



Uma Abordagem de Apoio à Solução Técnica em Ambientes de Desenvolvimento de Software Orientados à Organização

Sávio Figueiredo, Ana Regina Rocha, Gleison Santos, Mariano Montoni

COPPE/UFRJ – Programa de Engenharia de Sistemas e Computação

Caixa Postal 68511 – CEP: 21945-970

Rio de Janeiro – RJ

{savio, darocha, gleison, mmontoni}@cos.ufrj.br

Abstract: *The size and complexity of software products have been continuously increasing over the years. Moreover, the number of software projects that fail to finish inside the constraints of budget and schedule established and to deliver a product that addresses the needs of the customer, also increases significantly. In this context, the research with focus at software processes has began because people have realized that the software product quality is extremely dependent of the software process that has been adopted in order to develop the product. This work presents an approach to support the Technical Solution process, which is executed during the software development.*

Resumo: *Os produtos de software estão se tornando cada vez maiores e mais complexos. Além disso, o número de projetos de software que não conseguem ser terminados dentro dos prazos e custos estabelecidos e que não implementam um produto que satisfaça às necessidades do cliente também cresce. Neste contexto, se situam as pesquisas com foco em processos de software, pois se percebeu que a qualidade do produto a ser desenvolvido está fortemente relacionada com a qualidade do processo utilizado para desenvolvê-lo. Este trabalho apresenta uma abordagem de apoio ao processo de Solução Técnica, que é executado durante o desenvolvimento de software.*

1. Introdução

A atenção atual no estudo de processos de software é fruto da percepção de que a qualidade do produto de software desenvolvido está fortemente relacionada com a qualidade do processo de software utilizado para o seu desenvolvimento [Fuggeta 2000, Gomes 2001, Golubié 2005]. Cada vez mais os desenvolvedores de software percebem que a sua habilidade em satisfazer os contratos está fortemente relacionada com os processos que utilizam para desenvolver o produto [Sheard 1997].

O processo de Solução Técnica, um dos processos executados durante o desenvolvimento de um produto, é iniciado quando os requisitos para o problema a ser resolvido pelo software estiverem definidos, desenvolvidos e aprovados. Este processo pode ser executado tanto no contexto do software a ser desenvolvido quanto no contexto do sistema onde o software será integrado. O objetivo do processo de Solução Técnica é elaborar o projeto (*design*) do software e, também, implementar a solução de projeto para os requisitos em questão.



O processo de Solução Técnica é intenso em conhecimento e durante a sua execução é necessário tomar diversas decisões, pois podem existir diversas maneiras de solucionar um mesmo problema. Dessa forma, o conhecimento sobre Solução Técnica é de grande importância para auxiliar nas escolhas a serem tomadas. É importante armazenar o conhecimento organizacional adquirido durante a execução do processo, mantendo o registro do raciocínio pelo qual uma determinada decisão foi tomada durante a etapa de projeto. Isso fica claro quando se tem um problema de rotatividade de pessoal na empresa. A alta rotatividade da equipe na indústria de software, associada com longos períodos de vida dos produtos, aumenta a probabilidade de que os projetistas originais não estejam presentes quando evoluções nos produtos forem necessárias e os problemas começarem a ocorrer [Burge e Brown 2002].

Outro problema comum nas organizações é a dificuldade de aproveitar o conhecimento de membros mais experientes durante o treinamento de novos membros das equipes, pois a dinâmica de trabalho não permite que os experientes parem a execução de suas atividades para compartilhar o seu conhecimento. Assim, em situações de tomada de decisão, os membros iniciantes tendem a repetir os mesmos erros cometidos por outros membros das equipes que já passaram por situações semelhantes [Montoni 2003]. Além disso, muito freqüentemente, o entendimento necessário para realizar manutenções depende da compreensão de quais decisões de projeto foram consideradas, que suposições foram feitas, que soluções alternativas foram rejeitadas e que critérios e requisitos foram satisfeitos no processo de deliberação [Conklin 1989].

Este tipo de documentação é chamado de *Design Rationale*. *Design Rationale*, ou simplesmente DR, é a explicação do porquê de um artefato ou alguma parte dele ter sido projetado do jeito que foi [Lee e Lai 1991]. Souza *et al.* [1998] definem DR como a documentação das decisões de projeto com suas respectivas justificativas, as opções consideradas, as avaliações e a argumentação que levaram a determinada decisão.

Um tipo mais específico de decisão que as organizações geralmente têm que tomar durante seus projetos é a decisão de desenvolver, comprar ou reutilizar um determinado componente do produto. Este tipo de decisão é muito importante para uma organização e deve levar em consideração, principalmente, dois fatores que são o custo de cada opção e o impacto nas competências centrais dessa organização que o componente possui. É vital para as organizações que os custos de desenvolvimento dos produtos sejam reduzidos, assim como seu prazo de chegada ao mercado (*time to market*). Um meio de conseguir estes dois objetivos é delegar à outras empresas a confecção de componentes do produto, desde que estes componentes não façam parte da competência central da organização.

É necessário, portanto, a existência de uma abordagem que permita a uma organização decidir o que lhe é mais vantajoso: desenvolver um determinado componente internamente, contratar uma outra empresa para fazer este desenvolvimento ou reutilizar um componente já disponível na organização. As organizações devem ser capazes de escolher as partes do produto que serão produzidas internamente e as partes dos produtos que serão produzidas por empresas contratadas.

Este artigo apresenta uma abordagem de apoio à Solução Técnica em projetos de software. São descritos os processos que fornecem suporte à seleção de alternativas de projeto e a análise entre desenvolver, comprar ou reutilizar um determinado componente do produto e as ferramentas TechSolution e MBR que apóiam,



respectivamente, a execução destes processos. As duas ferramentas estão integradas aos ambientes de Desenvolvimento e Manutenção TABA [Vilela 2004] para apoiar a execução de projetos de software de uma organização.

A seção 2 deste artigo discute a importância da utilização do DR quando o foco é a qualidade do software a ser desenvolvido. A seção 3 descreve alguns sistemas de DR existentes na literatura. Na seção 4 são apresentados os processos e as ferramentas definidos nesta abordagem para apoiar a Solução Técnica.

