

Gerência de Projeto Baseado em Design Orientado a Encapsulamento de Dados e a Troca de Mensagens entre Subsistemas Autônomos

Maria Luiza d'Almeida Sanchez
Depto. de Eng. Telecomunicações- UFF
Depto de Informática - PUC - Rio

Bruno Maffeo*
Depto de Informática - PUC - Rio

*Trabalho parcialmente financiado pelo convênio entre o Centro Técnico-Científico da PUC-Rio e o Ministério da Ciência e Tecnologia.

Resumo

Este trabalho apresenta uma proposta para gerenciar projetos de desenvolvimento de sistemas de tempo-real e de grande porte. Focaliza, primordialmente, a construção do software e a administração a ser efetuada em cada fase do projeto. Como hipótese básica, reconhece a necessidade de mapear na estrutura do processo de gerenciamento a estrutura do produto a ser gerado. Também, pressupõe o emprego de Métodos Estruturados, sendo a fase de Design orientada a Encapsulamento de Dados e a Troca de Mensagens entre Subsistemas Autônomos. Essa abordagem favorece o Reuso de subsistemas anteriormente desenvolvidos e a estratégia de Desenvolvimento Incremental.

Abstract

This work presents a proposal for the project management of large real-time systems development processes. The emphasis is on software construction and phase administration. As a fundamental hypothesis, it recognizes the necessity of establishing a mapping of the structure of the product to be constructed onto the structure of the managerial process. It also supposes the application of Structured Methods, the Design phase being oriented to Information Hiding and Exchange of Messages between Independent Subsystems. This approach facilitates the Reuse of subsystems previously developed and the Incremental Development strategy.

1- Introdução

Este trabalho apresenta uma proposta gerencial que aborda os aspectos mais relevantes envolvidos no planejamento, controle e execução das fases de um projeto de desenvolvimento de sistemas. Busca evidenciar uma administração de projeto que vise assegurar qualidade de software obtida por construção, minimizar custos e maximizar a produtividade do processo de desenvolvimento.

A implantação de um método de desenvolvimento adequado às características do sistema a ser construído é fundamental para garantir o sucesso do projeto. No entanto, para implantar um método e utilizá-lo de forma correta e eficiente, é necessário conceber a administração (planejamento, controle, organização e liderança) do projeto de modo que sua estrutura resulte de um mapeamento a partir da estrutura conceitual do produto a ser gerado.

Um processo de desenvolvimento de sistemas compreende um conjunto de atividades que objetiva, como resultado, a obtenção de um sistema atendendo a requisitos especificados. Esse é um processo iterativo de busca de alternativas pois, a cada alternativa explorada, as mesmas atividades são repetidas, até que uma solução satisfatória seja encontrada. Nessas condições, a estrutura do gerenciamento de um projeto deve sobrepor-se a um processo com características sequenciais e paralelas, com marcos bem definidos a intervalos de tempo previsíveis, visando assegurar que a totalidade do processo de desenvolvimento fique contida em prazos e recursos preestabelecidos, garantida a qualidade do produto final.

Nossa preocupação principal refere-se ao desenvolvimento de sistemas de tempo-real de grande porte, cuja complexidade intrínseca é geralmente superior à de sistemas de mesmo porte de outra classe. Comumente, o projeto de tais sistemas consome tempo significativo até sua implantação (cerca de 3 a 8 anos) e envolve equipes grandes (com mais de 15 pessoas). Os procedimentos para garantir uma troca eficiente de informação entre membros de uma dada equipe, e entre equipes, bem como para estabelecer o acompanhamento das modificações introduzidas no período de desenvolvimento, são complexos e necessitam de métodos rígidos de

controle. Esse controle será tanto mais facilmente realizado quanto mais realisticamente estruturado for o gerenciamento do projeto: a segmentação de atividades deve corresponder à estrutura do produto a ser gerado, permitindo estabelecer marcos de controle significativos e proporcionar visibilidade à evolução do projeto [13]. Este trabalho define marcos que permitem o mapeamento da estrutura conceitual do sistema para o processo gerencial, baseando-se em princípios que fundamentam os chamados Métodos Estruturados. Esses métodos segmentam a representação do sistema em três modelos [13,22,24]: Essência, Implementação e Automação. Para o Modelo da Implementação, é pressuposto o uso da técnica de Design Orientado a Encapsulamento de Dados e Troca de Mensagens entre Subsistemas Autônomos [19].

Na seção 2, é apresentado um resumo dos conceitos subjacentes à estrutura de modelagem preconizada à implantação de Reuso.

Na seção 3, define-se uma proposta para o conjunto de atividades a serem executadas durante o desenvolvimento de um sistema, os pontos de controle e os modelos resultantes de cada atividade, e aborda-se a estratégia de desenvolvimento incremental.

Na seção 4, focaliza-se a organização das equipes do projeto, os aspectos de liderança envolvidos e a dinâmica de sua interação no decorrer do processo de desenvolvimento.

A seção 5 aborda os aspectos relativos ao controle (cumprimento de requisitos e qualidade assegurada por construção).

A seção 6 contém a conclusão deste trabalho.

2 - O Processo de Desenvolvimento de um Sistema

2.1 - Métodos Estruturados

Tradicionalmente, os chamados **Métodos Estruturados** têm sido orientados a atividades, buscando primordialmente a identificação das funções que constituem o sistema. Modernamente, incorporam a modelagem da informação armazenada [13,22,24].

A estrutura de modelagem consiste de três modelos, com graus de abstração diferentes, que detalham toda a concepção e implementação do sistema. Existe uma relação de precedência lógica entre esses modelos que, entretanto, não define necessariamente a ordem cronológica mais adequada para sua construção. São eles:

- **Modelo da Essência (ME)**, representando o resultado da elicitação, análise e especificação dos requisitos essenciais (isto é, independentes de qualquer alternativa de implementação) do sistema em questão. Neste trabalho, não serão apresentados detalhes referentes a esse modelo, que podem ser encontrados, por exemplo, em [13];
- **Modelo da Implementação (MI)**, sucessor lógico do ME incorporando os requisitos essenciais ao Design do sistema, é caracterizado sucintamente ao longo do trabalho e discutido mais detalhadamente, por exemplo, em [22,4].
- **Modelo da Automação (MA)**, sucessor lógico do MI incorporando a alternativa de implementação selecionada, representa o sistema em termos de seus recursos de mais baixo nível de abstração (código, processadores, ...).

Cada modelo engloba o conteúdo do modelo que o precede logicamente.

2.2 - Modelo da Implementação

Neste trabalho, o Modelo da Implementação é segmentado em dois submodelos: o **Projeto Básico**, que apresenta, em diversos níveis de abstração, a decomposição do sistema até a determinação de um conjunto de subsistemas com características de **Subsistemas Básicos**, e a **Especificação Detalhada** de cada **Subsistema Básico**. O **Projeto Básico** permite visualizar esquematicamente o funcionamento do sistema como um todo e as opções de Design que levam a um dado particionamento do sistema em subsistemas.

Os **Subsistemas Básicos** são **Unidades de Implementação** e **Unidades de Reutilização**. Cada **Subsistema Básico** é uma unidade de código fonte, indivisível quando reutilizada, desenvolvida e mantida por uma equipe pequena (no máximo 3 pessoas). A separação entre a documentação do Design do sistema como um todo e a documentação da **Especificação Detalhada** de cada **Subsistema Básico** facilita a divisão do trabalho em equipes, cada uma responsável por um ou mais **Subsistemas Básicos**, bem como a identificação dos **Subsistemas Básicos**, já desenvolvidos, que podem ser reutilizados.

Um Subsistema Básico é caracterizado por ser, alternativamente:

- **Responsável por objeto físico:** por exemplo, a um dispositivo de hardware é associado um Subsistema Básico, que o controla e o apresenta ao restante do sistema com um conjunto de operações necessárias a seu uso (Dispositivo Inteligente);
- **Responsável por estrutura de dados:** orientando a implementação para encapsulamento de dados, o que permite homogeneizar a segmentação do sistema, já que um dispositivo de hardware pode ser entendido como uma estrutura de dados encapsulada em uma peça de hardware;
- **Responsável por funções do sistema:** quando não associadas diretamente a objetos físicos ou a estruturas de dados, funções relacionadas segundo algum critério sistêmico devem ser reunidas em um único Subsistema Básico;
- **Responsável por relacionamento de funções:** estes são os Subsistemas Básicos que coordenam a execução e essa coordenação determina a forma de operação do sistema.

