

engenharia  
de software

magazine

Edição 63 :: Ano 5



DEV MEDIA

# GESTÃO DE CONHECIMENTO

Tendências e influências nos softwares modernos e os softwares de GC nas organizações

Requisitos  
Gerencie requisitos com MPS.BR

Requisitos  
Gestão de riscos nos requisitos

Desenvolvimento  
Arquiteturas de sistemas Web 3.0



**Processo**  
Gerência de variabilidades em linhas de processos

**Agilidade**  
Responsabilidades de uma equipe Scrum

**BABOK**  
Introdução ao guia de conhecimento de análise de negócio

# Do Instituto Infnet para as melhores empresas do mercado.

## PÓS-GRADUAÇÃO

MIT em Engenharia de Software com Java

MIT em Engenharia de Software com .NET

## GRADUAÇÃO

Análise e Desenvolvimento de Sistemas

Engenharia de Computação

## FORMAÇÕES

Desenvolvedor Java

Desenvolvedor Java: Sistemas Distribuídos

Desenvolvedor Web .NET

MCITP Server Administrator

SQL Server

Aulas 100% práticas, corpo docente atuante no mercado, a mais atualizada biblioteca de TI do Rio, laboratórios com tecnologia de ponta e sala de estudos e exames.

**Faça Infnet e seja um profissional valorizado.**

“ Entrei na Graduação de ADS em março de 2009 (...). Hoje já sou Desenvolvedor Java Jr. de uma das maiores multinacionais de TI do mundo. ”

**Rafael Santos**

Aluno da Graduação de ADS - 2010.1

Acesse nosso site e conheça todos os nossos  
programas: [www.infnet.edu.br/esti](http://www.infnet.edu.br/esti)

ESCOLA SUPERIOR DA TECNOLOGIA DA INFORMAÇÃO  
[www.infnet.edu.br](http://www.infnet.edu.br) | 21 2122-8800





Ano 5 - 63ª Edição - 2013

#### Corpo Editorial

##### Editor

Rodrigo Oliveira Spínola

##### Colaboradores

Marco Antônio Pereira Araújo  
Eduardo Oliveira Spínola

##### Consultor Técnico

Daniella Costa

##### Jornalista Responsável

Kaline Dolabella - JP24185

##### Na Web

<http://www.devmedia.com.br/revista-engenharia-de-software-magazine>

magazine



#### **Rodrigo Oliveira Spínola**

[rodrigo.devmedia@gmail.com](mailto:rodrigo.devmedia@gmail.com)

Doutor e Mestre em Engenharia de Software pela COPPE/UFRJ. Realizou seu Pós-Doutorado na University of Maryland Baltimore County e Centro Fraunhofer nos Estados Unidos. Atualmente é Professor Titular do Programa de Pós-Graduação em Sistemas e Computação da Universidade Salvador - UNIFACS.



#### **Marco Antônio Pereira Araújo**

[maraajo@devmedia.com.br](mailto:maraajo@devmedia.com.br)

Doutor 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 e Coordenador do curso de Bacharelado em Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora, Professor e Diretor do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Fundação Educacional D. André Arcorverde, Analista de Sistemas da Prefeitura de Juiz de Fora, Colaborador da Engenharia de Software Magazine.



#### **Eduardo Oliveira Spínola**

[eduspinola@gmail.com](mailto:eduspinola@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).

#### 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:

[\(21\) 3382-5038](http://www.devmedia.com.br/central)

#### Publicidade

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

[publicidade@devmedia.com.br](mailto: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 à 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 ES Magazine, entre em contato com os editores, informando o título e mini-resumo do tema que você gostaria de publicar.



## Conhecimento faz diferença!

**Agilidade:** Acompanhamento de projetos ágeis distribuído através do Daily Meeting

**Processo:** Negociação de contratos em projeto

**enGENHARIA de software** magazine

**Edição 29 :: Ano 3**

**SOA**  
Processo e levantamento de requisitos de negócios – Parte 2

**Qualidade de Software**  
Definição, características e importância

**Automação de Testes**

**Cuidados a serem tomados na implantação**

**Projetos**

**Aulas de**

**+ de 290 vídeos para assinantes**

**Aulas desta edição:**

- Estratégia de Teste Funcional baseada em Casos de Uso – Partes 5 a 9
- Executar testes funcionais com Hudson e Selenium RC
- Importância da comunicação no processo de software

**Teste**  
Execute testes funcionais com Hudson e Selenium RC

**Processo**  
A importância da comunicação no processo de software

**ISSN 1678-2227-1**

Faça já sua assinatura digital! | [www.devmedia.com.br/es](http://www.devmedia.com.br/es)

## Faça um upgrade em sua carreira

Em um mercado cada vez mais focado em qualidade, ter conhecimentos aprofundados sobre requisitos, metodologia, análises, testes, entre outros, pode ser a diferença entre conquistar ou não uma boa posição profissional. Sabendo disso a DevMedia lançou uma publicação totalmente especializada em Engenharia de Software. Todos os meses você pode encontrar artigos sobre Metodologias Ágeis; Metodologias tradicionais (document driven); ALM (application lifecycle); SOA (aplicações orientadas a serviços); Análise de sistemas; Modelagem; Métricas; Orientação à Objetos; UML; testes e muito mais. **Assine Já!**



# Sumário

Conteúdo sobre Agilidade

## 06 – Conhecendo as responsabilidades de uma equipe Scrum

[ Anderson Dias Ribeiro ]

Conteúdo sobre Novidades

## 11 – O MPS.BR em sua revisão de 2012

[ Márcio Assis Miranda ]

Artigo no estilo prático

## 17 – Gerenciando requisitos com MPS.BR

[ Ivânia Ramos dos Santos e Giovanni De Bona ]

Conteúdo sobre Boas Práticas

## 24 – Gestão de riscos nos requisitos

[ Fábio Sarturi Prass ]

## 29 – A gestão de conhecimento e a TI – Parte 1

[ Diógenes Firmiano ]

Artigo no estilo Prático

## 38 – Gerência de variabilidades em linhas de processos de software

[ Edson A. Oliveira Junior, Maicon Giovane Pazin e Sandra Ferrari ]

Conteúdo sobre Boas Práticas

## 45 – Introdução ao guia de conhecimento de análise de negócio

[ Ivânia Ramos dos Santos e Fabio Pereira dos Santos ]

Artigo no estilo Prático

## 51 – Automação de testes funcionais para aplicações da plataforma Android

[ Jeanne de Castro Trovão e Arilo Claudio Dias Neto ]

## 59 – Arquiteturas de sistemas Web 3.0

[ Elaine G. M. de Figueiredo ]



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!

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

# VOÇÊ É ASSINANTE **MVP?**

ENTÃO PODE TER CERTEZA:  
**O MERCADO  
PREFERE VOCÊ!**



**DESENVOLVEDOR ATUALIZADO É O MAIS  
VALIOSO DO MERCADO!**

Muitos profissionais estão acomodados, não se atualizam, não leem livros e não vão a eventos. Acham que é a empresa que deve mantê-los atualizados. Gastam dinheiro com tudo, mas acham caro investir menos de R\$60,00 por mês em suas carreiras.

Mas você não! Você é MVP! Parabéns!

TENHA ACESSO A:

+ DE **260** CURSOS ONLINE

**09** REVISTAS MENSais

**7.850** VÍDEO-AULAS

**POR APENAS 59,90**  
MENSais



QUEM TEM ESTÁ TRANQUILO.



Acesse: [www.devmedia.com.br/mvp](http://www.devmedia.com.br/mvp)

Nesta seção você encontra artigos voltados para as práticas e métodos ágeis.

## Conhecendo as responsabilidades de uma equipe Scrum

Gerente de projeto VS Scrum Master



**Anderson Dias Ribeiro**

[andersonr@gmail.com](mailto:andersondr@gmail.com)

Graduado em Tecnologia da Informação e pós-graduado em Engenharia de Software. Atua como analista de sistemas há 10 anos e há 15 anos no mercado de TI, participou do planejamento e desenvolvimento de diversos sistemas nas plataformas WEB, Java, DotNET, dentre outros. Atualmente é Analista de Sistemas Sênior da Nova Tecnologia, prestando serviço a instituições financeiras de grande porte no ramo de Seguro. Também atuou por 7 anos junto à MRS Logística, como Analista de Sistemas e especialista em tecnologias Oracle como SOA e BI e como pesquisador da plataforma SOA/Java e gerenciamento de projetos SGP, baseados em PMBOK.

**A**tualmente as organizações que buscam um diferencial e uma qualidade real de seus produtos no mercado vêm buscando a cada dia melhorar e aperfeiçoar suas metodologias e processos, principalmente quando estamos falando de desenvolvimento de Software, um produto totalmente abstrato e intangível, que por sua vez possui um grande valor estratégico para as empresas que buscam alto desempenho, produtividade e como consequência obter maior lucratividade.

Ao longo do tempo, diversas metodologias foram aplicadas, umas mais bem sucedidas outras menos, se destacando o PMBOK (Project Management Body Of Knowledge) do PMI (Project Management Institute) como referência consagrada pela engenharia de software moderna e por seus usuários nos mais diversos ramos de atuação. No entanto, quando falamos em metodologias ágeis, precisamos ir além dos conceitos tradicionais, avaliar as reais necessidades que um projeto vai demandar de acordo com

### **Porque esse artigo é útil:**

A atuação dos profissionais e suas responsabilidades em um projeto, seja ele orientado por uma metodologia tradicional, quanto por paradigmas ágeis, devem estar bem claras e definidas para que conflitos desta natureza não venham a intervir no sucesso do projeto, que é o objetivo de todos os envolvidos.

Neste contexto, este artigo, além de conceituar e exemplificar as características de um projeto ágil e como deve ser composta uma equipe de maneira eficiente, busca detalhar as responsabilidades e papéis nos quais estão envolvidos os profissionais de gerência de projetos bem como o Scrum-Master, que tem grande importância no âmbito de um projeto ágil, assim como o Product Owner, o Time e os demais interessados no sucesso de um projeto.

Este artigo busca esclarecer questões e conflitos que possam surgir durante um projeto que aplique a metodologia ágil Scrum, no que se refere principalmente aos pontos de atuação do gerente de projetos e do ScrumMaster, visando dirimir quaisquer dúvidas que possam impactar no projeto.

suas características essenciais, podendo ser necessário aplicar conceitos de gerenciamento e desenvolvimento ágeis, ou até mesmo utilizar uma abordagem mais flexível, ajustando os processos às peculiaridades do projeto.

Para atender a esta crescente demanda, a metodologia largamente utilizada é o Scrum, sendo um conjunto de ferramentas que nada mais é que um processo interativo e incremental, segmentado em diversas tarefas, que pode ser utilizado para o gerenciamento e desenvolvimento de quaisquer tipos de produto aplicado a quaisquer tipos de projeto, sendo ele de software ou não. Podemos dizer que o Scrum é mais um framework que uma metodologia, talvez mais a atitude da equipe envolvida do que um processo propriamente dito, ficando claro que o comprometimento da equipe é a base do sucesso do projeto.

## Projetos e gerenciamento

Neste exato momento inúmeros projetos estão com seus prazos atrasados, possivelmente a grande maioria deles, mas quais as causas principais que ocasionam isto? Quais os principais fatores que estão envolvidos? São apenas questões técnicas, ou também políticas e culturais das organizações? E o que podemos fazer para sanar este problema, ou ao menos amenizá-lo?

Ao longo de vários anos de experiência no mercado de TI, pude observar diferentes políticas e metodologias utilizadas pelas organizações para controlar seus projetos, tais como PRINCE2 (*PRojects IN Controlled Environments, version 2*), IPMA (*International Project Management Association*), PMBOK, que citamos anteriormente como sendo a mais consagrada delas na atualidade. No entanto, infelizmente, muitas destas organizações não utilizam nenhuma metodologia, aplicando apenas alguns processos aos quais julga necessário, esquecendo-se dos fundamentos de qualquer engenharia, em que os princípios que a regem são a organização e o planejamento seguidos de métricas que possibilitem uma projeção o mais precisa possível dos objetivos a serem alcançados. Para isto, os guias já citados disponibilizam inúmeras técnicas e ferramentas.

A seguir citarei alguns fatores que considero fundamentais e que contribuem para o sucesso de um projeto.

## Planejamento da qualidade

Por meio da participação no planejamento global do projeto, o Planejamento da Qualidade tem como principal objetivo identificar os padrões de qualidade relevantes a este projeto e identificar a forma como estes devem ser satisfeitos.

Deve-se ter em mente que neste escopo entram tanto os padrões de qualidade que poderíamos dizer “voluntários”, quanto os padrões de qualidade definidos por normas e legislação, que virão a ser requeridos conforme a finalidade ou produto final do projeto e questões do ambiente interno e externo das organizações envolvidas. Assim, o processo de planejamento explicita as ações dos processos de Garantia e de Controle de qualidade.

Desta forma, estes processos visam garantir que o projeto irá satisfazer as necessidades para as quais ele foi empreendido, e deve ocorrer pelo menos uma vez a cada fase do projeto direcionado para a gerência do projeto e para o produto do projeto.

## Qualidade

Quando falamos do produto software, qualidade se traduz na satisfação dos objetivos pelos quais aquele produto foi projetado e ainda que o funcionamento esperado do produto esteja isento de imprevistos que possam impactar no seu desempenho em geral. Segundo o PMI, define-se por qualidade a totalidade das características de uma entidade que afetam sua habilidade em satisfazer necessidades declaradas ou implícitas.

Os princípios básicos da qualidade podem resumir-se em:

1. Fazer certo da primeira vez, economiza tempo e dinheiro;
2. Qualidade é um processo preventivo;
3. Qualidade é incorporada aos produtos como resultado da atenção dedicada às necessidades dos clientes;
4. Qualidade é responsabilidade de todos os envolvidos;
5. Qualidade é um processo de melhoria contínua (lições aprendidas usadas para otimizar produtos e serviços).

## Gerenciamento da Qualidade

O gerenciamento da qualidade consiste em uso de ferramentas e técnicas, as quais possibilitam alcançar determinado objetivo, citamos algumas mais interessantes:

- **Análise Custo/Benefício:**

- Usar medidas financeiras como retorno de investimento ou período de reembolso para avaliar a qualidade das alternativas identificadas;
- O principal benefício de atender aos requisitos de qualidade é o menor retrabalho ou maior produtividade;
- O principal custo são os gastos associados com as atividades de gerência da qualidade.

- **Benchmarking:**

- Comparar as práticas reais ou planejadas do projeto com as de outros projetos para gerar ideias de melhoria e fornecer um padrão pelo qual se possa medir o desempenho.

- **Fluxogramas:**

- Diagramas que mostram como os vários elementos do sistema se relacionam e sejam adequados para análise de problemas potenciais e determinação de padrões de qualidade como, por exemplo, os diagrama causa e efeito e os fluxogramas de processos.

É um conhecido axioma da qualidade que os benefícios superam os custos.

## Recursos humanos

Os recursos humanos são de suma importância em projeto, tanto que geralmente as organizações possuem um departamento específico para tal. Cabendo também ao gerente de projetos garantir o efetivo uso das pessoas.

## Comunicação

A maior ameaça para os projetos é a falha de comunicação. Esta deve ser a principal habilidade para um bom gerente de projetos, pois comprovadamente é observado que a grande maioria dos casos, 90% do tempo do gerente é gasto com as comunicações.

## Aquisição

O papel do gerente de projetos nas aquisições deve ser atribuído previamente, que por sua vez providencia a análise de riscos antes mesmo das contratações, evitando e se antecipando a eventuais problemas neste sentido. Também é papel do gerente de projetos:

- Alocar os riscos no contrato;
- Ajudar na elaboração do contrato para atender às necessidades;
- Colocar o cronograma de acordo com o plano do projeto;
- Participar da fase de negociação do contrato;
- Acompanhar para garantir que o trabalho seja feito;
- Proteger o relacionamento com o contratante.

## Papéis e responsabilidades no Scrum

Ao descrevermos anteriormente alguns requisitos básicos, os quais consideraram de caráter essencial para o sucesso de um projeto, abordamos as principais características de metodologias tradicionais para o gerenciamento de projetos, em que o papel do gerente é fortemente destacado.

Como já definimos anteriormente, o Scrum é um conjunto de ferramentas, constituindo-se por um processo interativo e incremental, segmentado em diversas tarefas, que geralmente é utilizado para o gerenciamento e desenvolvimento de quaisquer tipos de produto aplicado a quaisquer tipos de projeto, sendo ele de Software ou não. Considerando este paradigma ágil aplicado ao gerenciamento de projetos, o papel do gerente ou “Product Owner” como é mais conhecido nesta metodologia, deve ser claramente definido, bem como do ScrumMaster e o Time Scrum, todos de suma importância para o sucesso de um projeto.

Ilustramos na **Figura 1** os membros que compõem uma equipe de Scrum.

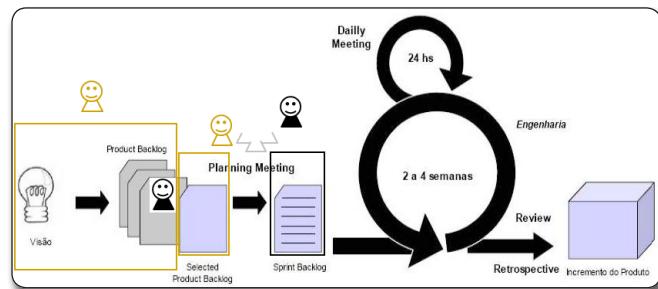


**Figura 1.** Membros da Equipe

Na **Figura 2** temos a estrutura básica dos processos envolvidos no Scrum, bem como a definição dos papéis e responsabilidades dos membros da equipe.

O Product Owner ou “PO” define a Visão do Produto. Esta Visão é o que representa sua necessidade, é o que deve ser satisfeita ao fim do projeto. Para definir esta Visão o PO colhe informações junto a clientes, usuários final, time,

gerentes, stakeholders, executivos, etc. O PO cria uma lista inicial de necessidades que precisam ser produzidas para que a Visão do projeto seja bem sucedida. Esta lista de necessidades é chamada de Product Backlog e o ScrumMaster deve auxiliar o PO na elaboração desta lista. No início de cada iteração (Sprint) o time deve se reunir para realizar a Planning Meeting. Nesta reunião o time realizará o planejamento do que será entregue ao final do ciclo da Sprint (que deve ser de 2 a 4 semanas).

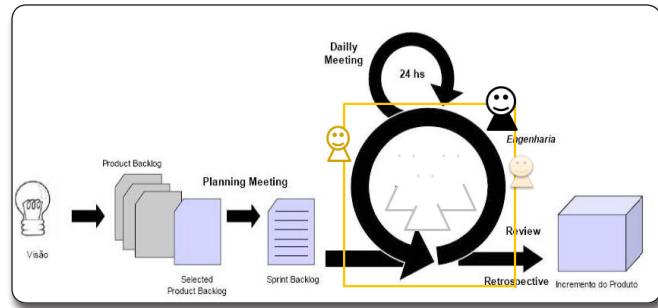


**Figura 2.** Estrutura do Scrum/Responsabilidades

Na primeira parte da Planning Meeting o PO deverá definir a meta da Sprint e falar para o time sobre os Itens mais prioritários do Product Backlog. Desta forma, o time deve estimar os itens em tamanho (caso ainda não estejam estimados) e selecionar o que acredita que possa ser feito durante a Sprint. Essa lista selecionada chama-se Selected Product Backlog. O facilitador desta reunião é o ScrumMaster.

Na segunda parte da Planning Meeting o time deverá colher mais detalhes dos itens do Selected Product Backlog e decompô-los em tarefas, gerando assim o Sprint Backlog, para isso pode ser necessária a ajuda de Especialistas. Após a decomposição, cada membro do time deve selecionar as tarefas que deseja executar durante a Sprint (sempre negociando com o time) e estimá-las em horas. Neste caso, o facilitador desta reunião é o ScrumMaster. É importante ressaltarmos que durante a execução da Sprint, valem as práticas de engenharia de software previamente definidas para o projeto.

O ScrumMaster, através da sua capacidade de liderança e colaboração, facilita o trabalho do time removendo os impedimentos encontrados e garantindo a boa aplicação do Scrum. Veja a **Figura 3**.



**Figura 3.** Estrutura do Scrum – Sprint

Deve ser observado durante a Sprint se o time sente a necessidade de consultar especialistas ou mesmo o Product Owner que também tem esta atribuição de estar atento às necessidades e dificuldades que posam surgir e impactar em algum ponto do projeto. Diariamente o time deve realizar uma reunião de 15 minutos (Daily Meeting) na qual cada membro deve responder algumas questões, como:

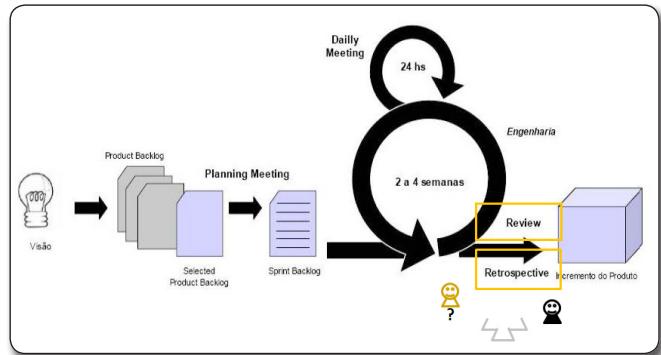
- O que fiz desde a última reunião?
- O que pretendo fazer até a próxima?
- Tive (estou tendo) algum impedimento?

Através desta reunião o time ganha visibilidade de como está o caminho para a meta e planeja o dia seguinte de trabalho e o ScrumMaster novamente é o facilitador, e deve lembrar-se que a reunião é para o time e não para ele.

Ao final da execução da Sprint deve ser realizada a Review Meeting, mais uma reunião que tem como propósito apresentar o que foi feito para o Product Owner e eventuais convidados, em que o time é quem realiza a apresentação, e o Product Owner avalia se a Meta da Sprint foi alcançada com sucesso ou não, faz anotações que poderão se transformar em novos itens para o Product Backlog. Observe a Figura 4.

Finalmente, a última etapa de um Sprint Scrum, chamada de Retrospectiva, consiste em uma reunião de lições aprendidas, facilitada pelo ScrumMaster, na qual o time deve avaliar:

- O que foi bom na última Sprint?
- O que deve ser melhorado?
- Quem está no controle?



**Figura 4.** Estrutura do Scrum – Review e Retrospective

Esta reunião representa o espírito de Inspeção-Adaptação dentro do Scrum. É uma reunião do time, mas, caso todos os membros estejam de acordo o PO também pode participar.

### Conclusão

Diante do que foi discutido neste artigo, notamos que em uma metodologia de desenvolvimento ágil como o Scrum,

## CURSOS ONLINE

A Revista Clube Delphi oferece aos seus assinantes uma série de Cursos Online de alto padrão de qualidade.



### CONHEÇA ALGUNS DOS CURSOS:

- **Curso de Multicamadas com Delphi e DataSnap**
- **Delphi para Iniciantes**
- **Criando componente Boleto em Delphi**
- **Loja Virtual em Delphi Prism**

Para mais informações :

[www.devmedia.com.br/cursos/delphi](http://www.devmedia.com.br/cursos/delphi)  
(21) 3382-5038

suas peculiaridades influenciam diretamente nos papéis e responsabilidades dos membros da equipe, diferentemente de metodologias mais tradicionalistas.

Em uma visão macro do assunto, podemos concluir que as principais atividade dos membros da equipe Scrum, conforme explanamos ao longo deste artigo, são as seguintes:

- Product Owner:

- Responsável por garantir o ROI (Retorno sobre o Investimento);
- Responsável por conhecer as necessidades do(s) cliente(s);
- Integrador em ambientes com mais de um cliente.

- ScrumMaster:

- Responsável por remover os impedimentos do time;
- Responsável por garantir o uso de Scrum;
- Protege o time de interferências externas.

- Time:

- Definir metas das interações;
- Autogerenciamento;
- Produzir produto com qualidade e valor para o cliente.

Com isto, podemos observar as principais responsabilidades dos diferentes membros de uma equipe Scrum, reduzindo quaisquer dúvidas que possam impactar no projeto.

### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



### Links e Referências:

**A Guide to the Project Management Body of Knowledge (PMBOK® Guide)**

- Fourth Edition

[www.pmi.org/](http://www.pmi.org/)

**Scrum Guide 2011, escrito por Ken Schwaber e Jeff Sutherland**

[www.scrum.org/scrumguides](http://www.scrum.org/scrumguides)

**AMBLER, Scott W Imperfectly Agile: You Too Can Be Agile. Dr. Dobb's Portal. Setembro, 2006.**

[www.ddj.com/architect/192700252](http://www.ddj.com/architect/192700252)

**BECK K. (1999). Programação Extrema Explicada. Bookman, 1999.**

**BECK, K ; FOWLER, M. (2000). Planning extreme programming. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2000.**

<http://www.mip.sdu.dk/~brianj/Extreme%20Programming%20Explained%20-%20Kent%20Beck%3B%20Addison-Wesley,%201999.pdf>

**BECK, KENT et al. (2001). Agile Manifesto, 2001.**

<http://www.agilemanifesto.org>

**COCKBURN, A. (2000). Agile Software Development Draft version: 3b. Highsmith Series Editors, 2000.**

<http://zsiie.icis.pcz.pl/ksiazki/Agile%20Software%20Development.pdf>

**FOWLER, Martin. The New Methodology, 2005.**

[www.martinfowler.com/articles/newMethodology.html](http://martinfowler.com/articles/newMethodology.html)

**SCHWABER, Ken e BEEDLE, Mike (2002). Agile Software Development with SCRUM. Prentice Hall, PTR Upper Saddle River, NJ, USA, 2002.**

Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros

## O MPS.BR em sua revisão de 2012

### Melhoria de processos com MPS.BR

**N**o cenário atual em que as organizações de desenvolvimento de software e prestadoras de serviços se encontram, a busca pela qualidade dos produtos desenvolvidos e dos serviços prestados se torna cada vez mais frequente, seja para diminuir custos e aumentar a produtividade ou para conquistar novos mercados. Também houve um aumento considerável na dependência em serviços de suporte, e os diversos universos de tecnologia disponíveis, onde provedores de serviços lutam para manter altos níveis de serviços aos clientes. Com o trabalho de forma reativa, as empresas investem pouco tempo em planejamento, treinamento, análise crítica, investigação e no trabalho com seus clientes. O desenvolvimento e a melhoria dos processos de software e das práticas de serviços são pontos importantíssimos para um melhor desempenho, aumento da satisfação do cliente e para o alcance da lucratividade.

#### *Porque esse artigo é útil:*

Este artigo é útil para profissionais e empresas que buscam conhecer o programa de Melhoria de Processos de Software e Serviços e consequentemente aplicar as melhores práticas do modelo em seus processos e serviços, podendo futuramente ingressar em um programa de auditoria da qualidade. Para isso, ele demonstra, através de uma visão geral, como o modelo de qualidade MPS.BR (Melhoria de Processo de Software Brasileiro) pode contribuir no processo de desenvolvimento de software e na prestação de serviços. Serão apresentados os conceitos, características, objetivos, representações e alguns dados estatísticos referentes ao modelo.



**Márcio Assis Miranda**

*assismiranda@gmail.com*

Especialista em Melhoria de Processo de Software pela UFLA, Bacharel em Sistemas de Informação pelo UNILESTE. Atuante na área de TI, com mais de 10 anos de experiência profissional em gerencia de projetos, análise e desenvolvimento de sistemas, sendo os últimos anos dedicados a carreira acadêmica, à liderança de equipes e à gerência de projetos. Professor no UNILESTE e em algumas escolas técnicas da região do Vale do Aço, Minas Gerais.

A qualidade é um fator crítico tanto para as empresas desenvolvedoras de software quanto para as prestadoras de serviços. Essa qualidade do produto ou serviço é determinada principalmente pela qualidade dos processos utilizados em seu desenvolvimento e isso tem motivado as empresas a modificarem as suas estruturas organizacionais e processos produtivos, fugindo da visão tradicional e indo em direção a redes

de processos centralizados no cliente. A competitividade vem exigindo cada vez mais que as organizações estabeleçam conexões com essas redes e que crie elos essenciais na cadeia produtiva, inclusive atendendo padrões internacionais de qualidade.

Esta visão orientou a elaboração de diversos modelos que visam a avaliação e melhoria do processo de software e serviços, tais como o CMMI, MPS.BR, ISO/IEC 15504-5 entre outros. Dentre os modelos citados, destaca-se o modelo MPS.BR, pois o mesmo foi desenvolvido para a realidade do mercado brasileiro que é formado em sua maioria por empresas de pequeno e médio porte. Importante destacar que o modelo também atende empresas de grande porte, inclusive isso é uma realidade que será mostrada em alguns dados estatísticos apresentados mais adiante.

Serão apresentados os dois modelos de referência do MPS, MR-MPS-SW (Modelo de Referência MPS para Software) e MR-MPS-SV (Modelo de Referência MPS para Serviços), sendo o último incluído no MPS em 2012.

## Por que investir em melhoria de processos?

O Brasil possui um mercado de software que se encontra em constante expansão nos últimos anos. Se fizermos uma pesquisa sobre o assunto, provavelmente todas as empresas responderão que o que mais buscam são formas de melhorar a produtividade e qualidade, ou seja, como produzir ótimos produtos, numa escala de produtividade alta, custos baixos e pagando salários justos, inclusive baseados na competência dos profissionais.

Sabe-se que não só o mercado brasileiro, mas também o mercado internacional estão cada vez mais exigentes e intolerantes à variação de qualidade, prazo e custo. Sendo assim, como achar um ponto de equilíbrio? Qual fórmula adotar para atender essas necessidades?

Uma forma de obter alguma referência da qualidade do software que uma empresa produz é exigindo algum certificado de maturidade de processos, conforme modelos já citados na introdução desse artigo.

Segundo McConnell, empresas que investiram em capacitação e melhoria de processo de software obtiveram retorno de investimento da ordem de 5:1 e, em alguns casos, até 9:1.

Diversos estudos de caso, conduzidos durante mais de duas décadas, fornecem evidências comprovadas de que a melhoria do processo de desenvolvimento trazem diversos benefícios, sendo citados alguns a seguir:

- Aumento da qualidade do software produzido;
- Aumento da satisfação do cliente;
- Aumento da produtividade da empresa;
- Redução dos custos de desenvolvimento;
- Diminuição da rotatividade de pessoal;
- Aceleração da curva de aprendizado do desenvolvedor;
- Aumento da moral da equipe.

Alguns casos de sucesso, conforme o Portal SEI, são apresentados na Tabela 1.

Tópico	Melhorias
Custo	<ul style="list-style-type: none"> <li>• 33% de diminuição no custo médio de correção de defeitos (Boeing, Austrália);</li> <li>• 15% de diminuição nos custos de detecção e correção de defeitos (Lockheed Martin M&amp;DS);</li> <li>• Melhoria na precisão da estimativa de custos (Raytheon North Texas Software Engineering).</li> </ul>
Cronograma	<ul style="list-style-type: none"> <li>• Diminuição de desvios de custos e prazos em projetos para abaixo de 10% (Ci&amp;T, Brasil);</li> <li>• Aumento na previsibilidade e exatidão do cronograma de entrega (JP Morgan Chase).</li> </ul>
Qualidade	<ul style="list-style-type: none"> <li>• 50% de redução na taxa de defeitos ao chegar ao nível 3 de maturidade CMM (Ci&amp;T, Brasil);</li> <li>• Aumento na qualidade do código (Sanchez Computer Associates, Inc.);</li> <li>• Alcançada a meta de <math>20 \pm 5</math> defeitos por KLOC (Northrop Grumman IT).</li> </ul>
Produtividade	<ul style="list-style-type: none"> <li>• 30% de aumento na produtividade de software (Lockheed Martin M&amp;DS);</li> <li>• 25% de aumento de produtividade em 3 anos (Siemens Information Systems Ltd, Índia).</li> </ul>
Satisfação do Cliente	<ul style="list-style-type: none"> <li>• Recebimento de 98% de todas as premiações do cliente por desempenho (Northrop Grumman IT);</li> <li>• Classificado como “excepcional” em todas as categorias das avaliações de desempenho dos serviços (Northrop Grumman IT).</li> </ul>

**Tabela 1.** Casos de sucesso da Melhoria de Processos

Focando em processos, as organizações se tornam mais produtivas, produzem softwares com menos defeitos, lidam melhores com os riscos, melhoraram suas estimativas de prazos e custos, aumentam a moral da equipe de desenvolvimento e aumentam as chances de sucesso, principalmente em projetos grandes, que geralmente são mais críticos.

## MPS.BR

O MPS.BR é um programa mobilizador, de longo prazo, criando em dezembro de 2003, coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) que conta com apoio do Ministério da Ciência, Tecnologia e Inovação (MCTI), Financiadora de Estudos e Projetos (FINEP), Serviço Brasileiro de Apoio às Micros e Pequenas Empresas (SEBRAE) e do Banco Interamericano de Desenvolvimento (BID/FUMIN).

Ainda de acordo com o guia do MPS.BR, o programa conta com uma Unidade de Execução do Programa (UEP) e duas estruturas de apoio para a execução de suas atividades, o Fórum de Credenciamento e Controle (FCC) e a Equipe Técnica do Modelo (ETM). Por meio destas estruturas, o MPS.BR pode contar com a participação de representantes de universidades, instituições governamentais, centros de pesquisa e de organizações privadas, os quais contribuem com suas visões complementares que agregam valor e qualidade ao programa.

