*

*Indivíduos e Interações primeiro que Processos e Ferramentas.*

*Software funcionando primeiro que Documentação Compreensiva.*

*Colaboração do Cliente primeiro que Negociação Contratual.*

*Resposta a Mudanças primeiro que Conformidade com Planejamento.*

*Isto é, embora reconheçamos que há valor nos itens à direita, nós valorizamos primeiramente os itens à esquerda.”*

É importante destacar que não há ruptura entre os itens da esquerda e os da direita e sim de ênfase. A comunicação irá auxiliar em cada um dos itens da esquerda e será o grande facilitador desse processo.

O interessante deste tipo de abordagem é que não se tem a procura por um culpado. Todos estão imbuídos na busca da melhor solução para a organização, seja o analista, desenvolvedor, gerente de projeto ou o cliente. E cada um tem o seu papel claramente determinado frente ao processo de desenvolvimento. Em metodologias ágeis, uma das premissas básicas é ter o cliente sempre por perto e fazê-lo um participante ativo. É importante que ele entenda suas responsabilidades e sua grande parcela de contribuição para o sucesso do projeto.

Algumas questões diferenciam metodologias ágeis no planejamento do projeto. Nas metodologias ágeis, o planejamento é feito continuamente, durante todo o projeto, e baseado em um goal (objetivo), onde são definidas as tarefas

necessárias para se iniciar o mesmo. Este objetivo serve como um guia para todos os envolvidos e é a meta a ser perseguida. Normalmente é um parágrafo que resume, em três ou quatro frases, o foco maior que o projeto deve ter. Seu estabelecimento visa delegar à equipe do projeto um espaço de agilidade que a permita tomar decisões rápidas ao longo do projeto, norteando-o quanto a alterações de prioridades, de requisitos, práticas ou outras quaisquer.

Se todo o planejamento fosse feito no início do projeto, após as primeiras semanas, o longo planejamento feito poderia estar defasado, pois as mudanças ocorreriam e o forçariam a esta situação. Requisitos não devem ser intensamente esgotados no início do projeto, eles sim, são incrementados a cada iteração, experimentados pelos envolvidos, colocados a prova, avaliados e refinados.

É importante construir o todo aos poucos, refinando os requisitos, equalizando o entendimento sobre eles e corrigindo o rumo sempre que uma mudança provoque alterações significativas.

Planejar detalhadamente é ir contra a natureza do processo de desenvolvimento de software. As tarefas que constituem este tipo de abordagem fazem parte de um processo criativo, não linear e não palpável, fazendo com que modelos ágeis se apresentem como uma alternativa interessante. Por isso, planos menos detalhados e feitos em frequência maior se mostram mais adequados.

A forte presença de iterações curtas também contribui para o planejamento contínuo citado. Elas garantem ritmo a

todos os envolvidos e levam ao feedback real e imediato dado pelo cliente. Como resultado, auxiliam nos possíveis ajustes - que não acontecem mais tardiamente - e ajudam na entrega de software de valor. Cada uma das iterações é composta de todas as tarefas necessárias a esta entrega: levantamento de dados, análise, projeto, implementação, testes e integração a versão anterior.

Outro ponto interessante é a maneira pela qual o monitoramento e a medição do projeto são executados em metodologias ágeis. Cada tarefa definida é verificada quanto ao seu percentual de andamento (o mais usual) e também quanto à sua qualidade através de testes e entregas que são feitas com maior frequência. O progresso do projeto não é medido levando-se em consideração o plano inicial feito, como em metodologias tradicionais, ou seja, não é traçado um quadro comparativo entre o que foi planejado no início e o que foi executado até o momento. Mudanças são aceitas e algumas “linhas de bases” farão parte do projeto. Dessa forma, o atendimento ao goal definido deve ser a principal medida de progresso sendo verificado, constantemente, para garantir o máximo de retorno para o negócio.

A identificação e o gerenciamento dos riscos em metodologias ágeis são feitos em taxas diárias (através de reuniões curtas e diárias) e durante toda a iteração, o controle da qualidade dos trabalhos é avaliado. Esta tarefa pode ser executada através de testes, revisão por pares, inspeção contínua e acompanhamento pelo cliente, que seguem o mesmo processo.

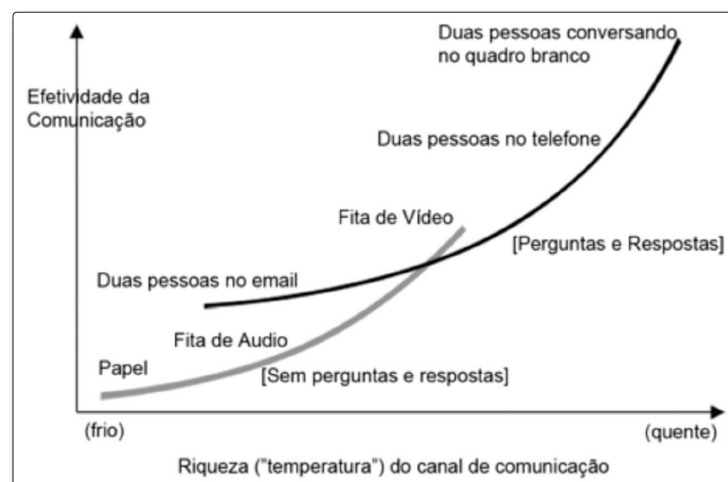


Figura 1. Temperatura da comunicação segundo Alistair Cockburn.

Impedimentos são levantados e devem ser resolvidos no prazo máximo de 24 horas para garantir que ações de resolução rápidas são executadas e para que todos conheçam os impedimentos que podem se tornar potenciais riscos para o projeto.

Além disso, mudanças no projeto são bem aceitas, pois elas existem e irão acontecer. É por isso que o comprometimento de todos os envolvidos é tão importante em um projeto. Caso o cliente queira mudar uma solicitação ou adequar-se a uma mudança de mercado, pode-se alterar o rumo rapidamente sem afetar todo o projeto pois, o planejamento é refeito a todo o momento. E ainda, caso uma iteração não corra conforme o esperado pelo cliente pode-se mudar a abordagem de levantamento de dados, os recursos envolvidos, a forma de troca de informações e corrigir o rumo a tempo do final do projeto.

Pode ser (e na maioria das vezes, é) que o plano inicial não represente mais a necessidade real de nosso cliente. Ele deixa de pedir alterações, pois o contrato já foi assinado, ele não tem mais horas adicionais para pagar, o prazo está curto e ele necessita do produto e então... ele “evita” solicitar mudanças. Pergunta: o que ele ganha? De que adianta entregar um software que o cliente não irá usar? E o que o fornecedor do serviço ganha? Ok, entrega-se o produto, o cliente usa poucas funcionalidades, reclama das funcionalidades prioritárias que ele não recebeu e em contrapartida, recebeu várias outras que não vai utilizar. E isso pode ser considerado um projeto de sucesso?

O sucesso de um projeto de software utilizando metodologias ágeis não é somente medido se comparando projeções iniciais de prazo, custo e escopo e sim através de entrega de software

de valor ao cliente. Estes três pilares – prazo, custo e escopo - devem ter seus pesos definidos. É claro que todos nós desejaríamos que nossos projetos terminassem dentro do prazo previsto, com o custo dentro da margem determinada e com todo o escopo entregue. Todos nós já passamos por fases posteriores de manutenções e correções que consomem grande esforço de recursos e também traz desgaste no relacionamento das pessoas envolvidas.

Neste artigo, faremos uma introdução ao Scrum. Mas antes disso, para auxiliar no entendimento, utilizaremos um exemplo buscando dar uma visão mais simplista da metodologia ágil Scrum. Como diz o provérbio chinês: “O que ouço, esqueço. O que vejo, lembro. O que faço, entendo”. Portanto, segue um relato que muitas famílias já vivenciaram. Algumas correlações serão feitas no decorrer da estória – colocamos os termos Scrum entre parênteses. Mais tarde, estes conceitos serão desvendados.

Resolvemos planejar uma viagem em família. Para isso, algumas perguntas surgiram:

- Qual seria o projeto?
- Quem seriam nossos clientes?
- Como melhor atendê-los sabendo que mudanças podem ocorrer e mudar nosso rumo?
- Como e quais decisões tomar para reagir a tempo não frustrando as expectativas?
- Como saber se o projeto está alcançando o resultado esperado?
- Como aprender com a experiência vivida?

Projeto: viagem de família à praia.

Clientes: nossos filhos – Luiza, de 9 anos, Gabriel, de 7 anos e Bernardo - 13 anos (existem clientes mais exigentes?).

Custo: menor possível.

Prazo: final de semana prolongado.

Nossos filhos faziam planos para este feriadão há muito tempo e não viam a hora de chegar à praia. O carro já estava na revisão e as malas todas prontas.

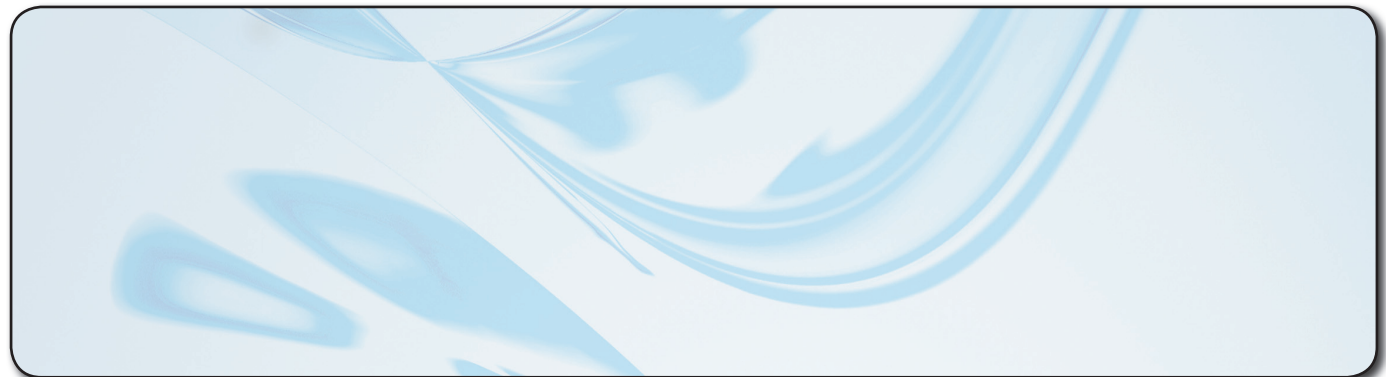
Começaríamos a viagem no final da tarde de quinta-feira. Poderíamos chegar à noite, aproveitar os dias de sexta e sábado e retornar no domingo. A viagem começou e as crianças estavam eufóricas, muita gritaria, bagunça, fome, comida, pedidos para parar, ir ao banheiro, colo de mãe. E perguntas do tipo: quando vamos chegar? Já chegou? Como é lá? Problemas com o carro. Será que conseguiríamos chegar? Noite em hotel... decepção. Carro ok, continuamos a viagem, e enfim, chegamos...

Bom, tudo isto ocorreu conosco durante nossa ida e para a volta, resolvemos fazê-la a “La Scrum”.

Seguindo a metodologia, decidimos dividir a viagem de volta (Release) em etapas (Sprints), com duração prevista de duas horas ou 200 km. Cada uma com um resultado esperado (Sprint Goal), que eram acompanhadas de perto por nossos filhos. O projeto também teve seu objetivo maior definido (Release Goal ou Vision). Sendo assim, as expectativas de nossos “clientes” ficaram claramente conhecidas.

Release Goal: Chegar em casa com segurança aproveitando cada momento juntos.

Durante cada etapa, estávamos sempre verificando o andamento das atividades planejadas - o que foi feito ontem?, planejando as próximas - o que vai fazer amanhã? e removendo problemas que surgiam – que impedimentos ocorreram (Daily Scrum). Além disso, atualizávamos em nosso mapa (Agile Radiator) nossa evolução. Dessa forma, todos os envolvidos tinham visualmente o real





andamento e poderíamos identificar qualquer desvio indesejado. Tudo isso tendo como guia o Sprint Goal, também estipulado por eles.

Ao final de cada etapa parávamos para comer alguma coisa, descansar, esticar as pernas e perguntar a nossos filhos o que estavam achando de nossa viagem. Realizávamos também uma avaliação (Retrospective Meeting), analisando pontos positivos e negativos e seguíamos para a etapa seguinte (Novo Sprint).

Todos tinham suas responsabilidades e papéis definidos e nos dávamos o direito de rearranjar a próxima etapa se algum imprevisto ocorresse. Tendo sempre em mente o que foi determinado (goal), vez por outra, resolvíamos parar mais em uma agradável cidade, alterávamos o horário do jantar para curtirmos um pôr do sol em outra ou ir a uma festa típica que estava acontecendo.

Terminada a viagem, já em casa, fizemos a última retrospectiva de tudo o que ocorreu para nos prepararmos melhor para a próxima viagem, que terá uma duração maior, vários tipos de transporte e destinos e ainda levaremos nossos primos.

Esta foi somente uma estória de como podemos utilizar as práticas do Scrum em nossa vida cotidiana. Na próxima seção, explicaremos um pouco mais sobre o framework de processos.

### O Scrum

O Scrum é um framework de processo ágil utilizado para gerenciar e controlar o desenvolvimento de um produto de software através de práticas iterativas e incrementais. É composto por um conjunto de boas práticas de gestão que admite ajustes rápidos, acompanhamento e visibilidade constantes e planos realísticos.

Os principais papéis do Scrum são: Product Owner, Scrum Master e Scrum Team (equipe do projeto). Não há como fazermos um mapeamento direto entre os papéis do Scrum e os papéis convencionais conhecidos. Não existe a figura única do Gerente de Projetos. Suas responsabilidades estão diluídas entre os papéis citados. Cada um conhece sua participação frente ao projeto e trabalha em conjunto para conseguir alcançar o goal definido.

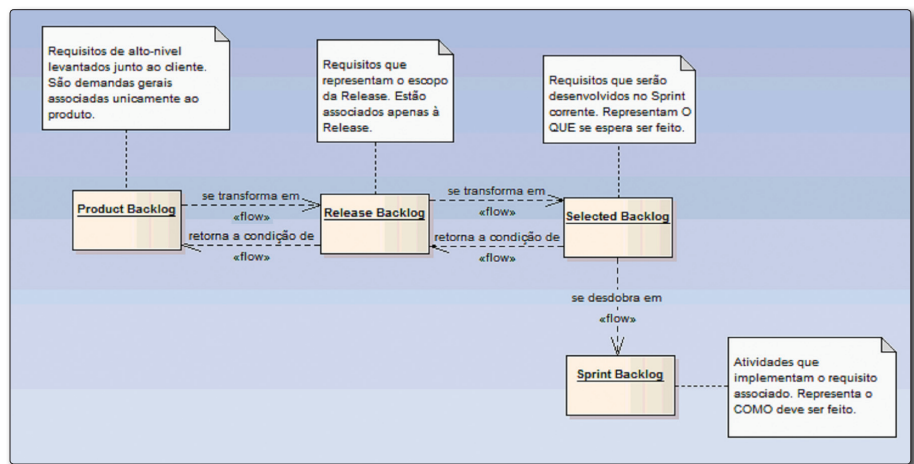


Figura 2. Ciclo de vida de um requisito e desdobramento em atividades

Seus artefatos principais são o Product Backlog e Sprint Backlog – artefatos que representam seus requisitos/atividades além de Burndown charts e impediment backlogs. Para representar o ciclo de vida de um requisito, usa-se derivações do Product Backlog, como Release Backlog e Selected Backlog. A Figura 2 exemplifica o ciclo mencionado..

Inicialmente um requisito deve fazer parte do Product Backlog, que possui descrições de necessidades de clientes de alto nível levantadas. A tarefa de manter a descrição e refinamento destes requisitos é do Product Owner. Ele é responsável por definir, para cada nova release de um produto, o objetivo (Release Goal ou Vision). Ele interage a todo momento com seu cliente final e, dessa forma, conhece suas expectativas e as do mercado. Esta lista de demandas nunca acaba e de tempos em tempos (a cada nova release), um item de Product backlog é promovido a Release backlog para integrar ao escopo de uma ou mais Releases. O Product Owner deve comunicar e discutir o novo goal com todos os envolvidos em reuniões chamadas de Release Planning.

Portanto, a primeira reunião oficial de Release que ocorre é a Release Planning. Nesta reunião, são discutidos os recursos envolvidos – tanto computacionais quanto humanos, riscos identificados, prazo acertado e requisitos previstos acordados junto ao cliente e coerentes com a demanda de mercado. Seu propósito principal é definir e comunicar as funcionalidades macro de uma Release do produto e fazer com que os envolvidos no projeto entendam e se comprometam com o Release

Goal definido. As funcionalidades não devem ser discutidas em detalhes. Isso será feito em cada Iteração durante reuniões de Sprint Planning. Ela pode ter a presença de diversos papéis: seja cliente externo, desenvolvedor, área comercial, etc.

Cada release é composta de iterações curtas, chamadas de Sprints. A Figura 3 traz um desenho das reuniões existentes durante o período de uma Release, composta por 3 Sprints. Conforme dito, a primeira reunião é a Release Planning. Cada Sprint tem reuniões de planejamento, Sprint Planning, que detalham o requisito apresentado durante o Release Planning e ainda possuem seu Sprint Goal. Semelhante ao goal do Release, tem o mesmo cunho de comunicar a todos os envolvidos o objetivo maior de Sprint corrente. É realizado de forma iterativa, através de ciclos de “tempo fechado” (Time-Boxed) em 30 dias corridos. Ainda sobre a reunião, são conhecidos os requisitos do Release Backlog que serão promovidos como Selected Backlog. Eles são discutidos em maiores detalhes e servem para elucidar O QUE se espera como resultado final. Em um Sprint, portanto, são executadas atividades de refinamento de requisitos, análise, projeto, desenvolvimento e testes pelo Scrum Team, devendo resultar em um incremento de produtos funcionando e demonstrável para o Product Owner ao final do Sprint, na reunião de Sprint Review.

O Scrum Master tem o papel de liderança muito importante para o processo. Ele deve remover todo e qualquer obstáculo que surgir durante o desenvolvimento, garantindo que o Scrum Team

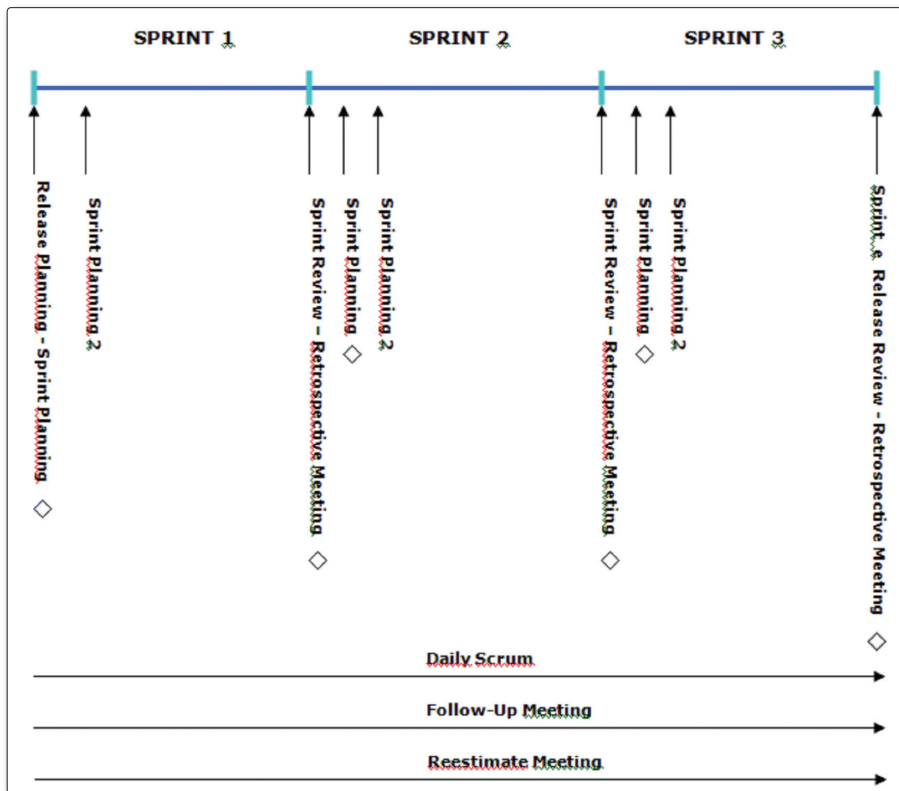


Figura 3. Reuniões Scrum

possa focar no real objetivo definido. Além disso, ele é responsável por fazer com que a equipe siga e pratique o processo e ainda por criar uma atmosfera de ajuda mútua entre a equipe (o resultado é sempre da equipe e não individual).

O Scrum Team é responsável por se organizar e determinar a melhor estratégia de se entregar as funcionalidades de maior prioridade. Interessante observar que iterações curtas garantem ritmo à equipe e facilita o entendimento do processo, além de proporcionar visibilidade ao cliente sobre o andamento do projeto.

Diariamente são executadas reuniões de acompanhamento e monitoramento do Sprint através da reunião de Daily Scrum. Esta reunião deve ter a presença do Scrum Team e Scrum Master obrigatoriamente. Deve ter a duração máxima de 15 minutos (é interessante até utilizar um cronômetro para isso) e são permitidas somente 3 perguntas:

- O que você fez hoje?
- O que fará amanhã?
- Que impedimentos surgiram e que atrapalharam sua produtividade?

Outros assuntos deverão ser discutidos em outras reuniões, como Follow-Up Meetings para detalhamentos e elucida-

ções relativas aos requisitos e Reestimate Meetings para aplicação de técnicas como o Poker Planning para estimativa de requisitos.

Há uma reunião de Sprint Planning 2 que promove o desdobramento do Selected Backlog em Sprint Backlog onde aspectos técnicos são discutidos. É a hora do COMO as coisas devem ser feitas. Neste momento, pode-se discutir sobre a arquitetura do produto, reuso de componentes, mudança em interfaces, etc.

Ao final do Sprint, é feita uma reunião final de apresentação do trabalho executado ao Product Owner chamada Sprint Review. Ele será responsável por validar a apresentação e concluir se o Sprint Goal foi atingido. Após esta, reuniões de debrief são executadas, chamadas Retrospective Meetings. Reuniões de retrospectiva levam a reflexão acerca do que passou e quais atitudes devem ser tomadas para o correto atingimento dos objetivos. São levantados aspectos positivos (WWW - what went well) e negativos (WCBI - what can be improved) tanto organizacionais quanto relativos a Release. Todos os envolvidos devem estar presentes e são responsáveis por minimi-

zar todo e qualquer risco ao projeto e também por manter os aspectos positivos levantados.