As duas últimas categorias são necessárias para permitir a segmentação de porções não reusáveis, presentes nos serviços de um subsistema. Porções reusáveis, quando não são usadas em conjunto, também devem pertencer a Subsistemas Básicos diferentes.

A concepção de Subsistemas Básicos permite um alto grau de Reuso. Esses subsistemas têm apenas conhecimento de si próprio, isto é, não apresentam referências a outros subsistemas. Visando obter essa independência, a comunicação entre subsistemas é estabelecida por **envio/recebimento de mensagens** através de **Portas** de dados que são parte de cada subsistema. O sistema e os subsistemas que o compõem são definidos como prestadores de serviços. A cada subsistema está alocada uma fração dos serviços prestados pelo sistema, muitas vezes de forma concorrente. Um **serviço concorrente** está associado a um **Processo** de execução periódica ou ativado por um **Pedido de Serviço** - recebimento de uma mensagem através de uma Porta de entrada. Essas mensagens são transportadas por **Interfaces** que conectam **Portas de Subsistemas**.

Subsistemas prontos e testados precisam ser integrados, formando o sistema total. Para isso, as **Interfaces** devem ser implementadas e os **Processos** que compõem cada **Subsistema Básico** precisam ser mapeados na **Arquitetura de Hardware - Configuração do Sistema**. Para facilitar a migração para novas arquiteturas de hardware e software básico (Portatibilidade), a **Configuração do Sistema** não deve ser incorporada a cada **Subsistema Básico**; deve ser feita através de uma **Linguagem de Configuração** que permita modelar as **Interfaces** dos subsistemas, verificar a consistência das **Portas** conectadas (**Linguagem de Configuração de Sistemas**) e mapear os **Processos** de cada **Subsistema Básico** na **Arquitetura de Hardware (Linguagem de Configuração de Hardware)**. A **Configuração do Sistema** constitui um módulo de software independente, o que facilita alterações de **Arquitetura de Hardware e Software**.

2.3 - A Implantação de Reuso

Após o desenvolvimento de alguns sistemas segundo este método, a Área de Informática da organização disporá de uma biblioteca significativa de Subsistemas Básicos reusáveis. Durante a concepção do Projeto Básico, deve ser feita uma pesquisa nessa biblioteca, visando determinar os Subsistemas Básicos existentes cujas características, obtidas a partir de suas Especificações, sejam compatíveis com as do sistema em desenvolvimento. A decomposição do sistema em subsistemas deve ser feita com esse conhecimento prévio e direcionada para o aproveitamento desses Subsistemas Básicos. Um sistema pode até vir a ser construído totalmente pela conexão de Subsistemas Básicos existentes, como hoje em dia constroem-se módulos de hardware pela interligação de componentes integrados disponíveis no mercado.

Para a implantação da política de reutilização, é necessário que o método de desenvolvimento esteja integrado a uma política administrativa que:

- gerencie a implementação de Subsistemas Básicos reusáveis, visando garantir o respeito a métricas de Reuso [16], tais como: ser o mais genérico possível, não empregar variáveis globais, ser parametrizável, possuir boa documentação etc.;
- dê visibilidade ao conjunto dos Subsistemas Básicos já existentes e efetive sua aplicação em sistemas novos. A identificação de subsistemas que mais se adaptem a um ou outro requisito não é tarefa fácil. Suporte automatizado, como um sistema de biblioteca baseado em conhecimento [16], pode ajudar;

- gerencie as diversas versões de Subsistemas Básicos e os sistemas nos quais estão empregadas. Alteração em um desses subsistemas deve, pelo menos, implicar sinalização para as equipes responsáveis pela manutenção dos outros sistemas que o utilizam.

3 - Atividades do Processo de Desenvolvimento

O Ciclo de Vida de um sistema é conceito bastante polêmico [13] e existem várias propostas para organizar as fases básicas do processo de desenvolvimento. Entretanto, independentemente de ordem cronológica, existem atividades que devem sempre ser executadas e isso deve ocorrer segundo as diretrizes do método escolhido para o desenvolvimento. Algumas dessas atividades apresentam precedência lógica em relação a outras. No entanto, uma atividade ainda incompleta pode produzir um modelo parcial que permita o início da execução de outra atividade da qual seja prerequisite. Essa constatação tem grande valor do ponto de vista gerencial pelos ganhos de produtividade que podem ser auferidos pelo paralelismo a ser utilizado na execução de atividades logicamente seqüenciais.

Dentro do método proposto neste trabalho, um projeto compõe-se basicamente das seguintes atividades:

- Modelagem da Essência (composta de Modelagem do Contexto - Definição do Problema e Modelagem do Comportamento - Solução Abstrata em relação a qualquer alternativa de implementação) correspondendo à especificação dos requisitos essenciais: funcionais e informacionais;
- Modelagem da Implementação;
- Especificação dos Requisitos de Concepção do Sistema;
- Especificação da Interface Usuário - Máquina;
- Projeto Básico de Software e Hardware;
- Projeto de Subsistemas Básicos e Módulos de Hardware;
- Modelagem da Automação
- Desenvolvimento dos Subsistemas Básicos e do Hardware correspondente (incluindo design, implementação e teste isolado de cada Subsistema Básico);
- Integração e Testes.

A precedência lógica dessas atividades é apresentada na Figura 1. No entanto, essas atividades não são executadas de forma necessariamente seqüencial; ao longo do tempo, pode ocorrer paralelismo. O sistema gradualmente toma forma quanto mais se aprende sobre o problema e tenta-se resolvê-lo de forma abstrata (Modelagem da Essência) e quanto mais alternativas de implementação são exploradas (Modelagem da Implementação). Construções parciais válidas desses modelos de alto nível de abstração permitem incremento gradual de funcionalidade até ser atingida a totalidade dos requisitos desejados.

3.1 - O Resultado de Cada Atividade

Uma atividade é caracterizada pelos produtos que gera. Esses produtos são modelos do sistema em níveis de abstração diferenciados em relação ao uso de recursos de implementação. O resultado de cada atividade deve ser:

- usado como marco de fim da atividade;
- associado a recursos e prazos bem definidos;
- revisado, de modo que sua qualidade seja assegurada por construção;
- usado para proporcionar visibilidade quanto ao ponto em que se encontra o projeto no caminho percorrido durante o desenvolvimento do sistema.

3.1.1.- Estudo de Viabilidade de Custo x Benefício x Importância

No início, o projeto baseia-se em uma solicitação de serviço com a definição preliminar do escopo [6,24]. Isso pode dar origem a uma Lista de Eventos Externos, também preliminar, e a uma visão global dos Requisitos de Concepção de Sistema mais importantes e críticos. Esses modelos iniciais constituem uma definição concisa do problema sobre a qual serão estabelecidos objetivos (qualitativos e quantitativos) a serem atingidos; por exemplo: tecnologia a empregar, custos e prazos. O estabelecimento desses objetivos resulta de uma análise de viabilidade econômica e de custo x benefício x importância, o que deve corresponder a cerca de 5% a 10% dos recursos e prazos totais do projeto [24]. É uma atividade fundamental pois permite decidir sobre o prosseguimento do projeto.

