

## **GESTÃO DE ESCOPO EM PROJETOS DE APLICAÇÕES WEB**

### **SCOPE MANAGEMENT IN WEB APPLICATION PROJECTS**

**Cristina Coelho de Abreu Pinna**

Mestre em Engenharia

Universidade de São Paulo

Sistemas Digitais

Av. Prof. Almeida Prado, tr. 2, nº 128

[cpinna@uol.com.br](mailto:cpinna@uol.com.br)

**Marly Monteiro de Carvalho**

Professora Livre-docente

Universidade de São Paulo

Departamento Engenharia de Produção

Av. Prof. Almeida Prado nº 531 2º Andar CEP: 05508-900 - Cidade Universitária - São Paulo

Tel.: (11) 3091-5363 – [marlymc@usp.br](mailto:marlymc@usp.br)

### **RESUMO**

Os projetos de *software* estão sujeitos a uma série de incertezas e riscos, especialmente aqueles que envolvem aplicações Web, em virtude da competição mercadológica na busca de disponibilizar cada vez mais novidades em serviços eletrônicos e virtuais em tempos cada vez menores (*Internet timing*).

Outro ponto crítico está no foco adotado para a gestão de projetos de aplicações Web – foco na tecnologia ou no processo de negócio ?

Este fato gera dificuldades na homologação do *software* pelo cliente, que poderiam ser contornadas com um bom gerenciamento do escopo do projeto desde fases iniciais do projeto (concepção e planejamento).

O trabalho irá apresentar uma abordagem que pode ser utilizada na gestão de projetos de aplicações Web para minimização das incertezas e riscos na definição de escopo e de requisitos que afetam projetos desta natureza. Será realizada uma análise comparativa e crítica entre diversos instrumentos e técnicas que podem ser utilizados na definição de escopo e

prioridades, tais como: *Quality Function Deployment* (QFD), Modelo do Processo de Negócio e Particionamento.

**Palavras-chave:** Gestão de Projetos, Gestão de Escopo, Aplicação Web, QFD.

## ABSTRACT

Software projects are subjected to certain risks and uncertainties, especially those that include Web applications, due to the fact that market competition aims to have electronic and virtual services available in a shorter and shorter period of time (also known as “Internet timing”).

Another critical factor relates to the focus to be adopted into the management of Web application projects – should technology or business process be emphasized?

This fact produces some difficulties in the acceptance of the software by the customer. This problem could be minimized by appropriate management of the project scope from its earliest phases (conception and planning).

The article will present an approach that can be utilized on the management of Web applications projects for the minimization of uncertainties and risks when defining scope. A comparative and critical analysis will be carried out among several instruments and techniques that can be utilized on the definition of scope and priorities such as: QFD, Business Process Modeling and Partitioning.

**Key-words:** Project Management, Scope Management, Web applications, QFD.

## 1. INTRODUÇÃO

Os projetos de *software*, devido a sua natureza, estão sujeitos a uma série de incertezas e riscos relacionados com custos, escopo/requisitos, recursos humanos e materiais, prazos, expectativas do cliente, da equipe, do investidor, dentre outros. Tais incertezas e riscos são acentuados em projetos que envolvem aplicações Internet, desde a construção de um simples site institucional até sites de comércio eletrônico (B2C – *Business-to-Consumer* e B2B – *Business-to-Business*), em virtude da competição mercadológica na busca de disponibilizar cada vez mais novidades em serviços eletrônicos e virtuais em tempos cada vez menores (denominado *Internet timing*) (BROWN, 2000).

Quando estas incertezas e riscos não são adequadamente gerenciados, a qualidade do produto final pode ser comprometida, a expectativa do cliente pode não ser atendida, e a equipe que precisa conviver com ansiedades e conflitos durante a vida do projeto pode ter sua produtividade reduzida.

Em projetos de desenvolvimento de aplicações Web, muitas vezes, os prazos de implantação dos sistemas são curtos e pré-definidos pelo cliente (conceito de *time-boxing*), cabendo então uma definição de escopo e requisitos adequada para garantir a qualidade e adequação da expectativa em relação ao produto a ser entregue.

Outro ponto crítico está no foco adotado para a gestão de projetos de aplicações Web – foco na tecnologia ou foco no processo de negócio. Muitas vezes, por preciosismo, o projetista ou o desenvolvedor se foca nos requisitos tecnológicos mais sofisticados ou inovadores, mas que não trazem benefícios para o cliente do ponto de vista de negócio. Este fato causa então dificuldades na homologação do *software* pelo cliente, por este não atender suas expectativas e/ou necessidades (McCONNEL, 1996).

