

# Desenvolvimento de um Ambiente de Apoio à Tomada de Decisões em Projetos de Software

Hélio R. Costa<sup>1,2</sup>, Gustavo B. Vieira<sup>1,2</sup>, Luiz Claudius Coelho F. Leite<sup>1,2</sup>, Ana Regina Rocha<sup>1</sup>

<sup>1</sup>COPPE/UFRJ – Programa de Engenharia de Sistemas e Computação  
Caixa Postal 68511 – CEP: 21945-970 - Rio de Janeiro – RJ

<sup>2</sup>CCA-RJ – Ponta do Galeão s/nº - Ilha do Governador – RJ, CEP 21941-520

heliorcosta@terra.com.br, gustavo\_barbi@yahoo.com.br,  
luiz.claudius@gmail.com, darocha@centroin.com.br

**Abstract.** *Almost everything we do in our lives is about making decisions under an uncertain scenario. Software Engineers and Project Managers also experience this problem during a software development process. Nevertheless, up to the present moment, there not exist a body of knowledge that may help managers and practitioners during a decision process. This paper describes the steps that are being taken in order to build an environment for decision making in software projects based on a formal process, the reuse of knowledge and decision support methods.*

**Resumo.** *Quase tudo o que fazemos na vida envolve a tomada de decisão sobre um cenário de incerteza. Engenheiros de software e gerentes de projetos também vivenciam este problema ao longo do processo de desenvolvimento de software. Contudo, não existe, até o presente momento, um corpo de conhecimento organizado que possa auxiliar gerentes e engenheiros de software durante um processo decisório. Este artigo descreve os passos que estão sendo realizados para a construção de um ambiente de apoio à tomada de decisões em projetos de software, baseado em um processo formal, no reuso de conhecimento e na aplicação de métodos de apoio à decisão.*

## 1. Introdução

Tomar decisões é uma atividade central na vida de todas as pessoas. Quase tudo o que fazemos envolve algum tipo de tomada de decisão, seja ela simples ou complexa. A Engenharia de Software, assim como diversas áreas de conhecimento, também requer o uso de técnicas gerenciais, pois decisões precisam ser tomadas ao longo de todo o processo de desenvolvimento e evolução dos sistemas. Questões como: tipos de tecnologias, processos, recursos e ferramentas são fundamentais para a garantia da qualidade de produtos e serviços. Ruhe [Ruhe 2003] comenta que a tomada de decisões afeta significativamente todos os estágios do ciclo-de-vida de um projeto e que processos e sistemas de apoio à decisão são fundamentais para aumentar a eficiência, a qualidade e a relação custo-benefício de sistemas.

Normalmente, a experiência obtida durante o desenvolvimento de software é armazenada de forma implícita e dificilmente está acessível. Este fato acontece devido à falta de uma infra-estrutura que permita que o conhecimento adquirido não seja perdido e que forneça meios dele circular dentro da organização [Hoffmann 1997].

É certo que, com a atual complexidade dos sistemas, a rapidez da evolução dos requisitos e tecnologias, a elevada concorrência do mercado e a restrição cada vez maior de recursos, não há mais espaço para decisões equivocadas ou que provoquem re-trabalho sem uma adequada justificativa. No entanto, tem-se observado que: (i) problemas são, normalmente, mal-entendidos e/ou mal-descritos; (ii) decisões são tomadas sob pressão de tempo; (iii) decisões não são baseadas em dados confiáveis, modelos avaliados experimentalmente ou em boas práticas comprovadas; (iv) decisões são tomadas sem considerar a perspectiva de todos os interessados na questão; (v) decisões não são esclarecidas ou tornadas transparentes para todos os envolvidos; e (vi) decisões são tomadas sem nenhuma metodologia ou utilizando metodologias não confiáveis [Clemen e Reilly 2004].

O apoio à decisão na Engenharia de Software é de grande interesse, tanto para a academia quanto para a indústria. Contudo, Ruhe [Ruhe 2003] comenta que, atualmente, muitas decisões cruciais são tomadas de forma *ad hoc*, baseadas em impressões e sem conexão com as melhores práticas, modelos e experiências. É preciso, pois, envidar esforços no sentido de definir processos, estabelecer metodologias adequadas e integrar o maior número possível de informações disponíveis, para que seja criado um ambiente adequado para auxiliar tomadores de decisão em organizações desenvolvedoras de software.

Durante a realização deste trabalho, diversos Sistemas de Apoio a Decisão (SAD) foram estudados e analisados, visando identificar a possibilidade de atender gerentes e engenheiros de software nas suas atividades rotineiras de tomada de decisão. Exemplos de Sistemas de Apoio à Decisão (SADs) aplicados à Engenharia de Software podem ser encontrados em [Collofello et al. 1998; Raffo et al. 2002; Fenton et al. 2004]. No entanto, todos os sistemas encontrados apóiam um tipo específico de problema com base em modelos voltados para o domínio representado.

Este trabalho tem como meta final desenvolver um Ambiente Apoio à Tomada de Decisão (Oráculo) acoplado ao Ambiente Taba de Desenvolvimento [Santos et al. 2005]. Partindo dos resultados esperados de todos os processos do MR-MPS.BR [SOFTEX 2007] que envolvem tomada de decisão, será realizado um mapeamento entre esses resultados esperados e os processos e atividades do Taba. Dessa forma ao utilizar o Ambiente Taba, um usuário poderá ser assistido durante todo o tempo pelo Ambiente Oráculo, melhorando a qualidade e a velocidade do processo decisório.

O Oráculo terá como pilares três componentes principais: (i) um processo formal de tomada de decisão, (ii) um mecanismo de armazenamento e recuperação de conhecimento e a (iii) seleção de métodos que apóiem a tomada de decisão.