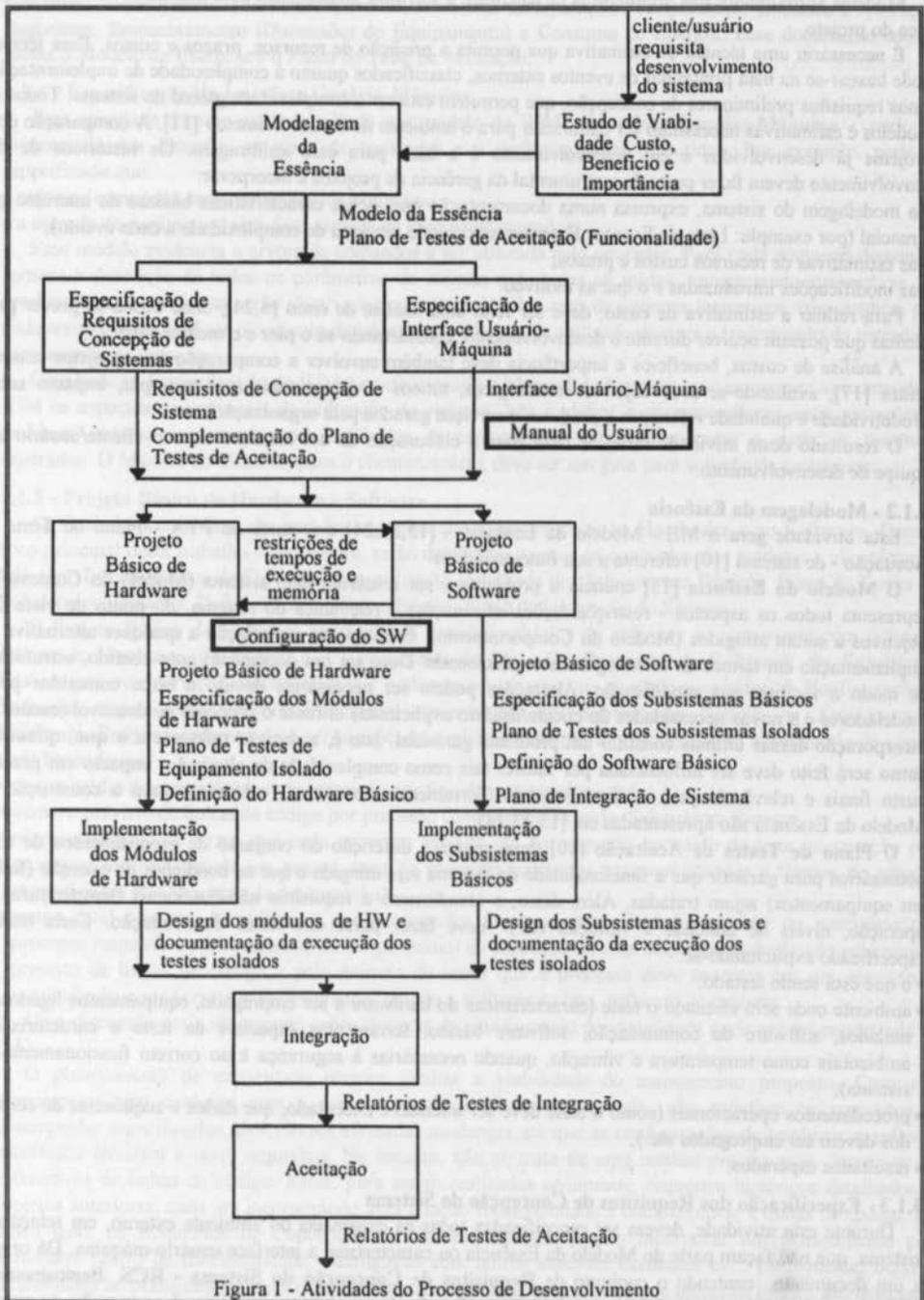


Figura 1 - Atividades do Processo de Desenvolvimento

Modelos aproximados das arquiteturas de hardware e software possibilitam uma análise de viabilidade técnica do projeto.

É necessária uma técnica de estimativa que permita a predição de recursos, prazos e custos. Essa técnica pode basear-se na lista preliminar de eventos externos, classificados quanto à complexidade de implementação, e nos requisitos preliminares de concepção, que permitem estimar a complexidade global do sistema. Todos os modelos e estimativas necessitam ser calibrados para o ambiente de desenvolvimento [11]. A comparação com projetos já desenvolvidos e em desenvolvimento é a base para essa calibragem. Os históricos de desenvolvimento devem fazer parte do instrumental da gerência de projetos e incorporar:

- a modelagem do sistema, expressa numa documentação que inclua características básicas de interesse gerencial (por exemplo: Lista de Eventos Externos associando um grau de complexidade a cada evento);
- as estimativas de recursos custos e prazos;
- as modificações introduzidas e o que as motivou.

Para refinar a estimativa de custo, deve ser feita uma análise de risco [6,24], onde busca-se prever problemas que possam ocorrer durante o desenvolvimento, considerando-se o pior e o melhor caso.

A análise de custos, benefícios e importância deve também envolver a comparação com projetos concorrentes [17], avaliando-se seus objetivos estratégicos, táticos e operacionais (por exemplo, impacto sobre produtividade e qualidade referentes a produtos e serviços gerados pela organização).

O resultado desta atividade serve de base para a elaboração de um contrato entre o cliente/usuário e a equipe de desenvolvimento.

3.1.2 - Modelagem da Essência

Esta atividade gera o ME - Modelo da Essência - [13,22,24] e a parte do PTA - Plano de Teste de Aceitação - de sistema [10] referente a sua funcionalidade.

O **Modelo da Essência** [13] enuncia o problema a ser resolvido pelo sistema (Modelo do Contexto) e representa todos os aspectos - restrições/ações/informações - relevantes do sistema, do ponto de vista dos objetivos a serem atingidos (Modelo do Comportamento). É invariante em relação a qualquer alternativa de implementação em termos da tecnologia a ser selecionada. Deve ser um documento auto-contido, estruturado de modo a facilitar sua modificação. Alterações podem ser necessárias devido a erros cometidos pelos modeladores e a novas necessidades do cliente/usuário explicitadas durante o processo de desenvolvimento. A incorporação dessas últimas constitui um problema gerencial. Isto é, a decisão referente a **o que, quando e como** será feito deve ser influenciada por fatores tais como complexidade da alteração, impacto em prazo e custo finais e relevância para o cliente/usuário. Ferramentas conceituais e técnicas para a construção do Modelo da Essência são apresentadas em [13,22,24].

O **Plano de Testes de Aceitação** [10] deve conter a descrição do conjunto de procedimentos de teste necessários para garantir que a funcionalidade do sistema seja atingida e que as condições de exceção (falhas em equipamentos) sejam tratadas. Além disso, o atendimento a requisitos não-funcionais (temperaturas de operação, níveis de umidade e vibração etc.) deve fazer parte dos testes de aceitação. Cada teste é especificado explicitando-se:

- o que está sendo testado;
- ambiente onde será efetuado o teste (características do hardware a ser empregado, equipamentos ligados/simulados, software de comunicação, software básico, ferramentas especiais de teste e características ambientais como temperatura e vibração, quando necessárias à segurança e ao correto funcionamento do sistema);
- procedimentos operacionais (como o teste deve ser iniciado e executado, que dados e seqüências de comandos devem ser empregados etc.);
- resultados esperados.

3.1.3 - Especificação dos Requisitos de Concepção do Sistema

Durante esta atividade, devem ser especificadas todas as exigências do ambiente externo, em relação ao sistema, que não façam parte do Modelo da Essência ou caracterizem a interface usuário-máquina. Dá origem a um documento contendo o conjunto de **Requisitos de Concepção do Sistema - RCS**. Basicamente, a estrutura desse documento [19,5,9] permite a especificação de requisitos necessários à concepção do sistema

tais como: Desempenho, Tolerância a Falhas, Ambiente de Operação (Temperatura, Umidade e Vibração), Segurança, Empacotamento (Dimensões do Equipamento) e Consumo de Energia. Esse documento complementa o Modelo da Essência e o Plano de Teste de Aceitação.

3.1.4 - Especificação da Interface Usuário-Máquina

Essa atividade fornece como resultado um modelo da IUM - Interface Usuário-Máquina -, onde fica evidenciada sua filosofia, os tipos de comandos e a dinâmica neles embutidas. Por exemplo, pode ser especificado que:

- todos os comandos serão selecionados por menu;
- a entrada de parâmetros será feita através de telas preformatadas.