E o mesmo se repete por todos os Sprints planejados, com a diferença de que o último Sprint possui uma reunião de Release Review, que tem como guia o Release Goal definido. Nunca é demais lembrar que o planejamento é feito durante toda a Release e portanto, alterações podem ser feitas. Se constitui em uma boa prática poder refazer o planejamento durante cada Sprint Planning, mas durante o Sprint, o Scrum Master deve procurar “blindar” o Scrum Team de grandes modificações.

As estimativas de tamanho de cada funcionalidade/requisito são executadas obrigatoriamente pelo Scrum Team. Dessa forma, todos sabem para onde ir (goal) e é desta forma que se obtém o comprometimento da equipe para com o projeto. A viabilidade de continuidade dos planos é constantemente verificada.

A comunicação de um projeto que utiliza o Scrum é mais efetiva e direta, além de informações estarem sempre à tona com a utilização de quadros brancos (Agile Radiator). A Figura 4 exemplifica um Agile Radiator monitorando um projeto real. Eles garantem visibilidade do projeto a todos os envolvidos. Não há como mascarar o real andamento. O goal fica afixado e os requisitos – através de Post-its - (Selected backlogs) e seus desdobramentos (Sprint backlogs) são posicionados na situação onde se encontram (se ainda não iniciados - planejados, se sendo executados no momento - em andamento e se terminados – 100% concluídos). Eles devem ser posicionados de acordo com a prioridade dos mesmos – Business Value declarado pelo Product Owner. Post-its localizados no topo nos dizem ser de maior prioridade que os posicionados no rodapé do quadro branco.

Impedimentos (Impediment Backlog) que ocorrem durante o Sprint também são evidenciados. O Scrum tem um gráfico, de nome BurnDown Chart, que exibe o real andamento do mesmo. O Scrum Team diariamente, através de reuniões chamadas Daily Scrum, atualiza o Agile Radiator com a situação atual, movendo post-its e indicando o trabalho executado durante o período.

Vale destacar que a granularidade de cada Sprint Backlog deve ser pequena. Sua estimativa não deve ultrapassar 8 horas de trabalho e isso se constitui em uma boa prática no gerenciamento do andamento das atividades. Através desta reunião, o time consegue avaliar se é necessário mudar o plano para acertar o rumo, se deve priorizar alguma outra atividade, se muitos impedimentos estão ocorrendo e como minimizá-los, se há algum recurso que está necessitando de ajuda, etc. É real! Isso tudo sempre à luz do goal definido. Além de todas estas vantagens, ainda proporciona visibilidade a todos os envolvidos.

Durante cada reunião de Daily Scrum, post-its são “movimentados” pelo Scrum Team contemplando cada uma das perguntas associadas a esta reunião. Atividades executadas (o que foi feito hoje) são movidas para a seção “Concluídos”. Atividades planejadas (o que será feito amanhã) são colocadas na seção “Em Andamento” e impedimentos são registrados (que impedimentos surgiram? Repare a seção em vermelho no rodapé a direita).

Além destes artefatos, o Burndown Chart deve ser atualizado ao final da reunião contabilizando o tamanho total entregue pelo Scrum Team no dia a

ser representado. O Burndown possui duas retas: uma que exhibe o planejado e outra que reflete o realizado. Ele está registrado no topo à esquerda. Ele contém o total de trabalho restante e só de “batermos o olho” no gráfico conseguimos verificar a evolução do trabalho. Todo o Agile Radiator tem esta característica. Se estivermos vendo muitos impedimentos, temos que verificar o que está sendo feito para prevenir novos e se estes estão sendo corretamente removidos. Se atividades de maior prioridade estão sendo deixadas para trás, a própria equipe pode verificar o que está havendo. O que foi bom (WWW) e o que pode ser melhorado (WCBI) também ficam afixados no quadro branco.

A cada novo Sprint, o Agile Radiator deve ser modificado para retratar a situação atual novamente. E este ciclo se inicia: planejamento, execução, controle e avaliação do que foi feito para cada nova Release necessária.

### Considerações Finais

Uma confusão ou uma análise superficial que não podemos deixar de citar é o entendimento de que metodologias ágeis são pouco formais ou que indo por este caminho, você estará li-

vre de documentações ou burocracias. É necessária muita disciplina para seguir esta abordagem. Seus princípios e valores são baseados em dedicação e bom senso de todos os envolvidos. Suas práticas pregam inspeções constantes - para o feedback rápido - e aceitação das mudanças e adaptações que o processo deva passar. O comprometimento entre todos os envolvidos também se constitui como um grande diferencial do framework. O processo de gestão está permeado entre os diversos papéis existentes como falado anteriormente.

Estas são somente práticas de um framework de processo que nos guia em como fazer. Ele está baseado no Manifesto da Agilidade, que em 2001 reuniu vários gurus para a discussão sobre processos de desenvolvimento de software. Não é uma anarquia! É um convite a pensar diferente. Cabe a cada um de nós descobrir o melhor processo dentro da organização que trabalhamos. Nossa experiência nos leva a combinar algumas abordagens. E elas têm se demonstrado bastante efetivas. Mas, conforme dissemos, isso se dará de forma diferente para cada um. O importante é encontrar o seu caminho. ●



Figura 4. Agile Radiator

### Bibliografia

Agile Alliance, 2002. Agile Manifesto, <http://www.agilealliance.org>  
 Artigo: “Get Ready for Agile Methods, with Care”, Computer, 35, 1 (January 2002) 64-69 escrito por Boehm, B.  
 Agile Software Development with Scrum, Prentice Hall, (October 2001). Schwaber, K; Beedle, M,  
 Agile Software Development, Cockburn Highsmith Series Editors, Alistair Cockburn 2000

### Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:  
[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)





## Tendências na área de Gestão de Riscos em Ambientes de Desenvolvimento de Software

A gerência de risco não deve, em absoluto, ser entendida e utilizada sob uma conotação negativa, pela qual se visaria tão só avaliar e resolver os eventos adversos. Ao contrário, deve visualizar oportunidades: vantagens estratégicas e diferenciais competitivos através da execução de atividades preventivas. As organizações e empresas devem ser proativas, monitorando os riscos de seus projetos com a finalidade de agregar valor e de alçar novas oportunidades não só de negócios, mas de conhecimento adquirido.

Muitas organizações podem alegar que não necessitam executar nenhum estudo ou qualquer outro procedimento desta natureza, pois empregam pessoas suficientemente competentes e processos absolutamente seguros; confiança nos conhecimentos e na experiência de seus funcionários, o que elimina a possibilidade de ocorrência de falhas em seus domínios industriais.

### **De que se trata o artigo:**

Aborda temas relacionados ao gerenciamento de riscos nos ambientes de desenvolvimento de software.

### **Para que serve:**

Fornecer uma visão horizontal sobre o gerenciamento de riscos através de variados temas gerenciais e estratégicos. Também facilita a aplicação de conceitos de maturidade organizacional nos ambientes de desenvolvimento.

### **Em que situação o tema é útil:**

Além de ser atual, a preocupação e conscientização dos ambientes organizacionais no gerenciamento dos riscos é um primeiro passo para a minimização das falhas ainda existentes no gerenciamento de projetos em ambientes de desenvolvimento de software.

Estas, pois, são as mais indicadas a serem surpreendidas por situações adversas e não previstas que resultam, muitas vezes, em graves problemas causando grandes perdas, tanto mercadológicas como materiais.



### **Cristine Gusmão**

*cristine@dsc.upe.br*

Professora Assistente do Departamento de Sistemas Computacionais da Escola Politécnica da Universidade de Pernambuco (POLI – UPE), onde leciona várias disciplinas na graduação e pós-graduação (especialização e mestrado) e das Faculdades Integradas Barros Melo. Doutora e Mestre em Ciência da Computação pela Universidade Federal de Pernambuco. Graduada em Engenharia Elétrica – Eletrotécnica pela Universidade Federal de Pernambuco.

### Gerência de Múltiplos Projetos

O resultado de todos os projetos desenvolvidos por uma organização tem grande parcela de contribuição no seu sucesso. Projetos individuais influenciam a organização, mas também sofrem a influência de todos os outros projetos que estejam sendo iniciados, ou mesmo, executados num mesmo período. Nenhum projeto é desenvolvido isoladamente. Existem os projetos estratégicos, projetos embargados e os projetos de manutenção. Priorizar e garantir que os projetos mais importantes sejam realizados é um dos grandes, se não vital, objetivos organizacionais.

Outro grande desafio encontrado pelos profissionais de projetos de Tecnologia da Informação e Comunicação (TIC) é estabelecer um método para seleção, rastreamento e controle de projetos. A grande maioria das organizações não tem condições de manter uma equipe dedicada a cada um dos seus projetos. Os funcionários vão sendo deslocados entre os projetos de acordo com a necessidade de cada um deles.

Outra característica importante e bastante comum nestas organizações é que o orçamento mensal de cada projeto pode ficar totalmente comprometido ou extrapolar o planejado, devido a imprevistos não tratados. Neste caso, a solução é remanejar recursos financeiros de outros projetos que não estejam tão comprometidos.

Este ambiente dinâmico no qual a alocação de recursos é elemento-chave é conhecido como ambiente de múltiplos projetos. Pouco mais do que 90% de todos os projetos são conduzidos neste tipo de ambiente [Danilovic e Borjesson 2001].

Portanto, além de complexas variáveis que cercam um único projeto, outras dificuldades surgem quando passam a existir diversos projetos executados simultaneamente. É comum o lançamento de projetos faltando recursos e com programação deficiente. Isto promove a re-priorização dos projetos, subprojetos e tarefas, ou seja, no momento em que o prazo de algum dos projetos esteja vencendo ele passa a ser o foco das atenções [Freitas 2005].

Em um momento posterior ele pode ser relegado ao segundo plano em detrimento de outro que esteja na mesma situação.

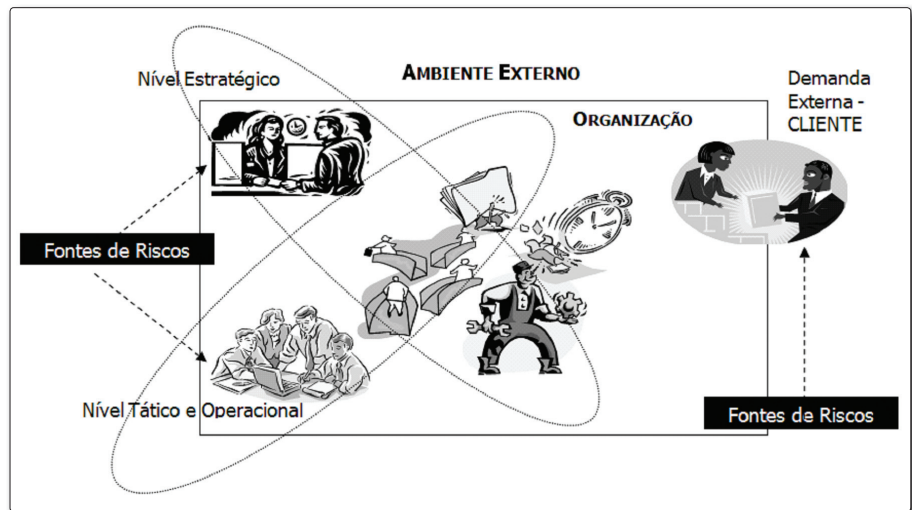


Figura 1. Níveis hierárquicos de uma organização

A alocação de recursos então deve ser feita no momento em que os projetos precisam e não através de um planejamento prévio. O resultado pode ficar comprometido pela ausência do recurso no momento em que o mesmo é necessário, recorrendo a soluções paliativas drásticas que comprometem o orçamento, a qualidade e o cronograma do projeto.

Considerando então a natureza mutável dos recursos entre os projetos, o problema da comunicação toma proporções ainda maiores. Isso gera conflitos, sentimento de insegurança, estresse e desconforto, entre a equipe de desenvolvimento, pois a mobilidade das pessoas entre os projetos por muitas vezes não permite que elas tenham um conhecimento mais aprofundado do que estão desenvolvendo.

As organizações estão estruturadas primariamente em três níveis, conforme mostra a Figura 1: estratégico, tático e operacional. O nível estratégico é composto pela alta administração executiva da organização e é responsável pela definição das metas de médio e longo prazo que estejam alinhadas às estratégias da organização. É no nível estratégico que ocorre a seleção e priorização dos projetos, também conhecida como Gerência de Portfólio de Projetos [Dye e Pennypacker 2000].

O nível tático tem a preocupação de definir as tarefas a serem realizadas, para que os projetos de longo e médio prazo, definidos no nível estratégico, aconteçam. Este nível é composto pelos gerentes de projeto e o foco do trabalho é no gerenciamento diário das

atividades planejadas e na alocação dos recursos necessários para o andamento das atividades.

O nível operacional é composto pelos demais membros do projeto, os encarregados de executarem as atividades definidas pelo nível tático.

A Gerência de Múltiplos Projetos consiste no monitoramento contínuo dos diversos projetos de um ambiente de múltiplos projetos pela gerência, manifestando-se primordialmente no nível tático.

Na literatura de Engenharia de Software não existe uma abordagem que, explicitamente, tenha foco nos Ambientes de Múltiplos Projetos. Algumas das abordagens como o RUP (Rational Unified Process), CMMI (Capability Maturity Model Integration) e o próprio Guia PMBOK (Project Management Body of Knowledge) apresentam a Gerência de Projetos, mas ainda possuem deficiências quando se trata de múltiplos projetos [Freitas 2005].

Alguns trabalhos de cunho acadêmico vêm sendo desenvolvidos com o intuito de preencher esta lacuna. Por exemplo, Bruno Freitas [Freitas 2005] apresenta a proposta do MGMPs – Modelo para Gerenciamento de Múltiplos Projetos de Software, tendo, inclusive, definido uma ferramenta para suporte a esses ambientes – GMP (<http://www.cin.ufpe.br/~gmp/>) – gerenciador de multiprojetos. O modelo tem por base metodologias como RUP, CMMI e PMBOK. Alguns dos conceitos relacionados às metodologias são adaptados para a efetiva utilização nos Ambientes de Múltiplos Projetos de Software.

## Modelos de Maturidade

Conquistar níveis de maturidade organizacional, através da melhoria dos processos utilizados, exige esforços árduos, sistemáticos e eficazes. Atualmente têm sido relatadas muitas metodologias de Engenharia de Software sob a ótica de processos modulares, de métricas e da qualidade de software, objetivando a maturidade e a competência integral dos mesmos [Hall 1998].

A melhoria do processo de desenvolvimento visa a garantir a qualidade do processo e também a qualidade do produto. Da mesma forma, a evolução do grau de maturidade da organização com respeito à utilização de técnicas e procedimentos para o gerenciamento de riscos em projetos de software pode ser resumida nos seguintes estágios [Hall 1998]:

**Problemático.** Neste estágio a organização é caracterizada pela falta de comunicação causando falta de coordenação. Nestas organizações as pessoas estão tão ocupadas resolvendo problemas presentes que não conseguem pensar no futuro. Não existe a preocupação na identificação dos riscos, logo não são endereçados até que se tornem problemas. Desta forma a gerência de crise é uma constante. Notícias sobre riscos são encaradas como má notícia e as pessoas que se preocupam com riscos são hostilizadas.

**Atenuado.** Este estágio é caracterizado pela mudança da gerência de crise para a utilização de conceitos básicos da Gerência de Riscos. As organizações passam a introduzir os conceitos relacionados aos riscos. As pessoas passam a se preocupar com riscos, mas não os endereçam de forma sistemática e estruturada. A ênfase destes novos conceitos está nas fases iniciais do desenvolvimento.

Como as pessoas têm pouco conhecimento e experiência, se sentem inseguras em como reportar os riscos. Gerentes, na sua grande maioria, utilizam a gerência de risco para reduzir a probabilidade e consequência de riscos críticos, elaborando planos de contingência.

**Prevenido.** A Gerência de Riscos deixa de ser vista como uma atividade de gerência e passa a ser vista como uma atividade da equipe de desenvolvimento de projetos de software. Passa-se a uma fase de transição da abordagem “evitar os sintomas dos riscos” para “identificar e eliminar as raízes das causas de riscos”.

Maior visibilidade do envolvimento da equipe e eventualmente dos clientes com o gerente, entendendo que a Gerência de Riscos é um processo dinâmico e que não pode ser tratado de forma isolada.

Embora as pessoas já possuam experiência em identificar riscos, ainda se sentem inseguras em quantificá-los. As ações reativas dão lugar às ações proativas.

**Antecipado.** Neste estágio o enfoque está na transição da gerência subjetiva para a Gerência Quantitativa de Riscos. A utilização de métricas para antecipar falhas e prever eventos futuros é sistematizada. As equipes apresentam habilidade para aprender, adaptar e antecipar mudanças.

Equipe e clientes utilizam a Gerência de Riscos para quantificar riscos com razoável certeza e para focar nas reais prioridades.

**Criando Oportunidades.** Este estágio tem duas finalidades essenciais, primeiramente a visão positiva da Gerência de Riscos utilizada para inovar e criar um novo futuro, e em segundo lugar, a mudança de paradigma para “riscos percebidos como perspectiva para economizar dinheiro e fazer melhor do que o planejado”.

Desta forma, o risco, assim como qualidade, passa a ser responsabilidade de todos dentro da organização. Risco é um processo ético e contínuo, identificado e resolvido em um ambiente aberto e sem ameaças.

A maturidade das atitudes profissionais favorece uma comunicação aberta e contribuições individuais. As pessoas passam a entender que existe um custo de oportunidade associado a cada escolha e que a busca desse equilíbrio melhora o processo de decisão.

## Gerência de Portfólio de Projetos

As organizações vivem atualmente uma grande competitividade mercadológica demandando rápidas decisões, melhor alocação de recursos e uma clara definição de foco. As organizações estarão cada vez mais aptas a gerir as incertezas de seus projetos ou problemas potenciais, se nas fases iniciais da execução de projetos houver um efetivo balanceamento dos fatores de riscos associados a cada um dos projetos em execução. Buscando soluções para estes ambientes dinâmicos, a Engenharia de Software trouxe os conceitos da Gerência de Portfólio de Projetos, da área de administração e finanças [Moura et al 2004].

A Gerência de Portfólio de Projetos é uma forma de organizar o gerenciamento de ambientes de múltiplos projetos. Cooper define quatro objetivos para a utilização da Gerência de Portfólio [Cooper et al 2001]:

- Maximizar o valor do portfólio;
- Procurar o melhor balanceamento entre os projetos do portfólio;
- Assegurar o alinhamento do portfólio de projetos com a estratégia organizacional;
- Garantir que não existem muitos projetos para os recursos disponíveis.

Cooper apresenta um modelo de Gerência de Portfólio composto por dois processos [Cooper et al. 2001, Moura et al. 2004, Correia 2005]:

- Sessões de Revisão: este processo tem o objetivo de periodicamente avaliar, através de abordagem holística, o grupo de projetos analisados.
- Processo Stage/Gate: processo formal utilizado pelas organizações para decidir se o projeto será continuado ou encerrado.

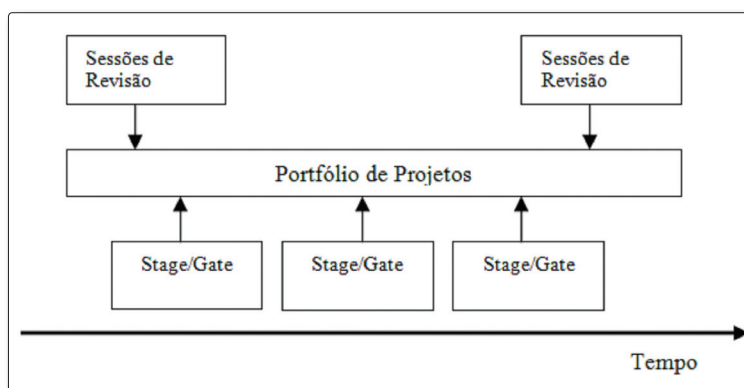


Figura 2 . Gerência de Portfólio de Projetos – Elementos do Modelo de Cooper

De acordo com Cooper [Cooper et al. 2001], a Gerência de Portfólio de Projetos é um processo dinâmico de decisão com base no tempo, conforme a **Figura 2**, onde uma lista de projetos ativos é constantemente avaliada e revisada. Neste processo, novos projetos são avaliados, selecionados e priorizados; projetos existentes podem ser acelerados, finalizados ou ter sua prioridade diminuída; recursos podem ser alocados e realocados entre os projetos ativos.

Outro modelo disponível na literatura é apresentado por Archer e Ghasemzadeh [Archer e Ghasemzadeh 1998]. Este modelo aborda a Gerência de Portfólio de Projetos como um processo passo-a-passo, conforme mostra a **Figura 3**.

A abordagem apresentada por Archer e Ghasemzadeh toma como base a Gerência de Portfólio de Projetos como um fluxo contínuo de ações, que são influenciadas principalmente pelos resultados do desenvolvimento estratégico da organização e pela metodologia de seleção de projetos escolhida [Correia 2005].

Inicialmente as propostas de projetos são avaliadas. Cada projeto é analisado individualmente. O próximo passo é escolher o portfólio com base nos projetos analisados, dentro do conjunto avaliado. Os projetos selecionados compõem um portfólio ótimo, podendo sofrer ajustes de prioridade e recursos para, então, se obter o comprometimento das principais pessoas responsáveis pela decisão sobre a necessidade de quais projetos serão desenvolvidos. Ao

final, este conjunto de projetos sugerido é submetido ao nível estratégico da organização para validação e possíveis ajustes [Moura et al. 2004].

Outros modelos foram desenvolvidos, através de estudos acadêmicos, como exemplo o modelo adaptado de André Pereira [Pereira 2002] que propõe um modelo que integra várias características importantes dos modelos de Archer e Ghasemzadeh e de Cooper, mas também incorpora outras características, que não foram consideradas e que são importantes para a Gestão de Inovação de Produtos.

O segundo modelo disponível na literatura, apresenta uma abordagem da Gerência de Portfólio de Projetos voltada para a indústria de software. O modelo apresentado por Breno Correia leva em

consideração as características dos modelos de Archer e Ghasemzadeh e de Cooper, como também algumas das considerações levantadas por André Pereira [Pereira 2002]. Uma das características importantes do modelo Portfolius é sua modularização e divisão em níveis organizacionais [Correia 2005].