2. Design Rationale X Qualidade de Software

A abordagem proposta baseia-se na utilização do DR para apoiar a Solução Técnica em projetos de software. Sob o ponto de vista dos estudos relacionados com qualidade de software, existem muitos motivos para a utilização do DR, sendo inúmeras as suas formas de aplicação em benefício da qualidade de software.

Em relação à verificação, o DR pode ser utilizado para verificar se o projeto satisfaz os requisitos definidos e para verificar a intenção do projetista. Também é possível utilizar o DR com a finalidade de avaliar projetos e escolhas de projeto a fim de detectar inconsistências. Sob o foco da manutenção de software, DR é útil na localização de fontes de problemas no projeto com o propósito de indicar onde são necessárias modificações e de assegurar que as opções rejeitadas não serão inadvertidamente reimplementadas. A documentação do projeto também é beneficiada, pois se pode ter, além da “fotografia” das decisões finais, informações justificando as decisões tomadas [Burge e Brown 2002, Lee 1997].

Além disso, DR também traz benefícios para a qualidade do software a ser desenvolvido ao fornecer suporte para o projeto. Neste contexto, DR apóia as discussões, verifica o impacto das modificações, realiza uma checagem de consistência e ajuda na mitigação de conflitos ao observar as violações de restrições entre os múltiplos projetistas. Raciocínios representados explicitamente podem fornecer um vocabulário comum e manter a memória dos projetos, facilitando as negociações e o consenso [Burge e Brown 2002, Lee 1997].

Outros aspectos do desenvolvimento de software também são beneficiados pela utilização do DR. O ensino do projeto, por exemplo, é facilitado pois o DR pode ser utilizado para treinar novos membros da equipe. Em relação à comunicação, o DR pode ser usado para comunicar as razões pelas quais as decisões foram tomadas. Finalmente, sob o contexto da reutilização, DR é útil para determinar que porções do projeto podem ser reutilizadas e, em alguns casos, sugerir aonde e como serão modificadas para satisfazer um novo conjunto de requisitos [Burge e Brown 2002, Lee 1997].

Deste modo, pode-se dizer que a utilização do DR está fortemente relacionada à atividade de garantia da qualidade, cujo objetivo é monitorar o produto e os processos, para tentar assegurar que eles estejam em conformidade com os requisitos especificados e os planos estabelecidos. A utilização do DR auxilia a alcançar os objetivos mencionados, sendo, portanto um elemento importante na garantia da qualidade do software.



3. Trabalhos Correlatos

Os sistemas de DR oferecem suporte a captura, armazenamento e recuperação de DR, visando auxiliar o entendimento dos projetos e registrar as experiências obtidas em toda sua elaboração, servindo para projetos futuros [Francisco 2004].

Geralmente, os sistemas de DR propostos ou acarretam muito esforço do desenvolvedor na captura de DR e facilitam sua recuperação, ou acarretam pouco ou nenhum esforço ao desenvolvedor e dificultam sua recuperação. Também se observa que quanto maior a exigência na estrutura da representação do DR, melhor é a sua recuperação, mas menor é a efetividade na sua captura e vice-versa [Francisco 2004].

O DR pode ser representado através de três formas diferentes que são a argumentação, a comunicação e a documentação [Shipman e McCall 1997]. Argumentação e documentação focam nas decisões de projeto e nas razões por trás das mesmas. A diferença é que o objetivo da documentação é fornecer conhecimento sobre o projeto a pessoas externas ao projeto, enquanto que a argumentação tem o objetivo adicional de estruturar como o projetista abordou o problema. A perspectiva da comunicação é uma tentativa de capturar naturalmente a comunicação do projeto enquanto a mesma ocorre, como e-mails, minuta de reunião etc [Burge 2005]. A Tabela 1 resume as diferenças entre as três perspectivas [Francisco 2004].

Tabela 1 – Diferenças entre as perspectivas de representação do DR

Perspectivas	Argumentação	Comunicação	Documentação
Características das Informações			
Conteúdo	Semi-estruturado	Não Estruturado	Estruturado
Nível de detalhamento da informação armazenada em relação à produzida durante as discussões	Médio – analítica e sintetizada	Alto - Analítica	Baixo – Sintetizada
Momento da Captura	Durante a discussão	Durante a discussão	Após a discussão
Esforço/atividade requerido(a) no momento da discussão	Classificar os nós e links	Não possui	Não possui
Esquemas de representação	Nós e links	Diversas mídias	Linear

O sistema gIBIS [Conklin e Burgess-Yakemovic 1995] utiliza a notação de argumentação IBIS para representar a atividade de argumentação dos desenvolvedores. O sistema gIBIS almeja facilitar, capturar, estruturar e gravar a discussão entre esses desenvolvedores, construindo uma rede de forma a representar tal discussão. A notação IBIS também é utilizada por itIBIS [Conklin e Burgess-Yakemovic 1995].

SIBYL é um sistema de apoio à decisão em grupo, também não dedicado a um domínio específico, que conta com representação e gerenciamento de aspectos qualitativos do processo de decisão, tais como os objetivos, as alternativas e a avaliação dessas alternativas de acordo com os objetivos. O SYBIL é uma extensão do sistema de hipertexto simples gIBIS. Para a criação e gerência da rede de argumentação, o sistema SYBIL adota o modelo DRL [Lee 1990].



JANUS é um sistema para o ambiente de projeto de cozinhas planejadas que utiliza a notação PHI [McCall 1991]. A notação PHI também é utilizada por PHIDIAS, um sistema hipermídia usado para projetos gráficos 2D e 3D. Uma representação simplificada da notação PHI é utilizada por DocRationale [Francisco 2004].

InfoRat utiliza um subconjunto significativo da notação DRL. Este sistema pode ser utilizado de forma integrada à plataforma *eclipse*¹ e fornece uma ontologia de critérios a serem utilizados durante a avaliação e seleção das alternativas. Através de InfoRat é possível realizar inferências sobre o DR para verificar a sua estrutura sintática e semântica [Burge 2005]. SYBIL também verifica o DR ao verificar se o raciocínio que levou a uma determinada decisão está completo. Outros sistemas, como JANUS, criticam o projeto e fornecem aos projetistas um DR para apoiar as críticas.