Considerando o ciclo de vida de um projeto, as incertezas e riscos relacionados com o escopo do projeto e requisitos do produto são grandes nas fases iniciais do projeto (concepção e planejamento) e tendem a diminuir nas fases posteriores.

Por outro lado, os custos envolvidos na identificação de problemas e inconsistências nas fases iniciais de um projeto são muito menores se comparados com os custos na identificação de problemas em fases posteriores, em virtude dos retrabalhos. Segundo Pressman (2005), uma mudança identificada na fase de desenvolvimento custa de 1,5 a 6 vezes mais do que se ela fosse identificada na fase de definição de requisitos e de 60 a 100 vezes mais na fase de manutenção, depois que o sistema está implantado.

Portanto, a adoção de métodos e técnicas que apóiam a definição clara e precisa do escopo de um projeto de *software*, bem como a priorização e aderência dos requisitos do produto com as necessidades do cliente já nas fases iniciais do projeto permitem garantir a qualidade do produto final, reduzindo custos de retrabalho e expectativas inadequadas.

De maneira resumida, pode-se afirmar que os projetos de aplicações Web estão frequentemente sujeitos a aumento de custos, prazos e retrabalhos.

O objetivo deste trabalho é demonstrar que a adoção de alguns métodos e ferramentas conceituais de gestão já nas fases iniciais do projeto podem apoiar na minimização dos riscos e das incertezas referentes à definição de escopo e de requisitos para desenvolvimento de projetos de *software*, com enfoque especial para projetos de aplicações Web. Assim, a abordagem proposta visa aumentar a qualidade e a produtividade do desenvolvimento, bem como garantir a redução de custos em projetos de aplicações Web.

O trabalho também visa apresentar o resultado da análise crítica da aplicação destas técnicas e ferramentas em projetos de *software*, através da verificação dos benefícios alcançados e das dificuldades encontradas na sua utilização.

Este artigo está estruturado em seis seções. Na seção 2 apresenta-se a síntese do quadro teórico utilizado para o desenvolvimento deste trabalho. Na seção 3 são apresentados os aspectos metodológicos da pesquisa. As seções 4 e 5 apresentam uma aplicação do método desenvolvido e a discussão dos resultados do caso analisado. Finalmente, na seção 6, apresentam-se as conclusões deste trabalho.

## 2. CONCEITUAÇÃO

Para o desenvolvimento deste trabalho foram utilizados os seguintes fundamentos conceituais: Modelo do Processo de Negócio, Particionamento, Arquitetura de Sistema e Desdobramento da Função Qualidade (QFD - *Quality Function Deployment*). Uma breve explicação de cada um destes fundamentos é apresentada a seguir.

### **Modelo do Processo de Negócio**

O modelo do processo de negócio permite estabelecer os fluxos dos processos e informações de um processo de negócio, independente do grau de automação previsto.

A formalização destes modelos pode ser feita utilizando-se diversas notações. As mais utilizadas atualmente são o SADT/IDEF0, que permite a descrição dos processos e sub-processos de negócio, destacando as entradas, saídas, ferramentas, mecanismo e regras de negócio (MARCA, 1988) e o UML Use Case, que permite a descrição detalhada das diversas situações de uso do processo de negócio segundo as necessidades do usuário.

Tais modelos formalizados podem ser validados com o cliente ou usuário, garantindo o entendimento e alinhamento do processo e das regras de negócio.

O estabelecimento do escopo e dos requisitos do produto de *software* de acordo com este modelo permite o alinhamento da automação com os objetivos do negócio. Este é o foco mais adequado para um sistema de *software*: ser uma ferramenta que apóia um processo de negócio e nunca o contrário, ou seja, o foco deve ser eficácia e não eficiência.

Portanto, destaca-se que o modelo de negócio é uma entrada fundamental para o estabelecimento do escopo do projeto de *software*.

No desenvolvimento de *software*, escopo pode ser definido como:

Capturar o contexto, os requisitos e restrições mais importantes para assim, definir os critérios de aceite do produto final. (KRUCHTEN, 1998)

A utilização de técnicas de gerenciamento de projetos em organizações focadas em Tecnologia da Informação (TI) tem se intensificado nos últimos anos. No contexto organizacional, os modelos de maturidade têm se difundido entre eles destacam-se o *Capability Maturity Model* (CMM e CMMI) e o *Organizational Project Management Maturity Model* (OPM3) (CARVALHO *et al*, 2003). Já no âmbito do gerenciamento de projetos o uso de guias de referência como o *Project Management Body of Knowledge* (PMBOK) e a certificação profissional *Professional Project Management* (PMP), fornecido pelo *Project Management Institute* (PMI), têm crescido significativamente entre os profissionais da área de TI (CARVALHO; RABECHINI Jr, 2005).

