

JEAN CARLO CAMARGO FERREIRA

**GERENCIAMENTO DE ESCOPO E
QUALIDADE EM PROJETOS
VOLTADOS À TECNOLOGIA DA
INFORMAÇÃO®**

**Monografia apresentada a
Universidade Federal do
Paraná como parte dos
requisitos para a obtenção do
título de Especialista em
Gerenciamento de Projetos.**

Orientador: Prof. Dr. José Amaro dos Santos

Curitiba 2005

“Dedicatória”

A Deus por me guardar e iluminar, a minha família pela intensa dedicação, e por meus amigos e colegas pelo apoio.

*Quem não sonha não realiza sonhos, fazer nascer em cada dia que te é ofertado,
a beleza de nossa essência, é contemplar a existência.*

“Ignoto”

Sumário

Resumo	3
Abstract	4
Lista de Abreviaturas	5
Lista de Figuras	5
Lista de Tabelas	5
1. Introdução	6
1.1.1 O que é um projeto.....	7
1.1.2 Diferentes Fases do Projeto	8
1.1.3 A tomada de decisão em Projetos	10
1.1.4 Ciclo de Vida do Projeto.....	10
1.1.5 Ciclo de Vida Representativos do Projeto	13
1.1.6 Histórico e acompanhamento do software.....	14
1.1.7 O Papel evolutivo do software.....	15
1.1.8 Aplicações do Software	16
2. Gerenciamento de Projetos (GP)	18
2.1.1 Atribuições do Gerente de Projeto.....	19
2.1.2 Gerenciamento de Projetos em Software.....	20
2.1.3 Processos e Sistemas nas organizações para projetos.....	21
2.1.4 O Conceito Sistêmico em projetos.....	22
2.1.5 Hierarquia dos sistemas e suas interfaces	24
2.1.6 Gestão do conhecimento aplicado a Sistemas	25
3. Escopo em Projetos	26
3.1.1 Gerência de Escopo em projetos.....	27
3.1.2 Início de Escopo.....	28
3.1.3 Seleção do projeto pelo escopo.....	29
3.1.4 Planejamento do Escopo	29
3.1.5 Detalhamento do Escopo	30
3.1.6 Verificação de Escopo	31
3.1.7 Controle de Mudanças do Escopo.....	31
3.1.8 O Escopo em Software.....	32
4. Qualidade em Projetos	35
4.1.1 Aspectos da Qualidade.....	36
4.1.2 Medidas da Qualidade.....	38
4.1.3 A trilogia da Qualidade de Juran	39
4.1.4 Qualidade Total – Prevenção e Inspeção	43
4.1.5 O Papel do Gerente de Projetos na Qualidade.....	43
4.1.6 CMM – Capability Maturity Model.....	44
4.1.7 ISO 9000	55
4.1.8 A Série ISO-9000 Aplicada ao Software.....	57
4.1.9 Atividades do Ciclo de Vida do Software.....	58
4.2.0 Protótipo em software.....	61
4.2.1 O Modelo em espiral.....	63
5. Qualidade e o Modelo RUP	65
5.1.1 A Concepção da excelência em RUP.....	65

5.1.2 As Fases do RUP	65
5.1.3 Workflows do Ciclo de Desenvolvimento	66
5.1.4 Artefatos e Modelos	68
6. Estudo de caso “Migração de Tecnologia”	68
6.1.1 Breve Relato	68
6.1.2 Histórico	69
6.1.3 Objetivos do Negócio	69
6.1.4 Controles de Auditoria	70
6.1.5 Impactos no negócio	71
6.1.6 Impactos no Ambiente Operacional	71
6.1.7 Descrição Geral	72
6.1.8 Processo Operacional	73
6.1.9 Visão da arquitetura	74
6.2.0 Visão do Negócio	75
6.2.1 Estratégia de Implantação	75
6.2.2 Descrição Geral do Processo	78
6.2.3 Processo aliado à Qualidade	79
6.2.4 Resumo Qualitativo	79
7. Conclusão.....	81

Resumo

O crescimento acentuado nas grandes corporações em busca de uma qualidade assegurada, um produto ou mesmo serviços, são fatores propulsores a que as organizações desempenham com maior intensidade e foco. Em virtude deste fato, se torna mais presente a adoção de ferramentas e técnicas que lhes permitam gerenciar de uma forma mais eficiente e orientada a resultados, prezando pelos fatores de escopo e qualidade na gerência e gestão de projetos voltados a Tecnologia de Informação.

O escopo está intimamente relacionado a todos os estágios de desenvolvimento e andamento do projeto, se faz mais presente no início do projeto e geralmente está passível de alteração no desenvolver e caminhar do projeto.

A qualidade é um dos fatores que está presente nas mais diversas áreas de atuação profissional, se faz presente em diversos ramos do conhecimento humano aplicado a processos, projetos dentro das organizações e demais áreas de atuações.

Na atual conjuntura e sistemas de forma de trabalho a que as organizações e empresas estão sujeitas, se faz presente e necessário uma forma de abordagem direta e eficaz sobre como fazer o gerenciamento de projetos, atingindo os propósitos e metas estipuladas no início do projeto para sua implementação, gestão e gerenciamento em sua continuidade.

O gerenciamento de projetos, ou mesmo a tecnologia da informação com uma abordagem aos sistemas de informação é uma área que está em constante transformação e evolução. Os conceitos e formas a qual as pessoas, processos e projetos estão sujeitos a alterações e mudanças de acordo com a filosofia das empresas.

Neste contexto se revela de extrema valia o aperfeiçoamento, nos quesitos de escopo e qualidade para o sucesso dos projetos corporativos. Este instrumento traz a conhecimento técnicas e práticas que vão desde o acompanhamento do ciclo de vida do projeto, gerência de escopo e qualidade, atribuições do gerente de projetos, qualidade em projetos, qualidade total, CMM, ISO 9000, até RUP, as quais embasam as organizações pela excelência na execução de projetos corporativos.

Sobre esta perspectiva e visão de atuação, é fundamental conhecer e compreender os elementos que constituem um escopo claro e definido, associado a padrões de conduta e qualidade a qual este instrumento se ressalva a fazer a área de *TI Tecnologia da Informação(TI)*.

Abstract

The accentuated growth in the great corporations in search of an assured quality, of product, or services, is one of the factors that impels organizations with great intensity and focus. Due to this fact, the adoption tools and technology have become more present, that allow them to manage a more efficient form directed to results, esteem for the factors of target and quality in the management of projects directed to the Information Technology.

The target is intimately related to all stages of training development and course of the project, it makes more present in the beginning of the project and generally it is subject to alteration in developing and course of the project.

The quality is one of the factors that are present in most diversified areas of professional interaction, it is present in many branches of the human knowledge applied to processes, projects inside of the organizations and also different areas of interaction.

In the current reality and systems of work form, the organizations and companies are citizens, makes present and necessary a direct form and efficient approach on how to make the management of projects, reaching the intentions and goals target in the beginning of the project for its implementation, management and in its continuity.

The management of projects or same the information technology, with an approach to the information systems is an area that is in constant transformation and evolution. The concepts and forms which the people, processes and projects are subordinated the alterations and changes in accordance to companies policies.

It's revealing that extreme value the perfection, in the questions of target and quality for the success of the corporative projects. This instrument brings the knowledge of practical techniques and they go since the accompaniment of the cycle of life the project, management of target and quality, attributions of the manager projects, quality in projects, total quality, CMM, ISO 9000, until RUP, which base the organizations for the excellence, in the execution of corporative projects.

On this perspective and vision of performance, it is basic to know and understand the elements that constitute and define a clearly scope and associate standards conduct and quality which this instrument proposes to make the area of Information Technology (TI).

Keywords: management project, information technology, software development, quality, scope, changes, objectives.

Lista de Abreviaturas

PMO	Project Management Office
TI	Tecnologia da Informação
PMI	Project Management Institute
PMBOK	Project Management Body of Knowledge
GP	Gerente de Projetos
PMP	Project Management Professional
RUP	Rational Undefined Process
CMM	<i>Capability Maturity Model</i>
ISO	<i>International Organization for Standardization</i>
WBS	<i>Work Breakdown Structure</i>

Lista de Figuras

Figura.	Página.
Figura 1 – Ciclo de vida do Projeto, segundo PMBOK	11
Figura 2 – Níveis de Atividade do Projeto por tempo	12
Figura 3 – Relacionamentos para requisitos do projeto	14
Figura 4 – Interações entre sistemas	24
Figura 5 – Sequência de Escopo	29
Figura 6 – Planejamento de Escopo	30
Figura 7 – Detalhamento de Escopo	30
Figura 8 – Estrutura de Escopo	31
Figura 9 – Controles de mudança de Escopo	32
Figura 10 – Processo de Controle	33
Figura 11 – Diagramas da Trilogia de Juran	42
Figura 12 – Os cinco níveis do CMM	48
Figura 13 – Planejamento de Projeto CMM	53
Figura 14 – Protótipos em software	61
Figura 15 – O Modelo em espiral	64

Lista de Tabelas

Tabela	Página
Tabela 1 – Aspectos da Qualidade	36
Tabela 2 – Gerenciamento para a Qualidade	41
Tabela 3 – Nível de maturidade das Organizações	46
Tabela 4 – Resumo dos níveis de Maturidade	54
Tabela 5 – Referências a ISO	57

1. Introdução

È cada vez mais presente à forma de gerenciar projetos com alto grau de qualidade e um escopo bem definido. Nas mais diversas áreas de atuação do conhecimento humano, se faz necessário atingir um padrão de qualidade e eficácia de gerenciamento e gestão de projetos independente da área de atuação. A forma, métodos, conceitos e a própria maneira de conduzir um projeto estão intrinsecamente ligados ao sucesso do projeto em seu andamento e conclusão.

Atualmente tem se acrescido o valor agregado à importância do gerenciamento de projetos nas organizações e grandes empresas de âmbito nacional e internacional. Em face a este fato, os padrões de condução de projetos tem acrescido uma considerável evolução nos quesitos imprescindíveis a seu sucesso.

Com tal desenvolvimento em constante grau de evolução e estudos sobre as principais fundamentações para o sucesso de projetos se chegou ao patamar de obter-se um referencial ao gerenciamento de projetos denominado como *Project Management Body of Knowledge* ou mais popularmente conhecido como PMBOK, a qual representa um somatório do conhecimento do gerenciamento de projetos. O PMBOK é mais vinculado à prática de um conjunto de conhecimentos contribuídos por profissionais e gerentes que aplicam estes conhecimentos, como uma forma de obtenção em excelência em projetos.

Uma das áreas que se observa um acentuado crescimento na prospecção de projetos é a tecnologia da informação. Muitos projetos estão integrados e gerenciados por ferramentas de tecnologia da informação.

Apesar da não importância de seus muitos usuários, hoje sua aplicação se dá nas maiores áreas da sociedade de bens de consumo, como indústria, áreas de pesquisas e diversos outros ramos que chegam a nossa realidade através de caixas eletrônicos, e-mails, celulares, microondas, computadores de bordo de automóveis e inúmeros outros segmentos presentes em nosso convívio social.

As empresas de Tecnologia da informação estão sobre um cenário de constante transformação e mutação sobre seus projetos tecnológicos, assim como também a demanda de serviços e desenvolvimento de softwares para implementação de seus projetos nas mais diversas áreas de atuação.

Em relação a este cenário, devida às necessidades de se ter um alto critério de padrão de qualidade nos quesitos de desenvolvimento de software e delineamento preciso do escopo de projeto, a gerencia de projetos de TI para o desenvolvimento de software, se torna um quesito fundamental.

As atuais desenvolvimentos para o desenvolvimento de software, como fator de andamento para projetos de TI, é o principal alicerce para a conclusão de um protótipo final, que pode estar associado ao gerenciamento de um processo ou mesmo a efetivação de uma solução que se adere às necessidades corporativas.

Para se obter um apreciável grau de qualidade voltado aos projetos corporativos de cunho tecnológico, é importante conhecer as práticas que assegurem uma conformidade em relação a toda iteração que fará frente ao produto final do projeto. A conformidade no ciclo de vida do projeto está embasada no escopo. O escopo por sua vez, define as sucessivas etapas de maturação do projeto e em qual etapa se faz mais presente e necessário, o comprimento de condutas e práticas voltado a excelência em projetos de *Tecnologia da Informação*.

1.1.1 O que é um projeto

Segundo Victor Roldão Vr (04), em seu livro *Gestão de Projetos*, o trabalho envolve *serviços continuados e/ou projetos*, embora possa haver superposição entre os dois. Serviços continuados e projetos possuem muitas características comuns; por exemplo, ambos são:

Executados por pessoas

- Restringidos por recursos limitados
- Planejados, executados e controlados.

Projetos são frequentemente implementados como meios de realizar o plano estratégico da organização. Serviços continuados e projetos diferem principalmente porque enquanto os primeiros são contínuos e repetitivos, os projetos são temporários e únicos. Assim, um projeto pode ser definido em termos de suas características distintas, *um projeto é um empreendimento temporário com o objetivo de criar um produto ou serviço único*.

Temporário significa que cada projeto tem um começo e um fim bem definidos. *Único* significa que o produto ou serviço produzido é de alguma forma diferente de todos os outros produtos ou serviços semelhantes. Para muitas organizações, projetos são o meio de responder a requisitos que não podem ser atendidos através dos limites normais de operação da organização.

Os projetos são desenvolvidos em todos os níveis da organização. Eles podem envolver uma única pessoa ou milhares delas. Podem durar poucas semanas ou mais de cinco anos. Os projetos podem envolver uma unidade isolada da organização ou atravessar as fronteiras organizacionais, como ocorre com consórcios e parcerias. Os projetos são críticos para a realização da estratégia de negócios da organização por que projetos são os meios pelos quais as estratégias são implementadas. Pode-se citar como exemplos de projetos:

- Desenvolver um novo produto ou serviço
- Implementar uma mudança organizacional no nível de estrutura, de pessoas ou de estilo gerencial.
- Planejar um novo veículo de transporte
- Desenvolver ou adquirir um sistema de informação novo ou modificado
- Construir um prédio ou instalações
- Desenvolver um sistema de abastecimento de água para as comunidades em desenvolvimento
- Levar a cabo uma campanha política
- Implementar um novo processo ou procedimento organizacional

1.1.2 Diferentes Fases do Projeto

Concepção do Projeto

De início, ainda em fase exploratória, grande parte dos aspectos inerentes ao projeto são ainda pouco claros. Segundo Victor Roldão Vr. (04), a fase de concepção vai fundamentalmente clarificar esses aspectos, através de discussões que envolvem o Gerente de Projeto os seus colaboradores e o cliente. Com uma clara compreensão do objetivo final, o gerente de projeto pode então definir as principais linhas de ação, utilizando nomeadamente a seguinte metodologia:

1. Escrever a definição do projeto
2. Fixar os resultados finais do projeto
3. Listar o que é imperativo e desejável
4. Gerar estratégias alternativas
5. Avaliar essas alternativas
6. Escolher linhas de ação

Freqüentemente a concepção do produto tem duas subfases:

A – Realização de estudos prévios, estudos de sector e de mercado, e eventualmente estudos prévios de investimento.

A. - Após ter decidido que o projeto vai avançar, realização de estudos mais detalhados, o que conduz ao aprofundamento dos estudos prévios (freqüentemente com o apoio de consultores!).

- Trata-se de um projeto de uma fábrica, esta fase normalmente inclui:
 - Realização de consultas prévias
 - Realização de estudos técnicos incluindo estimativas de custo
 - Definição do processo
 - Traçado definitivo de desenhos e especificações (Engenharia de detalhe)
 - Definição de quantidades a fabricar
 - Determinação do custo final e providenciar para obtenção dos fundos necessários ao financiamento do projeto.
 - Pré-qualificação de Sub-contratados.
 - Consideração dos aspectos operatórios da manutenção
 - Consideração do processo de recrutamento e formação
 - Obtenção de autorizações definitivas, licenças e etc.

Planejamento e Orçamento do Produto

Nesta fase é definida a estratégia de implementação, sendo fixados objetivos e os recursos necessários ao cumprimento desses objetivos. É nesta fase que se antecipam muitos dos problemas, e são negociados os financiamentos.

O planejamento engloba nomeadamente:

- O programa de implementação
- A determinação do encadeamento de ações paralelas.
- A estimativa das durações das várias ações paralelas
- A estimativa dos recursos necessários em cada momento
- O estudo antecipado de cenários possíveis

Implementação do Projeto

Edificar o empreendimento dentro do custo, tempo e qualidade acordados, constitui o objetivo central da implementação. As previsões são permanentemente confrontadas com a realidade e os desvios são medidos através de um processo de decisão contínuo.

Implementar é executar e controlar liderando uma equipe, no que são especialmente relevantes os seguintes aspectos:

Organização, recrutamento e formação da equipe de projeto, e sua articulação com a estrutura da empresa.

Liderança da equipe de projeto

- Redefinição da Engenharia de detalhe
- Contratação e relação dos itens necessários ao projeto
- Negociação definitiva do financiamento
- Acompanhamento e revisão permanente do Planejamento e Orçamento

Em suma, na implementação do projeto é fundamental:

- Gestão da execução
- Motivação
- Resolução dos problemas
- Medição de resultados

Conclusão do Projeto

Esta fase consiste fundamentalmente em:

Proceder à recepção e testes do Produto

Apoiar o arranque da nova instalação (se for caso disso)

Dissolver a equipe do projeto e dá-lo por concluído.

1.1.3 A tomada de decisão em Projetos

Uma vez elaborado o planejamento geral, já na fase de concretização, surge a necessidade de implementar o projeto e de tomar medidas corretivas necessárias à prevenção de problemas maiores. Segundo Victor Roldão Vr. (04), se o gerente consegue gerir os pequenos problemas, os grandes poderão nunca aparecer, sendo certo que “é pior não tomar decisões do que tomá-las de forma errada” – com uma decisão errada o gerente sabe onde errou, com uma “não decisão” não sabe nada.

O interesse de tomar muitas pequenas decisões, reside na aprendizagem que se faz com a experiência e com os pequenos erros. Há no entanto, vantagem em não cometer grandes erros de forma a ganhar a confiança de si próprio, da administração e do cliente, e de gerar uma imagem de competência e conhecimento perante os colaboradores.

Porque só se pode atacar um problema se o percebermos, uma metodologia simples que se pode seguir no ataque aos problemas é a seguinte:

1. Qual o problema?
 2. Quais as alternativas?
 3. Qual a melhor alternativa?
- (só se pode atacar um problema se o percebermos)

1.1.4 Ciclo de Vida do Projeto

Segundo o PMBOK (02), o ciclo de vida do projeto serve para definir o início e o fim de um projeto. Por exemplo, quando uma organização identifica uma oportunidade dentro de sua linha de atuação, normalmente ela solicita um estudo de viabilidade para decidir se deve criar um projeto. O ciclo de vida do projeto determina se o estudo de viabilidade constituirá a primeira fase do projeto ou se deve ser tratado como um projeto à parte.

A definição do ciclo de vida do projeto também determina os procedimentos de transição para o ambiente de operação que serão incluídos ao final do projeto, distinguindo os dos que não serão. Desta forma, o ciclo de vida do projeto pode ser usado para ligar o projeto aos processos operacionais contínuos da organização executora.

A seqüência de fases, definida pela maioria dos ciclos de vida de projeto, tais como “solicitações” para “design”, “construção para operações” ou “especificação” para “manufatura”, geralmente envolve alguma forma de transferência de tecnologia ou handoff.

Os subprodutos oriundos de uma fase normalmente são aprovados antes do início da próxima fase. Entretanto, quando os riscos são considerados aceitáveis, a fase subsequente pode iniciar antes da aprovação dos subprodutos da fase precedente. Esta prática de sobreposição de fases é usualmente chamada de *fast tracking*.

Os ciclos de vida dos projetos geralmente definem:

- Que trabalho técnico deve ser realizado em cada fase (por exemplo, o trabalho do arquiteto deve fazer parte da fase de definição ou da fase de execução?).
- Quem deve estar envolvido em cada fase (por exemplo, Implementadores que necessitam ser envolvidos com levantamento de requisitos e especificação).