O modelo é baseado nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de software e serviços correlatos e também para a melhoria da qualidade e produtividade dos serviços prestados. Já o programa MPS.BR tem como meta definir e aprimorar um modelo de melhoria e avaliação de processo de

software e serviços que visa preferencialmente às micro, pequenas e médias empresas (mPME) de forma a atender as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software e às empresas prestadoras de serviços. O MPS estabelece dois modelos de referência de processos de software e serviços, e um processo/método de avaliação de processos. Esta estrutura fornece sustentação e garante que o modelo seja empregado de forma coerente com as suas definições. O modelo MPS estabelece também um modelo de negócio para apoiar a sua adoção pelas empresas desenvolvedoras de software e prestadores de serviços.

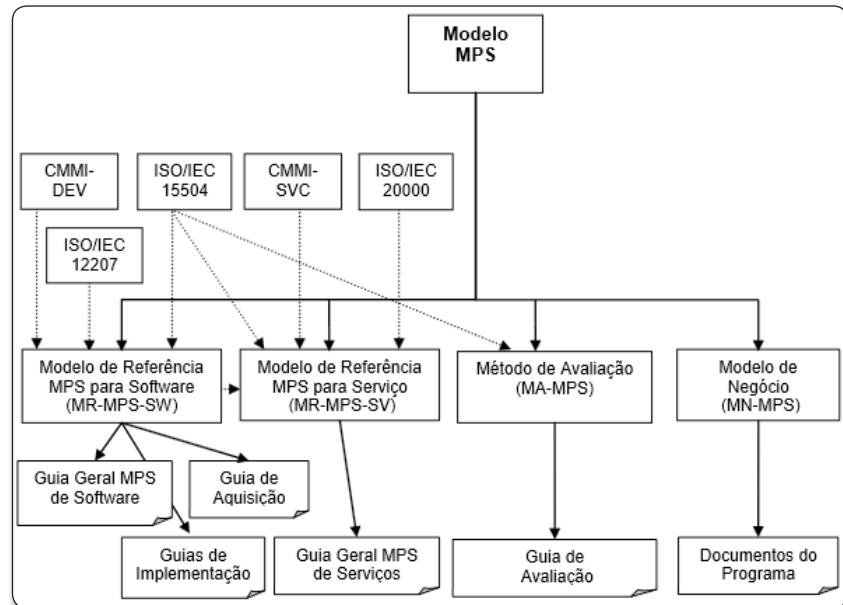
O modelo MPS está dividido em quatro componentes, como se pode observar na **Figura 1**: MR-MPS-SW, MR-MPS-SV, MA-MPS (Método de Avaliação) e MN-MPS (Modelo de Negócio). Cada componente é descrito por meio de guias e/ou documentos do Programa MPS.BR. Mais informações sobre isso podem ser encontradas na seção **Links**.

Como já foi citado anteriormente, o artigo dará destaque para o MR-MPS-SW e MR-MPS-SV.

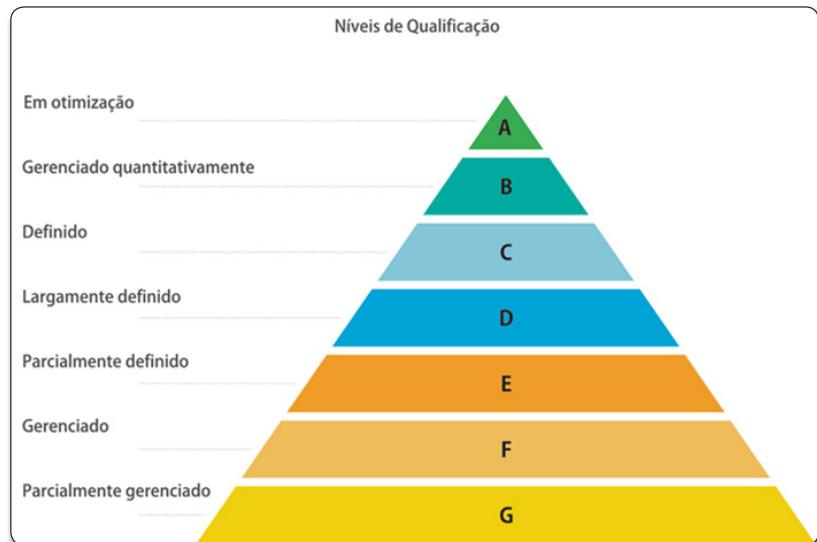
O primeiro modelo contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o MR-MPS-SW. Ele contém as definições dos níveis de maturidade, processos e atributos do processo. O MR-MPS-SW está em conformidade com os requisitos de modelos de referência de processo da Norma Internacional ISO/IEC 15504-2.

O segundo modelo contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o MR-MPS-SV. Ele contém as definições dos níveis de maturidade, processos e atributos do processo. O MR-MPS-SV está em conformidade com os requisitos de modelos de referência de processo da Norma Internacional ISO/IEC 15504-2.

Os dois modelos de referência definem níveis de maturidade que são uma combinação entre processos e sua capacidade.



**Figura 1.** Componentes do modelo MPS



**Figura 2.** Níveis de maturidade do MPS

A definição dos processos segue os requisitos para um modelo de referência de processo apresentados na ISO/IEC 15504-2, declarando o propósito e os resultados esperados de sua execução. Isso permite avaliar e atribuir graus de efetividade na execução dos processos.

A capacidade do processo é a caracterização da habilidade do processo para alcançar os objetivos de negócios, atuais e futuros; estando relacionada com o atendimento aos atributos de processo associados aos processos de cada nível de maturidade. O próximo dá um

destaque nos níveis de maturidade do MPS.BR e como eles são organizados.

## Níveis de maturidade

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos. O MR-MPS-SW e o MR-MPS-SV definem sete níveis de maturidade, como apresentado na **Figura 2**.

Para cada um destes sete níveis de maturidade é atribuído um perfil de processos que indicam onde a organização deve colocar o esforço de melhoria. O progresso e o alcance de um determinado nível de maturidade do MR-MPS-SW e do MR-MPS-SV se obtém quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e os resultados esperados dos atributos de processo estabelecidos para aquele nível.

A divisão em sete estágios tem o objetivo de possibilitar uma implementação e avaliação adequada às micros, pequenas e médias empresas. A possibilidade de se realizar avaliações considerando mais níveis também permite uma visibilidade dos resultados de melhoria de processos em prazos mais curtos.

### Processo

Segundo Sommerville, um processo de software é um conjunto de atividades relacionadas que leva à produção de um produto ou serviço de software. No MR-MPS-SW e no MR-MPS-SV, os processos são descritos em termos de propósito e resultados.

O propósito descreve o objetivo geral a ser atingido durante a execução do processo e os resultados esperados do processo estabelecem os resultados a serem obtidos com a efetiva

implementação do processo. Estes resultados podem ser evidenciados por um produto de trabalho produzido ou uma mudança significativa de estado ao se executar o processo.

### Capacidade do processo

A capacidade do processo é representada por um conjunto de atributos de processo descrito em termos de resultados esperados e expressa o grau de refinamento e institucionalização com que o processo é executado na organização/unidade organizacional. Nos dois modelos apresentados, à medida que a organização/unidade organizacional evoluí nos níveis de maturidade, um maior nível de capacidade para desempenhar o processo deve ser atingido.

O atendimento aos atributos do processo (AP), pelo atendimento aos resultados esperados dos atributos do processo (RAP), é requerido para todos os processos no nível correspondente ao nível de maturidade, embora eles não sejam detalhados dentro de cada processo. Os níveis são acumulativos, ou seja, se a organização está no nível F, esta possui o nível de capacidade do nível F que inclui os atributos de processo dos níveis G e F para todos os processos relacionados no nível de maturidade F (que também inclui os processos de nível G). Isto significa que, ao passar do nível G para o nível F, os processos do nível de maturidade G passam a ser executados no

# CURSOS ONLINE

A Revista .net Magazine oferece aos seus assinantes uma série de Cursos Online de alto padrão de qualidade.



### CONHEÇA OS CURSOS MAIS RECENTES:

- **Curso Padrões de Projeto com C#**
- **Curso básico de ASP .NET**
- **Curso de Introdução ao .NET Framework**
- **Curso Básico de C#**
- **C# 5 e suas novidades**
- **ASP.NET MVC – Sistema de Vestibular**

Para mais informações :

[www.devmedia.com.br/curso/netmagazine](http://www.devmedia.com.br/curso/netmagazine)  
**(21) 3382-5038**

nível de capacidade correspondente ao nível F. Em outras palavras, na passagem para um nível de maturidade superior, os processos anteriormente implementados devem passar a ser executados no nível de capacidade exigido neste nível superior. Veja as **Tabelas 2 e 3**.

Os diferentes níveis de capacidade dos processos são descritos por nove atributos de processo (AP), sendo:

- AP 1.1 O processo é executado;
- AP 2.1 O processo é gerenciado;
- AP 2.2 Os produtos de trabalho do processo são gerenciados;
- AP 3.1. O processo é definido;
- AP 3.2 O processo está implementado;
- AP 4.1 O processo é medido;
- AP 4.2 O processo é controlado;
- AP 5.1 O processo é objeto de melhorias incrementais e inovações;
- AP 5.2 O processo é otimizado de forma continua.

#### Nota

Os atributos de processo AP 4.1, AP 4.2, AP 5.1 e AP 5.2 somente devem ser implementados para os processos críticos da organização/unidade organizacional, selecionados para análise de desempenho. Os demais atributos de processo devem ser implementados para todos os processos.

A descrição detalhada de todos os processos, bem como seus propósitos e resultados esperados, estão disponíveis no site da SOFTEX (vide seção **Links**).

## MPS em números

De acordo com o portal SOFTEX, vale apresentar os seguintes números sobre MPS atualizados em julho de 2013:

- **Avaliações realizadas:** 482 Software e 2 Serviços = TOTAL 484;
- **Instituições Implementadoras (II):** 20;
- **Instituições Organizadoras de Grupos de Empresas (IOGES):** São 17 instituições e 65 projetos;
- **Instituições Avaliadoras:** 12;
- **Cursos realizados:** 248;
- **Provas realizadas:** 53;
- **Profissionais Capacitados:** mais de cinco mil.

Isso mostra que tanto as empresas como os profissionais estão investindo cada vez mais em melhoria de processo, buscando sempre aperfeiçoar as rotinas de trabalho, reduzindo

Nível	Processo	Atributo de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI Desenvolvimento para Reutilização – DRU Gerência de Decisões – GDE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
D	Verificação – VER Validação – VAL Projeto e Construção do Produto – PCP Integração do Produto – ITP Desenvolvimento de Requisitos – DRE	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	Gerência de Projetos – GPR (evolução) Gerência de Reutilização – GRU Gerência de Recursos Humanos – GRH Definição do Processo Organizacional – DFP Avaliação e Melhoria do Processo Organizacional – AMP	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
F	Medição – MED Garantia da Qualidade – GQA Gerência de Portfólio de Projetos – GPP Gerência de Configuração – GCO Aquisição – AQU	AP 1.1, AP 2.1 e AP 2.2
G	Gerência de Requisitos – GRE Gerência de Projetos – GPR	AP 1.1 e AP 2.1

**Tabela 2.** Níveis de maturidade do MR-MPS-SW [SOFTEX, 2012]

Nível	Processo	Atributo de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Trabalhos – GTR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Capacidade – GCA Gerência da Continuidade e Disponibilidade dos Serviços – GCD Gerência de Decisões – GDE Gerência de Liberação – GLI Gerência de Riscos – GRI Gerência da Segurança da Informação – GSI Relato de Serviços – RLS	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
D	Desenvolvimento do Sistema de Serviços – DSS Orçamento e Contabilização de Serviços – OCS	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
E	Avaliação e Melhoria do Processo Organizacional – AMP Definição do Processo Organizacional – DFP Gerência de Mudanças – GMU Gerência de Recursos Humanos – GRH Gerência de Trabalhos – GTR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
F	Aquisição – AQU Gerência de Configuração – GCO Garantia da Qualidade – GQA Gerência de Problemas - GPL Gerência de Portfólio de Trabalhos – GPT Medição – MED	AP 1.1, AP 2.1 e AP 2.2
G	Entrega de Serviços - ETS Gerência de Incidentes – GIN Gerência de Nível de Serviço – GNS Gerência de Requisitos – GRE Gerência de Trabalhos – GTR	AP 1.1 e AP 2.1

**Tabela 3.** Níveis de maturidade do MR-MPS-SV, [SOFTEX, 2012]

tempo e custo. Ressalto que essa busca incessante pela melhoria não é somente do setor de TI, mas sim de qualquer setor, pois, estamos atravessando um momento em que a qualidade está como meta de qualquer empresa que busca um lugar de respeito no mercado. Outro exemplo que destaco sempre é o setor de construção civil, que aperfeiçoa cada vez mais os métodos de trabalho.

## Conclusão

A qualidade dos processos reflete internamente e externamente nas empresas. A adoção de práticas de engenharia de software alinhadas aos padrões nacionais e internacionais de produção gera ganhos significativos em produtividade, com a redução do tempo e do investimento nos projetos. Uma empresa certificada além de ter os seus processos bem definidos e monitorados, passa a ter diferencial competitivo, visto pelos clientes como um indicativo de qualidade dos produtos e serviços oferecidos por ela.

Num ambiente extremamente competitivo como o atual, a utilização de metodologias e melhores práticas é condição obrigatória para obter a melhoria nos processos, a qualidade necessária, o cumprimento dos prazos e a satisfação dos clientes.

Dessa forma, o objetivo de muitas empresas tem sido obter qualificações para atender as exigências explícitas do mercado. O MPS.BR descreve princípios e práticas relacionadas ao processo de desenvolvimento de software e serviços. O modelo visa ajudar organizações produtoras de software e provedoras de serviços a melhorarem a capacidade de seus processos, isso por meio de um caminho evolucionário que considera desde processos com resultados imprevisíveis, e até mesmo caóticos, a processos disciplinados e definidos, com resultados previsíveis, com possibilidade de melhoria contínua e satisfação dos clientes.

Para finalizar, destaco que a melhoria do processo de desenvolvimento deve ser um processo contínuo, exige planejamento

anticipado e, o mais importante, apoio da alta direção e gerência da empresa.

No próximo artigo será apresentado um estudo de caso, onde foi aplicado o modelo MPS.BR para definição e implantação do processo de gerência de configuração em uma empresa de desenvolvimento de software.

### Links e Referências:

#### Portal SOFTEX – MPS.BR

<http://softex.br/mpsbr>

#### Portal SEI

<http://www.sei.cmu.edu/cmmi/>

#### Portal FUNSOFT

[http://www.funsoft.org.br/qualidade/modelo\\_mpsbr](http://www.funsoft.org.br/qualidade/modelo_mpsbr)

#### Blog CMMI.

<http://www.blogcmmi.com.br/qualidade/construcao-de-um-predio-de-6-andares-em-1-dia-video>

**McConnell, Steve.** *Professional software development: shorter schedules, higher quality products, more successful projects, enhanced careers.* Addison Wesley. ISBN 0-321-19367-9. 2003.

**SOMMERVILLE, Ian.** *Engenharia de Software; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hirama.* – 9. ed. – São Paulo: Pearson Prentice Hall, 2011.

### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros

## Gerenciando requisitos com MPS.BR



**Ivânia Ramos dos Santos**

[ivania.ramos.santos@gmail.com](mailto:ivania.ramos.santos@gmail.com)

Bacharel em Sistemas de Informação pela Faculdade Mater Dei (2008), especialista em Engenharia de Software pela Faculdade Mater Dei (2011). Atualmente é Gerente de Serviços de TI da empresa Alnova., professora na Faculdade Mater Dei, FESC e Sesu/ Senai. Experiência gestão de processos de qualidade utilizando metodologia MPS.BR.



**Giovanni De Bona**

[giovannidebona@gmail.com](mailto:giovannidebona@gmail.com)

Graduado em Sistemas de Informação e Especialista em Engenharia de Software na Faculdade Mater Dei. Atualmente Analista/Programador de software na empresa Sponte Informática. Pato Branco – PR.

O desenvolvimento de um software de qualidade e de fácil manutenibilidade não deve ser baseado apenas em código fonte e documentação, mas em um processo coerente com as características da organização desenvolvedora, que reflete a rotina de trabalho da equipe e as necessidades do negócio.

As solicitações de clientes e usuários devem ser registradas e atendidas em todas as fases do projeto, refletindo em um produto com excelência. Para isso, é imprescindível manter um controle padrão sobre implementações e manutenções através do gerenciamento de requisitos.

O gerenciamento de requisitos é um dos processos fundamentais em projetos de desenvolvimento de software. Por serem os requisitos a constituição da base para a definição da arquitetura do sistema, é importante que haja tarefas definidas para garantir sua qualidade, como modelagem, testes, validação e criação de baseline de requisitos (entende-se como a linha de base para

### **Porque esse artigo é útil:**

O nível G do MPS.BR define um conjunto de resultados esperados que devem ser alcançados por empresas que pretendam ser avaliadas neste nível. Uma das áreas é a de Gestão de Requisitos. Esse artigo apresenta os resultados esperados pelo modelo MPS.BR (Melhoria de processo de software brasileiro), para Gestão de requisitos (GRE), correspondentes ao nível G. Serão abordadas as necessidades de equipes sobre o controle e gerência de requisitos, através de entrevistas, o que resultou em protótipos de um software para gerenciamento de requisitos.

A análise deste tema é útil para organizações e equipes que buscam modelos que lhe auxiliem a estruturar o processo de gestão de requisito. As experiências reportadas aqui auxiliarão na escolha de um caminho a seguir durante as atividades de implantação de processo em sua organização.

acompanhamento de projetos ou requisitos, ou seja, um caminho pelo qual o projeto ou requisito deverá seguir). Este processo visa manter planos, artefatos, prioridades, status, especificações e atividades de desenvolvimento consistentes com o conjunto de requisitos definidos para o software.

A utilização de um modelo de qualidade de processo auxilia organizações de TI a gerenciar seus processos, assim como o de gestão de requisitos, como o MPS.BR que apresenta resultados esperados no processo GRE (Gerenciamento de requisitos), os quais se alcançados possibilitam à organização avançar níveis para futuras certificações, aumentar sua produtividade e melhorar seus processos organizacionais.

Este artigo visa abordar aspectos relacionados ao controle de mudanças em requisitos, ao gerenciamento da configuração, à rastreabilidade, à definição de prioridades, a classificação e à qualidade em requisitos seguindo os processos do MPS.BR. A coleta destas informações servirá para definir o que um controle de requisitos automatizado deverá possuir.

## Requisitos de software

Requisitos de software são sentenças que expressam as necessidades dos clientes e que condicionam a qualidade do produto, ou especificações de serviços que o sistema deve prover, suas restrições no sistema e conhecimento necessário para desenvolvê-lo. Uma vez criados, requisitos devem ser modelados, documentados, validados e acompanhados.

Requisitos podem ser classificados segundo alguma forma de categorização, que pode ser definida de acordo com as práticas de cada organização de software. Eles são geralmente classificados em:

- Funcionais: declarações de funções que o sistema deve satisfazer, reações a entradas e comportamento. Por exemplo: "o sistema deve permitir que cada professor lance as notas das turmas lecionadas". Também conhecidos como requisitos de negócio, estes representam objetivos de alto nível da organização ou cliente que requisitou o sistema.
- Não funcionais: requisitos que expressam condições que o software deve atender ou qualidades específicas que deve ter. Em vez de informar o que o sistema fará, os requisitos não funcionais colocam restrições no sistema. Por exemplo: "As consultas do sistema devem ser respondidas em menos de três segundos".

## Engenharia de Requisitos

Um processo de engenharia de requisitos é um conjunto atividades estruturadas que derivam, validam e mantém um documento de requisitos de um sistema do início ao fim do projeto.

Uma descrição completa de um processo inclui as atividades que devem ser conduzidas, a estrutura ou agenda destas atividades, as entradas e saídas de cada atividade e as ferramentas utilizadas para suportar a engenharia de requisitos.

Poucas organizações possuem um processo de engenharia de requisitos definido, preocupando-se apenas em obter como resultado a documentação de requisitos. Não existe um processo único que possa ser utilizado por todas as organizações. Cada uma deve desenvolver seu próprio processo para adaptar-se aos tipos de sistema em desenvolvimento, cultura organizacional, experiência dos envolvidos, entre outros.

## Principais atividades da engenharia de requisitos

Um processo de engenharia de requisitos possui atividades específicas e detalhadas para que o mesmo se torne viável e obtenha resultados, dentre eles se destacam:

- Levantamento (Elicitação): Consiste na identificação dos requisitos necessários à implementação de um sistema. Este processo engloba atividades de descoberta, refinamento, modelagem, documentação, especificação e manutenção do conjunto de requisitos.
- Análise: o objetivo da análise é descobrir problemas, incompletude, aplicabilidade e inconsistência nos requisitos elicitados. Eles normalmente são retornados aos stakeholders para verificação das implementações e restrições impostas. A análise é intercalada com elicitação, pois problemas são descobertos quando os requisitos são elicitados. Esta fase envolve resolução de conflitos através de técnicas como negociação de priorização e levantamento de pré-requisitos técnicos.
- Especificação: A especificação dos requisitos do software resulta do fluxo de requisitos. Este fluxo começa com a definição de um problema, que é descrito em uma proposta de especificação do software; e termina com uma especificação dos requisitos do software, que descreve, de forma detalhada, um conjunto dos requisitos que devem ser satisfeitos por uma solução implementável para o problema.
- Validação e Verificação: O processo de validação objetiva garantir que os requisitos refletem as necessidades do cliente em relação ao sistema (completeza), através da inspeção por padrões organizacionais, conhecimento sobre a organização e do sistema a ser desenvolvido, garantindo que os requisitos estejam livres de inconsistências.

Uma incorreta identificação dos requisitos de um software pode levar ao desenvolvimento de um produto que não atende aos objetivos para o qual foi planejado, sendo total ou parcialmente desperdiçado. Durante o processo de desenvolvimento, problemas nos requisitos podem gerar a necessidade de um novo ciclo de especificação, projeto, codificação e teste, afetando diretamente os custos e prazos envolvidos.

## Características e categorização dos requisitos

A primeira atividade na aplicação de uma gestão de requisitos consiste em categorizar as informações relevantes para a gerência de requisitos. Assim, é possível elaborar uma lista de informações para cada categoria. No decorrer da execução de um projeto, são necessárias informações detalhadas sobre os requisitos e as requisições dos stakeholders.

Durante a análise, criação, desenvolvimento e testes dos requisitos, o analista, o testador ou o programador podem informar as características apresentadas na **Tabela 1** no controle de requisitos.

Um bom controle de requisitos deve possuir uma seção de relatórios, onde os requisitos desejados são filtrados e trazidos para pequenas ou grandes reuniões para as mais diversas definições de desenvolvimento e análise.

Projeto/Produto	Identificação do projeto ao qual o requisito faz parte.
Controle de versão do requisito	Em que versão o problema ocorre, caso seja uma correção.
Identificação	Funcional / Não Funcional.
Prioridade	A priorização ajuda o gerente de projeto a resolver conflitos, planejar iterações e fazer as compensações necessárias. Quando as expectativas do cliente são altas e o tempo é curto, faz-se necessário entregar o produto com as funcionalidades mais relevantes, o mais cedo possível. A escala de importância a se utilizar é: Baixa, Média, Alta e Urgente.
Tipo	Sugestão de cliente, correção, implementação da empresa, erro de versão Beta.
Situação	As situações mais utilizadas e importantes são: Proposto, em desenvolvimento, implementado, em avaliação, em teste, atendido (pelo cliente), verificado (testado), liberado (para o cliente) e rejeitado.
Requisito Suplementar	Representa que o requisito é aplicável ao sistema como um todo.
Descrição	Descrição sucinta, simples e objetiva.
Detalhamento	Além dos detalhes, neste campo é possível informar também riscos e restrições associados.
Fonte	Indica quem identificou o requisito (cliente ou técnico ou analista). Para possíveis esclarecimentos.
Justificativa	Motivo de fazer o requisito.
Característica de Qualidade	Indica a característica de qualidade (Usabilidade, Confiabilidade, Eficiência, Portabilidade, Manutenibilidade, etc.) vinculada ao requisito não funcional em questão.
Requisitos Associados	Permite relacionar o requisito a outros já incluídos. Esta associação tem a finalidade de relacionar requisitos que tem afinidade entre si. Importante também na rastreabilidade.
Versão	Informa a versão em que o requisito será desenvolvido.
Comentário do desenvolvimento	Seção onde o desenvolvedor insere comentários sobre o desenvolvimento.
Anexos	Seção onde se pode anexar arquivos para complementar o requisito.
Interessados	Seção onde os clientes interessados no requisito são inseridos.
Log de situações	Log onde se apresenta o histórico de situações que o requisito já passou.
Desenvolvedor	Quem desenvolverá o requisito.
Testador	Quem testará o requisito.
Horas previstas	Tempo previsto para o desenvolvimento.
Associação ao controle de fontes	Associação ao controle de código fonte (se existir).
Análise de impacto das alterações requisitadas	Entende as possíveis implicações de se fazer a mudança. Identifica todos os arquivos, modelos e documentos que devem ser modificados se a mudança ocorrer. Identifica tarefas necessárias para implementar a mudança e estimar o esforço, tempo e recursos necessários.
Rastreabilidade	Objetiva acompanhar a vida de um requisito, desde a origem até a implementação. Garante que os requisitos especificados são associados às necessidades dos clientes. Garante também que todo produto de trabalho está associado aos requisitos identificados.
Gerência de iterações de teste	Controla as iterações feitas pelo setor de testes até o requisito ser finalizado e estiver pronto para liberação.

**Tabela 1.** Característica e categorização dos requisitos

Relatórios também servem para acompanhar quantidades de requisitos dentro de um determinado intervalo de tempo e em determinada categoria e classificação, ajudando no acompanhamento das fases do projeto.

Complementa-se o controle de requisitos com uma seção de gráficos e estatísticas para também serem levados em reunião ou apresentados ao gerente de projetos como resultados do produto.

Em relação à economia de tempo de atendimento a uma solicitação de cliente, pode ser disponibilizado um cadastro de sugestões onde o próprio cliente insere suas solicitações, contendo campos que deixam implícitos o que o cliente deseja, ou seja, possibilitando o total entendimento do analista de requisitos.

## MPS.BR – Melhoria de Processo de Software Brasileiro

Segundo o Guia Geral MPS de Software (2012), o MPS.BR é um modelo de qualidade de processos das empresas de

desenvolvimento de software que visa melhorar processos e serviços a médio e longo prazo, buscando atender as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software e serviços.

O MR-MPS-SV define sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A.

Para cada um dos níveis de maturidade é atribuído um perfil de processos, que indicam onde a organização deve colocar o esforço de melhoria. O progresso e o alcance de um determinado nível de maturidade se obtêm quando são atendidos os propósitos e todos os resultados esperados dos respectivos processos e os resultados esperados dos atributos de processo estabelecidos para aquele nível.

Nesse artigo são adotados os resultados esperados em Gerenciamento de Requisitos, do nível G. O nível G é chamado 'Parcialmente Gerenciado', por ser a primeira fase do modelo, onde os processos de gestão de projetos e requisitos são mapeados e padrões definidos para o gerenciamento.

Sua implementação deve ser executada com cautela por estabelecer o início dos trabalhos em implantação de melhoria dos processos de software na organização. Ao final da implantação deste nível a organização deve ser capaz de gerenciar parcialmente seus projetos de desenvolvimento de software.

No nível G, o projeto pode usar os seus próprios padrões e procedimentos, não sendo necessário que se tenha padrões organizacionais. Se, porventura, a organização possuir processos já definidos e os projetos necessite se adaptar aos processos existentes, deve-se registrar essa adaptação durante o planejamento do projeto. Adaptações podem incluir alteração em processos, atividades, ferramentas, técnicas, procedimentos, padrões, medidas, dentre outras.

## **Gerenciamento de requisitos**

Um dos critérios para se alcançar o nível G é alcançar os resultados esperados em gestão de requisitos, que tem como propósito gerenciar os requisitos de trabalho e dos componentes de trabalho e identificar inconsistências entre os requisitos, os planos de trabalho e os produtos de trabalho. Os resultados esperados em GRE no nível G são descritos na **Tabela 2**.

Resultados Esperados	
<b>GRE 1</b>	O entendimento dos requisitos é obtido junto aos fornecedores internos ou externos de requisitos;
<b>GRE 2</b>	Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido;
<b>GRE 3</b>	A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
<b>GRE 4</b>	Revisões em planos e produtos derivados do trabalho são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
<b>GRE 5</b>	Mudanças nos requisitos são gerenciadas ao longo do trabalho.

**Tabela 2.** Resultados esperados

Através destes resultados que o gerenciamento se torna simples e objetivo, fazendo com que os principais problemas na gestão possam ser evitados ou contornados de forma fácil.

## **Análise de resultados**

A fim de coletar necessidades de envolvidos nos processo de gestão de requisitos e alcançar os GRE's 1 ao 5, do nível G – MPS.BR – foram aplicados questionários, obtidos em pesquisas realizadas com 25 profissionais de empresas de software no sudoeste do Paraná.

A pesquisa relatou que o principal problema nas organizações é ocasionado pelo não entendimento da equipe em relação aos requisitos e suas alterações.

Dos entrevistados, 63% relatam que o único momento onde se realiza a análise de requisitos, na empresa onde trabalha, é

em reunião com a equipe, já 30% relata que é através de troca de e-mail. Outros 7% possuem outras maneiras de análise, informada. Este resultado mostra que as reuniões ainda são predominantemente adotadas por melhorarem o entendimento perante todos os setores, principalmente na análise de novos requisitos, sendo fundamental haver um processo formal, onde haja registro e feedback.

Antes de realizar a construção de um projeto, também em momentos estratégicos, a viabilidade deve ser analisada, tendo em vista que o produto a ser desenvolvido deve estar alinhado às expectativas do negócio e da equipe (existência de recurso suficiente, tempo viável, viabilidade financeira, de mercado, de ramo e exportação, entre outros). Nas entrevistas notou-se que a análise de viabilidade é, na maioria das vezes, realizada informalmente, e apenas quando o projeto já foi iniciado. Como comprova os 71% dos profissionais que relatam que a análise é realizada na fase de arquitetura dos requisitos, outros 17%, surpreendem, informando que nas organizações onde é realizada no desenvolvimento. Sendo ressaltado que a preocupação esta relacionada a critérios técnicos e não estratégicos. Outros 12% fazem a análise em outras etapas.

Outro fator que surpreende é em relação à criação de artefatos (planos de projetos, especificação de requisitos e modelagem), os quais são critérios básicos de qualidade. Dos entrevistados, 56% dizem que esses artefatos são criados parcialmente, já 40%, afirma que há a criação e apenas 4% não cria artefatos que auxiliariam no gerenciamento. Destacando que 83% dos entrevistados, que relatam criarem artefatos, fazem também a revisão desses durante o desenvolvimento do requisito.

Foi levada em consideração a questão da utilização de um software que apoie a organização na gestão de requisitos: 76% afirma já utilizar um sistema para esse fim, sendo que 80% dessas organizações desenvolveram sua própria ferramenta e 20% utiliza software pago, de terceiros. Este resultado se dá pela economia financeira e pela personalização do controle perante as necessidades da empresa.

A importância de um software de gestão é reforçada por 96% dos entrevistados defenderem que o controle de requisitos melhoraria a tarefa que executam na organização.

Em relação à criação de um requisito novo para o sistema, 84% dos entrevistados afirmam que os mesmos são formatados em pacotes para cada projeto, ou seja, são vinculados à um software específico, juntamente com os outros. Já 8% dizem criar requisitos sem associar a um projeto específico, o que dificulta a rastreabilidade.

Os profissionais foram questionados sobre a forma que é realizada a gestão de requisitos, se a mesma satisfaz as necessidades da empresa. 63% afirmaram que satisfaz parcialmente e apenas 25% defende que o controle de requisitos utilizado satisfaz totalmente as necessidades da empresa, sendo que 12% afirmam que não satisfaz.

A abordagem mostra a carência nas organizações de um processo coerente com suas necessidades, não havendo um controle específico sobre o rastreio de requisitos, gerando principalmente perda de tempo. Outro ponto é a análise de

impacto, que pode gerar problemas no projeto, devido a alterações não previstas, assim como pela falta de um controle de iterações dos requisitos, de gravação de um histórico.

Um processo de gerenciamento de requisitos coerente com as características da organização e automatizado é apontado pelos entrevistados como ponto de melhora em outros critérios fundamentais. 25% apontando o tempo e o prazo, outros 25% dos entrevistados descrevem como um dos critérios o controle de prioridades, 21% o histórico de implementações, 18% a 20% compartilhamento e o controle do conhecimento e controle de versão de requisitos, defendido por 16% dos entrevistados.

As organizações buscam automatizar seus processos, mas muitas vezes não satisfazem as necessidades apontadas pelos envolvidos, talvez pela falta de um modelo que norteie o propósito da ferramenta e pela falta de pesquisa com os reais usuários do processo. Por conta disso, esse artigo também apresenta protótipos baseados em necessidades identificadas pelos entrevistados e nos GREs, do nível G do MPS.BR.

## Geração de protótipos e análise de cenários de uso

A partir de agora apresentaremos alguns protótipos com o intuito de apresentar sugestões de telas de um sistema que satisfaça os resultados esperados pelo processo de requisitos, do modelo MPS.BR nível G, também as necessidades apontadas por usuários nas entrevistas descritas na seção anterior.