Segundo o guia PMBOK (PMI, 2004), existem cinco processos relevantes para a condução da gerência do escopo do projeto, quais sejam: planejamento do escopo, definição do escopo, criação da estrutura analítica do projeto - EAP (*work breakdown structure* - WBS), verificação do escopo e controle de mudanças do escopo.

Este trabalho enfatiza os três processos iniciais da Gestão do Escopo, ou seja, o desdobramento dos principais subprodutos do projeto de *software* em partes menores e mais fáceis de gerenciar.

A abordagem do PMBOK (2004) sugere a utilização de modelos de estrutura analítica do projeto (EAP) e decomposição como ferramentas para o detalhamento do escopo.

Não obstante, no contexto da engenharia de *software*, a técnica de particionamento proposta por Pressman (2005) pode trazer uma contribuição relevante como ferramenta para o processo de detalhamento do projeto, aliada ao método QFD para a decomposição sucessiva em partes mais gerenciáveis para a equipe, mas garantindo o alinhamento com os objetivos do negócio.

## **Particionamento**

Segundo Pressman (2005), o particionamento é uma técnica que permite dividir um projeto complexo em partes menores, garantindo um melhor gerenciamento e controle dos requisitos e escopo da partição (conceito do “dividir para conquistar”). Esta técnica também permite que os prazos de entrega sejam reduzidos, em virtude de entregas parciais e desenvolvimentos simultâneos.

Assim, quando se trata de projetos para Internet, esta técnica pode ser bem aplicada, garantindo entregas mais rápidas, mesmo que com escopo reduzido, porém controlado e negociado com o cliente. Versões posteriores do produto irão complementar as entregas parciais iniciais, permitindo o desenvolvimento incremental e evolutivo.

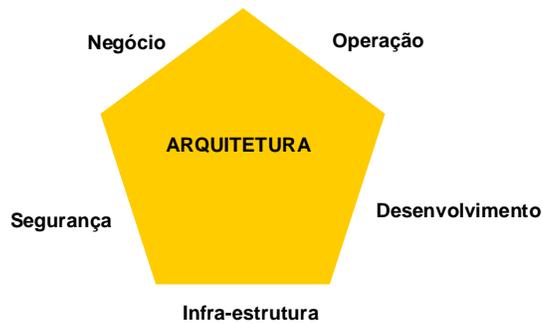
O desafio na aplicação desta técnica consiste em escolher as partições a serem entregues para o cliente e que lhes sejam úteis. Neste sentido, esta técnica é utilizada em conjunto com o modelo de processo de negócio, ou seja, o particionamento é orientado pelo processo de negócio, o que consiste em uma estratégia adequada para o cliente.

## **Arquitetura de Sistema**

Quando o usuário estabelece os requisitos de um software, em geral, ele se preocupa com os requisitos e funcionalidades de negócio, pois são estes que atendem diretamente suas necessidades. No entanto, outros requisitos são de fundamental importância para garantir a qualidade do produto de software como um todo, tais como infra-estrutura de *hardware*, *software*, comunicação e segurança.

A arquitetura de um sistema pode ser caracterizada por um conjunto de visões, como apresentado na Figura 1 (ver ISO 9126/ NBR 13596, ABNT, 1997).

Esta abordagem permite estabelecer todos os requisitos considerando os requisitos solicitados pelo cliente (geralmente os de negócio) e também os demais requisitos fundamentais para a qualidade do produto final. Assim, a abordagem da Arquitetura de Sistema permite garantir completude dos requisitos de um sistema de *software*.



Visão	Abrangência
Negócio	Funcionalidades, Interação Homem-Máquina, Modelos de Processo de Negócio.
Operação	Processo que sustentam o sistema, monitoração e parametrização de regras de negócio pelos gestores.
Segurança	Integridade, privacidade, confiabilidade e rastreabilidade.
Desenvolvimento	Ambientes, processos de desenvolvimento, testes, homologação, implantação, ferramentas e controle de qualidade.
Infra-estrutura	Hardware, software, Bases de Dados e comunicação.

Figura 1: Visões da Arquitetura de um Sistema.

Fonte: ISO 9126 / NBR 13596 (ABNT, 1997).

### Desdobramento da Função Qualidade - QFD

O QFD (*Quality Function Deployment*) é um método que permite o desdobramento e priorização dos requisitos do cliente nas características e processos de qualidade a serem implementados, garantindo alinhamento com as necessidades do cliente. O QFD orienta a organização dos recursos disponíveis, priorizando-os de acordo com a visão do cliente.