Além desta introdução, a seção 2 do artigo faz uma revisão da literatura sobre as condições de tomada de decisão, das classes de problemas e objetivos e das perspectivas de tomada de decisão. A seção 3 descreve o processo definido para apoiar a tomada de decisão em projetos de software. A seção 4 apresenta os resultados da aplicação do processo proposto. A seção 5 descreve como o ambiente trata os aspectos de

armazenamento e recuperação de conhecimento. A seção 6 enfoca como os métodos deverão ser utilizados nos processos de tomada de decisão. Na seção 7 são apresentadas algumas perspectivas futuras, visando esclarecer os próximos passos deste trabalho e, finalmente, na seção 8 são traçadas conclusões.

## 2. Tomada de Decisão

### 2.1 Condições de Tomada de Decisão

Em todas as formas de tomada de decisão existem sempre dois elementos básicos a serem analisados: os fatores que fazem parte do problema e da solução e suas respectivas probabilidades de ocorrência. Quanto aos fatores, pode-se obter dois estados: conhecimento total ou parcial. Quanto às probabilidades têm-se, também, dois estados: 100% ou algum valor entre 0% e 100%. Dessa forma podemos encontrar três condições onde decisões são tomadas: *Certeza*, *Certeza Parcial* e *Ignorância*. [Klein 1999].

### 2.2 Classes de Problemas e Objetivos

Existem duas classes de problemas e de objetivos: os bem-definidos e os mal definidos [Reitman 1965]. Problemas bem-definidos são aqueles cujos objetivos, caminhos e obstáculos estão claros e baseados em informações confiáveis. Por sua vez, problemas mal definidos são caracterizados pela ausência de um caminho claro que leve à solução. Os objetivos bem-definidos são aqueles que proporcionam ao solucionador uma linha clara de ação em sua direção. Já nos objetivos mal definidos, as metas a serem alcançadas não são claras.

### 2.3 Perspectivas de Tomada de Decisão

De acordo com [Klein 1999] existem duas perspectivas nas quais os seres humanos tomam decisões: a *Natural* e a *Racional*. Na primeira, os decisores estão, geralmente, envolvidos com problemas ou objetivos mal definidos e as decisões são baseadas na experiência. Já na Racional, existe um processo formal de tomada de decisão, ou linha de raciocínio a ser seguida, onde, passo-a-passo, o decisor é levado a atingir o objetivo proposto pelo processo.

Em relação à perspectiva Natural, Klein [Klein 1999] verificou que as pessoas possuem uma gama de habilidades para tomar decisões de forma Natural, às quais ele chamou de Fontes de Poder: Intuição, Simulação Mental, Pontos de Apoio, Analogias e Metáforas, Mente da Equipe, Criação de Estórias e Visão do Invisível.

A perspectiva Racional, segundo o próprio Klein [Klein 1999], também é uma importante fonte de poder, pois fornece os benefícios de ordenar e sistematizar abordagens de solução para problemas tanto complexos como simples. Por sua vez, nossa habilidade de analisar situações requer, apenas, pensamento racional, que é independente de experiência. Algumas outras vantagens da Perspectiva Racional são: (i) permite a discriminação de idéias; (ii) reduz a chance de uma possível alternativa de solução ser negligenciada; (iii) possibilita a ampla busca por muitas opções; (iv) permite o registro de lições aprendidas para reutilização em decisões futuras; (v) permite ao decisor usar seu conhecimento de forma declarativa; e (vi) aproxima-se mais de resoluções livres de erros do que outras abordagens.

Diversas críticas, contudo, são apresentadas à perspectiva racional [Gigerenzer e Selten 2002]: (i) requer um maior tempo de análise comparada à perspectiva natural; (ii) normalmente é aplicada a problemas bem-definidos; (iii) depende de muitas informações para ser colocada em prática; (iv) complexidade ao se aplicar cálculos, pois existe uma grande dificuldade em se obter dados precisos ou estimativas para serem aplicados aos métodos. Vários outros estudos discutem as vantagens e as desvantagens tanto da abordagem natural quanto da racional, tais como [Klein e Weitzenfeld 1978; Lipsitz e Bar-Ilan 1996]. No entanto, alguns fatores tendem a levar o processo de tomada de decisão, no contexto da Engenharia de Software, para a Perspectiva Racional, motivo pelo qual este trabalho adotou a Perspectiva Racional como forma de apoiar o processo de tomada de decisões durante o ciclo-de-vida de projetos de software:

- A Engenharia de Software faz parte de um contexto financeiro e é uma atividade econômica como qualquer outra. Neste sentido, tanto gerentes como técnicos precisam, em muitos casos, embasar e justificar suas decisões de maneira formal [Costa et al. 2004];
- Durante um processo de desenvolvimento de software, geralmente há tempo suficiente para se tomar decisões baseadas em uma análise mais detalhada como a sugerida pela perspectiva racional;
- Permite que o registro dos processos seja reutilizado em futuras decisões, facilitando a geração de conhecimento, o aprendizado organizacional, o aperfeiçoamento do processo e a melhoria dos parâmetros de decisão; e
- Modelos de referência e normas internacionais, tais como o CMMI - Capability Maturity Model Integration [Chrissis et al. 2006], a ISO/IEC 12207 - Software life-cycle processes – Amendment 2 [ISO/IEC 2004], a ISO/IEC 15504 – Software Process Assessment [ISO/IEC 2003] e o MR-MPS.BR – Modelo de Referência para Melhoria de Processo do Software Brasileiro [SOFTEX 2007] - exigem que processos formais de tomada de decisão sejam utilizados em situações onde isto seja necessário para se atingir determinados níveis de maturidade e capacitação em processos de software.