Para melhor visualizar as similaridades existentes entre as atividades do processo de Gerência de Portfólio e as do processo de Gerência de Riscos, a **Tabela 1** apresenta um estudo baseado no balanceamento do portfólio [Moura et al. 2004], com as abordagens de Cooper e Archer, através de suas atividades de seleção de projetos em comparação com as atividades propostas pelo processo de Gerência de Riscos do Guia PMBOK [PMI 2004].

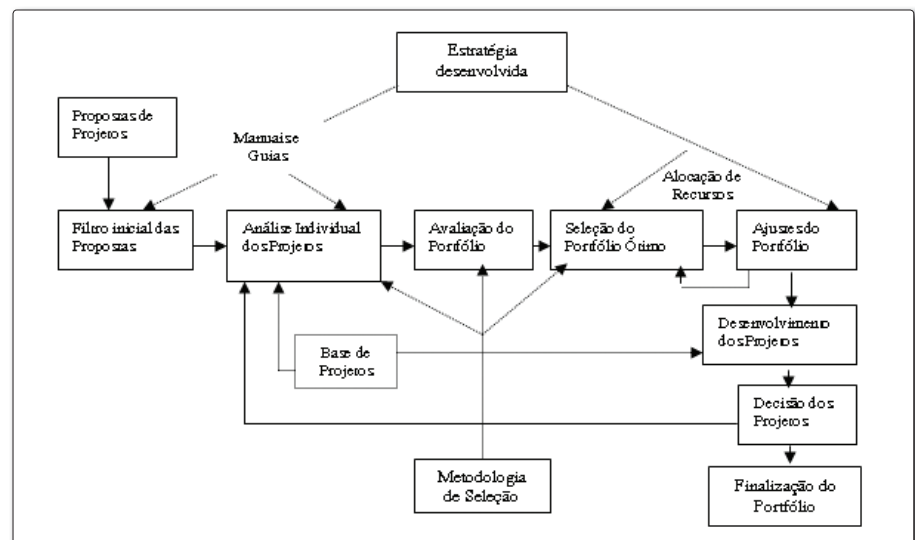


Figura 3. Gerência de Portfólio de Projetos – Elementos do Modelo de Archer

Gerência de Riscos PMBOK	GERÊNCIA DE PORTFÓLIO	
	COOPER	ARCHER
Planejar a Gerência de Risco	Faz menção a importância de definição da estratégia organizacional	Menciona uma definição clara da estratégia organizacional para os gestores
Identificar Riscos	Métodos de pontuação; Sessões de Revisão de Projetos	Métodos de pontuação – Visão de Benefícios
Analisar Riscos Quantitativamente	Processo Estágio/Passagem	Pré-seleção e Análise individual de Projetos (repositório de projetos) e Priorização; Ajustes do Portfólio
Analisar Riscos Qualitativamente	Processo Estágio/Passagem	Pré-seleção e Análise individual dos projetos / Interdependência de recursos e limitações
Controlar Respostas aos Riscos	Não existe atividade relacionada, mas acredita-se que pode ser realizada através dos Métodos de pontuação e das Sessões de Revisão de Projetos	Não existe atividade relacionada, mas podem ser controladas através das Avaliações de Fases / Ajustes do Portfólio
Controlar e Monitorar Riscos	Sessões de Revisão de Projetos	Avaliação de Fases / Ajustes do Portfólio

Tabela 1. Estudo analítico – Modelos de Portfólio de Projetos



**Planejar a Gerência de Riscos** é uma atividade implícita nos dois modelos aqui apresentados, uma vez que ambos mencionam a necessidade de uma estratégia organizacional que será base para seleção dos projetos que formarão o portfólio. É desenvolvida a estratégia para a gestão de portfólio e, a partir desta definição, são criados os tutoriais e guias para alocação dos recursos.

A atividade de **Identificar Riscos** está relacionada aos modelos de pontuação utilizados para avaliar os níveis de riscos de cada projeto em análise. No modelo de Cooper, revisões do portfólio são realizadas periodicamente para verificar a concordância dos projetos selecionados com a alocação dos recursos e limitações organizacionais impostas. Todos os projetos ativos são revisados e comparados uns aos outros e o balanceamento do portfólio é considerado sobre várias dimensões. Já na abordagem de Archer os modelos de pontuação também analisam os projetos de acordo com os aspectos positivos que podem trazer benefícios para a organização.

**Analisar Riscos Quantitativamente** é uma atividade que no modelo de Cooper está representada pelo processo Estágio/Passagem (Stage/gate) usado pelas organizações para tomar decisões do tipo Continuar/Finalizar (Go/Kill), sobre projetos individuais. Decisões de Continuar/Finalizar projetos, nas passagens e priorização de projetos, são baseadas na

pontuação final de um projeto. No modelo de Archer esta atividade está representada pela filtragem de projetos e aferição de dados através de um repositório de projetos anteriores.

A atividade de **Analisar Riscos Qualitativamente** no modelo de Cooper é realizada conjuntamente com a análise quantitativa dos dados dos projetos em avaliação. A exceção está no modelo de Archer que utiliza para a seleção de projetos várias formas de interações. Estas interações incluem interdependências, competição por recursos e sincronização, com o valor de cada projeto determinado a partir de um conjunto de parâmetros comuns que foram estimados para cada projeto no estágio anterior.

As atividades de sessões de revisões de projetos e os consequentes ajustes de processo subsidiam as atividades de **Controlar Respostas aos Riscos**.

**Controlar e Monitorar Riscos** são atividades cíclicas tanto para Cooper como Archer. Na abordagem de Archer o estágio final do processo é realizado pelos ajustes do portfólio, provendo uma visão global dos projetos selecionados. Diferentes representações visuais podem ser usadas para ilustrar o portfólio em diferentes dimensões. O objetivo do ajuste do portfólio é alcançar o equilíbrio entre os projetos em discussão. Tomadores de decisões devem ser capazes de efetuar mudanças no portfólio também nesse estágio, se as mudanças diferem substancialmente do

portfólio sugerido, então se faz necessário um novo cálculo dos parâmetros do portfólio como, por exemplo, dependência de disponibilidade de recursos.

### **Algumas Limitações**

Uma limitação atual da Gerência de Riscos é a quantidade pequena de estudos práticos divulgados, apresentando indicadores de sucesso dos projetos de software desenvolvidos [Leopoldino 2004] e [Coelho 2004].

As abordagens de Gerência de Riscos, por vezes, endereçam um número limite de objetivos, muitas vezes estas limitações têm uma pequena relevância na prática e não necessariamente indicam que o processo não funciona.

De uma forma geral, estas limitações dizem respeito a:

#### **1. Cronogramas (prazos), custos e qualidade do produto.**

Existem projetos onde outras variáveis como: impacto mercadológico do negócio, habilidade de reuso dos resultados de projetos anteriores, restrições e escassez associada aos recursos do projeto, necessidade de ser compatível com outros sistemas e garantia da conformidade dos requisitos (gerenciamento de mudanças) com o conjunto de limitações (restrições), afetam o sucesso do projeto.

#### **2. Poucas são as abordagens que reconhecem explicitamente as necessidades e expectativas dos stakeholder's envolvidos no projeto e seus objetivos.**



Muitas vezes a equipe e o cliente de um projeto têm um envolvimento grande na identificação e avaliação dos riscos do projeto mas, mesmo o envolvimento das partes não garante que os interesses estejam presentes na fase de análise de riscos. É fundamental a promoção de discussões e comunicação aberta entre os membros do projeto. Além de mecanismos que possam tratar as informações geradas.

### 3. Muitas organizações adotam como atividade de identificação de riscos os *checklists* ou taxonomia de risco.

O uso de listas de verificação (*checklists*) e de taxonomias pode ser bastante útil na identificação de riscos, previamente relacionados. Mas podem aumentar a tendência dos participantes e equipe do projeto a limitarem-se a lista de riscos definida, tolhendo a habilidade de ficar atento a novos riscos. Outro fator importante que deve ser levado em consideração é que, principalmente as taxonomias, têm seus riscos identificados dependentemente do ambiente, além de ser extremamente fatigante manter um nível de detalhamento dos riscos listados. Em 1997, Tony Moynihan [Moynihan 1997] em seu relatório intitulado *How experienced Project Managers Assess Risk*, fez uma série de estudos com um grupo de 14 gerentes de projetos, baseado na taxonomia do SEI, onde foi apresentado, de acordo com os resultados obtidos, que as principais preocupações no uso de taxonomias devem ser a evolução das necessidades específicas de cada projeto e o escopo da lista de riscos, criada com base nos tipos e categorias de riscos utilizados.

### 4. A quantificação e a priorização dos riscos é um grande desafio para a Gerência de Riscos.

Normalmente é baseada em estimativas, probabilidades e perdas. As estimativas podem ser baseadas em três tipos de abordagens: dados históricos, estimativas subjetivas e tabelas. Estes tipos de abordagens também apresentam algumas limitações.

A utilização de dados históricos raramente é utilizada na estimativa da probabilidade de riscos, presumivelmente porque as organizações não coletam os dados dos projetos com a frequência necessária.

A subjetividade está associada ao tipo de experiência e crenças. Logo, as estimativas subjetivas refletem muito o quanti-

tativo de conhecimento e experiência da pessoa em questão. Os estudos de avaliação subjetiva dos riscos vêm crescendo e ganhando adeptos rapidamente. A razão para esta mudança é a rapidez com que se podem avaliar os riscos através de uma análise qualitativa baseada em uma visão quantitativa, provavelmente um dos mecanismos mais realistas de estimar a probabilidade de eventos futuros.

### A Busca pelo Conhecimento através da Experiência

Na busca constante pela melhoria de processos, a Engenharia de Software, apoiada pela Inteligência Artificial, desenvolve diversas aplicações atualmente que suportam a aquisição, organização, reuso e compartilhamento de conhecimento sobre processo de software. O emprego de ontologias, por exemplo, favorece o compartilhamento e o reuso de bases de conhecimento, o seu uso como guia para o processo de aquisição de conhecimento e uma mais fácil compreensão e interação entre pessoas.

De uma forma geral, gerentes de projetos têm que alocar, ratear recursos entre projetos, gerenciá-los dentro do orçamento e tempo disponíveis e, por fim, garantir que estes recursos limitados sejam implementados de acordo com o planejado.

Para o alcance desse objetivo uma grande quantidade de variáveis pode ser considerada tanto ameaças quanto oportunidades. Para uma melhor identificação desses fatores críticos de sucesso dos projetos é importante a definição de uma terminologia única para a captura, avaliação e controle dos riscos.

A *mPRIME Ontology*, projeto desenvolvido no Centro de Informática (CIn - UFPE), é uma ontologia que permite

a identificação de riscos através do uso de um vocabulário comum que pode ser utilizado para representar conhecimento útil dentro de um ambiente de desenvolvimento de software [Gusmão 2007].

Ontologias são úteis para apoiar a especificação e implementação de qualquer sistema de computação. Uma ontologia pode ser desenvolvida por motivos diversos mas, de uma forma geral, traz os seguintes benefícios (de certa área de conhecimento):

**Melhor compreensão:** no desenvolvimento de uma ontologia, as pessoas envolvidas no processo se vêem diante de um desafio: explicar seu entendimento sobre o domínio em questão, o que as faz refletir e melhorar sua compreensão sobre esse domínio;

**Entendimento e consenso:** geralmente, para uma determinada área de conhecimento, especialistas têm entendimento distinto sobre os conceitos envolvidos, podendo levar a problemas de comunicação. Na construção de uma ontologia, essas possíveis diferenças são explicitadas e busca-se um consenso sobre seu significado e importância;

**Aprendizagem:** na existência de uma ontologia sobre uma determinada área de conhecimento desenvolvida, uma pessoa que deseje aprender mais sobre essa área não precisa se reportar sempre a um especialista. Ela pode estudar a ontologia e aprender sobre o domínio em questão, absorvendo um conhecimento geral e de consenso.

Alguns trabalhos podem ser encontrados na área de Engenharia de Software ratificando a importância da utilização de ontologias como instrumento para a modelagem de domínios [Duarte e Falbo 2000, Siebra et al 2004, Medeiros 2004]. A utilização de ontologias, no entanto, requer a criação de uma cultura de uso e



a existência de ferramentas e metodologias adequadas para suporte efetivo das informações manipuladas e do conhecimento gerado.

Outra técnica bastante interessante é o Raciocínio baseado em Casos (RCB) que busca uma solução para um problema novo baseando-se em experiências passadas.

No contexto da Gerência de Projetos, as informações mais relevantes no início de um novo projeto são as informações sobre projetos passados. Esse conhecimento passado pode ser utilizado tanto nas estimativas de custos e prazos, quanto em outros processos como o processo de identificação de riscos, uma forma de aprender com as experiências passadas. A maioria das técnicas de identificação de riscos faz uso da experiência passada das pessoas envolvidas no processo de identificação.

Projeto deste cunho vem sendo desenvolvido no Departamento de Sistemas e Computação (DSC – UPE) através da pesquisa de modelos que suportem a busca por experiências passadas – CBR Risk Method (na Web: [pma.dsc.upe.br](http://pma.dsc.upe.br)).

O CBR Risk é um método que objetiva auxiliar gerentes de projetos nas suas atividades de Gerência de Riscos, através de suporte à identificação de riscos para novos projetos baseado em analogia. Esse

método já está disponibilizado como funcionalidade da ferramenta de Gestão de Riscos *mPRIME Tool* (na Web - versão demo: [www.cin.ufpe.br/~suppera](http://www.cin.ufpe.br/~suppera)).

### Considerações Finais

O gerenciamento de riscos é uma ciência que permite ao homem uma convivência mais segura com os riscos a que estão expostos. Tem a função de proteger os seres humanos, seus recursos materiais e o meio ambiente. Em uma organização, um programa de gerenciamento de riscos tem o objetivo de identificar, analisar e avaliar os riscos existentes e assim decidir como serão tratados. De uma forma geral, existem duas formas de tratar o risco: financiando ou controlando.

Apesar das dificuldades de se precisar a avaliação de riscos, o objetivo principal é indicar o conjunto de prioridades no plano de ação, para que se possam tomar as medidas de prevenção de maneira correta e sistemática e, assim, aperfeiçoar os resultados do próprio desenvolvimento tecnológico, a partir da redução dos riscos apresentados pelas atividades surgidas na sociedade moderna.

A relação custo-benefício para a implementação de um plano de gerenciamento de riscos em uma organização poderá

ser favorável, pois os custos envolvidos não podem ser comparados com os benefícios que acompanham uma maior proteção dos recursos humanos, materiais, financeiros e ambientais.

A conscientização mundial de elaborar e adotar políticas de desenvolvimento sustentável, ou seja, políticas que conciliem desenvolvimento econômico com eliminação de desperdícios, respeito ao ser humano e ao meio ambiente, revela o quanto é fundamental a adoção de tecnologias como o gerenciamento de riscos pelas organizações brasileiras. O Brasil gastou em 2005 menos que 0,92% de seu Produto Interno Bruto (US\$ 14,6 bilhões) com produtos e serviços de TI. É necessário investir em prevenção para que haja uma redução dos erros e falhas (riscos) antes que os mesmos ocorram, tornando as organizações mais competitivas. ●

#### Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



### Referências

- [Archer e Ghasemzadeh 1998] Archer, N., Ghasemzadeh, F. A Decision Support System for Project Portfolio Selection. *International Journal of Technology Management*, Vol. 16, No. 1-3, pp.105-114, 1998.
- [Coelho 2004] Coelho, P. Identificação das Estratégias de Aprendizado utilizadas pelos PMP's e Aspirantes a Certificação PMP. Projeto PMK – Environment Learning. CIn /UFPE – Centro de Informática – Universidade Federal de Pernambuco. 2004.
- [Cooper et al 2001] Cooper, R. G.; Edgett, S.; Kleinschmidt, E. J. (2001) *Portfolio Management for New Products*. 2ª ed. Perseus Publishing, New York.
- [Correia 2005] Correia, B. C. S. (2005) *Portfólios: Um Modelo de Gestão de Portfólio de Projetos de Software*. Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife.
- [Danilovic e Borjesson 2001] Danilovic, M. e Borjesson, H. (2001) *Managing the MultiProject Environment*. In: *The Third Dependence Structure Matrix (DSM) International Workshop, Proceedings*, Massachusetts Institute of Technology (MIT), Massachusetts, Boston, Cambridge, USA.
- [Duarte e Falbo 2000] Duarte, K. C. e Falbo, R. A. Uma Ontologia de Qualidade de Software. *Anais do VII Workshop de Qualidade de Software, XIV Simpósio Brasileiro de Engenharia de Software*, João Pessoa, Paraíba, Brasil, Outubro 2000.
- [Dye e Pennypacker 2000] Dye, L.; Pennypacker, J. (2000) *Project Portfolio Management and Managing Multiple Projects: Two Sides of the Same Coin?*. In: *Proceedings of the Project Management Institute Annual Seminars & Symposium*, Houston, Texas, USA.
- [Freitas 2005] Freitas, B. C. C. (2005) *Um Modelo para o Gerenciamento de Múltiplos Projetos de Software aderente ao CMMI*. Monografia (Trabalho de Graduação) – Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife.
- [Gusmão 2007] Gusmão, C. M. G. (2007) *Um Modelo de Processo de Gestão de Riscos para Ambientes de Múltiplos Projetos de Desenvolvimento de Software*. Tese de Doutorado. Universidade Federal de Pernambuco – Recife/PE – Brasil.
- [Hall 1998] Hall, E. M. (1998) *Managing Risk – Methods for Software Systems Development*. Addison-Wesley. pp 88-103.
- [Leopoldino 2004] Leopoldino, C. B. *Avaliação de Riscos em Desenvolvimento de Software*. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul – Escola de Administração. Porto Alegre. 2004.
- [Moura et al 2004] Moura, H. P.; Gusmão, C. M. G.; Correia, B. C. S. (2004) *Portfolio Management: A Critical View of Risk Factors Balancing*. *Anais do NORDNET – International PM Conference*. Helsinki – Finlândia.
- [Moynihan 1997] Moynihan, T. (1997) *How experienced Project Managers Access Risk*. *IEEE Software*. Volume 14. Nº 3. 35-41.
- [Pereira 2002] Pereira, A. R. (2002) *Modelo de Gestão de Portfólio para Alinhar os Projetos de Novos Produtos às Estratégias Corporativas*. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.
- [PMI 2004] PMI - Project Management Institute. (2004) *A Guide to the Project Management Body of Knowledge – ANSI/PMI 99-01-2004*. Project Management Institute. Four Campus Boulevard. Newtown Square. USA.
- [Siebra et al. 2004] Siebra, S. A. et al. *SmartChat - An Intelligent Environment for Collaborative Discussions*. *ITS - Intelligent Tutoring Systems 2004*: pp 883-885. 2004.



# Out Sourcing Total

Preparados para nova era em qualidade de serviços, garantimos operações full-time dentro de uma gestão única desenvolvida pela nossa equipe para atuar preventivamente, somos a única empresa a operar por 25hs por dia garantindo disponibilidade total.

## Escalabilidade, Conectividade e Disponibilidade.

### CERTIFICAÇÕES

- ↳ Servidor Dedicado
- ↳ Gerenciamento
- ↳ Banda IP
- ↳ Cross Connection
- ↳ Link PAP
- ↳ Serviços compartilhados
- ↳ Sites
- ↳ E-mail
- ↳ Backup

MICROSOFT

LINUX

CISCO

ITIL

Data Center Próprio

## Segurança e certeza de bons negócios e serviços

[contato@localdatacenter.com.br](mailto:contato@localdatacenter.com.br)

+55(21) 3527-6510



# Desenvolvimento de Software Dirigido por Caso de Uso

## Parte III: Caso de Uso de Negócio

**N**os artigos anteriores apresentamos como trabalhamos com os casos de uso e seu papel no desenvolvimento de software. Foram apresentados os erros cometidos na criação e na especificação de casos de uso e também como evitá-los, assim como os conceitos, as características e sua importância no desenvolvimento de software de uma forma geral. Mostramos também como especificar um caso de uso utilizando boas práticas. Foi mostrado que juntamente com o caso de uso existe um documento chamado de *Especificação Suplementar*, que é um complemento para os casos de uso sendo preenchido com as regras não funcionais, pois casos de uso possuem apenas regras funcionais. Foi possível entender que com o caso de uso é possível fazer uma análise dos requisitos e um projeto físico de forma mais eficaz e termos um projeto centrado em arquitetura que formará a base de todo o desenvolvimento do software. Os ca-

### **De que se trata o artigo:**

Uso do caso de uso de negócio no mapeamento do processo de negócio do cliente e objetivos do negócio. Neste artigo serão mostrados detalhes sobre o modelo de caso de uso de negócio e sua especificação.

### **Para que serve:**

Fornecer um meio de mapear o processo de negócio do cliente de forma mais clara e, com isso, termos um melhor entendimento do negócio para assim podermos dar uma solução sistêmica mais efetiva.

### **Em que situação o tema é útil:**

Muito útil para o levantamento e entendimento do processo de negócio do cliente, tanto para validação do cliente quanto para a equipe de desenvolvimento.

Os casos de uso dos artigos anteriores são chamados de casos de uso sistêmicos, pois eles têm por objetivo mapear a solução do software (ou a solução sistêmica) em termos de regras, passos entre os atores (usuários) e o sistema (GUI), mensagens do sistema para os atores e exceções das regras.



### **Vinicius Lourenço de Sousa**

*vinicius.lourenco.sousa@gmail.com*

Atua no ramo de desenvolvimento de software há mais de 10 anos, é autor de diversos artigos publicados pelas revistas ClubeDelphi e SQL Magazine. É Graduado em Tecnologia da Informação pela ABEU Faculdades Integradas e Pós-Graduado em Análise, Projeto e Gerência de Sistemas pela PUC-RJ, IBM Certified: Especialista Rational Unified Process e instrutor de UML, Análise OO e Java. Atualmente trabalha na CPM Braxis como Especialista nas áreas de arquitetura, especificação de requisitos, levantamento e modelagem de processo de negócio e projetista de software em soluções com componentes de negócio, SOA e BPM.

Neste artigo veremos a parte final do desenvolvimento de software dirigido por caso de uso, mostrando outro tipo de caso de uso, chamado de caso de uso de negócio. A diferença entre o caso de uso sistêmico e o caso de uso de negócio está no objetivo de sua utilização, ou seja, na visão que cada um representa. Enquanto o primeiro serve para mapear a solução sistêmica, o segundo serve para mapear o processo de negócio do cliente, independente de ser automatizado (envolvendo sistemas legados) ou manual. Na próxima seção vamos entender mais detalhadamente sobre o caso de uso de negócio e seu modelo.