CoMo-Kit [Dellen *et al.* 1996] utiliza um modelo de processo de software para obter as decisões de projeto e as dependências de causa entre as mesmas. Estas dependências de causa são, então, consideradas como sendo o DR. O sistema observa tarefas (objetivos a serem alcançados durante o processo), produtos (os produtos produzidos, como especificações), métodos (o modo como a tarefa é realizada) e agentes (a pessoa ou computador que executa a tarefa). Quando o modelo de processo é gerado, o fluxo da informação entre as tarefas é capturado e usado para deduzir a dependência entre as mesmas. O raciocínio consiste da justificativa pela escolha de um determinado método. CoMo-Kit [Dellen *et al.* 1996] não é um sistema de argumentação – não registra as alternativas tentadas e rejeitadas e as razões por trás das mesmas. Por outro lado, usa as dependências para encontrar a justificativa para cada decisão.

ABCDE-DR [Souza *et al.* 1998] é um editor de diagramas de objetos que implementa um modelo cooperativo para o registro do DR. O sistema apóia: (i) a representação semi-formal do DR, (ii) o registro do DR de uma maneira semi-automática através da reflexão na teoria da ação, e (iii) a integração entre o artefato e o DR. O modelo de DR foi desenvolvido baseado no modelo de cooperação de SOUZA *et al.* [1997], que utiliza as anotações como forma de cooperação entre os usuários. O modelo de DR também usa as anotações como forma de representar a informação de DR. Este modelo é chamado semi-automático porque coleta a informação do DR por fora da interação normal dos autores e revisores durante o processo de melhoria do projeto deles, ou seja, o DR é obtido quase que automaticamente a partir da interação entre os projetistas.

WinWin [Boehm e Bose 1994] é uma abordagem que objetiva a coordenação das atividades de tomada de decisão feitas pelos vários *stakeholders* envolvidos no processo de desenvolvimento de software. Bose [1995] definiu uma ontologia para o raciocínio de decisão com a finalidade de manter a estrutura de decisão. O objetivo era modelar o raciocínio da decisão para apoiar a manutenção ao permitir que o sistema determine o impacto de uma mudança e propague os efeitos da modificação.

4. A Abordagem de Apoio à Solução Técnica

Com o objetivo de apoiar a Solução Técnica em projetos de software, foram definidos dois processos: um processo que apóia a seleção de alternativas de projeto, mantendo

¹ *Eclipse*: é uma comunidade de código aberto cujos projetos focam no fornecimento de plataformas de desenvolvimento e *framework* de aplicações para o desenvolvimento de software.



assim o registro das decisões tomadas e o porquê de cada decisão; e um processo que apóia um tipo mais específico de decisão. que é a decisão entre desenvolver, comprar ou reutilizar um determinado componente do produto.

Em relação à representação do DR, pode-se dizer que a abordagem implementada utiliza a argumentação, tipo de representação explicada na seção 3 deste artigo, e dispõe dos seguintes nós:

- *Avaliação*: Problema para o qual se busca a melhor solução dentre uma lista de possíveis soluções. Uma avaliação possui objetivos, uma lista de possíveis soluções alternativas e critérios que serão utilizados para que seja possível selecionar a solução mais adequada.
- *Objetivo*: Objetivos da avaliação. Este nó poderá conter informações sobre possíveis restrições, como por exemplo, restrições de tempo ou esforço da avaliação em questão. Os objetivos também podem indicar os requisitos do projeto que devem ser atendidos pela solução a ser selecionada.
- *Critério*: As soluções alternativas identificadas serão avaliadas em relação a uma lista de critérios. Cada critério possui um peso que indicará a importância do critério em questão para o projeto. Um critério pode conter uma lista de sub-critérios quando houver necessidade de refinamento. Neste caso, apenas os sub-critérios serão avaliados.
- *Solução Alternativa*: Soluções para o problema sob avaliação. Estas soluções serão avaliadas em relação aos critérios a fim de selecionar a solução mais indicada.
- *Avaliação do Critério*: Os critérios da avaliação serão avaliados em relação às soluções alternativas. Esta avaliação compreende a atribuição de um valor que indica o quanto a solução alternativa satisfaz o critério em questão. Além disso, deve-se justificar o porquê da atribuição de um valor a um determinado critério durante a avaliação de uma solução alternativa.

Este esquema de representação foi construído com base nos outros modelos de representação de DR já apresentados neste artigo. Comparando-o com o modelo DRL [Lee e Lai 1991], por exemplo, pode-se dizer que o nó “Avaliação” armazenaria as mesmas informações que o nó “Problema de Decisão”, “Solução Alternativa” representaria o nó “Alternativa”, e “Avaliação dos Critérios” teria como nó correspondente no outro modelo o nó “Alegação”. O objetivo é manter a documentação das decisões tomadas, das soluções alternativas avaliadas, dos critérios utilizados durante a avaliação e das avaliações propriamente ditas destas alternativas em relação aos critérios.

Após a definição dos processos, duas ferramentas foram implementadas para fornecer suporte à execução dos mesmos: a ferramenta TechSolution que implementa as atividades do processo de seleção de alternativas de projeto e a ferramenta MBR que implementa as atividades do processo de análise entre fazer, comprar ou reutilizar um determinado componente. Estas ferramentas encontram-se integradas à Estação TABA, podendo ser executadas a partir de um ADSOrg [Villela 2004] instanciado para um projeto específico.



4.1 Processo de Seleção de Alternativas de Projeto

O processo para apoiar a seleção de alternativas de projeto, mantendo desta forma o raciocínio por trás das decisões tomadas, foi definido baseado principalmente nas áreas de processo Solução Técnica e Análise de Decisão e Resolução do CMMI (*Capability Maturity Model Integration*) [Chrissis *et al.* 2003], nos processos de Solução Técnica e de Análise de Decisão e Resolução do MPS.BR [MPS.BR 2005] e na norma ISO/IEC 14102 [ISO/IEC 14102 1995]. Além disso, outros sistemas de DR encontrados na literatura e apresentados na seção 3 deste artigo também influenciaram na elaboração do processo.