### **3. Processo Formal**

A fim de viabilizar a construção do ambiente de tomada de decisão foi elaborado um processo que conduzisse o decisor a uma solução mais adequada para o problema, devendo ser associado a problemas com risco médio, alto ou que afetem a possibilidade de alcançar os objetivos do projeto, bem como nos momentos em que o impacto da decisão envolver uma quantia determinada do orçamento, alteração significativa do cronograma ou qualidade, decisões técnicas não triviais, etc. Ele pode ser usado tanto para problemas técnicos (escolher o tipo de arquitetura a ser empregado) como para problemas não-técnicos (definir o melhor fornecedor de um produto). No entanto, deve-se atentar para o fato de que o custo de um processo de avaliação formal deve ser aceitável quando comparado ao impacto da decisão. Os critérios para a utilização do processo de decisão formal em uma organização devem estar definidos na sua Política Organizacional.

Não existe uma lista completa de situações onde se deve aplicar um processo formal de decisão, pois sua utilização é extremamente dependente do tipo de organização, do projeto ou até mesmo do produto. No entanto, alguns exemplos passíveis da utilização de um processo formal de decisão são: definição de componentes, decisão sobre construir ou adquirir um produto, seleção de ferramentas, definição de estratégias de contingências de riscos, priorização de recursos, contratação de pessoal e plataformas de sistemas.

### **3.1 Contextualizar a Decisão**

O primeiro passo em um processo de tomada de decisão formal é a contextualização da decisão, a fim de esclarecer dúvidas, definir responsabilidades e o ambiente em que o processo será utilizado. Deve-se considerá-la inicialmente para que seja possível compreender e deixar claro o contexto de utilização do processo. Nesta atividade são definidas questões tais como: a empresa ou o local onde o problema será resolvido, o projeto em pauta, os responsáveis, participantes, o contexto em si e, caso necessário, um glossário dos termos que serão utilizados, para que todos tenham ciência do assunto que está sendo tratado.

### **3.2 Definir o Problema**

Em seguida é necessário definir exatamente qual o problema se deseja resolver, o que terá forte influência sobre as possíveis soluções adotadas. Definir um problema erroneamente pode conduzir o decisor a um caminho que não o levará à solução do problema real. Esta atividade visa assegurar que se pretende resolver o problema correto e deve estar alinhada com os objetivos de negócio da organização. Para uma correta definição do problema e embasamento para a elaboração das possíveis soluções, sugere-se atentar para os detalhes propostos por [Gomes et al. 2004].

### **3.3 Definir Critérios de Seleção**

Em muitos casos, mais de uma variável pode influenciar na escolha da melhor solução. Essas variáveis são chamadas critérios. Dessa forma, os critérios de avaliação devem ser priorizados e/ou ponderados, bem como definidos os parâmetros de aceitação de cada critério, para que eles possam ser aplicados e a melhor solução possa ser escolhida. A priorização e/ou a ponderação dos critérios pode ser feita por uma ou mais pessoas, devendo-se registrar o resultado do trabalho e os motivos que levaram à escolha dos critérios e à sua priorização e/ou ponderação. Pode-se, também, registrar os motivos que levaram à exclusão de alguns critérios. Os critérios escolhidos não devem ser tendenciosos, devendo ser escolhidos apenas aqueles que colaboram para que o objetivo seja alcançado. Na priorização e/ou ponderação de critérios, estes devem ser ordenados de tal forma que aquele com maior grau de prioridade seja o que realmente tem maior influência no processo de decisão.

### **3.4 Definir o(s) Método(s) de Avaliação**

Não existe um consenso sobre qual o melhor método a ser utilizado em um processo de decisão formal, pois eles dependem de diversos fatores, tais como: o nível de precisão requerido na resposta; o tempo disponível para a tomada de decisão; os recursos a serem empregados; o grau de conhecimento da equipe na aplicação de um método específico; a complexidade do problema; e, até mesmo, as informações disponíveis para a tomada

de decisão. Alguns problemas podem necessitar o uso de apenas um método de avaliação, outros, por sua vez, podem requerer diversos métodos para se determinar que alternativa de solução melhor se aplica ao problema definido.

Os métodos a serem usados para avaliação podem variar desde uma simples reunião, a simulações e ao uso de modelos probabilísticos complexos, chegando ao desenvolvimento de sistemas especialistas para situações mais específicas. Exemplos de métodos de avaliação tais como simulação, árvores de decisão, análise de custo-benefício podem ser encontradas em [Clemen e Reilly 2004]. Detalhes sobre métodos mais simples, tais como Multivotação Ponderada, Análise de Pareto e Comparação aos Pares podem ser encontrados em [Wildman e Warner 2003]. Outros métodos mais complexos, tais como Redes Bayesianas [Jensen 1996], Análise multi-critério [Bana e Costa e Vansnick 1995] e Dinâmicas de Sistemas [Barros 2001] também podem ser utilizados.

### **3.5 Identificar Alternativas de Solução**

A identificação de alternativas de solução deve ser realizada de forma a permitir uma boa avaliação e uma implementação correta. Sempre que possível, os principais envolvidos no problema devem estar presentes na execução desta atividade, bem como especialistas e demais pessoas que serão afetadas pelo problema ou pela(s) solução(ões). Uma boa prática para a identificação das possíveis soluções é realizar um trabalho de grupo ou reuniões de *brainstorming*, onde, além das alternativas de solução, sejam levantados os riscos, problemas, vantagens e desvantagens das referidas alternativas, bem como possíveis premissas e restrições para a implementação de uma solução.