Esse modelo evidencia a árvore de comandos a ser utilizada pelo operador. Para cada comando disponível, fornece a descrição de todos os parâmetros de entrada e de todas as saídas, apresentando sua sintaxe, sua semântica e em que forma serão observados pelo operador. No caso de sistemas interativos, isso implica, para cada comando, especificar o conjunto de telas de entrada e saída utilizado durante a transmissão da entrada e a correspondente resposta.

Com base na Especificação da IUM é confeccionado o **Manual do Usuário**, que acrescenta ao conteúdo da IUM os aspectos operacionais de utilização do sistema. Isto é, define a correspondência entre situações do ambiente externo e comandos ou seqüências de comandos a serem empregados e quais os resultados esperados. O Manual do Usuário, para o cliente/usuário, deve ser um guia para solução de problemas.

3.1.5 - Projeto Básico de Hardware e Software

Ao final desta atividade são gerados modelos que dizem respeito ao **Hardware** e ao **Software**. Como o foco principal deste trabalho é o software, serão detalhados apenas os conteúdos dos modelos a ele referentes. São eles: Projeto Básico de Software, Especificação Detalhada dos Subsistemas Básicos, Plano de Testes dos Subsistemas Isolados, Definição do Software Básico, Plano de Integração de Sistema.

O Projeto Básico de Software apresenta os grandes módulos de software e, juntamente com o Projeto Básico de Hardware, dá origem a um documento de CS - **Configuração do Software**. Os Subsistemas Básicos possuem serviços concorrentes que podem executar em mais de um processador. A Configuração do Software mapeia essas frações (processos) concorrentes dos Subsistemas Básicos na Arquitetura de Hardware. Esse mapeamento deve ser justificado explicitamente por um planejamento aproximado de capacidade baseado em:

- tempo médio de execução de uma instrução por tipo de processador usado;
- número previsto de linhas de código por processo concorrente em cada Subsistema Básico;
- número máximo de vezes que cada processo executa em um segundo (extraído de taxa interrupção para processar informações de um sensor, taxa de envio de comandos para um atuador e tempos de resposta especificados nos requisitos de desempenho);
- tempo total previsto de utilização por processador por segundo, calculado a partir do somatório (por processo mapeado no processador) do tempo médio de execução de uma instrução multiplicado pelo número previsto de linhas de código e pelo número de vezes que o processo deve executar em um segundo. O resultado deve ser acrescido do "overhead" devido a software básico, software de rede etc..

Uma discussão mais abrangente sobre técnicas de planejamento de capacidade pode ser encontrada em [15].

O planejamento de capacidade permite avaliar a viabilidade do mapeamento proposto. Caso esse mapeamento, em conjunto com a arquitetura de hardware selecionada, não satisfaça os requisitos de desempenho especificados, deverão ser efetuadas mudanças até que as configurações de hardware e software escolhidas atendam a esses requisitos. No entanto, não se trata de uma análise precisa pois baseia-se em estimativas de linhas de código. Estas, para serem realizadas seriamente, requerem históricos detalhados de projetos anteriores, cada um incorporando uma relação funcionalidade x complexidade x linhas de código. Por outro lado, os Requisitos de Concepção do Sistema também podem incorporar um planejamento para acréscimo futuro de funcionalidade, interligação com outros sistemas existentes etc.. Comumente, ocorrem dificuldades no planejamento para expansão do sistema e falhas na previsão de linhas de código são detectadas nas fases de implementação e testes. Assim, o planejamento de capacidade deve ser complementado por uma

análise de alternativas futuras de otimização ou de expansão; esse estudo deve abranger as configurações de hardware e software e permitir a superação de problemas de desempenho ou o acréscimo de funcionalidade.

O **Projeto Básico de Software** contém a definição de um conjunto de Subsistemas Básicos e de sua forma de interação. O Modelo resultante apresenta a estrutura de alto nível do software e a relação de concorrência e seqüencialização entre os elementos que compõem essa estrutura - comportamento do sistema. É expresso em termos de uma linguagem diagramática hierárquica onde são modeladas as **Operações** executadas pelo sistema, através de seus subsistemas, os **Dados** processados/produzidos por esses subsistemas, que trafegam nas mensagens, e as **Relações de Precedência** na execução das operações.

Para que a implementação seja realizada por equipes diferentes, cada Subsistema Básico necessita de uma **Especificação Detalhada** independente. Isto é, o conjunto de Serviços contido no Projeto Básico não está ali descrito com o nível de detalhe necessário à implementação. Entretanto, a partir do Projeto Básico e do Modelo da Essência, pode ser realizada essa especificação. A cada Especificação Detalhada deve estar associado um **Plano Isolado de Testes**. Em linhas gerais, esse plano deve seguir a mesma estrutura do Plano de Testes de Aceitação de Sistema. Geralmente, nesta fase, é necessária a simulação de diversos subsistemas para que o teste possa ser executado. Assim sendo, será necessária a implementação de um Programa de Teste, que deve ter sua especificação e documentação (diagrama de estrutura e fonte comentado) anexadas ao Plano de Testes.

De posse do Projeto Básico, o Planejamento define o processo de desenvolvimento, que deve refletir uma estratégia incremental onde Subsistemas Básicos são escolhidos para geração de versões parciais do software. O **Plano de Integração** resultante determina a integração parcial do sistema e a ordem em que os Subsistemas Básicos serão especificados e implementados.

3.1.6 - Implementação dos Subsistemas Básicos

Gerada a Especificação Detalhada de um Subsistema Básico, passa-se a dispor da planta a ser usada para sua implementação. Devem ser empregadas ferramentas e técnicas análogas às usadas na construção do Projeto Básico, evitando-se descontinuidades no processo de desenvolvimento.

3.1.7 - Integração do Sistema

A atividade de integração do sistema é executada sempre que estiver concluído um conjunto de Subsistemas Básicos (testados a partir de seu Plano de Teste Isolado) que componha uma versão do sistema. Ela dá origem ao **Relatório de Testes de Integração** (um por integração realizada) que enumera os problemas detectados na integração, as soluções encontradas e o impacto dessas soluções em modelos já gerados.

3.1.8 - Aceitação do Sistema

A aceitação do sistema será efetuada incrementalmente (no caso de entrega de versões parciais operacionais intermediárias) e no final do processo de desenvolvimento. Constitui a aceitação formal, pelo cliente/usuário, do sistema desenvolvido (ou da versão parcial entregue). A execução desta atividade baseia-se no Plano de Testes de Aceitação, aprovado formalmente pelo cliente/usuário. Os resultados devem ser registrados em **Relatório de Testes de Aceitação** onde são registradas as pendências eventualmente existentes, a serem resolvidas pela equipe de desenvolvimento para que o sistema possa ser finalmente aceito.

3.2 - A Estratégia de Desenvolvimento Incremental

A opção por uma estratégia de Desenvolvimento Incremental é decisão de cunho estritamente gerencial e visa a apresentação de produtos operacionais ao cliente/usuário em prazos muito menores do que o tempo necessário ao desenvolvimento do sistema completo.

Essa estratégia determina quais versões parciais do sistema serão entregues ao cliente/usuário e em que ordem. Essas versões serão entregues como produtos válidos para uso produtivo. Isso traz resultados políticos importantes:

- consegue-se maior comprometimento do cliente/usuário com a continuidade do projeto. Cada versão operacional em uso permite-lhe fazer críticas e requisitar adaptações que transformem o produto final naquele mais adequado a suas necessidades, garantindo-se desta forma sua satisfação ao fim do projeto;

• o cliente/usuário não perderá a confiança na equipe de desenvolvimento pois receberá resultados intermediários, parciais porém operacionais, em prazo curto, o que lhe permitirá observar o retorno do capital investido bem antes da conclusão do projeto.

Da Lista de Eventos Externos do Modelo da Essência, é possível selecionar cada subconjunto de eventos que dará origem a uma versão operacional de parte do sistema. Por relacionar facilmente a totalidade das necessidades a serem atendidas pelo sistema com cada parte bem definida dessa totalidade, essa visão de alto nível de abstração permite explicitar para o cliente/usuário, em linguagem que lhe é familiar, como está ocorrendo a evolução do projeto no sentido de sua conclusão.