A norma ISO/IEC 14102 descreve um conjunto de processos que possuem como objetivo avaliar ferramentas CASE e selecionar a mais apropriada. As descrições das atividades destes processos da norma influenciaram a definição das atividades do processo proposto. Em relação à área de processo Análise de Decisão e Resolução do CMMI, esta área apresenta a descrição de práticas a serem executadas por um processo de avaliação de alternativas e seleção da mais apropriada. No MPS.BR, são descritos os resultados esperados da execução do processo de Análise de Decisão e Resolução. O objetivo do processo de Análise de Decisão e Resolução é analisar possíveis decisões através de um processo formal avaliando alternativas em relação a critérios pré-estabelecidos de modo a identificar a alternativa mais adequada.

O processo de seleção de alternativas de projeto definido é composto pelas seguintes atividades:

- **Definir Objetivos e Restrições da Seleção:** Definir os objetivos e as restrições relativas à avaliação e seleção que serão realizadas nas etapas seguintes. Devem ser identificadas, por exemplo, restrições de cronograma, recursos, e custos. Durante esta atividade o projetista pode buscar por uma avaliação que já tenha ocorrido e que o ajude durante esta avaliação ou então consultar se a ferramenta possui conhecimento sobre a avaliação a ser executada. Por exemplo, se o projetista precisa tomar uma decisão de projeto que envolve decidir se as chaves primárias das tabelas do banco de dados serão geradas pelo banco de dados ou se serão geradas pelo próprio software sendo desenvolvido, e caso esta decisão já tenha sido tomada em um projeto anterior, o projetista poderá consultar o DR do outro projeto para tomar a decisão mais apropriada. Caso o projetista precise decidir que estilo arquitetural utilizar, então o projetista poderá utilizar o conhecimento fornecido pela ferramenta para realizar identificar as possíveis soluções e avaliá-las a fim de selecionar uma solução. Os exemplos citados acima são úteis para projetos de desenvolvimento e também para projetos de manutenção. Mas é nesta atividade também que o projetista pode procurar por um determinado DR para saber o porquê de uma decisão tomada. Esta situação é especialmente útil durante a manutenção, para facilitar o entendimento do sistema, evitar a repetição de erros etc.

- **Estabelecer Critérios de Seleção:** Baseado nas expectativas e objetivos desenvolvidos na atividade anterior é necessário estabelecer critérios de seleção. Estes critérios de seleção serão utilizados na avaliação e seleção do conjunto de soluções alternativas a serem consideradas. Os critérios devem ser ponderados de modo que o critério de maior peso exerça a maior influência, ou seja, deve-se atribuir um valor a estes critérios que indique a importância em se satisfazer o critério para o projeto em questão. Também é necessário definir as faixas e escalas associadas a cada critério para que se possa posteriormente avaliar os critérios de seleção em relação às soluções



alternativas ao selecionar um valor na faixa ou escala estabelecida. Para problemas que o projetista não tenha encontrado um projeto que possui DR explicando porque uma solução foi selecionada, ou então se a abordagem não possui conhecimento de como realizar a avaliação pode-se visualizar uma lista de critérios que são sugeridos pela ferramenta, a partir dos quais o usuário acrescentará novos, refinará os existentes ou removerá de acordo com a situação. Assim, ao executar esta atividade, as pessoas que estão executando a manutenção do software poderão tomar conhecimento dos critérios utilizados nas avaliações das soluções.

- **Desenvolver Soluções Alternativas:** Nesta atividade, o projetista desenvolverá soluções alternativas detalhadas para o problema em questão. Uma ampla quantidade de alternativas aparece ao solicitar ao maior número de stakeholders possível por soluções para a questão. Sessões de Brainstorming também podem estimular alternativas inovadoras através de uma rápida interação e feedback [Chrissis *et al.* 2003]. As soluções alternativas identificadas devem ser documentadas. Neste momento, os requisitos devem ser mapeados com as soluções alternativas desenvolvidas. Deste modo, será possível rastrear os requisitos em relação à solução selecionada no futuro. A execução desta atividade é muito importante para a manutenção, pois ao analisar um projeto, o projetista terá conhecimento das soluções alternativas analisadas no momento em que o projeto foi elaborado, e também do porquê de uma solução ter sido escolhida em detrimento de outras. É importante ressaltar que no caso da abordagem já possuir conhecimento sobre a decisão sendo tomada, ou no caso desta decisão já ter sido tomada anteriormente durante outro projeto, então as soluções alternativas serão sugeridas para o projetista.

- **Evoluir Cenários e Conceitos Operacionais:** Durante esta atividade, os conceitos operacionais, cenários e ambientes para descrever as condições, modos de operação e estados de operação específicos de cada componente do produto são evoluídos. Os conceitos operacionais e cenários são evoluídos para facilitar a seleção de soluções para os componentes de produto. Conceitos operacionais e cenários documentam a interação dos componentes do produto com o ambiente, usuários e outros componentes do produto, sem levar em conta a disciplina de engenharia. Esta atividade é necessária para fornecer ao projetista maiores informações para que o mesmo possa avaliar adequadamente as soluções alternativas e por consequência, selecionar a solução alternativa que melhor satisfaz os critérios pré-estabelecidos.

- **Realizar Avaliação:** O propósito da avaliação é produzir um relatório técnico que será utilizado como entrada durante a atividade de seleção. Cada solução alternativa deve ser avaliada em relação aos critérios de seleção pré-estabelecidos. Toda a atividade de avaliação deve ser documentada e esta documentação deve conter, entre outras coisas, a justificativa pela qual uma determinada solução alternativa recebeu uma pontuação em um critério de seleção. No fim da avaliação será gerado um relatório sobre a avaliação que além de descrever o processo de avaliação executado, também deve apresentar os critérios de seleção identificados, as soluções alternativas com os requisitos mapeados e a avaliação das soluções alternativas em relação aos critérios pré-estabelecidos. O relatório também apresenta as atividades do processo de avaliação em um nível de detalhe necessário para permitir ao leitor tanto entender o escopo e profundidade da avaliação, quanto repetir a avaliação caso deseje. Novamente, nas situações em que o problema para o qual está sendo procurada uma solução já foi solucionado anteriormente ou caso a abordagem possua conhecimento sobre o mesmo,



então as avaliações dos critérios sugeridos para o projetista em relação às alternativas também sugeridas será feita automaticamente. O projetista poderá alterar o resultado de uma avaliação de um critério caso não concorde.