É de suma importância, neste momento, a avaliação quantitativa dos riscos envolvidos na implementação (ameaças e oportunidades) de cada solução, pois caso alguma solução seja considerada inviável, devido a uma ameaça que comprometa os objetivos do projeto, esta não deverá ser levada para a próxima fase do processo. No entanto, a análise das oportunidades que uma alternativa pode produzir também deverá ser foco de atenção. Esta avaliação deve considerar a probabilidade de ocorrência, o impacto e, se a implementação desta solução afetará o processo de desenvolvimento, o produto final ou qualquer outra atividade em alguma fase futura.

### **3.6 Avaliar Alternativas**

Avaliar as alternativas significa realizar o trabalho necessário para aplicar os métodos selecionados às possíveis soluções listadas. Durante esta fase do processo, cada alternativa de solução deve ser submetida ao método escolhido e os resultados registrados para futura análise. A aplicação do método deve ser o mais imparcial possível, a fim de não influenciar a avaliação das alternativas, lembrando sempre que os resultados não serão analisados nesta etapa do processo.

### **3.7 Analisar Resultados**

Nesta atividade deve-se comparar os resultados obtidos na avaliação de cada alternativa em relação aos critérios estabelecidos. Ao realizar a análise de uma alternativa, faz-se necessário verificar se ela está adequada às restrições e premissas impostas, tanto pelo problema quanto pela própria alternativa em questão. Deve-se elaborar um breve parecer do resultado obtido após a aplicação dos critérios de seleção a cada alternativa

analisada, sugerindo a solução proposta para que o responsável pela decisão final aprove ou não a solução, visto que nem sempre o responsável pela decisão é a mesma pessoa que executou o processo.

É importante que se verifiquem os riscos associados a cada alternativa, os quais já foram listados e avaliados ao identificá-la. Possíveis respostas aos riscos (contenções ou contingências) necessárias para sua eliminação ou mitigação devem ser definidas. Algumas abordagens para gerenciamento de risco podem ser encontradas em [Farias 2002 e Carr et al. 2003; Chrissis et al. 2006].

### **3.8 Selecionar a Solução**

Selecionar uma solução implica escolher aquela que melhor se enquadra nos critérios determinados e faz com que o problema seja resolvido. Todo o processo de escolha da solução deve ser documentado para que questionamentos futuros possam ser esclarecidos. Os motivos que justificam a escolha de uma solução devem ser descritos, bem como os motivos que levaram à exclusão das demais alternativas.

### **3.9 Implementar a Solução**

Após a escolha de uma alternativa de solução é aconselhável elaborar algumas recomendações para a sua implementação. Isto significa traçar as linhas gerais da forma como a solução escolhida deve ser implementada, podendo conter informações adicionais, procedimentos, responsáveis, etc. Caso sejam necessárias modificações no Plano do Projeto ou em *baselines* previamente estabelecidas, estas deverão ser registradas para que os responsáveis tomem ciência desses fatos.

## **4. Aplicações do Processo**

Para se chegar à atual versão, testar sua validade e realizar aperfeiçoamentos, o processo foi utilizado pela empresa BL Informática que teve os seus processos avaliados em relação às áreas de processo dos níveis 2 e 3 do CMMI e uma avaliação oficial SCAMPI e foi considerada aderente ao nível 3 do CMMI. Durante a iniciativa de melhoria de processos, com o objetivo de apoiar a implantação dos processos de Análise de Decisão e Resolução e Solução Técnica que deveriam estar aderentes ao CMMI, a BL Informática utilizou a ferramenta TechSolution [Figueiredo 2006] e o processo proposto.

Antes da implantação deste processo, as decisões eram simplesmente registradas em ata de reunião e tomadas com base na experiência dos envolvidos, não existindo registro detalhado dos critérios utilizados e a causa da não utilização de uma determinada alternativa. Os maiores benefícios constatados até então, foram: (i) a diminuição da subjetividade em torno das decisões; (ii) o registro detalhado das decisões que começaram a ser utilizados como lição aprendida, apoio e informação para novas decisões; (iii) maior assertividade nas soluções propostas; (iv) maior entendimento dos envolvidos sobre os benefícios e riscos das alternativas, facilitando a decisão; e (v) disseminação do conhecimento obtido durante a execução do projeto.

## 5. A Gestão do Conhecimento

O conhecimento adquirido em um projeto de software, por meio da utilização de técnicas de Engenharia de Software, pode ser melhor aproveitado pela sua transferência para outros projetos. A disseminação das experiências aprendidas permite o melhoramento contínuo do desenvolvimento de projetos de software e também proporciona a diminuição de seus custos. Porém, para que a transferência de conhecimento seja possível e viável, é necessário que as experiências adquiridas sejam catalogadas e armazenadas, de modo que elas não sejam futuramente esquecidas ou perdidas. Além disso, é necessária a identificação de características de cada experiência e do projeto no qual ela se originou para que este conhecimento possa ser transferido entre projetos similares [Souza 2002].

O segundo passo para a construção do ambiente de tomada de decisão foi descobrir um mecanismo ou tecnologia que fosse simples e eficiente no sentido, tanto de armazenar quanto de recuperar informações e conhecimentos adquiridos ao longo do tempo, a fim de auxiliar decisões futuras. Abordagens tradicionais de Gestão de Conhecimento, tais como as de [Nonaka 1994; Meehan e Richardson 2002] não se mostraram ágeis o suficiente para atender aos requisitos do ambiente.

A solução encontrada foi aplicar os conceitos de Raciocínio Baseado em Casos (RBC) [Schank 1982; Aamodt e Plaza 1994] e registrar os passos de realização do processo de decisão formal em forma de casos, bem como palavras-chaves, para que estes possam ser recuperados por questão de similaridade com problemas futuros e, desta forma, auxiliar decisores em suas atividades.