O SQFD (*Software Quality Function Deployment*) representa a transferência da utilização do QFD em um ambiente industrial para o mundo do desenvolvimento de *software* (Spinola, 2000). As principais vantagens da utilização do SQFD são:

- Aumento da produtividade no período de programação, por garantia de escopo bem definido;
- Menor necessidade de realizar modificações futuras no *software*;
- Menor necessidade de manutenção, possibilitando que os recursos sejam destinados para o desenvolvimento de novos projetos.

O Quadro 1 apresenta um resumo das ferramentas conceituais apresentadas nesta seção, bem como seus benefícios e pontos críticos.

Quadro 1: Resumo - Ferramentas, Benefícios e Pontos Críticos

<i>Ferramenta</i>	<i>Minimiza incertezas de ...</i>	<i>Benefício</i>	<i>Ponto Crítico</i>
Particionamento	- Prazo, escopo e custo (projetos menores são mais gerenciáveis).	- Entregas parciais e mais rápidas; - Validações intermediárias pelo usuário.	- Integração das partes (comunicação e arquitetura são fundamentais); - Expectativas devem ser bem gerenciadas.
Arquitetura e Processo de Negócio	- Expectativas (aderência ao negócio); - Escopo e requisitos.	- Completeza de requisitos; - Aderência ao negócio.	- Aderência com as demais visões da arquitetura de sistema.
QFD	- Escopo, expectativas e prioridades (alinhamento com as necessidades de negócio do cliente)	- Direcionamento e otimização dos esforços despendidos.	- Complexidade de utilização (quando a quantidade de requisitos for muito grande, trabalhar com particionamento).

### 3. ASPECTOS METODOLÓGICOS

O planejamento de um projeto de *software* deve englobar, basicamente, dois aspectos: “o quê” deve ser feito e “como” deve ser feito.

Em relação ao “o quê” deve ser feito, deve ser realizado o entendimento e modelagem do processo de negócio para, a partir daí, serem estabelecidos os requisitos de automação e o escopo do projeto.

O “como” pode ser respondido pela definição das prioridades e estratégias que serão utilizadas na obtenção do *software*.

A abordagem proposta compreende os seguintes passos:

**Passo 1 - Modelagem de Negócio:** A partir do levantamento junto ao cliente/ usuário, formalizar o processo de negócio em um modelo, por exemplo, utilizando a notação IDEF0, que descreve para cada processo, as entradas, saídas, ferramentas, mecanismos, regras e controles. Tal modelo deve ser validado com o cliente/ usuário e serve de referência para estabelecimento dos itens a serem automatizados em um *software*.

**Passo 2 - Requisitos de Automação:** Com base no modelo de processo de negócio, levantar junto ao cliente/ usuário os requisitos desejados para o *software*. Em geral, o cliente relaciona aspectos funcionais e de negócio, esquecendo-se de outros aspectos relevantes para o processo de desenvolvimento de *software*, denominados requisitos não-funcionais como por exemplo, modularidade, reutilização, segurança e infra-estrutura. Assim, a lista de requisitos deve ser acrescida com estes outros requisitos, utilizando como estratégia as visões de arquitetura de um sistema.

**Passo 3 - Particionamento e Priorização:** Como os tempos envolvidos no desenvolvimento de aplicações Web são pequenos e muitas vezes preestabelecidos, é importante realizar a priorização dos requisitos solicitados pelo cliente, de forma a estabelecer o que será disponibilizado na primeira versão e o que fica para as próximas versões. Esta estratégia exige um bom gerenciamento e negociação com o cliente. Assim, pode-se utilizar o particionamento com base no modelo de processo de negócio, visando disponibilizar para o cliente algo útil e que possa entrar em operação. Partições de negócio englobam um conjunto de requisitos de negócio e podem ser desenvolvidas simultaneamente. As partições devem ser priorizadas pelo cliente.

**Passo 4 - Estabelecimento das Características do Software e Construção da Casa da Qualidade:** Os requisitos do cliente devem ser traduzidos em características técnicas que possam ser tratadas pelas equipes técnicas de projeto. Além disso, é importante saber quais características da arquitetura do *software* devem ser reforçadas durante o projeto. A utilização do SQFD ajuda nesta definição. Considerando a matriz da Casa da Qualidade (Akao, 1990) (Carvalho, 1997), a coluna referente à Voz do Consumidor (VOC) deve ser preenchida com os requisitos de negócio estabelecidos pelo cliente, ou pelas partições, nos casos de projetos com muitos requisitos. Na linha referente às características da qualidade (CQ) devem ser alocados os requisitos técnicos de acordo com as visões de Arquitetura de Sistema. A correlação entre estes itens da Casa da Qualidade permite estabelecer os requisitos de negócio prioritários do ponto de vista do cliente e como tais requisitos impactam e priorizam os aspectos técnicos a serem tratados no desenvolvimento do projeto de *software*.