As descrições do ciclo de vida de projeto podem ser genéricas ou detalhadas. Descrições muito detalhadas podem conter uma série de formulários, diagramas e checklists para prover estrutura e consistência. Estas abordagens detalhadas são frequentemente chamadas de metodologias de gerência de projeto.

A maioria das descrições do ciclo de vida de projeto apresentam algumas características em comum:

O custo e a quantidade de pessoas integrantes da equipe são baixos no início do projeto, sofre incrementos no decorrer do mesmo e se reduzem drasticamente quando seu término é vislumbrado. Este modelo é ilustrado na figura 1, logo a seguir:

Segundo o PMBOK (02), no início do projeto, a probabilidade de terminá-lo com sucesso é baixo e, portanto, o risco e a incerteza são altos. Normalmente a probabilidade de sucesso vai aumentando à medida que o projeto caminha em direção ao seu término.

A capacidade das partes envolvidas de influenciar as características finais do produto do projeto e o seu custo final é alto no início e vai se reduzindo com o andamento do projeto. Isto acontece, principalmente, porque o custo de mudanças e correção de erros geralmente aumenta à medida que o projeto se desenvolve.

Deve-se tomar cuidado para distinguir *ciclo de vida de projeto* de *ciclo de vida do produto*. Por exemplo, um projeto para lançar no mercado um novo computador de mesa é somente uma fase ou estágio do ciclo de vida deste produto.

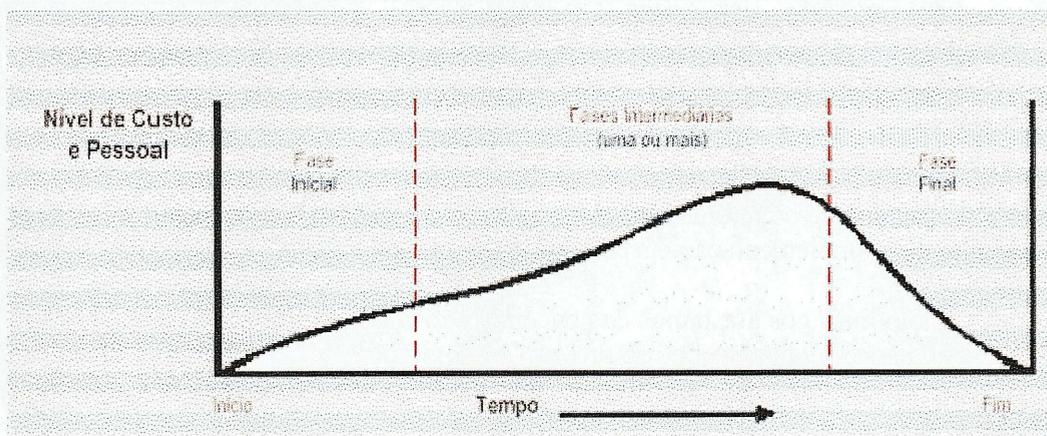


Figura 1

Ainda que muitos ciclos de vida de projeto apresentem nomes de fases similares com resultados de trabalho similares, poucos são idênticos. Embora a maioria tenha quatro ou cinco fases, alguns chegam a ter nove ou mais. Mesmo numa mesma área de aplicação, temos variações significativas – numa organização, o ciclo de vida para desenvolvimento de

software pode ter uma única fase de design, enquanto em outra, pode apresentar duas fases, uma para especificação funcional e outra para design detalhado.

Subprojetos, dentro dos projetos, podem ter ciclos de vida separados. Por exemplo, uma empresa de arquitetura contratada para projetar um novo prédio de escritórios estará inicialmente envolvida com a fase de definições do contratante, quando da elaboração do projeto, e com a fase de implementação, quando fornecendo suporte à construção. O projeto de desenho arquitetônico, no entanto, terá sua própria série de fases desde a especificação conceitual, passando pela definição e implementação, até o encerramento.

O arquiteto pode, ainda, tratar o design do prédio e o suporte à construção como projetos separados com suas próprias fases.

Segundo Victor Roldão, autor de “Gestão e Projetos” editora monitor, denota-se sobre uma outra perspectiva de visão, onde se pode inferir sobre o nível de Atividade como mostra na figura 2:

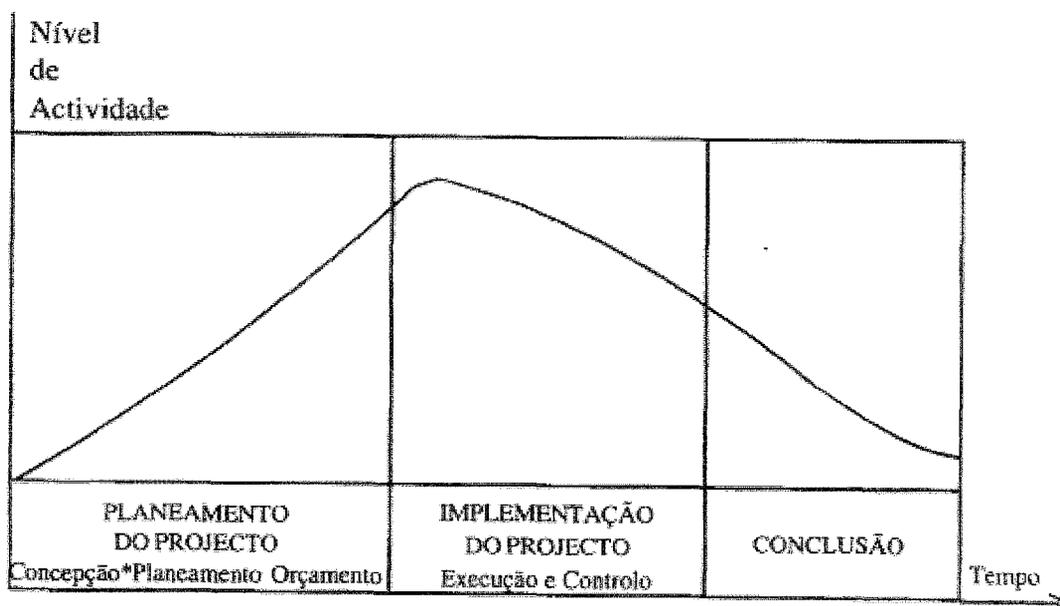


Figura 2

Normalmente o ciclo de vida do projeto inicia-se com a concepção, a que se segue o planejamento, a implementação e a conclusão do projeto.

Quanto mais nos afastamos da concepção, maior é o volume de recursos consumidos e mais difícil se torna fazer refletir o projeto, como demonstrado na Figura 2 Planejamento.

Assim, quanto maior o valor do investimento em causa, maiores deverão ser os estudos realizados em:

Possibilidade de Influenciar os Resultados

- Estudos de Mercado
- Ensaios
- Simulações de computador

Possibilidade de Influenciar os Resultados		Consumo Acumulado de Recursos
PLANEAMENTO DO PROJECTO Concepção*Planeamento Orçamento	IMPLEMENTAÇÃO DO PROJECTO Execução e Controlo	CONCLUSÃO

Figura 2 Planeamento

1.1.5 Ciclo de Vida Representativos do Projeto

Segundo o PMBOK (02), os seguintes ciclos de vida foram selecionados para ilustrar a diversidade de abordagens em uso. Os exemplos apresentados são típicos; eles não são nem recomendados nem preferidos.

Aquisição pelo Sistema de Defesa. A instrução 5000.2 do Departamento de Defesa Americano, em sua revisão final, abril de 2000, descreve uma série de fases e marcos para o processo de aquisição, como ilustrado na figura 3 a seguir:

- Conceituação e desenvolvimento tecnológico – estudos de alternativas conceituais para encontrar a missão solicitada. Desenvolvimento de subsistemas / componentes e demonstração conceitual/ tecnológica dos novos conceitos do sistema. Termina com a seleção da arquitetura de sistema e a maturidade tecnológica a ser usada.
- Desenvolvimento de sistema e demonstração – integração do sistema; redução de risco; demonstração dos modelos de engenharia desenvolvidos; desenvolvimento e testes iniciais de operação e avaliações. Encerra com a demonstração do sistema no ambiente operacional.
- Produção e operação – início de produção em pequena escala (LRIP) desenvolvimento completo da capacidade de fabricar; as fases sobrepõem os processos contínuos de operação e suporte.

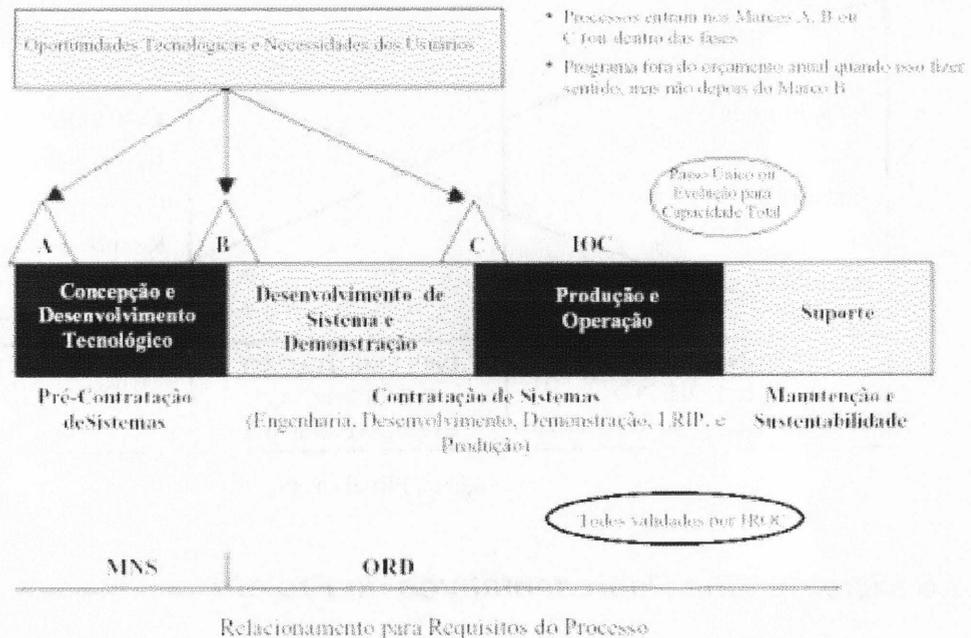


Figura 3

Suparte – essa fase é parte do ciclo de vida do produto, mas é certamente um gerenciamento contínuo. Vários projetos podem ser conduzidos durante essa fase para desenvolver capacidades e correção de defeitos, etc.

Construção, descreve o ciclo de vida de construção.

- Viabilidade - formulação do projeto, estudos de viabilidade e formulação e aprovação da estratégia. Uma decisão de continuidade (go / no-go) do projeto faz parte da finalização desta fase.
- Planejamento e Projeto - projeto básico, custo e cronograma, termos e condições contratuais, e planejamento detalhado. A maioria dos contratos são fechados ao final desta fase.
- Produção - fabricação, entrega, obras civis, instalação e teste. As instalações estão substancialmente completas ao final desta fase.
- Adaptação e Lançamento - teste final e manutenção. As instalações estão em plena operação ao final desta fase.

1.1.6 Histórico e acompanhamento do software

Segundo Roger Presmann Rp.(95) em seu livro, engenharia de software, durante as três primeiras décadas da era do computador, o principal desafio era desenvolver um hardware que reduzisse o custo de processamento e armazenagem de dados. Ao longo da década de 1980, avanços na microeletrônica resultaram em maior poder de computação a um custo cada vez mais baixo. Hoje, o problema é diferente. O principal desafio durante a

década de 1990 é melhorar a qualidade (e reduzir o custo) de soluções baseadas em computador – soluções que são implementadas com software.

O poder de um computador *mainframe* da década de 1980 agora está à disposição sobre uma escrivadinha. As assombrosas capacidades de processamento e armazenagem do moderno hardware representam um grande potencial de computação. O software é o mecanismo que nos possibilita aproveitar e dar vazão a esse potencial.

Quando se iniciava a década de 1980, uma reportagem de primeira página da revista *Business Week* apregoava a seguinte manchete: “Software: A Nova Força Propulsora”. O software amadurecera – tornara-se um tema de preocupação administrativa. Em meados da década de 1980, uma reportagem de capa de *Fortune* lamentava “Uma Crescente Defasagem de Software” e, ao final da década, a *Business Week* avisava os gerentes sobre a “Armadilha do Software – Automatizar ou Não”. No começo da década de 1990, uma reportagem especial da *Newsweek* perguntava: “Podemos Confiar em Nosso Software?” enquanto o *Wall Street Journal* relacionava as “dores de parto” de uma grande empresa de software com um artigo de primeira página intitulado “Criar Software Novo: Era Uma Tarefa Agonizante...”. Essas manchetes, e muitas outras iguais a elas, eram o anúncio de uma nova compreensão da importância do software de computador – as oportunidades que ele oferece e os perigos que apresenta.

Com uma abordagem aos fatores relevantes ao gerenciamento de projetos em TI, é importante compreender que *Software*: é todo o programa que faz o funcionamento e operações em computadores e máquinas afins, e *Hardware*: é toda a parte que pode ser “tocada” ou seja o *equipamento físico* onde os programas ou softwares funcionam em seu interior.

O software agora ultrapassou o hardware como a chave para o sucesso de muitos sistemas baseados em computador. Seja o computador usado para dirigir um negócio, controlar um produto ou capacitar um sistema, o software é um fator que *diferencia*. A inteireza e a oportunidade das informações oferecidas pelo software (e bancos de dados relacionados) diferenciam uma empresa de suas concorrentes. O projeto e a capacidade de ser “amigável ao ser humano” (*human-friendly*) de um produto de software diferenciam-nos dos produtos concorrentes que tenham função idêntica em outros aspectos. A inteligência e a função oferecidas pelo software muitas vezes diferenciam dois produtos de consumo ou indústrias idênticas. É o software que pode fazer a diferença.

1.1.7 O Papel evolutivo do software

Segundo Roger Presmann Rp.(95), o contexto em que o software foi desenvolvido está estreitamente ligado a quase cinco décadas de evolução dos sistemas computadorizados. O melhor desempenho de hardware, menor tamanho e custo mais baixo precipitaram o aparecimento de sistemas baseados em computadores mais sofisticados. Mudam-se dos processadores a válvula para os dispositivos microeletrônicos que são capazes de processar 200 milhões de instruções por segundo.

Em livros populares sobre a “revolução do computador”, Osborne caracterizou uma “nova revolução industrial”, Toffler chamou o advento da microeletrônica de “a terceira onda de mudança” na história humana e Naisbitt previu que a transformação de uma sociedade industrial em uma “sociedade de informação” terá um profundo impacto sobre

nossas vidas. Feigenbaum e McCorduck sugeriram que a informação e o conhecimento (controlados por computador) serão o foco principal de poder no século XXI, e Stoll argumentou que a “comunidade eletrônica” criada por redes e software é a chave para a troca de conhecimentos em todo o mundo.

1.1.8 Aplicações do Software

Roger Pressman Rp.(95), afirma que o software pode ser aplicado a qualquer situação, em que um conjunto previamente especificado de passos procedimentais (isto é, um algoritmo) tiver sido definido (notáveis exceções a essa regra são o software de sistema especialista e o software de rede neural). O conteúdo de informação e a determinância são fatores importantes na determinação da natureza de um aplicativo. A palavra *conteúdo* refere-se ao significado e à forma das informações que entram e saem. Por exemplo, muitas aplicações comerciais fazem uso de dados de entrada altamente estruturados (um banco de dados) e produzem “relatórios” formatados. O software que controla uma máquina automatizada (por exemplo, um torno de controle numérico) aceita itens de dados distintos com estrutura limitada e produz comandos de máquina individuais em rápida sucessão.

Determinância de informação refere-se à previsibilidade da ordem e da oportunidade da informação. Um programa de análise de engenharia aceita dados que têm uma ordem predefinida, executa o(s) algoritmo(s) de análise sem interrupções e produz dados resultantes em relatório ou em formato gráfico. Tais aplicações são determinadas. Um sistema operacional multi-usuário, por outro lado, aceita entradas que têm conteúdo variado e regulagem de tempo arbitrária, executa algoritmos que podem ser interrompidos por condições externas e produz saída que varia como uma função do ambiente e do tempo. Aplicações com essas características são indeterminadas.

Segundo Rp.(95), desenvolver categorias genéricas para as aplicações de software é uma tarefa um tanto difícil. À medida que a complexidade do software cresce, desaparece a clara divisão em compartimentos. As seguintes áreas de software indicam a amplitude das aplicações potenciais:

Software Básico Software básico é uma coleção de programas escritos para dar apoio a outros programas. Alguns tipos de software básico (por exemplo, compiladores, editores e utilitários de gerenciamento de arquivos) processam estruturas de informação complexas, mas determinadas. Outras aplicações de sistema (por exemplo, componentes do sistema operacional, *drivers*, processadores de telecomunicações) processam dados amplamente indeterminados. Tanto num como noutro caso, a área do software básico é caracterizada por forte interação com o hardware de computador, intenso uso por múltiplos usuários, operações concorrentes que exigem escalonamento (*schedule*), compartilhamento de recursos e sofisticada administração do processo, estruturas de dados complexas e múltiplas interfaces externas.

Software de Tempo Real Um software que monitora / analisa / controla eventos do mundo real é chamado software de *tempo real*. Entre os elementos do software de tempo real incluem-se um componente de coleta de dados que obtém e formata as informações provenientes de um ambiente externo, um componente de análise que transforma as

informações conforme a aplicação exige, um componente de controle /saída que responde ao ambiente externo e um componente de monitoração que coordena todos os demais componentes de forma que a resposta em tempo real (que, tipicamente, varia de 1 milissegundo a 1 minuto) possa ser mantida. Deve-se notar que o termo “tempo real” difere de “interativo” ou *time-sharing* (tempo compartilhado). “Um sistema de tempo real deve responder dentro de restrições de tempo estritas. O tempo de resposta de um sistema interativo (ou *time-sharing*) pode ser normalmente ultrapassado sem resultados desastrosos.

Software Comercial O processamento de informações comerciais é a maior área particular de aplicação de software. Distintos “sistemas” (por exemplo, folha de pagamentos, contas a pagar e a receber, estoques etc.) evoluíram para o software de sistemas de informação administrativa (MIS) que dão acesso a um ou mais bancos de dados contendo informações comerciais. As aplicações dessa área reestruturam os dados de uma forma que facilita as operações comerciais e as tomadas de decisões administrativas. Além da aplicação de processamento de dados convencional, as aplicações de software comerciais também abrangem a computação interativa (por exemplo, processamento de transações em pontos-de-venda).

Software Científico e de Engenharia O software científico e de engenharia tem sido caracterizado por algoritmos de processamento de números. As aplicações variam da astronomia à vulcanologia, da análise de fadiga, mecânica de automóveis à dinâmica orbital de naves espaciais recuperáveis e da biologia molecular à manufatura automatizada. Porém, novas aplicações dentro da área identificação de engenharia estão-se afastando dos algoritmos numéricos convencionais.

O projeto auxiliado por computador (CAD), simulação de sistemas e outras aplicações interativas começaram a assumir características de tempo real e até mesmo de sistema básico.

Software Embutido Produtos inteligentes têm-se tornado comuns em quase todo o mercado industrial e de consumo. O software embutido (*embedded software*) reside na memória só de leitura (*read-only*) e é usado para controlar produtos e sistemas para os mercados industriais e de consumo. O software embutido pode executar funções muito limitadas e particulares (por exemplo, controle de teclado para fornos de microondas) ou oferecer recursos funcionais de controle significativos (por exemplo, funções digitais em automóveis, tais como controle de combustível, mostradores no painel, sistemas de freio etc.).

Software de Computador Pessoal O mercado de software para computadores pessoais entrou em efervescência na última década. Processamento de textos, planilhas eletrônicas, computação gráfica, diversões, gerenciamento de dados, aplicações financeiras pessoais e comerciais, redes externas ou acesso a bancos de dados são apenas algumas das centenas de aplicações. De fato, o software de computador pessoal continua a representar os mais inovadores projetos de interface com seres humanos de toda a indústria de software.