### 5.1 Raciocínio Baseado em Casos

RBC é um enfoque de aprendizado contínuo e incremental que utiliza o conhecimento de situações concretas acontecidas no passado (casos) para resolver novos problemas e aprender por meio da experiência adquirida. O principal objetivo da abordagem RBC é o desenvolvimento de um conjunto de experiências práticas e o fornecimento de um amplo suporte ao desenvolvimento de sistemas apoiado nesta base de conhecimento [Wangenheim 1998].

O uso de RBC em Engenharia de Software não é uma novidade e diversos exemplos bem sucedidos podem ser observados. Na área de estimativas encontramos [Bisio e Malabochia 1995; Finnie et al. 1997]. Em relação ao reuso de software tem-se o exemplo de [Ostertag et al. 1992]. Outras aplicações podem ser encontradas em [Rodrigues 2000]. Desta forma, acredita-se que quanto mais casos forem cadastrados no ambiente, maiores serão as possibilidades de um decisor beneficiar-se de casos anteriores, eliminando dúvidas, aproveitando erros e acertos passados e contribuindo cada vez mais para o aperfeiçoamento, tanto de suas decisões quanto da base de conhecimento.

Em um processo RBC, a descrição inicial de um problema define um novo caso. Este novo caso é utilizado para recuperar um ou mais casos da coleção de casos passados que estão armazenados na base de experiências. O problema de representação de conhecimento em RBC está em decidir o que armazenar em um caso, encontrar uma estrutura apropriada para descrever o conteúdo dos casos, e decidir como uma base de

casos deverá ser organizada e indexada para uma efetiva recuperação e reutilização do conhecimento contido nela [Souza 2002].

Esses problemas foram solucionados da seguinte forma: a partir da definição dos processos e atividades definidos na estação Taba, foram mapeadas todas as situações onde sejam possíveis tomadas de decisão. Em seguida, foram analisados todos os resultados esperados dos processos do MR-MPS.BR a fim de se obter uma visualização de quais resultados eram passíveis de tomada de decisão. Finalmente foi realizado um mapeamento entre esses resultados esperados e as situações problema dos processos e atividades da estação Taba. Assim, já se pode fornecer ao usuário do ambiente uma série de situações onde possivelmente ele poderá tomar decisões formais.

Em relação à estrutura dos casos eles serão registrados de acordo com o padrão estabelecido para o processo formal adaptado à situação que se deseja resolver. Por exemplo, ao ser iniciado um processo de tomada de decisão durante a fase de definição de arquitetura, todos os itens do processo formal referentes a esta situação, já estarão previamente preenchidos, restando ao usuário o preenchimento dos itens referentes ao seu caso específico.

Desta forma, dois problemas foram automaticamente resolvidos, quais sejam: o que armazenar, definido pelas situações-problema; a estrutura de armazenamento dos casos, que ficou definida como sendo a do processo formal.

O último problema a ser resolvido era como seria o cálculo grau de similaridade entre os casos cadastrados na base de casos. Isto foi resolvido da seguinte maneira: a indexação dos casos será composta por dois níveis. O primeiro nível será referente ao Resultado Esperado do MR-MPS-BR a que ao caso faz referência.. O segundo nível será relativo ao Contexto do Caso (primeiro passo do processo formal) e à Definição do Problema (segundo passo do processo formal). Nesse segundo nível a indexação será feita por palavras-chaves associadas tanto ao contexto quanto à definição do problema do caso em questão, sendo esta associação feita pelo próprio usuário que está sendo responsável pelo caso. Vale ressaltar que no Ambiente Taba já existem diversas palavras-chaves já cadastradas que serão utilizadas como base para os cadastros dos índices, sendo que um usuário será capaz de cadastrar novas palavras-chaves.

A partir dos dados fornecidos pelo usuário, a busca será feita por um algoritmo que realizará uma pesquisa na base de casos e disponibilizará uma listagem de casos em ordem de grau de similaridade com os dados fornecidos. Assim, se um caso possuir em seu banco de palavras-chaves os mesmos dados fornecidos e estiver associado ao mesmo Resultado Esperado pelo usuário no momento da busca, ele será 100% semelhante ao seu caso e, possivelmente terá grande valia para auxiliar a solução do problema em questão. Consequentemente, quanto menos índices coincidentes, menos similares serão os casos.

A fim de ilustrar a proposta, considere o seguinte exemplo: 03 (três) casos cadastrados e diversas palavras-chave (índices hipotéticos) representadas pelas letras de A a M, fornecidas no momento do armazenamento do caso na base de casos, conforme a Tabela 5.1

**Tabela 5.1 – Base de casos**

Caso 1		Caso 2		Caso 3	
Contexto	Índices	Contexto	Índices	Contexto	Índices
Texto descritivo para contexto do caso 1.	A,B,C, D	Texto descritivo para contexto do caso 2.	B,C,E	Texto descritivo para contexto do caso 3.	A, F, G
Problema	Índices	Problema	Índices	Problema	Índices
Texto descritivo para o problema do caso 1.	H, I,J	Texto descritivo para o problema do caso 2.	J, K, L	Texto descritivo para o problema do caso 3.	I, K, M

Supondo que um usuário esteja tentando resolver um novo problema referente a um dado resultado esperado do MR-MPS-BR (por exemplo, RE1) e queira buscar casos semelhantes para auxiliar seu processo decisório. Ele indicará que seu caso se refere a este Resultado Esperado e a busca se restringirá aos casos vinculados a RE1 na base de casos.