## Entendendo o que é um Modelo de Caso de Uso de Negócio

O modelo de caso de uso de negócio descreve a direção e a intenção do negócio. Direção é provida na forma dos objetivos do negócio que são chamados de *goals*.

O modelo de caso de uso de negócio é usado pelos usuários-chaves do negócio, analistas do processo de negócio e os projetistas do negócio para entender e melhorar o caminho que o negócio interage com seu ambiente, e pelos analistas de sistema e arquitetos de software para prover o contexto para o desenvolvimento de software. O gerente de projeto também usa o modelo de caso de uso de negócio para planejar o conteúdo das iterações durante a modelagem de negócio pela equipe de desenvolvimento e o rastreamento do progresso pela equipe.

Como explicado antes, o caso de uso de negócio serve para mapear o processo de negócio da empresa do cliente. Entende-se por processo de negócio, de uma forma resumida, toda atividade que deve ser feita, como deve ser feita, quando deve ser feita, por quem deve ser feita e o que deve ser gerado ao final das atividades. Neste artigo não entrarei em detalhes sobre processo de negócio, pois

é muito extenso e é necessário um ou vários artigos para uma explicação mais abrangente. Abaixo veremos algumas regras para entendermos como funciona um caso de uso de negócio.

**Um caso de uso de negócio é uma sequência de interações entre o ator de negócio (alguém ou algo que interage com o processo de negócio) e o processo de negócio, que acontece de forma atômica, na perspectiva do ator de negócio.**

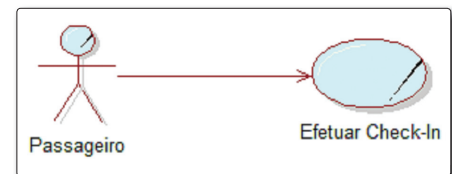
Perceba que esta regra é bem parecida com a regra do caso de uso sistêmico, mas ao invés de termos a interação entre o ator e o sistema, temos a interação entre o ator de negócio e o processo de negócio. O ator de negócio é aquele que não participa do processo de negócio, sendo o estímulo para o processo ser iniciado, ou seja, ele é o cliente do negócio. Como exemplo veja a **Figura 1**.

Na **Figura 1** vemos um diagrama de caso de uso de negócio para a realização de check-in no aeroporto. Como podemos perceber de início, já existe uma diferença visual entre o diagrama de caso de uso sistêmico e o diagrama de caso de uso de negócio. Essa diferença visual é porque os elementos visuais da UML que são o ator e o caso de uso utilizam estereótipos diferentes. Os estereótipos são utilizados para classificar os elementos da UML, além de introduzir novos elementos visuais e conceitos à linguagem de modelagem. Um estereótipo permite a criação de novos tipos de blocos de construção que são derivados dos já existentes, estendendo a semântica dos elementos e não a estrutura das classes pré-existentes no metamodelo da UML. Um estereótipo é considerado como um metamodelo, pois cada um cria o equivalente a uma nova classe no metamodelo da UML.

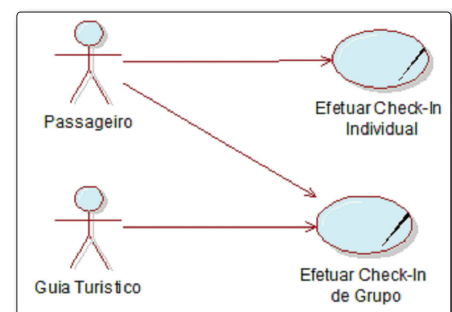
No caso da **Figura 1** foi indicado que os elementos teriam o estereótipo de caso de uso de negócio e de ator de negócio. O caso de uso de negócio *Efetuar Check-In* mapeia todo o processo de negócio para a realiza-

ção de check-in e o ator de negócio representa os passageiros que irão utilizar desse “serviço de negócio”. O processo de check-in possui várias etapas ou passos para sua realização como a identificação do passageiro, a confirmação do voo (data, horário, portão, etc.), despacho de bagagem e etc. No final do processo o passageiro receberá a confirmação da realização de seu check-in sem ter o conhecimento do processo como um todo. É por isso que o ator de negócio não participa do processo de negócio, pois serão outras pessoas (funcionários da empresa) que irão trabalhar em conjunto para o processo ser executado. O ator de negócio depois de ter iniciado o processo, apenas aguardará o resultado final. Repare que essas etapas do processo de negócio estando automatizadas se tornam as funcionalidades do sistema ou passos de uma funcionalidade, ou seja, seriam mapeadas por um ou mais casos de uso sistêmicos. Então podemos dizer que um caso de uso de negócio pode ser “realizado” por um ou vários casos de uso sistêmicos.

**Ao ser executado, um caso de uso de negócio deve fornecer um resultado observável e significativo para o ator de negócio.**



**Figura 1.** Realização de Check-In no aeroporto



**Figura 2.** Realização de Check-In individual e de grupo



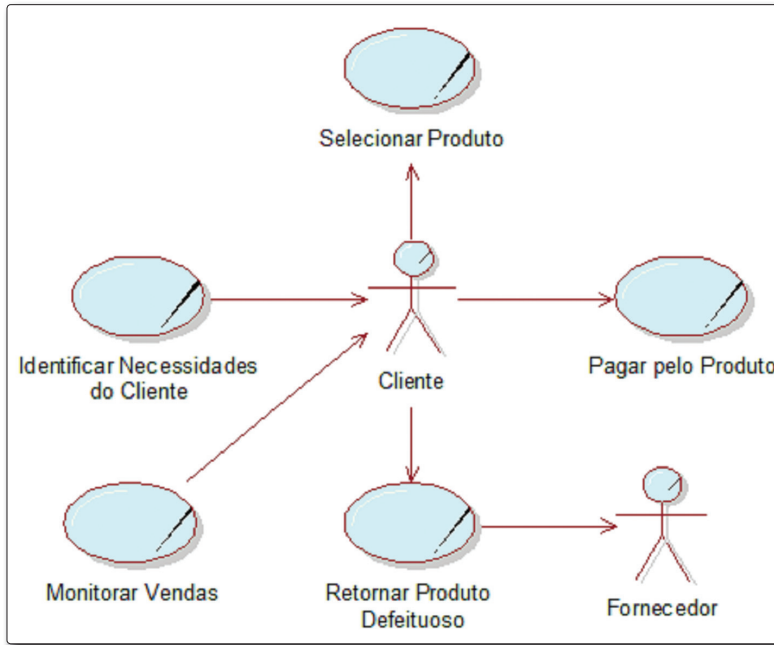


Figura 3. Diagrama de caso de uso de negócio de uma loja de produtos virtual

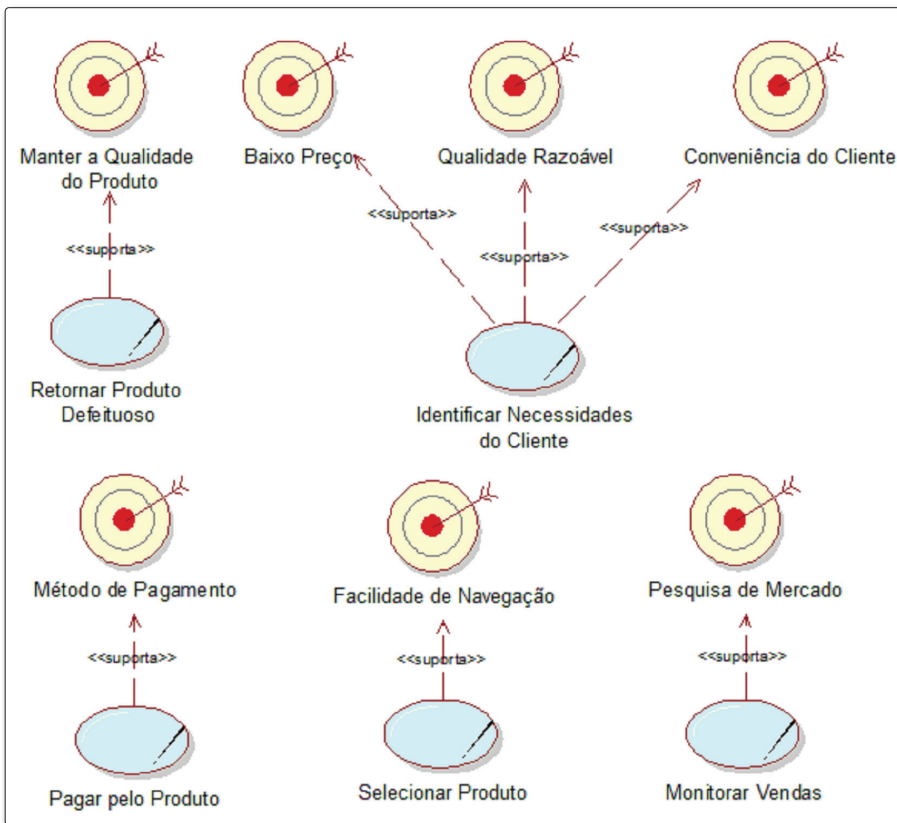
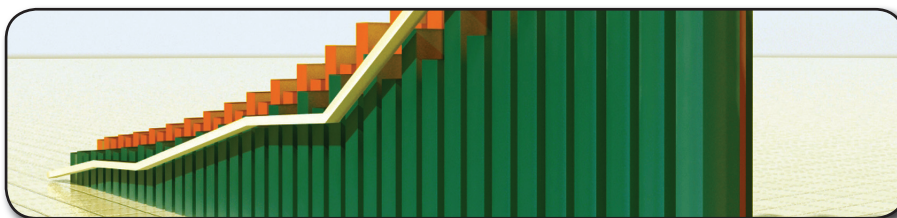


Figura 4. Diagrama de link entre os casos de uso de negócio e os goals



Todo caso de uso de negócio deve representar um processo de negócio de algum “serviço” do cliente.

Como exemplo, posso citar o próprio caso de uso de negócio da Figura 1. O passageiro ao solicitar o seu check-in terá como resultado a confirmação da realização ou não do check-in. Portanto, igualmente como em um caso de uso sistêmico, o resultado do caso de uso de negócio também está dentro do contexto do processo de negócio, pois o que se deseja é a realização do check-in.

### Características Importantes de um Diagrama de Caso de Uso de Negócio

Ao criarmos um diagrama de caso de uso de negócio, devemos nos atentar para certas características para que o diagrama não fique errado ou confuso no momento da especificação. Vou me ater somente às características diferentes das do diagrama de caso de uso sistêmico, pois tanto o sistêmico quanto o de negócio possuem características semelhantes. Veja as características de um diagrama de caso de uso de negócio:

- **Ator de Negócio:** Alguém ou algo que interage com o processo de negócio. Um ator de negócio pode ser um sistema, um cliente, um fornecedor, um parceiro de negócio ou potenciais clientes.

O nome do ator de negócio deve refletir seu papel para o negócio. Para atores que representam pessoas, não podemos nomeá-los com os nomes das pessoas. Essa regra é a mesma que é utilizada para os atores sistêmicos. Por exemplo: na Figura 2, temos os atores de negócio *Passageiro* e *Guia Turístico*. O primeiro pode efetuar um check-in individual ou um check-in de grupo e o segundo somente poderá efetuar check-in de grupo. O nome *Passageiro* e *Guia Turístico* refletem o papel nesse contexto do negócio;

- **Caso de Uso de Negócio:** Como já explicado, um caso de uso de negócio mapeia a interação entre o ator de negócio e o processo de negócio. Nesta seção apenas incrementarei as características de caso de uso de negócio dentro do diagrama de caso de uso de negócio.

Assim como os casos de uso sistêmicos, um dos grandes problemas em diagramas de caso de uso de negócio é a sua nomeação. Um caso de uso de negócio deve ter um nome que esteja dentro do contexto

do negócio do cliente. A forma do nome deve ser ativa, normalmente descrito pelo gerúndio do verbo ou um verbo e um substantivo conjunto. Os nomes podem descrever as tarefas no caso de uso de negócio a partir de um ponto de vista externo ou interno. Embora o caso de uso de negócio descreva o que acontece dentro do negócio, muitas vezes é mais natural nomear o caso de uso de negócio a partir do ponto de vista do ator de negócio principal. Como exemplo, podemos ver a **Figura 2** que os nomes dos casos de uso de negócio estão voltados para o ponto de vista dos atores de negócio.

- **Goals:** Os goals (objetivos do negócio) são fatores determinantes e de extrema importância na identificação dos casos de uso de negócio. Durante o levantamento do negócio é muito importante que os analistas do processo de negócio entendam e mapeiem todos os goals relacionados ao negócio. Os goals dos casos de uso de negócio devem ser especificados a partir de duas perspectivas:

- Para os atores de negócio o processo de negócio interage com eles e especifica o valor que o ator de negócio espera obter a partir do negócio (goals externo).

- Do ponto de vista do desempenho do processo de negócio da organização, definir quais os objetivos dos processos de negócio e aquilo que se espera atingir através de sua realização (goals interno).

É muito importante que os casos de uso de negócio sejam “linkados” aos seus respectivos goals em um diagrama a parte, pois assim poderemos ter o mapeamento de quais casos de uso de negócio dão suporte a um ou vários goals. Veja na **Figura 3** e **Figura 4** um exemplo desse link.

Na **Figura 3** vemos um diagrama de caso de uso de negócio de uma loja de produtos na internet. Neste diagrama existem dois atores de negócio, onde o primeiro que é o *Cliente* pode selecionar produtos, pagar pelos produtos e devolver os produtos defeituosos. O segundo ator de negócio é o *Fornecedor*, que recebe os produtos defeituosos para efetuar sua manutenção. Perceba que os casos de uso de negócio *Identificar Necessidades do Cliente* e *Monitorar Vendas* não têm nenhum ator de negócio iniciando seu processo e sim apenas notificando o cliente. Isso é porque esses dois ca-

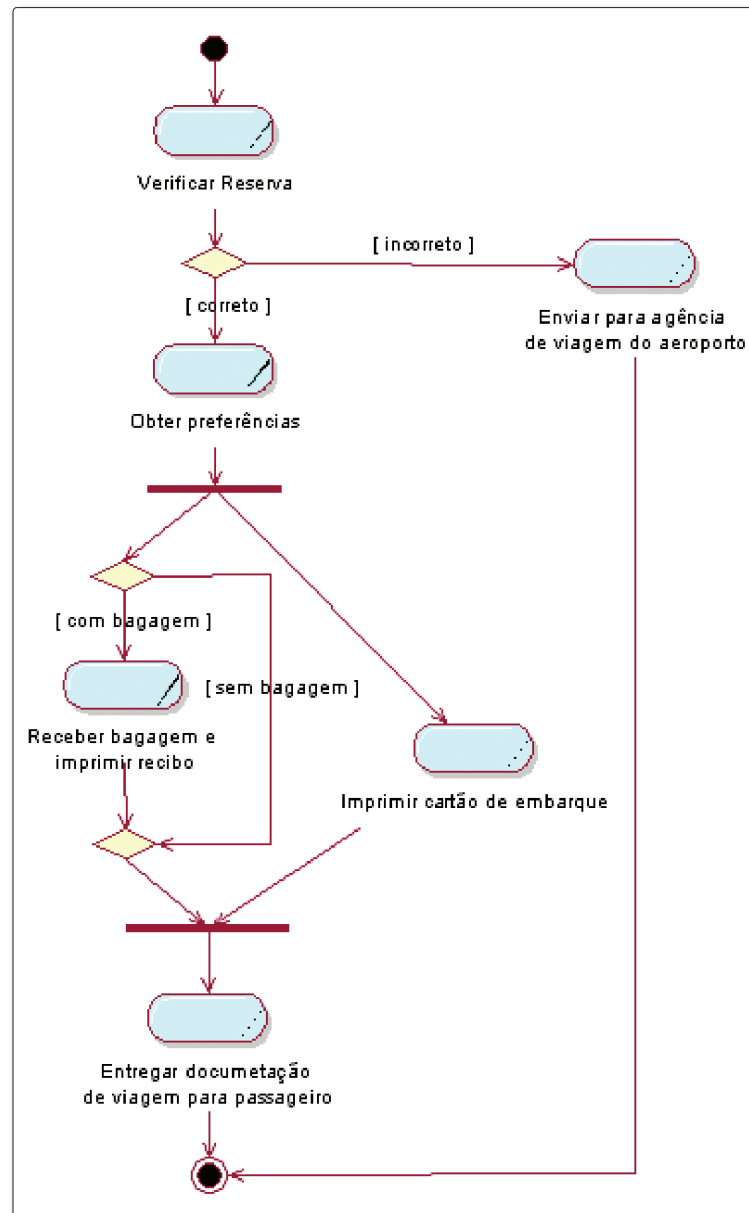
sos de uso de negócio são iniciados internamente, ou seja, quem inicia esses processos faz parte do próprio processo de negócio e como explicado antes os atores de negócio interagem com o processo sendo seus clientes, não fazendo parte dele. Esses casos de uso de negócio são geralmente chamados de casos de uso de negócio administrativos.

Já na **Figura 4**, vemos um diagrama de caso de uso utilizado para fazer o link entre os casos de uso de negócio da **Figura 3** e os goals do negócio. Veja que o link entre o caso de uso de negócio e o goal é feito através de dependência usando o estereótipo <<suporta>>, para indicar que o caso de uso em questão suporta aque-

le objetivo do negócio. Vemos que para cada goal existe um caso de uso de negócio, mas poderiam existir vários casos de uso de negócio dando suporte a um goal e vice-versa, como o caso de uso *Identificar Necessidades do Cliente*, que dá suporte aos goals *Baixo Preço*, *Qualidade Razoável* e *Conveniência do Cliente*.

## Especificação de Caso de Uso de Negócio

Na edição anterior apresentei três formas de especificação de caso de uso e expliquei que a forma mais usada para casos de uso de negócio é a **Descrição Contínua**. Existe outra forma de criarmos uma especificação para o caso de



**Figura 5.** Diagrama de atividade de negócio do processo de check-in

uso de negócio que é a utilização do diagrama de atividade de negócio da UML. A vantagem da utilização desse diagrama é o rápido entendimento do processo de negócio que o caso de uso de negócio representa pelos envolvidos, sejam eles o cliente ou a equipe de desenvolvimento. Ao invés de terem que ficar lendo páginas do processo descrito, eles poderão validar de forma mais rápida todos os cenários, condições de negócios e as atividades. Veja na **Figura 5** um exemplo de diagrama de atividade de negócio.

Observe que na **Figura 5** o diagrama de atividade é bastante semelhante ao diagrama de atividade do caso de uso sistêmico mostrado no artigo anterior. A única diferença é no visual, pois esse diagrama de atividade está usando o estereótipo para o diagrama de atividades de negócio. Na figura podemos entender claramente como funciona o processo de negócio da realização de check-in. Estão detalhadas todas as atividades, os seus cenários e condições de negócio que são representados pelos símbolos de decisão.

Um caso de uso de negócio não precisa obrigatoriamente ser especificado usando o diagrama de atividade da UML quando desejarmos fazer uma especificação visual. Hoje em dia, muitas equipes de desenvolvimento utilizam o BPMN (**Business Process Management Notation / Notação de Gerenciamento do Processo de Negócio**) que é uma notação concorrente a UML para mapeamento de processo de negócio bem parecida com

o diagrama de atividade. Sendo que esse diagrama pode ser executado através de ferramentas próprias e assim sabermos se o processo mapeado está aderente ao negócio do cliente ou se é necessário fazer alterações no processo através de métricas de negócio obtidas. Este artigo não visa explicar sobre o BPMN, pois isso é bastante amplo e deve ficar para um artigo próprio. Apenas quis mencionar que hoje em dia a UML não é a única a ser utilizada quando o assunto é modelagem de processo de negócio.

### Link entre Caso de Uso de Negócio e Caso de Uso Sistêmico

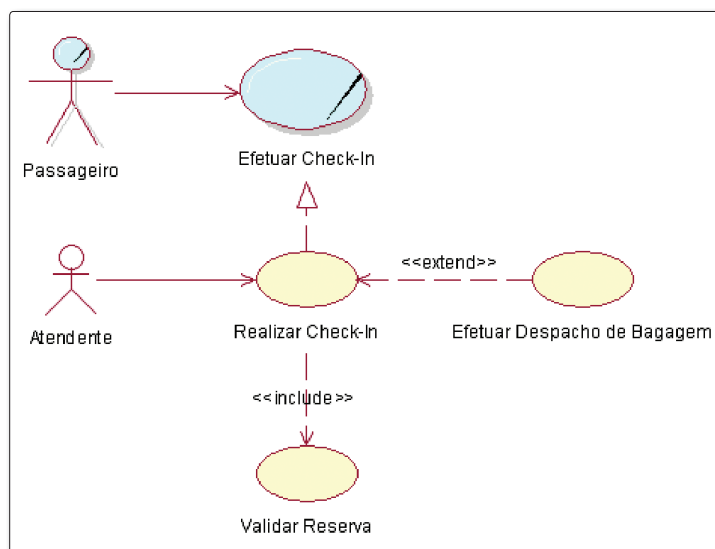
Da mesma forma que é feito um link entre os casos de uso de negócio e os goals, também é uma boa prática criarmos um link entre os casos de uso de negócio e os casos de uso sistêmicos. Com isso proporcionarmos uma visão mais detalhada das partes do processo de negócio que estão sendo automatizadas. Com base na **Figura 5** veja a **Figura 6**.

Observe no diagrama da **Figura 6** que foi feito o link entre o caso de uso de negócio *Efetuar Check-In* e o caso de uso sistêmico *Realizar Check-In*. Com esse link é possível termos um mapa de rastreabilidade com todos os casos de uso sistêmicos que estão automatizando o processo de negócio ou parte dele. Comparando com o diagrama de atividade de negócio da **Figura 5**, vemos que as atividades *Verificar Reserva*, *Obter Preferências*, *Receber Bagagem e Imprimir Recibo* e *Imprimir Cartão de Embarque* foram automatizadas.

Também percebemos que existem mais dois casos de uso sistêmicos no diagrama que são o *Efetuar Despacho de Bagagem* e *Validar Reserva*. O primeiro caso de uso é estendido porque pelo processo de negócio a bagagem pode ser despachada ou não e o segundo caso de uso é incluído porque a reserva sempre será validada. A separação dessa parte do processo de negócio em dois casos de uso sistêmicos está levando em consideração as regras e boas práticas apresentadas na primeira parte desse artigo, ou seja, o caso de uso *Efetuar Despacho de Bagagem* tem suas regras e passos e o caso de uso *Validar Reserva* pode ser utilizado por outros casos de uso como, por exemplo, um caso de uso que cria a reserva.