Software de Inteligência Artificial O software de inteligência artificial (Artificial Intelligency – AI) faz uso de algoritmos não-numéricos para resolver problemas complexos que não sejam favoráveis à computação ou à análise direta. Atualmente, a área de AI mais

ativa é a dos *sistemas especialistas*, também chamados *sistemas baseados em conhecimento*. Porém, outras áreas de aplicação para o software de AI são o reconhecimento de padrões (voz e imagem), jogos e demonstração de teoremas. Nos últimos anos, desenvolveu-se um novo ramo do software de AI, chamado *redes neurais artificiais*. Uma rede neural simula a estrutura dos processos cerebrais (a função do neurônio biológico) e pode, por fim, levar a uma nova classe de software que consegue reconhecer padrões complexos e aprender com a “experiência” passada.

2. Gerenciamento de Projetos (GP)

A atividade de Gerenciamento de Projetos tem tido ultimamente grande desenvolvimento na sua utilização. A competitividade necessária aos negócios de hoje obrigou as empresas a procurarem técnicas capazes de acelerar o lançamento de produtos e serviços, reduzir o custo, melhorar os controles. Segundo Dalton Valeriano Dv.(95), em seu livro *Gerencia em Projetos*, a gerência, administração e gestão estão intimamente ligadas, sendo quase sinônimas, em qualquer dicionário. Tanto que o “Aurélio” tem apenas um verbete para gerente, no qual recorre aos verbos gerir e administrar: “Gerente: que ou quem gere ou administra negócios, bens ou serviços”.

Estas palavras dizem respeito ao processo definido no parágrafo anterior. Mas é de toda a conveniência estabelecer um critério de aplicação no que se refere a projetos. Aliás, este item está procurando deixar clara e bem-estabelecida uma prática que já vem tendo contornos mais definidos no campo do gerenciamento de programas e projetos.

Assim, pode-se compreender que todos os termos derivados de administrar referem-se ao nível da organização. Neste campo situam-se os problemas típicos das organizações: finanças (contabilidade, taxas e impostos etc.), pessoal (efetivos, contratações, direitos e deveres etc.), patrimônio (imóveis, máquinas, veículos) etc.

Os termos derivados da palavra gerenciar referem-se às ações no nível do projeto: planejamento do projeto, controle do projeto, orçamento, cronograma etc.

Aqueles derivados de gerir referem-se a parcelas das atribuições do gerente do programa / projeto. São partes do gerenciamento delegadas pelo gerente, das quais este se destacam o que se chama de gestões específicas: gestão da documentação, gestão de interfaces, gestão de riscos, gestão da qualidade, gestão da configuração, gestão ambiental, gestão de programas de auditoria.

Por sinal, as quatro últimas são objeto e títulos de normas ISO 9000, 10000 e 14000. Estas mesmas normas, ao se referirem à responsabilidade das organizações, usam a expressão “alta administração” ou “administração”, reservando a palavra gestão a campos específicos da administração e da gerência de projetos: gestão da qualidade, gestão da configuração, entraremos em maiores detalhes em qualidade de software.

Segundo Dv.(95), evidentemente há zonas cinzentas que devem ser definidas à luz das conveniências em cada caso: controle, contratos, entre outros, tanto podem estar no âmbito da organização como no do projeto conforme acordo entre a organização e o projeto.

Deve-se adiantar, neste ponto, que muitas funções na organização têm a denominação de gerência (de departamentos ou outros órgãos internos), distinguindo-se, portanto, dois tipos de gerentes: os chamados gerentes *departamentais*, *funcionais* ou de linha (fixos, permanentes) e os gerentes de projeto (temporários, transitórios). Segundo

Kezner a *Gerência de Projetos* é a aplicação de conhecimentos, habilidades, e técnicas para projetar atividades que visem atingir os requerimentos do projeto. O Gerenciamento do Projeto é acompanhado através do uso de processos tais como: iniciação, planejamento, execução, controle e encerramento.

A equipe de projeto gerência o trabalho do projeto que tipicamente envolve:

- Demandas concorrentes: escopo, tempo, risco e qualidade.
- Partes envolvidas com diferentes necessidades e expectativas
- Identificação de requerimentos

É importante notar que muitos processos dentro da gerência de projetos são naturalmente iterativos. Isto é, em parte, devido a existência e a necessidade da elaboração progressiva no projeto durante todo o ciclo de vida do projeto, e quanto mais você conhece acerca do seu projeto, melhor você é capaz de gerenciá-lo.

O termo *gerência de projetos* é algumas vezes usado para descrever uma abordagem organizacional para gerenciamento dos processos operacionais contínuos.

Esta abordagem, mais conhecida como *gerência por projetos*, trata muitos aspectos dos serviços continuados como projetos, objetivando aplicar também a eles, os conceitos de gerência de projetos. Embora seja óbvio que o conhecimento de gerência de projetos é essencial para uma organização que aplica a gerência por projetos, uma discussão detalhada dessa abordagem, está fora do escopo deste documento.

2.1.1 Atribuições do Gerente de Projeto

Segundo Victor Roldão Vr (04) em seu livro gestão de projetos, o gerente de projeto deve comportar-se como um empresário, que tem de:

- Planejar o que fazer e quando fazer
- Recrutar a equipe
- Organizar a equipe
- Orçar o projeto
- Realizar o projeto
- Fazer a entrega
- Dissolver a equipe

Para o que tem de tomar:

1. Decisões Técnicas

- Direção da concepção
- Seleção de equipamentos a usar
- Identificação do tipo e âmbito dos testes

2. Decisões Comerciais

- Decisões de "Fazer ou Subcontratar"
- Seleção de Subcontratantes ou de vendedores

3. Decisões Administrativas

Seleção do Pessoal

Programação de recursos (pessoal, equipamentos...)

4. Decisões Financeiras

- Determinação de gasto dos fundos orçamentados

Esta autoridade tem como contrapartida um conjunto de responsabilidades, de que se destacam:

- Manter a equipe dinâmica e conseqüente
- Realizar despesas criteriosas
- Manter as despesas com pequenos desvios em relação ao orçamento
- Evitar situações críticas de ruptura financeira
- Manter os objetivos visíveis
- Evitar erros graves
- Controlar a seqüência do planejamento e desvios emergentes

2.1.2 Gerenciamento de Projetos em Software

O Processo de Gerencia:

Segundo Rp.(95), antes que um projeto possa ser planejado, os objetivos e o escopo devem ser estabelecidos, soluções alternativas devem ser consideradas e as restrições administrativas e técnicas, identificadas. Sem essas informações, é impossível definir estimativas de custo razoáveis (e precisas), uma divisão realística.

O desenvolvedor de software e o cliente devem reunir-se para definir os objetivos e o escopo do projeto. Em muitos casos, essa atividade ocorre como parte do processo de engenharia de sistema. Os objetivos identificam as metas globais do projeto sem levar em consideração como essas metas serão atingidas. O escopo identifica as funções primárias que o software deve realizar e, o que é mais importante, tenta *delimitar* essas funções de uma forma quantitativa.

Logo que os objetivos e o escopo do projeto forem compreendidos, soluções alternativas serão consideradas. Ainda que poucos detalhes sejam discutidos, as alternativas possibilitam que gerentes e profissionais selecionem uma abordagem “melhor”, dadas as restrições impostas por prazos de entrega, orçamento, disponibilidade de pessoal, interfaces técnicas e uma infinidade de outros fatores.

Gerencia de Projeto de Software:

A gerência de projetos é a primeira camada do processo de engenharia de software. Se chama *camada*, em vez de etapa ou atividade, porque ela abrange todo o processo de desenvolvimento, do começo ao fim.

Para conduzir um projeto de software bem-sucedido, devemos compreender o escopo do trabalho a ser feito, os riscos em que incorreremos, os recursos exigidos, as tarefas a serem executadas, os marcos de referência a serem acompanhados, o esforço (custo) despendido e a programação a ser seguida. A gerência de projetos de software oferece essa compreensão. Ela começa antes do trabalho técnico, prossegue à medida que o software se desenvolve do modelo conceitual para a realidade e encerra somente quando o software se torna obsoleto e é aposentado.

Uma vez que a gerência de projetos de software é tão importante para o sucesso de um projeto, seria razoável presumir-se que todos os gerentes de projetos entendem como colocá-la em prática e que todos os profissionais entendem como trabalhar dentro dos limites estabelecidos por ela. Infelizmente, muitos não a entendem. Nas seções que se seguem, apresentamos uma visão geral dos elementos-chave da gerência de projetos de software.

2.1.3 Processos e Sistemas nas organizações para projetos

Ao se descrever ou estudar qualquer assunto, é comum fazê-lo segundo uma determinada abordagem (ou enfoque), isto é, tomando um ponto de vista para realçar um ou mais predicados do objeto ou para dedicar especial atenção a ele. Uma pessoa pode ser estudada como um ser vivo, como um componente de um determinado grupo social (família, escola ou empresa), como uma entidade psicológica ou, ainda, por seu trabalho, como um dos fatores da produção etc. Isto é, ela pode ser vista sob o enfoque biológico, sociológico, psicológico, econômico etc.

Segundo Dv (00) em seu livro, Gerenciamento Estratégico e Administração por Projetos, os assuntos aqui mencionados podem ser estudados como um processo (abordagem processual), ou um sistema (abordagem sistêmica). Assim, o projeto, seu planejamento, a execução e o controle, bem como a documentação técnica do projeto, tanto podem ser vistos sob a abordagem processual como sob a abordagem sistêmica.

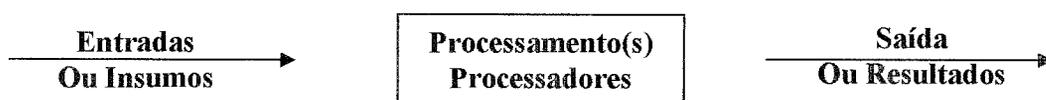
Entende-se por processo um conjunto de recursos e atividades inter-relacionadas (os sub-processos) que transformam insumos em produtos ou resultados. Os insumos são genericamente chamados de entrada (“*input*”) e os produtos, de saída (“*output*”).

Insumo é tudo aquilo que é fornecido ao processo para:

- Utilização (uma informação, uma instrução, um instrumento, um serviço de uma máquina ou trabalho humano etc.);
- Transformação (energia, matéria-prima, por exemplo);
- Consumo (energia, material de escritório etc.);
- Um produto ou resultado pode ser:
 - Tangível (materiais processados, por exemplo)
 - Intangível (uma informação ou conceitos) e ainda:
 - Intencional, isto é, um bem ou serviço, constituindo o objetivo do processo;
 - Não-intencional, aquele que se forma e é debitado como resultado não procurado, geralmente subproduto, alguns destes podendo ser poluentes ou com efeitos indesejáveis e perniciosos.

Evidentemente por se tratar de atividades de pessoas ou grupo de pessoas ou ainda de máquinas, um processo não se realiza sem uma base física, quer seja esta uma pessoa ou um grupo de indivíduos, um conjunto de máquinas ou uma organização etc. Ou seja, o processo tem lugar em um conjunto organizado de componentes, partes, órgãos e ou pessoas.

Correntemente denomina-se processamento o conjunto das ações que realizam as transformações dos insumos em produtos ou resultados e processador é qualquer parte interna do processo que desempenha estas ações. Então, sinteticamente, o processo consiste no emprego de insumos, que passam por um processamento, para produzir resultados. Como demonstrado no fluxo a seguir:



2.1.4 O Conceito Sistêmico em projetos

Por sua vez, sistema é o conjunto de partes, elementos ou componentes inter-relacionados (subsistemas e / ou processos) e que visa à realização de determinados objetivos ou efeitos situados no meio exterior (ou no ambiente) em que está inserido.

Segundo Dv (00), para alcançar os objetivos do sistema, cada subsistema tem seu objetivo próprio e se acha conectado a outro, do qual recebe insumos ou entradas e pode atuar sobre ele, ajustando as saídas desse outro para atender as necessidades próprias, para alcançar o objetivo. Isto acarreta uma interdependência de partes interligadas que atuam entre si para atingir um objetivo definido para o conjunto.

Um sistema é definido:

- Por seu objetivo, resultado ou efeito sobre o ambiente (sua finalidade e relacionamento externo);
- Por seus limites ou fronteiras (seu campo de atuação);
- Pelos subsistemas constitutivos (sua estrutura);
- Pelas funções e inter-relacionamento de seus subsistemas;

Os sistemas podem ser classificados segundo vários critérios, como, por exemplo:

- Concretos (físicos, materiais) /abstratos (conceituais);
- Naturais (espontâneos) /artificiais (intencionalmente feitos pelo ser humano); abertos (interagem com o ambiente)/fechados (desprezível interação);
- Estáticos (sem atividade, em uma dada escala dimensional ou temporal) dinâmicos;
- Inanimados (sem vida)/biológicos, sociais (formados de seres vivos);

Normalmente os sistemas e cada um de seus subsistemas assumem uma modalidade, em cada critério. Veja-se, por exemplo, um sistema que pode ser chamado de “sistema

pessoa-máquina”, por compreender a necessária interação de pessoas e equipamentos para realizar seu objetivo.

Tome-se o sistema aeronave / tripulação / equipe de terra / passageiros. Ele é concreto e artificial (existe fisicamente e é fabricado), tem subsistemas inanimados, sendo alguns dinâmicos (aeronaves) e outros estáticos (aerportos), e também biológicos (as pessoas que os integram) e sociais (a comunidade a que serve) e é aberto, interagindo com o ambiente de várias formas (econômica e socialmente, poluindo o meio ambiente etc.).

Entende-se por ambiente de um sistema tudo o que está situado fora dos limites ou das fronteiras do sistema considerado. Segundo a norma ISO 14001, o meio ambiente é a "circunvizinhança em que uma organização opera, incluindo ar, água, solo, recursos naturais, flora, fauna, seres humanos e suas inter-relações. Neste contexto, circunvizinhança estende-se do interior das instalações para o sistema global”.

O ambiente tanto é afetado pelos sistemas que encerra como também atua sobre eles. Assim, se considerar uma empresa industrial como um sistema, o ambiente influirá sobre ela segundo vários aspectos ou agentes como os seguintes:

- Governo (legislação: direitos, deveres);
- Clientes (necessidades: quantidade, qualidade, prazos);
- Competidores (concorrência no mercado);
- Fornecedores (condições de fornecimento: quantidade, qualidade, prazos);
- Tecnologia (disponibilidade externa: nível de adequação, evolução);
- Ecologia (restrições, influências);
- Sociedade (padrões morais, éticos, religiosos, políticos etc.);
- Associações de Classe, Sindicatos (reivindicações, conflitos, greves) etc.

Segundo Dv (00), cada componente do sistema, quando visto isoladamente, pode ser considerado um outro sistema de menor nível, enquanto o sistema original passa a seu ambiente imediato. A menor parte de um sistema ou subsistema é chamada de processo unitário, aquele que transforma entradas em saídas, como visto anteriormente.

Por outro lado, cada sistema é parte de outro sistema maior, no qual ele está situado, e assim por diante. Desta forma, os sistemas exibem uma espécie de hierarquia que se estabelece pela existência de componentes dentro de componentes, como se vê na figura 4 a seguir. Nesta, acha-se um trecho de um grande sistema S, dele vendo-se um subsistema inteiro S1, composto, por sua vez, de outros quatro subsistemas e uma parte do sistema S2. As setas indicam as interações entre os componentes.

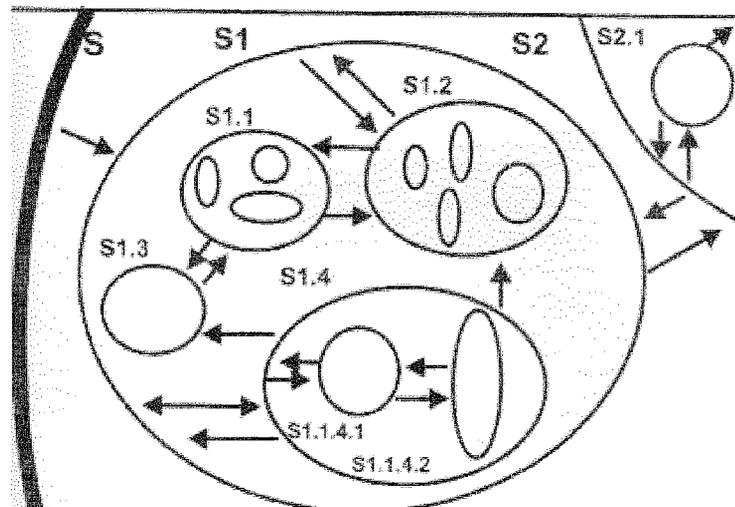


Figura 4

2.1.5 Hierarquia dos sistemas e suas interfaces

Verifica-se que a abordagem sistêmica é mais abrangente que a processual, pois implica o conhecimento ou a definição do efeito ou da finalidade do resultado sobre o meio ambiente, e deste sobre o sistema. O conceito de processo limita-se ao estudo do comportamento interno dos agentes que trabalham os insumos (não interessando de onde vêm), para obter um resultado (não interessando para quê), enquanto o de sistema preocupa-se também com as interações com o ambiente.

As mais diversas ciências costumam classificar os sistemas quanto à interação com o ambiente, em sistemas abertos e sistemas fechados, respectivamente aqueles com e sem interação com o exterior. Ora, os sistemas fechados, como mencionado, são uma abstração, absolutamente teóricos, pois não se conhecem sistemas isolados do ambiente.

Segundo Dv (00), é importante ressaltar a alternância do processo versus sistema o que podem surgir algumas questões: para que servem estes dois enfoques, isso e o de sistema? Qual deve ser utilizado? Qual o melhor?

Uma boa resposta consiste em utilizar os dois, cada qual em seu lugar, a seu m sua finalidade, e isto pode ser esclarecido por meio de um simples exemplo a elaboração de um documento técnico (um relatório, digamos).

Sobre o ponto de vista como processo, esta atividade utiliza como insumos alguns dados selecionados durante alguma parte da execução do projeto. O ente principal consiste em transformar estes dados em informações, sendo processador mais importante, auxiliado por outros, como "softwares", um como processo, o autor deve estar preocupado em bem empregar e produzir seu resultado: um relatório que deverá satisfazer necessidades ou de alguma equipe do projeto, o cliente etc. A esta altura convém adotar sistêmico.

2.1.6 Gestão do conhecimento aplicado a Sistemas

Como sistema, deve-se considerar que o documento fará parte de um conjunto maior, a documentação do projeto, seu ambiente mais imediato. Este relatório certamente produzirá efeitos em seu projeto (o ambiente em que se situa): as informações que produziu serão utilizadas como insumos em outros processos ou sub-sistemas. Por outro lado, está condicionado a ele, sujeito a algumas regras: de identificação do documento, de trato de originais, de cópias, distribuição, arquivamento etc.

Sua exatidão depende do insumo: os dados de um laboratório, digamos. Agora não é um processo isolado.

A mudança de focalização, mostrada em um singelo exemplo, baseado na elaboração de um documento técnico, deve ser repetida muitas vezes durante qualquer tarefa. Outro exemplo? Observe um pássaro silvestre que busca seu alimento no nosso jardim: ele alterna, em poucos segundos, sua atenção para o processo (a alimentação) e para o sistema (o ambiente e seus predadores).

É importante se ter um papel de observador, como se estivéssemos ajustando o zoom de uma máquina para se ter sucessivamente a visão de detalhe e a visão de conjunto.

A capacidade de poder abarcar um panorama amplo, como o do sistema em que se trabalha, está sendo cada vez mais requerida atualmente, no gerenciamento e na execução de projetos, como, de resto, em muitas outras atividades.

Segundo Dv.(00), um estudo sobre o corpo de conhecimentos de uma organização reconhece *quatro níveis de intelecto profissional* que operam em uma organização, aqui resumidos:

Conhecimento cognitivo ("*know-what*"), que é o domínio básico de uma disciplina que o profissional adquire por meio de treinamento e certificação. É um conhecimento essencial, mas geralmente está longe da suficiência para o sucesso comercial.