Baseado na análise dos fatores envolvidos, o usuário informa que as palavras-chave que mais caracterizam o Contexto seu Problema são os índices (A, B e C) e as que mais caracterizam sua Definição de Problema são os índices (I e J). Portanto, estes índices serão usados como palavras-chaves na tentativa de busca de casos similares.

Com base nestas informações, o cálculo de similaridade é realizado da seguinte forma:  $N_c / N_T$ , onde (i)  $N_c$  - nº de coincidências entre as palavras-chaves fornecidas e os índices definidos para o Contexto e a Definição do Problema e (ii)  $N_T$  - nº total de palavras-chaves fornecidas. O cálculo é efetuado separadamente tanto para o Contexto quanto para a Definição do Problema. Finalmente, calcula-se o somatório da média ponderada destes resultados. Para efeito de simplificação os pesos estão sendo considerados idênticos (1/2) tanto para o Contexto quanto para a Definição do Problema. Dessa forma, a busca teria um resultado como e exibido na Tabela 5.2.

**Tabela 5.2. Resultado da Busca**

Caso	Similaridade	Explicação
Caso 1	100 %	Das 3 ( $N_t$ ) palavras-chaves informadas na busca para Contexto (A,B,C), as 3 ( $N_c$ ) A, B e C foram definidas como índices do caso 2. Contexto= $3/3=1$ . Das 2 ( $N_t$ ) palavras-chaves informadas para a busca para problema (I, J), 2 ( $N_c$ ) I e J foram definidas como índice do caso 2. Problema= $2/2=1$ Similaridade = $(1 \times 1/2) + (1 \times 1/2)=1$
Caso 2	58,3 %	Das 3 ( $N_t$ ) palavras-chaves informadas na busca para Contexto (A,B,C), apenas 2 ( $N_c$ ) B e C foram definidas como índices do caso 2. Contexto= $2/3=0,666$ . Das 2 ( $N_t$ ) palavras-chaves informadas para a busca para problema (I, J), apenas 1 ( $N_c$ ) J foi definida como índice do caso 2. Problema= $1/2=0,5$ Similaridade = $(0,666 \times 1/2) + (0,5 \times 1/2)=0,583$
Caso 3	41,7 %	Das 3 ( $N_t$ ) palavras-chaves informadas na busca para contexto (A,B,C), apenas 1 ( $N_c$ ) A foi definida como índice do caso 3. Contexto= $1/3=0,333$ . Das 2 ( $N_t$ ) palavras-chaves informadas para a busca para problema (I, J), apenas 1 ( $N_c$ ) I foi definida como índice do caso 2. Problema= $1/2=0,5$ Similaridade = $(0,333 \times 1/2) + (0,5 \times 1/2)=0,417$

Com base no resultado, o usuário do sistema tem condições de avaliar que casos apresentam maior grau de similaridade em relação ao seu problema, o que o ajudará a

reutilizar adequadamente experiências anteriores para avaliar possíveis soluções e optar por uma alternativa adequada.

## 6. Métodos de Tomada de Decisão

Uma das maiores dificuldades observadas durante a aplicação do processo em empresas é escolher e aplicar um método às alternativas encontradas de forma a determinar qual a mais adequada à solução do problema.

Conforme descrito na seção 3.4, diversos métodos podem ser aplicados, desde os mais simples aos mais elaborados, dependendo de quão crítico é o problema. No entanto, observa-se que métodos mais complexos tais como Simulações, Redes Bayesianas, Dinâmica de Sistemas e Redes Neurais requerem um longo tempo para que se desenvolvam modelos confiáveis, bem como são mais utilizados para sistemas especialistas. Levando-se em conta que o objetivo do ambiente é fornecer apoio genérico para a tomada de decisão, buscou-se na literatura um conjunto de métodos e ferramentas que fossem simples de serem empregados, mas que oferecessem confiabilidade em suas aplicações, de tal forma que a grande maioria dos problemas vivenciados no dia-a-dia de projetos de software pudessem ser atendidos – Princípio de Pareto [Aron 1937].

Wildman e Warner (2000) realizaram um extenso estudo sobre métodos mais simples que têm sido usado ao longo da história para auxiliar o processo de decisão em diversas áreas de conhecimento. Desta forma, pretende-se aproveitar a experiência do autor para aplicá-la na proposta deste trabalho. São eles: Árvores de Decisão, Matriz de Avaliação, *Force Field Analysis*, Matriz de Análise de Decisão, Multivotação, MAUT, Comparação aos Pares, PMI (*Plus, Minus, Interesting*) e *Six Thinking Hats*.

A fim de descobrir que métodos eram adequados a quais tipos de problema, foi realizada uma busca por taxonomias de classificação de métodos de tomada de decisão, pois, dependendo das características do problema (tipos de informações disponíveis), um ou outro método é mais apropriado. A taxonomia de classificação que mais se enquadrou no contexto deste trabalho foi a proposta por [Ullman e D’Ambrosio 1995]. Originalmente, esta taxonomia foi utilizada para classificar Problemas e Sistemas de Apoio à Decisão. No entanto, demonstrou-se perfeitamente adequada aos propósitos deste trabalho, ou seja, classificar problemas e Métodos de Tomada de Decisão. A taxonomia é apresentada na Tabela 6.1.