### Conclusão

Nesta terceira e última parte vimos como criar e especificar um caso de uso de negócio e para que ele serve. Vimos também a importância de criarmos um link entre os casos de uso de negócio e os goals que são “realizados” por esses casos de uso e as principais características do modelo de caso de uso de negócio, além da especificação visual onde podemos entender o processo de negócio de uma forma mais rápida. Espero que essa série de artigos do desenvolvimento de software dirigido por caso de uso tenha sido de grande ajuda em tirar dúvidas e na obtenção de novos conhecimentos. ●



**Figura 6.** Diagrama de link entre os casos de uso de negócio e os casos de uso sistêmicos

#### Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



#### Links

**OMG: The Object Management Group**  
[www.omg.org](http://www.omg.org)

**RUP: Rational Unified Process 7.0**  
<http://www-306.ibm.com/software/awdtools/rup/>

#### Bibliografia

**UML Essencial – 3ª Edição – Martin Fowler**

**Utilizando UML e Padrões – Larman, Craig**



# 10 Motivos para hospedar seu site na RedeHost!

30 dias  
GRÁTIS  
para testar.

Mais de 15 mil  
clientes e 6  
anos no  
mercado.

Até 50Gb de  
espaço para  
e-mails ou  
site.

Planos a  
partir de  
R\$ 13,90.

Webmail em  
Ajax com  
calendário,  
tarefas,  
anotações e RSS.

ASP, PHP,  
.net e MySQL  
no mesmo  
plano.

Domínios  
adicionais  
ilimitados  
(apontam para  
o site principal).

Data center  
no Brasil com  
Links  
Redundantes.

Infra-estrutura  
com servidores  
Dell e redes  
Cisco.

Estatísticas de  
acesso com 75  
relatórios.



# Eclipse Process Framework

Um ambiente de Engenharia de Software livre para publicar e manter métodos e processos



**Gustavo Serafim**

[gustavo@soacorporativa.com.br](mailto:gustavo@soacorporativa.com.br)

[www.soacorporativa.com.br](http://www.soacorporativa.com.br)

Especialista IBM RUP. Atualmente, apóia iniciativas SOA/BPM na CPM (Adoção e Consolidação). Especialista em Engenharia de Software, Orientação a Objetos, Reutilização e SOA. Experiência como consultor/instrutor em: adaptação de processos de desenvolvimento baseado no RUP (foco na arquitetura, dirigido por UC e desenvolvimento iterativo); técnicas de design de software (incluindo as técnicas de análise e design orientados a objetos); modelagem de processos de negócios e exploração de automação do processo; desenvolvimento baseado em componentes (CBD).

## **De que se trata o artigo:**

Uma ferramenta livre, desenvolvida com o Eclipse, para a manutenção e publicação de processos e métodos. Fornece uma terminologia comum, o UMA (*Unified Method Architecture*), para definição de processos e métodos. Esta terminologia permite que métodos ou práticas sejam reaproveitados em processos diferentes, possui mecanismos de extensão para adaptar práticas em contextos diferentes.

## **Para que serve:**

O EPF tem como principal objetivo fornecer meios para manter uma base de conhecimento de capital intelectual que você possa procurar, gerenciar e criar. Este conteúdo pode ser externo e, o mais importante, você pode incluir seu próprio conteúdo como, white papers, diretrizes, modelos, boas práticas, procedimentos e regulamentações internas, material de treinamento e qualquer outra descrição geral de seus métodos. Equipes de desenvolvimento precisam ser instruídas sobre os métodos aplicáveis aos papéis que desempenham. O EPF funciona como uma ajuda online.

## **Em que situação o tema é útil:**

Equipes de desenvolvimento têm enfrentado questões como a falta de uma terminologia comum para documentar processos e métodos; a dificuldade de adaptar e estender o conhecimento para projetos diferentes; a inexistência de um ambiente central de publicação, para facilitar a disseminação da base de conhecimento, entre outras.

Mesmo em corporações que têm adotado abordagens mais ágeis de desenvolvimento, encontramos desafios em coordenar e gerenciar várias equipes simultâneas, que estão frequentemente desenvolvendo partes diferentes do mesmo sistema. Isto resulta na necessidade de uma abordagem fácil de planejar, divulgar e reaproveitar processos de governança e práticas ágeis. Essas práticas e processos de governança podem ser publicados, reutilizados e compostos com ajuda do Eclipse Process Framework (EPF), apoiando também o desenvolvimento ágil em escala corporativa.

Existem vários fatores que ditam se um processo de desenvolvimento será mais formal ou mais ágil, tais como tamanho e cultura da equipe, localização geográfica dos membros, complexidade da arquitetura, tecnologias envolvidas, padrões utilizados, entre outros. Projetos têm necessidades únicas, que podem ser suportadas em processos e métodos de desenvolvimento distintos. Há ainda as boas práticas do desenvolvimento de software, que beneficiam toda a equipe de projeto, tornando-a mais eficaz. Essas abordagens podem ser adaptadas ou estendidas para as necessidades específicas de cada projeto/equipe.

Conseqüentemente, tal singularidade dos projetos atuais provoca novos desafios. Muitas equipes de desenvolvimento têm enfrentado questões como a falta de uma terminologia comum para documentar processos e métodos; a dificuldade de adaptar e estender o conhecimento para projetos diferentes; a inexistência de um ambiente central de publicação, para facilitar a disseminação da base de conhecimento, entre outras.

Mesmo em corporações que têm adotado abordagens mais ágeis de desenvolvimento, encontramos desafios em coordenar e gerenciar várias equipes simultâneas, que estão freqüentemente desenvolvendo partes diferentes do mesmo sistema. Isto resulta na necessidade de uma abordagem fácil de planejar, divulgar e reaproveitar processos de governança e práticas ágeis. Essas práticas e processos de governança podem ser publicados, reutilizados e compostos com ajuda do Eclipse Process Framework (EPF).

Muitas organizações supõem que as pessoas sabem como realizar suas tarefas e não documentam seus métodos. Porém, cabe salientar que, para que as empresas tornem os seus sucessos repetíveis, não dependendo exclusivamente de talentos individuais, elas devem estabelecer práticas comuns.

EPF é um ambiente que permite que engenheiros de processo, engenheiros de software e desenvolvedores implementem, desenvolvam e façam a manutenção de processos para organizações ou para projetos individuais.

**Onde o EPF pode ajudar?**

Quando uma organização possui mais de um projeto de desenvolvimento de software, na grande maioria das vezes, existirá um processo de governança e um processo de desenvolvimento de software. Este processo pode não ser explícito, muitas vezes instável com alterações descontroladas, mas ele existe.

Definir e explicitar processos de governança, principalmente em empresas com vários projetos simultâneos, facilita o planejamento, a divulgação e a adoção de praticas mais ágeis. Processos de governança e de desenvolvimento devem ser simples, fácil de evoluir e manter, para promover agilidade.

Segundo a documentação do EPF, ele pode ajudar em dois aspectos principais:

- Primeiramente, as equipes de desenvolvimento precisam ser instruídas sobre os métodos aplicáveis aos papéis que desempenham. Os desenvolvedores de software precisam aprender quais práticas devem usar; os testadores precisam aprender como testar as aplicações baseando-se nas práticas definidas; os gerentes precisam aprender como gerenciar o escopo do projeto, e assim por diante. Funciona como uma ajuda online.
- Em segundo lugar, as equipes de desenvolvimento precisam compreender como aplicar esses métodos durante todo o ciclo de vida do desenvolvimento. Isto é, elas precisam definir ou selecionar um processo de desenvolvimento. As equipes também precisam entender claramente como as diferentes tarefas dos métodos estão relacionadas entre si; por exemplo, como o método de gerenciamento de mudanças influencia no método de gerenciamento de requisitos. Até mesmo as equipes auto-organizadas,

com pessoas experientes, precisam definir um processo que fornece, no mínimo, alguma orientação sobre como o desenvolvimento terá o escopo definido em todo o ciclo de vida, quando marcos serão atingidos e verificados, etc.

O EPF tem como principal objetivo fornecer meios para manter uma base de conhecimento de capital intelectual que você possa procurar, gerenciar e criar. Este conteúdo pode ser externo e, o mais importante, você pode incluir seu próprio conteúdo como, white papers, diretrizes, modelos, boas práticas, procedimentos e regulamentações internas, material de treinamento e qualquer outra descrição geral de seus métodos. Essa base de conhecimento pode ser utilizada para referência e disseminação de conhecimento. Vejamos abaixo os principais recursos do EPF:

- Fornece ferramentas para autoria, configuração, visualização e publicação de métodos e processos;
- Fornece editores de texto avançados e intuitivos para criar descrições de conteúdo ilustrativas. Os editores permitem o uso de estilos, imagens, tabelas, hyperlinks e edição direta no HTML;
- Permite criar processos e diagramas de fluxo de trabalho. Suporta diferentes visualizações de processo: visualização de divisão de trabalho, visualização de produto de trabalho e visualização de alocação de perfis;
- Possui recursos de reutilização e capacidade de extensão. Suporta padrões de processo reutilizáveis e vinculados dinamicamente para rápida montagem do processo através de arrastar e soltar;
- Fornece uma terminologia comum para documentar processos e métodos, o UMA (Unified Method Architecture);

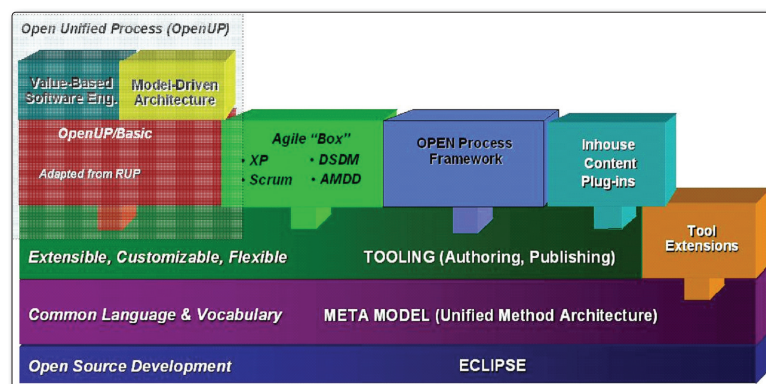


Figura 1. Arquitetura do EPF



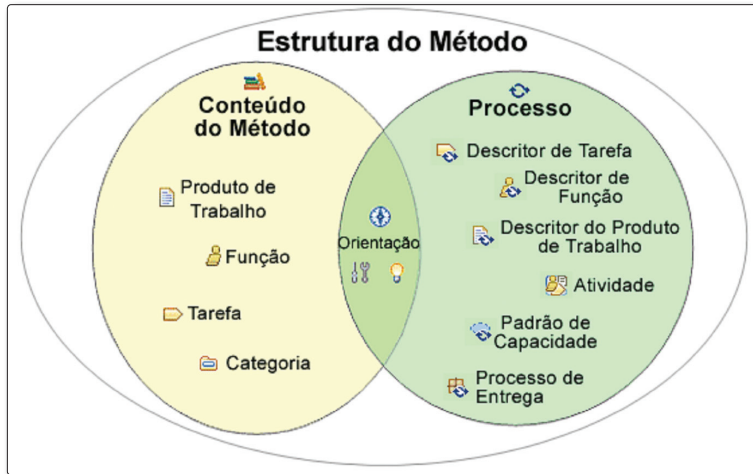


Figura 2. Visão geral dos conceitos principais do EPF que são baseados na UMA

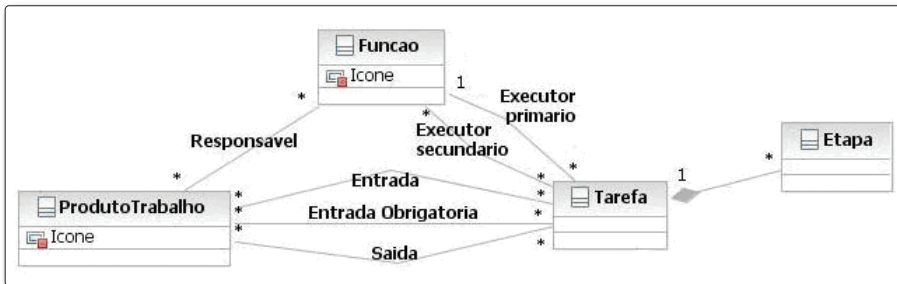


Figura 3. Núcleo reutilizável do conteúdo do método no UMA

• Existem plug-ins com processos pré-definidos como: XP, OpenUp e Scrum. Desenvolvido com o Eclipse (Open Source), como pode ser visto na **Figura 1**.

### O que é UMA?

O EPF utiliza o UMA (Unified Method Architecture), que é um metamodelo de engenharia que define esquema e terminologia para representar métodos e processos. Os conceitos do UMA norteiam todas as funcionalidades do EPF. O UMA separa o Conteúdo do Método, que é o núcleo reutilizável, dos Processos. Isso assegura a reutilização em atividades de autoria de processo no EPF.

A **Figura 2** fornece um resumo dos elementos-chave utilizados no UMA, e como eles estão relacionados ao Conteúdo do Método ou Processo. Como se pode ver, o Conteúdo do Método é expresso, principalmente, utilizando produtos de trabalho, funções, tarefas e orientação. O lado direito do diagrama representa os processos no EPF. Os Processos de Entrega representam um modelo de processo completo e integrado para executar um tipo específico de projeto. Eles descrevem um ciclo de vida completo, de ponta

a ponta do projeto, e são utilizados como uma referência para executar projetos com características similares.

O Conteúdo do Método descreve o que deve ser produzido, as habilidades necessárias e a explicação passo a passo que descreve como as metas de desenvolvimento específicas são atingidas, independentemente do posicionamento desses itens dentro de um ciclo de vida de desenvolvimento. Já os Processos obtêm esses elementos do método e os relatam para seqüências semi-ordenadas que são personalizadas para tipos específicos de projetos. Pode-se, assim, reutilizar os elementos do Conteúdo do Método em processos diferentes, como desenvolvimento de sistemas WEB e desenvolvimentos de pacotes.

### Principais conceitos do UMA

O Conteúdo do Método é fundamentalmente descrito pela definição de Tarefa relatada com Etapas que têm Produtos de Trabalho como entrada e saída e que são desempenhadas por Funções (papéis). As Funções também definem relacionamentos de responsabilidades importantes para Produtos de Trabalho, ver **Figura 3**.

Os principais elementos do Conteúdo do Método são:

- **Produto de Trabalho** - Um Produto de Trabalho é algo significativo resultante da execução de uma tarefa. As pessoas utilizam Produtos de Trabalho para desempenhar Tarefas e produzir Produtos de Trabalho, durante o desempenho de suas Tarefas;

- **Função** - Uma Função define um conjunto de habilidades, competências e responsabilidades relacionadas;

- **Tarefa** - Uma Tarefa descreve uma unidade de trabalho. Cada Tarefa é desempenhada por Funções específicas. A granularidade de uma Tarefa geralmente é algumas horas em poucos dias.

A Orientação é um conceito abstrato que generaliza o conteúdo cuja finalidade principal é fornecer explicações e ilustrações adicionais a elementos, por exemplo: Funções, Tarefas, Produtos de Trabalho ou Processos.

A Orientação é fornecida em vários tipos, como:

- **Lista de Verificação** - Identifica uma série de itens que precisam ser concluídos ou verificados. As listas de verificação são frequentemente utilizadas em revisões, como tentativas ou inspeções;

- **Conceito** - Destaca idéias principais e princípios básicos dando suporte a um tópico central para o método. Os Conceitos normalmente endereçam tópicos mais gerais e se estendem através de vários Produtos de Trabalho, Tarefas ou atividades;

- **Exemplo** - Representa uma instância de amostra normal, parcialmente concluída de um ou mais Elementos de Conteúdo. Geralmente, os exemplos são fornecidos para Produtos de Trabalho;

- **Diretriz** - Fornece detalhes adicionais sobre como tratar de um Elemento de Conteúdo específico. As Diretrizes geralmente são mais aplicadas a Tarefas e Produtos de Trabalho;

- **Prática** - Apresenta uma maneira comprovada ou uma estratégia de realização do trabalho, para atingir uma meta que possui impacto positivo no Produto de Trabalho ou qualidade do processo;

- **Recurso Reutilizável** - Descreve um recurso que pode ser reutilizado em um contexto diferente;

- **Roteiro** - Resume um Processo, muitas vezes, de uma perspectiva específica, como por exemplo, focando em produtos ou papéis do processo;

- **Gabarito** - Especifica a estrutura de um Produto de Trabalho com um formato padronizado, fornecendo descrições do conteúdo;
- **Definição de Termo** - Define conceitos que são utilizados para construir o Glossário;
- **Mentor de Ferramentas** - Apresenta como utilizar uma ferramenta específica para criar parte de um Produto de Trabalho no contexto;
- **White Paper** - Tipo de orientação para documentos publicados externamente, que podem ser lidos e entendidos isoladamente de outros Elementos do Conteúdo.

O **Processo** estabelece as definições de trabalho estruturadas que delineiam o trabalho a ser desempenhado no ciclo de vida. Um processo adquire elementos de métodos do núcleo reutilizável como, por exemplo, as Tarefas e Produtos de Trabalho, e os relaciona em seqüências ordenadas.

### Organização da Biblioteca de Métodos do EPF

Uma Biblioteca de Métodos é um contêiner físico para plug-ins de método. Um plug-in de método é um contêiner para

pacotes de métodos que contêm o Conteúdo de Método. Os plug-ins de método e pacotes de métodos organizam o conteúdo para possibilitar a reutilização.

Um recurso importante fornecido pelo EPF são os plug-ins extensíveis. Esse mecanismo permite a customização do conteúdo gerado, sem alterar diretamente o conteúdo base. O conteúdo do método base incluído no EPF é protegido contra a modificação direta. Os plug-ins de leitura são esmaecidos na visualização de biblioteca, indicando que estão bloqueados.

Quando for adaptar algum processo para suas necessidades, sempre deve ser criado um novo plug-in estendendo o original, isso separa seu conteúdo do conteúdo original. Permite também que você atualize sua biblioteca com novos releases, sem afetar o conteúdo criado em seus próprios plug-ins.

### Visão geral

A documentação do EPF foca em quatro conceitos principais: autoria do Conteúdo do Método; autoria de Processo; configurações de métodos e publicação. Com eles podemos obter uma visão geral do funcionamento do EPF.

Na Autoria do Conteúdo do Método criamos o núcleo reutilizável que são: os papéis (funções), tarefas, produtos de trabalho e seus relacionamentos, conforme mostrado na **Figura 3**. Na autoria de Processo estabelecemos uma seqüência lógica de execução para as tarefas criadas anteriormente. Em uma configuração de método você pode selecionar e cancelar seleção dos pacotes de conteúdo disponíveis. As seleções feitas ajudam a determinar o conteúdo de sua publicação. Publicação é um Web site com o método e processos que podem ser utilizados por uma equipe de projeto.

### Visão geral de autoria do Conteúdo do Método

A **Figura 4** ilustra o Conteúdo do Método representado no EPF. Muitos métodos de desenvolvimento são descritos em publicações, como manuais, artigos, material de treinamento, padrões, regulamentações e outras formas de documentação. Essas origens geralmente documentam métodos, fornecendo explicações etapa por etapa para uma maneira específica de atingir uma meta no desenvolvimento. Alguns exemplos são:

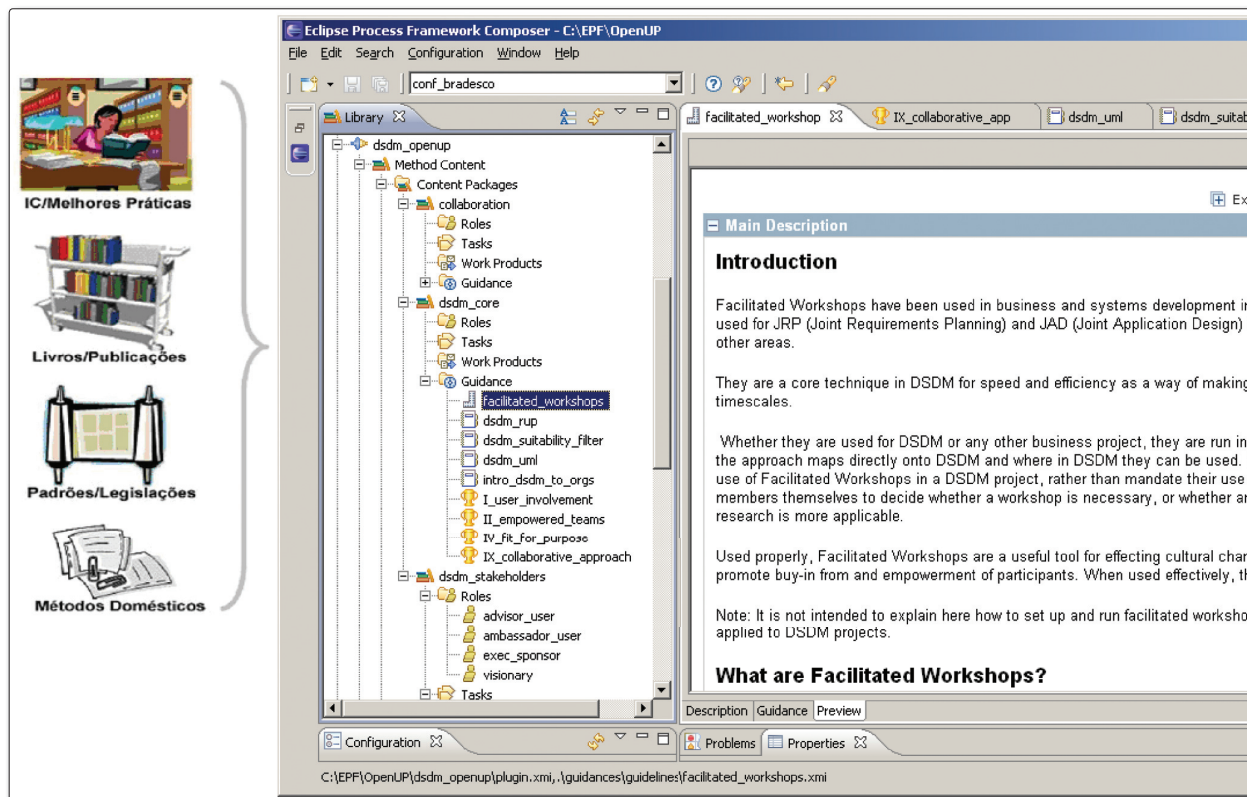


Figura 4. Autoria do conteúdo do método

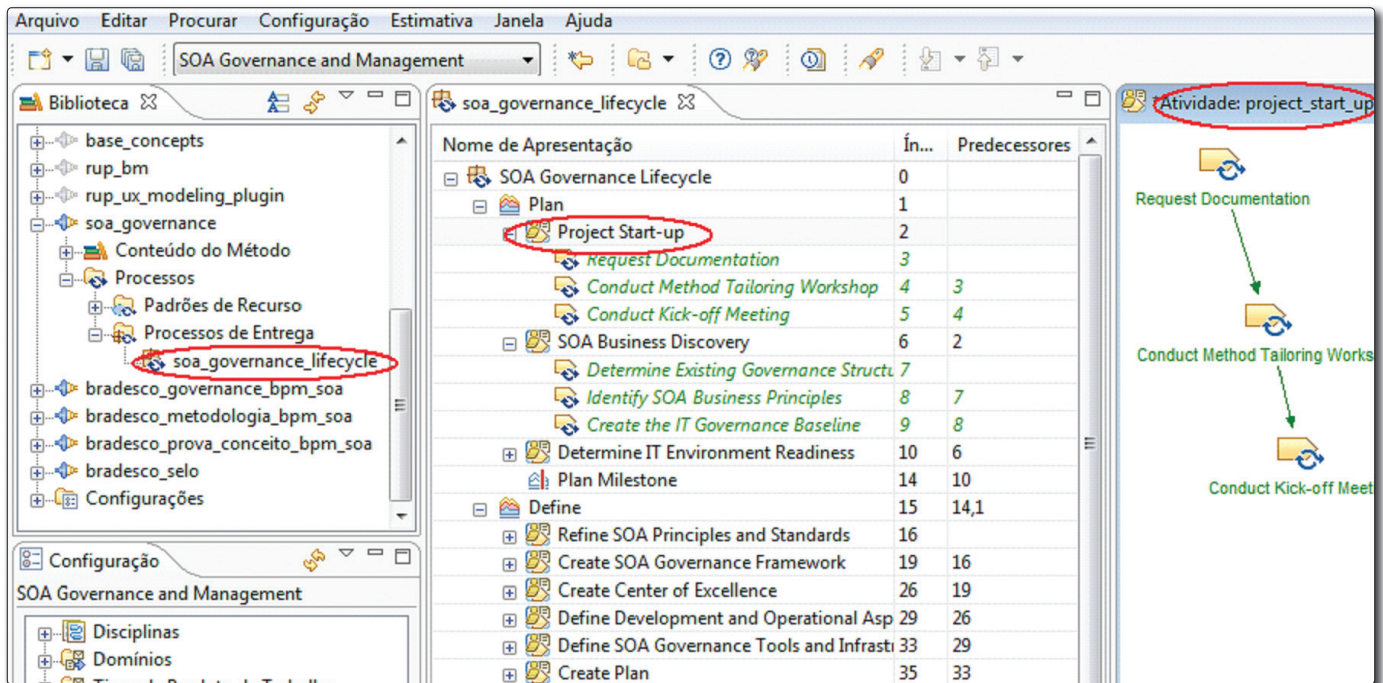


Figura 5. Autoria de Processo

transformar um documento de requisitos em um modelo de análise; definir um mecanismo arquitetural baseado em requisitos funcionais e não funcionais; criar um plano de projeto para uma iteração de desenvolvimento; definir um plano de garantia de qualidade para requisitos funcionais; projetar novamente uma organização de negócios com base em uma nova orientação estratégica, e assim por diante.