- **Selecionar Solução:** Na última atividade do processo, aplica-se um algoritmo de seleção aos resultados dos esforços de avaliação das soluções alternativas. Esta atividade tem início assim que o relatório de avaliação for completado. Com a aplicação do algoritmo de seleção, uma solução alternativa é recomendada. Os resultados da avaliação podem ser avaliados e então indicar a necessidade de informações adicionais, o que implica em re-executar algumas das atividades anteriores deste processo.

Para a representação gráfica do processo, foi utilizada a linguagem de modelagem de processos desenvolvida por VILLELA [2004]. A Figura 1 exibe o processo de seleção de alternativas de projeto definido nesta abordagem.

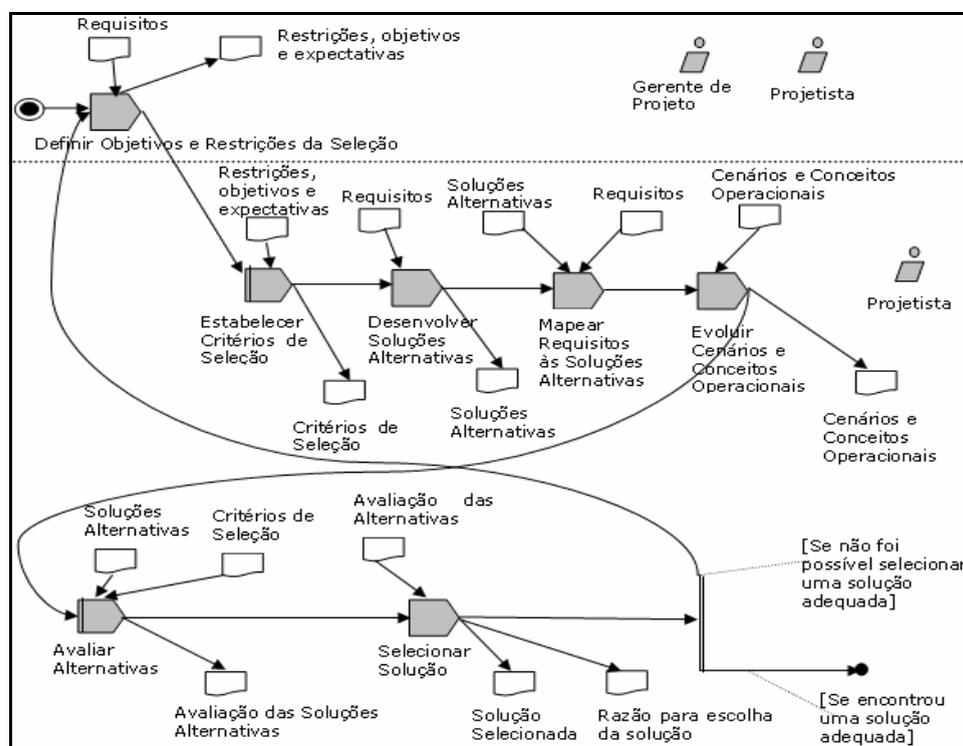


Figura 1 – Processo de Seleção de Alternativas de Projeto

4.2 Processo de Análise entre Desenvolver, Comprar ou Reutilizar

Em virtude da atual globalização, da forte competitividade e do crescimento da complexidade dos produtos de software, as empresas buscam melhores formas de se organizarem. As organizações tentam centralizar-se em suas atividades principais, a fim de melhorar a gerência dos custos. Isto levanta questões como: Que recursos deveriam ser desenvolvidos para aumentar as competências da organização? Que atividades deveriam ser feitas por outras empresas e a que parceiros em potencial deveriam ser contratados para executar estas atividades? Que atividades internas deveriam ser preservadas e desenvolvidas pela própria organização? Como os recursos da organização deveriam ser alocados em relação às atividades [Bouchriha *et al.* 2002]?



As abordagens existentes na literatura para responder a estas questões possuem duas correntes principais de pensamento: a primeira busca responder a pergunta sob um ponto de vista econômico e a segunda foca no ponto de vista estratégico [Cánez *et al* 2001]. Cánez *et al.* [2001] descrevem as principais abordagens criadas na década passada para apoiar a decisão de fazer ou comprar.

Além disso, assim como as tecnologias evoluem, o raciocínio para fazer ou comprar um componente do produto também evolui. Por um lado, esforços de desenvolvimento complexos favorecem a compra de um componente de produto de prateleira, mas por outro, avanços na produtividade e ferramentas favorecem a opção de desenvolvimento. Durante esta escolha, deve-se levar em consideração que produtos de prateleira podem ter documentação incompleta ou inexata e as organizações que os fornecem podem não dar suporte para os mesmos no futuro [Chrissis *et al.* 2003].

Chrissis *et al.* [2003] sugerem fatores que podem ser considerados durante o processo de escolha entre fazer, comprar ou reutilizar: (i) funções que os produtos ou serviços irão fornecer e como essas funções se encaixarão no projeto, (ii) recursos e habilidades disponíveis no projeto, (iii) datas de entrega e integração críticas, (iv) uniões de negócio estratégicas, incluindo requisitos de negócio de alto nível, (v) pesquisa do mercado sobre produtos disponíveis, incluindo produtos COTS, (vi) funcionalidade e qualidade dos produtos disponíveis, (vii) habilidades e capacidade dos potenciais fornecedores, (viii) impacto do produto ou serviço nas competências centrais da organização, (ix) licenças, garantias, e limitações associadas com os produtos sendo adquiridos, (x) disponibilidade do produto, (xi) questões proprietárias e (xii) redução de riscos.

A decisão entre fazer, comprar ou reutilizar um determinado componente do produto também é uma escolha de alternativa de projeto, entretanto, pode ser melhor apoiada através da utilização de um processo com atividades específicas para este tipo de decisão. Neste sentido, foi definido um processo, com base no processo de seleção de alternativas de projeto definido, para apoiar a escolha entre fazer, comprar ou reutilizar um componente do produto. Este processo é composto pelas seguintes atividades:

- **Caracterizar o Produto/Componente:** Durante a execução desta atividade, deve-se identificar os requisitos que o produto/componente necessita satisfazer. Estes requisitos serão fundamentais durante todas as atividades do processo de seleção de uma alternativa de compra, reutilização ou desenvolvimento. Tais requisitos foram desenvolvidos no processo de desenvolvimento de requisitos, mas podem não estar agrupados de forma a indicar que serão atendidos por um produto/componente.