No **Projeto Básico**, mais uma vez, o sistema como um todo precisa ser levado em conta, isto é, a Lista de Eventos Externos deve ser completamente mapeada em serviços prestados. Com base nesse mapeamento, deve ser proposta a arquitetura de Subsistemas Básicos. Essa visão sistêmica global deve garantir uma segmentação adequada do sistema em Subsistemas Básicos.

Existem Subsistemas Básicos que podem ser identificados desde o início da concepção do sistema. Muitos representam áreas de risco para o desenvolvimento. Neste caso, é recomendável iniciar sua Especificação Detalhada e seu Design o mais rápido possível (eventualmente, em paralelo com a modelagem da essência) pois, visando diminuir o risco associado ao desenvolvimento, precisa ser eliminado o perigo de erros no seu entendimento, de falhas na sua especificação e de incapacidade técnica para gerar sua implementação.

Após o Projeto Básico, a implementação de Subsistemas Básicos deve ser executada de acordo com a ordem prescrita na estratégia de desenvolvimento. Para aqueles subsistemas com maior prioridade, a **Especificação Detalhada** terá sido feita mais cedo e, em seguida, será iniciada sua implementação. Os subsistemas prontos serão integrados gradativamente, gerando versões operacionais parciais. A estratégia de desenvolvimento pode determinar a implementação imediata de alguns dos serviços prestados por um dado Subsistema Básico enquanto que outros serão implementados posteriormente. Neste caso, antes de ser iniciada a implementação da parte de maior interesse, deve-se dispor da Especificação Detalhada completa desse Subsistema Básico.

4 - Organização das Equipes

Para cada atividade, o perfil profissional de cada membro da equipe deve ser determinado em função do tipo de atividade e o número de profissionais da equipe deve ser determinado em função da quantidade de trabalho e do grau de paralelismo que se deseje imprimir em sua execução. No entanto, deve ser ressaltado que, em equipes numerosas, a produtividade é, em geral, fortemente inibida por problemas de comunicação [21]. Neste trabalho, a proposta de estrutura de equipe visa otimizar os canais de comunicação, empregando uma hierarquia que facilite a interação e o acompanhamento.

4.1 - Equipes x Fases do Projeto

As atividades iniciais (Modelo da Essência, Especificação de RCS, Especificação da IUM e Projeto Básico) do processo de desenvolvimento necessitam de um número menor de pessoas envolvidas. No entanto, deve ser um grupo experiente, que manterá o comando do projeto por todo o desenvolvimento. Este grupo será aqui chamado de **Grupo de Sistemas**. Deve ser constituído por **Engenheiros de Software**, **Engenheiros de Hardware** e **Engenheiros de Sistemas** (provenientes da área usuária [12]) que possuam visão ampla de todo o sistema, exercendo as seguintes funções:

- **Engenheiros de Software (na função de Analistas de Sistemas)**, em conjunto com **Engenheiros de Sistemas (na função de Representantes da Área Usuária)** serão capazes de perceber o sistema em desenvolvimento como um conjunto {software, hardware} e detectar, junto ao cliente/usuário, as necessidades a serem atendidas. Esses profissionais serão os responsáveis pela definição da funcionalidade do sistema e pela definição das condições de exceção que possam ocorrer no ambiente externo. Eles deverão construir o Modelo da Essência e definir os Requisitos de Concepção do Sistema.
- **Engenheiros de Software (na função de Projetistas de Interface)**, em conjunto com **Engenheiros de Sistemas (na função de Representantes da Área Usuária)**, a partir do Modelo da Essência, deverão especificar a Interface Usuário-Máquina e reponsabilizar-se pela elaboração do Manual de Usuário.
- **Engenheiros de Software (na função de Projetistas de Sistemas)**, a partir do Modelo da Essência, deverão elaborar o Projeto Básico de Software. Em virtude da independência do Projeto Básico de Software em

relação à escolha do hardware [19], a Arquitetura de Hardware poderá ser definida em paralelo por **Engenheiros de Hardware**.

- **Engenheiros de Software (na função de Projetistas de Sistemas)**, em conjunto com **Engenheiros de Hardware**, baseando-se no Projeto Básico de Software, nas alternativas possíveis para a Arquitetura de Hardware, nos Requisitos de Concepção do Sistema e nos tempos de execução previstos para cada Subsistema Básico, deverão selecionar uma Arquitetura de Hardware e de Software Básico, bem como executarão o mapeamento dos Subsistemas Básicos na Arquitetura de Hardware - **Configuração do Software**.
- **Engenheiros de Sistemas (na função de Representantes da Área Usuária)**, a partir do Modelo da Essência, dos Requisitos de Concepção de Sistemas e da Especificação da Interface Usuário Máquina, deverão especificar o Plano de Testes de Aceitação.

Dentre os membros do Grupo de Sistemas, um deles deve ser o "**Gerente do Projeto**". Será o responsável por todas as decisões referentes a:

- adaptação do método de desenvolvimento do projeto ao problema em questão;
- coordenação dos trabalhos;
- provisão dos recursos necessários à equipe.

Atividade do Projeto	Equipes Envolvidas	Coordenação
Modelagem da Essência	Grupo de Sistemas	Gerente do Projeto
Modelagem dos Requisitos de Concepção	Grupo de Sistemas	Gerente do Projeto
Elaboração do Projeto Básico	Grupo de Sistemas	Gerente do Projeto
Especificação da IUM	Grupo de Sistemas	Gerente do Projeto
Manual do Usuário	Grupo de Sistemas	Gerente do Projeto
Plano de Testes de Aceitação	Grupo de Sistemas	Gerente do Projeto
Implementação de Subsistemas Básicos	Equipes de Implementação de Software	Líder de Equipe
Implementação de Módulos de Hardware	Equipes de Implementação de Hardware	Líder de Equipe
Integração	Equipes de Implementação envolvidas	Elemento do Grupo de Sistemas responsável pela Integração (Gerente de Integração)
Controle de Mudanças	Grupo de Sistemas	Gerente do Projeto e Coordenador de Área

Tabela 1 - Atividades x Equipes x Liderança

Seus principais objetivos serão o cumprimento de prazos e custos e a geração de um produto com qualidade assegurada por construção. Sua função específica mais importante será garantir a integração entre os diversos desenvolvimentos concorrentes e, em particular, garantir a consistência entre hardware e software visando assegurar a integridade funcional do sistema.

Concluídas as atividades de responsabilidade do Grupo de Sistemas, este pode ser reduzido, ficando apenas com três ou quatro integrantes (Engenheiros de Software, Hardware e Sistemas), responsáveis pela gerência total do projeto, das áreas de software, hardware, integração e aceitação do sistema. Os demais integrantes do Grupo de Sistemas passarão a liderar as novas equipes que se formam.

Iniciada a implementação dos Subsistemas Básicos, a equipe total deve crescer e, em função do porte do sistema, várias sub-equipes podem ser formadas, cada uma responsável pela implementação de um conjunto de Subsistemas Básicos. Essa divisão em sub-equipes permite a divisão de responsabilidades, constituindo uma hierarquia, o que permite maior controle sobre o processo de desenvolvimento. A Equipe de Projeto fica então constituída por:

- **Gerente de Projeto;**

- **Grupo de Sistemas**, responsável pelo Modelo da Essência, Especificação dos RCS, Especificação da IUM, Projeto Básico de Software e Hardware, Especificação Detalhada dos Subsistemas Básicos, Manual do Usuário e Plano de Testes de Aceitação;
- **Gerente de Integração**, quando o porte do projeto assim o exigir;
- **Coordenador do Desenvolvimento do Hardware e Coordenador do Desenvolvimento do Software**, aqui chamados de **Coordenadores de Área**, quando o porte do projeto assim o exigir.
- **Líderes de Equipe** das diversas sub-equipes de software e hardware que efetivarão o desenvolvimento do sistema. O Líder de Equipe (Engenheiro de Software e Engenheiro de Hardware) é o primeiro responsável pela qualidade do serviço, pela adesão ao método escolhido e pelo cumprimento da especificação dos subsistemas modelados pelo Grupo de Sistemas.
- **Equipes de Implementação**, responsáveis pelo design e implementação de módulos de hardware ou Subsistemas Básicos.