Na tela “Cadastro de requisito”, apresentada na **Figura 1**, a guia ‘Dados Principais’ possibilita ao analista de requisitos informar nos campos correspondentes ao produto - versão, prioridade (baixa, média, alta ou urgente), tipo (sugestão de cliente, correção, implementação da empresa, erro de versão), situação (proposto, em desenvolvimento, implementado, em avaliação, em teste, atendido, verificado, liberado ou rejeitado), identificação (funcional / não funcional) – também se é suplementar ou não, descrição, detalhamento, justificativa, fonte (quem identificou) e a

**Figura 1.** Cadastro do requisito

**Figura 2.** Dados do desenvolvimento e testes

característica de qualidade (usabilidade, confiabilidade, eficiência, portabilidade, manutenibilidade, etc.).

No guia ‘Desenvolvimento’, da tela “Cadastro de Requisito”, como pode ser observado na **Figura 2**, podem ser registradas informações pelo desenvolvedor na fase de desenvolvimento como situação, desenvolvedor, testador responsável e a previsão de tempo para a conclusão da fase. No campo “Comentários” devem ser descritos os detalhes e peculiaridades, tanto para outros programadores visualizarem, quanto para o setor de testes.

Na área “Log de situação e iteração de testes” constará um histórico de todas as situações pelas quais o requisito passou. Esta área possuirá também registros das iterações de testes, ou seja, se foi testado, se o teste passou ou falhou, se foi revisado ou se está pronto para liberação.

Para obter o alcance do GRE 1 – ‘O entendimento dos requisitos é obtido junto aos fornecedores internos ou externos de requisitos’ e do GRE 2 – ‘Os requisitos são avaliados com base em critérios objetivos e um comprometimento da equipe técnica com estes requisitos é obtido’, foram criados os protótipos mostrados pelas **Figuras 3 a 5**.

Na tela “Avaliação de Requisitos”, ilustrada pelo protótipo da **Figura 3**, o usuário, seja ele, desenvolvedor, analista ou testador, pode consultar uma lista de requisitos pré-filtrados, onde pode realizar sua aprovação ou não, clicando sobre o botão ‘Visualizar requisito completo’.

O botão ‘Visualizar requisito completo’ direciona o usuário para a guia ‘Avaliações’, como apresenta **Figura 4**.

No guia ‘Avaliações’ o usuário pode selecionar a opção ‘Aprovado’ ou ‘Reprovado’ e informar o motivo e

alguma observação pertinente, nos campos específicos. A seção “Alterações do requisito” será relatada no GRE 5.

O usuário poderá visualizar as avaliações de outros usuários do requisito, conforme **Figura 5**, clicando no botão “Visualizar outras avaliações”.

Nesta tela podem ser visualizados os usuários do sistema e suas avaliações, com justificativa e resultados totalizados na lateral. A mesma tem por objetivo gerar uma documentação para avaliar o requisito como um todo para o seu desenvolvimento.

Abaixo da parte tracejada encontram-se as avaliações dos clientes, seus comentários (histórico) e os resultados totalizados. Estes resultados podem ser inseridos pelo próprio cliente, ou pela equipe técnica que entrou em contato com o mesmo.

Para alcançar o resultado esperado do GRE 3 – ‘A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida’, existe a preocupação em avaliar o impacto de mudanças, ou seja, no que o desenvolvimento daquele requisito afetará. Na rastreabilidade podem ser vinculados documentos no requisito, para visualização de usuários, auxiliando na execução de tarefas do projeto.

Na **Figura 6** pode ser observada a guia “Análise de Impacto”, onde no campo de mesmo nome pode ser inserido texto pelo analista de requisitos informando dados sobre o impacto de desenvolvimento do requisito. Já no campo “Classes e documentos associados” serão inseridos nomes e caminhos de classes ou documentos que poderão estar envolvidos no desenvolvimento do requisito.

No grupo “Requisitos Associados” podem ser visualizados requisitos associados a ações individuais para identificar o que foi inserido na análise de impacto, quais os stakeholders do requisito, quais as classes de projeto associadas (link de rastreabilidade) e a visualização completa do requisito (um link de rastreabilidade é um relacionamento entre itens de rastreabilidade, por exemplo, um requisito a uma classe de projeto ou a um caso de teste).

O GRE 4 – ‘Revisões em planos e produtos derivados do trabalho são realizadas visando identificar e corrigir inconsistências em relação aos requisitos’ e o GRE 5 – ‘Mudanças nos requisitos são gerenciadas ao longo do trabalho’ buscam revisões, correções e controle de mudanças. Através do alcance deles podem ser identificados problemas ou detalhes não registrados na criação do requisito, além de uma gerência de alterações informando todos os interessados.

A cada requisito criado ou alterado, é enviada uma mensagem pelo sistema

ID 12345 - Descrição resumida do requisito	<input type="button" value="Visualizar requisito completo"/>	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado
ID 54321 - Descrição resumida do requisito 2	<input type="button" value="Visualizar requisito completo"/>	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado
ID 13524 - Descrição resumida do requisito 3	<input type="button" value="Visualizar requisito completo"/>	<input type="radio"/> Aprovado	<input checked="" type="radio"/> Reprovado

**Figura 3.** Requisitos a serem avaliados

Avaliação do requisito:	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar outras avaliações"/>			
Motivo:	<input type="text"/>					
Observação:	<input type="text"/>					
Alterações do requisito:	Data da alteração:	Usuário	Alteração:	Avaliação:	Outras avaliações	
	12/05-2013	Pedro (desenvolvedor)	<input type="button" value="Visualizar alteração"/>	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar"/>
	13/05-2013	João (testador)	<input type="button" value="Visualizar alteração"/>	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar"/>

**Figura 4.** Avaliações do Requisito

Usuário Fulano:	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar comentários"/>	Resultados:
Usuário Ciclano:	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar comentários"/>	2 aprovações
Usuário Beltrano:	<input type="radio"/> Aprovado	<input checked="" type="radio"/> Reprovado	<input type="button" value="Visualizar comentários"/>	1 reprovação
Cliente Fulano:	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar comentários"/>	4 sem resposta
Cliente Beltrano:	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar comentários"/>	
Cliente Ciclano:	<input checked="" type="radio"/> Aprovado	<input type="radio"/> Reprovado	<input type="button" value="Visualizar comentários"/>	

**Figura 5.** Avaliações dos usuários e clientes

Análise de impacto:	<input type="text"/>				
Classes e documentos associados:	<input type="text"/>				
Requisitos associados	ID: 12345 - Situação: Desenvolvido	<input type="button" value="Análise de impacto"/>	<input type="button" value="Classes associadas"/>	<input type="button" value="Interessados"/>	<input type="button" value="Visualizar requisito"/>
	ID: 12346 - Situação: Proposto	<input type="button" value="Análise de impacto"/>	<input type="button" value="Classes associadas"/>	<input type="button" value="Interessados"/>	<input type="button" value="Visualizar requisito"/>
	ID: 12345 - Situação: Em Testes	<input type="button" value="Análise de impacto"/>	<input type="button" value="Classes associadas"/>	<input type="button" value="Interessados"/>	<input type="button" value="Visualizar requisito"/>
	ID: 12345 - Situação: Recusado	<input type="button" value="Análise de impacto"/>	<input type="button" value="Classes associadas"/>	<input type="button" value="Interessados"/>	<input type="button" value="Visualizar requisito"/>

**Figura 6.** Análise de Impacto

gerenciador de requisitos e enviando um e-mail para os interessados (usuários, desenvolvedores, testadores, analistas) a fim de que avaliem e revisem a alteração ou implementação, sendo gravadas ações em histórico do requisito.

Para os stakeholders aprovarem estas alterações, uma seção (Figura 4, alterações do requisito) dentro do requisito é aberta, que consta o histórico de alterações do requisito, sendo que cada alteração gera nova avaliação para todos os interessados. Para facilitar a identificação de requisitos ainda não avaliados, a guia correspondente é identificada por um asterisco conforme Figura 7.

Informações de mudanças de requisitos auxiliam no tempo de desenvolvimento e de avaliação durante o trabalho. A gerência destas mudanças ajuda as partes interessadas a ficarem sempre atualizadas perante alterações de requisitos e evitam possíveis erros de desenvolvimento.

## Conclusão

Um controle de requisitos juntamente com o MPS.BR faz com que todo o ciclo de vida de um requisito ou uma solicitação possa ser controlado, acompanhado, detalhado e melhorado.

Baseando-se nos GRE's, as organizações conseguem melhorar tempo, custo e otimização de processos. Protótipos de telas permitem um controle mais abrangente dos stakeholders e da equipe técnica, sobre os requisitos e suas alterações.

Um controle de requisitos completo e bem alimentado faz com que não só o gerenciamento se torne melhor, mas também a documentação e a agilidade que os setores da empresa podem usufruir para as mais diversas tarefas relacionadas ao projeto.

As organizações buscam suprir suas necessidades através da automatização de processos, como o de gestão de requisitos, buscando a velocidade de processamento e resposta de processos internos, o entendimento geral da equipe e garantia de qualidade.

Novo requisito ou Requisito xxx					
Dados Principais	Análise de Impacto	Desenvolvimento	Anexos	Interessados	Avaliações *

**Figura 7.** Requisito não avaliado

### Referências:

SAYÃO, Miriam; BREITMAN, Karin Koogan. **Gerência de Requisitos. Faculdade de Informática da PUC-RS e DI/PUC-Rio.**

AUDY, Jorge; PRIKLADNICKI, Rafael. **Desenvolvimento distribuído de software.** Rio de Janeiro, Elsevier, 2007;

ESPINDOLA, Rodrigo; LOPES, Leandro; PRIKLADNICKI, Rafael; AUDY, Jorge Luiz. **Uma Abordagem Baseada em Gestão do Conhecimento para Gerência de Requisitos em Desenvolvimento Distribuído de Software.** Pontifícia Universidade Católica do Rio Grande do Sul;

Sommerville, Ian; Sawyer, Peter. "Requirements Engineering – a good practice guide". New York: John Wiley & Sons Ltd, 1997, 391p.

Kotonya e Sommerville (1998). **Requirements Engineering: Processes and Techniques.** Gerald Kotonya, Ian Sommerville. Wiley, 1998.

HORRIAN, Hossein. MAHMUD, Shafquat. KARTHIKEYAN, Srinivasan. **Requirements Engineering in Agile methods.** Dept. of Computer Science, University of Calgary, Canada, 2003.

IEEE. IEEE Std. 830 – 1993. **IEEE Recommended Practice for Software Requirements Specifications,** in [IEEE94].

### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros



## Gestão de riscos nos requisitos

Conheça os passos para a identificação de riscos durante o levantamento de requisitos

**A** engenharia de requisitos é uma área de investigação que procura atacar um ponto fundamental no processo de desenvolvimento, que é a definição do que se pretende produzir. Pode-se dizer que a engenharia de requisitos é o ponto de encontro entre o desenvolvimento de software formal e o informal.

A fim de produzir a necessidade do cliente, dentro do prazo e orçamento estipulados, é essencial desenvolver um processo que inclua a gestão dos riscos associados com a fase dos requisitos, de modo a contribuir para a melhoria gradual do desenvolvimento de processos e também de gerenciamento de projetos software com intuito de atingir a satisfação do cliente.

Ao falar em riscos em projetos, tenha a percepção que estes afetam eventos futuros. Qualquer projeto que possui um ciclo de vida que pode ser alterado pelo acontecimento de um evento, e este sendo no início do projeto qualificado como risco, evitando a sequência normal do projeto. Por exemplo, se uma pessoa possui uma reunião em Porto Alegre e necessita se

### **Porque esse artigo é útil:**

A gestão de riscos associados aos requisitos de software, individuais ou em grupos, desde a fase de engenharia de requisitos pode permitir minimizá-los, evitá-los e controlá-los. O envolvimento dos riscos que possam afetar o desenvolvimento ou a qualidade dos requisitos e ações para evitá-los permite minimizar os problemas que persistem no desenvolvimento de software. Desta forma, os riscos mais importantes são os associados com as principais características dos requisitos de qualidade.

A discussão deste tema é útil quando tratamos de aspectos relacionados aos processos da engenharia de requisitos como a elicitação e a gerência de requisitos e seus pontos críticos no sentido de antecipar e restringir riscos por meio de ressalvas que possam garantir a qualidade do projeto que se está desenvolvendo além de evidenciar como pequenos erros podem se tornar decisivos para o projeto e até impedir a sua conclusão.



**Fábio Sarturi Prass**

[fabiopras@gmail.com](mailto:fabiopras@gmail.com)

Diretor da FP2 Tecnologia ([www.fp2.com.br](http://www.fp2.com.br)), possui Mestrado em Ciência da Computação pela Universidade Federal de Santa Maria (UFSM) e atua como professor na Antônio Meneghetti Faculdade, no curso de Sistemas de Informação. Domínio em Java, .NET e UML. Sólido conhecimento em Engenharia de Software e Banco de Dados.

deslocar de avião, mas há um atraso no voo, esse atraso é um risco. Um risco pode vir a se tornar um grande problema caso não venha a ser gerenciado.

Os riscos estão presentes de forma permanente em termos de incerteza (risco que pode ou não ocorrer) e perda (se o risco se tornar uma realidade, perdas indesejáveis ocorreram). Sendo assim, têm-se definidas as categorias acerca dos riscos: riscos de projetos, que ameaçam o plano de execução do projeto; riscos técnicos, que ameaçam a qualidade e o cronograma do projeto; riscos conhecidos, aqueles descobertos em avaliações; riscos previsíveis, conhecidos a partir da experiência; e riscos imprevisíveis, que podem ocorrer, mas é muito difícil identificar previamente.

A gestão de riscos na área de software procura formalizar o conhecimento orientado à minimização de riscos em projetos de software mediante a geração de princípios e boas práticas de desenvolvimento. No entanto, até o momento têm sido propostas e utilizadas diferentes abordagens para a gestão riscos. É claro que algumas organizações ainda utilizam métodos específicos de modo explícito e sistemático para gerenciar os riscos em seus projetos de software.

Este artigo pretende demonstrar que a gestão de riscos associada ao desenvolvimento de software deve iniciar pela etapa de engenharia de requisitos, criando, desde as etapas iniciais, um plano de contingência para evitar e monitorar os riscos durante o projeto.

## Regras e responsabilidades da gestão de riscos

O gerenciamento de riscos de um projeto contém os processos que abordam a realização da identificação, análise, planejamento, monitoramento e controle, resolução e comunicação. A maior parte desses processos é atualizada durante todo o ciclo de vida do projeto. Com isso, têm-se as seguintes etapas:

- **Identificar:** produzir os riscos que podem afetar o projeto, gerando sempre a documentação das suas características;
- **Analisa:** estimar a probabilidade e a intensidade da perda integrada com cada item de risco e com a possível composição desses riscos;
- **Planejar:** definir como administrar as atividades de gerenciamento dos riscos de um projeto;
- **Monitorar:** acompanhar o status do risco e das ações executadas e identificar novos riscos;
- **Resolver:** executar as ações planejadas e reportar os resultados da resolução do risco;
- **Comunicar:** prover informações para e entre entidades e níveis organizacionais envolvidos no projeto. Inclui níveis dentro do projeto, da organização, da organização do cliente e a comunicação entre desenvolvedor, cliente e usuário.

Cada um desses processos pode ser relacionado com os membros que fazem parte do projeto. Assim, tem-se a distribuição conforme a **Tabela 1**.

Através desses processos, o objetivo da gestão de riscos é minimizar a probabilidade e as consequências de riscos negativos (ou ameaças) e maximizar a probabilidade e consequências de riscos positivos (ou oportunidades) identificadas no projeto para que seus objetivos sejam atendidos. Isto é alcançado seguindo algumas orientações:

- Identificar todos os riscos conhecidos do projeto;
- Realizar uma avaliação da probabilidade de ocorrência e o potencial impacto;
- Quantificar o que seria o custo dos riscos se vier a ocorrer;
- Criar planos de ação para gerenciar os riscos de alta prioridade;
- Reconhecer e gerenciar o risco o mais cedo possível.

Os benefícios obtidos através da realização de uma boa gestão de riscos são:

- Reduz os custos do projeto;
- Melhora a satisfação do cliente;
- Aumenta a capacidade e probabilidade de sucesso;
- Facilita o desenvolvimento do projeto;
- Diminui drasticamente as surpresas nos projetos;
- Ajuda a empresa a alcançar os objetivos de negócios evitando problemas no projeto que poderiam causar perdas inesperadas e não planejadas.

Em todo o gerenciamento de risco é necessário desenvolver um plano de gestão de risco, que deve descrever:

- A estratégia de gestão de risco;
- Domínio do esforço de gerenciamento de riscos;
- Como está o planejamento para realizar a identificação de riscos;
- Como se realizará a análise de risco (qualitativa, quantitativa, priorização);
- Como se realizará o monitoramento e controle;
- Orçamento de Gestão de Risco;
- Calendário das atividades de gestão de risco;
- Funções e responsabilidades.

## Atividades de gestão de riscos

A gestão de riscos do projeto inclui os processos de planejamento associados ao gerenciamento de riscos, identificação e análise, respostas aos riscos, monitoramento e controle dos riscos do projeto. A maioria destes processos é atualizada durante o projeto. A **Figura 1** mostra como essas atividades estão relacionadas.

Estes processos interagem uns com os outros e com processos de outras áreas. Cada processo pode envolver o esforço de uma ou mais pessoas ou grupos de pessoas, dependendo das necessidades do projeto. Cada processo ocorre pelo menos uma vez em todos os projetos e ocorre em uma ou mais fases do projeto, se este for dividido em fases.

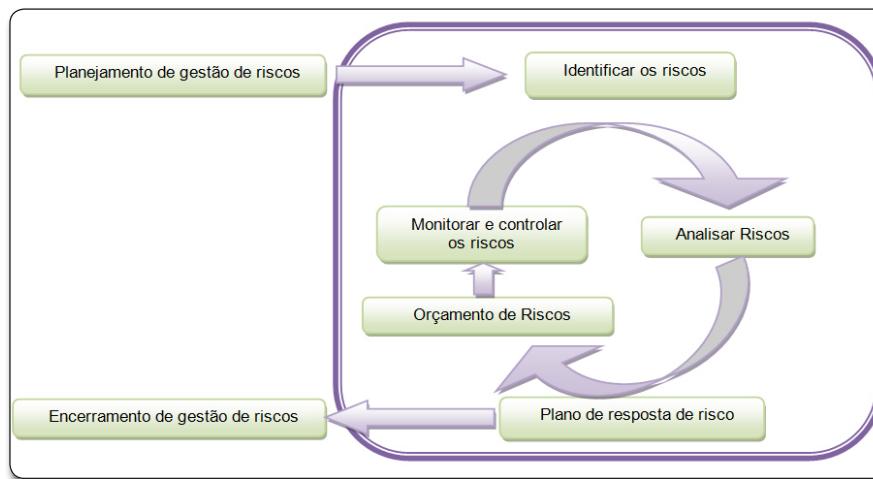
## Gestão de risco nos requisitos

Um risco é um fator que influencia negativamente o sucesso do projeto e deve ser considerado por três componentes: o evento, a possibilidade de ocorrer e o impacto do evento. Constantemente é avaliado como um evento de implicações negativas, quanto ao custo, o tempo, e a qualidade. Por este motivo os gerentes se empenham em como evitá-los e em como lidar com eles. Esquecendo que, eventualmente, os riscos podem ter implicações positivas.

Essa premissa mostra que o risco está, implicitamente, associado a qualquer atividade e acompanha qualquer mudança, pois envolve a escolha e incerteza de um processo ocorrer. No momento de iniciar uma atividade de elicitação de requisitos

de software, se inicia a identificação dos riscos. Esta responsabilidade é do gerente de projetos (ou outro cargo semelhante) que antecipa os riscos que possam afetar o desenvolvimento ou a qualidade dos requisitos e toma medidas para evitá-los.

Esta atividade assegura que, a partir do início do processo de desenvolvimento de software, se executará as tarefas que visam assegurar a qualidade do produto. Neste entendimento, devem-se adotar os passos apresentados nos tópicos a seguir.



**Figura 1.** Atividades da Gestão de Riscos

### **Passo 1 – Identificação de Riscos**

Problemas potenciais podem ocorrer no processo de engenharia de requisitos, ou na especificação de requisitos de software (ERS), tais como orçamento, pessoal, usuário, organizacionais, técnicas ou outras comunicações. Deve-se começar com a análise de risco genérico, que constitui uma ameaça potencial para todos os projetos de software e que possam estar presentes no projeto atual. Em seguida, deve-se identificar os riscos específicos, envolvendo um conhecimento profundo do projeto que estão relacionados com o ambiente de desenvolvimento, tecnologia, experiência e tamanho da equipe.

	Gerente de projeto	Envolvido nos negócios	Cliente	Especialistas do negócio	Equipe do projeto	Responsável por um risco
Planejamento de gestão de riscos	Desenvolve e mantém o plano de gestão de riscos	Fornecer informações se o nível de risco é aceitável	Fornece dados sobre os critérios para aceitação das entregas que podem influenciar o risco do projeto			
Identificação de riscos	Identifica os riscos do projeto	Fornecer informações históricas que deverá ajudar a identificar os riscos do projeto		Fornece informações históricas que deverá ajudar a identificar os riscos do projeto	Trabalha com o gerente de projeto para identificar riscos	
Análise de riscos	Analisa os riscos do projeto	Valida as suposições feitas durante o planejamento do projeto e fornece a entrada sobre a probabilidade e o impacto dos riscos		Valida as suposições feitas durante o planejamento do projeto e fornece a entrada sobre a probabilidade e o impacto dos riscos		
Planejamento de resposta aos riscos	Direciona as respostas do processo de planejamento, identifica os participantes e define os planos de resposta aos riscos, com a ajuda da equipe do projeto	Participa do desenvolvimento de planos de resposta para cada risco individual e assume a responsabilidade de seus planos				
Controlar e monitorar riscos	Responsável pelo fim do monitoramento e controle de riscos. É responsável por manter plano de risco	Identifica riscos novos e que tenham mudado. Avalia a eficácia da gestão de riscos, planos de resposta e qualquer ação de resposta				Responsável pelo plano de resposta ao risco
Encerramento da gestão de riscos	Registra as lições aprendidas durante a gestão de risco e apresenta os resultados para o encerramento do projeto					

**Tabela 1.** Funções e responsabilidades

Requisito	Resposta	Impacto	Componente	Medida
Requisito_01	Sim	Sim	Custo	Critico
Requisito_02	Não	Não	--	--
Requisito_03	Não	Sim	Tempo	Critico

**Tabela 2.** Identificação dos riscos

Risco	Tipo de Risco	Descrição
Equipe	Projeto, produtos e negócios	Pessoas experientes deixando o projeto antes do término
Alteração de Requisitos	Projeto e do produto	Existência de mais mudanças do que os requisitos previstos inicialmente
Atrasos na especificação	Projeto e do produto	Especificações de interface essenciais final
Subestimando o tamanho	Projeto e do produto	A exigência de tamanho (o processo de ER/ERS) tem sido subestimada
Fraç desempenho da ferramenta CASE	Produto	Ferramentas CASE que ajudam o projeto não tem o desempenho e características esperadas

**Tabela 3.** Lista de riscos e classificação

Para os requisitos e ERS deve-se começar esta tarefa respondendo a pergunta: que características especiais tem esse requisito, ou o grupo de requisitos, que podem estar ameaçadas?

Um método comprovadamente eficaz para a identificação de riscos é criar uma lista de fatores de risco. Esta lista deve incidir sobre os riscos relacionados com o tamanho do produto, o impacto sobre o projeto e organização, características do cliente, definição de processos, ambiente de desenvolvimento, tecnologia de construção, o tamanho da equipe e experiência do pessoal.

Nesta atividade, o procedimento é estabelecido a fim de criar uma lista de verificação para as necessidades individuais e um para o ERS. Os resultados são convertidos em tabelas para facilitar a análise subsequente. Os riscos serão considerados a partir do comportamento das características individuais e de grupo (ambiguidade, clareza, integridade, consistência, rastreabilidade, entre outros). Dependendo da intensidade do comportamento da característica e exigência de qualidade, é possível controlar de forma eficaz o risco no processo de engenharia de requisitos antes de passar para a fase seguinte do ciclo de vida do projeto.

Risco, se não for gerido na fase inicial, pode ser difícil de controlar, uma vez que se iniciou o processo de desenvolvimento. Desequilíbrios no comportamento das características de qualidade dos requisitos que resultam no surgimento de riscos tornam o projeto vulnerável e impraticável à aplicação de qualquer plano de qualidade,

Risco	Tipo de Risco	Riscos
Requisito_01	Da Equipe	A equipe não tem o conhecimento necessário para lidar com a complexidade da exigência
		Os membros da equipe não estão disponíveis em momentos críticos
	De Requisitos	Mudanças nas condições que requerem modificações no desenho
		Os clientes que não entendem o impacto das mudanças nos requisitos
	Estimativa	O tempo necessário para desenvolver o processo de engenharia de requisitos é subestimado
	Comunicação	O cliente é incapaz de participar de revisões e reuniões

**Tabela 4.** Riscos por requisitos

Tipo de Risco	Riscos potenciais
Equipe	Incapaz de recrutar pessoal com as habilidades exigidas.
Organizacional	A organização é reestruturada e uma nova administração é responsável pelo projeto.
Ferramentas	As várias ferramentas CASE não estão disponíveis
Requisitos	Mudanças nos requisitos que necessitam de modificações ao projeto.
Estimativas	O tamanho do sistema em desenvolvimento é subestimado.

**Tabela 5.** Riscos por tipo

razão pela qual se convém detectar e tratar os riscos com antecedência.

Um exemplo da aplicação deste dispositivo é a seguinte:

- **Característica do Requisito:** Ambiguidade.
- **Pergunta:** O requisito é ambíguo?
- **Aplicação:** Aplicado a todos os requisitos especificados (conforme observado na Tabela 2).

Este procedimento é estabelecido para definir e listar os riscos que podem afetar o atual processo da engenharia de requisitos, como mostrado nas Tabelas 3 a 5.

### Passo 2 – Avaliação e Projeção dos Riscos

O objetivo fundamental das técnicas de avaliação de riscos incide sob a organização dos dados obtidos (que determinam

os riscos) com a finalidade de permitir a aprendizagem para novos projetos. Muitas das técnicas existentes têm um elevado grau de subjetividade envolvida uma vez que depende da análise dos profissionais envolvidos no processo de avaliação.

Já a projeção dos riscos consiste em determinar a probabilidade de uma ocorrência de risco e as consequências que esta pode ter, como por exemplo, o aumento dos custos, o cancelamento do projeto, a insatisfação do cliente. Trata-se, desta forma, de ordenar a lista de riscos, tendo em conta a probabilidade de ocorrência e o impacto de cada risco. Atribui-se o nível de probabilidade, que pode ser alto, médio ou baixo. Assim, têm-se os efeitos (consequências) sobre o escopo (o quanto é afetado) e duração (quanto tempo se manifesta).

Risco	Probabilidade	Efeitos
Os problemas financeiros da organização reduzem o orçamento do projeto	Baixo	Catastrófico
Incapaz de recrutar profissionais com as habilidades necessárias	Alto	Catastrófico
Pessoas chave doentes ou não estão disponíveis em momentos críticos	Moderado	Sério
Alterações nos requisitos requerem modificações na codificação	Moderado	Sério
O tempo necessário para desenvolver o processo de Engenharia de Requisitos é subestimado	Alto	Sério
Os clientes não entendem o impacto das mudanças nos requisitos	Moderado	Tolerável

**Tabela 6.** Riscos ordenados por efeito

Risco	Estratégia
Problemas financeiros da organização	Preparar um breve documento para a direção da empresa que mostra que o projeto tem contribuições muito importantes para as metas do negócio.
Problemas de recrutamento	Organizar cursos de capacitação para as pessoas do projeto e/ou investigar a possibilidade de contratar pessoas de outras regiões ou países.
Enfermidades pessoais	Reorganizar a equipe para que eles se sobressaiam ao trabalho e os membros entendem o trabalho dos outros.
Mudança de requisitos	Controlar a informação para avaliar o impacto dos requisitos, maximizar as informações ocultas neles.
Tempo de Engenharia de Requisitos subestimado	Alertar o cliente para as dificuldades potenciais e as possibilidades de atraso.

**Tabela 7.** Estratégias dos Riscos

Tipo de Risco	Identificador
Tecnologia	Atraso na entrega de equipamentos de hardware. Existência de relatórios sobre questões de tecnologia.
Pessoal	Baixa moral dos funcionários, relações ruins entre os membros da equipe.
Organizacional	Rumores. A falta de iniciativa da direção.
Ferramenta	Membros da equipe rejeitam o uso de ferramentas. Reclamações sobre ferramentas Case.
Requisitos	Muitas mudanças nos requisitos. Reclamações de clientes.
Estimação (tempo)	Deixar de cumprir os horários programados.

**Tabela 8.** Indicadores potenciais de Riscos

Na análise de risco se considera cada risco em separado e se avalia em intervalos sua probabilidade e impacto:

- Probabilidade do risco estimado em muito baixa (<10%), baixa (10-25%), moderada (25-50%), elevada (50-75%) ou muito elevada (> 75%);
- Efeitos do risco avaliado como catastrófico, sério, tolerável ou insignificante.

O resultado é armazenado em uma tabela ordenada por probabilidade ou efeito de risco. Feito isso, se escolhe, do total, quais são os mais importantes. Assim, se considera os riscos chaves durante o projeto (deve ser um número administrável).

Por exemplo, todos os riscos graves ou catastróficos com qualquer probabilidade. Observe a Tabela 6.

### **Passo 3: Planejamento de Risco**

Esta etapa tem como objetivo desenvolver uma estratégia para enfrentar os riscos. Se a equipe tem uma abordagem proativa para o risco, a ajuda é sempre a melhor estratégia. Isto é alcançado através do desenvolvimento de planos de mitigação de risco e contingência.

No planejamento dos riscos se considera cada um dos riscos chaves identificados e as estratégicas para administrá-los, o que

será dado pelo julgamento e experiência do gerente de projeto.

Estratégias de prevenção tentam reduzir a probabilidade de ocorrência das estratégias de diminuição de riscos: tentar restringir o seu impacto. Os planos de contingência são desenvolvidos a fim de se preparar no caso de o risco passar a atuar com uma estratégia particular. Observe a Tabela 7.

### **Passo 4: Supervisão dos Riscos**

Consiste em ter um plano de supervisão, dado o caso em que se aceite continuar com o projeto. Indicar que ações e decisões serão executadas quando um problema for identificado.

A supervisão dos riscos acompanha cada um dos riscos identificados para decidir se é provável e quando os seus possíveis efeitos se modificaram. Devem-se controlar os fatores que podem indicar mudanças na probabilidade e impacto. Observe a Tabela 8.

### **Conclusão**

Um adequado processo de engenharia de requisitos possui implicações positivas na qualidade do produto final, assim como na satisfação do cliente. Devido a isto, este processo tem que estar bem definido e ser desenvolvido de forma disciplinada, coerente e repetitiva, garantindo a obtenção de experiências que permitam a aplicação de melhores práticas.

A gestão proativa dos riscos associados aos requisitos de software permite ao gestor adotar, desenvolver e implementar atividades dessa gestão, em função de obter produtos de qualidade que satisfaçam as necessidades dos clientes, mantendo o equilíbrio do prazo e orçamento do projeto. A gestão dos riscos associados com os requisitos, organizadas e geridas através de diferentes propostas pode se tornar uma ferramenta útil para os gestores e equipes de desenvolvimento.

Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros

## A gestão de conhecimento e a TI

As tendências e influências da gestão de conhecimento nos softwares modernos e os softwares de GC nas organizações



**Diógenes Firmiano**

diogenesjf@gmail.com

Gestor Sênior de Engenharia de Software atuando nas áreas de Engenharia de Software, Arquitetura, Metodologia e Desenvolvimento de Sistemas. Formado em Ciência da Computação na Universidade de Brasília, sendo pós-graduado com MBA em Gerenciamento de Projetos tendo feito vários cursos de aperfeiçoamento em sua área de atuação. É professor de disciplinas como Gerência de Projetos, Engenharia de Software, Sistemas de Informação e Programação. Possui experiência profissional de mais de 15 anos em tecnologia da informação sendo especialista em metodologias de software, arquitetura de sistemas e gerência de projetos de software.

Este artigo trata sobre um tema muito abordado nas organizações hoje em dia: a gestão do conhecimento. As necessidades modernas de melhoria de desempenho, aumento de produtividade e rapidez no processo decisório exigem cada vez mais dos colaboradores. A tomada de decisão tem sido uma atividade cada vez mais exigida de toda a empresa e muitas vezes, as melhorias dos processos falham nestas questões. As diretrizes mais acertadas exigem um nível de informações cada vez maior e a gestão deste conhecimento necessário para decidir ganha uma importância cada dia maior. Abordaremos aqui os conceitos iniciais de gestão de conhecimento e as principais interações entre GC e tecnologia.