Habilidades avançadas ("*know-how*"), que traduz o "*conhecimento Livresco*" em cativa execução. A habilidade de aplicar regras de uma disciplina aos problemas complexos do mundo real é a mais difundida habilidade profissional capaz de criar valor.

Compreensão de sistemas ("*know-why*"), é o profundo conhecimento da trama de relacionamento causa-efeito subjacente a uma disciplina. Ele permite ao profissional mover-se para além da execução de tarefas para resolver problemas mais complexos – e criar extraordinário valor. Os profissionais com "*know-why*" podem ver sutis interações e conseqüências não-intencionais.

Criatividade automotivada ("*cure-why*"), que consiste na vontade, na motivação e na adaptabilidade para o sucesso. Os grupos altamente motivados e criativos comumente superam os grupos que dispõem de maiores recursos físicos e financeiros.

O gerente de projeto e cada membro de sua equipe precisam ter uma competente capacidade de ver os detalhes nos níveis em que devem operar, ao mesmo tempo em que dividem o ambiente em que trabalham. Sem esta capacidade de ajustamento de foco, os principais aspectos da gerência e da execução de projetos, nos dias atuais, não serão sequer compreendidos, e muito menos praticados. O gerente de projeto e sua equipe, além do profundo conhecimento de seus respectivos papéis no projeto precisam ter esta visão sistêmica para bem poderem compreender os pontos e atuação sobre o projeto como entre outros e deles participar ativamente:

- o “guarda-chuva” da engenharia de sistemas, sob o qual os projetos atuam;
- o exercício da engenharia simultânea;
- o ambiente e a prática da qualidade total;
- o minucioso trato do desdobramento da função qualidade;
- a formação de equipes, especialmente as auto-gerenciadas etc.

Para os desprovidos desta visão sistêmica, estas áreas, entre outras igualmente importantes, serão universos estranhos e até mesmo hostis.

Entretanto é importante estabelecer os limites mais convenientes para a visão de conjunto e para a visão de detalhes. No primeiro caso, um exagerado e desnecessário campo de visão certamente prejudicará a compreensão do posicionamento do sub-sistema em estudo.

No caso inverso, com a atenção agora voltada para o processo, o excessivo e inútil detalhamento trará à tona o supérfluo que prejudicará o trabalho. Desta forma, nas varreduras e nas sucessivas alternâncias do zoom, é imprescindível encontrar o limite superior e o inferior, necessários e suficientes para a análise do processo e sua síntese no ambiente.

Certamente o limite superior não será o da ampla perspectiva de uma galáxia aparentemente estática e nem o limite inferior vai ser aquela minuciosa ilusão dos fugazes quarks.

3. Escopo em Projetos

Segundo Dv (00), o escopo constitui uma “descrição documentada de um projeto quanto a seus resultados, sua abordagem e conteúdo” ou seja, ele é um resumo das partes capitais do projeto e de suas esperadas conseqüências, de forma a permitir uma compreensão do que se pretende fazer e com que finalidade. Ele é considerado um dos nove módulos do Corpo de Conhecimento do Gerenciamento de Projeto, juntamente com a integração, o prazo, custo, qualidade, risco, recursos humanos, compras e comunicações^o.

O escopo deve conter as seguintes partes:

“Os resultados do projeto: o que será criado, em termos de forma e tamanho, geografia, quantidade, especificações de desempenho técnico e operacional, características de custos, utilidade, e assim por diante”.

A metodologia a empregar: as tecnologias (novas ou existentes?), os insumos internos e externos, a descrição das interfaces ou limites entre o projeto e seu ambiente.

O conteúdo do projeto: o que está incluído e excluído do trabalho a ser executado e a descrição das interfaces ou limites entre as tarefas do projeto e destas com outras relacionadas com os resultados do projeto ou com seu ambiente.

O escopo é elaborado durante o planejamento preliminar e pode sofrer modificações durante a etapa de proposta e negociação com a organização, com os patrocinadores, com os clientes etc. Entretanto, uma vez aprovado o projeto, tanto o escopo como o objetivo tornam-se peças que só podem ser alteradas com a formal concordância de todos que tornaram parte na aprovação.

3.1.1 Gerência de Escopo em projetos

De acordo com o PMBOK (02), o Gerenciamento do Escopo do Projeto é composto dos “processos para garantir que o projeto inclua todo o trabalho exigido, e somente o trabalho exigido, para completar o projeto com sucesso”

As finalidades do Gerenciamento do Escopo do Projeto incluem a definição do trabalho necessário para concluir o projeto, servir como guia (ou ponto de referência) para determinar que trabalho não está incluído (ou não é necessário) no projeto.

O escopo é o “foco” do projeto. O escopo do projeto difere-se do escopo do produto na medida em que o escopo do projeto define o trabalho necessário para fazer o produto, e o escopo do produto define os recursos (atributos e comportamentos) do produto que está sendo criado.

Os projetos não desviam frequentemente do foco de negócios da empresa, e geralmente estão relacionados à sua atividade fim.

Segundo o PMBOK a gerência do escopo do projeto, inclui os processos requeridos para assegurar que o projeto inclua todo o trabalho necessário, e tão somente o trabalho necessário, para complementar de forma bem sucedida o projeto. A preocupação fundamental compreende definir e controlar o que está ou não incluído no projeto.

- **Iniciação** – comprometer a organização a iniciar a próxima fase do projeto.
- **Planejamento do Escopo** – desenvolver uma declaração escrita do escopo como base para decisões futuras do projeto.
- **Detalhamento do escopo** – subdividir os principais subprodutos do projeto em componentes menores e mais manejáveis.
- **Verificação do Escopo** – formalizar a aprovação do escopo do projeto.
- **Controle de Mudanças do Escopo** – controlar as mudanças do escopo do projeto.

Estes processos interagem uns com os outros e também com os processos das demais áreas de conhecimento. Cada processo pode envolver esforço de um ou mais indivíduos ou grupos de indivíduos dependendo das necessidades do projeto. Cada processo geralmente ocorre pelo menos uma vez em cada fase do projeto.

Embora os processos sejam aqui apresentados como elementos discretos e interfaces bem definidas, na prática eles podem se sobrepor e interagir de outras maneiras. No contexto de projeto, o termo escopo deve se referir a:

- Escopo do produto – aspectos e funções que devam ser incluídos no produto ou serviço.

- Escopo do projeto – o trabalho que deve ser feito com a finalidade de entregar um produto de acordo com os aspectos e as funções especificados.

Os processos, ferramentas e técnicas usados para gerenciar o escopo do *produto* variam conforme a área de aplicação e são usualmente definidos como parte do ciclo de vida do projeto.

Um projeto consiste em um único produto, mas esse produto pode incluir elementos subsidiários, cada um deles com seu próprio e distinto, porém interdependente, escopo de produto. Por exemplo, um novo sistema de telefonia geralmente inclui quatro elementos subsidiários – hardware, software, treinamento e implementação.

A conclusão do escopo do *produto* é mensurada contra as exigências, enquanto a conclusão do escopo do *projeto* é mensurada contra o plano. Ambos os tipos de gerência de escopo devem ser bem integrados para garantir que o trabalho do projeto resulte na entrega do produto especificado.

3.1.2 Início de Escopo

A iniciação é o processo de reconhecimento formal que um novo projeto existe ou que um projeto existente deve continuar em sua próxima fase. A iniciação formal liga o projeto com o trabalho em execução na organização. Em algumas organizações um projeto é formalmente iniciado somente depois da conclusão de um estudo de viabilidade, de um plano preliminar ou de qualquer outra forma equivalente de análise que foi iniciada separadamente.

Segundo o PMBOK (02), alguns tipos de projetos, especialmente projetos de serviços internos e projetos para o desenvolvimento de novos produtos, são iniciados informalmente e alguma quantidade limitada de trabalho é feita para assegurar as aprovações necessárias para a iniciação formal. Os projetos são, tipicamente, autorizados como resultado de uma ou mais das seguintes situações:

Uma demanda de mercado (por exemplo, uma companhia de óleo autoriza um projeto para construir uma nova refinaria em resposta à uma escassez crônica de gasolina).

Uma necessidade do negócio (por exemplo, uma companhia de treinamento autoriza um projeto para criar um novo curso para incrementar as receitas).

Um pedido (uma exigência) de cliente (por exemplo, uma empresa pública de energia elétrica autoriza um projeto para construção de uma nova subestação para servir um novo parque industrial).

Um avanço tecnológico (por exemplo, uma firma eletrônica autoriza um novo projeto para desenvolver um jogo para vídeo após a introdução do vídeo cassete).

Uma exigência legal (por exemplo, um fabricante de tintas autoriza um projeto para estabelecer orientações para manuseio de materiais tóxicos).

Esses estímulos podem ser, também, chamados de problemas, oportunidades ou exigências do negócio. O tema central de todos esses termos é que a gerência deve, geralmente, tomar a decisão sobre como responder.

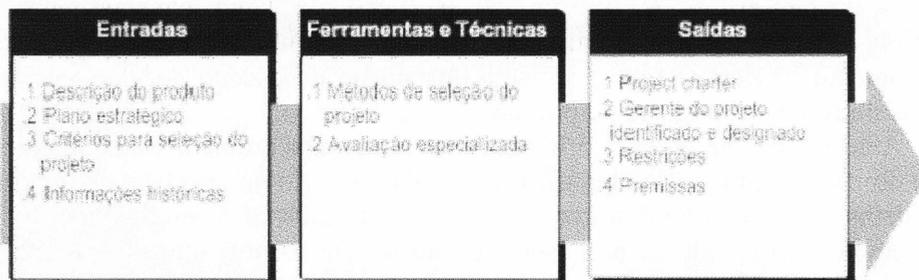


Figura 5

3.1.3 Seleção do projeto pelo escopo

A maioria dos projetos passa por um processo para determinar seu custo e valor. Eles são selecionados com base em diversas condições: oportunidade, necessidade, demandas do cliente, entre outros. O escopo do projeto deve ser criado para dar suporte à finalidade e à necessidade do projeto.

O processo de seleção baseado no valor percebido normalmente é uma relação custo benefício. Esse tipo de estudo é comparativo, e determina se o projeto vale a pena ser feito.

Outro modelo de seleção é o método de otimização restrito. Esses métodos são utilizados mais frequentemente em projetos grandes, e se concentram em equações matemáticas relativamente complexas.

3.1.4 Planejamento do Escopo

O planejamento do escopo do projeto é feito através do processo chamado “plano de gerenciamento do escopo”.

Para determinar qual é o escopo do projeto, existe muito planejamento do escopo. Para isso, tanto o gerente quanto a equipe precisam ter uma visão unificada sobre quais são os componentes do projeto, dos seus requisitos, da expectativa dos stakeholders do projeto, e de onde o projeto se encaixa nas necessidades de negócio destes stakeholders.

O resultado dos processos de planejamento de escopo é a declaração de escopo. A declaração de escopo diz o que está dentro e o que está fora do projeto, de maneira clara e sem ambigüidades.

É importante que a declaração de escopo seja bem-feita, e que haja acordo sobre ela. Quando a declaração de escopo estiver pronta, a equipe do projeto, os stakeholders, o patrocinador do projeto e o gerente de projetos não deverão mudar o escopo – a menos que haja um motivo muito forte que justifique essa mudança (que quase certamente implica em impactos no custo do projeto).

Segundo o PMBOK (02), o planejamento do escopo é o processo de desenvolvimento de uma declaração escrita do escopo como base para decisões futuras do projeto incluindo, em particular, os critérios usados para determinar se o projeto ou fase foi completado com sucesso.

A declaração escrita do escopo é necessária tanto para projetos como para subprojetos. Por exemplo, uma firma de engenharia contratada para projetar uma usina para processamento de petróleo deve ter uma declaração do escopo definindo as fronteiras de seus trabalhos nos subprojetos do projeto.

A declaração do escopo forma as bases para um acordo entre a equipe do projeto e o cliente do projeto através da identificação de objetivos do projeto bem como dos principais subprodutos do projeto. Se todos os elementos da declaração do escopo já estão disponíveis (por exemplo, um pedido de proposta pode identificar os principais subprodutos, o project charter deve definir os objetivos do projeto) esse processo pode envolver um pouco mais que criar fisicamente o documento escrito.

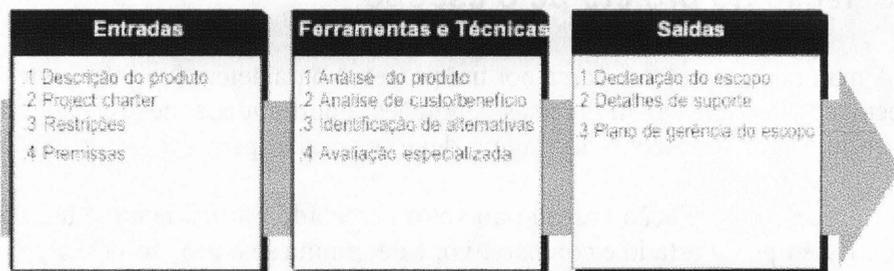


Figura 6

3.1.5 Detalhamento do Escopo

Em consonância ao PMBOK (02), o detalhamento do escopo significa na subdivisão dos principais subprodutos do projeto (conforme identificados na declaração do escopo) em componentes menores e mais manejáveis para se ter condição de:

- Melhorar a precisão das estimativas de custo, tempo e recurso.
- Definir um baseline para medir e controlar o desempenho (desempenho).
- Facilitar uma atribuição clara de responsabilidades.

A definição apropriada do escopo é um ponto crítico para sucesso do projeto. "Quando existe uma definição pobre do escopo, pode ser esperado um custo final do projeto mais alto por causa de inevitáveis mudanças que rompem com o ritmo do projeto, causam retrabalho, aumentam o tempo do projeto e diminuem a produtividade e o moral da força de trabalho".

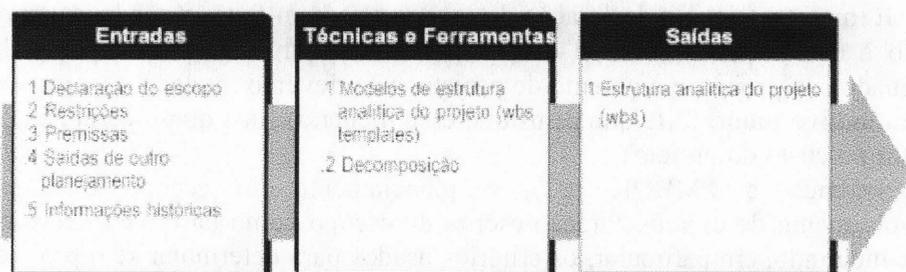


Figura 7

Uma estrutura analítica do projeto - EAP, de um projeto anterior pode ser usada como modelo em um novo projeto. Embora cada projeto seja único, EAP's podem, frequentemente, ser "reusadas" uma vez que a maioria dos projetos se assemelha a um outro em alguma extensão. Por exemplo, a maioria dos projetos de uma determinada organização terá ciclos de vida de projeto iguais ou similares e, conseqüentemente, terá os subprodutos requeridos iguais, ou similares, para cada fase. Modelo EAP:

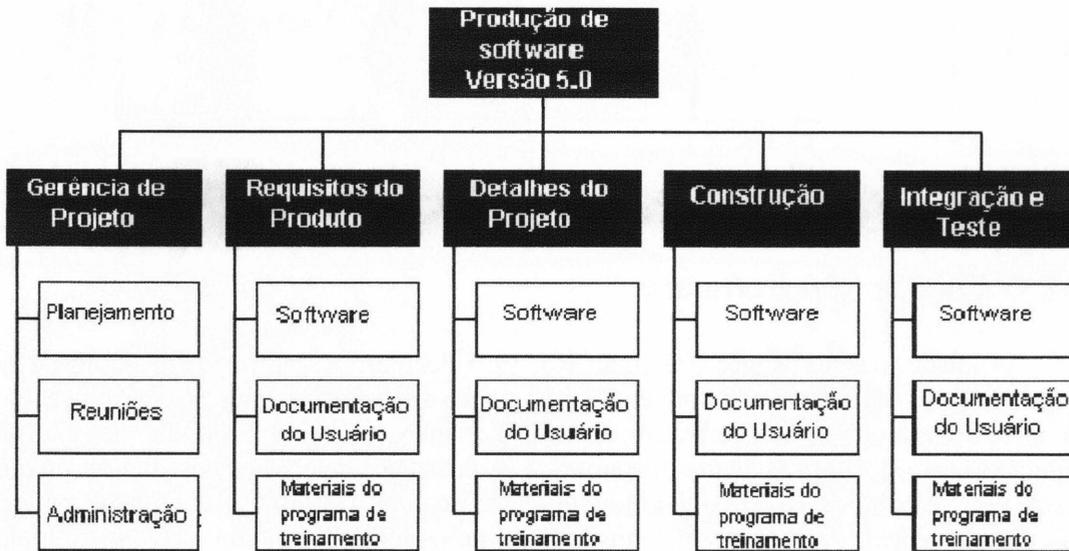


Figura 8

3.1.6 Verificação de Escopo

A verificação do escopo é o processo de formalização do aceite do escopo do projeto pelas partes envolvidas (patrocinador, cliente, freguês, etc). Isto exige uma revisão dos produtos e resultados do trabalho para garantir que tudo foi completado correta e satisfatoriamente.

Se o projeto terminar mais cedo, o processo de verificação do escopo deve estabelecer e documentar o nível e extensão da complexidade. A verificação do escopo difere do controle da qualidade já que é fundamentalmente relacionada com a aceitação do resultado do trabalho enquanto o controle da qualidade é fundamentalmente relacionado com a exatidão dos resultados do trabalho.

3.1.7 Controle de Mudanças do Escopo

Pelo conceito do PMBOK (02), o controle de mudanças do escopo consiste em (a) influenciar os fatores que criam mudanças no escopo para garantir que as mudanças sejam benéficas, (b) determinar que uma mudança no escopo ocorreu, e (c) gerenciar as mudanças reais, quando e se elas ocorrem. O controle das mudanças de escopo deve ser

completamente integrado com os outros processos de controle (controle de prazo, controle de custo, controle de qualidade, e outros). Este item de controle de mudança será melhor descrito em Baseline associado a qualidade em projetos de software.

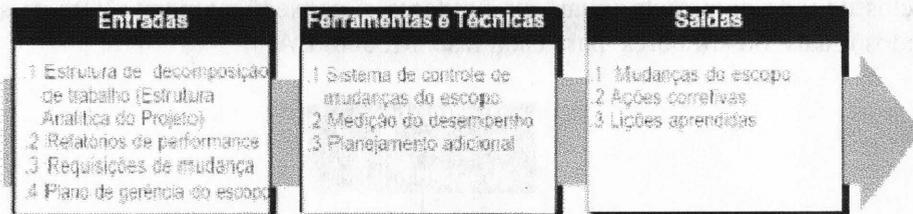


Figura 9

3.1.8 O Escopo em Software

As afirmações de Roger Pressman Rp. (95) denotam que a primeira atividade no planejamento de projetos de software é a determinação do *escopo do software*. A função e o desempenho atribuídos ao software durante o trabalho de engenharia de sistema computadorizado devem ser avaliados para que se possa estabelecer um escopo de projeto que seja claro e compreensível tanto em nível técnico como administrativo. Uma declaração do escopo do software deve ser *delimitada*. Ou seja, os dados quantitativos (por exemplo, número de usuários simultâneos, quantidade de clientes, tempo máximo de resposta permitido) são declarados explicitamente, as restrições e ou as limitações (por exemplo, custo do produto que restringe o tamanho da memória) são anotadas e os fatores mitigantes (por exemplo, algoritmos desejados que são bem compreendidos e estão disponíveis) são descritos.