O EPF permite a distribuição do conteúdo e a sua estruturação em um esquema específico de Funções (Papeis), Produtos de Trabalho, Tarefas e Orientações. Tal esquema suporta a organização de grandes quantidades de descrições para métodos e processos de desenvolvimento. Esse Conteúdo do Método e Processos não precisa ser limitado à engenharia de software, mas também pode abranger outras disciplinas de engenharia, como engenharia mecânica, transformação de negócios, ciclos de venda, etc.

Ele também permite que você categorize seu conteúdo com base em um conjunto de categorias pré-definidas (por exemplo, categorizar suas tarefas em disciplinas de desenvolvimento ou seus produtos de trabalho em domínios) ou crie seus próprios esquemas de categorização para o conteúdo, indexando da maneira desejada.

### Visão geral da autoria de Processo

Um processo define seqüências de tarefas desempenhadas por Funções e os Produtos de Trabalho produzidos ao longo do tempo.

A Figura 5 mostra que os processos são geralmente representados como fluxos de trabalho. Portanto, para definir um processo, um indivíduo pode pegar o Conteúdo do Método e combiná-lo em estruturas que especificam como o trabalho deve ser organizado ao longo do tempo. Isto ocorre para atender às necessidades de um tipo específico de projeto de desenvolvimento. O EPF suporta processos com base em diferentes abordagens de desenvolvimento em vários modelos de ciclo de vida, incluindo ciclos de vida em cascata, incrementais e iterativos. Você também pode definir processos no EPF que utilizam um conjunto mínimo de Conteúdo do Método para definir processos para equipes ágeis e auto-organizadas.

Na Figura 5 temos um exemplo de processo apresentado como uma estrutura de divisão de atividades aninhadas, bem como um fluxo de trabalho ou diagrama de atividades.

O EPF fornece um editor de processo que suporta diferentes visualizações de estrutura de divisão de trabalho, bem como apresentações gráficas do

processo. Como autor do processo, você geralmente começa criando uma divisão de trabalho, dividindo o processo em fases, iterações e atividades de alto nível. Em vez de criar suas atividades no editor de estrutura de divisão, você pode, como alternativa, trabalhar em um editor de diagrama de atividades gráfico, que permite criar graficamente um fluxo de trabalho para as atividades.

### Visão geral das configurações de métodos

Todos os conteúdos e processos no EPF são organizados em plug-ins de métodos, os quais são estruturados em pacotes de métodos. Uma configuração de método é, simplesmente, uma seleção dos plug-ins e pacotes de métodos.

Você cria e especifica uma configuração utilizando o editor de configurações, como pode ser visto na Figura 6. Você pode começar criando sua própria configuração de método ou copiando uma das configurações existentes no EPF e modificá-la para que esta se adapte às suas necessidades. Você pode incluir ou remover os plug-ins de método por completo, bem como fazer a seleção com cada plug-in, marcando ou desmarcando pacotes. As configurações de métodos são utilizadas para a publicação.

**Visão geral de publicação**

Uma configuração publicada é um Web site HTML que apresenta todos os conteúdos do método e processos da configuração de método em forma passível de procura e de navegação. Ela utiliza os relacionamentos estabelecidos durante a autoria dos Processos e do Conteúdo do Método para gerar hyperlinks entre elementos, bem como fornece navegadores de árvore com base na visualização de configuração e nas categorizações do conteúdo definidas pelo usuário. A **Figura 7** mostra um exemplo da configuração do método publicada. No endereço <http://epf.eclipse.org/wikis/openuppt/> é possível ver um exemplo de publicação.

Para publicar, basta criar e selecionar uma configuração. Por exemplo, você pode criar uma publicação com processos importantes para uma equipe de requisitos utilizando o Conteúdo do Método corporativo. Para isso, basta criar uma configuração com apenas os plug-ins e pacotes de métodos referentes a requisitos. O assistente de publicação fará o restante, e publicará

apenas o conteúdo que faz parte da configuração do método. Ela também adotará, automaticamente o conteúdo para a configuração, removendo referências de elementos do Conteúdo do Método não selecionados na configuração, ou removendo atividades dos processos que contêm Tarefas que estão fora da configuração da publicação. Portanto, a publicação incluirá apenas o conteúdo que for realmente necessário. Você pode visualizar uma configuração publicada utilizando a perspectiva de procura do EPF.

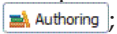
**Como criar um conteúdo de método**

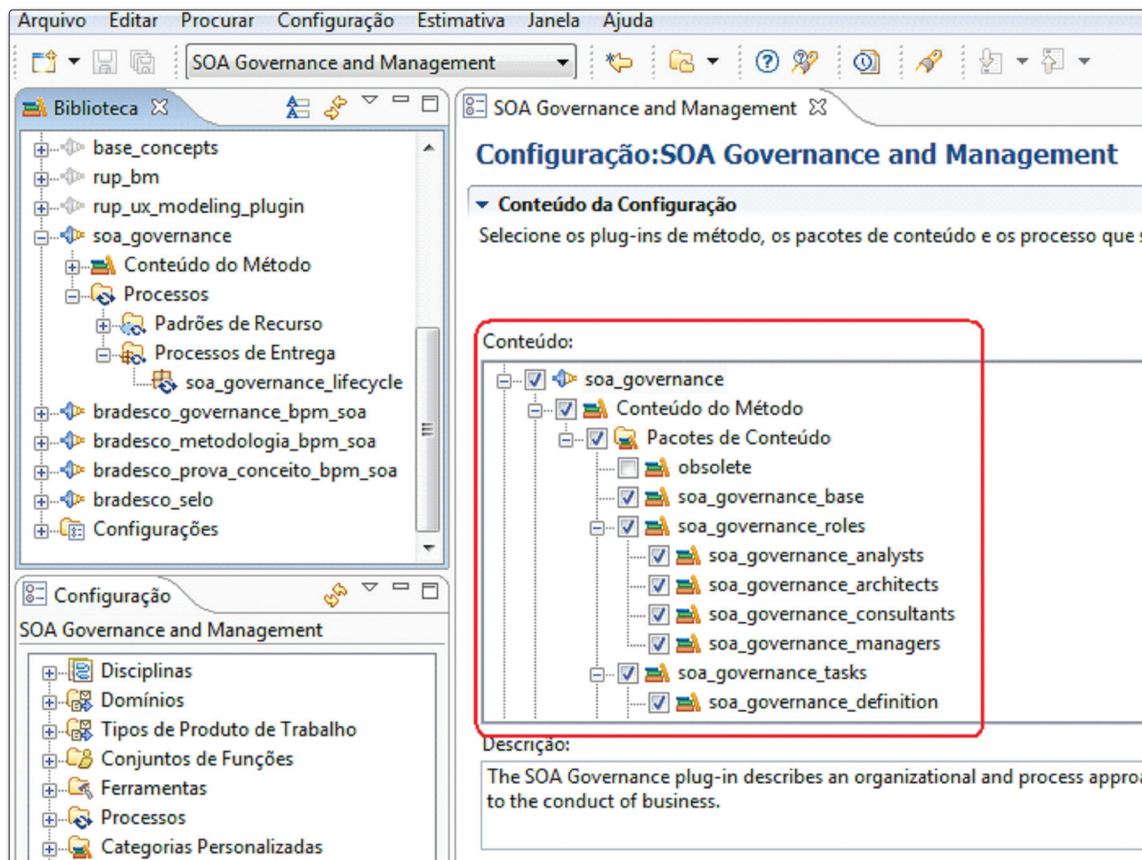
Vamos agora mostrar passo a passo como criar o conteúdo de método seguindo a documentação do EPF. Com o conteúdo de método criado você poderá criar seus processos e publicações. Outros tópicos como estender um conteúdo de método existente, como criar/adaptar processos e configurar publicações serão apresentados nos próximos artigos.

Os elementos do método e do processo utilizam dois nomes: **Nome** e **Nome da Apresentação**. O campo **Nome** é utiliza-

do como o nome de arquivo para o item. É recomendável utilizar nomes de arquivos com todas as letras em minúsculo, sem espaços e caracteres especiais (para manter compatibilidade com outras plataformas). O **Nome da Apresentação** é o nome mostrado nas páginas publicadas. Diferentemente do campo **Nome**, o campo **Nome da Apresentação** pode conter caracteres maiúsculos, espaços e símbolos especiais, como o caractere de marca registrada.

Primeiramente, criaremos um novo plug-in de método; em seguida, criaremos um novo pacote de conteúdo dentro dele. Para isso:

- Certifique-se de que esteja na perspectiva **Autoria** ;
- Utilize o menu **Arquivo** e selecione **Novo**, em seguida, **Plug-in de Método**, como na **Figura 8**. Dessa forma o assistente **Novo Plug-in de Método** é apresentado;
- No assistente **Novo Plug-in de Método** (**Figura 9**), forneça um nome para o novo plug-in. Neste exemplo chamamos de "my\_plug-in";
- Selecione **base\_concepts** (**Figura 9**) no painel **Plug-ins referenciados**: e clique em **Concluir**. Isso significa que seu



**Figura 6.** Configurações de Método





Figura 7. Exemplo de publicação

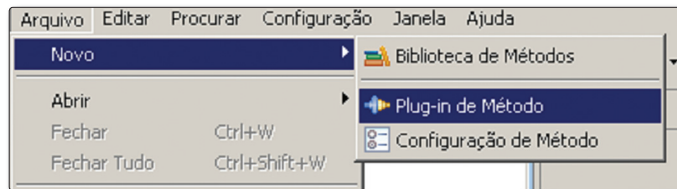


Figura 8. Ativando o assistente de publicação

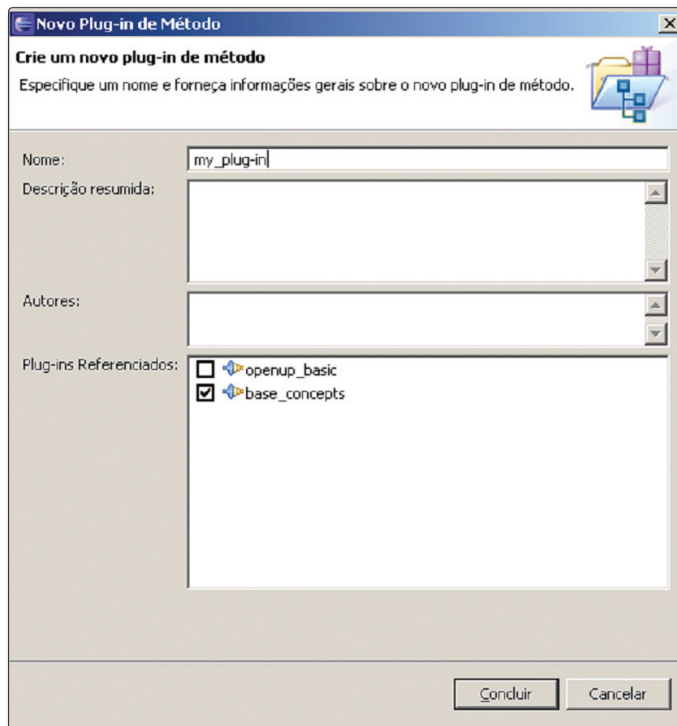


Figura 9. Assistente Novo Plug-in de Método

plug-in estenderá o plug-in **base\_concepts** e você conseguirá utilizar o conteúdo deste enquanto cria o conteúdo de seu próprio plug-in, mantendo uma referência para os elementos do plug-in **base\_concepts** (esse procedimento deve ser feito sempre que for necessário reutilizar elementos de outro plug-in como, por exemplo, papéis e tarefas). O **base\_concepts** contém conceitos básicos necessários para a utilização do site publicado com seu processo/método, por exemplo, instruções de navegação e conceitos básicos do UMA. Por isso, essas orientações devem estar em todas as publicações;

- Agora vamos criar um pacote de conteúdo do método no novo plug-in. No painel de visualização Biblioteca (Figura 10) expanda o nó da árvore em my\_plug-in clicando no símbolo +, e expanda o nó para **Conteúdo do Método**;

- Clique com o botão direito do mouse em **Pacotes de Conteúdo** (Figura 11), selecione **Novo** e depois **Pacote de Conteúdo** para criar um novo Pacote de Conteúdo;

- O Editor de Pacote de Conteúdo é exibido (Figura 12). Digite o nome do pacote no campo **Nome** e algum texto no campo **Descrição resumida**. Posteriormente, o campo Dependências será



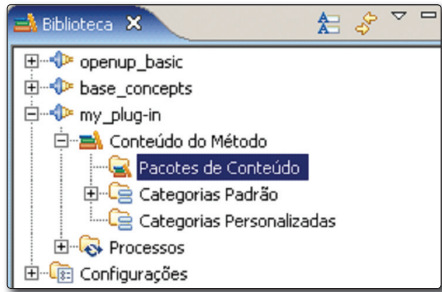



Figura 10. Árvore de visualização de Biblioteca

automaticamente preenchido quando algum conteúdo do pacote depender do conteúdo de outro pacote. Por exemplo, uma tarefa criada no pacote A estende uma tarefa que está no pacote B, acrescentando mais passos na tarefa. Dessa forma, o pacote A dependerá do pacote B.

- Salve o novo conteúdo (Ctrl + S) e feche o painel do editor;

- Agora criaremos o conteúdo do Método. Em geral, os novos elementos (tarefas, produtos de trabalho, funções ou papéis e orientações) são criados clicando com o botão direito do mouse em uma pasta de destino para o novo elemento e selecionando **Novo**. Por exemplo, se você deseja criar um novo **Produto de Trabalho** em um pacote de conteúdo, clique com o botão direito na pasta **Produto de Trabalho** existente na árvore de navegação, selecione **Novo** e clique em **Artefato**, como na Figura 13;

- Digite algo nos campos **Nome** e **Nome da Apresentação** (Figura 14). Você pode ainda, nos campos **Objetivo** e **Descrição Principal**, descrever seu artefato com um editor de texto avançado usando, por exemplo, cores e figuras. Você pode abrir o editor de texto para qualquer item que tenha o ícone de editor de texto  próximo a ele (por exemplo: veja o campo **Descrição Principal** na Figura 14). Com o editor aberto também é possível usar diretamente HTML para criar o conteúdo (Figura 15). Observação: O texto copiado de documentos do Microsoft Word contém códigos HTML que podem causar problemas de formatação quando colados em qualquer uma das janelas do editor. Desta forma, recomenda-se copiar o texto de uma ferramenta que não utilize marcações "fechadas" para manter a estrutura do documento.

- Na Figura 14, clique na guia **Visualizar** para ver como ficou o estado do novo Artefato. Salve-o (Ctrl + S) e feche a janela do editor;

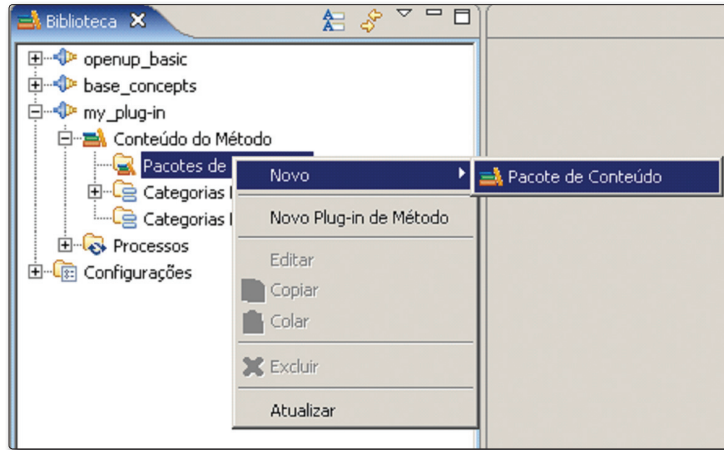


Figura 11. Criando um novo Pacote de Conteúdo

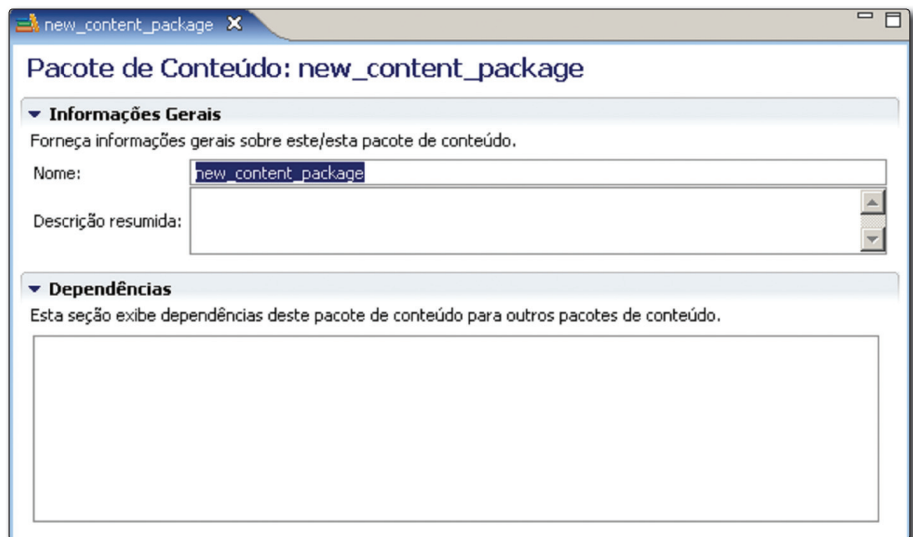


Figura 12. Editor de Pacote de Conteúdo

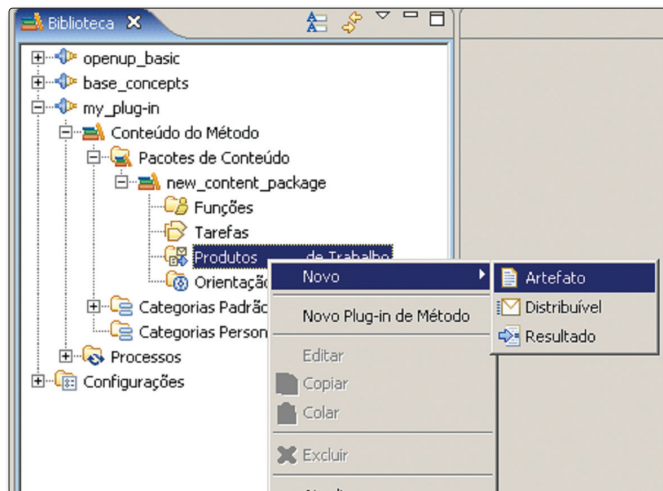


Figura 13. Criando um conteúdo de método

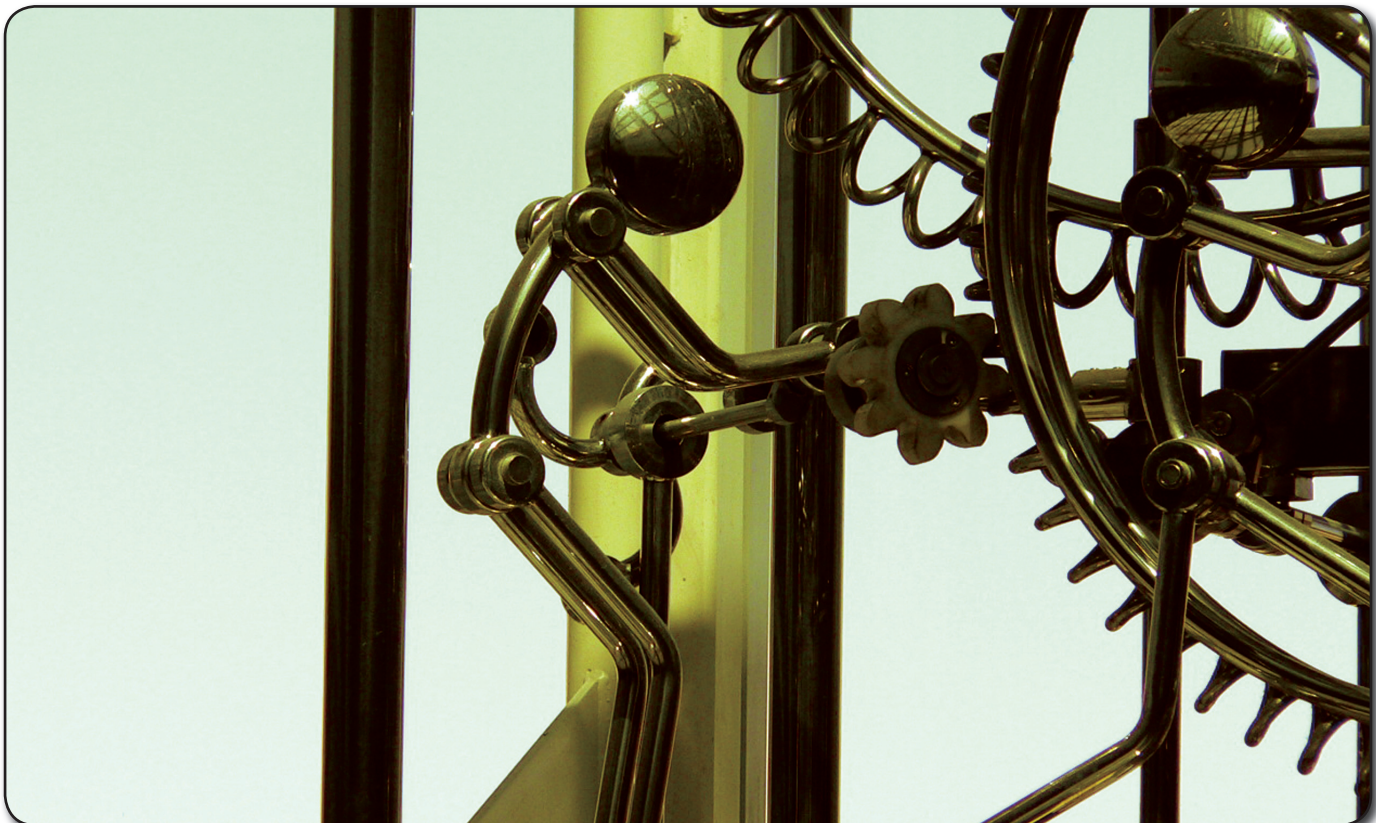




Figura 14. Editando um conteúdo de método

Figura 15. Editor de texto

Figura 16. Guias para criação de relacionamentos do conteúdo de método (tarefas, produto de trabalho e orientações)