O mundo moderno gira em torno do conhecimento. A necessidade de conhecer uma infinidade de assuntos e temas é cada vez maior na vida das pessoas. A sociedade está inundada por informações. Porém, nem todas estas

### *Porque esse artigo é útil:*

Gestão do conhecimento é a criação de valor a partir de bens intangíveis da organização. Neste sentido, a gestão do conhecimento inclui a identificação e mapeamento dos ativos intelectuais, para gerar novos conhecimentos e criar vantagem competitiva. No mundo corporativo moderno, a gestão do conhecimento será cada vez mais importante. É necessário saber como lidar com esses novos conhecimentos e tendências na engenharia de software. As organizações como um todo necessitam de conhecimento, principalmente para uma melhor tomada de decisões. A melhor tomada de decisão e o aumento do desempenho das empresas exige e exigirá cada vez mais da integração das informações, geração e gestão de conhecimento. Neste cenário, este artigo apresenta o que é a gestão de conhecimento, de que forma ela vem impactando as organizações e como a tecnologia da informação e a engenharia de software serão impactadas e podem se beneficiar.

informações serão transformadas em conhecimento. Mesmo os conhecimentos que são gerados precisam ser geridos para que seu melhor uso seja realizado.

Saber aproveitar e utilizar, da melhor maneira possível, os conhecimentos, é prioridade das organizações modernas que precisam aperfeiçoar seus negócios. A tecnologia é uma importante aliada neste processo facilitando a gestão do conhecimento. As empresas de tecnologia, no processo de construção de softwares também podem se beneficiar com as técnicas e vantagens da gestão do conhecimento. Mas o que vem a ser gestão do conhecimento?

A Gestão do Conhecimento não é mais uma moda de eficiência operacional. Faz parte da estratégia empresarial. Visando compreender Gestão do Conhecimento, são necessários os conceitos de dado, informação, conhecimento e, por fim, de Gestão do Conhecimento.

No que diz respeito ao conceito de dado, este pode ter significados distintos, dependendo do contexto no qual a palavra é utilizada. Os dados geralmente são caracterizados como elementos em sua forma primária. Para uma organização, dado é o registro estruturado de transações. Dado pode ser definido como um conjunto de fatos distintos e objetivos, relativos a eventos. É informação bruta, descrição exata de algo ou de algum evento. Os dados em si não são dotados de relevância, propósito e significado, mas são importantes porque compõem a matéria-prima essencial para a criação da informação.

Já informações são dados interpretados, dotados de relevância e propósito. Informação é uma mensagem com dados que fazem diferença, podendo ser audível ou visível, e onde existe um emitente e um receptor. É o insumo mais importante da produção humana. É um fluxo de mensagens, um produto capaz de gerar conhecimento. É um meio ou material necessário para extrair e construir o conhecimento. A informação afeta o conhecimento acrescentando-lhe algo ou reestruturando-o.

O conhecimento deriva da informação assim como esta deriva dos dados. O conhecimento não é puro nem simples, mas é uma mistura de elementos, fluido, formalmente estruturado, intuitivo e, portanto, difícil de ser colocado em palavras ou de ser plenamente entendido em termos lógicos.

O conhecimento pode ser entendido dentro de um contexto. As situações e contextos que envolvem problemas e soluções a serem dadas envolvem pessoas e necessidades de raciocínio humano, desta forma, acaba sendo complexo e imprevisível. O conhecimento pode ser comparado a um sistema vivo, que cresce e se modifica à medida que interage com o meio ambiente. Os valores e as crenças integram o conhecimento, pois determinam, em grande parte, o que o conhecedor vê, absorve e conclui a partir das suas observações. O conhecimento, diferente da informação, refere-se a crenças e compromisso. O conhecimento é classificado como tácito e explícito.

O conhecimento tácito é o conhecimento que está contido dentro das pessoas, é difícil de ser expresso em palavras ou escrito. Envolve fatores intangíveis como, por exemplo, crenças pessoais, perspectivas, sistema de valor, insights, intuições, emoções, habilidades. Só pode ser avaliado por meio da ação. Conhecimento explícito é aquele que pode ser escrito ou verbalizado de alguma forma, o que permite que o mesmo seja mais facilmente transmitido, sistematizado e comunicado. Ele

pode ser transmitido formal e facilmente entre os indivíduos. Esse foi o modo dominante de conhecimento na tradição filosófica ocidental.

Os conhecimentos tácito e explícito são unidades estruturais básicas que se complementam e a interação entre eles é a principal dinâmica da criação do conhecimento na organização. Para se tornar uma “empresa que gera conhecimento”, a organização deve completar uma “espiral do conhecimento” (veja a Figura 1). Espiral esta que vai de tácito para tácito, de explícito a explícito, de tácito a explícito, e finalmente, de explícito a tácito. Logo, o conhecimento deve ser articulado e então internalizado para tornar-se parte da base de conhecimento de cada pessoa. A espiral começa novamente depois de ter sido completada, porém em patamares cada vez mais elevados, ampliando assim a aplicação do conhecimento em outras áreas da organização.



Figura 1. Espiral do conhecimento

A espiral do conhecimento prevê quatro formas de aprendizado e assimilação de conhecimento: socialização, externalização, combinação e internalização. Socialização é o compartilhamento do conhecimento tácito, por meio da observação, imitação ou prática (tácito para tácito). Externalização/Articulação é a conversão do conhecimento tácito em explícito e sua comunicação ao grupo (tácito para explícito). Combinação ou padronização do conhecimento é juntá-lo em um manual ou guia de trabalho e incorporá-lo a um produto (explícito para explícito). Internalização é quando novos conhecimentos explícitos são compartilhados na organização e outras pessoas começam a utilizá-los para aumentar, estender e reenquadrar seu próprio conhecimento tácito (explícito para tácito).

Gestão do conhecimento também pode ser entendida como sendo a criação de valor a partir de bens intangíveis da organização. Neste sentido, a gestão do conhecimento inclui a identificação e mapeamento dos ativos intelectuais, para gerar novos conhecimentos e criar vantagem competitiva. O Gartner Group, por sua vez, define a gestão do conhecimento

como sendo uma disciplina que promove, com visão integrada, o gerenciamento e o compartilhamento de todo o ativo de informação possuído pela empresa, podendo este estar localizado em um banco de dados, em documentos, em procedimentos, bem como em pessoas, através de suas experiências e habilidades.

As ferramentas de gestão do conhecimento pretendem auxiliar o processo de coleta e estruturação do conhecimento de grupos de indivíduos, disponibilizando esse conhecimento em uma base compartilhada por toda a organização.

A fim de desenvolver os sistemas de conhecimento, é necessário ter foco externo (benchmarking de outras organizações), tecnologias facilitadoras (groupware), gestão de desempenho (mensuração, recomendação, recompensas para equipes, obrigações contratuais) e gestão de pessoas (equipes virtuais, coordenadores de conhecimento, busca do perfil do disseminador do conhecimento).

Somente a organização pode oferecer a continuidade básica de que os trabalhadores do conhecimento precisam para serem eficazes. Apenas a organização pode transformar o conhecimento especializado do trabalhador em desempenho. Porém, avançou-se muito pouco sobre como se deveria gerenciar o conhecimento.

### A gestão de conhecimento nas organizações

A gestão do conhecimento é um processo corporativo focado na estratégia empresarial e que envolve a gestão das competências, a gestão do capital intelectual, a aprendizagem organizacional, a inteligência empresarial e a educação corporativa.

### Gestão das Competências

Competência engloba o conhecimento real, a habilidade, a experiência, os julgamentos de valor e as interações sociais. É o elo entre conhecimento e estratégia. Não pode ser copiada com exatidão, sendo transferida pela prática. Competência é a capacidade de gerar resultados observáveis, necessários ao alcance dos objetivos definidos no posicionamento estratégico da empresa e no propósito de manutenção e desenvolvimento de sua ideologia essencial. É um conceito pelo qual se define quais são as atitudes, as habilidades e os conhecimentos necessários para alcançar resultados diferenciados, o conjunto de qualificações que a pessoa tem para executar um trabalho com um nível superior de performance. Está associada às realizações das pessoas, àquilo que elas produzem e/ou entregam.

Desta forma, a competência não é um estado ou um conhecimento que se tem e nem é resultado de treinamento. Competência é, na verdade, colocar em prática o que se sabe em um determinado contexto. Há competência apenas quando há competência em ação. A gestão de recursos humanos está em amplo processo de transformação. Sistemas que tradicionalmente são utilizados como referencial – centrados em cargos – vêm mostrando sua fragilidade em articular sistematicamente as necessidades de gestão das organizações. A abordagem de gestão de pessoas, que tem no seu núcleo o conceito de

competência, apresenta imensas possibilidades de articular as relações entre as diferentes ações de gestão de RH, como a conjugação de desempenho, desenvolvimento e potencial, aumentando a sinergia do sistema.

### Gestão do Capital Intelectual

Capital intelectual é a soma do conhecimento de todos em uma organização; é a capacidade mental coletiva, a capacidade de criar continuamente e proporcionar valor de qualidade superior. É criado a partir do intercâmbio entre:

- Capital humano - capacidade organizacional que uma empresa possui de suprir as exigências do mercado. Está nas habilidades dos colaboradores, em seus conhecimentos tácitos e nos obtidos nas suas informações profissionais, na busca permanente de atualizar seus conhecimentos, nas informações alcançáveis, nas informações documentadas sobre clientes, concorrentes, parceiros e fornecedores. Essencialmente diz respeito às pessoas, seu intelecto, seus conhecimentos e experiências;
- Capital estrutural – soma das patentes, processos, manuais, marcas, conceitos, sistemas administrativos, bancos de dados disponibilizados, tecnologia, estruturas organizacionais;
- Capital de clientes – é o valor dos relacionamentos de uma empresa com as pessoas e fornecedores com os quais faz negócios.

**Não perca tempo  
reinventando a roda!**

**COBREBEMX**

**Componente completo para sua  
Cobrança por Boleto Bancário  
e Débito em Conta Corrente**

**Mais de 40 exemplos  
em diversas linguagens  
de programação**

**Geração e leitura de arquivos  
(remessa e retorno) nos padrões  
FEBRABAN e CNAB**

**Testes e Downloads  
gratuitos em nosso site**



ACESSE E CONHEÇA O COMPONENTE EM:  
**WWW.COBREBEM.COM**

O capital intelectual se encontra em três formas e em três lugares distintos:

- a) Na forma de conhecimento dentro da cabeça de cada pessoa;
- b) Na forma de conhecimento adicional que é gerado quando as pessoas se relacionam e compartilham o seu conhecimento;
- c) Na forma de conhecimento armazenado em livros, revistas, jornais, fotografias, desenhos, fitas, discos, CD, CD ROM, bases de dados, etc.

A introdução do conceito de capital digital – o capital intelectual digitalizado – é consequência do uso da internet, que permite que as pessoas se contatem, se relacionem e compartilhem seu conhecimento (capital intelectual individual) independentemente da distância e armazenem e disponibilizem na internet boa parte do conhecimento de que dispõem. Esses fatos fizeram com que o conhecimento que as pessoas produzem e a que têm acesso, uma vez digitalizado e disponibilizado na internet, aumente exponencialmente, produza uma explosão de novas ideias, fundamentando uma nova forma de gerar riqueza.

### **Aprendizagem organizacional**

O tema central da gestão do conhecimento é aproveitar os recursos que já existem na organização para que as pessoas procurem, encontrem e empreguem as melhores práticas em vez de tentar criar algo que já havia sido criado. Cuida de agregar valor às informações, filtrando, resumindo e sintetizando-as e, dessa forma, desenvolvendo um perfil de utilização pessoal que ajuda a chegar ao tipo de informação necessário para passar à ação. Através da aprendizagem contínua, a organização exerce a sua competência e inteligência coletiva para responder ao seu ambiente interno (objetivos, metas, resultados) e externo (estratégia). Nas “organizações que aprendem”, as pessoas expandem continuamente sua capacidade de criar resultados que elas realmente desejam, onde maneiras novas e expansivas de pensar são encorajadas, onde a aspiração coletiva é livre, e onde as pessoas estão constantemente aprendendo a aprender coletivamente”.

A organização que aprende possui a capacidade de, continuamente, criar o futuro que realmente deseja. Para isto, reflete sobre o desempenho atual e os fatores que o geram, pensa sobre os diversos futuros possíveis e qual entre eles é o desejado, e planeja e programa as ações para se mover da situação atual para a desejada. A aprendizagem organizacional é, portanto, o processo contínuo de detectar e corrigir erros. Errar significa aprender, envolvendo a autocrítica, a avaliação de riscos, a tolerância ao fracasso e a correção de rumo, até alcançar os objetivos. É a capacidade das organizações em criar, adquirir e transferir conhecimentos e em modificar seus comportamentos para refletir estes novos conhecimentos e insights.

Isso implementa um mecanismo pelo qual os colaboradores contribuem para o desempenho da empresa por meio da aplicação dos seus conhecimentos e habilidades em resolver problemas e inovar. Então, é criada a organização que aprende e que gera conhecimento. No entanto, nenhuma mudança

organizacional significativa pode ser realizada sem que se efetuem profundas mudanças nas formas de pensar e interagir das pessoas. A base de ideias que sustenta as “organizações que aprendem” estabelece o pensamento sistêmico, os modelos mentais, o domínio pessoal, a visão compartilhada, a aprendizagem em grupo e o diálogo como elementos inevitáveis do seu desenvolvimento, cada um proporcionando uma dimensão vital na construção de organizações realmente capazes de “aprender”, de ampliar continuamente sua capacidade de realizar suas mais altas aspirações.

### **Educação corporativa**

As organizações enfrentam dificuldades na velocidade de sua adaptação à nova realidade. O desconhecimento do futuro, a permanência de traços da cultura anterior, momentos de dificuldades empresariais, a descrença em uma visão compartilhada de futuro, a falta de capacitação para os novos papéis e a “falta de tempo” são alguns dos fenômenos que fazem parte das dificuldades percebidas. Esse novo contexto empresarial redefine o perfil do colaborador da era do conhecimento. São necessários profissionais que aprendam de forma não convencional e que saibam trabalhar cooperativamente para gerar soluções inovadoras.

É necessária uma nova abordagem na formação para que as pessoas permaneçam produtivas, em condições de acompanhar as mudanças e otimizando seu tempo. Além disso, a tecnologia vem permitindo desenvolver experiências para treinar mais pessoas com maior economia. As empresas começam a perceber a necessidade de transferir o foco dos esforços de treinamento e educação de eventos em sala de aula, cujo objetivo é desenvolver qualificações isoladas, para a criação de uma cultura de aprendizagem contínua, em que os colaboradores aprendem uns com os outros e compartilham inovações e melhores práticas visando solucionar problemas organizacionais reais.

O foco do treinamento vai além do empregado isoladamente para o desenvolvimento da capacidade de aprendizado da organização, criando situações que permitam a discussão de problemas comuns e soluções por meio da aprendizagem coletiva. Os ambientes de aprendizagem passam a ser reestruturados de forma a tornarem-se proativos, centralizados, determinados e estratégicos e o resultado esperado é o “aprender fazendo”, desenvolvendo a capacidade de aprender e dar continuidade a esse processo na volta ao trabalho. Outra forma de educação surgiu, aproveitando as facilidades proporcionadas pela melhoria crescente nos cursos por correspondência gerando o ensino à distância – EAD, que pode ser definido como qualquer interação entre estudante e instrutor, onde os participantes são separados pela distância, pelo tempo, ou por ambos. A aprendizagem torna-se, além de contínua, flexível para poder responder às necessidades do momento.

### **Interações entre gestão de conhecimento e tecnologia**

O sucesso dos negócios está ficando cada vez mais dependente da inovação e do conhecimento, que estão mudando

as formas tradicionais de organizar negócios nas empresas. As suposições tradicionais em coordenação, controle e apropriação de recursos estão perdendo sua relevância, e as habituais formas de administrar as organizações estão se tornando inadequadas.

Com o surgimento da Tecnologia da Informação (TI), novas formas de organização estão emergindo e a importância de redes informais dentro e entre organizações está se tornando amplamente concebida. Na rede da sociedade do conhecimento, as empresas terão novos tipos de profissionais, considerados experts, e os gerentes de negócio precisam estar cientes de que deverão entender de diferentes sistemas de valor. Essas mudanças implicam no design organizacional, na estratégia, nas práticas de gestão e tecnologias organizacionais, conduzindo às novas teorias e práticas de gestão do conhecimento (GC).

Novas informações, comunicação e tecnologias computacionais estão mudando fundamentalmente a organização e o conteúdo de trabalho, enfatizando que pelos menos para alguns membros da sociedade, carreiras de trabalho de longa vida estão se tornando um mosaico, onde o trabalho produtivo, o aprendizado e o desenvolvimento de competência são inseparáveis.

Esse quadro de integração do trabalho produtivo, aprendizado e competências, gera para a organização vantagens competitivas e oportunidades de melhorar a estratégia empresarial e a integração entre a GC e TI aparece como um imperativo para atender esses requisitos que são importantes para toda empresa. Contudo, essa integração é extremamente complexa, pois envolve tanto a gestão de ativos intangíveis de diferentes naturezas – pessoas, conhecimentos tácitos, explícitos, individuais, organizacionais e de redes – quanto conhecimentos estruturais, que servem de base tecnológica para estocagem, para melhoria e para o fluxo dos bens intangíveis, e sistemas de informação com aplicativos que possibilitem o aumento da interação entre pessoas nos ambientes interno e externo, agregando fornecedores e clientes à cadeia de valor das organizações.

Diante dessa complexidade, a TI tem se tornado o centro nervoso das empresas, um fator estratégico de competitividade e sobrevivência. Embora essa afirmação esteja correta, as empresas precisam ter precaução para que não cometa o erro de considerar a TI, em si, como a solução para o sucesso das organizações. Algumas organizações, equivocadamente, presumiram que a tecnologia poderia substituir a qualificação e o julgamento de um trabalhador humano experiente, o que se tem revelado falso. A informação se movimenta pelas organizações por redes hard e soft. As redes hard têm uma infraestrutura definida, formada por fios, utilitários de entrega, antenas parabólicas, centrais de correio, endereços, caixas postais eletrônicas, etc.. Envolvem conhecimento estruturado, qualificações técnicas e experiência profissional. Já as redes soft são menos formais e visíveis; são circunstanciais; envolvem um claro senso dos aspectos culturais, políticos e pessoais do conhecimento e da transferência do

conhecimento. A integração adequada entre ambas é o que permite o bom posicionamento da empresa no mercado, sua resposta acertada às demandas.

O objetivo das ferramentas de GC é modelar parte do conhecimento existente nas cabeças das pessoas e nos documentos corporativos, disponibilizando-o para toda a organização. A mera existência do conhecimento na empresa é de pouco valor se ele não estiver acessível e não for utilizado como um dos seus recursos mais importantes. Com essas ferramentas, almeja-se que o conhecimento possa fluir por meio de redes de comunidades, transformando a tecnologia em um meio e o conhecimento em um capital, em uma mensagem.

Conforme dito anteriormente, o conceito de gestão do conhecimento pode assumir diversos significados. Com o advento da TI e com o avanço nas práticas de gestão organizacional, a GC tem sido entendida sob a forma de diferentes estratégias, a partir das quais as organizações lidam com o conhecimento, interna e externamente, para obter vantagens competitivas.

A GC pode ser sintetizada como um processo articulado e intencional, destinado a sustentar ou promover o desempenho global de uma organização, tendo como base a criação e a circulação de conhecimento.

Existe uma forte tendência em correlacionar a GC com desempenho organizacional em sentido amplo, o que tem sido comprovado em diversas pesquisas realizadas com métodos diferentes, junto a empresas de distintos perfis, portes e locais. Três estudos recentes a esse respeito feitos por amostragem quantitativamente significativa, detalhando as relações entre GC, TI e desempenho organizacional, apontam-nas como positivas.

O primeiro desses estudos descreve como empresas alemãs utilizam a GC. Este estudo investigou o estado-da-arte dos sistemas de gestão do conhecimento (Knowledge Management System – KMS) utilizados nas 500 maiores firmas da Alemanha e nas 50 mais importantes companhias dos setores bancário e de seguros do país, sendo que para esse estudo foram enviados questionários, que revelaram que a GC, embora pareça absorver todos os tipos de abordagens teóricas e práticas, muitas vezes não leva em consideração o valor estratégico ou de negócios. Sendo assim, os autores concluíram que “os esforços de GC das organizações respondentes, em média, ainda têm elementos para ir até os mais avançados benefícios que podem ser colhidos”, e concluíram também que a “(...) GC, na prática, parece ser um esforço que comprehende todos os tipos de atividades, medidas e tecnologias”.

As estratégias de GC são afetadas pelo uso de tecnologias de informação e comunicação que influenciam todos os níveis de intervenção. A cultura corporativa subjacente, a estrutura organizacional, as funções e processos, classificados como tecnologias que suportam a GC permitem inferir que as tecnologias de informação utilizadas são, de alguma forma e em alguns casos, confundidas com estratégias de GC ou que estratégias têm dimensão secundária como ações integrativas em um sistema de GC.

O segundo estudo analisa as relações entre GC e estratégia, GC e gestão de competências e GC e resultados, em 99 empresas brasileiras a partir de um amplo referencial teórico. Entre os 16 conceitos de GC revisados pela autora, a maioria traduz a expressão como processos, fluxos ou redes de valor, para que as empresas possam chegar a uma posição superior em relação à concorrência. Ela adotou o conceito de Wegmann (1997), que considera a GC como um processo contínuo, relacionado à criação de valor em uma cadeia de produção, o que segundo esse autor, confere-lhe dinamicidade.

Entre as principais conclusões deste estudo, destaca-se que em muitas organizações as iniciativas para a GC parecem iniciativas isoladas de áreas funcionais, não sendo raro observar esforços de departamento de recursos humanos ou de tecnologia da informação tentando desenvolver ou implementar projetos de GC, ou seja, a cúpula da maior parte das empresas ainda não apoia as iniciativas de GC, nem tampouco os empregados compreendem o que significa GC. Acrescenta ainda que a potencial contribuição da TI não chega a ser um destaque, e restringe-se ao desenvolvimento e uso dos registros dos ativos intelectuais e dos sistemas de processamento. A autora concluiu que “(...) as empresas brasileiras ainda possuem poucas práticas que relacionam a gestão do conhecimento à estratégia empresarial, gestão de competências e de resultado...”. Inferiu também que essas empresas não enfrentaram obstáculos que se impõem à implementação da GC.

O terceiro estudo teve como objetivo verificar até que ponto as organizações adotam práticas de GC positivamente relacionadas com desempenho organizacional e se desempenho organizacional é, em troca, positivamente relacionado a desempenho financeiro. Foi elaborado um quadro com 12 práticas de GC, a partir de 30 pesquisas anteriores, que também relacionaram GC com desempenho organizacional. Pode-se afirmar que a base conceitual dos autores está assentada em uma revisão bibliográfica que permite identificar, com expressivo grau de profundidade, os problemas associados à implantação das estratégias de GC. Também foi elaborado um questionário para estudar as relações entre GC e desempenho organizacional em empresas do Canadá e Estados Unidos. Com base nos resultados obtidos, concluíram que a GC deixou de ser um conceito emergente para tornar-se uma função comum nas organizações. Mesmo assim, indicam que na prática ainda há problemas a resolver. Mencionam que a cultura talvez seja o fator mais influente na promoção ou inibição de práticas de GC.

A partir desses estudos foi confirmado o vínculo entre GC e desempenho organizacional, já referido por pesquisas quantitativas e qualitativas, e certa relação positiva, ainda que fraca, entre desempenho organizacional e desempenho financeiro. As práticas de GC estão diretamente relacionadas a várias medidas intermediárias de desempenho organizacional estratégico (proximidade do consumidor, liderança em produto e excelência operacional) e essas medidas intermediárias estão associadas ao desempenho financeiro.

## Impactos da tecnologia na gestão de conhecimento

A TI tem papel importante na gestão do conhecimento como habilitadora de processos de negócios que visam criar, armazenar, disseminar e aplicar conhecimento.

A tecnologia, se bem aplicada, pode ser uma valiosa habilitadora aos objetivos da gestão do conhecimento. Pode-se dizer que a finalidade de uma ferramenta tecnológica de apoio à gestão do conhecimento não é, por si só, gerenciar conhecimento, mas facilitar a implementação das atividades ou processos do conhecimento, tais como, criar ou adquirir conhecimento, armazenar, utilizar, compartilhar, etc. As tecnologias podem ser usadas para capturar conhecimento tácito, acelerar e facilitar comunicações, conectar pessoas, clarificar suposições, construir e catalogar ideias e sugestões; e, em outros casos, podem também automatizar certos tipos de trabalhos do conhecimento. Mas, de forma geral, o papel da tecnologia é puramente habilitador, sendo sempre de responsabilidade dos gestores definirem e gerenciarem as atividades de conhecimento.

A informação é uma das ferramentas para transferência do conhecimento que pode ser associada a um software para gerenciamento destas informações. Os softwares destinados à transferência do conhecimento devem ter filtros, que identificam seus usuários através das senhas de acesso, liberando ou não, certas informações, de acordo com o perfil de cada cargo. Este filtro é chamado de colaboradora ou social. Outro filtro é o psicológico, onde o software identifica o usuário através de seu perfil. E por último o filtro adaptativo, onde o software é capaz de “aprender” de acordo com o perfil e preferências dos usuários.

Para desenvolver um programa que facilite o acesso a informações, deve-se considerar a importância da organização, e da formação de uma equipe composta por tecnólogos, especialistas em informação e gerentes do conhecimento, como também as consultorias que vendem tecnologias. Estas equipes trabalharão com usuários chaves do sistema, colhendo as informações necessárias para desenvolver o perfil de cada colaborador que irá utilizar o programa. A mente humana é a fonte de dados para alimentar um sistema, portanto as pessoas que mais detêm as informações são aquelas interessantes no projeto de gestão do conhecimento, representando um elo entre a tecnologia e o indivíduo.

Assim, podemos entender que os sistemas de informação podem promover aprendizagem organizacional capturando, codificando e distribuindo tanto o conhecimento explícito quanto o tácito. Tendo a informação sido coletada e organizada em um sistema, pode ser utilizado muitas vezes, o que possibilita a preservação do conhecimento como memória organizacional.

É dentro deste contexto que surge o Knowledge Management Systems (KMS), entendido como o Sistema de Gestão do Conhecimento (SGC). A literatura caracteriza o KMS como sendo uma maneira mais eficiente e sistemática de gerenciar conhecimento, que envolve a utilização de tecnologia de informação e outros recursos organizacionais para gerenciar o

conhecimento estrategicamente. Dos anos 90 até os dias atuais, a evolução e a disseminação da internet e das intranets têm concentrado as principais aplicações da TI para a gestão do conhecimento. Esta evolução traz novas funcionalidades, o que veio a facilitar sua interatividade tanto para uso individual quanto em grupo. Exemplos dessa integração são as ferramentas de apoio de trabalho em grupo, suportadas por mecanismos de gerenciamento de documentos eletrônicos, ferramentas de navegação inteligente na internet, etc.

Os recursos de TI facilitam o trabalho em rede, podendo manter os conhecimentos descentralizados junto aos locais em que são mais gerados e/ou utilizados, melhorando o grau e interatividade do usuário com os registros de conhecimentos. Contudo, para a efetividade da TI para a gestão do conhecimento, esta deverá ser empregada utilizando-se a interatividade dos sistemas com a atividade humana.

O emprego de sistemas/tecnologias de informação na empresa deve ser condicionado às definições e escolhas da estrutura organizacional e não o contrário, cabendo a estes o papel de facilitadores, contribuindo com o tratamento e transmissão do conhecimento. A utilização de TI focalizada na internet/intranets para a GC representa também a adoção de uma tecnologia base de padrões abertos e universais, o que facilita a integração com outros sistemas internos ou externos à empresa, resultando em uma tendência recente que é a formação de portais com o objetivo de centralizar o acesso intranet da empresa e a sites relacionados ou de interesse da empresa na internet.

As intranets corporativas, bem como outros sistemas além do ERP, podem permitir o gerenciamento dos conteúdos de conhecimentos da empresa. Porém, uma importante parte da solução, para que estes sistemas funcionem, passa por se estabelecer muito bem quais são os processos-chave e os principais papéis e fluxos de trabalho (workflow) dentro desses processos.

O papel principal da TI na gestão do conhecimento é ampliar o alcance e acelerar a velocidade de transferência do conhecimento. Contudo, a TI fornece apenas a infraestrutura para as atividades de gestão do conhecimento. Uma ferramenta de GC pode ser definida como sendo um tipo específico de software que oferece suporte a, pelo menos, uma das atividades de geração, codificação ou transferência de conhecimento.

A tipologia das ferramentas de GC pode ser baseada em quatro parâmetros: funcionalidade essencial, processo da

gestão do conhecimento, tipo de conhecimento e área de origem dos conceitos. A funcionalidade essencial é a característica principal da ferramenta, dando assim o nome para a categoria. O processo da gestão do conhecimento refere-se a um ou mais de um dos processos de geração, codificação e transferência. O tipo de conhecimento indica para qual conhecimento (explícito, tácito ou ambos) a ferramenta trabalha. A área de origem dos conceitos refere-se aos campos de conhecimento implementados na categoria da ferramenta. Assim, classifica as ferramentas de gestão do conhecimento em seis categorias principais: ferramentas voltadas para a intranet, sistemas de gerenciamento eletrônico de documentos, sistemas de groupware, sistemas de workflow, sistemas de mapas do conhecimento (data warehousing) e sistemas para descoberta de conhecimento em bases de dados (data mining). Estas ferramentas pretendem auxiliar no processo de geração, codificação e transferência do conhecimento de grupos de indivíduos, disponibilizando este conhecimento para toda a instituição. Observe a **Tabela 1**.

### **Benefícios da gestão de conhecimento na engenharia de software**

A engenharia de software é uma atividade orientada ao conhecimento e envolve várias pessoas trabalhando em diferentes fases e atividades. O conhecimento na engenharia de software é disperso, de proporção imensa e de crescimento contínuo. Algumas necessidades de conhecimento de organizações ligadas ao desenvolvimento de software são:

1. Aquisição de conhecimento sobre novas tecnologias: a necessidade de monitoração do ambiente em busca de novas tecnologias é constante nas organizações que desenvolvem software;
2. Acesso a novos domínios de conhecimento: o desenvolvimento de software não envolve só conhecimento sobre o software ou as tecnologias relacionadas. Envolve também o domínio do conhecimento do campo para o qual o software está sendo desenvolvido como, por exemplo, conhecimento médico no caso de um software para a área de medicina;
3. Compartilhamento do conhecimento sobre políticas e práticas institucionais: novos membros de uma organização precisam conhecer sobre a cultura da organização, assim como a infraestrutura de trabalho e as práticas institucionais;

Categoria	Processos	Tipo(s) de Conhecimento	Áreas de Origem
Ferramentas voltadas para a Intranet	Codificação e transferência	Explícito e tácito	Redes de computadores
Sistemas de gerenciamento eletrônico de documentos	Codificação e transferência	Explícito	Ciência da informação
Sistemas de groupware	Geração, codificação e transferência.	Explícito e tácito	CSCW (Trabalho cooperativo apoiado por computador)
Sistemas de workflow	Codificação e transferência	Explícito e tácito	Organização e métodos
Sistemas de mapas do conhecimento	Geração, codificação e transferência.	Explícito e tácito	Ciência da informação e gestão do conhecimento
Sistemas para descoberta de conhecimento em bases de dados	Geração, codificação e transferência.	Explícito	Estatística e Inteligência artificial

**Tabela 1.** Parâmetros de Gestão do Conhecimento

4. Captura de conhecimento e saber quem faz o quê: o indivíduo é o ponto chave para o sucesso de qualquer projeto de engenharia de software. Saber o tipo de conhecimento possuído por cada empregado é indispensável na criação de uma estratégia que previna o desaparecimento de conhecimentos valiosos;

5. Colaboração e compartilhamento do conhecimento: a colaboração está relacionada com a troca mútua de conhecimento. Membros de uma equipe de desenvolvimento de software precisam de um meio de colaboração e troca de conhecimento independente de tempo e espaço.

A gestão do conhecimento requer esforço e alocação de recursos. No modelo chamado de "A Organização Fábrica de Experiência", não são os desenvolvedores que externalizam o conhecimento e sim uma equipe dedicada a esta finalidade. O conceito base da Organização Fábrica de Experiência (OFE) é que os projetos de desenvolvimento de software podem obter melhores resultados se alavancados por experiências de projetos anteriores. Com cronogramas, expectativas quanto à qualidade e produtividade, e desafios técnicos, a maioria dos projetos não pode dedicar recursos suficientes para explicitar o conhecimento. Contudo, esta atividade fica com a equipe chamada de fábrica de experiência. Esta equipe analisa e sintetiza todos os tipos de

experiência, incluindo lições aprendidas, dados de projetos, relatórios e explicitam estas experiências através da criação de repositórios.

A fábrica de experiência agrega valor ao conhecimento através da criação de modelos baseados em documentos ou indivíduos. As atividades de externalização e internalização são integradas de modo que a equipe do projeto trabalha em harmonia com a fábrica de experiência. A implantação da OFE envolve mudanças culturais na organização, devido à criação de equipes e processos distintos de trabalho. A essência da OFE não é a experiência e sim o novo conhecimento gerado a partir da experiência.

Na pesquisa de DINGOSOYR, metade das instituições consultadas utiliza alguma implementação semelhante à OFE. Esta pesquisa observa a presença de estratégias de gestão de conhecimento como personalização e codificação em todas as instituições pesquisadas. Observa-se a utilização da estratégia de codificação em práticas como: transferência de conhecimento entre projetos no sentido de resolver problemas; prevenção do retrabalho de ter que explicar a várias pessoas sobre a mesma solução técnica; e melhora no ambiente de trabalho dos desenvolvedores através de dicas de melhor configuração das ferramentas técnicas. Já a estratégica de personalização é mais utilizada em: busca por competências para solucionar problemas técnicos; alocação

# DÊ UM SALTO EM CONHECIMENTO!



Acesse o maior  
portal para  
desenvolvedores  
da América  
Latina!