A relação Atividades do Projeto x Equipes x Liderança encontra-se resumida na Tabela 1.

4.1.1 - A Função do Gerente de Projeto

O Gerente de Projeto é responsável pela administração de todo o processo de desenvolvimento. Inicia-se esse processo com as diversas interações entre cliente/usuário e analistas até que haja acordo quanto às necessidades a serem atendidas pelo sistema. O gerente, tendo percebido as características básicas do sistema, deve:

- definir a estratégia a ser adotada no desenvolvimento, de forma a iniciar, o mais rapidamente possível, o atendimento às necessidades do cliente/usuário. Isso significa decidir quais versões parciais serão produzidas antes da versão final e o que deve ser implementado primeiro;
- definir o ambiente de desenvolvimento e quais ferramentas serão usadas, tais como linguagens, utilitários e pacotes (por ex: pacote de rotinas gráficas - GKS).
- garantir o controle de qualidade durante a fase de modelagem da essência, da especificação dos requisitos de concepção de sistemas e da especificação da interface usuário-máquina e obter a aprovação, junto ao usuário, do conteúdo desses modelos;
- garantir o controle de qualidade durante a fase de elaboração do Projeto Básico e a consistência do resultado dessa fase em relação à especificação;
- participar do processo de modelagem como Líder da Equipe de Projeto Funcional. Nessa qualidade, sua responsabilidade será a produção do Modelo da Essência e do Projeto Básico, de acordo com padrões de qualidade preestabelecidos;
- providenciar os recursos (por exemplo: aquisição de equipamentos) necessários ao desenvolvimento e implantação do sistema;
- definir que Subsistemas Básicos necessitam ser integralmente desenvolvidos e quais podem re-utilizar, mediante pequenas adaptações, subsistemas já desenvolvidos. Para isso, o Gerente do Projeto faz uso dos Subsistemas Básicos definidos no Projeto Básico e de uma biblioteca de módulos reusáveis;
- de posse da definição anterior, fazer um cronograma detalhado da fase de implementação, durante a qual desempenhará um papel de mediador, administrando as solicitações de mudança em relação às decisões tomadas em fases anteriores e, quando necessário, autorizando alterações. Essas mudanças podem ser de especificação, de design e de estratégia
- definir a ordem em que serão integrados os diversos Subsistemas Básicos e coordenar os Testes de Integração.

As atividades do Gerente, em relação a cada área, e relatam para o gerente apenas as mudanças que afetem a área complementar.

4.1.2 - A Função do Líder de Equipe

Em projetos que exijam a constituição de sub-equipes, é necessário manter um controle mais rígido sobre o desenvolvimento de cada subsistema, de forma a garantir que as interfaces entre Subsistemas Básicos sejam respeitadas.

Cada uma das Equipes de Implementação cuidará de um conjunto de Subsistemas Básicos correlacionados, sob a coordenação de um Líder de Equipe, que garantirá o controle de qualidade dos produtos gerados, a

adesão ao método de desenvolvimento adotado e o cumprimento das metas fixadas no Projeto Básico. Dessa forma, o projeto total fica segmentado em termos de subprojetos menores, com melhor controle e planejamento mais realista.

Os Líderes de Equipe desempenham, em pequena escala, o papel do Coordenador de Área ou do Gerente de Projeto, com a ressalva de que não possuem autonomia nas decisões que fujam aos aspectos internos dos subsistemas pelos quais são responsáveis. Isto é, quem toma decisões sobre a "caixa-preta" vista da parte interna é o Líder de Equipe, mas as decisões sobre sua aparência externa é do Coordenador de Área e do Gerente de Projeto.

4.2 - Interação entre Equipes

É importante instituir-se uma burocracia para o relacionamento entre equipes, já que a visão do todo - sistêmica - é obtida pelo Grupo de Sistemas, nas fases iniciais, e pelo Gerente de Projeto, nas fases de implementação.

Nas fases iniciais do processo de desenvolvimento, o Grupo de Sistemas funciona em conjunto sob a coordenação do Gerente de Projeto, que participa das discussões técnicas e responsabiliza-se pela validação de seus resultados. Nas fases posteriores, quando a equipe do projeto cresce, forma-se um conjunto de equipes independentes, cada uma sob a coordenação de seu Líder de Equipe visando a implementação de subsistemas. Nessa situação, é fundamental que cada líder mantenha o Gerente de Projeto informado do estado em que se encontra o desenvolvimento, permitindo o acompanhamento e a revisão do planejamento. O desenvolvimento do Hardware é paralelo ao do Software. Um controle rigoroso das modificações introduzidas deve ser exercido, visando a integração mais suave possível entre Hardware e Software. Além disso, como dificilmente as especificações dos Subsistemas Básicos e de suas interfaces são elaboradas corretamente desde o início, sempre haverá a necessidade de modificações quando iniciar-se a fase de implementação. Essas modificações serão discutidas em reuniões envolvendo os Líderes de Equipe, Coordenadores de Área e o Gerente de Projeto, onde serão decididas quais mudanças serão incorporadas à modelagem do sistema e como isso será feito.

É importante que cada alteração sugerida, além de ser registrada nos documentos do projeto, o seja também em formulário próprio, preenchido pelo solicitante, contendo sua descrição detalhada, o motivo técnico que a originou, o parecer da equipe responsável pela avaliação e a decisão quanto a sua efetivação. Esse histórico de alterações diminui a probabilidade de incorrer-se em erros já cometidos.

5 - Controle

O controle do processo de desenvolvimento é fundamental para o ajuste do plano de desenvolvimento. Devem ser minimizadas, o mais rápido possível, as diferenças entre o **Cronograma Previsto** e o **Cronograma Executado** e entre os **Recursos Previstos** e os **Recursos Utilizados**. O controle de qualidade é outra meta a ser atingida: ao final de cada fase, os produtos gerados devem satisfazer os requisitos de qualidade prefixados.

Controlar o desenvolvimento significa reconhecer a possibilidade de ocorrência de desvios referentes a custo/cronograma/qualidade, em relação ao planejado. Esses desvios devem e podem ser detectados com a antecedência necessária à correção [17].

O controle utiliza marcos específicos (modelos, parciais ou completos) e uma hierarquia gerencial constituída por Líderes de Equipe, Coordenadores de Área e Gerente do Projeto. Cada nível dessa hierarquia exerce controle de uma execução específica e reflete uma divisão real de responsabilidades.

5.1 - Acompanhamento do Desenvolvimento

Nas fases iniciais do processo de desenvolvimento, o acompanhamento do projeto é mais simples. O Gerente do Projeto, responsável pelo controle de todo o processo, é parte integrante de uma equipe reduzida, o Grupo de Sistemas. Isso o aproxima das dificuldades e de suas implicações em custo, prazo e qualidade dos produtos a serem gerados. O acompanhamento é diário e o nível de controle imposto à equipe pode ser elevado, devido à grande participação do gerente no processo e ao número reduzido de participantes. Isso permite ao gerente reavaliar, em função de situações imprevisíveis ou imprevistas, alocação de recursos, prazos e metas de qualidade, aprimorando a previsão inicial.

Terminado o Projeto Básico, o número de participantes da equipe cresce muito. É também dividida a responsabilidade em diversos níveis de gerência. Para que o Gerente do Projeto possa acompanhar a evolução

do estado de desenvolvimento, é necessário um controle de cada tarefa entregue às equipes participantes. O cronograma deve retratar marcos específicos (ou pontos de controle), capazes de dar uma noção do estágio em que se encontra o desenvolvimento. Esses pontos de controle são relativos a produtos que possam ser avaliados e/ou testados. Por exemplo: no desenvolvimento de um Subsistema Básico, o fim da fase de estruturação é um marco que possui como documento (modelo) o Design do subsistema. Os produtos resultantes serão submetidos a um controle de qualidade (um membro dedicado da equipe ou o Líder da Equipe), que avalie e produza um laudo do produto gerado (parte do Histórico do Projeto) e requisite as correções necessárias ao modelador. Esse laudo deve ser encaminhado à instância de gerência imediatamente superior, no caso, o Líder da sua Equipe, e às instâncias de gerência superiores à sua - Coordenadores de Área e Gerente do Projeto. Somente após aprovação do produto gerado, uma tarefa será considerada **pronta** no cronograma.