- Repita o processo para todo conteúdo necessário;
- Depois de criados os artefatos do seu método, crie os Relacionamentos desses artefatos selecionando uma das guias de associação (conforme mostrado da **Figura 16**). Cada guia destacada na **Figura 16** representa um tipo de artefato que podemos associar à tarefa que esta sendo editada, por exemplo, se selecionarmos a guia Funções (Papéis) poderemos escolher qual papel do seu método será responsável por essa tarefa.
- Finalmente, podemos gerar uma publicação com o conteúdo criado, onde é possível navegar com os links gerados automaticamente. Observe um exemplo de publicação, na **Figura 17**, onde podemos navegar no método criado usando a árvore de navegação, que possibilita

percorrer toda a hierarquia do conteúdo criado. Selecionando a função (Papel) de arquiteto na árvore de navegação, o EPF mostra graficamente os relacionamentos desta Função com suas Tarefas e seus Produtos de trabalho.

**Conclusão**

O EPF fornece ferramentas, um meta-modelo unificado e conteúdos que podem ser usados como a base para manter ou criar uma grande variedade de processos de TI, como processos de governança SOA e processos de desenvolvimento de Software. Essas ferramentas fornecem recursos para seleção, personalização e rápida montagem de processos, onde é possível utilizar processos pré-definidos, como o *XP* e o *OpenUP*, e adaptá-los às necessidades de um projeto específico ou da organização. ●

**Links**

**Eclipse Process Framework Project (EPF)**  
<http://www.eclipse.org/epf/>

**Getting Started**  
[http://www.eclipse.org/epf/general/getting\\_started.php](http://www.eclipse.org/epf/general/getting_started.php)

**EPF Wiki**  
<http://epf.eclipse.org/>

**Who will benefit from EPF**  
<http://www.eclipse.org/proposals/beam/Who%20will%20benefit%20from%20Eclipse%20Process%20Framework.pdf>

**Dê seu feedback sobre esta edição!**

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:  
[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



**OpenUP/Basic** Glossário

Onde eu estou | Conjuntos de Árvores |

**Função: Arquiteto** Edit/Info

Este papel é responsável por definir a arquitetura do software, incluindo a tomada das principais decisões técnicas que orientam o design e a implementação do projeto.

Conjuntos de Funções: OpenUP/Basic Roles

**Relacionamentos** Expandir Todas as Seções | Reduzir

**Navegação hierárquica** →

**Navegação gráfica** ↗

**Adicionalmente Desempenha:**

- Assess Results
- Define Vision
- Design the Solution
- Detail Requirements
- Find and Outline Requirements
- Manage Iteration
- Plan Iteration
- Plan Project

**Modifica:**

- Architecture Notebook
- Design

Figura 17. Exemplo de publicação com as possibilidades de navegação



# Introdução à Automação de Testes



**Cristiano Caetano**

[c\\_caetano@hotmail.com](mailto:c_caetano@hotmail.com)

É certificado CBTS pela ALATS. Consultor de teste de software sênior com mais de 10 anos de experiência, já trabalhou na área de qualidade e teste de software para grandes empresas como Zero G, DELL e HP Invent. É colunista na área de Teste e Qualidade de software do site [linhadecodigo.com.br](http://linhadecodigo.com.br) e autor dos livros "CVS: Controle de Versões e Desenvolvimento Colaborativo de Software" e "Automação e Gerenciamento de Testes: Aumentando a Produtividade com as Principais Soluções Open Source e Gratuitas". Criador e mantenedor do portal TestExpert: A sua comunidade gratuita de teste e qualidade de software ([www.testexpert.com.br](http://www.testexpert.com.br)).

Segundo Cem Kaner, autor do livro "Lessons Learned in Software Testing", o propósito da automação de testes pode ser resumidamente descrito como a aplicação de estratégias e ferramentas tendo em vista a redução do envolvimento humano em atividades manuais repetitivas.

A automação possibilita a execução de testes regressivos com maior amplitude e profundidade. Teste regressivo ou teste de regressão é o termo utilizado para o ciclo de re-teste de uma ou mais funcionalidades, a fim de identificar defeitos introduzidos por novas funcionalidades ou correção de defeitos.

A cada novo ciclo de teste, o time de testes normalmente executa os testes das novas funcionalidades e os testes regressivos das demais funcionalidades. Dessa forma, é possível encontrar algum efeito colateral ou instabilidade introduzida pela nova funcionalidade. O grande problema ocorre quando em um estágio avançado do desenvolvi-

mento, gasta-se mais tempo executando testes regressivos do que testando as novas funcionalidades.

Uma abordagem de testes baseada puramente em testes manuais, normalmente não consegue acompanhar as demandas e o volume de testes ao longo do ciclo de vida de desenvolvimento de software. Frequentemente o produto é liberado sem que tenha sido completamente testado em virtude de restrições de tempo.

A automação de testes quando utilizada corretamente permite a execução ininterrupta de testes regressivos a qualquer hora do dia ou da noite. A execução de testes automatizados é sempre mais rápida do que os testes manuais e menos suscetível a erros.

A decisão de usar uma abordagem de testes baseada em testes automatizados está em franca expansão na atualidade. Uma pesquisa realizada em 2006 pelo Forrester Research Inc, revela que 9% das empresas entrevistadas (empresas dos Estados Unidos e Europa) utilizam testes

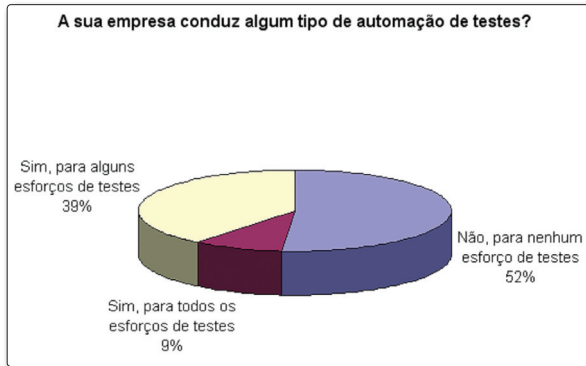


Figura 1. Avaliando Soluções de Testes Funcionais - The Forrester Wave™ Q2 2006.

automatizados em todos os esforços de testes e 39% das empresas responderam que utilizam testes automatizados em alguns esforços de testes (Figura 1).

Nas seções a seguir serão apresentados os principais paradigmas e tipos de testes automatizados, assim como, a comparação das vantagens e desvantagens de cada um deles.

## Paradigmas de automação de testes

Existem várias abordagens para a automação de testes. No entanto, neste artigo serão apresentados apenas os paradigmas mais importantes da atualidade. Além disso, o foco deste artigo é automação de testes funcionais, dessa forma, não serão apresentados ou discutidos os testes unitários (*Unit Tests*) e automação de testes de desempenho (*Performance Testing*).

Os tipos de automação são normalmente agrupados de acordo com a forma como os testes automatizados interagem com a aplicação. Em geral, os tipos de automação são agrupados em dois paradigmas (mas não são limitados a esses):

### Baseados na Interface Gráfica

Nesta abordagem os testes automatizados interagem diretamente com a interface gráfica da aplicação simulando um usuário. Normalmente as ações dos usuários são gravadas (*Capture*) por meio da ferramenta de testes automatizados. A ferramenta transforma as ações dos usuários em um *script* que pode ser reproduzido (*Playback*) posteriormente.

**Vantagens:** Não requer modificações na aplicação para criar os testes automatizados. Também não é necessário tornar a aplicação mais fácil de testar (testabilidade) porque os testes se baseiam na mesma interface utilizada pelos usuários.

**Desvantagens:** Existe uma forte dependência da estabilidade da interface gráfica. Se a interface gráfica mudar, os testes falham. Baixo desempenho para testes automatizados que exigem centenas de milhares de repetições, testes de funcionalidades que realizam cálculos complexos, integração entre sistemas diferentes e assim por diante.

### Baseados na Lógica de Negócio

Nesta abordagem os testes automatizados exercitam as funcionalidades da aplicação sem interagir com a interface gráfica. Normalmente é necessário realizar modificações na aplicação para torná-la mais fácil de testar (testabilidade). Essas modificações resultam em mecanismos para expor ao mundo exterior as funcionalidades da aplicação (APIs, Interfaces de Linha de Comando, *Hooks*, etc), como veremos mais adiante.

A interface gráfica é apenas uma casca (camada) que tem o objetivo de fornecer um meio para a entrada dos dados e apresentação dos resultados. A camada que abriga a funcionalidade e o comportamento da aplicação é a camada de lógica de negócio. Esta abordagem de testes é baseada no entendimento que 80% das falhas estão associados a erros na lógica de negócio (Figura 2).

**Vantagens:** Foco na camada onde existe maior probabilidade de existir erros. Independência das mudanças da interface gráfica. Alto desempenho para testes automatizados que exigem centenas de milhares de repetições, testes de funcionalidades que realizam cálculos complexos, integração entre sistemas diferentes e assim por diante.

**Desvantagens:** Requer grandes modificações na aplicação para expor as funcionalidades ao mundo exterior. Exige profissionais especializados em programação

Paradigma	Tipo de Teste Automatizado
Baseado na Interface Gráfica	Baseado na interface gráfica (Capture/Playback)
	Dirigido a dados (Data-Driven)
	Dirigido à palavra-chave (Keyword-Driven)
Baseado na Lógica de Negócio	Baseado na linha de comando (Command Line Interface)
	Baseado em API (Application Programming Interface)
	Test Harness

Tabela 1. Tipos de Automação de Testes.

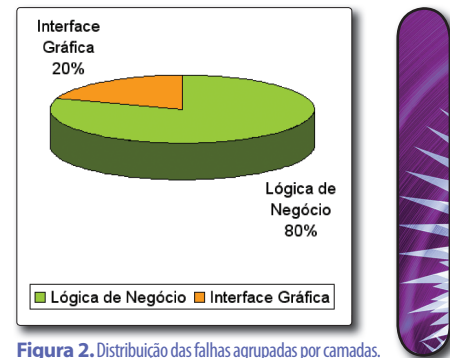


Figura 2. Distribuição das falhas agrupadas por camadas.

para criar os testes automatizados. Existem poucas ferramentas/*frameworks* que suportam essa abordagem (normalmente é necessário criar soluções caseiras).

## Tipos de automação de testes

Conforme mencionado anteriormente, os tipos de automação são normalmente agrupados de acordo com a forma como os testes automatizados interagem com a aplicação. A Tabela 1 apresenta o sumário dos tipos de testes automatizados que serão apresentados neste artigo.

Nas seções a seguir serão apresentados os tipos de testes automatizados, um exemplo de aplicação e as vantagens e desvantagens de cada tipo.

### Testes automatizados baseados na interface gráfica (Capture/Playback)

Nesta abordagem, os testes automatizados são realizados por meio da interface gráfica da aplicação. Normalmente a ferramenta de automação fornece um recurso para capturar (*Capture*) as ações do usuário enquanto o usuário estiver usando a aplicação.

Essas ações são traduzidas para comandos na linguagem de *script* suportada pela ferramenta de automação, para que então possam ser reproduzidas (*Playback*) posteriormente.

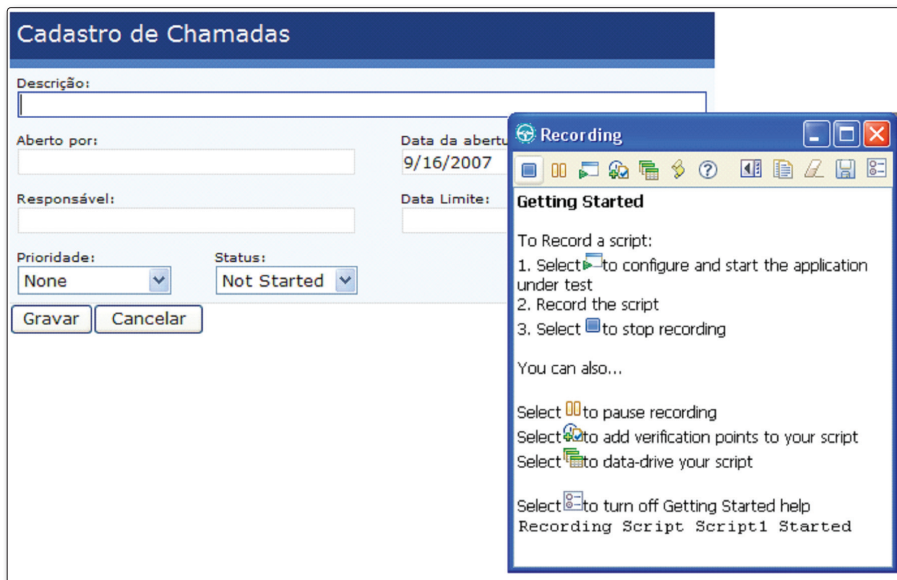


Figura 3. Gravação de um teste automatizado baseado na interface gráfica.

**Listagem 1.** Ações convertidas em comandos na linguagem Java

```
import resources.CadastroHelper;
import com.rational.test.ft.*;
import com.rational.test.ft.object.interfaces.*;
import com.rational.test.ft.script.*;
import com.rational.test.ft.value.*;
import com.rational.test.ft.vp.*;
public class Cadastro extends CadastroHelper
{
    public void testMain(Object[] args)
    {
        Descricao().click(atPoint(147,8));
        Formulario().inputChars("Problemas");
        AbertoPor().click(atPoint(147,8));
        Formulario().inputChars("Pedro Paulo");
        Responsavel().click(atPoint(115,13));
        Formulario().inputChars("José da Silva");
        BotaoGravar().click(atPoint(42,8));
    }
}
```

Via de regra, esta abordagem não requer modificações na aplicação para criar os testes automatizados. Também não é necessário tornar a aplicação mais fácil de testar (testabilidade) porque os testes se baseiam na mesma interface utilizada pelos usuários.

Para ilustrar um exemplo prático, observe na **Figura 3** um teste hipotético de uma aplicação de cadastro de chamadas técnicas. Para este exemplo, foi utilizada a ferramenta comercial IBM Rational Functional Tester.

Enquanto o teste está sendo executado pelo testador, as ações (cliques, digitação, etc) são gravadas pela ferramenta de automação (janela *Recording*). Ao final da gravação, a ferramenta converte (traduz) todas as ações em comandos em uma linguagem de *script*. No caso do IBM Rational Functional Tester, as ações são convertidas para comandos na linguagem Java, como pode ser visto no trecho de código da **Listagem 1**.

Fundamentalmente, podemos notar que todas as operações realizadas pelo testador foram convertidas para a linguagem Java de acordo com padrões adotados pelo IBM Rational Functional Tester. Este *script* pode ser salvo e agrupado em suítes para ser reproduzido (*Playback*) posteriormente.

Uma das principais vantagens dessa abordagem é a rapidez e facilidade para criar *scripts* de teste. Também devemos destacar o fato de que não são necessárias modificações na aplicação para a gravação e posterior reprodução dos testes. Além disso, não é necessário tornar a aplicação mais fácil de testar (testabilidade) porque os testes se baseiam na mesma interface utilizada pelos usuários.

Em contrapartida, a criação de muitos *scripts* de teste rapidamente, sem planejamento e conhecimento das melhores práticas da automação de testes, normalmente leva à criação de centenas de *scripts* com código spaghetti. Ou seja,

código confuso com baixa reutilização de rotinas e difícil de dar manutenção. Além disso, nesta abordagem existe uma forte dependência da estabilidade da interface gráfica. Se a interface gráfica mudar, os testes falham.

**Testes automatizados dirigidos a dados (Data-Driven)**

Os testes automatizados dirigidos a dados representam um refinamento dos testes baseados na interface gráfica. Basicamente, nesta abordagem, é utilizado um mecanismo para auxiliar a execução de testes que executam as mesmas ações repetidamente, porém com dados diferentes.

Tomemos como exemplo o cadastro de chamadas técnicas citado anteriormente. Digamos que seja necessário criar vários testes a fim de avaliar se a aplicação permite gravar os dados sem que todos os campos tenham sido preenchidos ou preenchidos com dados inválidos. Neste caso, as ações seriam as mesmas, apenas os dados preenchidos nos campos mudariam.

Na abordagem de testes baseados na interface gráfica, teríamos que capturar e gerar diversos *scripts* de testes. No entanto, à medida que se queira criar centenas ou milhares de *scripts* de testes, as coisas começam a se complicar. O tempo para capturar e gerar os *scripts* de teste aumenta consideravelmente. Além disso, a complexidade e o tempo para dar manutenção nesses *scripts* também aumentam exponencialmente.

Por meio da criação de testes automatizados dirigidos a dados, os dados deixam de ser constantes dentro dos *scripts* e tornam-se variáveis. Para ilustrar, vamos apresentar um exemplo na prática usando a ferramenta IBM Rational Functional Tester.

Esta ferramenta permite a criação de testes automatizados dirigidos a dados por meio da utilização de um mecanismo chamado *Test Datapool*. O *Test Datapool* é basicamente uma tabela que armazena os dados variáveis de um ou mais *scripts*.

No nosso exemplo, o *Test Datapool* irá armazenar os dados que serão repetidos no cadastro de chamada técnica para avaliar se a aplicação permite gravar os dados sem que todos os campos tenham sido preenchidos ou preenchidos com dados inválidos, como pode ser observado na **Figura 4**.

Dessa forma, apenas um único *script* deverá ser capturado. Durante a captura (*Capture*) ou posteriormente, todos os dados constantes, poderão ser trocados por referências aos dados contidos no *Test Datapool*, como pode ser visto na **Listagem 2**.

Como você deve ter notado no trecho de código da **Listagem 2**, os dados constantes foram trocados por comandos em Java que representam o mecanismo utilizado pelo Rational Functional Tester para referenciar os dados contidos no *Test Datapool*. Assim, em vez do dado constante "Marcos", você encontrará o comando `inputChars(dpString("AbertoPor"))`.

O Rational Functional Tester, por sua vez, durante a execução do teste automatizado trocará todas as referências pelos dados contidos no *Test Datapool*. Dessa forma, a execução do teste se repetirá enquanto existirem dados no *Test Datapool*.

Uma das principais vantagens dessa abordagem é a reutilização dos *scripts*, o que conseqüentemente diminui a complexidade e o tempo de manutenção.

Por outro lado, nesta abordagem também existe uma forte dependência da estabilidade da interface gráfica. Se a interface gráfica mudar, os testes falham. Além disso, os mecanismos oferecidos pelas ferramentas de automação que permitem a criação de testes automatizados dirigidos a dados não são muito robustos. Muitas vezes não é possível criar testes dirigidos a dados aninhados com outros testes dirigidos a dados ou definir critérios para usar apenas um subconjunto dos dados existentes com base em alguma condição durante a execução do teste.

Por fim, é importante destacar que o conceito de testes automatizados dirigidos a dados é muito mais amplo do que possa parecer. Apesar de classificarmos essa abordagem como uma abordagem de teste automatizado independente e baseada na interface gráfica, ela pode ser aplicada em qualquer outro tipo de teste automatizado descrito neste artigo.

### Testes automatizados dirigidos à palavra-chave (Keyword-Driven)

Esta abordagem foi criada para dar suporte aos testes de aceitação (*Acceptance Tests*) preconizados por metodologias ágeis, tais como o XP (*Extreme Programming*). Os testes de aceitação são normalmente definidos pelo usuário final em

conjunto com analistas e testadores. A principal função dos testes de aceitação é definir os passos e critérios para aceitar um requisito (ou uma *user story*).

Nesta abordagem, os testes automatizados são realizados por meio da interface gráfica da aplicação. No entanto, os testes são baseados em palavras-chaves (*keywords*). Normalmente a ferramenta de automação oferece um conjunto pré-definido de palavras-chaves para permitir a criação dos testes.