Como existe uma série de incertezas em relação a estes requisitos e prioridades; a aplicação das técnicas e ferramentas apresentadas pode minimizar estas incertezas. A Figura 2 apresenta as etapas do processo de planejamento de um projeto e as possíveis ferramentas conceituais a serem utilizadas.

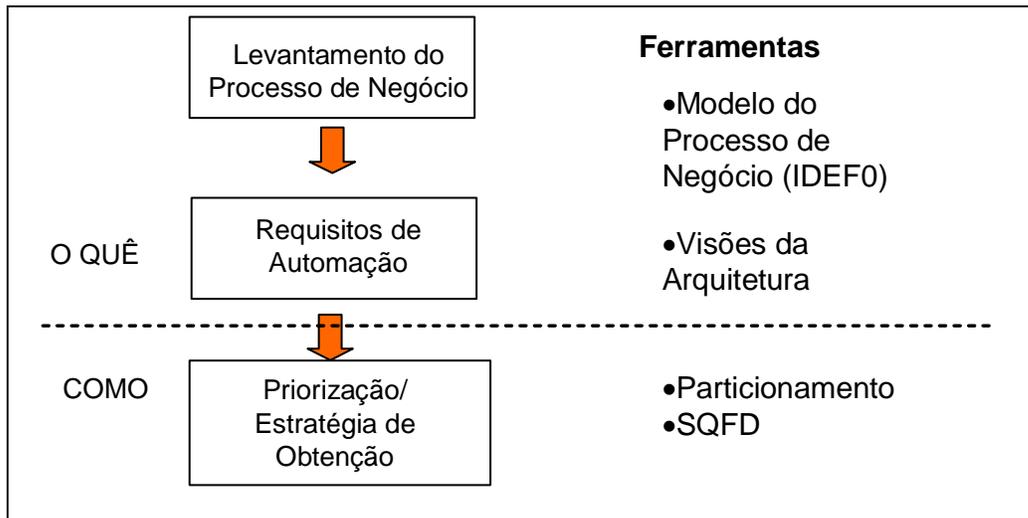


Figura 2: Etapas do Processo de Planejamento e Ferramentas.

#### 4. APLICAÇÃO DO MÉTODO: CASO DO BANCO ELETRÔNICO VIRTUAL

Para demonstração da abordagem proposta, a seguir é apresentado o estudo de um caso real aplicado a uma empresa do setor financeiro, onde o método foi utilizado na definição dos requisitos de um projeto de desenvolvimento de um Banco Eletrônico Virtual. Apenas o passo 4 – Casa da Qualidade não foi aplicado ao caso real, e será objeto de trabalho futuro. A Casa da Qualidade construída e apresentada neste item possui caráter ilustrativo, contribuindo para a análise dos resultados do método.

A Figura 3 apresenta o modelo do processo de negócio, passo 1 do método, para o Banco Eletrônico Virtual, descrito na notação IDEF0.