**Tabela 6.1. Taxonomia de Classificação de Problemas e Métodos**

<b>Estrutura</b>	<b>Espaço de Decisão</b>	1. Completeza do Problema
		2. Nível de Abstração
		3. Determinismo
	<b>Modelo de Preferência</b>	4. Funções Objetivas
		5. Consistência
		6. Base de Comparação
	<b>Modelo de Crença</b>	7. Dimensão
		8. Completeza de Crença
<b>Foco</b>	9. Foco do Problema	
<b>Faixa</b>	10. Faixa de Independência	
<b>Suporte</b>	11. Nível de Suporte	

Com base nessa taxonomia os problemas e os métodos acima listados podem ser classificados e, em seguida, os métodos adequados para utilização no problema em questão podem ser disponibilizados para o usuário do ambiente. Por exemplo, já foi realizada previamente a classificação de todos os métodos escolhidos para a utilização deste trabalho conforme a Tabela 6.2. A partir deste ponto, todas as situações-problema de cada fase dos processos Ambiente Taba também serão classificadas. Assim é possível identificar qual ou quais métodos são adequados para cada tipo de problema.

**Tabela 6.2. Classificação dos Métodos**

Características	Valores	Métodos								
		Force Field Analysis	Multivotação	PMI	Seis Chapéus	Árvore de Decisão	Matriz de Avaliação	Análise da Decisão	Comparação aos Pares	MAUT
Completeza do Problema	Completo	X	X		X	X	X	X	X	X
	Incompleto			X						
Nível de Abstração	Quantitativo			X		X	X	X	X	X
	Qualitativo	X	X	X	X		X	X	X	
Determinismo	Determinístico		X	X	X		X	X	X	
	Probabilístico	X			X	X				X
Funções Objetivas	Ponto Ótimo					X				
	Julgamento	X	X	X	X		X	X	X	X
Consistência	Consistente		X	X		X			X	
	Inconsistente	X			X		X	X		X
Base de Comparação	Absoluta		X	X	X	X				X
	Relativa	X					X	X	X	
Dimensão	Confiança	X	X	X			X	X	X	
	Conhecimento	X				X				
	Ambos	X			X					X
Completeza de Crença	Completa	X	X	X	X	X	X	X	X	X
	Incompleta									
Foco do Problema	Processo	X	X	X	X	X				X
	Produto		X		X	X	X	X	X	
Faixa de Independência	Independente	X	X	X			X	X	X	X
	Dependente			X		X				
	Interdependente				X					
Nível de Suporte	Representação	X	X							
	Definição dos Resultados				X	X	X	X	X	
	Análise da Decisão			X						X

## 7. Perspectivas Futuras

Após a definição dos três componentes básicos do ambiente, este trabalho terá continuidade com a construção do ambiente propriamente dito e sua disponibilização junto ao Ambiente Taba.

O objetivo final é gerar um corpo de conhecimento sólido com experiências reais e extrair informações relevantes no contexto de tomada de decisão em Engenharia de Software. Acredita-se que mantendo uma base centralizada de dados, meta-análises poderão ser realizadas e o conhecimento poderá ser gerado mais rapidamente dentro de uma empresa. Pretende-se, também, realizar pesquisas para uma melhor definição da similaridade entre casos, avaliando a definição de pesos para os fatores *Contexto* e *Definição do Problema*.

## 8. Conclusões

Nesse artigo, foram descritos os três principais passos realizados para a proposta de construção de um ambiente de tomada de decisões em projetos de software. O primeiro passo envolveu a definição de um processo formal que pudesse ser aplicado a qualquer tipo de problema dentro do contexto da Engenharia de Software. O processo definido foi aplicado em situações reais, aprimorado e estabelecido como padrão em virtude de sua comprovada eficiência. O segundo passo foi buscar uma forma de armazenar e recuperar conhecimento de soluções de problemas anteriores. Após uma extensa revisão na literatura, adotou-se o Raciocínio Baseado em Conhecimento como a abordagem utilizada para realizar esta tarefa. Finalmente, era necessário fornecer ao usuário do ambiente um conjunto de métodos que auxiliasse na escolha da solução mais adequada. Dessa forma, um conjunto de métodos foi selecionado e classificado para que, dependendo da necessidade do usuário, o método adequado seja disponibilizado.

## 9. Agradecimentos

Agradecemos à BL Informática e a Analia Irigoyen pela utilização do processo e valiosas sugestões. Agradecemos também ao Sávio Figueiredo, que possibilitou a avaliação prática inicial do processo com a ferramenta Techsolution, desenvolvida durante sua tese de mestrado [Figueiredo 2006].

## 10. Referências

- Aamodt, A., Plaza, E., (1994), "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *AI Communications*, vol. 17, nr. 1.
- Aron, R., (1937), "La sociologie de Pareto", *Zeitschrift für Sozialforschung*.
- Bana e Costa, C.A. e Vansnick, J.C., (1995) "A Theoretical Framework for Measuring Attractiveness by a Categorical Based Evaluation Technique (MACBETH)". In Clímaco, J. *Multicriteria Analysis*. Berlin: Springer Verlag.
- Barros, M. O., (2001) "Gerenciamento de Projetos Baseado em Cenários: uma Abordagem de Modelagem Dinâmica e Simulação", Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- Bisio, R. and Malabocchia, F., (1995), "Cost estimation of software projects through case base reasoning". 1st Intl. Conf. on Case-Based Reasoning Research & Development. pp11-22: Springer-Verlag.
- Carr, M. J., Konda, S.L, Monarch, I., Ulrich, F.C., Walker, C.F., (1993) "Taxonomy-Based Risk Identification", Technical Report CMU/SEI-93-TR-6, Software Engineering Institute, Carnegie Mellon University, EUA, July.