- **Estabelecer Critérios de Seleção:** Esta atividade é análoga à atividade “Estabelecer Critérios de Seleção” do processo de seleção de alternativas de projeto. A diferença está no fato de que neste momento, os critérios são definidos tendo como base os requisitos identificados na atividade anterior para o produto/componente. Outra diferença está na lista de critérios sugeridos, pois os critérios sugeridos são específicos para a análise entre desenvolver, comprar ou reutilizar um determinado componente do produto. Assim como no processo de seleção de alternativas de projeto, os critérios também devem ser ponderados e uma faixa ou escala deve ser atribuída aos mesmos.

- **Identificar Fornecedores e Produtos/Componentes Potenciais:** Esta atividade tem por objetivo identificar potenciais fornecedores do produto/ componente em questão e localizar produtos/componentes capazes de satisfazer aos requisitos



desenvolvidos. Devem ser identificados os fornecedores que poderiam entregar um produto\componente que atenda aos requisitos levantados, assim como os produtos\componentes já existentes que preencham a estes requisitos. Os produtos\componentes existentes na organização e que poderiam ser reutilizados devem ser levados em consideração. Sendo assim, as possíveis soluções alternativas do processo de análise entre fazer, comprar ou reutilizar são os produtos\componentes que poderiam ser reutilizados, os produtos\componentes existentes no mercado que poderiam ser adquiridos e os fornecedores que poderiam desenvolver um produto\componente satisfazendo os requisitos definidos.

- **Coletar Dados dos Produtos\Componentes e\ou Fornecedores:** Nesta atividade o projetista deve coletar os dados dos produtos\componentes e fornecedores identificados que serão necessários para a execução da atividade de avaliação. Em relação aos produtos\componentes, o projetista deve coletar informações como, por exemplo: a versão a ser utilizada, tipos e preços das licenças existentes, *site* onde é possível localizá-lo, funções do produto\componente, público alvo do mesmo e o ambiente de software onde é possível utilizá-lo.

- **Realizar Avaliação:** Assim como no processo de seleção de alternativas de projeto, o objetivo da atividade “Realizar Avaliação” é produzir um relatório técnico que será utilizado como entrada para a atividade de tomada de decisão. As alternativas de desenvolvimento, compra ou reutilização devem ser avaliadas em relação aos critérios de seleção. Toda a atividade de avaliação deve ser documentada e esta documentação deve conter, entre outras coisas, a razão pela qual uma determinada solução alternativa recebeu uma pontuação em relação a um critério de seleção.

- **Tomar Decisão:** Neste momento, deve-se decidir entre desenvolver, comprar ou reutilizar o produto\componente a partir do resultado da avaliação realizada das alternativas mencionadas em relação aos critérios pré-estabelecidos. Deve-se aplicar um algoritmo de seleção no resultado da avaliação para obter uma solução recomendada. Caso não consiga chegar a uma conclusão, o projetista pode re-executar alguma das atividades do processo com a finalidade de refinar as informações.

A Figura 2 exhibe o processo de análise entre desenvolver, comprar o reutilizar descrito.

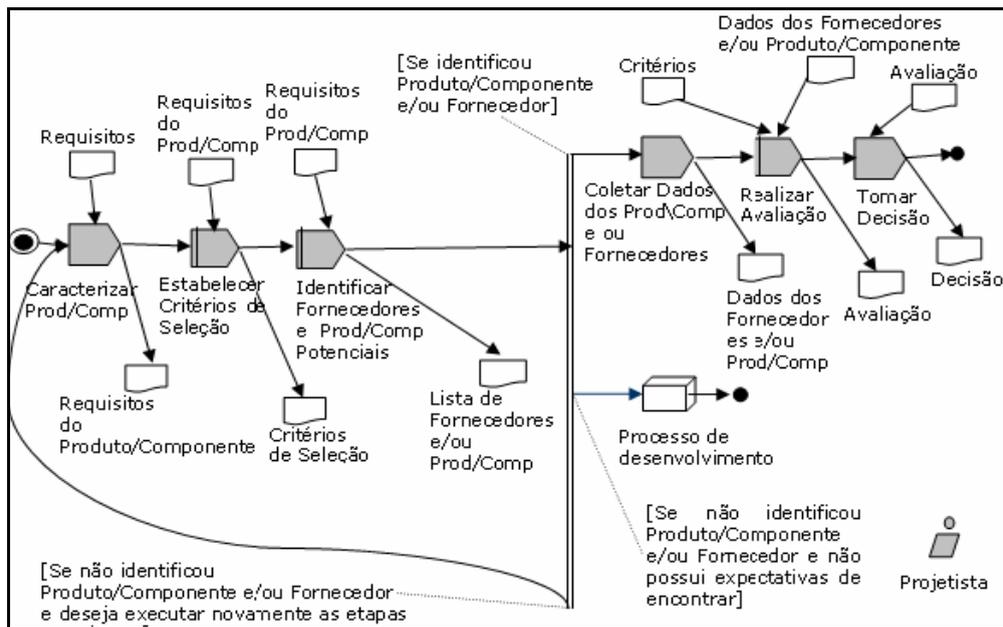


Figura 2 – Processo de Análise Fazer/Comprar/Reutilizar

4.3 As Ferramentas TechSolution e MBR

Para tornar viável a utilização da abordagem, era imprescindível fornecer apoio ferramental aos projetistas. Por outro lado, devido à necessidade de interferir o mínimo possível nas tarefas dos projetistas, decidiu-se por manter as ferramentas integradas a um ambiente que já estivesse sendo utilizado pelo projetista. Neste contexto, foram implementadas as ferramentas TechSolution [Figueiredo *et al.* 2005] e MBR que apóiam os processos de seleção de alternativas de projeto e de análise entre desenvolver, comprar ou reutilizar respectivamente, encontrando-se disponíveis nos ADS e AMS instanciados pela Estação Taba.