O escopo do software descreve a função, o desempenho, as restrições, as interfaces e a confiabilidade. As *funções* descritas na declaração do escopo são avaliadas e, em certos casos, refinadas para oferecer mais detalhes antes do início da estimativa. Uma vez que tanto as estimativas de custo como de prazo são funcionalmente orientadas, certo grau de decomposição muitas vezes é útil. As considerações referentes a *desempenho* abrangem os requisitos de processamento e de tempo de resposta. As restrições identificam os limites impostos ao software pelo hardware externo, memória disponível ou outros sistemas existentes.

Como exemplo de função, desempenho e restrições, consideremos um software que deva ser desenvolvido para impulsionar um sistema de classificação por correias transportadoras (Conveyor Line Sorting System – CLSS). A declaração do escopo para o CLSS é a seguinte:

O sistema de classificação por correias transportadoras (CLSS) classifica caixas que se movimentam ao longo de uma linha transportadora. Cada caixa será identificada por um código de barras que contém um número de peça e é classificada num dos seis depósitos ao final da linha. As caixas passarão por uma estação de classificação, a qual contém um leitor de código de barras e um PC. O PC da estação de classificação está ligado a um mecanismo de desvio que classifica as caixas para os depósitos. As caixas passam em ordem aleatória e

são espaçadas uniformemente. A linha move a cinco pés (1,524 m) por minuto. O CLSS é descrito esquematicamente na figura 10 a seguir.

O software do CLSS receberá entradas de informação de um leitor de códigos de barras a intervalos de tempo que condizem com a velocidade da linha transportadora. Os dados do código de barras serão decodificados no formato de identificação de caixas. O software executará uma busca num banco de dados de números de peças contendo no máximo 1.000 entradas para determinar a localização do depósito adequado para a caixa que se encontra atualmente sob o leitor (estação de classificação). A localização do depósito adequá-lo será passada a um mecanismo de desvio que posicionará as caixas no depósito conveniente.

Uma lista será usada para rastrear as posições de desvio para cada caixa à medida que ela passar pela estação de classificação. O software do CLSS receberá também entrada de um tacômetro a impulsos que será usado para sincronizar o sinal de controle ao mecanismo de desvio. Baseado no número de impulsos que será gerado entre a estação de classificação e o mecanismo de desvio, o software produzirá um sinal de controle ao mecanismo de desvio para posicionar adequadamente a caixa.

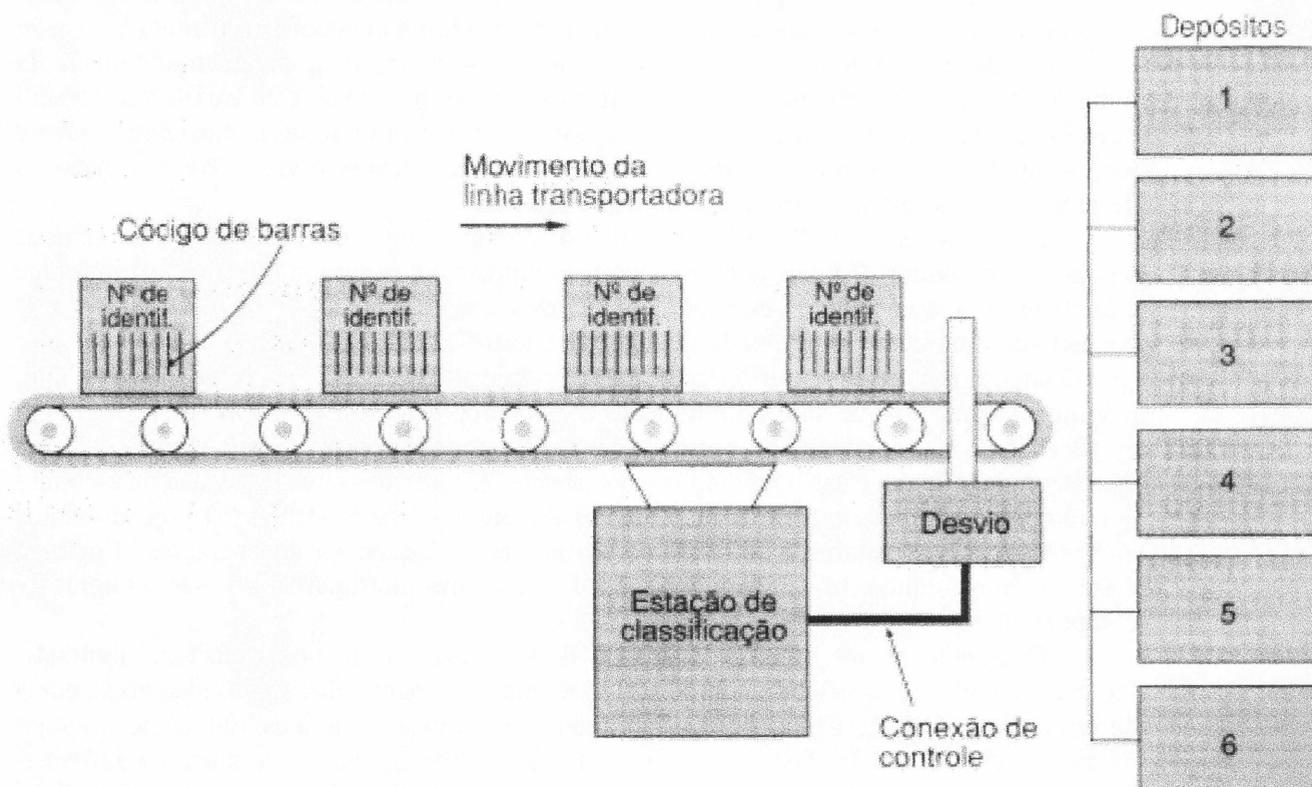


Figura 10

O planejador de projetos examina a declaração do escopo e extrai todas as funções de software importantes. Esse processo, denominado *decomposição*, resulta no seguinte:

- ler a entrada do código de barras;

- ler o tacômetro de impulso;
- decodificar os dados das peças;
- executar busca em banco de dados;
- determinar a localização do depósito;
- produzir sinais de controle para o mecanismo de desvio.

Nesse caso, o desempenho é ditado pela velocidade da linha transportadora. O processamento de cada caixa deve ser concluído antes que a caixa seguinte chegue ao leitor de códigos de barras. O software do CLSS sofre restrições do hardware que ele deve acessar [o leitor de códigos de barras, o mecanismo de desvio, o computador pessoal (PC)], a memória disponível e a configuração global da linha transportadora (caixas espaçadas uniformemente).

A função, o desempenho e as restrições devem ser avaliadas em conjunto. A mesma função pode precipitar uma diferença de ordem de magnitude no esforço de desenvolvimento quando considerada no contexto de diferentes limites de desempenho. O esforço e o custo exigidos para desenvolver o software do CLSS seriam drasticamente diferentes se a função permanecesse a mesma (isto é, a linha transportadora ainda colocasse caixas nos depósitos), mas o desempenho variaria. Por exemplo, se a velocidade média da linha transportadora aumentasse em um fator de 10 (desempenho) e as caixas não fossem mais espaçadas uniformemente (uma restrição), o software tornar-se-ia consideravelmente mais complexo – exigindo, portanto, mais esforço de desenvolvimento. A função, o desempenho e as restrições estão intimamente ligados.

Pelos preceitos de Rp.(95), o software interage com outros elementos do sistema baseado computador. O planejador considera a natureza e a complexidade de cada interface para determinar quaisquer efeitos sobre os recursos, custos e prazos de desenvolvimento. O conceito de interface é interpretado de maneira a ter os seguintes significados: (1) hardware (por exemplo, processador, periféricos) que executa o software e dispositivos (por exemplo, máquinas, *displays*) que são indiretamente controlados pelo software; (2) software que já existe (por exemplo, rotinas de acesso a bancos de dados, pacotes de sub-rotinas, sistema operacional e que deve ser ligado ao novo software; (3) pessoas que fazem uso do software por meio de terminais ou outros dispositivos de entrada / saída (EIS); (4) procedimentos que precedem ou sucedem o software como uma série seqüencial de operações. Em cada caso, a transferência de informações por meio da interface deve ser claramente compreendida.

O aspecto menos preciso do escopo do software é a discussão da confiabilidade. Existem medidas de confiabilidade de software, mas raramente elas são usadas nessa etapa de um projeto. Características de confiabilidade de hardware clássicas, tais como o tempo médio entre as falhas (MTBF, são difíceis de ser traduzidas para o domínio de software. Porém, a natureza geral do software pode determinar considerações especiais para garantir sua confiabilidade. Por exemplo, o software para um sistema de controle de tráfego aéreo ou do Space Shuttle [Ônibus Espacial] (ambos sistemas *human-rated*) não pode falhar; caso contrário, vidas humanas serão perdidas. Um software de sistema de controle de estoques ou de processamento de textos não deve falhar, mas o impacto da falha é consideravelmente menos drástico. Ainda que talvez não seja possível quantificar a confiabilidade do software tão precisamente como gostaríamos na declaração do escopo, podemos usar a natureza do

projeto para nos ajudar a formular estimativas de esforço e de custo para garantirmos a confiabilidade.

Se uma *Especificação de Sistema* tiver sido adequadamente desenvolvida, quase todas as informações exigidas numa descrição do escopo do software estará a disposição e será documentada antes que o planejamento do projeto de software se inicie. Em casos em que uma especificação não tenha sido desenvolvida, o planejador deve assumir o papel de analista de sistemas para determinar atributos e ligações que influenciarão as tarefas de estimativa.

4. Qualidade em Projetos

A qualidade deve ser uma preocupação exercida durante todas as fases do projeto. Durante a concepção as especificações são traçadas em estreita ligação ao cliente, e, durante a implementação os trabalhos são realizados "bem à primeira vez" segundo essa mesma especificação. A atitude preventiva, complementada com um plano de inspeções, e culminando na inspeção final (que precede a entrega da obra ao cliente) é sempre a mais vantajosa, pois, erros cometidos e detectados por inspeção, originam sempre modificações com custos mais ou menos elevados.

A qualidade do produto deve ser bem especificada, de forma a que as responsabilidades da boa execução (que incorporam essa qualidade) resultem bem claras; assim, as especificações devem ser acordadas com todas as partes e estar disponíveis perante todos os envolvidos no projeto. Segundo, Juran na liderança pela qualidade. Jj.(90) se no decurso dos trabalhos as especificações mudam, o gerente de projeto tem de assegurar-se que o cliente concorda com a sua revisão, e, com a conclusão do projeto o cliente deve proceder à aceitação final em termos da qualidade do produto. Assim, a qualidade no projeto implica atenção ao detalhe, especificações rigorosas e meios de verificar a qualidade (inspeções e testes).

O nível de qualidade deve ser definido pelo comprador que determina a natureza do controle do produto e do processo imposto no projeto. Embora a forma e intensidade deste controlo depende do tipo de produto em causa, ele existe sempre. Por outro lado, é o contrato ligando comprador e vendedor, que consigna logo à partida as especificações do produto. O reconhecimento de que essas especificações podem ser cumpridas é uma das mais importantes responsabilidades do gerente, devendo pois as especificações contratuais evitar: omissões, requisitos nebulosos, falta de rigor, inconsistência no detalhe, e requisitos de satisfação impossível.

Fluxo de Informação Comprador / Vendedor

À medida que a obra avança e o montante dos bens fornecidos sobem em valor, torna-se mais importante que a informação flua com maior rapidez, o que permite um planejamento mais eficaz, e ações corretivas (se necessárias) mais eficazes.

Em qualquer dos casos, as regras e as responsabilidades inerentes ao fluxo de informação devem ficar perfeitamente definidas de forma a evitar "mal entendidos", a atingir os objetivos do projeto, e permitindo que a superação de situações de desvio sejam mais eficazmente resolvidas.

4.1.1 Aspectos da Qualidade

O significado da Qualidade

Chegar a um acordo sobre que se entende por *qualidade* não é simples. O dicionário traz cerca de uma dúzia de definições. Para os gerentes nenhuma definição sucinta é realmente precisa, mas uma dessas definições obteve larga aceitação: qualidade é *adequação* ao uso.

Essa definição fornece um rótulo curto e compreensível, mas não fornece a profundidade necessária aos gerentes para escolherem os rumos da ação. Num exame mais detalhado descobre-se que a adequação ao uso divide-se em duas direções um tanto diferentes como ilustra na tabela abaixo:

CARACTERÍSTICAS DE PRODUTO QUE ATENDEM NECESSIDADES DE CLIENTES	AUSÊNCIA DE DEFICIÊNCIAS
Qualidade mais alta permite que a empresa: aumente a satisfação com o produto torne os produtos vendáveis atenda à competição aumente a participação no mercado forneça faturamento de vendas consiga preços vantajosos	Qualidade mais alta permite que a empresa: <ul style="list-style-type: none">• reduza freqüência de erros• reduza retrabalho, desperdício• reduza falhas de campo, despesas com garantia reduza insatisfação do cliente reduza inspeção, testes diminua o tempo necessário para colocar novos produtos no mercado aumente o rendimento, capacidade melhore o desempenho de entrega O principal efeito reside nos custos Geralmente, mais qualidade custa menos.

Tabela 1

Definições Subsidiárias

As definições da palavra *qualidade* incluem certas palavras-chaves que requerem, elas próprias, definições.

Produto

Produto é a saída de qualquer processo. Produto consiste principal-mente de bens, “software” e serviços.

Bens são coisas físicas, como lápis e aparelhos de televisão, e o software.

“*Software*” tem mais de um significado. Um dos principais significados é o de programa para computadores. Outro principal sentido é informação de um modo geral: relatórios, planos, instruções, conselhos e ordens.

Serviço é trabalho desempenhado para outra pessoa. Existem indústrias inteiras para prover serviços nas formas de, por exemplo, energia central, transporte, comunicação e entretenimento. Serviço também inclui trabalho desempenhado para outra pessoa dentro das

empresas, ou seja, preparação da folha de pagamento, recrutamento de novos empregados e manutenção da fábrica. Esses serviços são geralmente chamados serviços de apoio.

Características do Produto

Uma característica de produto é uma propriedade possuída por um produto com a intenção de atender a certas necessidades dos clientes e, dessa maneira, prover satisfação do cliente. As características do produto podem ser de natureza tecnológica – por exemplo, o consumo de combustível de um veículo, as dimensões de um componente mecânico, a viscosidade de um produto químico ou a uniformidade de voltagem de um fornecimento de energia elétrica. As características do produto também podem assumir outras formas – como pontualidade na entrega, facilidade de manutenção e cortesia no serviço.

Cliente

Um cliente é qualquer um que recebe ou é afetado pelo produto ou processo. Clientes podem ser internos ou externos.

Os *clientes externos* são afetados pelo produto mas não são membros da empresa que o produz. Os clientes externos incluem os clientes que compram o produto, órgãos governamentais de regulamentação e o público (que pode ser afetado por produtos inseguros ou por danos ao ambiente).

Os *clientes internos* são afetados pelo produto e também são membros da empresa que fabrica o produto. Eles são freqüentemente chamados clientes apesar do fato de não serem clientes no sentido encontrado nos dicionários; ou seja, eles não são fregueses.

Satisfação com o produto e satisfação do cliente

Atinge-se a satisfação com o produto quando as características do produto respondem às necessidades do cliente. Geralmente é um sinônimo de satisfação do cliente. A satisfação com o produto é um estímulo à eficácia de venda do produto. O principal impacto reside no faturamento das vendas.

Deficiências

Deficiência de um produto é a falta de um produto que resulta na *insatisfação com o produto*. As deficiências dos produtos ocorrem sob as formas de cortes de energia, falhas em cumprir os prazos de entrega, bens inoperáveis, aparência imperfeita e não conformidade à especificação. O principal impacto consiste nos custos implicados para refazer trabalho anterior e para responder as reclamações com o cliente.

4.1.2 Medidas da Qualidade

Cada uma das duas formas de qualidade apresentadas (na tabela acima) é medida na sua própria maneira. A medida usual não é em termos de ausência de deficiências, mas em termos de extensão de deficiências, tais como frequência de erros ou fração defeituosa. Essa medida pode ser generalizada pela expressão:

$$\text{Qualidade} = \frac{\text{Frequência de deficiências}}{\text{Oportunidades para deficiências}}$$

O numerador (frequência de deficiências) toma a forma de, por exemplo, número de defeitos, número de erros, número de falhas de campo, horas de retrabalho e custo da má qualidade.

O denominador (oportunidade para deficiências) toma a forma de, por exemplo, número de unidades produzidas, total de horas trabalhadas, número de unidades vendidas e faturamento das vendas.

Características do produto

Nessa forma de qualidade não existe uma unidade de medida universal. Em vez disso, o ponto de partida é o de descobrir dos clientes como eles avaliam a qualidade, ou seja, quais características de produtos eles consideram ser as mais importantes para atenderem suas necessidades.

Um fornecedor de serviços de transporte “expresso” mediu a qualidade baseado principalmente na porcentagem do espaço de transporte utilizado. Sua participação no mercado diminuiu porque os clientes mediam a qualidade baseados principalmente na pontualidade das entregas.

Um fabricante de aço inoxidável era um fornecedor para uma metalúrgica cujos clientes incluíam a National Aeronautics and Space Administration (NASA). O aço respeitava as especificações, mas mesmo assim o fornecedor perdeu parte da participação no mercado porque um concorrente embalava o aço com maior limpeza.

As características de produto são geralmente medidas em termos de *variáveis*; por exemplo, o tempo de entrega é medido em dias (ou horas, minutos e assim por diante); e a temperatura de recozimento é medida em graus. Por outro lado, as deficiências são geralmente medidas em termos de atributos; por exemplo, uma promessa é mantida ou não, ou um diâmetro está dentro das especificações ou não.

4.1.3 A trilogia da Qualidade de Juran

Aos preceitos de Juran em seu livro, *Juran na liderança pela qualidade* Jj. (90), o gerenciamento para a qualidade é feito pelo uso dos mesmos três processos gerenciais de planejamento, controle e melhoramento. Agora os nomes mudam para:

- Planejamento da qualidade
- Controle da qualidade
- Melhoramento da qualidade
- Vamos nos referir a esses processos como a Trilogia Juran.

As abordagens conceituais são idênticas àquelas usadas no gerenciamento das finanças. Entretanto, as etapas dos procedimentos são especiais, e as ferramentas usadas também são especiais.

Planejamento da Qualidade

É a atividade de desenvolver os produtos e processos necessários para atender às necessidades dos clientes. Envolve uma série de etapas universais, basicamente como se segue:

1. Determinar quem são os clientes.
2. Determinar quais são as necessidades dos clientes.
3. Desenvolver características de produtos que respondam às necessidades dos clientes.
4. Desenvolver processos que sejam capazes de produzir essas características de produto.
5. Transferir os planos resultantes às forças operacionais.

É importante frisar que estes fatores são constituintes de um Planejamento da Qualidade.

Controle da Qualidade

Segundo Jj. (90), esse processo consiste nos seguintes passos:

1. Avaliar o desempenho da qualidade real.
2. Comparar o desempenho real com as metas de qualidade.
3. Atuar nas diferenças.

Melhoramento da Qualidade

Esse processo é a maneira de elevar o desempenho da qualidade a níveis inéditos (inovação). A metodologia consiste de uma série de etapas universais:

1. Estabelecer a infra-estrutura necessária para assegurar um melhoramento da qualidade anual.

2. Identificar as necessidades específicas para melhoramento – os *projetos* de melhoramento.

1. A Trilogia Juran (Juran Trilogy) é uma marca registrada do Instituto Juran,

Como pensar sobre a Qualidade

3. Para cada projeto estabelecer uma equipe de projeto que tenha claramente a responsabilidade de fazer com que o projeto seja bem-sucedido.

4. Fornecer os recursos, motivação e treinamento necessário às equipes para Diagnosticar as causas Estimular o estabelecimento de uma solução Estabelecer controles para manter os ganhos.

Três seqüências universais

Note que cada um desses três processos foram generalizados na seqüência universal de etapas. Essas mesmas três seqüências universais foram descobertas e redescobertas várias vezes por gerentes.