Cada palavra-chave é um comando em alto nível (praticamente em linguagem nativa) que representa uma ação do usuário. Dessa forma, os testes são facilmente entendidos (e até escritos) pelos usuários finais em virtude do alto nível de abstração.

Para ilustrar, consideremos um exemplo prático. Suponha que seja necessário criar um teste para validar o mecanismo de simulação de crédito imobiliário de um website de uma instituição financeira. Neste cenário, o usuário final poderá escrever um teste de aceitação por meio de palavras-chaves (*keywords*). As palavras-chaves representam os passos para

abrir o endereço da página e a digitação dos valores nos campos (valor do imóvel, valor do financiamento, etc).

As palavras-chaves também poderão representar cliques nos botões para realizar o cálculo da simulação e a validação do resultado esperado do teste, ou seja, o critério de aceitação do teste. Não existem limites para as ações que uma palavra-chave possa realizar, exceto, as limitações impostas pela ferramenta de testes automatizados.

A título de exemplo, vamos demonstrar essa abordagem utilizando a ferramenta *Open Source Selenium IDE*. O Selenium IDE suporta a criação de testes automatizados para aplicações WEB dirigidos à palavra-chave (*Keyword-Driven*), como pode ser observado na **Figura 5**.

Como você pode perceber na **Figura 5**, a coluna "Command" representa a palavra-chave e as colunas "Target" e "Value" representam os argumentos. Apesar do fato que as palavras-chaves estão escritas em inglês, podemos notar claramente as ações para abrir o endereço da página (*Open*), digitação dos valores nos campos

	Titulo::	AbertoPor::	Responsavel::
	Problemas na impressora	Marcos	Pedro
1	Problemas na impressora	Marcos	
2	Problemas na impressora		Pedro
3			
4	@###%&=;/;		
5	@###%&=;/;	@###%&=;/;	@###%&=;/;

Figura 4. Test Datapool contendo os dados variáveis de um teste data-driven.

#### Listagem 2. Os dados constantes são convertidos em referências ao Test Datapool

```
import resources.CadastroHelper;
import com.rational.test.ft.*;
import com.rational.test.ft.object.interfaces.*;
import com.rational.test.ft.script.*;
import com.rational.test.ft.value.*;
import com.rational.test.ft.vp.*;
public class Cadastro extends CadastroHelper
{
    public void testMain(Object[] args)
    {
        Descricao().click(atPoint(147,8));
        Formulario().inputChars(dpString("Titulo"));
        AbertoPor().click(atPoint(147,8));
        Formulario().inputChars(dpString("AbertoPor"));
        Responsavel().click(atPoint(115,13));
        Formulario().inputChars(dpString("Responsavel"));
        BotaoGravar().click(atPoint(42,8));
    }
}
```

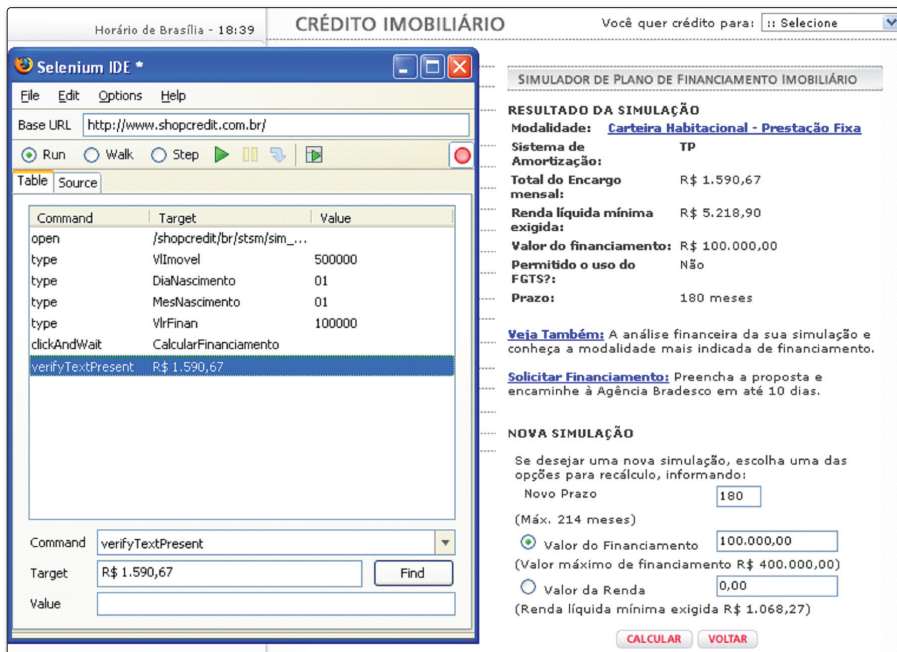


Figura 5. Exemplo de teste automatizado dirigido à palavra-chave.

(Type), clique no botão (ClickAndWait) e, por fim, a validação do resultado esperado (VerifyTestPresent).

Durante a execução dos testes, o Selenium IDE traduz as palavras-chaves em alto nível em ações para interagir com os elementos da interface gráfica do website (ou aplicação WEB).

É oportuno lembrar que esta abordagem não requer modificações na aplicação para criar os testes automatizados. Também não é necessário tornar a aplicação mais fácil de testar (testabilidade) porque os testes se baseiam na mesma interface utilizada pelos usuários.

O alto nível de abstração que descomplica a criação dos testes automatizados, até mesmo para usuários finais sem conhecimentos técnicos é, sem dúvida, a principal vantagem dos testes dirigidos à palavra-chave.

Por outro lado, essa abordagem não é muito robusta para a realização de testes que necessitem de laços de repetições e condições complexas. Além disso, nesta abordagem existe uma forte dependência da estabilidade da interface gráfica. Se a interface gráfica mudar, os testes falham.

### Testes automatizados baseados na linha de comando (Command Line Interface - CLI)

Uma Interface de Linha de Comando (Command Line Interface - CLI) fornece um mecanismo no qual o usuário pode

interagir com a aplicação por meio de um *prompt* ou *shell* do sistema operacional.

Via de regra, a lógica de negócio da aplicação pode ser exercitada por meio da execução de um conjunto de comandos e parâmetros pré-determinados. A CLI interpreta os comandos e parâmetros, executa a função selecionada e apresenta o resultado.

O objetivo da CLI é fornecer uma interface para o mundo exterior que não seja dependente da interface gráfica da aplicação. A automação de testes baseada na linha de comando faz uso dessa característica para orquestrar a execução dos testes.

Dessa forma, é possível criar *scripts shell* ou *batch* para exercitar algumas funcionalidades da aplicação sem que seja necessário utilizar uma interface gráfica.

A título de exemplo, observe no código abaixo, como é possível realizar testes automatizados de impressão do Microsoft Word por meio da linha de comando:

```
FOR %%c in (C:\*.doc) DO
winword.exe %%c /q /n /mFilePrintDefault /
mFileExit
```

Apesar de simples, o exemplo anterior é bastante interessante. Basicamente, o *script batch* varre todos os arquivos \*.doc do diretório. Para cada arquivo, o Word é aberto (instanciado sem interface gráfica), o arquivo é impresso e então o Word é fechado.

Com algumas poucas modificações, a impressão poderia ser redirecionada a um arquivo ao invés da impressora e podíamos

executar algum outro programa para comparar os arquivos gerados contra arquivos pré-definidos (gerados previamente para serem usados como base de comparação).

A principal vantagem dessa abordagem é a baixa exigência de modificações na aplicação para expor uma interface de linha de comando para o mundo exterior. Normalmente, essas modificações são simples e exigem pouca intervenção no código da aplicação.

Por outro lado, as interfaces de linha de comando são pouco flexíveis, principalmente quando é necessário passar parâmetros complexos para executar a funcionalidade.

Além disso, *scripts shell* ou *batch* não são muito robustos para capturar e manipular os resultados das operações, ou até mesmo, para realizar testes que necessitem de laços de repetições e condições complexas.

### Testes automatizados baseados em API (Application Programming Interface)

Uma API (Application Programming Interface ou Interface de Programação de Aplicativos) representa um conjunto de operações expostas por uma aplicação a fim de permitir que outras aplicações possam acessar ou consumir as suas funcionalidades.

O objetivo de uma API é fornecer uma interface para o mundo exterior que não seja dependente da interface gráfica da aplicação. A automação de testes baseada na API faz uso dessa característica para orquestrar a execução dos testes.

Para ilustrar um exemplo prático, observe na **Listagem 3** como é possível realizar testes automatizados baseados na API exposta pelo Microsoft Word.

Perceba no exemplo que o Word expõe via API praticamente todas as suas funcionalidades. Neste exemplo, foi possível instanciar o Word, abrir um documento, digitar um texto, mudar a formatação do texto, abrir a janela para localizar um texto e por fim finalizar o Word.

Em geral as APIs oferecem um mecanismo mais robusto para exercitar as funcionalidades da aplicação, se compararmos com os testes automatizados baseados na linha de comando.

Uma API pode expor virtualmente todas as funcionalidades de uma aplicação. Em termos práticos, isto significa que a principal vantagem dessa abordagem é



a criação de testes complexos e robustos por meio de uma linguagem de programação de alto nível (Java, C++, C#, etc).

Por esta razão, os testes automatizados baseados em API viabilizam a criação de testes automatizados com maior profundidade e amplitude, sem que seja necessário interagir com a interface gráfica da aplicação.

Os testes automatizados baseados em API representam a evolução natural dos testes automatizados baseados na linha de comando. No entanto, a robustez e a flexibilidade oferecidas por esta abordagem exigem grandes modificações no código da aplicação para criar e expor as APIs ao mundo exterior.

### Test Harness

O Test Harness é um tipo de automação de testes baseado na Lógica de Negócio que prega o uso racional e inteligente da automação. O Test Harness pode ser implementado por meio de um pequeno programa construído para testar uma API, uma interface de linha de comando, ou até mesmo ganchos “Hooks” projetados na aplicação para este fim.

Um ganho, ou *Hook*, é uma funcionalidade ou comportamento da aplicação que não tem valor do ponto de vista do usuário final. Normalmente não é documentado nem acessível pela interface gráfica. Mas, no entanto, é um recurso que tem a finalidade de tornar a aplicação mais fácil de testar (testabilidade), tanto do ponto de vista do teste manual, quanto do teste automatizado.

Nesta abordagem, não importa o meio no qual o teste será realizado (contanto que não ocorra interação com a interface gráfica). O objetivo é exercitar as funcionalidades críticas da aplicação que exigem dezenas e milhares de cálculos ou repetições virtualmente impossíveis (ou demoradas) de serem testadas por meios normais.

Para ilustrar, consideremos um exemplo prático e simples. Suponha que seja necessário simular 10 mil tipos de vendas diferentes numa aplicação hipotética. As variáveis destas vendas incluem centenas de tipos de produtos diferentes, diversas formas de pagamentos, alíquotas diferentes e combinações entre todas essas variáveis.

Entretanto, o objetivo do teste não é a realização das vendas, mas exercitar e validar o mecanismo (funcionalidade) que gera os

arquivos com os dados dos boletos de cobrança. Os bancos usam o padrão FEBRABAN CNAB 400 ou 240 para intercambiar informações digitalmente entre o sistema de informática do banco e o do cliente (ou seja, a aplicação que será testada).

A realização dos testes desse cenário hipotético por meio de uma abordagem de testes manuais seria enfadonha (repetitiva), demorada e suscetível a erros. A realização dos testes deste cenário por meio da automação baseada na interface gráfica também seria muito demorada. Tanto para gravar e criar os testes quanto para reproduzi-los.

Além disso, as ferramentas de automação de testes baseadas na interface gráfica nem sempre oferecem recursos para o desenvolvimento de um mecanismo robusto para realizar validações complexas, tal qual as validações que deverão ser realizadas nos arquivos bancários do nosso exemplo.

Por outro lado, um Test Harness é a combinação da criação de um pequeno programa utilizando uma linguagem de programação robusta (Java, C++, C#, etc) e a decisão inteligente de criar somente a API ou interface de linha de comando necessária para a realização do teste.

Assim, para executar os testes do cenário exposto anteriormente por meio de um Test Harness, será necessário primeiro criar uma API, *Hook*, ou interface de linha de comando para expor ao mundo exterior a funcionalidade que cadastra as vendas e a funcionalidade que gera os arquivos com os dados dos boletos de cobrança.

Também será necessário criar o Test Harness propriamente dito, ou seja, um pequeno programa desenvolvido em uma linguagem de programação (Java, C++, C#, etc) cujo único objetivo é realizar o teste em questão (por meio da API, *Hook*, ou interface de linha de comando criada para este fim).

Uma vez que esta infra-estrutura estiver pronta, será possível executar os testes baseados num Test Harness. Basicamente, o Test Harness deverá ler um arquivo de entrada contendo todos os dados das vendas. Com base nesses dados, o Test Harness exercitará as funcionalidades da aplicação por meio de uma API, ou outro mecanismo, sem interagir com a interface gráfica.

A aplicação, por sua vez, irá gerar os arquivos com os dados dos boletos de cobrança com base nas informações e ações realizadas pelo Test Harness. Por fim, o Test Harness deverá comparar os arquivos gerados pela aplicação contra arquivos pré-definidos (gerados previamente para serem usados como base de comparação) e apresentar os resultados indicando os testes que foram executados com sucesso ou falharam (**Figura 6**).

Dessa forma, conforme mencionamos anteriormente, este tipo de automação de testes prega o uso racional e inteligente da automação. Em termos práticos, isto significa que os esforços e as modificações requeridas para utilizar o Test Harness são muito menores se compararmos com os testes automatizados baseados em API.

Além disso, normalmente a velocidade da execução de um Test Harness é substancialmente maior do que outras abordagens, em virtude de que o Test Harness é um pequeno programa especializado apenas em uma única função. Isto torna o Test Harness a melhor opção para a realização de testes que exigem centenas de milhares de repetições, testes de funcionalidades que realizam cálculos complexos e assim por diante.

### Conclusão

Neste artigo notamos a diversidade das abordagens para a automação de testes. Em resumo, não existe uma solução que

#### Listagem 3. Teste automatizado baseado numa API utilizando uma linguagem de script

```
Set appWord = Wscript.CreateObject("Word.Application")
appWord.Visible = TRUE
appWord.Documents.Open("C:\Teste.doc")
appWord.Selection.TypeText Text:="teste de software"
appWord.Selection.HomeKey Unit:= wdLine
appWord.Selection.Range.HighlightColorIndex = wdYellow
appWord.Selection.Font.Color = wdColorRed
appWord.Selection.Font.Bold = wdToggle
appWord.Selection.Find.ClearFormatting
With appWord.Selection.Find
    .Text = "teste"
    .Forward = True
End With
appWord.Selection.Find.Execute
appWord.Quit
Set appWord = Nothing
```

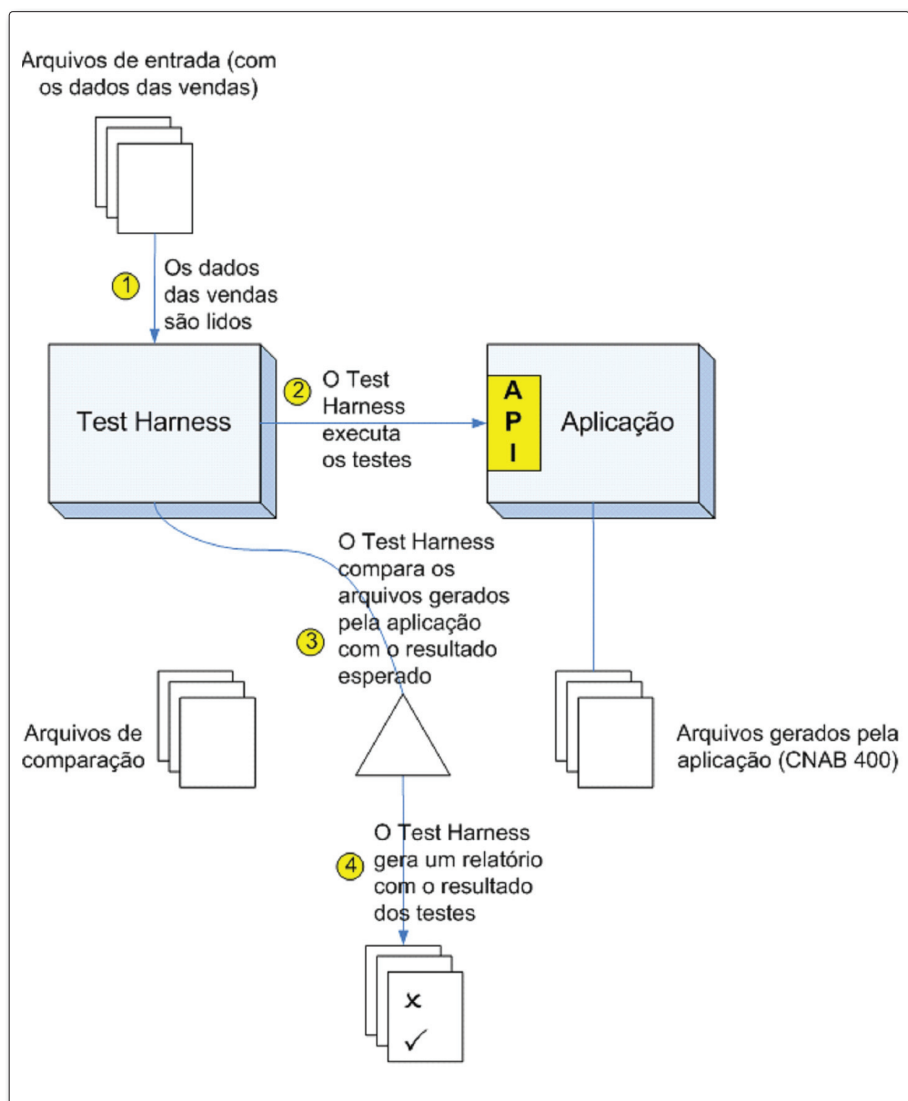


Figura 6. Exemplo das etapas de execução de um Test Harness.

sirva para todas as situações. É uma prática comum adotarmos soluções híbridas para atender necessidades diferentes de automação sem, no entanto, eliminar os testes manuais por completo. Testes manuais e automatizados são abordagens de testes diferentes que se reforçam mutuamente.

A Tabela 2 apresenta o sumário dos tipos de testes automatizados apresentados neste artigo descrevendo as vantagens e desvantagens de cada um deles. ●

#### Links

##### Sítio do Selenium

<http://www.openqa.org/selenium/>

##### IBM - Rational Functional Tester

<http://www-306.ibm.com/software/awdtools/tester/functional/>

##### Evaluating Functional Testing Solutions: The Forrester Wave™ Q2 2006

[www.forrester.com/Events/Content/0,5180,-1403,00.ppt](http://www.forrester.com/Events/Content/0,5180,-1403,00.ppt)

##### FEBRABAN

<http://www.febraban.org.br/>

##### Automação e Gerenciamento de Testes: Aumentando a Produtividade com as Principais Soluções Open Source e Gratuitas

<http://www.linhadecodigo.com.br/EBook.aspx?id=2951>

#### Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



Tipo de Teste Automatizado	Vantagens	Desvantagens
Gravação e reprodução	Rapidez e facilidade para criar scripts de teste;	Sem planejamento adequado os scripts ficam difíceis de dar manutenção.
	Não é necessário modificar a aplicação ou torna-la mais fácil de testar;	Forte dependência da estabilidade da interface gráfica;
Dirigido a dados	Alta reutilização dos scripts de teste;	Não é possível criar scripts de teste complexos;
	Não é necessário modificar a aplicação ou torna-la mais fácil de testar;	Forte dependência da estabilidade da interface gráfica;
Dirigido à palavra-chave	O usuário final pode criar os scripts testes;	Não é possível criar scripts de teste complexos;
	Não é necessário modificar a aplicação ou torna-la mais fácil de testar;	Forte dependência da estabilidade da interface gráfica;
Baseado na linha de comando	Baixa exigência de modificações na aplicação;	Pouca flexibilidade;
	Independente da interface gráfica;	Não é possível criar scripts de teste complexos;
Baseado em API	Possibilidade de criar scripts de teste robustos e complexos;	Alta exigência de modificações na aplicação;
	Independente da interface gráfica;	
Test Harness	Uso racional e inteligente da automação;	Alta exigência de modificações na aplicação e criação do Test Harness;
	Possibilidade de criar scripts de teste robustos e complexos;	Existem poucas ferramentas/frameworks que suportam essa abordagem (normalmente é necessário criar soluções caseiras).
	Melhor performance para testes que exigem muitas repetições e cálculos complexos;	

Tabela 2. Sumário das vantagens e desvantagens dos tipos de automação de testes.

# webdays developers 2008

**100%**  
**de aproveitamento!**  
22 e 23 de agosto – SP

- Conheça as principais novidades das tecnologias **Java e .NET**
- Palestras sobre **Boas práticas e Arquitetura**
- Certificado de Participação
- **100% de aproveitamento** - Apesar das palestras acontecerem simultaneamente, não haverá perda de conteúdo. No dia do evento você receberá um DVD com o conteúdo completo de todas as palestras em formato de vídeo-aula!

## Principais Temas:

**São + de 46 horas  
de conteúdo.**

### Java

Ajax  
Java FX  
Grails  
Android  
Groovy  
Jboss Seam  
JSF  
RIA (Flex, etc)  
Best practices em java

### .NET

Linq  
WPF  
Silverlight 2.0  
ASP.NET 3.5  
Ajax  
ASP.NET MVC  
Visual Studio 2008  
Novidades C# 3.0 e VB.NET 9.0  
Best practices em .NET

### Extras

SOA  
WEB 2.0  
WEB 3.0  
Arquitetura Web  
Scrum  
Microsoft XNA  
Ruby on Rails  
Domain Driven Design  
Mashups

**Inscriva-se já!**

Maiores informações:  
[www.devmedia.com.br/webdays2008](http://www.devmedia.com.br/webdays2008)  
evento@devmedia.com.br  
(21) 3382-5025

Realização



**DevMedia**  
group

# Chegou o Sistema de Créditos DevMedia

Agora o **conteúdo completo** do nosso  
site está **ao seu alcance!**



2.000 vídeos

A partir de agora todas as **2000 vídeo-aulas** do site DevMedia podem ser compradas individualmente.

**Economia** - Você não precisa mais assinar oito produtos diferentes para ter acesso ao conteúdo completo do site!

Todas as vídeos que você sempre quis ver a partir **R\$ 0,75!**

Saiba mais sobre o Sistema de Créditos!