O laudo, elaborado sobre cada produto gerado, serve também de base para avaliação dos profissionais que constituem a equipe. Permite melhor planejamento, fornecendo subsídios aos gerentes para "entregar o serviço certo à pessoa certa".

O Gerente do Projeto deve manter reuniões quinzenais ou mensais, para acompanhamento das atividades realizadas pelas diversas equipes - **Reuniões de Coordenação** -. Esse acompanhamento permite o conhecimento do estágio de desenvolvimento do projeto - refletido no **Acompanhamento do Cronograma Previsto**: relatório das dificuldades encontradas, dos produtos gerados, laudos associados e alterações de cronograma realizadas.

A reavaliação dos prazos das atividades internas a cada equipe é feita pelo Líder de Equipe. No entanto, esses cronogramas detalhados são apresentados nas Reuniões de Coordenação, para conhecimento do estágio de desenvolvimento de cada Subsistema Básico, e detecção dos casos de incoerência. Essa apresentação evita que um Líder de Equipe, para não atrasar o prazo final, faça um planejamento impossível de cumprir e permite uma avaliação posterior da qualidade de liderança das equipes.

5.2 - Qualidade do Desenvolvimento

Obter qualidade requer um processo de desenvolvimento orientado a qualidade. É necessário que o processo de desenvolvimento permita a geração de produtos intermediários, cujos atributos de qualidade sejam bem definidos e estejam associados a mecanismos de medida dessa qualidade [1].

O controle do processo deve ser tal que garanta a qualidade do produto final ao longo de sua construção. Para isso, segundo a proposta gerencial deste trabalho, algumas formas de controle devem ser exercidas:

- validação, efetuada pelo Grupo de Sistemas e membros externos à Equipe do Projeto definidos pelo cliente/usuário, do Modelo da Essência em relação às necessidade do Ambiente Externo ao sistema;
- validação, efetuada pelo Grupo de Sistemas, do Projeto Básico em relação ao Modelo da Essência;
- acompanhamento diário do Líder de Equipe, junto aos membros da equipe, das tarefas em execução, solucionando os problemas surgidos no decorrer do trabalho e inspecionando aleatoriamente partes do serviço para verificar se o método de desenvolvimento adotado está sendo seguido. Dependendo do porte da equipe, esta última tarefa pode também ser realizada por um integrante dedicado ao controle de qualidade;
- revisão, efetuada pelo controle de qualidade interno à equipe, de cada produto intermediário gerado, antes que este possa ser considerado pronto. Essa revisão objetiva detectar inconsistências internas ao produto ou violação das normas que regem sua construção;
- revisão, efetuada pelo Coordenador de Área e/ou Gerente do Projeto, visando garantir que os produtos construídos pelas diversas equipes sejam passíveis de integração;
- revisão externa, efetuada por integrante do Grupo de Sistemas orientado pelo Coordenador de Área e Gerente do Projeto, visando garantir que o produto gerado esteja de acordo com as especificações do Projeto Básico;

A organização pode, ainda, fazer uso de outro tipo de Controle de Qualidade, através de Equipe Dedicada, diferente da envolvida no desenvolvimento e incumbida de realizar inspeções de rotina e aplicar técnicas de "walkthrough" [24,26]. No entanto, a **Garantia de Qualidade** assegurada por construção será, no entender dos autores, menos custosa sem perda de eficácia quando efetuada segundo as formas acima expostas. A economia decorre da utilização da própria Equipe de Projeto, por estar culturalmente integrada ao

processo de desenvolvimento e poder produzir respostas mais imediatas. A eficácia deverá ser assegurada por uma atitude gerencial correta, em que a liderança demonstre a importância do fator qualidade.

5.3 - Objetivos de Qualidade

A avaliação dos produtos intermediários gerados deve ser realizada a partir de padrões predefinidos, estabelecidos no início do projeto.

Cada organização deve adotar um método de desenvolvimento consistente com a classe de sistemas que constrói - o domínio do problema. Normalmente, esse método será suficientemente abrangente, podendo tratar qualquer porte de sistema pertencente à classe. O estabelecimento de um método de desenvolvimento (ou de uma linha metodológica) é importante para que os membros das equipes falem a mesma linguagem. Um método de desenvolvimento deve apresentar padrões bem definidos para todos os modelos resultantes de cada atividade - **Plano de Modelagem**. Com base nesses padrões é que será possível o trabalho das lideranças (Gerente de Projeto, Coordenadores de Área e Líderes de Equipe) para acompanhar, avaliar e validar esses modelos. Esse esforço gerencial é a base do controle de qualidade.

Algumas adaptações podem fazer-se necessárias em função do porte do sistema ou de pequenas variações no domínio do problema. Essas adaptações devem conduzir a um padrão específico para o projeto - **Plano de Modelagem do Projeto**, concebido em suas fases iniciais. Esse plano é a base do contrato **cliente/usuário x modeladores** que especifica a qualidade desejada para o sistema a ser desenvolvido.

5.4 - Controle do Risco

Este trabalho preocupa-se com uma classe de sistemas considerada de alto risco, em função dos requisitos, custos, prazos e recursos envolvidos. A abordagem gerencial do projeto deve focalizar esse risco [3]. Para isso, além das ferramentas de Planejamento e Controle descritas, devem ser usadas Ferramentas de Integração Externa [6].

As **Ferramentas de Integração Externa** visam comprometer o cliente/usuário com o processo de desenvolvimento, através da criação de um comitê com participação do cliente/usuário, do Gerente do Projeto, dos Coordenadores de Área e Engenheiros de Sistemas (representantes da Área Usuária). Reuniões periódicas, bimestrais por exemplo, avaliam o estágio do projeto, problemas e soluções de interesse do cliente/usuário. A aprovação formal de produtos de apoio ao desenvolvimento também faz parte desse comprometimento. O cliente/usuário necessita aprovar

- Modelo da Essência;
- Especificação da Interface Homem-Máquina;
- Manual do Usuário;
- Plano de Integração;
- Plano de Testes de Aceitação

e deve participar do processo de aceitação e implantação do sistema.

6 - Conclusão

É reconhecido atualmente que a atividade gerencial é fator crítico de sucesso para um projeto de desenvolvimento de sistemas de software. Equipes tecnicamente qualificadas podem fracassar se a gerência do processo for incompetente. Além disso, engenheiros de software de grande experiência e renome proclamam que ganhos de ordem de grandeza em produtividade não devem ser esperados a partir de investimentos em novas tecnologias de software [8]. Com o mesmo vigor, afirmam que as grandes dificuldades para obter-se incrementos significativos de produtividade e qualidade não residem na vertente técnica da Engenharia de Software e sim na gerencial.

A proposta gerencial contida neste trabalho procura explorar ao máximo a estrutura conceitual do software concebida a partir dos esforços de pesquisa mais recentes na área de Métodos Estruturados. O mapeamento dos elementos dessa estrutura sobre fases de um processo de desenvolvimento, levando em conta simultaneamente as necessidades de hardware/dispositivos, permite administrar (planejar, controlar, organizar e liderar) o projeto de modo ágil e flexível. Por proporcionar grande visibilidade sobre a evolução do sistema a ser construído, a granularidade dessa estrutura permite controle rigoroso do processo de desenvolvimento em seus aspectos mais importantes: custo, prazo e qualidade.

Incorporando uma estratégia de Desenvolvimento Incremental, a proposta exige que versões operacionais parciais, inteiramente compatíveis com a estrutura do produto final, sejam entregues em prazos curtos ao longo da evolução do projeto. Com isso, visa-se satisfazer as necessidades do cliente/usuário de modo mais eficiente, permitindo-lhe, ainda, participar mais eficazmente do processo de desenvolvimento.