O diagrama da trilogia Juran

Os três processos da trilogia da qualidade estão inter-relacionados. A figura pós tabela reflete o diagrama da trilogia Juran, onde se mostra essa inter-relação. O diagrama da trilogia Juran é um gráfico com tempo no eixo horizontal e custo da má qualidade (deficiências de qualidade) no eixo vertical. A atividade inicial é o planejamento da qualidade. Os planejadores determinam quem são os clientes e quais são as suas necessidades. Os planejadores em seguida desenvolvem projetos de produto e processo que são capazes de responder a essas necessidades. Finalmente, os planejadores passam os planos para as forças operacionais. A tarefa das forças operacionais é a de operar os processos e produzir os produtos. À medida que as operações prosseguem, logo, se percebe que o processo não é capaz de produzir 100 por cento de trabalho útil. A figura a seguir, (pós tabela) gerenciamento da qualidade sobre a trilogia de Juran, mostra que vinte por cento do trabalho deve ser refeito como resultado das deficiências de qualidade. Esse desperdício torna-se então crônico porque o “*processo operacional sai planejado dessa maneira*”. Sob os padrões de responsabilidade convencionais, as forças operacionais são incapazes de se livrar daquele desperdício crônico planejado.

GERENCIAMENTO PARA A QUALIDADE		
PLANEJAMENTO DA QUALIDADE	CONTROLE DE QUALIDADE	MELHORAMENTO DA QUALIDADE
<p>Determinar quem são os clientes</p> <p>Determinar as necessidades dos clientes</p> <p>Desenvolver características de produto que respondam às necessidades dos clientes</p> <p>Desenvolver processos capazes de produzir as características do produto</p> <p>Transferir os planos para as forças operacionais</p>	<p>Avaliar o desempenho real do produto</p> <p>Comparar o desempenho real às metas do produto</p> <p>Atuar sobre a diferença</p>	<p>Estabelecer a infra-estrutura</p> <p>Identificar os projetos de melhoramento</p> <p>Estabelecer equipes de projeto</p> <p>Fornecer às equipes recursos, treinamento e motivação para:</p> <ul style="list-style-type: none"> • diagnosticar as causas • estimular os remédios • estabelecer controles para • manter os ganhos

Tabela 2

O que elas fazem, em vez disso, é executar *controle de qualidade* – para prevenir que as coisas piorem. O controle inclui a extinção de incêndios, como essa ponta esporádica.

O quadro também mostra que com o tempo o desperdício crônico é reduzido a um nível muito abaixo do nível originalmente planejado. Esse ganho é conseguido pelo terceiro processo na trilogia: melhoramento da qualidade. Com efeito, percebeu-se que o desperdício crônico é também uma oportunidade para melhora-mento, de modo que passos foram dados para aproveitar essa oportunidade.

O diagrama da trilogia Juran e deficiências de produto

O diagrama da trilogia, figura 11 demonstrada a seguir, relaciona-se a *deficiências de produto*. A escala vertical pode apresentar, portanto, unidades de medida como custo da má qualidade, frequência de erros, porcentagem de defeitos e frequência de chamadas da assistência técnica. Nessa mesma escala, a perfeição seria o zero. Quando os números sobem, significa que a situação piora.

O resultado de reduzir as deficiências dos produtos é a redução do custo da má qualidade, cumprir mais promessas de entregas e reduzir a insatisfação do cliente.

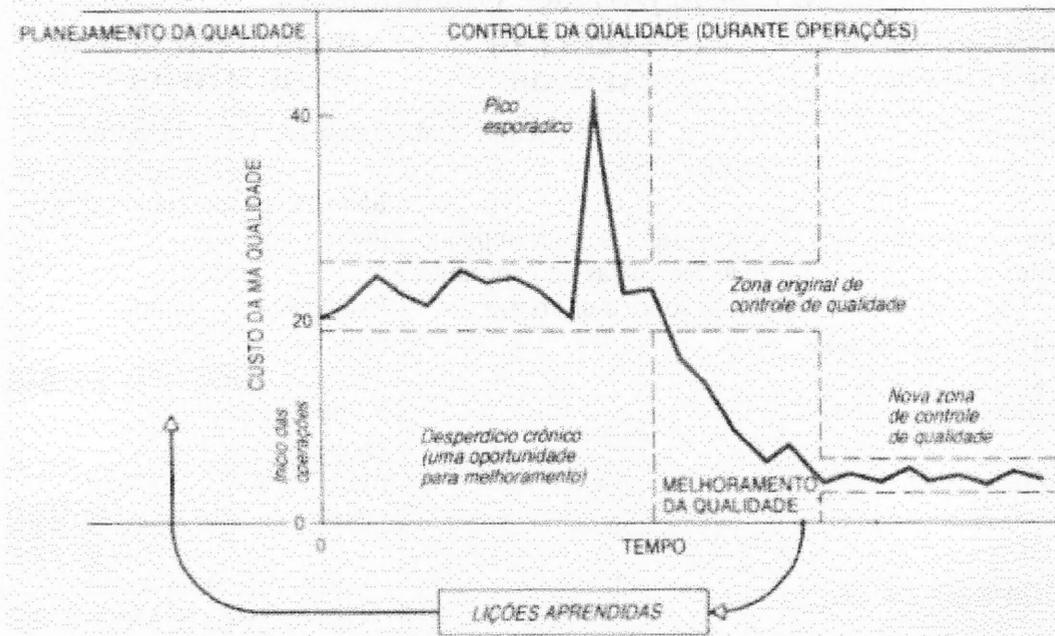


Figura 11

O diagrama da trilogia e as características de produto

Quando o diagrama da trilogia é aplicado a características de produto, a escala vertical muda. Agora a escala apresenta unidades de medida para características como horas de vida, milhões de instruções por segundo, capacidade de transporte de carga e tempo médio entre as falhas. Quando os números sobem, significa que a situação melhora.

Para algumas outras características de produto, o que sobe é ruim, por exemplo, tempo de espera nas filas, consumo de energia, horas de manutenção por 1.000 horas de operação e tempo necessário para restabelecer o serviço.

Para facilidade na interpretação dos diagramas, deveremos rotular as escalas verticais da seguinte maneira:

1. Para *características de produto* deveremos chamar a escala vertical de “Eficácia de venda do Produto”. Nesses diagramas, o que sobe é bom.
2. Para *deficiências de qualidade* deveremos chamar a escala vertical de “Custo da Má Qualidade”. Nesses diagramas, o que sobe é ruim.

4.1.4 Qualidade Total – Prevenção e Inspeção

Toda a lógica da Qualidade Total assenta em que cada pessoa ligada ao projeto, compare o seu trabalho com as especificações garantindo que todos os produtos que produz estão de acordo com as especificações e portanto que a sua produção tem zero defeitos. Cada pessoa envolvida deve saber quais as anomalias que podem surgir no seu trabalho, como as detectar e como as evitar.

Assim, o autocontrolo é realizado à priori e com confiança, partindo da definição de qualidade para uma medida de conformidade, o que permite que haja prevenção a caminho do "Zero Defeito".

Os dois vetores de qualidade são portanto, a prevenção e a inspeção, sendo a prevenção da responsabilidade de todos e a inspeção da responsabilidade dos especialistas de qualidade.

O responsável da qualidade:

- Fixa os requisitos da qualidade do projeto tendo presente as necessidades do utilizador.
- Supervisiona os testes de forma a assegurar que o produto cumpre a sua tarefa. Qualquer desvio deve ser rapidamente detectado através de um adequado sistema de informação determinando a sua extensão e corrigindo o erro.

4.1.5 O Papel do Gerente de Projetos na Qualidade

No que concerne à qualidade, segundo Vr (04), cabe ao gerente de projeto traçar estratégias alternativas que passam por definições do seguinte tipo:

- mais qualidade com mais custo dentro do orçamento ou
- menos custo com mais qualidade dentro da especificação ou
- mais qualidade com mais prevenção tentando minimizar custos

Os aspectos de gestão da qualidade que interessam especialmente ao gerente de projeto são:

- a) Assegurar-se que os meios físicos da produção garantem a qualidade do projeto.
- b) Prover o que a engenharia, desenhos e instruções fornecem e informação necessária e correta.
- c) Estabelecer um adequado plano de controlo assegurado a sua manutenção - no relatórios e na produção.
- d) Manter a eficácia da atuação da gestão da qualidade.
- e) Garantir que ao nível de aceitação, tudo é documentado.

4.1.6 CMM – Capability Maturity Model

Desenvolver software de qualidade assegurada, com elevada produtividade, dentro do prazo estabelecido e sem necessitar de mais recursos do que os alocados, tem sido o grande desafio da Engenharia de Software. Frequentemente surgem defensores de soluções miraculosas apresentadas sob a forma de ferramentas, métodos ou outra tecnologia qualquer. Sabe-se que, infelizmente, tais soluções não existem. Segundo os idealizadores do CMM, a principal causa dos problemas é a falta de um processo de desenvolvimento de software claramente definido e efetivo.

Conhecer os processos significa conhecer como os produtos e serviços são planejados, produzidos e entregues. Um processo, em um nível mais macro, cruza horizontalmente várias funções da organização.

Segundo Soeli Fiorini Sf.(98) em seu livro Engenharia de software com CMM, no gerenciamento com uma visão funcional (vertical) Fiorini afirma que cada pessoa vê apenas a sua parte no processo. Há pouca comunicação entre os envolvidos no processo fazendo com que as ligações (links), entre uma parte e outra do processo sejam ignoradas, limitando-se a apenas uma troca de produtos. Entretanto, é geralmente nessas ligações onde se concentra a maior parte dos problemas relacionados à otimização do processo.

Outra decorrência desta visão são os conflitos comuns com relação à priorização de funções ou áreas. Na visão por processos, horizontal, estes são definidos, ou redefinidos, e discutidos por todos os envolvidos, por exemplo, clientes e fornecedores internos, numa tentativa de melhorar as ligações, retirando-se o que não agrega valor ao negócio. Dessa forma, todos os envolvidos passam a ter uma visão única de como o processo “flui” na organização, entendem porque o trabalho de cada um é importante no que se refere aos objetivos do processo, e melhoram a comunicação. Ou seja, todos os participantes passam a falar a linguagem do processo estabelecido, melhorando o relacionamento cliente / fornecedor interno.

Cabe ressaltar que, a partir da definição do processo, é possível definir-se medições e coletar dados de execução. Isto dá visibilidade aos gerentes e técnicos sobre o andamento dos projetos, possibilitando ações para controlar as variações do projeto e dos processos por ele utilizados.

Uma vez definidos os processos e estabelecido o treinamento, o processo fluirá por si só, sem depender de pessoas chave. Ou seja, quando uma pessoa sai do projeto, outra poderá assumir o seu lugar, guiando-se por este processo.

O CMM enfatiza a documentação dos processos, seguindo a premissa de que, para realizar alguma melhoria no processo, é preciso primeiro conhecê-lo e entendê-lo, e que a qualidade de um produto é reflexo da qualidade e gerenciamento do processo utilizado em seu desenvolvimento.

Muitos gerentes e profissionais de software já observaram que uma das principais causas dos problemas com software é a desorganização do processo e a inexistência de padrões documentados visando o desenvolvimento e a manutenção de software. O cenário mais encontrado ainda é o do desenvolvedor heróico que utiliza as ferramentas que mais lhe agradam (frequentemente diferentes das utilizadas pelos seus colegas de trabalho), trabalha com um método que, muitas vezes, somente ele entende, e que promete o que não pode cumprir.

Este cenário tende a ser complementado com uma gerência passiva, senão inoperante, e que raras vezes tem noção do que está ocorrendo (a assim chamada síndrome dos 90% pronto). Na esperança de diminuir estes problemas dá-se aos líderes de projeto também o papel de gerentes, sobrecarregando-os com tarefas e obrigações que pouco contribuem para melhorar o desenvolvimento. O resultado é usualmente o caos. Programas que não funcionam a contento, custos excessivos, desentendimentos entre pessoas, culpando-se mutuamente pelos problemas coletivamente enfrentados. Pior, ainda, é o risco de perda de investimento que ocorre sempre que alguém deixa o projeto.

E qual a causa? Como reduzir o efeito negativo destes problemas? Voltam à cena os já citados vendedores de curas milagrosas, propondo métodos de gerência do tipo “tudo ou nada” e que, segundo eles obviamente, farão a felicidade de todos que utilizam tais métodos. Infelizmente, isto não tem funcionado. As organizações de desenvolvimento de software estão repletas de histórias de insucesso da adoção de assim chamadas “boas práticas”. Não existem soluções gerais, simplesmente porque cada projeto é um caso, cada empresa é um caso. Qualquer que seja o processo de desenvolvimento utilizado, ele precisa ser adaptado à empresa e aos projetos por ela desenvolvidos. Ou seja, a área de desenvolvimento e manutenção de software precisa aprender a se organizar, ao invés de esperar que poderá comprar soluções prontas.

Segundo Sf.(98), esta organização deve contribuir para melhorar o estado dos processos e não somente para burocratizar. E claro, deve ser capaz de mostrar através de resultados cuidadosamente medidos que a melhoria foi efetivamente alcançada.

O CMM – Capability Maturity Model for Software: ou em português Modelo de Maturidade da Capacitação para software, propõe um caminho gradual que leva as organizações e fazem com que desenvolvam software de forma mais organizada, cada vez com menos problemas de qualidade e menos erros de estimativa de prazo e de necessidade de recursos. A adoção do CMM traz consigo um contínuo aprimoramento da qualidade produzida e uma melhoria de produtividade.

Como será descrito mais adiante, o CMM reflete a experiência profissional adquirida e praticada ao longo de vários anos por um número significativo de pessoas. Ou seja, teve significativo tempo para ser experimentado e avaliado na prática. O CMM foi originalmente desenvolvido para grandes corporações, em particular visando empresas desenvolvendo software para o Departamento de Defesa dos Estados Unidos. Não impede, no entanto, a adoção de seus princípios também por pequenas empresas, desde que, é claro, façam as devidas adaptações. Como foi salientado, o CMM não só permite, como até sugere, que se façam adaptações, desde que as metas das áreas-chave do processo sejam atingidas. Segundo Sf (98), podemos observar diferentes pontos e situações que qualificam organizações maduras de organizações imaturas como demonstrado na tabela a seguir:

Nível de Maturidade das Organizações:

Organizações maduras	Organizações imaturas
Papéis e responsabilidades bem definidos	Processo improvisado
Existe base histórica	Não existe base histórica
É possível julgar a qualidade do produto	Não é viável julgar a qualidade do produto
A qualidade dos processos é monitorada	Qualidade e funcionalidade do produto sacrificada
O processo pode ser atualizado	Não há rigor no processo a ser seguido
Existe comunicação entre o gerente e grupo	Resolução de crises imediatas

Tabela 3

Modelo de Maturidade da Capacitação para Software

O CMM é uma estrutura (framework) que descreve os principais elementos de um processo de software efetivo. O CMM descreve os estágios através dos quais organizações de software evoluem quando elas definem, implementam, medem, controlam, e melhoram seus processo de software.

O CMM fornece uma diretriz para a seleção de estratégias de melhoria de processos permitindo a determinação da capacitação dos processos correntes e a conseqüente identificação das questões mais críticas para a melhoria do processo e da qualidade de software.

Desta forma prove e descreve um caminho de melhoria evolutiva a partir de um processo “ad hoc” para um processo maduro e altamente disciplinado. Este caminho de disciplina é definido por cinco níveis de maturidade: Inicial, Repetitiva, Definido, Gerenciado em otimização.

Cada nível de maturidade agrega áreas-chave’ de um processo de software (por exemplo, gerência de requisitos e definição do processo da organização) que formam um patamar coerente de competência a ser atingido, rotineiramente praticado e melhorado.

Cada área-chave possui um conjunto de metas e quando estas metas são satisfeitas rotineiramente, é aumentada a capacitação do processo em produzir resultados previsíveis e de qualidade assegurada.

O CMM focaliza na capacitação das organizações de software produzirem consistente e previsivelmente produtos de qualidade assegurada. Também focaliza na premissa básica ma gerência de processo na qual a qualidade de um produto. No caso software, é determinada principalmente na qualidade de seu processo de desenvolvimento.

O modelo "da maturidade da capacitação" do CMM revela esta premissa melhorando o processo – fortalecendo as propriedades de ser definido, praticado, gerenciado, medido e, controlado (maturidade do processo). É possível melhorar os produtos reduzindo a taxa de incerteza quanto a resultados esperados (capacitação do processo) que podem ser alcançados

seguinto este processo de software. Assim, conhecendo a capacitação do processo pode-se determinar seu nível de maturidade.

Logo, entende-se por maturidade da capacitação: o quanto capaz é o processo de assegurar a gerenciabilidade do projeto, a qualidade dos produtos gerados, a adaptação do processo as características específicas das empresas e dos projetos, e, também o quão os projetos são capazes de ser aprimorado.

O Desenvolvimento de software. É um modelo que auxilia na comunicação entre envolvidos com atividades de engenharia de software na medida em que estabelece linguagem comum de conceitos para falar do processo de software.

Segundo Sf.(98), o CMM difere de outros modelos pela sua ênfase na melhoria contínua de processos. O SEI estabeleceu uma proposta que descreve fases, atividades e recursos necessários para tomar iniciativas de melhoria de processo com esta medida.

Esta proposta é similar ao conhecido ciclo de melhoria PDCA - Plan, Do, Check e Act – Planejar, Fazer, Verificar e Agir) e é denominada EAL Segundo Peterson 95, e Radice 94. As fases do IDEAL, que formam, são descritas sucintamente a seguir:

Iniciar – através de estímulos visando a melhoria é estabelecida uma infra-estrutura para a melhoria, com o patrocínio da gerência sênior.

Diagnosticar - é gerado um diagnóstico documentado das práticas correntes (por exemplo, através de avaliações de processos) e são geradas recomendações para melhorias.

Estabelecer - são estabelecidas e documentadas as estratégias e prioridades, são identificados os processos e recursos necessários. É elaborado um plano de ação.

Gerenciar - são coletadas medidas detalhadas da qualidade do produto e processo de desenvolvimento de software. Tanto o produto quanto o processo de desenvolvimento de software são entendidos e controlados quantitativamente

Otimizar - o melhoramento contínuo do processo é conseguido através de um “feedback” quantitativo dos processos e pelo uso pioneiro de idéias e tecnologias inovadoras.

Cada nível de maturidade compreende um conjunto de metas de processo que, quando satisfeitas, estabilizam um importante componente do processo de software. Cada nível da estrutura de maturidade estabelece um componente diferente no processo de software, resultando em um aumento na capacitação do processo da organização segundo Sf.(98), como podemos observar na figura 12 a seguir:

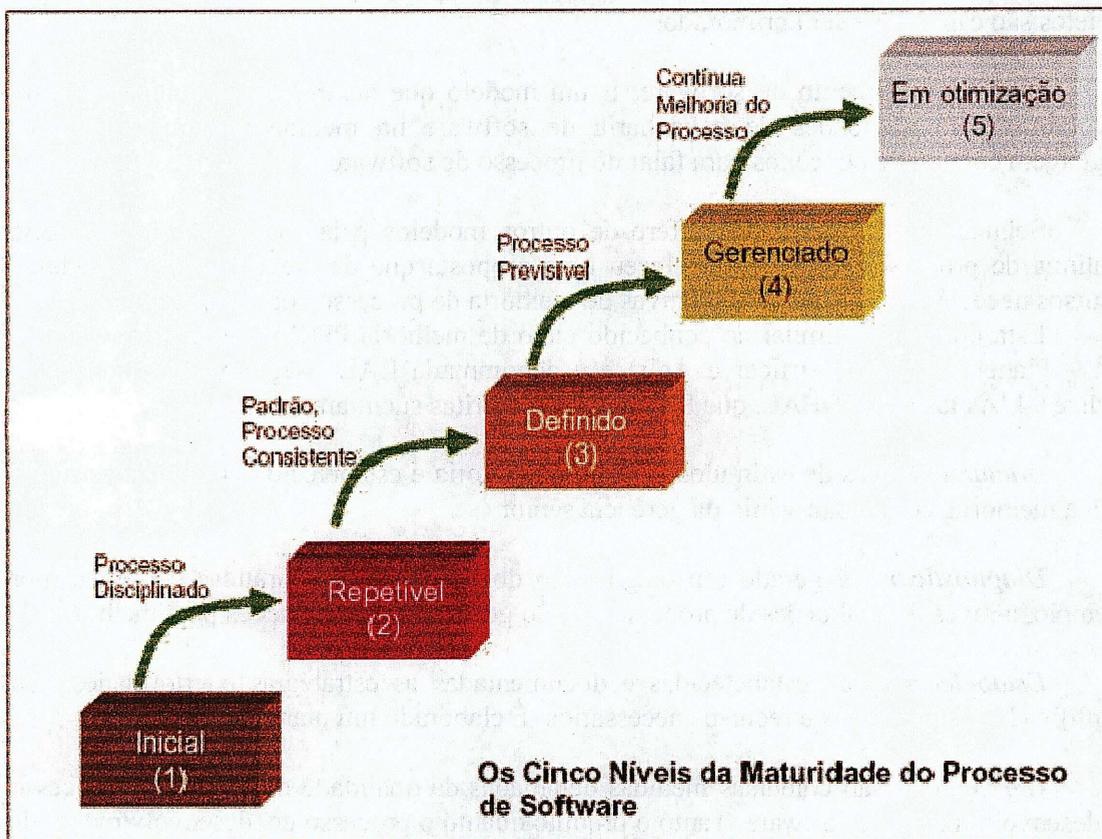


Figura 12

Existem cinco níveis de maturidade que caracterizam o nível de capacitação do processo da organização: Inicial, Repetitivo, Definido, Gerenciado e em Otimização.