20  
mil  
posts

430  
mil  
cadastrados

10  
milhões de  
page-views  
por mês

de recursos; e para o desenvolvimento de competências. O autor constata, ainda, que diferentes grupos de empregados utilizam diferentes ferramentas de gestão do conhecimento. Os desenvolvedores demandam por conhecimento mais detalhado enquanto que os outros grupos preferem o conhecimento mais abstrato.

## Conclusão

O conhecimento tem se tornado fundamental na vida das pessoas. Vivemos em uma era do conhecimento. A importância de realizar uma boa gestão do conhecimento nas organizações modernas tem sido cada vez maior. Neste processo, faz-se necessário compreender os diversos processos e metodologias necessários à gestão do conhecimento. A tecnologia da informação têm se tornado uma importante ferramenta para otimização da gestão de conhecimento e tem auxiliado na implementação dos processos e na metodologia de gestão de conhecimento.

A engenharia de software pode se beneficiar das técnicas de gestão de conhecimento. O aprendizado e a utilização de Organização de Fábricas de Experiências podem aperfeiçoar os processos de construção de softwares melhorando cada vez mais a atuação nos projetos de software. Ela também pode se beneficiar pelos diversos produtos de software que podem ser utilizados para auxiliar na gestão de conhecimento orientado aos projetos de construção de software.

Finalmente, é importante verificar que, dentro da visão de futuro de gestão de conhecimento, a construção de um software orientado à integração dos conceitos de produtos orientados à gestão de conhecimento é uma forte tendência. O software integrador de conhecimento será uma tendência de futuro para as organizações e deverá ser um desafio para a tecnologia da informação uma vez que envolve diversos conceitos inovadores e desafiante ao profissional de tecnologia. Além disso, os próprios profissionais poderão se utilizar dos softwares de integração para melhoria do processo de engenharia de software e otimização dos resultados nos projetos de software.

### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



### Links e Referências:

**ALBERTIN, A. L. (1996).** Aumentando as chances de sucesso no desenvolvimento e implementação de sistemas de informações. *Revista de Administração de Empresas*, São Paulo, v.36, n.3, p.61-69.

**ALAVI, M.; LEIDNER, D.** Review: Knowledge management and knowledge management systems: conceptual foundations and research issues. *MIS Quarterly*, v.25, n. 1, p. 107-136, Mar. 2001.

**ALTER, S.: Information Systems: a management perspective.** Addison-Wesley Publishing Co. Massachusetts, 1992.

**BALARINE, O. F. O. (2002).** Gestão da informação: tecnologia da informação como vantagem competitiva. *Revista de Administração de Empresas – eletrônica*, São Paulo, v.1, n.1.

<http://rae.com.br/eletronica>

**BASILI, V.R.,** "Software Development: A Paradigm for the Future," Proceedings. 13th International Computer Software and Applications Conference (COMPSAC 89) IEEE CS Press, Los Alamitos, California, 1989, p. 471-485.

**BENETT, Gordon.** Intranets: Como Implantar com Sucesso na sua Empresa. Rio de Janeiro: Campus, 1997.

**BERGER, P. L.; LUCKMANN, T.** The social construction of reality. New York: Anchor, 1966.

**BEUREN, I.M. (2000).** Gerenciamento da informação: um recurso estratégico no processo de gestão empresarial. 2. ed. São Paulo: Atlas.

**CRUZ, TADEU.** Workflow: A Tecnologia que vai Revolucionar Processos. São Paulo: Atlas, 1998.

**DAVENPORT, Thomas H.; PRUSAK, Laurence.** Conhecimento Empresarial: como as organizações gerenciam o seu capital intelectual. Rio de Janeiro: Campus, 1998.

**DELPHI GROUP.** Enterprise Portals Shape Emerging Business Desktop  
<http://www.delphigroup.com>

**DINGOSOYR, Torgeir.** Knowledge Management in Medium-Sized Software Consulting Companies. Trondheim: Norwegian University of Science and Technology, 2002. 256p. (Tese, Doutorado em Ciência da Computação).

**DRUCKER, Peter.** Desafios Gerenciais para o Século XXI. São Paulo: Pioneira, 1999.

**DRUCKER, Peter.** Sociedade pós-capitalista. São Paulo: Pioneira, 1993.

**FALCÃO, S. D.; BRESCIANI FILHO, E.** Gestão do conhecimento. *Revista da III Jornada de Produção Científica das Universidades Católicas do Centro-Oeste, Goiânia*, v. 2, set. 1999.

Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros

## Gerência de variabilidades em linhas de processos

Identificando e representando variabilidades com a abordagem SMartySPEM



**Edson A. Oliveira Junior**

[edson@din.uem.br](mailto:edson@din.uem.br)

Professor de Engenharia de Software na Universidade Estadual de Maringá (UEM). Doutor em Ciência da Computação pelo Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP). Principais temas de pesquisa: linha de processo de software, linha de produto de software, arquitetura de software e avaliação, métricas, engenharia de software experimental, modelos e metamodelos UML.



**Maicon Giovane Pazin**

[maicongazin@gmail.com](mailto:maicongazin@gmail.com)

Graduado em Informática pela Universidade Estadual de Maringá. Trabalha com sistemas para Web, com experiência em tecnologias como PHP, Javascript, CSS, HTML5 e Java.



**Sandra Ferrari**

[sferrari@din.uem.br](mailto:sferrari@din.uem.br)

Professora de Engenharia de Software na Universidade Estadual de Maringá (UEM). Doutora em Engenharia de Produção pela Universidade Federal de Santa Catarina. Principais temas de pesquisa: linha de produto de software e qualidade de software.

Durante o desenvolvimento de produtos de software, a definição de processos de software adaptados a determinados ambientes de produção é um importante fator para a redução de custos e melhoria da qualidade do produto final. Consequentemente, existe a necessidade de uma rápida e efetiva customização de processos de software que englobe a variedade de cenários, tecnologias, culturas e escalas envolvidas em cada novo projeto. Dessa forma, fica clara a necessidade de técnicas e mecanismos que auxiliem o processo de adaptação de processos que atendam a determinados domínios de aplicação. Recentemente diversos trabalhos relacionados a Linhas de Processo de Software (LPrS) vêm sendo desenvolvidos para auxiliar nessa demanda.

Entende-se por LPrS como um família de processos com um conjunto gerenciado de características que satisfazem necessidades específicas de uma organização e que são desenvolvidos

### Porque esse artigo é útil:

Neste artigo será apresentada a abordagem SMartySPEM que propõe uma formalização para representação de variabilidades em linhas de processo de software definidas com SPEM. Para isso, o SMartySPEM introduz um mecanismo baseado na abordagem SMarty para representar variabilidades em elementos de processos definidos pelo SPEM. Esse tema é útil para gerentes, arquitetos, estudantes e engenheiros de software interessados em aprofundar seus conhecimentos sobre métodos e técnicas para gerenciamento de variabilidades em processos de software.

a partir de um conjunto de processos básicos comuns. Para isso, é necessário identificar as similaridades e gerenciar as variabilidades dos elementos de processos, a fim de promover vantagens na adoção da abordagem.

Atualmente, o *Software & Systems Process Engineering Metamodel* (SPEM) é a notação padrão da OMG para modelagem de processos de desenvolvimento de software e seus componentes. Apesar de o SPEM propor

uma maneira para gerenciamento de variabilidades entre seus elementos de processos, diversos estudos contestam esses mecanismos. Neste artigo é apresentada a abordagem *Stereotype-based Variability Management for SPEM* (SMartySPEM) que busca introduzir um novo mecanismo para identificação e representação de variabilidades em elementos de processos de software baseados no SPEM.

## Modelagem de Processos de software com SPEM

O *Software and System Process Engineering Metamodel* (SPEM) é um metamodelo especificado pela OMG que suporta a modelagem, apresentação, gerenciamento, intercâmbio e definição de métodos e processos de desenvolvimento de software. A especificação formal da versão 2.0 do SPEM é dividida em duas partes:

- O SPEM 2.0 *Metamodel*, que define todas as regras de estruturação, especificadas como um modelo MOF e reutiliza algumas classes fundamentais da UML 2. Também define a notação de diagramas de processo específicos;
- O perfil do SPEM 2.0, que define um conjunto de estereótipos da UML 2. Tal definição abrange apenas sua representação, tornando-se dependente do SPEM 2.0 *Metamodel* para as declarações semânticas e de restrições.

O SPEM também suporta, como um recurso opcional, um conjunto de ícones para representação de elementos

definidos em seu metamodelo. Esses ícones são utilizados para representar seus estereótipos. A Tabela 1 apresenta alguns desses ícones, seus respectivos estereótipos e uma breve descrição.

A Figura 1 apresenta o *Workflow* de Análise do Processo Unificado modelado com SPEM. Nesse modelo existem quatro tarefas identificadas: *Architectural Analysis*, *Analyze a Use Case*, *Analyze a Class* e *Analyze a Package*. Cada uma dessas tarefas é representada por um ícone do tipo Uso da Tarefa. Associado às tarefas existem elementos do tipo Uso do Papel, com o elemento *Architect*, responsável pela condução da tarefa *Architectural Analysis*. Por fim, elementos do tipo Uso do Produto de Trabalho definem os artefatos gerados ou consumidos nas tarefas associadas.

## Linha de processo de software

Linha de processo de software (LPrS) refere-se à aplicação de técnicas e princípios de conceitos provenientes de linhas de produto, no contexto de processos de software. Seu objetivo é fornecer técnicas e mecanismos para:

(i) a modelagem de similaridades e variabilidades existentes em uma família de processos de software, e

(ii) a derivação automática de processos de software customizados que atendam às necessidades específicas de um determinado projeto de desenvolvimento de software.

O termo variabilidade em linha de produto se refere à forma como os membros de uma família de produtos podem se diferenciar entre si. Em uma LPrS, esses

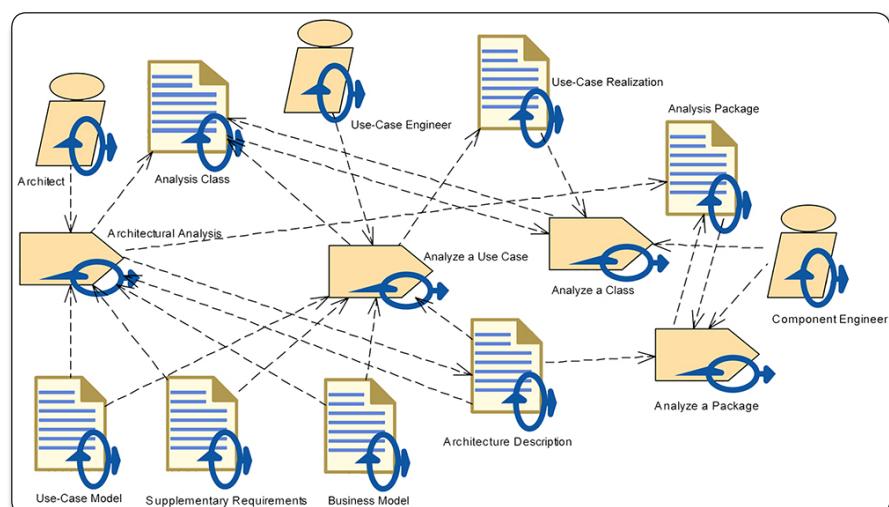


Figura 1. Workflow de Análise do Processo Unificado modelado com SPEM

Ícone	Estereótipo	Descrição
	Activity	Elemento Atividade, que representa um agrupamento de elementos, tais como, outras instâncias de Atividades (Activity), Uso de Tarefas (Task Uses), Uso de Papéis (Role Uses) e Uso de Produtos de Trabalho (Work Product Uses).
	TaskUse	Elemento Uso da Tarefa (Task Uses), que representa uma Tarefa sendo realizada por um Papel no contexto de uma Atividade.
	Step	Elemento Passo (Step), que representa um dos passos necessários para realizar a Tarefa.
	WorkProductUse	Elemento Uso do Produto de Trabalho (Work Product Uses), que representa um Artefato consumido ou produzido no contexto de uma Atividade específica.
	RoleUse	Elemento Uso do Papel (Role Uses), que representa um Papel responsável por uma ou mais Tarefas específicas.

Tabela 1. Principais ícones do SPEM 2.0

membros podem ser identificados pelos elementos de processos que se diferenciam em uma família de processos.

Assim como em linhas de produto, LPrS podem conter pontos de variação, os quais são elementos de processo que podem ser instanciados de diferentes formas, dependendo de uma situação específica. Para cada ponto de variação, existem elementos variantes, os quais são elementos do processo que podem ser selecionados para resolver um ponto de variação.

Uma LPrS possui similaridades e variabilidades associadas aos elementos do processo, como atividades, artefatos, papéis e ações. Exemplos de variabilidades que podem ser modeladas em uma LPrS são: (i) para representar um nível específico de modelo de maturidade de processo de software; (ii) para representar alternativas para o uso de uma determinada técnica de especificação de requisitos, modelagem do projeto de software ou atividades de execução de testes; (iii) para representar a escolha de um estilo de documentação da arquitetura do software; ou (iv) para representar a escolha de uma específica linguagem de programação. Essas variabilidades representam pontos onde elementos de processos podem ser estendidos ou refinados.

Nos últimos anos, vários trabalhos têm sido publicados explorando a adaptação de conceitos de linhas de produto voltadas à gerência de variabilidades em linhas de processo. Esses trabalhos propõem técnicas de modelagem que são utilizadas para promover uma gestão adequada dessas variabilidades e derivação de processos customizados. Entretanto, existem desafios a serem perseguidos já que esses trabalhos evidenciam que ainda existe a necessidade de estudos mais empíricos e avaliações de suas propostas.

## A abordagem SMarty para gerenciamento de variabilidade

A UML Stereotype-based Management of Variability (*SMarty*) é uma abordagem para gerenciamento de variabilidades em linhas de produto. Essa abordagem é composta por um perfil UML, o *SMartyProfile*, e um processo sistemático para o gerenciamento de variabilidades, o *SMartyProcess*.

O *SMartyProfile* contém um conjunto de estereótipos para representar variabilidade em modelos de linhas de produto. Basicamente, o *SMartyProfile* utiliza uma notação padrão orientada a objetos junto ao seu perfil UML para promover uma extensão da UML, permitindo assim, a representação gráfica dos conceitos de variabilidade. O *SMartyProcess* é um processo sistemático que guia um utilizador na identificação, delimitação, representação e rastreabilidade de variabilidades em modelos de linhas de produto.

Os principais conceitos envolvidos no gerenciamento de variabilidades em linhas de produto são adotados pelo *SMarty*. São identificados quatro principais conceitos: variabilidade, ponto de variação, variante e restrição entre variantes.

Com base nesses conceitos, o *SMartyProfile* é composto pelos seguintes estereótipos:

- <<variationPoint>> – que representa o conceito de ponto de variação em linhas de produto;

• <<variant>> – que representa o conceito de variante. Este estereótipo pode ser ainda especializado por quatro outros estereótipos para representar o comportamento das variantes: <<mandatory>>, <<optional>>, <<alternative\_OR>> e <<alternative\_XOR>>;

- <<mandatory>> – que representa variantes mandatórias ou obrigatórias que devem ser parte de todos os produtos derivados de uma linha de produto;

- <<optional>> – que representa variantes que podem ser selecionadas para resolver um ponto de variação ou uma variabilidade;

- <<alternative\_OR>> – que representa variantes em que diferentes combinações podem resolver pontos de variações ou variabilidades;

- <<alternative\_XOR>> – que representa variantes em que apenas uma dessas pode ser selecionada para resolver um ponto de variação ou uma variabilidade;

- <<mutex>> – que representa o conceito de restrição de variantes em linhas de produto. É uma relação mutuamente exclusiva entre duas variantes. Isso significa que quando um variante é selecionada, outra variante associada não pode ser selecionada para um mesmo produto;

- <<requires>> – que representa o conceito em que a seleção de uma variante requer a seleção de outra para um mesmo produto;

- <<variable>> – que indica que um determinado elemento é composto por outros elementos que possuem algum tipo de variabilidade como, por exemplo, um componente que é formado por classes que possuem variabilidade.

A **Figura 2** ilustra a modelagem UML de casos de uso do *Arcade Game Marker* (AGM) que utiliza a abordagem *SMarty* para representar as variabilidades do modelo. Nessa modelagem é possível identificar dois atores, *GamePlayer* e *GameInstaller*, os quais são associados a diferentes casos de uso como o *Save Game*, *Exit Game* e *Play Selected Game*. Note como os estereótipos definidos pelo *SMarty* são introduzidos no modelo.

## SMartySPEM para gerenciamento de variabilidades LPrS

A abordagem *Stereotype-based Variability Management for the SPEM metamodel* (*SMartySPEM*) tem como objetivo apoiar a identificação e representação de variabilidades em elementos de processos modelados com SPEM. Adicionalmente, o *SMartySPEM* também busca auxiliar na derivação de processos customizados a partir de linhas de processo. Entretanto, o processo de derivação ainda não ocorre de maneira automatizada.

O *SMartySPEM* é composto por um perfil UML, o *SMartySPEMProfile*, e um conjunto de diretrizes que direcionam um usuário na identificação e representação de variabilidades em modelos baseados no SPEM. A Figura 3 descreve as atividades necessárias para a definição de uma LPrS e derivação de processos customizados baseado na abordagem *SMartySPEM*. Observe que para a realização da primeira

atividade nomeada *Apply SMartySPEM Guidelines* é necessário ter em mãos artefatos do tipo *SPEM-based Process Models*. Esses artefatos são modelados utilizando o meta-modelo SPEM. O artefato resultante da execução da primeira atividade é uma LPrS baseada no SMartySPEM. Nesse momento, a LPrS já possui todas as variabilidades identificadas e representadas, o que habilita a possibilidade de executar a próxima atividade denominada *Resolve Variabilities/Derive Specific Processes*. Nessa atividade, diferentes processos customizados podem ser derivados a partir da LPrS previamente modelada. Essa derivação ocorre resolvendo os pontos de variação encontrados na LPrS, com base no projeto ou domínio no qual o processo derivado será utilizado.

As diretrizes fornecidas pelo SMartySPEM basicamente direcionam o processo de identificação de variabilidades e a modelagem de elementos de processos baseado no SPEM, aplicando estereótipos UML definidos pelo SMartySPEMProfile. Na Figura 4 é possível visualizar como esses estereótipos do perfil UML SMartySPEMProfile realizam a extensão de outros estereótipos do perfil UML do SPEM 2.0. O mesmo ocorre para as metaclasses da UML. Dessa forma, é possível aplicar os estereótipos definidos pelo SMartySPEM diretamente em elementos UML e elementos do SPEM.

Os estereótipos definidos pelo SMartySPEMProfile estendem aos seguintes estereótipos do perfil UML do SPEM: *<<Step>>*, *<<Activity>>*, *<<RoleUse>>*, *<<TaskUse>>* e *<<WorkProductUse>>*. Tais estereótipos representam os elementos Passo, Atividade, Uso do Papel, Uso da Tarefa e Uso do Produto de Trabalho respectivamente, os quais são utilizados para representar elementos de processos em modelos baseados no SPEM. Nesse contexto, apenas um conjunto específico de metaclasses UML são estendidas pelos perfis UML do SPEM e SMartySPEM.

O SMartySPEM é formado pelos seguintes estereótipos:

- *<<variability>>* – que representa o conceito de variabilidade em LPrS;

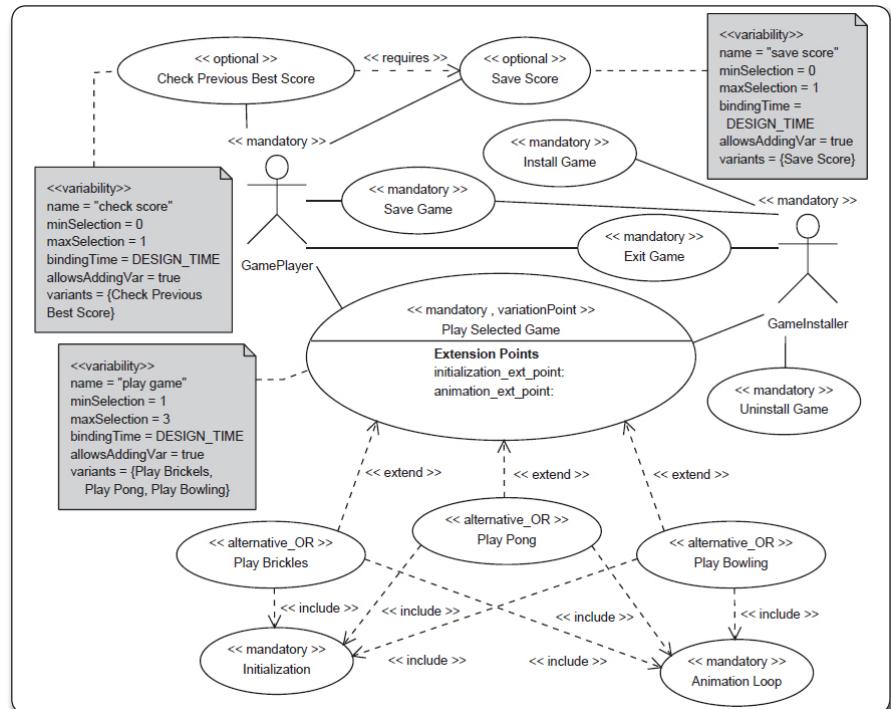


Figura 2. Arcade Game Marker modelado utilizando o SMartySPEM

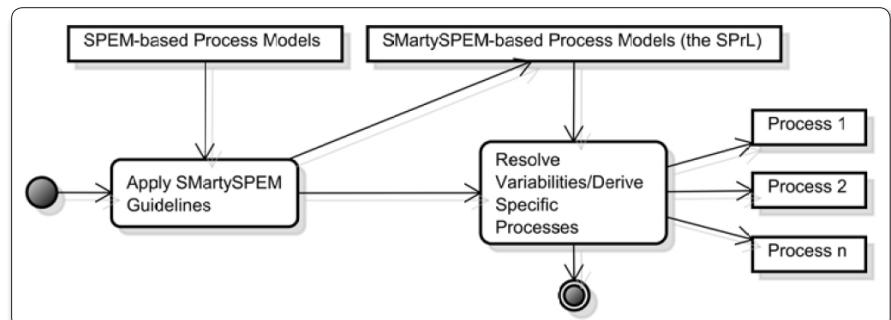


Figura 3. Modelando LPrS e derivando processos customizados baseados na abordagem SMartySPEM

- *<<variationPoint>>* – que representa específicos elementos de processo que variam de acordo com o ambiente;
- *<<variant>>* – que representa o conceito de elementos de processos variantes em LPrS;
- *<<mandatory>>* – que representa elementos de processos mandatórios que devem estar presentes em qualquer derivação de processo a partir de uma LPrS;
- *<<optional>>* – que representa elementos de processos opcionais que podem ser parte de um processo derivado de uma LPrS;
- *<<alternative\_OR>>* – que representa diferentes combinações de elementos de processos do tipo inclusivo, para resolver um ponto de variação ou variabilidade;
- *<<mutex>>* – que representa o conceito de restrição de variantes com relacionamento mutuamente exclusivo entre dois elementos de processo;
- *<<requires>>* – que representa o conceito de variante que ao ser selecionada implica na seleção de outro elemento para um específico processo;
- *<<variable>>* – que indica que uma partição em um diagrama de classes contém um conjunto de elementos de processos com variabilidades explícitas.

O SMartySPEM define o seguinte conjunto de diretrizes que auxiliam na identificação e representação de variabilidades em elementos de processos modelados, levando em consideração os estereótipos do SMartySPEMProfile:

D1. elementos DecisionNode em diagramas de atividade sugerem pontos de variação marcados com <<variationPoint>>; já que representam explicitamente múltiplos caminhos para diferentes atividades de processos de software;

D2. elementos Atividade do SPEM 2.0 em diagramas de atividade podem ser definidos como variantes obrigatórias ou opcionais marcadas, respectivamente, com <<mandatory>> e <<optional>>;

D3. elementos Atividade do SPEM 2.0 em diagramas de atividade, os quais representam fluxos alternativos de saída a partir de um DecisionNode sugerem Atividades variantes inclusivas (<<alternative\_OR>>) ou exclusivas (<<alternative\_XOR>>);

D4. elementos ActivityPartition em diagramas de atividades os quais contêm elementos com variabilidade associada, DecisionNode como um

ponto de variação ou Atividade como uma variante, podem ser marcados com <<variable>>;

D5. elementos Uso do Papel, Uso da Tarefa e Uso do Produto de Trabalho definidos no SPEM 2.0 podem sugerir pontos de variação marcados com <<variationPoint>> já que podem representar elementos passíveis de seleção em diferentes processos de software;

D6. elementos Uso do Papel, Uso da Tarefa, Passo e Uso do Produto de Trabalho definidos no SPEM 2.0 podem ser marcados como variantes obrigatórias ou opcionais, respectivamente, com <<mandatory>> e <<optional>>. Isso ocorre quando esses elementos especializam ou são composições/agregações de outros elementos do mesmo tipo;

D7. elementos Uso do Papel, Uso da Tarefa, Passo e Uso do Produto de Trabalho definidos no SPEM 2.0 que especializam ou são composições/agregações de elementos do mesmo tipo marcados com <<variationPoint>>, sugerem elementos variantes inclusivos (<<alternative\_OR>>) ou exclusivos (<<alternative\_XOR>>);

D8. elementos variantes que, ao serem selecionados para fazer parte de um processo, exigem a presença de outro(s) determinado(s) elemento(s) variante(s) do mesmo tipo, devem ter seus relacionamentos de dependência marcados com o estereótipo <<requires>>;

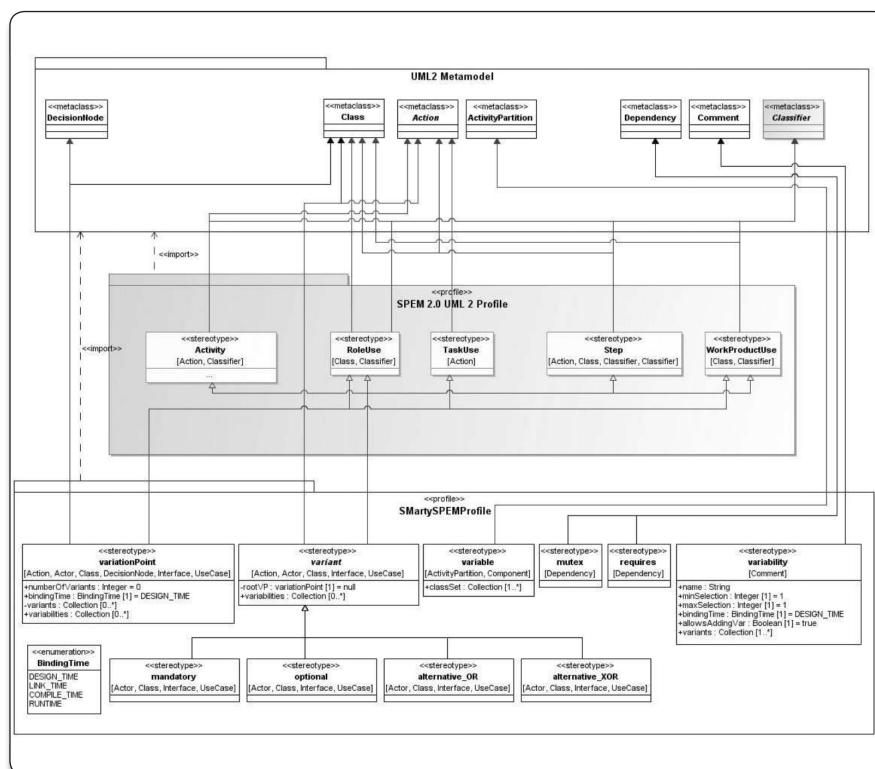
D9. elementos variantes mutuamente exclusivos do mesmo tipo para um determinado processo, devem ter seus relacionamentos de dependência marcados com o estereótipo <<mutex>>.

A abordagem SMartySPEM propõe o mesmo conjunto de ícones do meta-modelo SPEM conforme apresentado na **Tabela 1**. Entretanto, para cada ícone do SPEM, o SMartySPEM define diversos ícones que recebem marcações para identificação de variabilidades. Tais marcações são: MDT para elementos mandatórios, OPT para elementos opcionais, OR para elementos variantes inclusivos, XOR para elementos variantes exclusivos e VP para representar pontos de variação. Essas marcações são necessárias para melhorar a legibilidade de LPrS baseadas no SMartySPEM.

## Exemplo de aplicação do SMartySPEM

Para ilustrar a aplicação da abordagem SMartySPEM, foi projetada uma LPrS hipotética baseada no *Workflow* de Análise do Processo Unificado, o qual passou ainda por uma adaptação para representar a realização de atividades relacionadas ao desenvolvimento baseado em componentes.

A **Figura 5** apresenta o diagrama de atividades de uma LPrS baseada no SMartySPEM. As atividades desse modelo são representadas pelo elemento Atividade proveniente do SPEM e modificado pela abordagem SMartySPEM. Dessa forma, cada elemento do modelo recebe uma marcação baseada nos estereótipos do SMartySPEMProfile. Note que a primeira atividade nomeada *Architectural Analysis* está marcada como mandatória, de acordo com a diretriz D2. O primeiro *decision node* é marcado como ponto de variação, conforme diretriz D1. Associado ao *decision node* existe o elemento *Analyse Type Select* o qual informa detalhes como



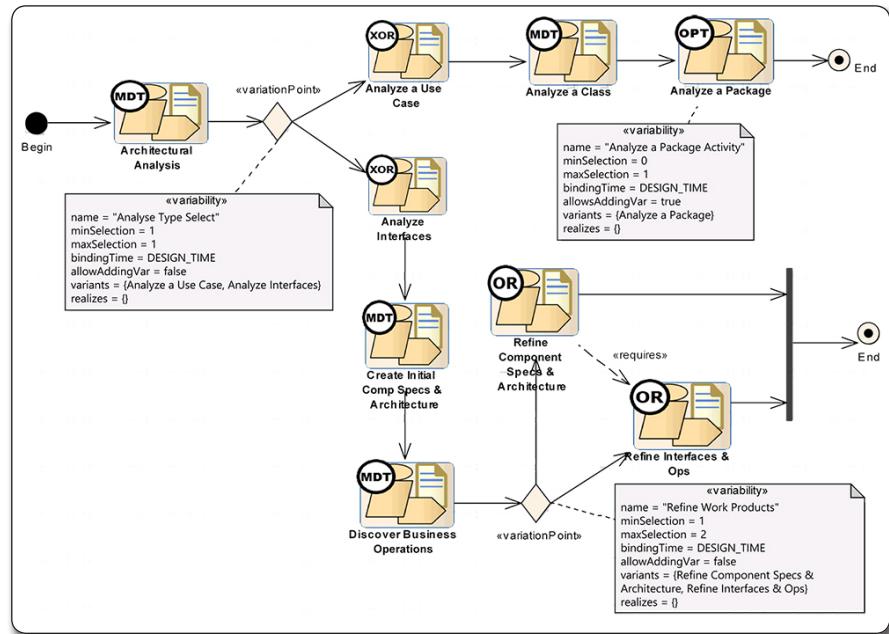
**Figura 4.** O SMartySPEMProfile e seus estereótipos

*name, min e max selection e binding time.* Os elementos *Analyze a Use Case* e *Analyze Interfaces* são as variantes associadas ao ponto de variação *Analyse Type Select*. A marcação desses elementos indicam que são variantes do tipo exclusivas, ou seja, apenas uma das opções pode ser selecionada para resolver o ponto de variação. Note que a atividade *Analyze a Package* é opcional, conforme diretriz D2 do SMartySPM. Sendo assim, essa atividade poderá ser executada opcionalmente caso a variante *Analyze a Use Case* seja selecionada para resolver o ponto de variação *Analyse type Select*.