Em resumo, a otimização de custo, prazo e qualidade é perseguida através de:

- implantação de uma estrutura gerencial que reflita a estrutura conceitual do sistema que está sendo desenvolvido;
- ajuste econômico do número de componentes da equipe;
- gerência matricial e hierárquica;
- qualidade assegurada ao longo do processo de construção.

Basicamente, são esses os aspectos que, por um lado, distinguem o modelo gerencial aqui apresentado de outras propostas [6,8,24,25] existentes na literatura e que, por outro lado, constituem os indicativos mais fortes para sua eventual adoção.

Uma dinâmica de desenvolvimento baseada em estrutura gerencial que reflita um conceito bem definido relativo ao produto a ser gerado evita desperdícios de tempo e recursos materiais. Esses desperdícios são incorridos seja construindo-se modelos incorretos ou desnecessários seja deixando de construir corretamente modelos importantes, o que poderia só ser descoberto tardiamente, introduzindo atrasos e queda de qualidade no produto final. O controle também é facilitado pois é efetuado sobre marcos nítidos resultantes do processo de modelagem. O método de desenvolvimento de sistemas que ilustra o modelo gerencial apresentado permite uma avaliação bastante acurada dos desperdícios possíveis bem como dos mecanismos para evitá-los. Além disso, o método abriga facilmente a estratégia de Desenvolvimento Incremental e encoraja Reuso.

O modelo apresentado recomenda o crescimento da equipe apenas durante as atividades de implementação, a serem realizadas sobre as especificações dos Subsistemas Básicos. Uma Equipe de Implementação preocupa-se apenas com os Subsistema Básicos a ela alocados. Essa característica permite economia de custos (membros da equipe são estritamente o necessário em cada atividade) e baixa vulnerabilidade do projeto em relação a mudanças nas diferentes equipe, comuns em projetos de longa duração. Apesar de nem todos os membros da Equipe de Projeto possuírem visão total do sistema, a integração dos Subsistemas Básicos fica garantida exclusivamente pelo Grupo de Sistemas pois só ele tem poder de decisão para alterar a Especificação Detalhada de um Subsistema Básico. A estrutura hierarquizada das equipes limita a comunicação entre equipes às interfaces dos Subsistemas Básicos, o que reduz os problemas advindos desse tipo de interação.

A coordenação da Equipe de Projeto, na figura do Gerente do Projeto, é repartida pelos Coordenadores de Área e Líderes de Equipe. Esses últimos executam um controle amíúde da modelagem em equipes de menor porte, mais facilmente auditadas. A limitação da capacidade de decisão de cada um desses subgerentes garante a consistência e integração dos modelos que estão sendo construídos em paralelo. Essa hierarquização da matriz gerencial permite validar diretamente, durante o processo de geração, cada modelo resultante do processo de desenvolvimento. Determina a qualidade do produto final ao longo de seu processo de construção. Nenhum produto intermediário é considerado "pronto" enquanto o Líder de Equipe não aprová-lo em função do laudo que o avalia. Inspeções de qualidade durante o desenvolvimento evitam o desperdício de tempo com uma solução inadequada.

A validação experimental dessa proposta ainda não foi realizada globalmente. Experimentos limitados, incidindo sobre seus aspectos básicos, encorajam os autores a sugerir sua adoção em organizações envolvidas com projetos incorporando sistemas de tempo-real de grande porte. Os princípios gerenciais nos quais baseia-se o modelo aqui apresentado são válidos e a estrutura proposta resulta de um mapeamento a partir do produto a ser gerado. Provavelmente, mesmo organizações que implementem diferentemente a gerência de desenvolvimento, ou que não visem ao porte de sistemas pressuposto pelo modelo, reconhecerão aspectos positivos que possam influenciar benéficamente a sua prática.

Bibliografia

- [1] Basili V. R. & Musa J. D.; The Future Engineering of Software: A Management Perspective; IEEE Computer; setembro de 1991
- [2] Boehm B.; Software Eng. Economics; Prentice Hall; Englewood Cliffs, N.J.; 1981
- [3] Bran Selic, Garth Gullekson, Jin McGee, Ian Engelberg; ROOM: An Object-Oriented Methodology for Developing Real-Time Systems; Case'92 Fifth International Workshop on Computer- Aided Software Engineering, Montreal, Quebec, Canada; Julho 6-10, 1992.
- [4] Clemente K.; Modelagem de Sistemas Sócio-Técnicos, Estudo de Caso de um Piloto Automático para Automóvel, Tese de Mestrado, Departamento de Engenharia Elétrica, PUC-Rio, abril de 1992
- [5] Cordeiro M. M., Vasconcelos M. G. P., Sanchez M. L.; Requisitos de Concepção de Sistemas; Publicação Interna do Instituto de Pesquisas da Marinha; 1988
- [6] Fernandes A. A., Kugler J. L. C; Gerência de Projetos de Sistemas; Livros Técnicos Científicos; 1990
- [7] Garbin E. J. da Rocha; Software e Processo de Desenvolvimento de Software - Algumas Reflexões sob uma Perspectiva de Modelagem da Essência; Tese de Mestrado; Departamento de Informática, PUC-Rio, maio de 1991
- [8] Glass R. L.; Software Conflict - Essays on the Art and the Science of Software Engineering; Prentice-Lall; 1991
- [9] IEEE; Guide to Software Requirements Especifications; ANSI/IEEE STD 830; Julho 1984
- [10] IEEE; Standard for Software Test Documentation; ANSI/IEEE; Std 829 - 1983
- [11] Kemerer, C.F. An Empirical Validation of Software Cost Estimation Models, Communications of the ACM, v 30, n5; Maio, 1987
- [12] Leveson N. G.; Software Safety: What, why and how; ACM Computer Surveys, vol. 18, n° 2, pp. 125-164, junho, 1986
- [13] Maffeo B.; Engenharia de Software e Especificação de Sistemas; editora CAMPUS; 1992
- [14] Mcmenamin M. e Palmer J. F.; Essential Systems Analysis; Yourdon Press; 1984
- [15] Menascé D. e Almeida V. A. S.; Planejamento de Capacidade de um Sistema de Computação - Análise Operacional como Ferramenta ; Editora Campus 1ª edição; 1985
- [16] Ramamoorthy C.V., Garg V. & Prakash A.; Programming in Large; IEEE Transactions on Software Engineering, Vol. SE- 12; fevereiro de 1986
- [17] Ritto A. C. A. & Maffeo B.; Funções do Gerente de Desenvolvimento de Sistemas - Algumas Reflexões; Monografia em Ciência da Computação, Departamento de Informática; PUC - Rio; 27/91
- [18] Salim C.S.; Notas de Aula em curso de Gerência de Projetos Sócio-Técnicos; DEI/PUC-Rio; 90.2
- [19] Sanchez M. L. & Maffeo B.; Gerência de Projeto ("Design") Orientado a Encapsulamento de Dados e a Troca de Mensagens entre Sistemas Autônomos; Monografia em Ciência da Computação, Departamento de Informática; PUC-Rio; 9/91
- [20] Sanchez M. L. & Maffeo B.; Projeto ("Design") Orientado a Encapsulamento de Dados e a Troca de Mensagens entre Sistemas Autônomos; Monografia em Ciência da Computação, Departamento de Informática; PUC - Rio; 8/91
- [21] Tarek K., Abdel-Hamid & Madnick S. E.; Lessons Learned from Modeling the Dynamics of Software Development; Communications of ACM; dezembro de 1989
- [22] Ward P. e Mellor S.; Structured Development for Real-Time Systems; Yourdon Press
- [23] Wiest J. e Levy F.; A Management Guide to PERT/CPM; Prentice-Hall; 1977
- [24] Yourdon E.; Modern Structured Analysis; Yourdon Press; 1986
- [25] Yourdon E.; Administrando Técnicas Estruturadas; Campus; 1988
- [26] Yourdon E.; Structured Walkthroughs; Yourdon Press; 1985