Para se estar em um determinado nível, o processo deverá satisfazer a todas as ACPs deste nível e de seus níveis inferiores. Assim, mesmo que uma empresa satisfaça a todas, exceto uma das ACPs e produza software de excelente qualidade, estará no nível 1 caso a ACP não satisfeita seja do nível 2. Estará no nível 2 se ACPs não satisfeita for do nível 3, e assim por diante. Conseqüentemente, é possível que empresas do nível 1 satisfaçam diversas, mas não todas as ACPs dos níveis 2 a 5. Segundo Pfleeger 97, é recomendado que se adote ACPs mais adiantadas, para que estas possam contribuir para resolver problemas específicos de determinado projeto. Ou seja, o processo de adoção do CMM não precisa seguir uma forma seqüencial em que cada nível é resolvido para, só então, se prosseguir para o próximo nível.

CMM Nível 1:

No nível 1 o processo de desenvolvimento de software é uma caixa preta, de forma que as entradas e os produtos finais podem ser vistos com clareza. Os pretendem a ser diretamente dependentes de posturas individuais e a visibilidade limitada. Não há controle de requisitos e o cliente só os avalia na entrega do projeto. Como a seqüência das atividades é fracamente definida, gerentes têm dificuldade de estabelecer o estado das atividades e o progresso do projeto.

De maneira geral, a organização não dispõe, neste nível, de um ambiente organizável para o desenvolvimento e a manutenção de software. Usualmente as gerenciais mínimas estão ausentes. Isto provoca um planejamento e acomodo do desenvolvimento quando realizado pouco efetivo, compromete qualquer benefício oriundo da adoção de boas práticas de engenharia.

Ocorrendo uma crise, os projetos normalmente abandonam procedimentos planejados caso existam, e passam a utilizar o reconhecidamente ineficaz processo de codificação e testes. O sucesso depende da disponibilidade de um excepcional gerente de desenvolvimento, bem como de uma equipe de desenvolvimento altamente eficiente. Ou seja, as deficiências do processo são superadas com esforços heróicos. Alguns gerentes de desenvolvimento mais capazes resistem às pressões em manter em uso um processo, por mais rudimentar que seja, evitando seguir atalhos que somente aumentam custos e deterioram qualidade. No entanto, esta resistência e sua influência estabilizadora lhes acompanha quando estes gerentes deixam o projeto. Mesmo um forte processo de engenharia não consegue separar a instabilidade criada pela ausência de sólidas práticas gerenciais.

A capacitação do processo de software do nível 1 é imprevisível, pois o processo explicitamente definido ou é constantemente alterado com o transcorrer do trabalho. Os cronogramas, orçamentos, funcionalidades e qualidade de projeto são geralmente imprevisíveis.

Segundo dados do SEI 97, referentes a 533 organizações que reportaram seus resultados em ações realizadas desde 1992 e até o final de 1996 revelam que 61,5% das organizações são nível 1.

Entretanto, estar no nível I não significa que uma organização não possa produzir um bom software. Significa que a organização desenvolve produtos muitas vezes satisfatórios, entretanto orçamento e cronograma tendem a ser excedidos significativamente, e o sucesso só pode ser repetido se as mesmas pessoas forem alocadas em outro projeto. Ou seja, a capacitação não é da organização e sim das pessoas.

CMM Nível 2:

Desenvolvimentos bem sucedidos podem ser repetidos.

No nível 2, o processo passa a ter um nível básico de controle. Ao invés do processos de desenvolvimento ser uma única caixa preta ele passa a ser uma seqüência de caixas pretas (tarefas) que asseguram a visibilidade em determinados pontos, tipicamente nos chamados marcos de acompanhamento de progresso do projeto (mile-stones). São controlados também a evolução dos requisitos, de tal forma que o cliente possa avaliá-los ao final de cada marco do projeto, e são controlados, ainda, a evolução das configurações. Embora a gerência não saiba como cada uma das tarefas é realizada, os produtos do processo e os marcos são identificados e controlados, permitindo verificar se o processo está sendo executado de acordo com os planos estabelecidos. Neste nível a gerência ainda não é pró-ativa, mas sim reativa. Ou seja, ao invés de conduzir o desenvolvimento de modo que não ocorram crises, toma iniciativas somente quando observa que está defronte a uma delas.

No nível repetitivo são estabelecidas políticas para gerenciar projetos de desenvolvimento de software bem como procedimentos para implementá-las. O planejamento e a gerência de novos projetos é baseada em experiências adquiridas com projetos similares já realizados. A capacitação do processo é melhorada, projeto a projeto, através do estabelecimento de disciplinas básicas de gerência de processo. Para cada projeto são estabelecidos processos que são definidos, documentados, praticados, executados, treinados, medidos, obedecidos e passíveis de melhoria.

Os projetos de organizações de nível 2 possuem controle básico de gerência do desenvolvimento de software. Os compromissos de um projeto são baseados tanto

CMM Nível 3:

No nível 3 a organização interna das tarefas estaria definida e visível (Processo de Software Definido do Projeto. Ou seja, a caixa preta passa a ser caixa aberta. A estrutura interna representa as tarefas do processo este software definido do projeto e a maneira com que o processo de software padrão da organização tem sido aplicado a projetos específico. Os envolvidos (gerentes e técnicos conhecem seus papéis, responsabilidades e papéis a forma com que suas atividades interagem entre si. Há preparo gerencial para prevenção e observação de riscos. Dados relativos às alterações de projeto podem ser obtidos de forma rápida e segura.

No nível definido, o processo padrão e os processos utilizados para desenvolver e manter software estão documentados e permeiam toda a organização. O processo de software padrão da organização inclui processos de gerência e de engenharia de software. Esses processos são integrados, formando um todo coerente e consistente. Os processos estabelecidos no nível 3 são usados e alterados, quando apropriado, para ajudar os gerentes de software e a equipe técnica a desempenhar mais efetivamente suas atividades. Ao padronizar seus processos de software, a organização explora práticas eficazes de engenharia de software.

Na organização é destinado um grupo responsável pelas atividades de processo de software. É implementado um programa de treinamento que abrange toda a organização, visando assegurar que a equipe técnica e os gerentes tenham os conhecimentos e as

habilidades requeridos para desempenhar plenamente seus papéis. Dessa forma o conhecimento adquirido passa a ser da organização e não das pessoas com que nela atuam.

Segundo Sf.(98), o processo de software padrão da organização é adaptado às características; individuais de cada projeto, estabelecendo o seu principio Processo de Software Definido. Um processo de software definido contém um conjunto integrado, coerente e bem definido de processos de gerência e de engenharia de software. Um processo de software bem definido pode ser caracterizado como contendo critérios de prontidão (todos os requisitos necessários para iniciar uma atividade estão disponíveis). Entradas, padrões e procedimentos para realizar o trabalho, mecanismos de verificação, tais como revisão por parceiros, saídas e critérios de aceitação. Devido ao processo de software ser bem definido, a gerência tem uma boa percepção do progresso técnico dos projetos.

A capacitação do processo de software de organizações nível 3 pode ser resumida como padronizada e consistente porque ambas as atividades de gerência e engenharia de software são estáveis e repetíveis. Com relação às linhas de produtos estabelecidas, custos, cronograma e funcionalidades estes estão sob controle e a qualidade de software é acompanhada. Esta capacitação de processo é baseada num entendimento comum das atividades, papéis e responsabilidades num processo de software definido ao longo de toda a organização. A capacitação do processo deixa de ser uma habilidade das pessoas passando a ser uma habilitação da organização.

CMM Nível 4:

Processo medido e controlado. No nível 4 os processos de software definidos são instrumentados e controlados linhas de produtos estabelecidas. Os riscos envolvidos ao ascender à curva de aprendizado no domínio de uma nova aplicação são conhecidos e cuidadosamente gerenciados.

A capacitação do processo de software para organizações de nível 4 é quantificável e previsível, uma vez que o processo é medido e opera dentro de limites aceitáveis. O nível da capacitação do processo, neste nível, permite a uma organização prever tendências em processos e em qualidade de produtos dentro de limites quantitativos. Visto que o processo é estável e medido quando ocorrer algum evento inesperado as conseqüências poderão ser identificadas e abordadas minimizando seus impactos.

Quando os limites definidos forem excedidos poderão ser tomadas as ações para entender as causa e corrigir a situação de modo que problemas sistemáticos deixem de ocorrer. Finalmente os produtos são de alta qualidade.

CMM Nível 5:

No nível 5, tenta-se, de maneira continua e controlada, identificar, avaliar ao desenvolver novas e melhores maneiras de construir a software, de forma a melhorar a qualidade e a produtividade. As alterações são disciplinadas e as atividades com tendência a problemas são identificadas e revisadas ao substituí-las. O entendimento do processo ultrapassa os processos praticados, possibilitando compreender as efeitos de alterações potenciais no processo. Os gerentes são capazes de estimar e acompanhar quantitativamente

a impacto e a efetividade de alterações nos processos. Os clientes e a organização trabalham juntos para fortalecer ainda mais o seu relacionamento. Através dos cinco níveis da capacitação do processo interage com pessoas, tecnologia e medições, como uma organização madura.

No nível em otimização, a organização inteira esta voltada para a processo de melhoria contínua. A organização tem meios para identificar fraquezas e fortalecer o processo a de forma pró-ativa, prevenindo a ocorrência de defeitos. Os dados de efetividade do processo de software são usados para realizar análises de custo/benefício em novas tecnologias e alterações propostas para o processo de software da organização. As inovações que exploram as melhores práticas de engenharia de software são identificadas e disseminadas por toda a organização.

As equipes de software em organizações nível 5 analisam os defeitos para determinar suas causas. Os processos de software são avaliados, visando prevenir defeitos recorrentes, e as lições aprendidas são disseminadas por toda a organização.

Desperdício crônico, na forma de retrabalho, pode ser encontrado em qualquer sistema como consequência de causas aleatórias. Esforços organizados visando remover desperdícios levam a melhorias do processo, eliminando as causas comuns de ineficiência através de ações e atitudes preventivas. Embora isto seja verdade para todos os níveis de maturidade, é no nível 3 que se dá com maior ênfase.

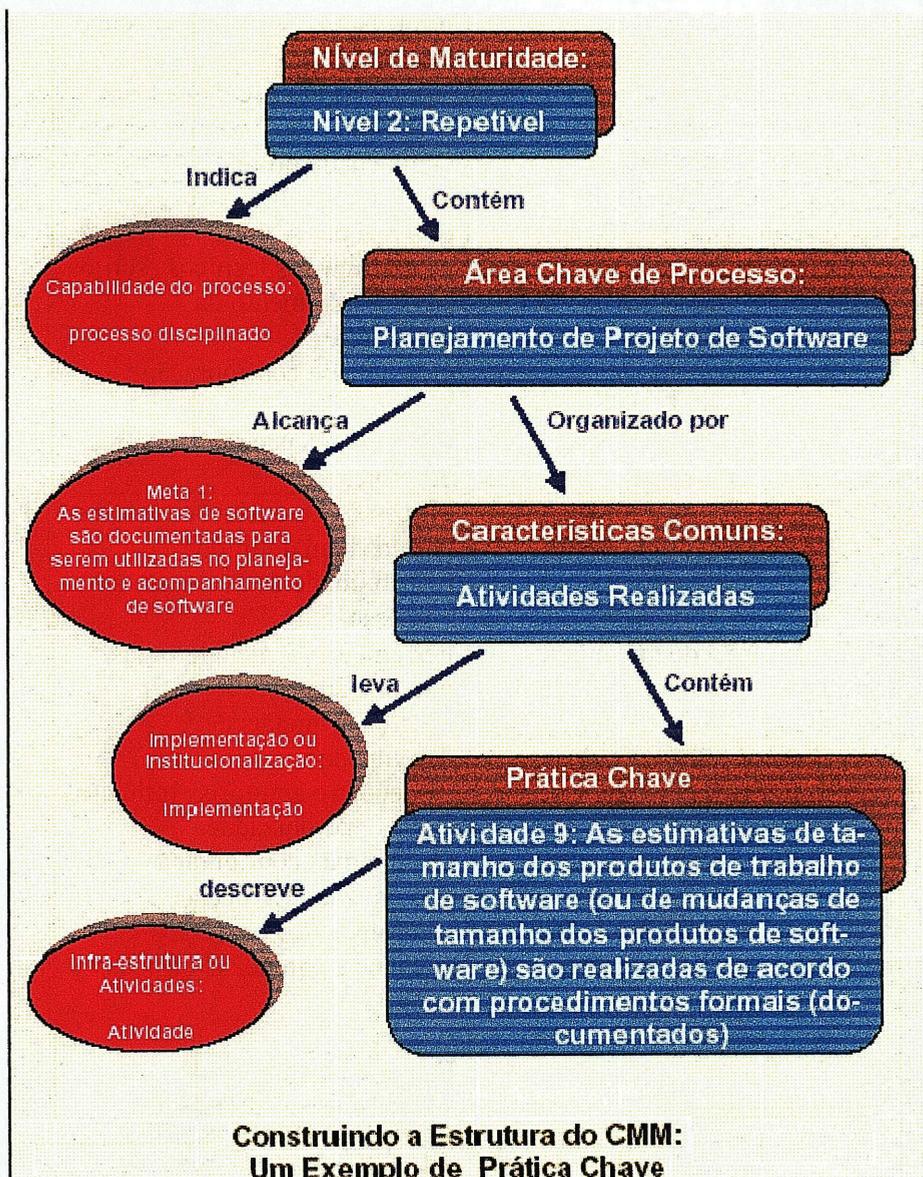
A capacitação do processo de software neste nível pode ser caracterizada como sendo de melhoria contínua, pois organizações nível 5 estão continuamente se empenhando para melhorar a capacitação do processo. As melhorias ocorrem tanto por avanços incrementais no processo existente quanto por inovações fazendo uso de novas tecnologias e métodos. Melhorias de tecnologias e processos são planejadas e gerenciadas como atividades de negócio rotineiras.

Estrutura

Segundo Sf.(98), em resumo o CMM é definido em função de um conjunto de:

- Níveis de maturidade;
- Áreas-chave de processo;
- Características comuns;
- Práticas-base.

Veja na figura 13 abaixo como estes elementos se interligam na estrutura do CMM usando como exemplo Planejamento de Projeto de Software do Nível 2.



Resumo dos níveis de maturidade

Nível	Resumo	Caracterização	Tipo da Capacitação
Inicial	O processo de software não está documentado e, usualmente, sequer existe. Poucos processos estáveis existem ou são usados, e o sucesso depende de esforços individuais.	Programadores consideram-se artistas. Padrões não existem, ou tendem a ser ignorados. Ferramentas são usadas ao acaso, muitas vezes por iniciativa pessoal. Metodologias são praticadas informalmente. Resultados não são previsíveis A coleta e análise de dados é "Ad Hoc"	Processo "Ad Hoc"
Repetitivo	Estão estabelecidos processos básicos de gerência de projeto para planejar e acompanhar custos, prazos e funcionalidades. Compromissos são firmados e gerenciados e sucessos podem ser repetidos.	Gerência de projetos estabelecida. Processo organizado. Alguns procedimentos técnicos escritos Acompanhamento da qualidade. Gerência de configuração inicial. O sucesso depende do gerente do projeto.	Processo Disciplinado
Definido	Tanto para as atividades de gerência básica como para as de engenharia de software, o processo de software é documentado, padronizado e integrado num processo único, chamado Processo de Software Padrão da Organização. Todos os projetos usam uma versão deste processo, adaptada às características específicas do projeto, contemplando o desenvolvimento e a manutenção de software.	Processos gerenciais e técnicos básicos bem definidos. Possibilidade de avaliação do processo. Ferramentas e metodologias padronizadas. Medições iniciais de desempenho. Inspeções e auditorias rotineiras Testes padronizados. Gerência de configuração generalizada. Evolução controlada dos processos técnicos e gerenciais básicos.	Processo Padronizado e Consistente
Gerenciado	São coletadas medições detalhadas do processo de software e da qualidade dos produtos. São gerados relatórios estatísticos. O	Está estabelecido e em uso rotineiro um programa de medições. Está estabelecido um grupo de garantia da qualidade.	Processo Previsível

	processo e os artefatos de software são avaliados quantitativamente e são, também, controlados.	A qualidade é planejada. A qualidade é rotineiramente avaliada e aprimorada	
Otimizado	É realizada rotineiramente a melhoria do processo como um todo. São realizados projetos piloto para a absorção e internalização de novas tecnologias.	Alto nível de qualidade é alcançado rotineiramente. Melhoria contínua. Alto nível de satisfação dos clientes	Processo em Melhoria Contínua

Tabela 4

4.1.7 ISO 9000

As Normas ISO série 9000 (ISO-9000) para a gestão e a garantia da qualidade representam o consenso de vários países para a normalização da qualidade. Elas orientam na busca da melhoria dos níveis da qualidade para produtos, serviços e relacionamento cliente / fornecedor, através de diretrizes para a implantação de sistemas da qualidade nas empresas.

A série 9000 especifica requisitos mínimos para que, através de ações gerenciais centradas na prevenção de problemas, as empresas possam assegurar a qualidade de seus produtos e serviços. Trata-se, em última instância, do emprego de normas básicas na busca de níveis de qualidade aceitáveis, tanto nos negócios locais como nos negócios internacionais.

A sigla ISO origina-se de *International Organization for Standardization*. Esse é o nome de um grupo internacional de normalização localizado em Genebra, Suíça. Essa organização foi fundada em 23 de fevereiro de 1947 e não possui ligações com órgãos governamentais.

Entidades de normalização de mais de noventa países fazem parte desse grupo, entre os quais o Brasil, através da ABNT – Associação Brasileira de Normas Técnicas.

A série 9000 foi elaborada em 1979 pelo ISO Technical Committee 176 (ISO-TC-176), o primeiro dos comitês do grupo ISO a tratar dos assuntos da gestão e garantia da qualidade. A primeira versão, porém, somente foi publicada oficialmente em 1987.

Empresas de todo o mundo estão se certificando ou simplesmente se adaptando a essas normas que já são condição obrigatória para negócios com a Comunidade Européia. Em princípio, estão previstas revisões para a série em intervalos de cinco anos, quando, então, novas versões serão homologadas.