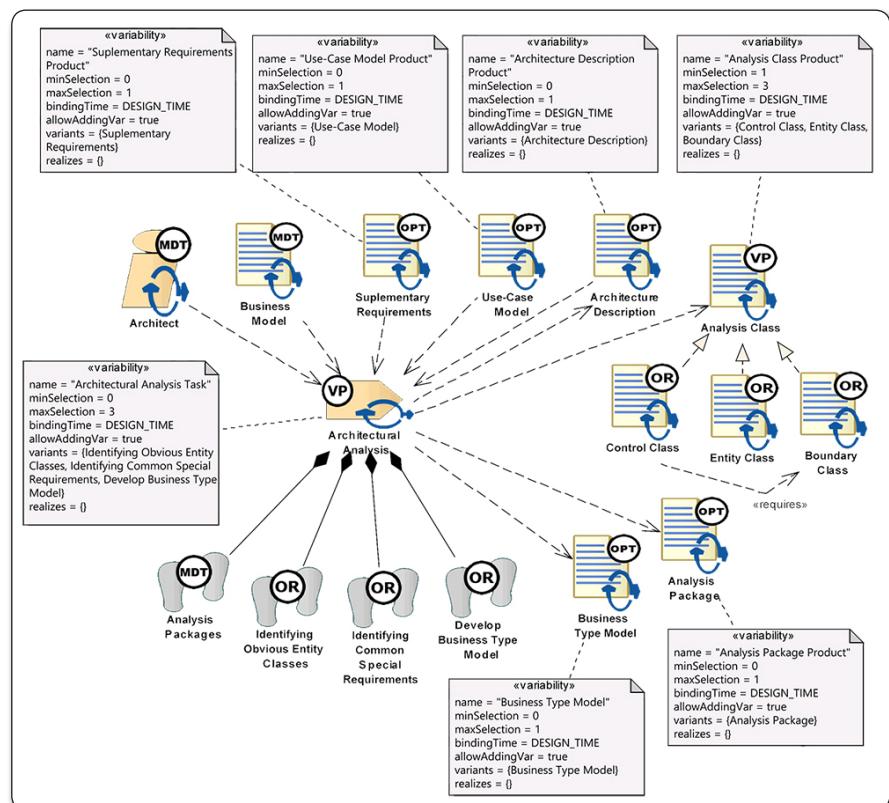
A Figura 6 apresenta a modelagem da atividade *Architectural Analysis*, assim como os elementos de processos relacionados. Neste modelo, o elemento *Architectural Analysis* representa um ponto de variação com três variantes inclusivas relacionadas. Essas variantes são: *Identifying Obvious Entity Classes*, *Identifying Common Special Requirements* e *Develop Business Type Model*. O elemento *Architect* associado é uma variante mandatória do tipo Uso do Papel que identifica o responsável por realizar a atividade *Architectural Analysis*. Outro ponto de variação encontrado no modelo é o elemento nomeado *Analysis Class*, o qual representa um dos artefatos gerados pela atividade e possui três variantes inclusivas associadas: *Control Class*, *Entity Class* e *Boundary Class*.

A partir da LPrS e da atividade *Architectural Analysis* apresentadas respectivamente nas **Figuras 5 e 6**, é possível derivar diferentes modelos customizados, de acordo com os critérios de cada projeto onde esses modelos podem ser aplicados. Para isso, é necessário resolver os pontos de variação encontrados e selecionar quais elementos serão parte do modelo final.

Na **Figura 7** é apresentado um exemplo da atividade *Architectural Analysis* que foi derivada a partir da LPrS hipotética utilizada para exemplificar a utilização da abordagem SMartySPEM neste artigo. Note que os elementos mandatórios encontrados na **Figura 6** também estão presentes na atividade derivada na **Figura 7**. Por outro lado, elementos opcionais como *Supplementary Requirements*



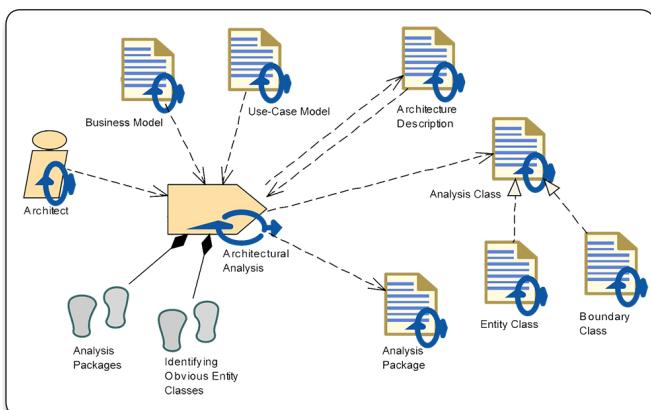
**Figura 5.** LPrS hipotética modelada com SMartySPEM



**Figura 6.** Atividade Architectural Analysis modelada com SMartySPERM

e *Business Type Model* não foram selecionados durante a derivação da atividade. Para resolver o ponto de variação *Architectural Analysis*, os elementos *Analysis Packages* e *Identifying Obvious*

*Entity Classes* foram selecionados. Da mesma forma, os seguintes elementos foram selecionados para resolver o ponto de variação *Analysis Class*: *Entity Class* e *Boundary Class*.



**Figura 7.** Modelo da atividade Architectural Analysis derivado da LPrS

## Conclusões

A abordagem SMartySPEM apresenta um mecanismo para gerenciamento de variabilidades em LPrS baseadas no SPEM. É composta pelo *SMartyProfile* e um conjunto de guias que auxiliam na identificação e representação de variabilidades em elementos de processos de software provenientes do SPEM.

O foco principal da abordagem é introduzir um mecanismo para representar precisamente elementos variantes opcionais, inclusivos e exclusivos, além de outras categorias de elementos que tornam a modelagem de LPrS fácil e efetiva. Dessa forma, novos mecanismos podem se beneficiar da abordagem para, por exemplo, introduzir uma maneira automática de derivação de processos customizados a partir de uma LPrS baseada no SMartySPEM. Como exemplo de aplicação da abordagem, foi apresentada uma LPrS hipotética baseada no *Workflow* de Análise do Processo Unificado, modificada para suportar atividades relacionadas ao desenvolvimento de produtos de software baseados em componentes.

### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



### Links e Referências:

1. Alegria, J.A.H., Bastarrica, M.C.: Building Software Process Lines with CASPER. In: Proceedings of the International Conference on Software and System Process. pp. 170-179. IEEE Computer Society, Zurich, Switzerland (2012).
2. Aleixo, F.A., Freire, M.A., Santos, W.C., Kulesza, U.: Automating the Variability Management, Customization and Deployment of Software Processes: A Model-Driven Approach. Lecture Notes in Business Information Processing 73(1), 372-387 (2011).
3. Armbrust, O., Katahira, M., Miyamoto, Y., Münch, J., Nakao, H., Ocampo, A.: Scoping Software Process Lines. Software Process: Improvement and Practice. Software Process: Improvement and Practice - Examining Process Design and Change 14(3), 181-197 (2009).
4. Barreto, A., Duarte, E., Rocha, A.R., Murta, L.: Supporting the Denition of Software Processes at Consulting Organizations via Software Process Lines. In: Proceedings of the International Conference on the Quality of Information and Communications Technology. pp. 15-24. IEEE Computer Society, Porto, Portugal (2010).
5. Bosch, J.: Preface. In: Proceedings of the 2nd Groningen Workshop on Software Variability Management: Software Product Families and Populations. pp. 1-2. University of Groningen, Groningen, The Netherlands (2004).
6. Fiori, D.R., Gimenes, I.M.S., Maldonado, J.C., Oliveira Junior, E.A.: Variability Management in Software Product Line Activity Diagrams. In: International Conference on Distributed Multimedia Systems. pp. 89-94 (2012).
7. Jacobson, I., Griss, M.L., Jonsson, P.: Software Reuse: Architecture, Process, and Organization for Business Success. Addison-Wesley Professional, Boston, MA, USA (1997).
8. Linden, F.J., Schmid, K.v.d., Rommes, E.: Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer-Verlag, New York (2007).
9. Oliveira Junior, E.A., Gimenes, I.M.S., Maldonado, J.C.: Systematic Management of Variability in UML-based Software Product Lines. Journal of Universal Computer Science 16(17), 2374-2393 (2010).
10. OMG: Software & Systems Process Engineering Metamodel (SPEM) <http://www.omg.org/spec/SPEM>

Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros

## Introdução ao guia de conhecimento de análise de negócio



### Porque esse artigo é útil:

O mercado de software vem expandindo para diferentes setores, buscando melhorar e agilizar rotinas de trabalho, automatizando o que antes era manual. Os sistemas comumente são carentes de rotinas que possibilitem a gestão específica do negócio correspondente devido à complexidade e a carência de analistas capacitados, o que resulta em processos de negócios parcialmente automatizados. A busca por alternativas ágeis e produtivas para descobrir o que seu usuário necessita e o que o satisfaz é a realidade atual das softhouses, porém a aplicação de ações básicas como o mapeamento do processo de negócio, entrevista a envolvidos, identificação das necessidades e possíveis soluções, bus-

cando transformar as necessidades do negócio em requisitos estruturados através da engenharia de software é importante. Em particular, esta preocupação com a análise do negócio é encontrada em diferentes frameworks, como o BABOK, que é um guia que norteia a análise de negócio sugerindo alternativas no processo de análise, auxiliando analistas na coleta, modelagem e validação de requisitos, redução tempo e custo na execução de atividades.

A discussão deste tema é útil para organizações de pequeno e médio porte, ou equipes que buscam modelos que lhe auxiliem a estruturar um processo de análise. O artigo se aplica também a profissionais que gostariam de conhecer o modelo para se especializarem na área.



**Ivânia Ramos dos Santos**

[ivania.ramos.santos@gmail.com](mailto:ivania.ramos.santos@gmail.com)

Bacharel em Sistemas de Informação pela Faculdade Mater Dei (2008), especialista em Engenharia de Software pela Faculdade Mater Dei (2011). Atualmente é Gerente de Serviços de TI da empresa Alnova., professora na Faculdade Mater Dei, FESC e Sesni. Experiência gestão de processos de qualidade utilizando metodologia MPS.BR.

**A** engenharia de software trata de aspectos relacionados ao estabelecimento de processos, métodos, técnicas, ferramentas e ambientes de suporte ao desenvolvimento de software. Assim como em outras áreas, em uma abordagem de engenharia de software, inicialmente o problema a ser tratado deve ser analisado e decomposto em partes menores, onde para cada

uma dessas partes, uma solução deve ser elaborada. Ao combinarmos métodos abrangentes para todas as fases de desenvolvimento do software, melhores ferramentas para automatizar esses métodos, blocos de construção mais poderosos para a implementação do software, melhores técnicas para garantia de qualidade do software e uma filosofia de coordenação predominante, controle



**Fabio Pereira dos Santos**

[fabiosantos@viisoft.com.br](mailto:fabiosantos@viisoft.com.br)

Tecnólogo em Análise e desenvolvimento de Sistemas pela UTFPR Pato Branco(2013). Atualmente é Coordenador de Atendimento no ramo de Agronegócios na empresa VIASOFT

e administração, podemos conseguir uma disciplina para o desenvolvimento do software – disciplina esta chamada engenharia de software.

Embora o software seja algo essencial, ainda traz consigo muitos problemas que afligem seu desempenho como o custo, muitas vezes alto, a falta de produtividade da equipe desenvolvedora, demonstrada pelo número de erros e principalmente o não atendimento das necessidades do usuário. Esses pontos negativos vêm sendo subterfugidos desde o início da história do software.

A engenharia de software pode ser descrita como métodos e princípios que têm como objetivo chegar a um software de melhor qualidade, que atenda as necessidades do usuário, com um custo mais baixo, sendo “a criação e a utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais”.

Para obter um software com melhor qualidade, buscando soluções eficientes, é imprescindível a aplicação de conceitos de análise de negócios, utilizando elementos da engenharia de software, em conjunto com uma ferramenta de análise de negócio, conforme sugerido pelo Guia BABOK, apresentado a partir de agora.

## O guia de conhecimento de análise de negócios (BABOK)

O IIBA (Instituto Internacional de Análise de Negócios) foi fundado em

Toronto, Canadá, em outubro de 2003, para apoiar a comunidade de análise de negócios através das seguintes iniciativas:

- Criação de desenvolvimento de consciência e reconhecimento do valor e da contribuição do analista de negócios;
- Definição de um Guia para o Corpo de Conhecimento de Análise de Negócios (BABOK);
- Estabelecimento de um fórum para compartilhamento do conhecimento e contribuição para a profissão de Análise de Negócios;
- Reconhecimento público e certificação dos praticantes de análise de negócios através de um programa de certificação internacionalmente reconhecido.

O guia BABOK contém a descrição de práticas geralmente aceitas no campo da análise de negócio. Os dados levantados deixam claro que as técnicas descritas nessa versão são utilizadas pela maioria dos praticantes. Destacando que o guia deve ser interpretado e adaptado conforme situação de sua aplicação, permitindo a implantação de outras práticas em conjunto.

Outro propósito do guia é definir a profissão do analista de negócio, discutindo tarefas e habilidades do profissional.

O BABOK é considerado um framework, por descrever tarefas de análise de negócios que devem ser executadas no intuito de compreender como uma solução irá gerar valor para a organização

patrocinadora. O modelo é constituído conforme mapa mental apresentado na Figura 1.

A abordagem do mapa mostra o framework composto por técnicas, competências, áreas de conhecimento e tarefas, sendo essas sugeridas pelo modelo à equipe desenvolvedora do trabalho.

As tarefas a serem executadas têm sua ordem sugerida pelo modelo, embora outros fatores permitam uma variação, mas cada uma contribui de alguma forma, direta ou indiretamente, para um objetivo global definido pelo negócio procurando um resultado que direcione a solução do problema da forma mais eficaz.

## Análise de negócio

A análise de negócio é um conjunto de atividades e técnicas utilizadas para servir como elo entre as partes interessadas no produto a ser desenvolvido com objetivo de compreender estrutura, políticas e operações de uma organização, procurando soluções para que a empresa alcance suas metas. Ela pode ser utilizada para compreender o estudo atual de uma organização ou para servir como base na identificação de necessidades de negócio, também na definição e validação de requisitos que atendam as necessidades do negócio, suas metas e objetivos.

Analistas de negócio são responsáveis por analisar informações repassadas por pessoas envolvidas na parte operacional do processo, sejam colaboradores, clientes ou profissionais de TI (Tecnologia de Informação). Sendo de responsabilidade dos analistas descobrirem as reais necessidades de cada usuário e buscar maneiras para facilitar a comunicação entre as partes interessadas.

Qualquer indivíduo pode ser analista de negócio, independente do cargo, desde que seja uma pessoa que desenvolva tarefas de análise, ou seja, analistas de processos, engenheiros, responsáveis pelo produto ou gerentes de produtos.

O analista deve compreender as áreas de conhecimento, destacando que esses tendem a executar tarefas de todas as áreas de conhecimento no processo de forma rápida, iterativa ou simultaneamente.

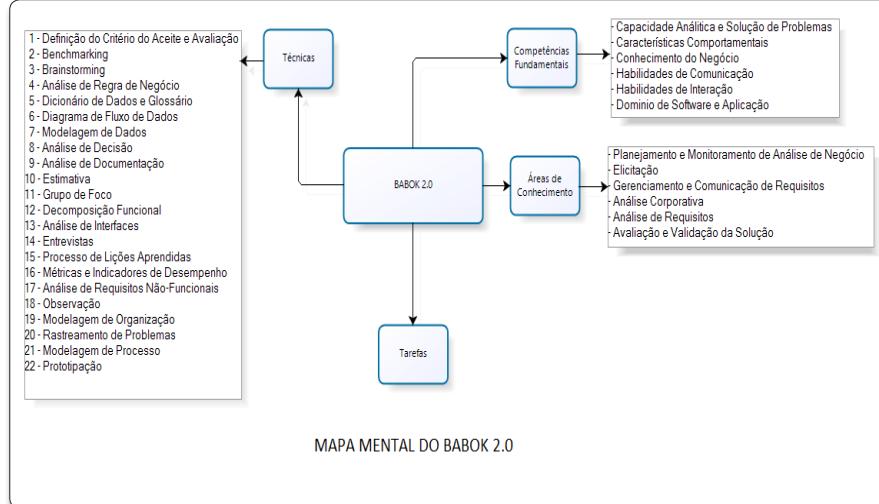


Figura 1. Mapa Mental BABOK

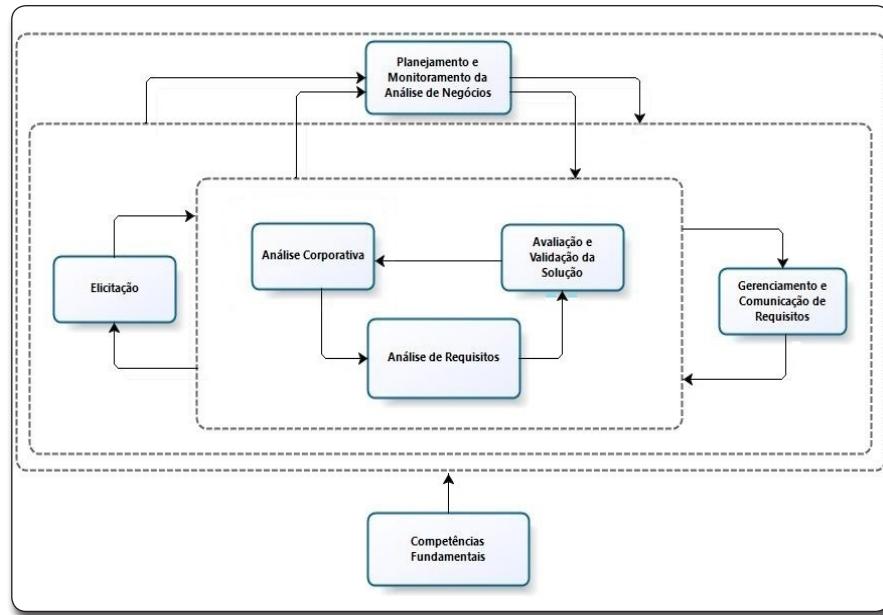
## Áreas de conhecimento

Áreas de conhecimento definem o que um praticante de análise de negócios precisa compreender e as tarefas que se deve executar em uma sucessão rápida, iterativa ou simultaneamente.

As áreas de conhecimento não representam fases em um projeto, mesmo sendo possível e permissível partir das atividades de análise corporativa para as atividades de análise de requisito e, então, para a avaliação e validação da Solução, e tratar cada uma como uma fase distinta em um projeto.

As áreas de conhecimento que compõe o BABOK são:

- Análise Corporativa: descreve como os analistas identificam uma necessidade de negócio, refinam e esclarecem a necessidade, também definem um escopo de solução, que pode ser implementado pelo negócio de forma viável descrevendo a definição e análise do problema, estudos de viabilidade e escopo da solução;
- Análise de Requisitos: define como os analistas priorizam e elaboram os requisitos das partes interessadas e da solução, com objetivo de que a equipe de projeto implemente a solução que irá atender as necessidades do solicitante. Envolve análise da necessidade, para definir soluções que atendam o solicitado, avaliando o estado atual do negócio para identificar e recomendar melhorias, e a verificação e validação dos requisitos resultantes;
- Avaliação e Validação da Solução: detalha como analistas avaliam as soluções propostas para determinar qual se encaixa melhor nas necessidades do negócio identificando lacunas e falhas de soluções. Também descreve como os analistas avaliam soluções entregues para ver quanto bem atendem à necessidade original para que a solicitante possa julgar o desempenho e eficácia da solução;
- Elicitação: registra como os analistas trabalham junto com as partes interessadas para identificar e compreender suas necessidades e preocupações. A elicitão garante que as reais necessidades das partes interessadas sejam compreendidas e não somente seus desejos explícitos e superficiais;



**Figura 2.** Relacionamento entre as Áreas de Conhecimento

- Gerenciamento e comunicação dos requisitos: representa como analistas gerenciam conflitos, questões e mudanças, no intuito de garantir que as partes interessadas e o time do projeto permaneçam em acordo com o escopo da solução. Também descreve como os requisitos são comunicados às partes interessadas e como o conhecimento obtido pelo analista de negócios é mantido para o futuro;
- Planejamento e monitoramento da análise de negócios: está relacionada em como os analistas determinam as atividades necessárias para que seja executada uma iniciativa de análise de negócio. Descreve e identifica as partes interessadas, seleciona as técnicas e o processo que será utilizado para gerenciar os requisitos e avaliar o progresso do trabalho;
- Competências fundamentais: descrevem comportamentos, conhecimentos e outras características que apoiam o desempenho efetivo da análise de negócio.

O guia BABOK não exige que você proceda desta forma e ele não deve ser imposto como uma metodologia para a execução da análise de negócios. As tarefas podem ser executadas em qualquer ordem, contanto que as entradas necessárias estejam disponíveis.

A ordem de execução das áreas de conhecimento, sugerida pelo BABOK, é apresentada pela **Figura 2**.

O processo apresentado na **Figura 2** inicia com a análise corporativa, que identifica uma necessidade de negócio e define um escopo de solução, o qual pode ser implementado pelo negócio de forma viável. Seguindo para a análise de requisitos, que tem como objetivo permitir que a equipe de projeto implemente a solução que irá atender as necessidades do solicitante, sendo posteriormente realizada a avaliação e validação da Solução, que descreve como analistas avaliam as soluções propostas para determinar qual se encaixa melhor nas necessidades do negócio identificando lacunas e falhas de soluções.

As três primeiras áreas a serem executadas são apresentadas pela **Figura 3**.

A próxima área a ser abordada, conforme ilustrado pela **Figura 4** é a de Elicitação, onde se tornam explícitos os requisitos, ou seja, é garantido que as reais necessidades das partes interessadas serão compreendidas. Na sequência temos a área de Gerenciamento e Comunicação de Requisitos.

A área de Gerenciamento e Comunicação de Requisitos aborda como analistas gerenciam conflitos, questões e mudanças, no intuito de garantir que as partes interessadas e o time do projeto permaneçam em acordo com o escopo da solução.

A fase de Planejamento e Monitoramento da Análise de Negócios, também apresentada pela **Figura 3**, tem influência e também é influenciada tanto pela primeira quanto pela segunda etapa, sendo essa área a responsável por determinar as atividades necessárias para que seja executada uma iniciativa de análise de negócio. Nela se identifica as partes interessadas, se seleciona técnicas a serem utilizadas e o processo que será implementado, para então gerenciar os requisitos e avaliar o progresso do trabalho.

As competências fundamentais que abrangem todas as áreas do modelo é a área que descreve comportamentos,

conhecimentos e outras características que apoiam o desempenho efetivo da análise de negócios.

Cabe esclarecer que essas áreas não precisam ser executadas na ordem descrita no guia, pois o modelo não exige um procedimento padrão imposto e sim um coerente com as características da organização, do negócio e da equipe.

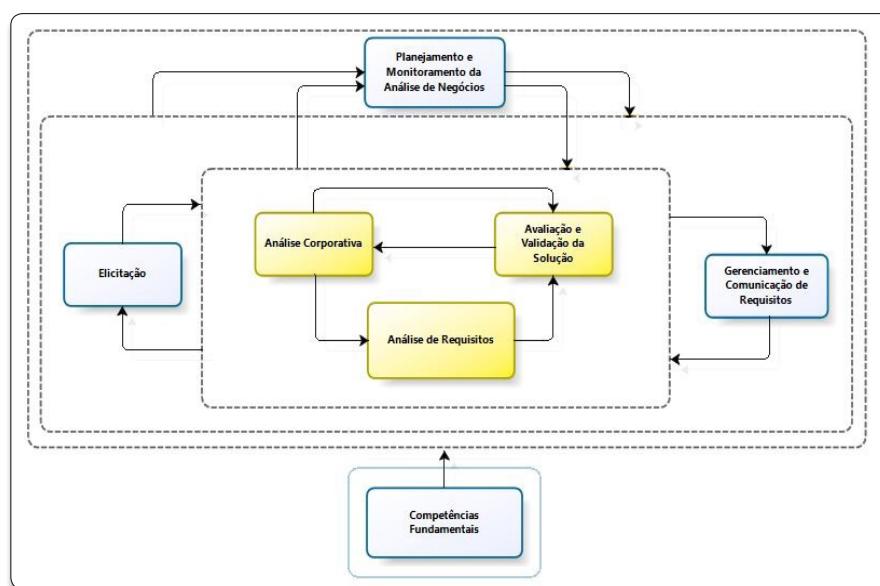
## Tarefas

Cada área de conhecimento descreve as tarefas desempenhadas por analistas de negócio para atingir um propósito da área correspondente, onde cada tarefa é apresentada no seguinte formato:

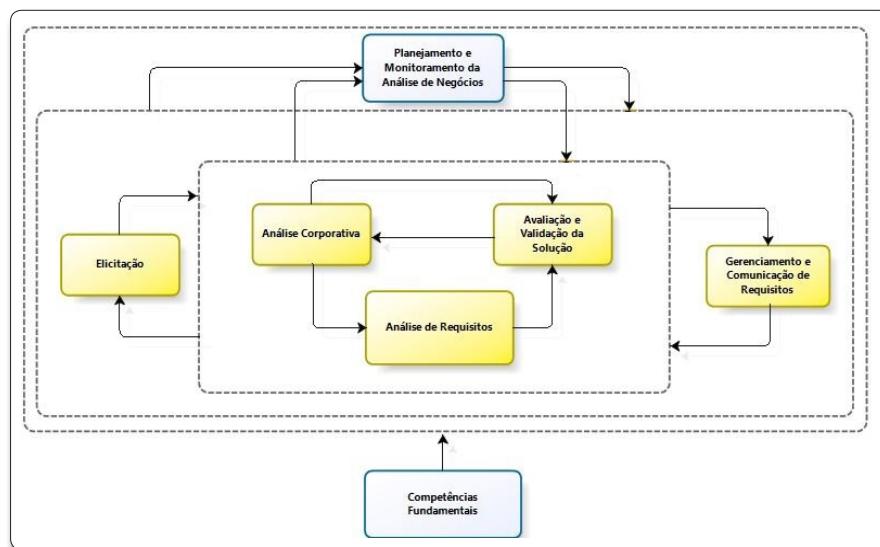
- Propósito: Breve descrição da razão pela qual a tarefa é executada, também o valor criado por sua execução.
  - Descrição: A tarefa é parte fundamental do processo e deve ser desempenhada como parte do negócio. Cada uma delas deve ser executada ao menos uma vez durante o processo, não havendo limite de quantidade. Porém, cada uma tem sua própria característica a qual deve estar detalhada na sua descrição possuindo características claras e permitindo o alcance de resultados na sua saída que gerem valor ao solicitante. Também é importante que seja completa, onde tarefas sucessoras que fazem uso de suas saídas devem poder ser executadas por outra pessoa ou grupo.
  - Entrada: Representa informações ou pré-condições necessárias para que uma tarefa seja iniciada. Pode ser gerada fora do escopo da análise de negócios, por exemplo, no caso da construção de um software, ou por uma tarefa de análise de negócio.
  - Elementos: O formato e a estrutura dessa seção são únicos para cada tarefa. Tal seção descreve os principais conceitos que são necessários para compreender como executar uma tarefa.
  - Técnicas: Proveem informações adicionais sobre as diferentes maneiras que uma tarefa pode ser executada, ou ainda, diferentes formas que uma saída pode assumir devendo essa estar relacionada a pelo menos uma técnica, como:

- Brainstorming: técnica dedicada a produzir um conjunto amplo ou diverso de opções auxiliando na resposta a questões específicas como: “Quais opções estão disponíveis para atuar sobre a questão em mãos?”, “Quais fatores estão impedindo o grupo de avançar com uma abordagem ou opção?” ou “O que poderia estar causando atraso nas atividades?”. Essa técnica funciona através do foco de um tópico ou problema, e então, levanta-se várias soluções possíveis para ele. Esta técnica é melhor aplicada em grupo, por se alimentar da experiência e criatividade de todos

– Análise de regras de negócio: técnica que permite a análise de políticas e



**Figura 3.** Primeira etapa das áreas de conhecimento



**Figura 4.** Primeira e segunda etapa

regras que direcionam e restringem a organização e a sua operação. A política de negócio trata-se de uma diretiva não acionável que apoia um objetivo do negócio. Já a regra de negócio é uma diretiva específica, acionável e testável que está sob o controle de uma organização e que apoia uma política do negócio. Um conjunto de princípios básicos guia o analista de negócios quando são declaradas ou gerenciadas as regras de negócio, as quais devem ser:

- Declaradas em termos apropriados para permitir que especialistas no assunto validem as regras;
- Documentadas independentemente de como elas são impostas;
- Declaradas em nível atômico e em formato declarativo;
- Separadas dos processos que a regra apoia ou restringe;
- Mantidas de forma que permita que a organização monitore e adapte as regras conforme as políticas do negócio mudam.
- Grupos focais: técnica que permite eliciar ideias e atitudes a respeito de um produto, serviço ou oportunidade em um ambiente de grupo interativo. Os participantes compartilham suas impressões, preferências e necessidades guiadas por um moderador.
- Entrevistas: abordagem sistemática projetada para eliciar informações junto a uma pessoa ou a um grupo de pessoas,

de maneira formal ou informal, através de uma conversa com um entrevistado, na qual são feitas perguntas relevantes e as respostas são documentadas.

- Observação: forma de eliciar requisitos através da condição de uma avaliação de ambiente de trabalho da parte interessada. Esta técnica é apropriada para documentar detalhes sobre processos atuais ou quando o projeto se destina a melhorar ou alterar um processo atual.
- Prototipagem: detalha os requisitos da interface dos usuários e as integra com outros requisitos como caso de uso, cenários, regras de dados e de negócio. Essa técnica pode ser categorizada de duas formas:
  - Escopo funcional: é um protótipo horizontal que modela uma visão superficial e abrangente da funcionalidade do sistema. Normalmente não tem qualquer lógica de negócio rodando por trás da visualização. Um protótipo vertical modela uma fatia profunda e limitada da funcionalidade completa do sistema.
  - Protótipo: visa detectar e esclarecer rapidamente os requisitos de interface, utilizando ferramentas simples, algumas vezes apenas papel e lápis. Como o nome sugere tal protótipo usualmente é descartado quando o sistema final é desenvolvido. O foco está na funcionalidade que não é facilmente elicitada por outras técnicas, que possui

## CURSOS ONLINE

A Revista Java Magazine oferece aos seus assinantes uma série de Cursos Online de alto padrão de qualidade.



### CONHEÇA ALGUNS DOS CURSOS:

- **Curso de noSQL (Redis) com Java**
- **Curso Básico de JDBC**
- **Java Básico: Aplicações Desktop**
- **JSF com Primefaces**
- **Conhecendo o Apache Struts**

Para mais informações :

[www.devmedia.com.br/curso/javamagazine](http://www.devmedia.com.br/curso/javamagazine)  
(21) 3382-5038

pontos de vista conflitantes ou que é difícil de compreender. As partes interessadas frequentemente consideram a prototipagem como um meio concreto de identificar, descrever e validar suas necessidades de interface.

- Análise de riscos: permite identificar e gerenciar áreas de incerteza que podem impactar em uma iniciativa, solução ou organização. Lembrando que um risco descreve uma ocorrência ou um evento incerto que podem ter um efeito na capacidade do analista de negócio, equipe do projeto ou organização de atingir um objetivo. Riscos podem ser positivos ou negativos. A análise de riscos envolve uma compreensão dos níveis de tolerância a risco da organização, avaliando riscos e identificação das respostas.
- Análise de causa-raiz: tem o propósito de determinar a fonte implícita do problema. É um exame estruturado de uma situação para estabelecer as causas-raízes e efeitos resultantes do problema. Um elemento chave da análise de causa-raiz é garantir que o pensamento do negócio e processo sejam desafiados.
- Pesquisa / Questionário: meio de eliciar informações de muitas pessoas, algumas vezes de forma anônima, em um período relativamente curto de tempo. Uma pesquisa pode coletar informações sobre clientes, produtos, práticas de trabalho e atitudes. Uma pesquisa pode também ser chamada de questionário.
- Histórias de usuário: descrições breves de funcionalidades que os usuários precisam para que uma solução atenda a um objetivo do negócio. É uma descrição textual de ações que a solução precisa permitir aos usuários. As histórias são geralmente sentenças que descrevem quem usa a história, a meta que se está tentando alcançar e quaisquer informações adicionais que possam ser críticas para compreender o escopo da história.
- Partes interessadas - Cada tarefa possui uma ou mais partes interessadas que tendem a participar da execução daquela tarefa. Representa uma classe de pessoas que provavelmente o analista de negócios irá interagir de alguma forma. Qualquer parte interessada pode ser uma fonte de requisitos, suposições ou restrições.
- Saída - Resultado necessário do trabalho descrito na tarefa. As saídas são criadas, transformadas ou mudam de estado como resultado bem sucedido de uma tarefa. Uma saída pode ser uma entrega, ou parte de uma entrega maior (uma tarefa pode ter mais de uma saída).

Esse detalhamento das tarefas mostra qual a função de cada uma delas, baseando-se no modelo BABOK.

A sequência prática de aplicação do modelo BABOK deve ser determinada pelas características do negócio e também da softhouse e do cliente.

## Conclusão

Esse artigo apresentou a metodologia BABOK, que possibilita a identificação de dificuldades e necessidades de organizações que pretendem implantar um sistema, bem como apontar uma possível solução através de um sistema automatizado gerencial.

A importância das organizações a se adequarem a um novo modelo de trabalho, mais atualizado e coerente com suas características para posterior automatização também se apresenta fundamental, criando um padrão de como devem ser executados os processos, levando em consideração que os usuários comumente demonstraram dificuldades técnicas na área da informática e na regra do negócio em questão.

A utilização de protótipos é uma das técnicas fundamentais para a compreensão da equipe de desenvolvimento e dos usuários do sistema a ser desenvolvido, permitindo a validação do produto com sucesso, o esclarecimento de dúvidas de usuários e comprovação visual de que o produto a ser desenvolvido facilita as atividades executadas.