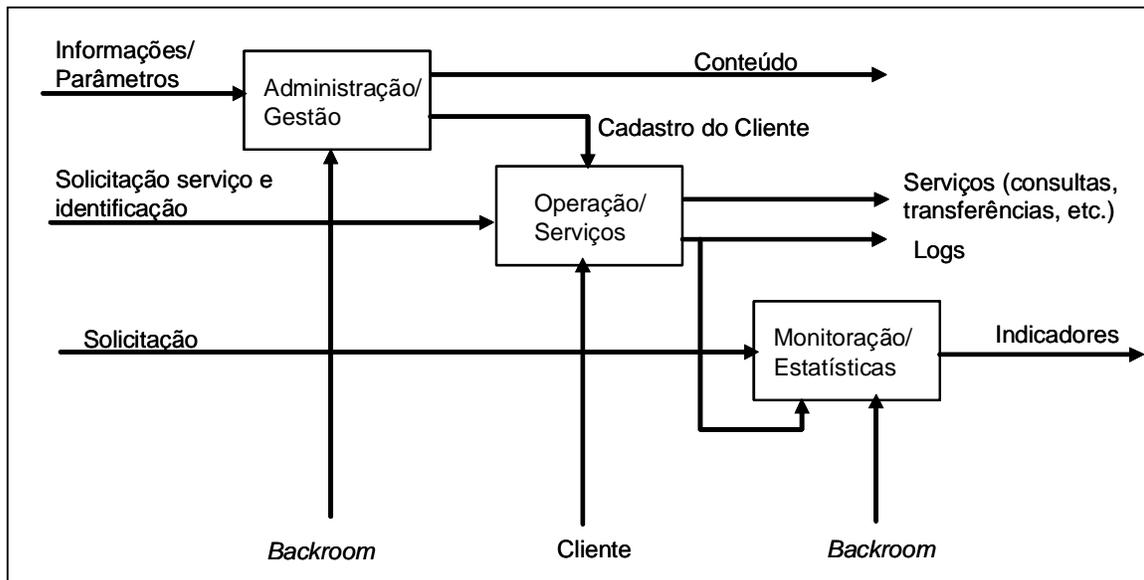


Figura 3: Modelo do Processo de Negócio do Banco Eletrônico Virtual – Notação IDEF0.

Este modelo foi construído em conjunto com os departamentos de negócio e de tecnologia da empresa envolvida no caso em estudo, tomando por base os processos reais do mundo físico e os processos levantados através de *benchmarks* realizados com outros bancos eletrônicos virtuais concorrentes. A Figura 3 apresenta apenas o primeiro nível do diagrama, que foi detalhado em mais dois níveis, mas que não são apresentados neste artigo.

O modelo apresentado, basicamente, compreende os processos de Administração e Gestão do site envolvendo gestão de conteúdo e de informações, cadastramento de clientes e tratamento de ocorrências; Operação e Serviços disponibilizados para os usuários finais, tais como transações financeiras de saldo, extrato, transferência de recursos e pagamento de contas e os processos de Monitoração e Estatísticas, através da disponibilização de indicadores de negócio e infra-estrutura e da extração de relatórios e estatísticas. Participam deste processo de negócio o cliente da instituição financeira (também denominado de usuário final) e o *backroom* (também denominado de *backoffice* ou equipe de retaguarda).

Os requisitos levantados a partir das solicitações dos representantes de negócio e de tecnologia da empresa foram agrupados de acordo com estes grandes processos. O Quadro 2 apresenta os requisitos levantados, compondo a Voz do Consumidor, considerando os pontos de vista do usuário final e do *backroom*.

Tais requisitos foram complementados pela equipe de projeto, de acordo com as cinco visões de Arquitetura, conforme estabelecido no *passo 2 - requisitos de automação*.

Quadro 2: Requisitos de Automação – Solicitações dos Representantes de Negócio e Visões da Arquitetura.

Solicitações dos Representantes de Negócio	Requisitos de acordo com as Visões de Arquitetura
Administração/ Gestão <ul style="list-style-type: none"> <li>- Gestão de conteúdo e informações;</li> <li>- Cadastro de clientes;</li> <li>- Tratamento de ocorrências;</li> </ul> Operação/ Serviços <ul style="list-style-type: none"> <li>- Transações financeiras de saldo, extrato, transferência de recursos e pagamentos;</li> </ul> Monitoração/ Estatísticas <ul style="list-style-type: none"> <li>- Indicadores de negócio e infra-estrutura;</li> <li>- Relatórios e estatísticas (quantidade de acessos, volumes, perfil do cliente);</li> </ul> Facilidade de Uso; Rapidez de Acesso; Tempo de entrega rápido.	Visão Negócio <ul style="list-style-type: none"> <li>- Funcionalidades de negócio (usuário final e <i>backroom</i>);</li> </ul> Visão Segurança <ul style="list-style-type: none"> <li>- Controle de acesso;</li> <li>- Rastreabilidade;</li> <li>- Integridade;</li> </ul> Visão Infra-estrutura <ul style="list-style-type: none"> <li>- Desempenho;</li> <li>- <i>Hardware</i> adequado;</li> </ul> Visão de Desenvolvimento <ul style="list-style-type: none"> <li>- Reuso;</li> <li>- Facilidade de Manutenção;</li> </ul> Visão Operação <ul style="list-style-type: none"> <li>- Monitoração.</li> </ul>

No Quadro 2, a coluna da esquerda representa a Voz do Consumidor e a coluna da direita as Características da Qualidade, o que permite a construção da Casa da Qualidade.

No entanto, como o conjunto de requisitos estabelecidos pelos clientes pode ser muito grande, recomenda-se a utilização do particionamento.

Neste caso, o particionamento proposto de acordo com o modelo do processo de negócio é o seguinte: partição 1: administração/gestão; partição 2: operação/serviços; e partição 3: monitoração/estatísticas.