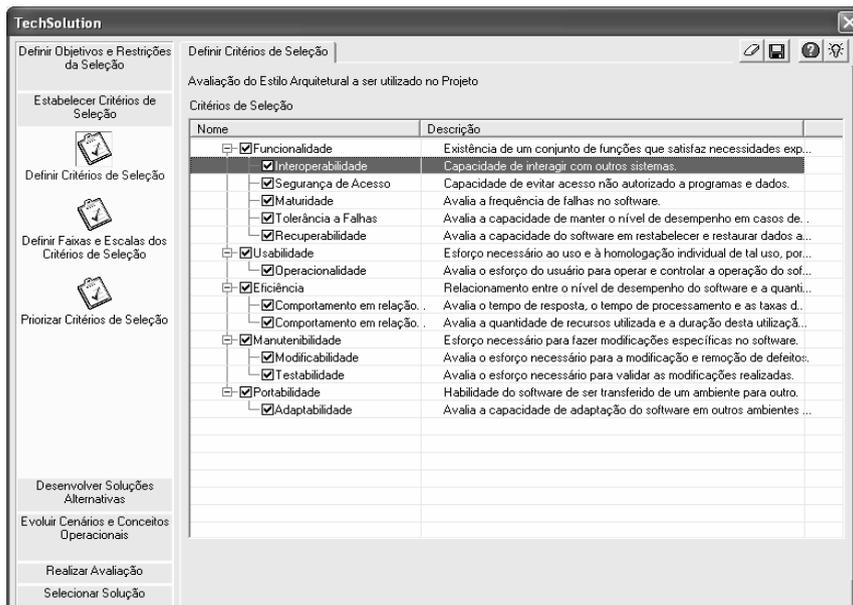


Figura 3 – TechSolution: Definir Critérios de Seleção



A Figura 3 exibe a interface principal da ferramenta TechSolution. Na parte esquerda da Figura 3, o sistema apresenta as atividades que guiam a execução da ferramenta, seguindo o padrão de interfaces de ferramentas da Estação Taba (estas atividades são descritas na seção 4.1). No lado direito da Figura, o sistema apresenta a atividade que está sendo executada no momento pela ferramenta, neste caso, a identificação dos critérios a serem utilizados para avaliação das soluções alternativas.

Neste exemplo, está sendo executada uma avaliação para determinar que estilo arquitetural deverá ser utilizado. Todo o raciocínio que levará a escolha de um determinado estilo arquitetural será registrado na ferramenta e poderá ser consultado posteriormente nos projetos de manutenção ou durante a execução do próprio projeto. Este conhecimento foi extraído da pesquisa realizada por Xavier [2001].

A Figura 4 apresenta a interface principal da ferramenta MBR. O exemplo apresentado na Figura 4 ilustra a situação na qual dois componentes estão sendo avaliados em relação aos critérios de seleção, que incluem os requisitos do projeto, a fim de determinar qual opção de compra de componente é mais vantajosa. Neste exemplo, a opção de desenvolvimento interno não está sendo avaliada pois a organização não possuía tempo e recursos para possibilitar o desenvolvimento interno do componente.

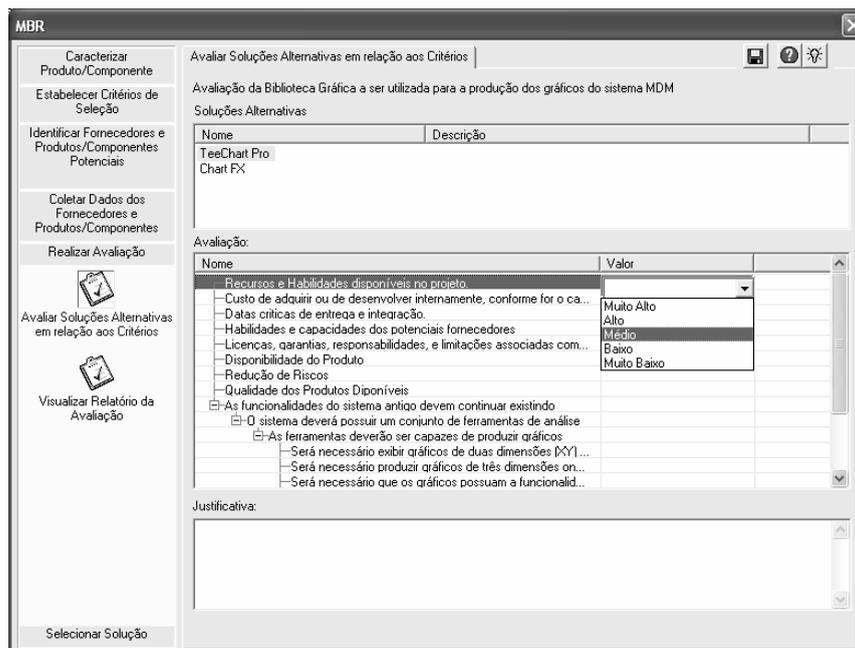


Figura 4 – MBR: Avaliar das Soluções Alternativas em relação aos Critérios

Um aspecto importante das ferramentas é o fornecimento de conhecimento ao projetista durante as atividades do processo da mesma. Este conhecimento facilita a utilização da ferramenta e por consequência, aumenta a probabilidade de que os DR serão registrados, visto que os projetistas possuem facilidades para registrar o DR. Como a captura do DR não é feita de forma automática, é necessário diminuir ao máximo as dificuldades do usuário, ou o DR não será registrado e portanto, não estará disponível para consulta posteriormente.

Algumas atividades do processo de Solução Técnica, como o projeto da solução selecionada e a implementação desta solução, não são executadas através ferramentas,



devendo ser supridas pelo processo de desenvolvimento. Estas atividades são apoiadas na Estação TABA através de roteiros específicos associados às atividades do processo de desenvolvimento e/ou manutenção previstas para gerá-los.

5. Conclusão

Este artigo apresentou uma abordagem de apoio à Solução Técnica através da utilização de *Design Rationale*. Como parte desta abordagem, foram descritos os processos de seleção de alternativas de projeto e de análise entre desenvolver, comprar ou reutilizar e as ferramentas TechSolution e MBR, que se encontram integradas aos ambientes instanciados TABA, foram apresentadas brevemente.