### Links e Referências:

**IIBA-International Institute of Business Analysis. Um guia para o Corpo de Conhecimento de Análise de Negócios (Guia Babok) Versão 2.0. Toronto, 2011.**

**A Engenharia de Software e a Otimização dos Resultados organizacionais.**  
[www.webartigos.com](http://www.webartigos.com)

**PFLEGER, Shari Lawrence. Engenharia de Software, Teoria e Prática. 2 ed. Pearson, 2004.**

**PREESMANN, Roger S. Engenharia de Software. São Paulo. Makron Books, 1995.**

**YIN, Roberto K. Estudo de caso: planejamento e métodos. 2ª Ed. Porto Alegre. Editora: Bookmam. 2001.**

### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros



**BUG**

## Automação de testes funcionais para aplicações da plataforma Android

Utilizando a ferramenta Robotium



**Jeanne de Castro Trovão**

[jeanne.trovao@icomp.ufam.edu.br](mailto:jeanne.trovao@icomp.ufam.edu.br)

Mestranda em Informática e Bacharel em Ciência da Computação pela Universidade Federal do Amazonas (UFAM). Faz parte do Grupo de Pesquisa em Experimentação e Teste de Software (ExperTS) da UFAM, onde participa de projetos de P&D em parceria com o Instituto Nokia de Tecnologia (INDT), atuando na área de testes. Possui as certificações internacionais: ISTQB Certified Tester e IREB Certified Professional for Requirements Engineering.



**Arilo Claudio Dias Neto**

[arilo@icomp.ufam.edu.br](mailto:arilo@icomp.ufam.edu.br)

Mestre e Doutor em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (COPPE/UFRJ). Possui 10 anos de experiência na área de Engenharia de Software. Atualmente é professor do Instituto de Computação da Universidade Federal do Amazonas (ICOMP/UFAM). Lidera o Grupo de Pesquisa em Experimentação e Teste de Software (ExperTS) da UFAM. Possui o certificado de Implementador do Modelo MR-MPS. Possui a certificação internacional ISTQB Certified Tester.

**O**s testes de software podem ser classificados por diferentes técnicas. No artigo “Introdução a Teste de Software”, publicado na edição 1 da Engenharia de Software Magazine, os autores comentam os tipos de técnicas de teste que podem ser aplicadas em projetos de software, destacando que os conceitos por trás de tais técnicas permanecem inalterados independente da plataforma do software ou paradigma de desenvolvimento aplicado no projeto. Além disso, o objetivo principal destas técnicas continua a ser o mesmo: revelar falhas durante a execução de um software.

As técnicas de teste são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste. Elas contemplam diferentes perspectivas do software e impõe-se a necessidade de se estabelecer uma estratégia de teste que conte com as vantagens e os aspectos complementares dessas técnicas. As técnicas existentes são: técnica funcional e estrutural.

### **Porque esse artigo é útil:**

Este artigo é útil quando já se tem certa familiaridade ou interesse no desenvolvimento de aplicações para a plataforma Android e encontra-se a necessidade de um apoio ferramental para desenvolver e executar testes funcionais para aplicações desta plataforma através da automatização dos testes. Para isso, o artigo irá abordar alguns conceitos sobre testes funcionais (caixa-preta) e automação de testes. Em seguida, será discutido o tema de teste de software de aplicações para dispositivos móveis e, por fim, será apresentado um framework, Robotium, para automação de testes funcionais de aplicações para a plataforma Android com um exemplo prático usando um ambiente real de desenvolvimento de software

O foco deste artigo é discutir a técnica de Teste Funcional, que se caracteriza como um dos tipos de teste mais importantes para a detecção de defeitos e, consequentemente, a garantia da qualidade em um software, pois ele verifica cada funcionalidade especificada pelos requisitos do software, observando seu comportamento e resultado para avaliar

se ela está executando conforme o esperado. Ao longo do artigo será discutida a aplicação de uma técnica de teste funcional por meio de uma ferramenta de automação em um aplicativo desenvolvido para uma plataforma de dispositivos móveis.

Para aplicar a técnica de teste funcional, devem ser criados vários cenários visando testar as diferentes possibilidades de uso para uma funcionalidade através de possíveis entradas para o sistema com resultados esperados já conhecidos. Para cada entrada e resultado esperado definido, executamos o teste e comparamos o resultado para saber se o sistema atingiu ou não o resultado ou comportamento esperado, em outras palavras, saber se o teste passou ou falhou.

O sucesso do teste funcional é influenciado pela cobertura dos testes, ou seja, quanto maior o número de entradas testadas, maior será a confiabilidade no software. No entanto, é quase impraticável, ou inviável, testar todas as possíveis entradas de um sistema (como por exemplo, quando o número de entradas corresponde ao conjunto de números naturais). Para resolver isso, existem critérios de geração de testes para a técnica de teste funcional que ajudam a encontrar um subconjunto de testes que diminuem o esforço de teste, mas não diminuem a confiabilidade do mesmo. Alguns exemplos de técnicas conhecidas são: particionamento por classes de equivalência e a análise do valor limite. Mais uma vez, o artigo “Introdução a Teste de Software”, publicado na edição 1 da ESM, contém detalhes sobre estes critérios de geração de testes funcionais.

## Automação de testes

Mesmo com a ajuda de técnicas que ajudam a diminuir o esforço de teste, quando se trata de sistemas mais complexos, a execução de testes funcionais de forma manual pode se tornar um gargalo no cronograma de desenvolvimento do sistema. O problema se torna ainda maior ao ter que executar o teste de regressão, onde todos os testes funcionais já executados anteriormente para antigas versões do software devem ser reexecutados para verificar, por exemplo, se uma nova versão do sistema continua estável ou se algum efeito colateral foi observado após a correção de falhas no software.

Por este motivo, cada vez mais profissionais da indústria procuram ferramentas que apoiam a automatização dos testes funcionais, na tentativa de diminuir o tempo de criação e execução dos testes.

Automação dos testes consiste no uso de algum apoio computacional, ferramentas, para controlar a execução dos testes, a comparação dos resultados e comportamentos obtidos com a execução dos testes em relação aos resultados e comportamentos esperados, a configuração das pré-condições dos testes e outras atividades do controle dos testes e relato de seus resultados. Comumente, a automação dos testes envolve automatizar um processo manual já estabelecido em uma organização que utiliza um processo de testes formalizado.

Conforme o artigo “Automatizando Testes Funcionais em Aplicações Web”, publicado na edição 24 da ESM, a automação dos testes é um processo de escrever um programa computacional para realizar testes que, caso contrário, seriam feitos

manualmente. Uma vez que os testes foram automatizados, eles podem ser executados rapidamente, logicamente que proporcionalmente à quantidade de casos de teste a serem executados. Como sua principal vantagem, este representa, na maioria dos casos, o método de teste com custo mais efetivo para produtos que possuem uma longa vida, com muitas manutenções, pois pequenas modificações em certas partes do software ao longo de sua vida podem fazer com que outras partes do software que estariam funcionando anteriormente deixem de funcionar. Neste cenário, existem duas abordagens gerais para automação dos testes: os testes dirigidos a código e os testes de interface. Este artigo possui como foco a aplicação de automação de testes de interface.

Os testes de interface consistem na avaliação de um software a partir da interação com sua interface gráfica, por meio de eventos como digitação de teclas ou cliques do mouse, onde são observadas as mudanças que ocorrem na interface do usuário e se estas atendem aos resultados esperados após cada evento realizado no software. Assim, podemos validar se o comportamento observado do software após tais eventos estão corretos.

Muitas ferramentas de automação provêm funcionalidades de gravar a execução de um software e rodar esta gravação que permitem aos usuários interativamente “filmar” as ações do usuário e repeti-la quantas vezes quiser, comparando os resultados e comportamentos obtidos com aqueles esperados. A vantagem desta abordagem, chamada *capture-replay*, é que ela requer pouco ou nenhum desenvolvimento de software. Nesta abordagem, uma ferramenta de teste grava as entradas de teste como se estivesse sendo submetido ao software que está sendo testado. Os casos de entrada armazenados podem então ser usados para reproduzir os testes posteriormente. Entre as ferramentas disponíveis atualmente para realização de testes de interface do tipo *capture-replay*, uma das mais utilizadas é o framework Selenium IDE, que é um ambiente implementado como um plugin do Firefox e permite a gravação, criação, edição e execução de scripts para testes funcionais para aplicações Web. Ao executar os testes, os scripts simulam a execução dos casos de teste no sistema e ao final da execução, uma barra de status indica se o teste passou ou não. A aplicação desta ferramenta em um exemplo prático foi tema do artigo “Automatizando Testes Funcionais em Aplicações Web”, publicado na edição 24 da ESM. Neste artigo, iremos trabalhar com a aplicação de automação de testes funcionais para aplicações desenvolvidas para dispositivos móveis, como iremos descrever a partir da próxima seção.

## Testes em aplicações para dispositivos móveis

O recente crescimento de dispositivos móveis (aparelhos celulares, *smartphones* e *tablets*) torna-os cada vez mais comuns no nosso dia-a-dia. Com essa evolução tecnológica, surge uma grande demanda por aplicações para esses dispositivos. É possível ter aplicações dos mais variados tipos que complementam as funções de um sistema para dispositivos móveis e compõe um novo mercado que é o de venda de aplicativos,

as chamadas “*app stores*”. Todo ano, milhões de aplicações são criadas para esse tipo de plataforma. Estima-se que em 2013 cerca de sete bilhões de downloads de aplicativos móveis arrecadarão em torno de \$4.2 bilhões de dólares somente nos Estados Unidos.

Com a atual geração de smartphones e tablets, várias empresas estão investindo neste mercado e a cada dia milhares de aplicações são desenvolvidas para as plataformas existentes, dentre as quais, a plataforma Android se destaca como a plataforma de código-fonte aberto mais popular disponível para o desenvolvimento de aplicações móveis. Para desenvolver aplicações Android, existe o ambiente de desenvolvimento padrão fornecido pela Google composto pelo popularmente conhecido ambiente Eclipse IDE, integrado com o Android SDK e com o plugin ADT, que oferecem bibliotecas e ferramentas necessárias para a construção das aplicações Android e podem ser facilmente instalados e configurados. O desenvolvimento destas aplicações é amplamente influenciado pela Google, possibilitando que qualquer pessoa que saiba programar em Java, inclusive pessoas de outras empresas, consiga criar seus próprios aplicativos e utilizá-los em qualquer dispositivo que possua a plataforma.

Com essa grande demanda por aplicativos para dispositivos móveis, faz-se necessário cada vez mais o aumento da produtividade num espaço curto de tempo, o que muitas vezes pode prejudicar a qualidade do produto final. A construção de aplicações de qualidade e que atendam a todas as necessidades dos usuários já deixou de ser um diferencial no mercado e tornou-se uma exigência. Uma das formas de garantir a qualidade destes aplicativos é por meio da aplicação de técnicas de teste de software. Para a plataforma Android, por exemplo, no ambiente de desenvolvimento citado anteriormente é possível realizar testes através do framework de teste da própria plataforma (*Android Testing Framework*). Porém, existem alternativas baseadas neste framework que facilitam a criação e execução de scripts de testes funcionais automatizados para Android, e esta é a proposta da ferramenta apresentada a seguir.

Nos testes de software desenvolvidos para uma plataforma móvel, dificuldades adicionais são introduzidas. Uma das principais dificuldades seria a simulação exata do ambiente onde tal software irá funcionar, devido às limitações de recursos existentes nestes dispositivos e a necessidade de interação do aplicativo com os demais serviços providos pelo dispositivo, tais como atender chamadas, oscilação de carga da bateria, comunicação por redes *wi-fi*, *bluetooth*, acesso à câmera fotográfica, dentre outros, o que não pode ser simulado totalmente pelos emuladores providos para cada plataforma. Assim, uma alternativa seria a realização dos testes com o software já embarcado no dispositivo. No entanto, há uma limitação de técnicas e ferramentas que possibilitam a automação de testes de software nos próprios dispositivos que são o ambiente final do software a ser desenvolvido.

Outra dificuldade está relacionada ao usual curto tempo destinado aos projetos de software para este plataforma devido à

necessidade de disponibilizá-lo rapidamente ao mercado (*time-to-market*), o que requer um gerenciamento de teste de software especializado para este cenário. Contextualizando o cenário de teste de software para plataformas móveis, observa-se que poucas soluções são providas para este cenário, enquanto que um número mais amplo de alternativas é disponibilizado para outras plataformas, como Web e Desktop. Assim, soluções para o teste de software para a plataforma móvel é um grande desafio que se superado pode trazer grande evolução e inúmeros benefícios para essa área.

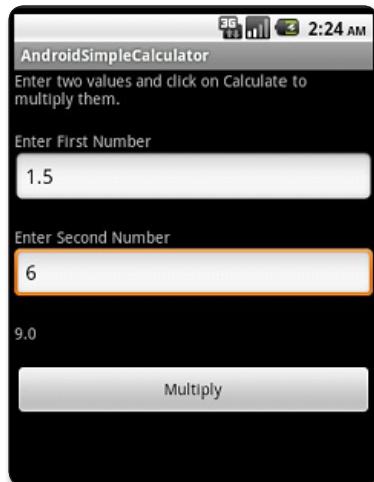
Existem alguns obstáculos em um contexto de testes baseados em modelos para interfaces de dispositivos móveis, tais como a preparação e manutenção dos modelos que representam a interface ao longo das mudanças que ocorrem nos projetos, tempo necessário para o aprendizado da técnica de modelagem, dificuldade da equipe em se adaptar a mudanças, dificuldade em convencer os usuários das vantagens desta estratégia de teste, falta de apoio ferramental e pessoas qualificadas para esta estratégia e existência de outras ferramentas aplicadas a outras plataformas que são adaptadas, nem sempre com sucesso, à plataforma de dispositivos móveis. Também é possível notar que há muito interesse entre os profissionais da área em relação ao assunto. No entanto, é necessário fazer mais pesquisas para tornar o teste de software para plataformas móveis tão fácil quanto possível.

## Robotium – Framework de Automação de Teste para Android

Robotium é um framework para criação de testes funcionais em aplicações para dispositivos Android que possibilita a criação de casos e suítes de teste mostrando o resultado da execução no próprio IDE. Neste, é possível visualizar quais os testes que passaram e quais falharam de forma semelhante aos testes realizados no ambiente Selenium IDE, citado anteriormente neste artigo e altamente conhecido na comunidade de desenvolvimento de software.

Este framework é disponibilizado gratuitamente pelos seus desenvolvedores (ver endereço na seção **Links** ao final do artigo), é regularmente atualizado, possui documentação de suas versões, além de ser facilmente configurado. Ele possui funções baseadas no framework de testes do Android que tornam mais fácil a escrita dos casos de teste. O framework de testes de Android, por sua vez, consiste em uma extensão do framework JUnit, bastante conhecido entre os desenvolvedores de aplicações Java. Outra vantagem, é que ele pode ser usado para testar tanto aplicações cujo código-fonte está disponível, quanto aplicações em que apenas o arquivo *.apk* (executável) está disponível e não se tem acesso aos detalhes de sua implementação.

A partir deste ponto iremos apresentar um exemplo da utilização do framework Robotium na realização de testes funcionais para um aplicativo simples de calculadora (**Figura 1**). Esta aplicação é um exemplo disponibilizado pelos desenvolvedores do Robotium e está disponível também na página do Robotium (ver endereço na seção **Links** ao final do artigo).



**Figura 1.** Tela inicial do aplicativo de calculadora

## Configuração do Framework Robotium

Dando início à parte prática deste artigo, iremos explicar quais os requisitos necessários para configuração do ambiente, e demonstrar como ele pode ser utilizado para apoiar a construção e execução de testes funcionais para aplicações da plataforma Android.

Para utilizar o Robotium, é necessário inicialmente configurar o ambiente com os seguintes requisitos:

1. Eclipse IDE 3.6.2 (ou superior) instalado com o JDK 6 (ou superior);
2. *Android SDK* e *Android Development Tools (ADT)* instalados e configurados de acordo com o passo a passo existente na comunidade de desenvolvedores Android;
3. Criar um *Android Virtual Device (AVD)* - dispositivo virtual onde será feita a simulação da aplicação a ser testada e deve corresponder à versão do Android à qual o aplicativo implementado atende;
4. Criar um *Android Project* no Eclipse IDE com a aplicação a ser testada.

Para quem já desenvolve aplicações Android, até aqui não há nenhuma novidade, pois todos os requisitos citados acima já fazem parte do ambiente de desenvolvimento da plataforma Android. Atualmente, existe até um pacote chamado *ADT Bundle* disponibilizado na página da comunidade de desenvolvedores (veja a sessão de links ao final do artigo) que já inclui os dois primeiros requisitos listados acima, além de outras facilidades que agilizam a configuração do ambiente. A principal diferença está na biblioteca *Robotium.jar* (disponível na página do Robotium referenciada na sessão de links), que deve ser adicionada à *build path* do projeto da aplicação.

## Construção do projeto de teste da aplicação usando Robotium

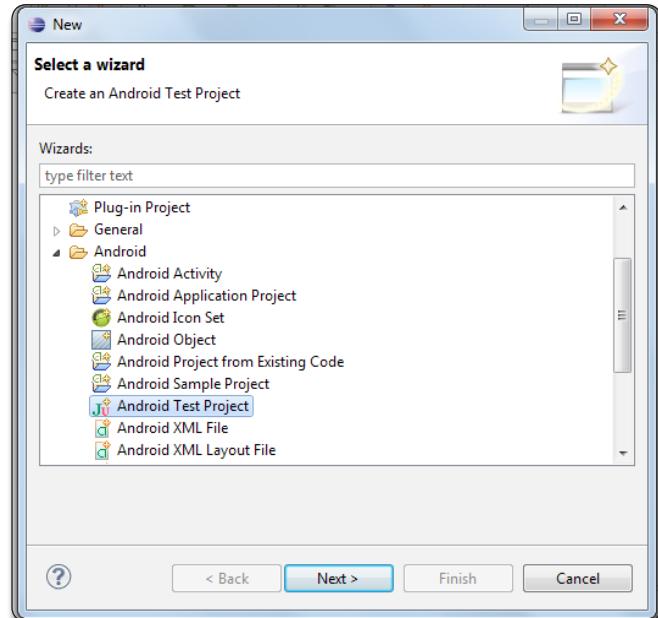
Para testar uma aplicação do Android usando Robotium, é preciso primeiramente criar um projeto de teste associado a esta aplicação, onde iremos adicionar as classes de casos de

teste para testar a principal classe da nossa aplicação (*AndroidCalculatorActivity.java*).

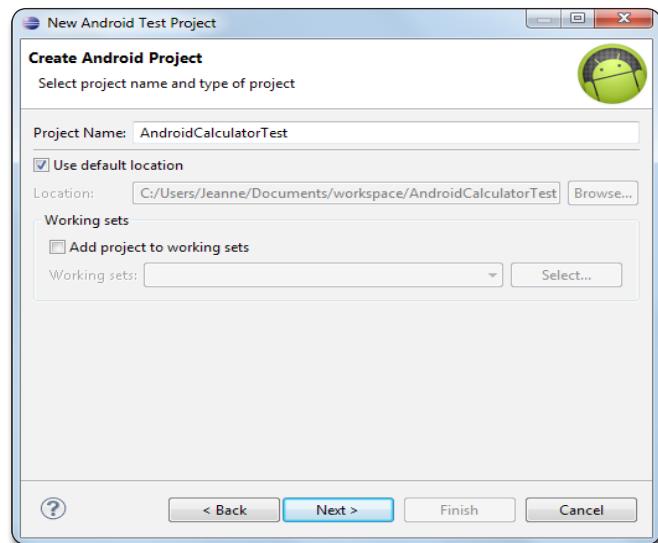
Em algumas versões anteriores do SDK e do Eclipse, era possível criar um projeto de teste no momento em que estava sendo criado o projeto da aplicação. Na versão que estamos usando, essa opção não está mais disponível, então iremos criar um projeto de teste para Android desde o começo.

O primeiro passo é clicar no menu **File**, em seguida **New → Other...**. Na nova janela, expanda a opção **Android** e selecione **Android Test Project** e clique em **Next**, conforme mostra a **Figura 2**.

Na janela a seguir (**Figura 3**), já que iremos testar a aplicação *Android Calculator*, podemos dar ao projeto o nome de “*AndroidCalculatorTest*” que deve ser digitado no campo **Project Name** e prosseguir clicando em **Next**.



**Figura 2.** Janela Select a Wizard para criação do Projeto de Testes



**Figura 3.** Janela Create Android Project

Na janela **Select Test Target** (Figura 4) devemos escolher o projeto ao qual o projeto de teste será associado. Neste caso, deve ser selecionado o projeto da nossa aplicação “*AndroidCalculator*”. Agora podemos finalizar clicando em **Finish**.

Com o projeto de teste criado, agora precisamos criar os casos de testes que irão avaliar o aplicativo.

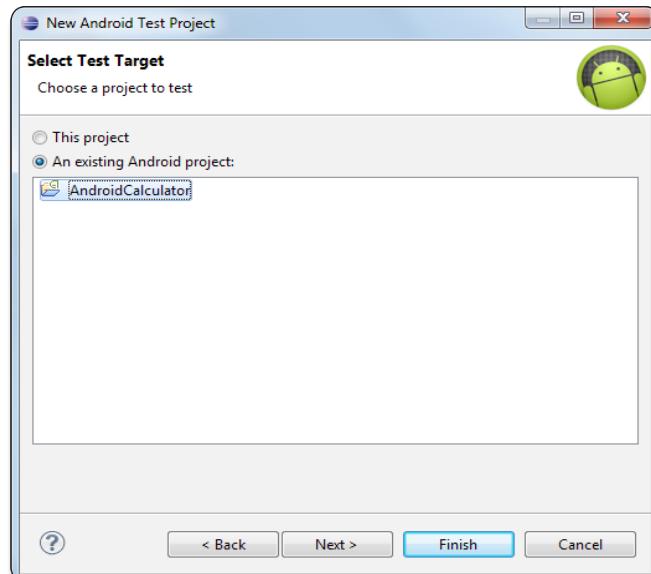


Figura 4. Janela Select Test Target

## Criação dos Casos de Teste

No projeto de teste (*AndroidCalculatorTest*) que aparece na aba **Package Explorer** do Eclipse, clique na pasta *src* e em seguida clique com o botão direito no pacote *com.calculator.test*, selecione **New** e em finalmente na opção *Others*. Na janela **New**, expanda a opção **Java** em seguida **JUnit** e selecione **JUnit Test Case**. Com isso, chegaremos à tela da Figura 5 e então devemos clicar em **Next**.

Na janela **New JUnit Test Case** (Figura 6), digite o nome do caso de teste no campo **Name** (o nosso será “*CalculatorTest*”). Em seguida marque as opções **setUp()**, **tearDown()** e **constructor**. Estas opções farão com que tais métodos sejam criados no caso de teste e eles significam:

- **setUp**: método a ser executado antes da execução do caso de teste. É útil para a configuração de pré-requisitos para o caso de teste sem interferir em sua execução.
- **tearDown**: método a ser executado após a execução do caso de teste. É útil para a finalização de objetos ou qualquer outra configuração a ser desfeita após a execução do caso de teste sem interferir em sua execução.
- **constructor**: método a ser executado no momento da criação do objeto de teste em questão.

Em seguida, clique em **Finish**.

Com isso, um novo caso de teste com o nome *CalculatorTest.java* será criado dentro do pacote *com.calculator.test* com os métodos citados anteriormente já criados, porém sem conteúdo, conforme mostra a Figura 7.

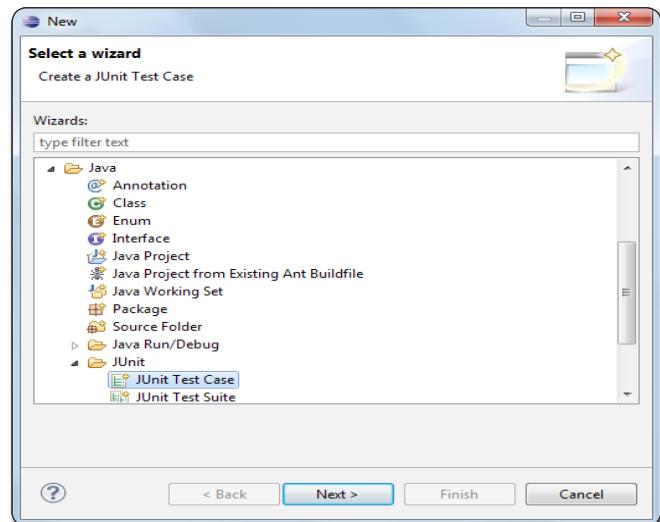


Figura 5. Janela Select a wizard

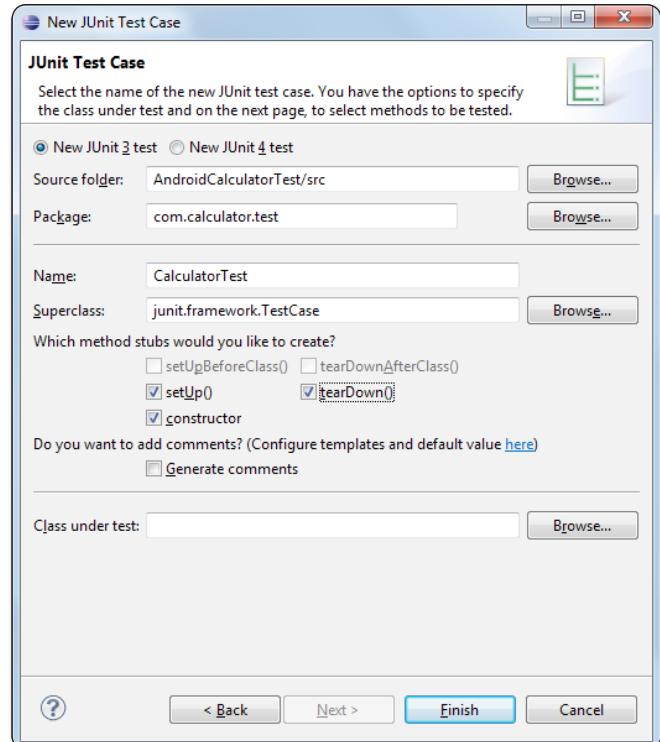


Figura 6. Janela JUnit Test Case

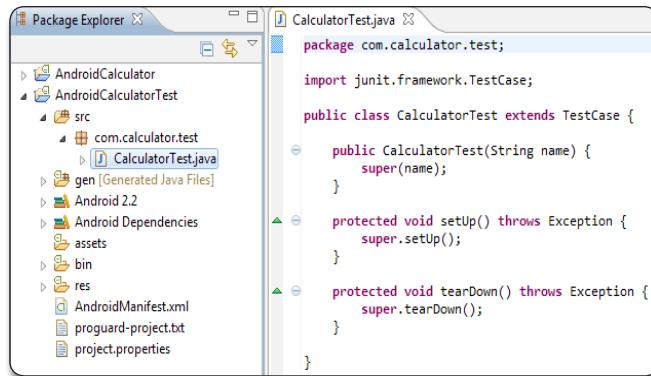
Antes de criarmos o código do nosso caso de teste, iremos adicionar o framework da ferramenta Robotium ao ambiente Eclipse, que possibilitará a automatização do nosso teste.

## Adicionando o arquivo *Robotium.jar* ao Eclipse

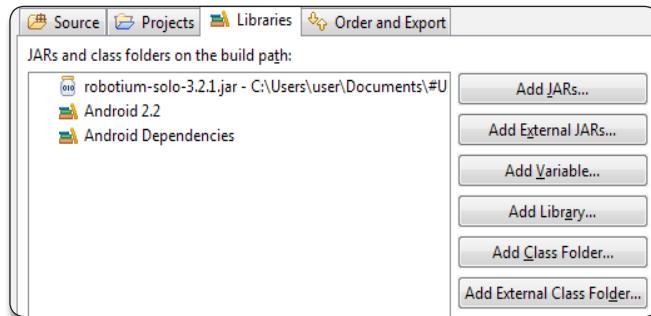
A versão do arquivo “*robotium.jar*” utilizada neste artigo é a versão 3.2.1, que pode ser baixada através da página de downloads no site do Robotium. Para uso do framework Robotium, precisamos referenciá-lo em nosso projeto. Isso pode ser feito clicando-se com o botão direito no projeto de teste,

selecionando **Build Path**, e então clicando em **Configure Build Path**. Na janela **Properties**, devemos clicar na aba **Libraries** e em seguida clique no botão **Add External JARs** para adicionar o arquivo “robotium.jar” baixado ao projeto (**Figura 8**).

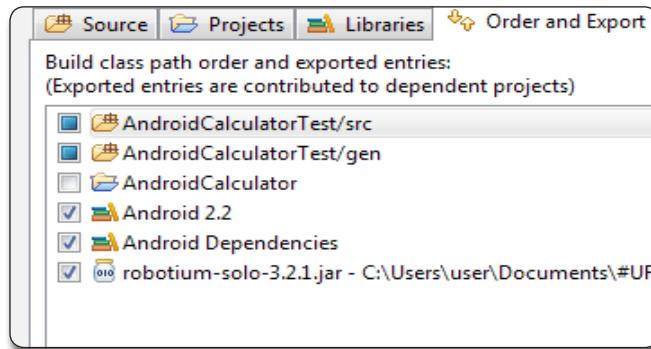
Em seguida, clique na aba **Order and Export** e marque as opções **Android 2.2**, referente à versão do Android que o aplicativo implementado atende **Android Dependencies** e o arquivo do Robotium (**Figura 9**).



**Figura 7.** Classe de teste “CalculatorTest.java”



**Figura 8.** Adição do Robotium ao Eclipse



**Figura 9.** Seleção das classes adicionadas na aba Libraries

## Escrevendo o código do caso de teste

O caso de teste deste exemplo, como pode ser visto na **Listagem 1**, irá acessar a aplicação *AndroidCalculator* e realizar dois testes com valores diferentes, onde cada teste realiza os seguintes passos:

1. Chamar/acessar os dois primeiros campos de edição (*EditField*).

2. Digitar os valores a serem multiplicados.
3. Acessar e clicar no botão *Multiply*.
4. Realizar uma assertiva para verificar o resultado da multiplicação no campo do resultado.

Este código deve ser copiado para a classe *CalculatorTest.java* criada no projeto de teste.

### Listagem 1. CalculatorTest.java

```

01 package com.calculator.test;
02
03 import android.test.ActivityInstrumentationTestCase2;
04 import com.calculator.AndroidCalculatorActivity;
05 import com.jayway.android.robotium.solo.Solo;
06
07 public class CalculatorTest extends
ActivityInstrumentationTestCase2<AndroidCalculatorActivity> {
08 private Solo solo;
09
10 public CalculatorTest() {
11     super("com.calculator", AndroidCalculatorActivity.class);
12 }
13@Override
14protected void setUp() throws Exception {
15     super.setUp();
16     solo = new Solo(getInstrumentation(), getActivity());
17 }
18public void testValues1() { //Primeiro caso de teste
19    solo.enterText(0, "1"); //Digita 1 no primeiro editfield
20    solo.enterText(1, "0"); //Digita 0 no segundo editfield
21    solo.clickOnButton("Multiply"); //Clica no botão Multiply
22    assertTrue(solo.searchText("0")); //Valida o resultado de 1x0
23 }
24public void testValues2() { //Segundo caso de teste
25    solo.enterText(0, "10"); //Digita 10 no primeiro editfield
26    solo.enterText(1, "20"); //Digita 20 no segundo editfield
27    solo.clickOnButton("Multiply"); //Clica no botão Multiply
28    assertTrue(solo.searchText("200")); //Valida o resultado de 10x20
29 }
30@Override
31protected void tearDown() throws Exception{
32     solo.finishOpenedActivities();
33 }
34}

```

Os testes criados neste exemplo são apenas demonstrativos e cobrem as duas possíveis saídas para o sistema (produto da multiplicação com valor 0 e outro produto com valor diferente de 0). Cada caso de teste utilizou três métodos do Robotium e um método do framework JUnit, outro conhecido framework para testes também utilizado pelo framework de teste nativo do Android. Os métodos do Robotium utilizados nos casos de teste foram:

- *void enterText(android.widget.EditText editText, String text)*: simula a entrada de dados em um campo de texto editável da interface por meio de seus parâmetros;
- *void clickOnButton(String name)*: clica no botão cujo nome é dado por parâmetro;
- *boolean searchText(String text)*: função que procura por uma string dada por parâmetro na interface do sistema e retorna verdadeiro se ao menos um item é encontrado na tela do aplicativo com o texto da string a ser buscada.

Em seguida, é utilizado o método `assertTrue(boolean condition)`, nativo do JUnit para validar se a condição dada por parâmetro é verdadeira. Caso seja, o teste passa, caso contrário, ele falha.

Para cada versão do framework Robotium que é atualizada, também é disponibilizado um arquivo `javadoc` onde são listados (entre outras informações) os métodos criados neste framework e uma breve descrição de cada um deles. Este documento pode ser encontrado na seção de downloads na página do Robotium (veja na sessão **Links**) e pode ser útil para que os desenvolvedores conheçam a abrangência desta ferramenta na aplicação de testes funcionais automatizados para a plataforma Android. Tais métodos implementam simulações de diferentes ações que podem ser realizadas por meio da interface gráfica de um aplicativo Android, tais como preenchimento de dados, clique de botões, seleção de itens em uma lista, tirar uma foto da tela, entre outras ações. Estes métodos se assemelham a métodos providos em outros frameworks para testes funcionais a partir de interface gráfica como, por exemplo, o framework Selenium para aplicações Web, já citado anteriormente.