A Figura 4 apresenta o particionamento a partir do modelo do processo de negócio do Banco Eletrônico Virtual apresentado na Figura 3. A representação gráfica do modelo apóia a identificação das partições, utilizando a notação IDEF0.

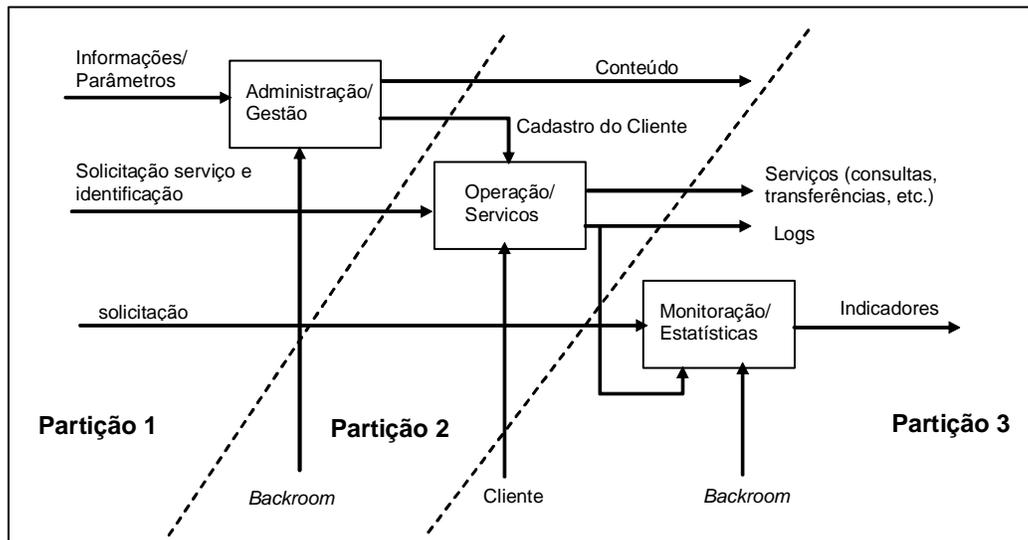


Figura 4: Particionamento com base no modelo do processo de negócio.

De acordo com as prioridades estabelecidas pelos representantes da empresa, a partição prioritária é a 2, pois compreende as transações financeiras. A primeira versão do *software* pode suportar as funcionalidades desta partição, sendo as demais funções executadas manualmente. No entanto, a arquitetura do *software* deve estar preparada para evoluir, considerando à automação do processo completo.

A estratégia de particionamento permite também o desenvolvimento simultâneo, em que para cada partição serão realizadas as etapas de análise, projeto, implementação e testes. Esta estratégia permite também agilizar a entrega do produto de *software* final.

No caso em estudo, a primeira partição implantada foi a 2, seguida das partições 1 e 3 respectivamente. O fato das implantações do banco eletrônico virtual serem realizadas de forma parcial e incremental minimizou as incertezas e ansiedades em relação às entregas do *software*.

Finalmente, a Figura 5 apresenta a Casa da Qualidade, considerando as partições estabelecidas e as características da qualidade.



## 5. ANÁLISE DE RESULTADOS

Os passos 1, 2 e 3 do estudo de caso apresentado garantiram um alinhamento dos requisitos com as necessidades do solicitante, bem como a organização e planejamento para o desenvolvimento simultâneo das partições.

Entre as dificuldades de uso desta abordagem destacam-se:

- Entendimento e levantamento do processo de negócio. Em algumas situações, a empresa não se sente confortável em expor o seu processo de negócio. Nestes casos, a técnica de observação do processo pode ser utilizada para facilitar o levantamento;
- Estabelecimento e gerenciamento das prioridades das partições, considerando um público com interesses diferentes (gestor, usuário final, etc.);
- O passo 4 não foi aplicado em um caso real, mas pode-se vislumbrar que a sua aplicação permite o enfoque dos esforços nos requisitos mais importantes da Arquitetura.

Como trabalhos futuros a serem realizados destacam-se:

- Aplicação do passo 4 da abordagem proposta no artigo (Estabelecimento das Características do *Software* e Construção da Casa da Qualidade) em projetos reais de desenvolvimento de aplicações Web para a indústria financeira;
- Estudo e análise da aplicação do AHP (*Analytic Hierarchy Process*) como instrumento para priorização quantitativa dos requisitos do cliente.