- Chrissis, M. B., Konrad, M., Shrum, S., (2006). “CMMI: Guidelines for Process Integration and Product Improvement” (2nd Edition). Addison-Wesley Professional.
- Clemen, R. T., and Reilly, T., (2004) “Making Hard Decisions”, Duxbury Thomson Learning, CA, United States.
- Collofello, J., Rus., I., Houston, D., Sycamore, D., Mith-Daniels, D., (1998), “A system Dynamics Software Process Simulator for Staffing Policies Decision Support”. In *Proceedings of the 31st Annual Hawaii International Conference on Systems Sciences*, pp. 103-111, Kohala Coast, Estados Unidos.
- Costa, H. R., Barros, M., O., Travassos, G., H., (2004). Software Project Risk Evaluation Based on Specific and Systemic Risks. In *Proceedings of the 16th International Conference of Software Engineering and Knowledge Engineering*.
- Farias, L. L., (2002), Planejamento de Riscos em Ambientes de Desenvolvimento de Software Orientados à Organização, Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- Fenton, N., Marsh, W., Neil, M., Cates, P., Forey, S., Tailor, M., (2004), Making Resource Decisions for Software Projects. In *Proceedings of the 26th International Conference on Software Engineering*.
- Figueiredo, S. M. (2006). Apoio à Tomada de Decisão no Processo de Solução Técnica em Ambientes de Desenvolvimento de Software Orientados à Organização. Dissertação de Mestrado COPPE/UFRJ. Rio de Janeiro, 2006.
- Finnie, G. R., Wittig, G. E. and Desharnais, J. M., (1997), “Estimating software development effort with case-based reasoning”. 2nd Intl. Conf. on Case Based Reasoning. pp13-22.
- Gigerenzer, G., Selten, R., (2002), “Bounded Rationality, the adaptive toolbox”, MIT Press.
- Gomes, L. F. A. M.; Araya, M. C. G.; Carignano, C., (2004) “Tomada de Decisões em Cenários Complexos”. Ed. Thompson, SP.
- Hoffmann, H., (1997), “A Decision Support System for Selection of Software Engineering Technologies Based on Models of their Application Domains”, Fraunhofer IESE, Germany.
- ISO/IEC, (2003), The International Organization for Standardization and The International Electrotechnical Commission, ISO/IEC 15504 - Software Process Assessment.
- ISO/IEC, (2004), The International Organization for Standardization and The International Electrotechnical Commission ISO/IEC 12207 – Software life cycle processes – Amendment 2.
- Jensen, F. V., (1996) “An introduction to Bayesian Networks”, UCL Press.
- Klein, G., Weitzenfeld, J., (1978) “Improvements of skills for solving ill defined problems”. *Educational Psychologist* 13:13-41.
- Klein, G., (1999), “Sources of Power: How people make decisions”, MIT Press, Massachusetts, USA.

- Lipshitz, R., Bar-Ilan, O., (1996), "How problems are solved: Reconsidering the phase theorem". *Organizational Behavior and Human Decision Process*, 65, 48-60.
- Meehan, B. and Richerdson, I., (2002), "Identification of Software Process Knowledge". *Software Process Improvement and Praactice*.
- Nonaka, I., (1994), "A dynamic theory of Organizational Knowledge Creation". *Organizations Science*, vol. 5, p. 14-37.
- Ostertag, E., et al., (1992), "Computing similarity in a reuse library system: an AIbased approach. *ACM Transactions on Software Engineering Methodology* 1(3): pp205-228.
- Raffo, D., Harrison, W., Vandeville, J., (2002), "Software Process Decision Support: Making Process Tradeoffs Using Hybrid Metrics, Modeling and Utility Framework". In *Proceedings of the SEKE 2002*, pp. 803-809, Ischia, Itália.
- Reitman, W. R., (1965), "Cognition and Thought". New York: Wiley.
- Rodrigues, M. R., (2000), "Desenvolvimento e Implementação de um Protótipo de Ferramenta para a Reutilização de Planos de Mensuração Utilizando Raciocino Baseada em Casos", Universidade Federal de Santa Catarina.
- Ruhe G., (2003), "Guest Editor's Introduction", *International Journal of Software Engineering and Knowledge Engineering* Vol. 13, No. 5.
- Rus I., Ilalling M., and Biffi S., (2003), "Supporting Decision Making in Software Engineering with Process Simulation and Empirical Studies". In *Proceedings of the 4th Workshop on Learning Software Organizations*, Chicago.
- Santos, G., Montoni, M., Rocha, A.R., Figueiredo, S., Mafra, S., Albuquerque, A., Diaz P., B., Amaral, M., (2005). "Using a Software Development Environment with Knowledge Management to Support Deploying Software Processes in Small and Medium Size Companies", 7th Int. Workshop on Learning Software Organizations (LSO' 2005), Kaiserslautern, Alemanha, Apr.
- Schank, R. C., (1982), "Dynamic Memory: A theory of reminding and learning in computers and people", Cambridge, UK: Cambridge University Press.
- SOFTEX, (2007), "MPS.BR – Melhoria de Processo do Software Brasileiro, Guia de Implementação Parte 5 (v1.1)". In: [http://www.softex.br/mpsbr/\\_guias/default.asp](http://www.softex.br/mpsbr/_guias/default.asp).
- Souza, M. P., (2002) "Identificação de Características Relevantes para reutilização de experiências de desenvolvimento de software". Dissertação de Mestrado da UFSC – Engenharia de Produção.
- Ullman, D. e D'ambrosio B., (1995), A Taxonomy for Classifying Engineering Decision Problems and Support Systems, *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, AI EDAM #9, pp. 427-438.
- Wangenheim, C. G., Wangenheim, A., Barcia, R. M., (1998), "Case-Based Reuse of Software Engineering Measurement Plans". In *Proceedings of the 10th Int. Conference on Software Engineering and Knowledge Engineering*.
- Wildman, P. and Warner, J, (2003) "The problem solving decision-making toolkit", HRD Press, Amherst , Massachusetts .