Os principais métodos providos pelo framework Robotium para tratar de testes para aplicações que rodam em dispositivos móveis são:

- `assertMemoryNotLow`: verifica se a memória disponível não é considerada baixa pelo sistema.
- `clearEditText`: limpa o valor de um `EditText`.
- `clickLongOnScreen`: longo clique nas coordenadas especificadas.
- `clickMenuItem`: clica no `MenuItem` relacionado ao texto indicado. Existem métodos similares para elementos do tipo `edit`, `image`, `checkbox`, `radiobutton`, `screen`, dentre outros.
- `drag`: simula tocar na tela e arrastar de um local para outro.
- `goBack()`: simula pressionar o botão de voltar do dispositivo.
- `hideSoftKeyboard()`: oculta o teclado virtual.
- `pinchToZoom`: realiza *zoom in* ou *zoom out*.
- `scrollDown()/scrollUp()`: rola a tela para baixo/cima.
- `scrollViewToSide`: rola a tela para o lado especificado.
- `sleep`: pausa o Robotium por um tempo especificado.
- `takeScreenshot()`: tira uma foto da tela.
- `waitForText`: espera um determinado texto aparecer na tela.

Além dos métodos específicos do framework Robotium, podem ser utilizados métodos nativos do framework JUnit, principalmente aqueles que realizam assertivas para verificação dos resultados dos testes funcionais.

Os principais métodos providos pelo framework JUnit (veja na sessão **Links**) e que são aplicáveis no framework Robotium são:

- `assertEquals`: verifica se dois números do tipo `double` ou `float` são iguais em relação a um número “*delta*” positivo.
- `assertFalse/True`: verifica se uma condição é falsa/verdadeira.
- `assertNull/NotNull`: verifica se um objeto é (ou não) nulo.
- `assertSame/NotSame`: verifica se dois objetos se referem (ou não) ao mesmo objeto.
- `fail`: falha um teste apresentando uma mensagem específica.

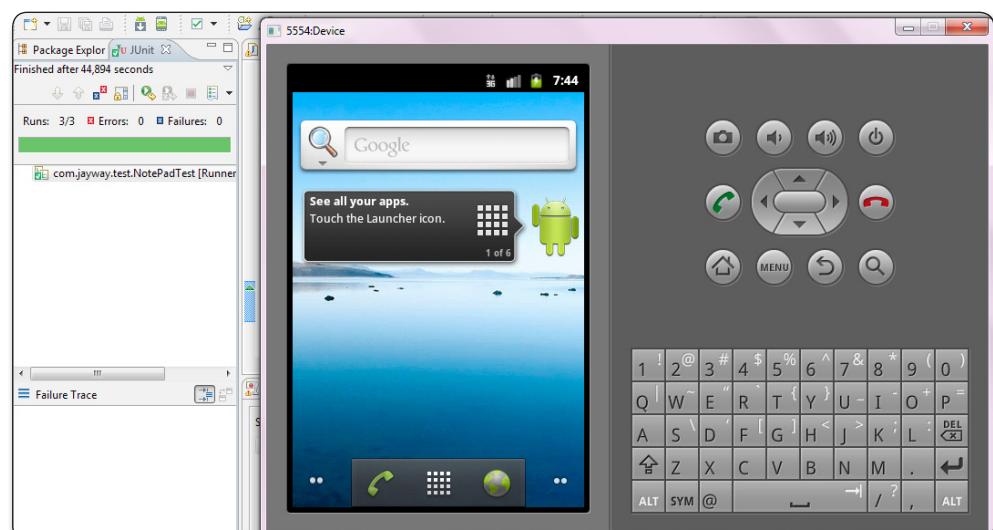
## Executando os testes

Para executar o teste criado, basta clicar com o botão direito no arquivo `CalculatorTest.java`, selecionar a opção *Run As* e finalmente clicar em *Android JUnit Test*. É necessário também selecionar o simulador, previamente criado durante o projeto da aplicação, e esperar enquanto o teste é carregado.

Ao ser executado com sucesso:

1. O simulador irá carregar e, caso necessário, desbloqueie a tela para que a aplicação possa ser executada.
2. A aplicação *AndroidCalculator* irá carregar.
3. Ele irá automaticamente entrar com os valores configurados nos casos de teste e clicar no botão *Multiply* (exatamente como nos passos do script).
4. Após ter executado a assertiva e o resultado for igual ao esperado pelo script, irá aparecer uma barra verde no Eclipse indicando que a execução foi bem sucedida e que todos os testes passaram, conforme mostra a **Figura 10**.

Em caso de falha na assertiva, por exemplo, se o valor que foi colocado como resultado esperado estiver incorreto, aparecerá uma barra vermelha no Eclipse indicando a falha. Neste último caso, mesmo que apenas um dos casos de teste tiver falhado e os outros passarem, a barra indicará uma falha, porém, acima da barra podemos ver a quantidade de testes que passaram e a quantidade dos que não passaram.



**Figura 10.** Resultado da execução do teste funcional

## Conclusão

Apesar de o framework de teste nativo do Android já oferecer inúmeras funções e ferramentas que possibilitam a criação de todo tipo de teste, a complexidade deste framework pode, em alguns casos, tornar a escrita dos casos de teses mais complicada.

Para facilitar o desenvolvimento dos testes, algumas alternativas têm sido criadas, e o framework Robotium tem sido uma das mais bem sucedidas já que é possível fazer quase tudo que o framework nativo faz, porém com métodos que facilitam e tornam a elaboração dos teses muitos mais simples e rápida. Ele ainda possui algumas limitações que outros frameworks alternativos tentam suprir, mas ainda assim é bastante útil, considerando também o suporte que é dado pela comunidade de seus desenvolvedores.

### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



### Links e Referências:

#### Página oficial do Robotium

<https://code.google.com/p/robotium/>

#### Aplicativo Calculadora (Exemplo)

[https://robotium.googlecode.com/files/ExampleTestProject\\_v3.0.zip](https://robotium.googlecode.com/files/ExampleTestProject_v3.0.zip)

#### Tutorial de Criação do Aplicativo Calculadora

<https://robotium.googlecode.com/files/CreateAndroidSampleApplication.pdf>

#### Arquivo javadoc do Robotium com a listagem dos métodos

<https://robotium.googlecode.com/files/robotium-solo-4.2-javadoc.jar>

#### Arquivo javadoc do JUnit com a listagem dos métodos

<http://junit.sourceforge.net/javadoc/org/junit/package-summary.html>

#### ADT Bundle

<http://developer.android.com/sdk/index.html#>

**Collins, Eliane F. et. al., "Experiência em Automação do Processo de Testes em Ambiente Ágil com SCRUM e ferramentas OpenSource", Simpósio Brasileiro de Qualidade de Software, 2010.**

**Dias-Neto, A. C.; Introdução a Teste de Software. Em: Engenharia de Software Magazine, Edição 01, pp. 54-59**

<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035>

# FÓRUM DEVMEDIA

O lugar perfeito para você ficar por dentro de tudo o que acontece nas tecnologias do mercado atual

No fórum da DevMedia você irá encontrar uma equipe disponível e altamente qualificada com consultores e colaboradores prontos para te ajudar a qualquer hora e sobre qualquer assunto.

Temos as salas de Java, .NET, Delphi, Banco de Dados, Engenharia de Software, PHP, Java Script, Web Design, Automação comercial, Ruby on Rails e muito mais!



**ACESSE AGORA**  
[www.devmedia.com.br/forum](http://www.devmedia.com.br/forum)

Nesta seção você encontra artigos voltados para diferentes abordagens de apoio ao desenvolvimento de projetos de software



## Arquiteturas de sistemas Web 3.0

Concepção arquitetural de sistemas concebidos pela Internet e pela engenharia de software modernas

**E**m tempos longínquos, a arquitetura de software era basicamente voltada para os sistemas cliente-servidor, onde uma máquina exercia o papel de cliente requisitante e a outra máquina o papel de servidor com a responsabilidade de atender as requisições. Deste modo, a arquitetura cliente-servidor tornou-se padrão na engenharia de software. Atualmente tal arquitetura ainda é muito utilizada e, sobretudo, mesclada a outras arquiteturas para atender ao objetivo de um software.

A mistura de estilos arquiteturais possibilita a existência de arquiteturas cada vez mais heterogêneas, onde estilos e tecnologias diferenciadas formam novas versões de representações arquiteturais. Contudo, por que as arquiteturas estão tão heterogêneas? Ao que se baseia essa afirmação? Tais questionamentos poderão ser inexistentes ao final da leitura, contudo cabe um simples raciocínio no próximo parágrafo que auxiliará no entendimento futuro.



**Elaine G. M. de Figueiredo**

[mira.figueiredo@gmail.com](mailto:mira.figueiredo@gmail.com)

Mestrado em Ciências da Computação com ênfase em engenharia de software, pós graduação em gerência de projetos, oito anos de atuação na área de desenvolvimento e qualidade de software. Atualmente é gerente de projetos na multinacional espanhola Indra Company, membro colaborador do grupo de pesquisa "ASSERT – Advanced System and Software Engineering Research Technologies Lab" pela universidade federal de Pernambuco.

### Porque esse artigo é útil:

Este artigo é relevante para aqueles que necessitam estruturar uma aplicação preparada para funcionar sobre as condições diversas que a Internet moderna pode oferecer. Na verdade, a utilidade é nortear os desenvolvedores para boas práticas de concepção arquitetural, ao passo que auxilia no entendimento de uma engenharia de software cada vez mais voltada para novos padrões de negócios e tecnologias.

A construção de sistemas está cada vez mais imersa nas inovações tecnológicas, inovações causadas pelo surgimento de novos negócios, novos frameworks, novas linguagens e novos protocolos que surgem em um intervalo cada vez menor de tempo. A tendência também é em projetar sistemas que são mais dependentes da Internet, a fim de atender aos modelos de negócio mais exigentes, com regras mais complexas.

Neste aspecto o que parece é haver um cenário muito favorável à existência de uma engenharia de software, onde a arquitetura de software está imersa, ambas focadas em acompanhar tais

exigências e inovações, bem como a trabalhar com uma Internet programável, caracterizada como uma moderna rede de computadores sociáveis e interoperáveis.

## A Internet como máquina propulsora

A Internet vem alcançando números gigantescos. Como exemplo, a quantidade de usuários apenas na América Latina chega a aproximadamente 143 (cento e quarenta e três) milhões, segundo dados do site INTERNET WORLD STATS especializado em pesquisas estatísticas sobre a internet. Os meios de acesso são variados, desde conexão banda larga até telefones celulares com tecnologia 3G. Os serviços também são diversos: ferramentas de buscas, blogs, sites empresariais, e-mail, ferramentas de comunicação instantânea, comércio eletrônico, abreviadores de URL, ferramentas de programação e etc.

Na Internet as empresas públicas, privadas e governamentais trocam informações importantes, vários paradigmas e tecnologias surgem em prol da Internet, além das transações bilionárias como, por exemplo, a compra de sites e serviços. Assim, a Internet se caracteriza como um rico instrumento de comunicação, negócios, transações, propagandas e ensino. Sobre o aspecto tecnológico, o grande feito da Internet foi sua evolução.

A Internet evoluiu de uma rede anteriormente classificada como Web 1.0, onde as informações somente poderiam ser visualizadas, para a rede programável, a Web 3.0. A Web 3.0 é classificada por meio de dois conceitos distintos. Um deles considera a Web 3.0 como uma plataforma de desenvolvimento de softwares independentes de infraestrutura física local, conceito aplicado por Marc Benioff. A outra ideia classifica a Web 3.0 como a Web Semântica, segundo BERNERS-LEE em 2001. Independente da conceituação, a Web 3.0 mudou definitivamente o aspecto da Internet, transformando a mesma em algo mais interativo, inteligente e independente.

É dentro desta temática de Internet programável, ou Web 3.0, que este artigo estabelecerá suas bases. Essa Web fez a engenharia de software se transformar procurando acompanhar tal evolução da Internet para que os aplicativos construídos por meio desta, ou para esta, fossem concebidos e estruturados de forma satisfatória. Isto já configura a questão das inovações tecnológicas citadas na introdução do texto.

## Uma nova geração de aplicativos web

Existe uma nova geração de aplicativos Web. Isto fica evidente quando pensamos que os mesmos serviços disponibilizados pela rede mundial há 10 (dez) anos, agora são disponibilizados pela mesma rede com mais completude e eficiência. Um exemplo disto se vê na comparação da maior rede social do mundo, o Facebook, tendo mais de setecentos milhões de usuários, com o SixDegrees, a primeira rede social do mundo, criada em 1997, que atingiu o máximo de um milhão de usuários.

O SixDegrees foi descontinuado no ano de 2000, pois o projeto apresentou sérios problemas financeiros que foram causados pelos poucos acessos feitos pelos usuários. Tais usuários não se sentiram atraídos pela rede, pois não havia muito que fazer

após “adicionar amigos”, apenas o envio de mensagem entre perfis da rede. Ou seja, o SixDegrees não possuía bons recursos tecnológicos para se gerar e utilizar *apps* na rede social, o que agregaria funcionalidades interessantes aos usuários, como: jogos, trocas de vídeos, fotos ou URLs. Apesar dos pontos negativos, o SixDegrees foi a primeira rede social a possibilitar a criação de um perfil virtual combinado com o registro e publicação de contatos, o que viabilizou a criação e navegação dos usuários por outras redes sociais que surgiram posteriormente, como o: Friendster, Ryze e Fotolog.

Em contrapartida, o Facebook é um serviço de alta disponibilidade que oferta desde uma ferramenta de conversação imediata até APIs Java para a reutilização de seus componentes por outras aplicações, ou seja, ele é uma rede social programável que qualquer instituição, ou grupo de pessoas, pode formar sua própria rede semelhante ao Facebook, utilizando para isso apenas a API da própria rede social. O Facebook fortaleceu consideravelmente a popularização das redes sociais. O trabalho dos usuários com estas redes é tão forte que uma pesquisa estatística feita pelo ConScore, Inc., mostrou um aumento de 4% nos acessos das mesmas somente nos primeiros seis meses do ano de 2010.

A facilidade em se reprogramar as redes sociais faz com que elas compõem a Web 3.0 reprogramável. Entretanto, nesta Internet ainda ocorre a proliferação dos Mashups, que são aplicações Web constituídas de outras aplicações. Com os Mashups é possível ter uma combinação de serviços em um único aplicativo (Benslimane, Dustdar, e Sheth, 2008). Um bom exemplo de Mashups são os sites de imobiliárias que inserem o serviço do Google Maps para que o usuário possa fazer a localização do imóvel nas ruas da cidade. Vale ressaltar que os Mashups personificam muito bem a comunicação e interação entre as aplicações Web.

As aplicações Web 3.0 adicionam um diferencial na Internet, inovações ao seu desenvolvimento e facilidade aos negócios, tudo isto a fim de atender aos modelos de negócio cada vez mais exigentes, com regras mais complexas. Dentro deste contexto a Internet desempenha um papel fundamental, além de instrumento, passa a ser provocadora de negócios, de inovações e de junções empresariais.

Por tudo que foi exposto até o momento, pode-se verificar que atualmente já se vive neste contexto de Web 3.0. Na próxima seção será possível verificar que a engenharia de software acompanhou tal evolução da rede mundial, ao mesmo tempo em que oferecer suporte aos desenvolvimentos das aplicações Web 3.0.

## Preocupações de uma Engenharia de Software Moderna

As aplicações Web 3.0 possuem características interessantes, pois elas são elementos absolutamente sociáveis, ou seja, interagem com a Internet e com os outros sistemas ao seu redor. Tais aplicações também devem ser independente de qualquer tecnologia, justamente para ter autonomia de funcionamento sobre qualquer circunstância. Tais aplicações

devem saber encontrar os serviços que lhe possam ser úteis na Internet. Outra importante característica de uma aplicação Web é a capacidade de oferecer seus serviços de modo que outras aplicações não necessitem saber, ou se preocupar com a forma como, quando e onde os serviços foram implementados.

Tendo conhecimento de tais características, a engenharia de software precisa atender as demandas de construção dessas aplicações Web modernas. Para isso, no entanto, precisa resolver problemas clássicos na construção de sistemas, como: manutenções demoradas e críticas; requisitos mal trabalhados; ausência de arquitetura; ausência de uma estratégia de negócio que adapte aplicações Web à resolução dos problemas, entre outros pontos.

Além das questões supracitadas, a engenharia de software precisará ofertar soluções que viabilizem:

- O desenvolvimento de aplicações interoperáveis que estão cada vez mais entrelaçadas e dependentes dos serviços umas das outras;
- O desenvolvimento de software sob demanda, pois estes necessitam de uma arquitetura diferenciada, haja vista, eles sejam desligados de um repositório físico para armazenamento, compartilhando tal repositório com outro software (conceitos de Software as a Service – SaaS, estes não precisam de servidor físico, eles são armazenados em servidores virtuais, os chamados *Cloud*, estes podem comportar vários SaaS e distribuem os recursos, como memória por exemplo, entre eles);
- O autogerenciamento de sistemas Web que compartilham atividades com outros sistemas distribuídos na rede.

Projetos de aplicações Web estão suscetíveis aos problemas pontuados. Uma vez que os problemas aconteçam, as iniciativas para saná-los devem ser precisas e rápidas. Dependendo da natureza do problema, o projeto pode até ser descontinuado. Apenas para exemplificar, este foi o caso de algumas ferramentas, o Google, no ano de 2009, precisou cortar gastos, com isso resolveu paralisar o desenvolvimento de alguns de seus produtos que geravam problemas, entre eles estava o Google Mashups Editor. O mesmo ocorreu com o Microsoft Popfly também em 2009. Este caso poderia ocorrer a qualquer aplicação Web que contivesse alguns dos problemas citados.

Alguns dos maiores problemas relacionados ao desenvolvimento de aplicações Web, os quais a engenharia de software deve tratar, são os oriundos da arquitetura. A arquitetura de software é uma ferramenta que auxilia na descoberta precoce de problemas que só se manifestam geralmente na codificação da aplicação ou quando a mesma está desenvolvida.

## Arquiteturas para a Web 3.0

Nesta seção serão expostos dois exemplos de arquitetura moderna. Trata-se de uma arquitetura *Cloud Computing*, que se preocupa em otimizar a disponibilização e o compartilhamento dos recursos computacionais na Internet. A outra arquitetura abordada será a das redes sociais.

## Arquitetura Cloud Computing

A Figura 1 retrata uma das arquiteturas concebidas para o paradigma *Cloud Computing*. Ela é constituída pela camada *Cloud Applications* que é a mais básica, trata-se de uma camada de aplicação onde os usuários finais podem manter seus programas na nuvem. Nesta camada também estão os aplicativos e os SaaS, todas as aplicações podem ser adquiridas pelos usuário de forma dinâmica conforme a necessidade do mesmo. Abaixo desta camada está a camada *Cloud Programming*, a relação entre elas reside no fato da camada inferior dar suporte à camada superior, sendo assim, a camada mais baixa *Cloud Programming* disponibiliza os serviços e os recursos para se implementar o software sob demanda.

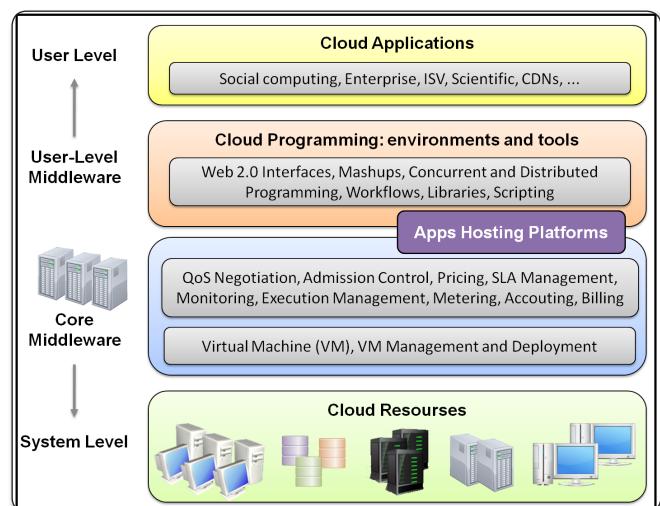


Figura 1. Arquitetura Cloud Computing – Fonte: Adaptada da literatura “Software platform for.NET-based Cloud Computing”

A *Cloud Programming* também pode ser chamada de *Middleware User-Level*. Nela podem ser desenvolvidos os aplicativos e regulado o suporte ao desenvolvimento. São basicamente ferramentas e ambientes que podem possuir interface Web 2.0, recursos de programação distribuída, bibliotecas e linguagens de programação, além de ferramentas para o desenvolvimento de sistemas Mashups. Esta camada é similar a uma PaaS.

A penúltima camada de *Middleware Core* é responsável pelo gerenciamento da infraestrutura física. Tal camada oferta serviços para a manutenção de QoS e SLAs, serviços de virtualização e outros. Todos os recursos dispostos nesta camada são acessados pela camada diretamente acima dela, pois somente assim, o usuário poderá usufruir dos recursos, haja vista o acesso à camada *Middleware Core* seja negado.

Por fim, a camada de mais baixo nível é a de infraestrutura física que contém *Data Centers* e conjuntos de CPUs, além de outros recursos de hardware, os quais podem ser agregados ou subtraídos com liberdade, oferecendo assim, bastante flexibilidade à arquitetura. Estas duas últimas camadas inferiores se comportam como IaaS.

O estilo da arquitetura Cloud Computing abordada é tipicamente em Camadas, toda camada tem uma responsabilidade distinta, além disto, os recursos de cada camada podem ser reorganizados, pois a própria vai se readaptando. Deste modo, quando houver necessidade de economia no investimento, e consequentemente a retirada de uma camada, as outras estarão preparadas; enquanto isso, o gerenciamento de cada camada pode ser feito independentemente.

Esse modelo arquitetural, em camadas, pode ofertar soluções para a manutenção da aplicação e independência entre as partes que a compõem, e certamente foi uma necessidade do negócio adotar tal estilo; tal necessidade só foi verificada durante o processo da engenharia de software voltado para a concepção arquitetural.

### **Arquitetura de redes sociais**

Aproveitando que o modelo arquitetural em camadas já foi comentado na arquitetura anterior, as arquiteturas investigadas para as redes sociais são também baseadas no estilo camadas. Entretanto, esse estilo arquitetural ostenta “formatos” diferenciados que dependem das características e não apenas dos requisitos de cada rede. O mesmo formato arquitetural é explorado tanto pelo microblog Twitter quanto pelo Facebook. No entanto, são abordagens diferentes da mesma arquitetura, explorando tecnologias diferentes.

Vale salientar que as redes sociais exploradas nessa seção são plataformas programáveis. Com elas, é possível criar novas aplicações utilizando-se apenas a Internet. A seguir, exploremos uma breve sinopse sobre cada rede, com as respectivas tecnologias adotadas e sua arquitetura. O Twitter iniciará a seção.

O Twitter foi aberto ao público em meados do ano de 2006, além de ser um microblog ele é uma rede social propriamente dita e ainda uma ferramenta de Broadcast. A rede, em abril do ano de 2010, segundo seu cofundador Biz Stones, conquistou a cifra de 105 milhões de usuários. Abaixo, delineamos algumas características importantes dessa rede.

O Twitter pode ser acessado via dispositivos móveis. Atualmente o microblog possui integração com outras redes sociais. No ano de 2008, foi incorporado um mecanismo de busca ao microblog. Outra característica importante do Twitter é a sua API. Ela oferece, entre outros, a busca de serviços que possam ser indexados em aplicações terceiras. A arquitetura do microblog é ilustrada na **Figura 2**.

Na arquitetura Twitter a camada de apresentação trabalha com os serviços dos clientes, algumas tecnologias são empregadas como: JSON e Rails (ver **BOX 1**), a camada realiza ainda renderização e caching. Logo abaixo está a camada de negócio, trabalhada por meio das linguagens C, Java e Scala. Esta última favorece a escalabilidade em relação ao crescimento do número de seguidores. Por último encontra-se a camada de dados suportada por bancos de dados não relacionais, os chamados NoSQL, que oferecem resiliência e escalabilidade; como exemplo desses bancos encontra-se o Cassandra e o HBase. Em seguida examinaremos a arquitetura do Facebook.

#### **BOX 1. JSON e Rails**

JSON é uma notação JavaScript que possibilita a troca de dados entre máquinas. Já o Rails, também chamado de Ruby on Rails, ou apenas RoR, é um framework de código aberto escrito na linguagem Ruby. Seu objetivo é aumentar a velocidade e a facilidade de desenvolvimento dos sites.



**Figura 2.** Arquitetura simplificada do Twitter

A arquitetura do Facebook contém uma camada a mais em relação ao Twitter, uma camada dedicada à comunicação, pois o Facebook trabalha com recursos de mídia mais complexos, como fotos e vídeos. Esta rede ainda possui um chat para conversação entre as pessoas conectadas e registradas como amigos. Por essas características e outras, o Facebook, que hoje é a maior rede social do mundo, tornou-se também um desafio para a escalabilidade.

Para prover a escalabilidade, os arquitetos do Facebook usam uma combinação de tecnologias, principalmente softwares para escalar, como o Map Reduce para processar e analisar grande volume de dados. Utiliza ainda o Varnish como平衡ador de carga e o Haystack para a recuperação de fotos.

O Facebook além de ser a rede social mais notável da atualidade, se caracteriza fortemente por ser uma plataforma para geração de aplicativos e serviços, que disponibiliza uma API e possui integração com plataformas diferenciadas. Apesar disso, as arquiteturas das duas redes possuem particularidades.

O serviço de Cloud Computing auxiliou as redes sociais no armazenamento de dados e na disponibilidade de recursos, o que facilitou a escalabilidade. Tanto Facebook quanto Twitter utilizam banco de dados NoSQL. O software Memcached (ver **BOX 2**), por sua vez, foi implementado dentro da camada de negócio de ambas as redes para melhorar o desempenho, visto que o Memcached armazena a informação em memórias distribuídas, recuperando-a mais rapidamente, evitando assim, tarefas repetitivas e demoradas de recuperação de informação em banco de dados.

Concluindo esta sessão, vale registrar que a arquitetura do Facebook não foi ilustrada neste capítulo, pois não foram encontrados materiais que explicassem em detalhes maiores como as camadas de tal arquitetura interagem.

Requisitos	Características e necessidades das aplicações Web	Solução de atendimento
Performance	Reatividade, Infraestrutura, Autonomia, expansão e Serviços	1. Controle sobre a geração de eventos; 2. Diminuir os elementos intermediários de comunicação; 3. Trabalhar bem com a infraestrutura e com a conectividade.
Disponibilidade	Constância, Sociabilidade, Reatividade, Infraestrutura, Serviços, Colaboração e Comunicação facilitada	1. Redundância: caso um elemento fique indisponível, a redundância (outro elemento igual) deve ocupar o seu lugar; 2. Mecanismos que verifiquem e alertem sobre possíveis falhas.
Modificabilidade	Sociabilidade, Manutenção facilitada, e Autonomia	1. Elementos arquiteturais que facilitem a manutenção; 2. Separar responsabilidades em camadas; 3. Manter a coerência semântica para facilitar a identificação do elemento arquitetural; 4. Isolar funcionalidades que possam ser facilmente alteradas; 5. Inserir elementos intermediários que diminuam a dependência.
Interoperabilidade	Colaboração, Infraestrutura, Sociabilidade e Conectividade.	1. Identificar e trabalhar com as interfaces providas e requeridas; 2. Trabalhar com padrões de interoperabilidade.

**Tabela 1.** Necessidades e soluções arquiteturais de aplicações web reprogramáveis

#### BOX 2. Memcached

É um sistema de cache de objetos em memória concebido para aumentar a velocidade de aplicações dinâmicas aliviando a carga do banco de dados.

Os materiais capturados faziam referência apenas às decisões de projeto e tecnologias adotadas. Aliás, as arquiteturas das redes sociais exploradas trabalham mais com questões tecnológicas. Neste aspecto foi possível notar que estas arquiteturas se diferem mais em relação às decisões de projeto, ou seja, de quais tecnologias utilizarem.

Para a definição de qualquer arquitetura é fundamental declarar as decisões de projetos. Estas estão diretamente ligadas ao design da arquitetura. Assim, os resultados da questão sobre as arquiteturas de redes sociais se adaptam mais às decisões de projeto.

#### Necessidades das arquiteturas web

O processo de concepção de uma arquitetura é por vezes mal executado ou não executado. Um exemplo clássico de problemas com arquitetura, ou projeto arquitetural, foi o caso do Twitter (arquitetura aqui já explorada). O micro blog trocou seu banco de dados MySQL pelo Cassandra em virtude do alto crescimento da taxa de dados transitados. No momento das decisões arquiteturais do Twitter não houve um raciocínio sobre o crescimento exacerbado do mesmo, com isto optou-se pelo banco de dados errado.

Uma forte característica da engenharia de software das aplicações Web atuais é especificar e desenvolver aplicações distribuídas e conectáveis, aplicando métodos e tecnologias que tratem corretamente suas particularidades, não as deixando vulneráveis aos problemas. A engenharia de software deve mensurar ainda na concepção da arquitetura o tratamento de muitos problemas.

Desta maneira a problemática focada se relaciona seriamente com a necessidade de se especificar corretamente sistemas da Web, neste sentido, a arquitetura que é o arcabouço de qualquer software, faria total diferença.

Toda arquitetura deve ser coerente aos requisitos não funcionais. Com isto, na exibição de uma arquitetura, é importante deixar claro quais os requisitos atendidos por elas. A Tabela 1 exibe outras necessidades e soluções (decisões de projeto) dos principais requisitos não funcionais de uma atual aplicação Web reprogramável.

#### Conclusão

Neste artigo houve a preocupação em mostrar características que diferenciam os sistemas e, sobretudo, as arquiteturas contemporâneas; mas do que isso mostrou-se brevemente soluções de projetos arquiteturais ou ainda estratégias que podem viabilizar a obtenção de requisitos fundamentais aos sistemas Web atuais.

Em se tratando de sistemas construídos para a Web e pela Web, a interoperabilidade é um ponto preocupante na análise deste estudo, os elementos arquiteturais devem trabalhar bem sobre este ponto, arquiteturas orientadas a serviços, e adoção de protocolos como REST e SOAP, são iniciativas inteligentes que podem gerar resultados produtivos para a colaboração, conectividade e sociabilidade.

Em se tratando de sistemas construídos pela Web, a Computação nas Nuvens, ou Cloud Computing, é um ponto de grande desafio, impulsionando os padrões que facilitam a comunicação e interoperabilidade em plataformas e aplicativos nas nuvens, bem como a criação de pacotes de desenvolvimento SDK que algumas linguagens de programação como Java e PHP possuem. Uma das arquiteturas exemplificada neste artigo mostra como a arquitetura Cloud é organizada para prover recursos necessários à criação de sistemas.

As arquiteturas Cloud Computing também podem constituir um modelo que compartilha recursos físicos, o que poderá ser um fator de insegurança para alguns sistemas, apesar de favorecer fortemente o crescimento da aplicação, em virtude da facilidade dos recursos, então aqui se tem um ponto de controvérsia.

No contexto de desenvolvimento de sistemas, tão importante quanto o aspecto tecnológico, é o processo de concepção arquitetural. É extremamente relevante conhecer o que se deseja construir, quais serviços são necessários, onde podem ser encontrados, como será feita a comunicação e como eles irão impactar no comportamento do software ou aplicação.

#### Você gostou deste artigo?

Dê seu voto em [www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)

Ajude-nos a manter a qualidade da revista!



#### Links e Referências:

##### Informações sobre a Web 3.0 reprogramável

<http://www.richappsconsulting.com/blog/blog-detail/web-30/>

##### Artigo sobre a Web 3.0

<http://www.zdnet.com/blog/saas/marc-benioff-heralds-web-3-0-at-dreamforce-europe/514>

##### Endereço com informações sobre o desenvolvimento de aplicativos para o Facebook

<http://developers.facebook.com/>

##### Endereço com informações sobre o desenvolvimento de aplicativos para o Twitter

<http://apiwiki.twitter.com/>

##### Blog de Silvio Meira. Especialista em inovação tecnológica

<http://terrasmagazine.terra.com.br/silviomeira/blog>

##### PANGEA: Comunidade de arquitetura de software

<http://pangeanet.org/>

# CURSOS ONLINE

A Revista SQL Magazine oferece aos seus assinantes uma série de Cursos Online de alto padrão de qualidade.



#### CONHEÇA OS CURSOS MAIS RECENTES:

- **Cursos: Curso de noSQL (Redis) com Java**
- **Desenvolvimento para SQL Server com .NET**
- **Curso PostgreSQL - Treinamento de banco de dados (Curso Básico)**

Para mais informações :

<http://www.devmedia.com.br/cursos/banco-de-dados>

(21) 3382-5038

# Somos tão apaixonados por tecnologia que o nome da empresa diz tudo.

**Porta 80** é o melhor que a Internet pode oferecer para sua empresa.

Já completamos 8 anos e estamos a caminho dos 80, junto com nossos clientes.

Adoramos tecnologia.  
Somos uma equipe composta de gente que entende e gosta do que faz,  
**assim como você.**



## Estrutura

100% NACIONAL.  
Servidores de primeira linha, links de alta capacidade.

## Suporte diferenciado

Treinamos nossa equipe para fazer mais e melhor. Muito além do esperado.

## Serviços

Oferecemos a tecnologia mais moderna, serviços diferenciados e antenados com as suas necessidades.

## 1-to-1

Conhecemos nossos clientes. Atendemos cada necessidade de forma única.  
Conheça!



# Porta 80

WEB HOSTING

Hospedagem | Cloud Computing | Dedicados | VoIP | Ecommerce |  
Aplicações | Streaming | Email corporativo

[porta80.com.br](http://porta80.com.br) | [comercial@porta80.com.br](mailto:comercial@porta80.com.br) | [twitter.com/porta80](http://twitter.com/porta80)

SP 4063-8616 | RJ 4063-5092 | MG 4063-8120 | DF 4063-7486