## 6. CONSIDERAÇÕES FINAIS

A abordagem de gestão apresentada neste trabalho incorpora técnicas e ferramentas que permitem minimizar as incertezas e riscos de um projeto de *software*, podendo ser utilizada nas fases iniciais do projeto.

Entre as vantagens desta abordagem, destaca-se: redução dos tempos de entrega, através de entregas parciais, desenvolvimento evolutivo e simultâneo, alinhamento do sistema de *software* com o processo de negócio, esforços direcionados às necessidades dos clientes,

redução dos custos decorrentes de retrabalho e garantia de maior completeza dos requisitos do projeto.

Apesar da abordagem proposta ser utilizada e analisada para o caso de um projeto de aplicação Web, a mesma pode ser utilizada no planejamento de projetos de *software* de outras áreas de aplicação. Assim os principais objetivos deste trabalho podem ser desdobrados em redução de custos, aumento de qualidade e de produtividade no desenvolvimento dos projetos de *software*.

A abordagem apresentada neste trabalho para minimização de riscos e incertezas de projetos de *software* pode ser aplicada de maneira complementar à estratégia de minimização de riscos e incertezas com base em requisitos da Arquitetura de *software* (PINNA, 2004). Estas duas abordagens podem ser utilizadas de maneira complementar, uma mais focada nas fases iniciais do ciclo de vida de desenvolvimento e a segunda aplicada ao ciclo todo.

No entanto, alguns pontos críticos podem ser destacados para utilização destas ferramentas, como a dificuldade de aplicação do QFD quando a quantidade de requisitos for muito grande (matriz fica muito grande). Neste caso, sugere-se trabalhar com partições.

Recomenda-se ainda: o gerenciamento e a negociação com o cliente em relação às entregas parciais devem ser feitos de forma transparente, a diversidade de interesses e de usuários em relação aos requisitos e suas prioridades deve ser tratada cuidadosamente, a visão do todo, da integração e gerenciamento das expectativas, quando se trabalha com particionamento e a postura de gestão forte e constante durante todo o projeto.

## REFERÊNCIAS

AKAO, Y. Quality Function Deployment: Integrating Customer Requirements into Product Design. Portland: Productivity Press, 1990.

ABNT – Associação Brasileira de Normas Técnicas - NBR 13596 (ISO 9126 ) – Tecnologia de Informação: Avaliação de Produto de Software – Características de Qualidade e Diretrizes para o seu Uso, 1997.

BROWN, A. W. Large-Scale, Component-Based Development. Prentice Hall PTR, 2000.

CARVALHO, M. M; LAURINDO, F.J.B.; PESSÔA, M.S.P. Information Technology Project Management to achieve efficiency in Brazilian Companies. In: KAMEL, Sherif. (Org.). Managing Globally with Information Technology. Hershey, pp.260-271, 2003.

CARVALHO, M.M.; RABECHINI JR.,R. Construindo Competências para Gerenciar Projetos: teoria e casos. São Paulo: Editora Atlas, 2005.

CARVALHO, M. M. QFD: uma ferramenta de tomada de decisão em projeto. 1997. Tese de Doutorado - Departamento de Engenharia de Produção e Sistema, Universidade Federal de Santa Catarina, Florianópolis, 1997.

KRUCHTEN, P.; *The Rational Unified Process: An Introduction*, Reading, MA: Addison-Wesley, 1998.

IDEF0 - Integration Definition For Function Modeling Federal Information Processing Standards Publication (FIPS Std 183), Dezembro, 1993 - URL [www.idef.com/Downloads/free\\_downloads.html](http://www.idef.com/Downloads/free_downloads.html).

MARCA, D. A.; McGOWAN, C. L. IDEF0/SADT: Business Process and Enterprise Modeling. Califórnia: Eclectic Solutions, 1988.

McCONNELL, S. Rapid Development: Taming wild software schedules. Washington: Microsoft Press, 1996.

PINNA, C. C. A.: Um Roteiro centrado em Arquitetura para Minimização de Riscos e Incertezas em Projetos de Software. 2004. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.

PMI. PROJECT MANAGEMENT INSTITUTE. A Guide to the Project Management Body of Knowledge (PMBOK). 3rd edition. Project Management Institute Inc., 2004.

PRESSMAN, R. S. Software Engineering – A Practitioner’s Approach. 6 ed. New York: McGraw-Hill, 2005.

SPINOLA, M. M.; CARDOSO, L. A. Aplicação de QFD para a Especificação de um Sistema de Informações. In: 2 Congresso Brasileiro de Gestão de Desenvolvimento de Produto, 2000, São Carlos, 2000.

Artigo recebido em 12/04/2007 e aceito para publicação em 01/03/2008