Um dos objetivos desta abordagem é beneficiar a qualidade do software a ser desenvolvido através de diversas formas discutidas ao longo deste texto como a não repetição de erros cometidos anteriormente na fase de projeto, o apoio à verificação do projeto, o apoio à gerência de conhecimento, o apoio à documentação e o apoio à manutenção.

Neste momento três empresas estão utilizando a abordagem e as ferramentas propostas neste artigo com o intuito de apoiar a implementação do processo de Solução Técnica em seus projetos. Estas empresas visam a obtenção do Nível de Maturidade C do MPS.BR e do Nível 3 do CMMI. Esta utilização torna possível uma avaliação, a ser realizada em breve, das ferramentas e dos processos definidos para obter um *feedback* sobre estes e permitir a sua melhoria através das questões levantadas pelas empresas.

Agradecimentos

Agradecemos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro ao projeto Ambientes de Desenvolvimento de Software Orientados à Organização.

Referências Bibliográficas

- Boehm B. e Bose, P. (1994), A Collaborative Spiral Software Process Model Based on Theory W, Third International Conference on the Software Process, Reston, VA, pp. 59-68.
- Bose, P. (1995), A Model for Decision Maintenance in the WinWin Collaboration Framework, Knowledge Based Software Engineering (KBSE '95), pp. 105-113.
- Bouchriha, H., D'Amours, S. e Ladet, P. (2002), A "make or buy" decision model with economies of scale, Article de conférence avec actes.
- Burge, J. E. (2005), "Software Engineering Using design RATIONALE", PhD Dissertation, CS Dept., WPI, May.
- Burge, J. E. e Brown, D. C. (2002), "Discovering a Research Agenda for Using Design Rationale in Software Maintenance", Computer Science Technical Report, Worcester Polytechnic University, WPI-CS-TR-02-03.
- Cánez, L., Robert, D. e Platts, K. (2001), Testing a Make-or-Buy Process, *Proceedings of the Twelfth Annual Conference of the Production and Operations Management Society*, POM-2001, March 30 – April 2, Orlando.
- Chrissis, M. B., Konrad, M. e Shrum, S. (2003), *CMMI: Guidelines for Process Integration and Product Improvement*, Addison-Wesley.
- Conklin, J., (1989), "Design Rationale and Maintainability". Vol. II: Software Track, *Proceedings of the Twenty-Second Annual Hawaii International Conference on Volume 2*, 3-6 Jan. Page(s): 533-539. Vol. 2 Digital Object Identifier 10.11 09/HICSS. 1989.48049.



- Conklin, J. e Burgess-Yakemovic, K., (1995), A Process-Oriented Approach to Design Rationale, In Design Rationale Concepts, Techniques, and Use, T. Moran and J. Carrol, (eds), Lawrence Erlbaum Associates, Mahwah, NJ, pp. 293-428.
- Dellen, B., Kohler, K. e Maurer, F. (1996), Integrating Software Process Models and Design Rationales, In: *Proceedings Knowledge-based Software Engineering*, Syracuse, NY, IEEE Computer Society Press, pp. 84-93.
- Figueiredo, S., Rocha, A.R., Santos, G., Montoni, M. e Natali, A.C., (2005), “Apoio à Manutenção de Software através de Design Rationale em Ambientes de Manutenção de Software Tabá”, *II Workshop de Manutenção de Software Moderna (WMSWM-05)*, Manaus, Brasil.
- Francisco, S. D. (2004), *DocRationale - uma ferramenta para suporte a Design Rationale de artefatos de Software*. São Carlos-SP, março de 2004. 123p. Dissertação de Mestrado. Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo.
- Fuggeta, A. (2000), Software Process: A Roadmap. In: *22nd International Conference on the Future of Software Engineering*, Limerick, Ireland: ACM-Association for Computing Machinery, p. 25-34.
- Golubié, S. (2005), Influence of Software Development Process Capability on Product Quality, In: *8th International Conference on Telecommunications*, Zagreb, Croatia, June 5-17.
- Gomes, A. (2001), Avaliação de processos de software baseada em medições, Tese de MSc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Jul.
- ISO/IEC 14102 (1995), Information technology - Guideline for the evaluation and selection of CASE tools.
- Lee, J. (1990), SIBYL: a tool for managing design rationale. In: *Proceedings of the Conference on Computer Supported Cooperative Work*, Los Angeles, EUA, p. 79-92.
- Lee, J. (1997), Design Rationale Systems: Understanding the issues. *IEEE expert/Intelligent Systems and Their Applications*, v. 12, n. 3, p. 78-85.
- Lee, J. e Lai, K. (1991), What's in design rationale. *Human-Computer Interaction*, 6(3-4): 251-280.
- McCall, R. (1991), PHI: A Conceptual Foundation for Design Hypermedia. *Design Studies* 12, pp.30-41.
- Montoni, M., (2003), *Aquisição de Conhecimento no Desenvolvimento de Software*, Tese de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- MPS.BR (2005), MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral (v. 1.0).
- Sheard, S, A. (1997), The Frameworks Quagmire, A Brief Look, Technical Report, Software Productivity Consortium.
- Shipman, F. e McCall, R. (1997), Integrating Different Perspectives on Design Rationale: Supporting the Emergence of Design Rationale from Design Communication. *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing*, 11(2):141-154.
- Souza, C, R. B., Wainer, J. e Rubira, C. M., R. (1997), An annotation model for the cooperative software development. In *III Workshop on Hypermedia and Multimedia Applications*, pages 143-154, São Carlos, Brazil, 23-25 May 1997.
- Souza, C, R. B., Wainer, J., Santos, D. B. e Dias, K. L. (1998), A model and tool for semi-automatic recording of design rationale in software diagrams. In: *Proceedings of the 6th String Processing and Information Retrieval Symposium & 5th International Workshop on Groupware*, Cancun, Mexico, 1998, p. 306-313.
- Villela, K. V. C. (2004), Definição e Construção de Ambientes de Desenvolvimento de Software Orientados à Organização, Tese de D. Sc., COPPE/UFRJ, Rio de Janeiro, Brasil.
- Xavier, J, R. (2001), Criação e Instanciação de Arquiteturas de Software Específicas de Domínio no Contexto de uma Infra-Estrutura de Reutilização, Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, Brasil.