A doutrina das Normas ISO-9000 não contempla a definição de regras e modelos rígidos ou a imposição de sistemas da qualidade preestabelecidos para serem implementados pelas organizações. Ao contrário, ela oferece um conjunto de diretrizes e requisitos mínimos a partir dos quais as empresas estabelecem modelos próprios de gestão da qualidade, condizentes com seu tipo de negócio e características. Ou seja, cada empresa tem seu modelo e suas regras.

Assim, um trabalho de Adequação ou Certificação às Normas ISO-9000 verifica primeiro se existe uma Política da Qualidade (estabelecida pela Alta Administração) e um Sistema da Qualidade documentado pelo qual essa Política é implementada. A seguir, é verificado se tanto a Política como o Sistema estão entendidos e implementados em todos os níveis da empresa.

A fase seguinte consiste em verificar se existem regras e procedimentos formalizados para todos os processos da empresa. Se não existem, a empresa tem que defini-los. Se existem, o próximo passo é saber se essas regras e esses procedimentos (que a própria empresa definiu) estão acessíveis, atualizados e sendo devidamente cumpridos (auditoria interna).

Outro ponto importante é que a auditoria prevista nas Normas tem por objetivo atuar sobre o Sistema de Qualidade da empresa, e não sobre a Qualidade do Software em si, que deve ser decorrência desse sistema. A Norma parte do princípio que a Qualidade do Software não pode ser superior à do processo que o produziu.

Adequação Versus Certificação

Entende-se por Adequação à ISO-9000 a implementação de um programa de qualidade segundo aquelas Normas (total ou parcialmente). A Certificação implica, necessariamente, a existência de uma empresa externa devidamente credenciada que realiza auditoria na empresa candidata e concede ou não o Certificado de Conformidade às Normas ISO.

A Certificação é concedida para cada base da empresa individualmente (*site*). Ou seja, a Certificação de uma divisão ou filial não se estende a todas as bases da empresa em questão. Também a Certificação obtida por meio de uma entidade de um determinado país não é, obrigatoriamente, reconhecida por outro. Isto vale também para clientes que exigem Certificação de seus fornecedores. O cliente pode reconhecer ou não a Certificação concedida por determinada entidade a seu fornecedor.

Os clientes que exigem Certificação devem informar a seus fornecedores os critérios ou entidades de Certificação que reconhecem.

Documentos Básicos da Série 9000

A série é composta, basicamente, de cinco documentos numerados seqüencialmente: ISO-9000, ISO-9001, ISO-9002, ISO-9003 e ISO-9004. Segue a titula desses documentos e o correspondente documento oficializado pela ABNT (NBR-190XX):

<i>Referência ISO/(ABNT)</i>	<i>Título do Documento</i>
ISO-9000 (NBR-19000)	Normas de Gestão da Qualidade e Garantia da Qualidade. Diretrizes para Seleção (da Norma mais adequada ao caso da empresa) e Uso.
ISO-9001 (NBR-19001)	Sistemas da Qualidade – Modelo para Garantia da Qualidade em Projeto Desenvolvimento, Produção, Instalação e

	Assistência Técnica.
ISO-9002 (NBR-19002)	Sistemas da Qualidade – Modelo para Garantia da Qualidade em Produção e Instalação.
ISO-9003 (NBR-19003)	Sistemas de Qualidade – Modelo para Garantia da Qualidade em Inspeção e Ensaios Finais.
ISO-9004 (NBR-19004)	Gestão da Qualidade e Elementos do Sistema da Qualidade – Diretrizes.

Tabela 5

Os documentos ISO-9000 e ISO-9004 contêm apenas diretrizes e regras orientativas. Os documentos ISO-9001, ISO-9002 e ISO-9003 contêm a essência básica das Normas. Só tem sentido em se falar de Certificação das empresas em relação a estas três últimas normas.

4.1.8 A Série ISO-9000 Aplicada ao Software

A ISO 9000-3 apresenta diretrizes para aplicação da ISO-9001 ao desenvolvimento, fornecimento e manutenção de software. Ela se espelha nos itens da ISO-9001 que podem ser aplicados aos processos de produção de software, fazendo a necessária adaptação.

A ISO-9000-3 aplica-se àqueles casos em que existe um contrato formal entre fornecedor e cliente. Ou seja, aos casos de desenvolvimento, fornecimento e manutenção de software, desde que específico para um cliente (relação de 1 para 1). Assim, *qualidade no contexto desta norma significa, basicamente, conformidade com as especificações contratuais firmadas entre as partes.*

A ISO-9000-3 teve seu esboço inicial em 1991 e somente foi publicada em junho de 1993. Tem por objetivo complementar, facilitar o entendimento e adequar a aplicação da ISO-9001 ao software. Para cada item da 9001 relacionado ao software existe um correspondente na ISO-9000-3 que o detalha e o adequa ao software.

As diretrizes da ISO-9000-3 cobrem questões como: entendimento comum para as partes (contratante e contratada) de requisitos funcionais; uso de metodologias consistentes para desenvolvimento de software e gerenciamento do projeto como um todo (da concepção até a instalação do software no cliente).

Apesar dessas questões estarem mais ligadas ao software aplicativo dedicado a um determinado cliente, *a ISO-9000-3 pode ser estendida para atender, parcialmente, fornecedores de pacotes de uso genérico (relação de 1 fornecedor para "n" clientes).*

A ISO-9000-3 encontra-se dividida em três partes principais:

- Estrutura do sistema da Qualidade;
- Atividades do Ciclo de Vida do Software
- Atividades de Suporte

Segue a análise do conteúdo e do objetivo de cada uma destas três partes. Os roteiros passo a passo cobrem os conteúdos a seguir de uma forma sistemática e prática. Portanto, a leitura desta parte ANTES da aplicação dos roteiros é fundamental.

Constata-se que ela também é útil como complemento e consulta durante a aplicação dos roteiros, pois as questões são curtas e objetivas, não comportando, portanto, todos os aspectos relevantes que devem ser considerados.

Estrutura do Sistema da Qualidade (*Quality Framework*)

Além de diretrizes gerais, a ISO-9000-3 descreve as responsabilidades e ações relacionadas à qualidade que devem ser tomadas tanto pelo fornecedor (empresa de software) como pelo cliente, ao longo das diferentes fases do ciclo de vida do software.

A aplicação da ISO-9000-3 deve propiciar aos fornecedores de software uma política de qualidade formal, documentada, divulgada e compreendida por todos os funcionários. Esses funcionários precisam ter como parte do seu trabalho a responsabilidade e a autoridade suficientes para implementar essas políticas. A empresa, por sua vez, deve possuir pessoas e recursos para verificar, de forma independente, o emprego correto das suas políticas de qualidade. Ou seja, o executor não pode ser o seu próprio auditor. Assim, a auditoria deve estar a cargo de um setor, de um grupo interno da qualidade ou de uma empresa independente de auditoria especializada que pode ou não estar credenciada pelo INMETRO. As empresas de auditoria não credenciadas não podem expedir Certificado de Conformidade. Podem, entretanto, realizar um trabalho importante nos casos em que a empresa de software procura somente sua adequação às Normas (e não a Certificação).

Diferentemente da ISO-9001, as diretrizes da ISO-9000-3 prevêm atribuições também para o cliente. É proposto que o cliente indique um representante para negociar com o fornecedor de software as questões contratuais, incluindo definição de requisitos, definição de critérios de aceitação e acordos de conclusão. Reuniões de revisão e acompanhamento entre fornecedor e cliente também são propostas nessa primeira parte da ISO-9000-3, cujas diretrizes se repetem para cada uma das fases do ciclo de vida do software.

Resumindo, são quatro os pontos cobertos pelo capítulo da Norma que trata da Estrutura do Sistema da Qualidade:

- a) Estabelecimento de Responsabilidades Gerenciais;
- b) Definição e Documentação do próprio Sistema da Qualidade;
- c) Procedimentos para Auditoria Interna do Sistema da Qualidade;
- d) Procedimentos para Ações Corretivas.

4.1.9 Atividades do Ciclo de Vida do Software

(Software Life Cycle Activities)

Faz parte do projeto de desenvolvimento de um software a definição do ciclo de vida em fases e, dentro destas fases, as atividades de cada uma delas. Segundo Rp(95), a norma define que o desenvolvimento de software deve ser feito segundo um determinado modelo de ciclo de vida, o qual pode ser estabelecido pela própria empresa.

Uma estrutura usual das fases deste ciclo de vida é a seguinte:

- Fase 1 – Definição de requisitos;
- Fase 2 – Projeto;
- Fase 3 – Implementação;
- Fase 4 – Teste;
- Fase 5 – Liberação para produção (*manufacturing release*);
- Fase 6 – Liberação para embarque (*shipment release*);

No projeto de desenvolvimento, a prevenção de problemas deve ser mais enfatizada do que a sua correção. Assim, o Sistema da Qualidade deve atuar sobre todo o ciclo de vida do software e a Qualidade deve ser buscada em cada atividade do ciclo de vida e não simplesmente avaliada no final.

Independentemente do modelo de ciclo de vida definido pela empresa, a Norma prevê que as atividades do ciclo podem ser agrupadas em nove grandes categorias.

Categorias das atividades relacionadas ao ciclo de vida do software (segundo a ISO-9000-3):

1. Análise Crítica de Contrato (Contract Review)

Cobre itens padrões que devem constar nos contratos relativos a compra e venda de software, tais como: abrangência do trabalho, contingências e proteção de informações proprietárias.

2. Especificação de requisitos do comprador

Trata da especificação de requisitos funcionais que devem ser preparados em conjunto pelo comprador e fornecedor. Deve incluir aspectos de performance, confiabilidade, segurança e privacidade.

3. Planejamento do Desenvolvimento (Development Planning)

Enfatiza a necessidade e define um plano de desenvolvimento do software. O plano deve incluir a definição do projeto, organização dos recursos, fases, cronograma e planos de teste. O plano deve incluir também formas de controle de entradas e saídas para cada fase do ciclo de vida e um método de monitorar e verificar o progresso.

4. Planejamento da Qualidade (Quality Planning)

Aborda a elaboração de um Plano da Qualidade *específico para o projeto* em pauta que englobe, também, itens não cobertos pelo Sistema da Qualidade geral da empresa, como requisitos ou atividades especiais previstas no contrato de um determinado cliente. Deve tratar dos Objetivos de Qualidade do produto de software (exemplos: redução do reprocessamento, redução do número de chamadas de assistência técnica, redução do número de horas de treinamento etc.), critérios de saída de cada fase e entrada na seguinte, planejamento detalhado de atividades de verificação e validação, bem como responsabilidades específicas para atividades da qualidade.

5. Projeto e Implementação (Design and Implementation)

Preconiza um projeto de software disciplinado, onde comprador e fornecedor concordam, previamente, sobre o conjunto de informações do projeto (do fornecedor) que serão fornecidas ao comprador. O desenvolvedor deve usar uma metodologia sistemática e, dentro do possível, valer-se da experiência passada. O projeto deve levar em consideração as futuras atividades de manutenção e aderir à regras e convenções de programação.

6. Teste e Certificação (Test and Validation)

Notifica a necessidade de teste e homologação do software em vários níveis. Um plano de testes deve ser sugerido cobrindo alguns fatores como: ambiente, documentação, *cases* de teste e dados. A validação do sistema completo e testes de campo devem também ser abordados pelo plano de testes

7. Aceitação (Acceptance)

Cobre os termos acordados previamente e condições impostas pelo comprador para aceitação do produto. Questões como: teste de aceitação, procedimentos para avaliação, ambientes e recursos de software e hardware são abordadas nessa categoria.

8. Reprodução, Expedição e Instalação (Replication, Delivery and Installation)

Trata do registro de considerações relativas ao número de cópias, tipo de meio físico utilizado, direitos autorais e licenças, critérios de envio e obrigações do fornecedor e do comprador ligadas à instalação.

9. Manutenção (Maintenance)

Identifica a manutenção como uma questão da qualidade, onde esse serviço é incluso no contrato de compra. Atividades de manutenção, normalmente, envolvem mudanças no software, solução de problemas, correção de defeitos, modificação de interfaces, melhorias de desempenho e expansões funcionais (upgrade). A Norma propõe a existência de um plano de manutenção, documentação e critérios de liberação em função da incorporação de alterações no software.

4.2.0 Protótipo em software

O Protótipo é uma fase beta de um software. O processo de construir ou gerar um protótipo denominamos neste instrumento como: Prototipação. Muitas vezes, o cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento e saída detalhados. Em outros casos, o desenvolvedor pode não ter certeza da eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação homem-máquina deve assumir. Segundo Rp(95), nessas e em muitas outras situações, uma abordagem de prototipação à engenharia de software pode representar a melhor abordagem.

A prototipação é um processo que capacita o desenvolvedor a criar um modelo do software que será implementado. O modelo pode assumir uma das três formas: (1) um protótipo em papel ou modelo baseado em PC que retrata a interação homem-máquina de uma forma que capacita o usuário a entender quanta interação ocorrerá; (2) um protótipo de trabalho que implementa algum subconjunto da função exigida do software desejado; ou (3) um programa existente que executa parte ou toda a função desejada, mas que tem outras características que serão melhoradas em um novo esforço de desenvolvimento.

A seqüência de eventos para o paradigma de prototipação é ilustrada na figura a seguir (figura 14). Como todas as abordagens ao desenvolvimento de software, a prototipação inicia-se com a coleta dos requisitos. O desenvolvedor e o cliente reúnem-se e definem os objetivos globais para o software, identificam as exigências conhecidas e esboçam as áreas em que uma definição adicional é obrigatória.

Ocorre então a elaboração de um “projeto rápido”. O projeto rápido concentra-se na representação daqueles aspectos do software que serão visíveis ao usuário (isto é, abordagens de entrada e formatos de saída). O projeto rápido leva à construção de um protótipo que é avaliado pelo cliente / usuário e é usado para refinar os requisitos para o software a ser desenvolvido. Um processo de iteração ocorre quando é feita uma “sintonia fina” do protótipo para satisfazer as necessidades do cliente, capacitando, ao mesmo tempo, o desenvolvedor a compreender melhor aquilo que precisa ser feito.

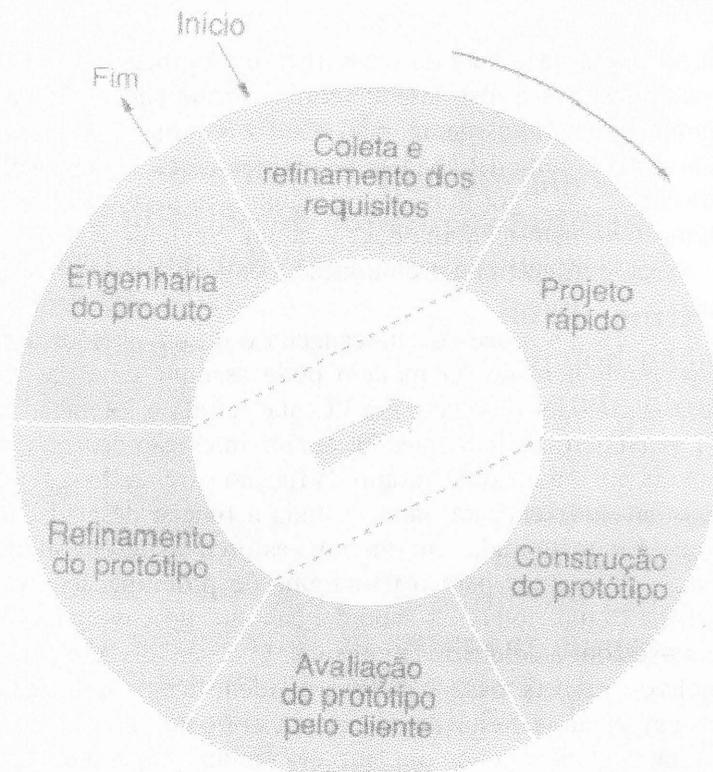


Figura 14

Idealmente, o protótipo serve como um mecanismo para identificar os requisitos de software. Se um protótipo de trabalho for construído, o desenvolvedor tentará usar fragmentos de programas existentes ou aplicará ferramentas (por exemplo, geradores de relatórios, gerenciadores de janela etc. que possibilitem que programas de trabalho sejam gerados rapidamente.

Mas o que faremos com o protótipo quando ele tiver servido ao propósito descrito anteriormente? Segundo Rp. (95) apresenta uma resposta:

Na maioria dos projetos, o primeiro sistema construído dificilmente é usável. Ele pode ser muito lento, muito grande, desajeitado em uso, ou todos os três. Não resta outra alternativa senão começar tudo de novo, dolorosamente, mas com mais habilidade, e reconstruir uma versão novamente projetada em que esses problemas sejam resolvidos ...

Quando um novo conceito de sistema ou nova tecnologia é usada, alguém tem de construir um sistema para jogar fora, porque até mesmo o melhor planejamento não é tão onisciente a ponto de fazê-lo corretamente logo na primeira vez. A questão administrativa, portanto, não é se deve construir um sistema-piloto e jogá-lo fora. Isso será feito. A única questão é se deve planejar antecipadamente a construção de algo que se vai jogar fora ou prometer entregar isso aos clientes.

O protótipo pode servir como “o primeiro sistema” sistema esse que Brooks recomenda que joguemos fora. Mas essa pode ser uma visão idealizada. Como o ciclo de vida clássico, a prototipação como paradigma da engenharia de software pode ser problemática pelas seguintes razões:

1. O cliente vê aquilo que parece ser uma versão de trabalho do software, desconhecendo que o protótipo se mantém unido “com goma de mascar e arame de enfardar”, sem saber que, na pressa de colocá-la em funcionamento, não levamos em consideração a qualidade global do software e a manutenibilidade a longo prazo. Quando informamos que o produto precisa ser reconstruído, o cliente grita improperios e exige que “alguns acertos” sejam *aplicados para torneiar o protótipo um produto de trabalho muito* freqüentemente, a gerencia (de desenvolvimento de software cede).

2. O desenvolvedor muitas vezes faz concessões de implementação a fim de colocar um protótipo em funcionamento rapidamente. Um sistema operacional ou linguagem de programação imprópria pode ser usada simplesmente porque está à disposição e é conhecida; um algoritmo ineficiente pode ser implementado simplesmente para demonstrar capacidade. Depois de algum tempo, o desenvolvedor pode familiarizar-se com essas opções e esquecer-se de todas as razões pelas quais elas são inadequadas. A opção menos que ideal se tornou então parte integrante do sistema.

Ainda que possam ocorrer problemas, a prototipação é um paradigma eficiente da engenharia de software. A chave é definir-se as regras do jogo logo no começo; ou seja, o cliente e o desenvolvedor devem ambos concordar que o protótipo seja construído para servir como um mecanismo a fim de definir os requisitos. Ele será depois descartado (pelo menos em parte) e o software real será projetado, levando-se em conta a qualidade e a manutenibilidade.

4.2.1 O Modelo em espiral

O modelo espiral para a engenharia de software foi desenvolvido para abranger as melhores características tanto do ciclo de vida clássico como da prototipação, acrescentando, ao mesmo tempo, um novo elemento – a análise dos riscos – que falta a esses paradigmas. O modelo, representado pela espiral da Figura em seqüência, define quatro importantes atividades representadas pelos quatro quadrantes da figura a seguir:

1. *Planejamento*: determinação dos objetivos, alternativa e restrições.
2. *Análise dos riscos*: análise de alternativas e identificação / resolução dos riscos.
3. *Engenharia*: desenvolvimento do produto no “nível seguinte”.
4. *Avaliação feita pelo cliente*: avaliação dos resultados da engenharia.

Um aspecto intrigante do modelo espiral torna-se claro quando consideramos a dimensão radial descrita na figura a seguir. Com cada iteração ao redor da espiral (iniciando-se ao centro e avançando para fora), versões progressivamente mais completas do software são construídas. Durante o primeiro giro ao redor da espiral, os objetivos, alternativas e restrições são definidos e os riscos são identificados e analisados. Se a análise dos riscos indicar que há incertezas nos requisitos, a prototipação pode ser usada no quadrante da engenharia. Simulações e outros modelos podem ser usados para definir ainda mais o problema e refinar o protótipo.

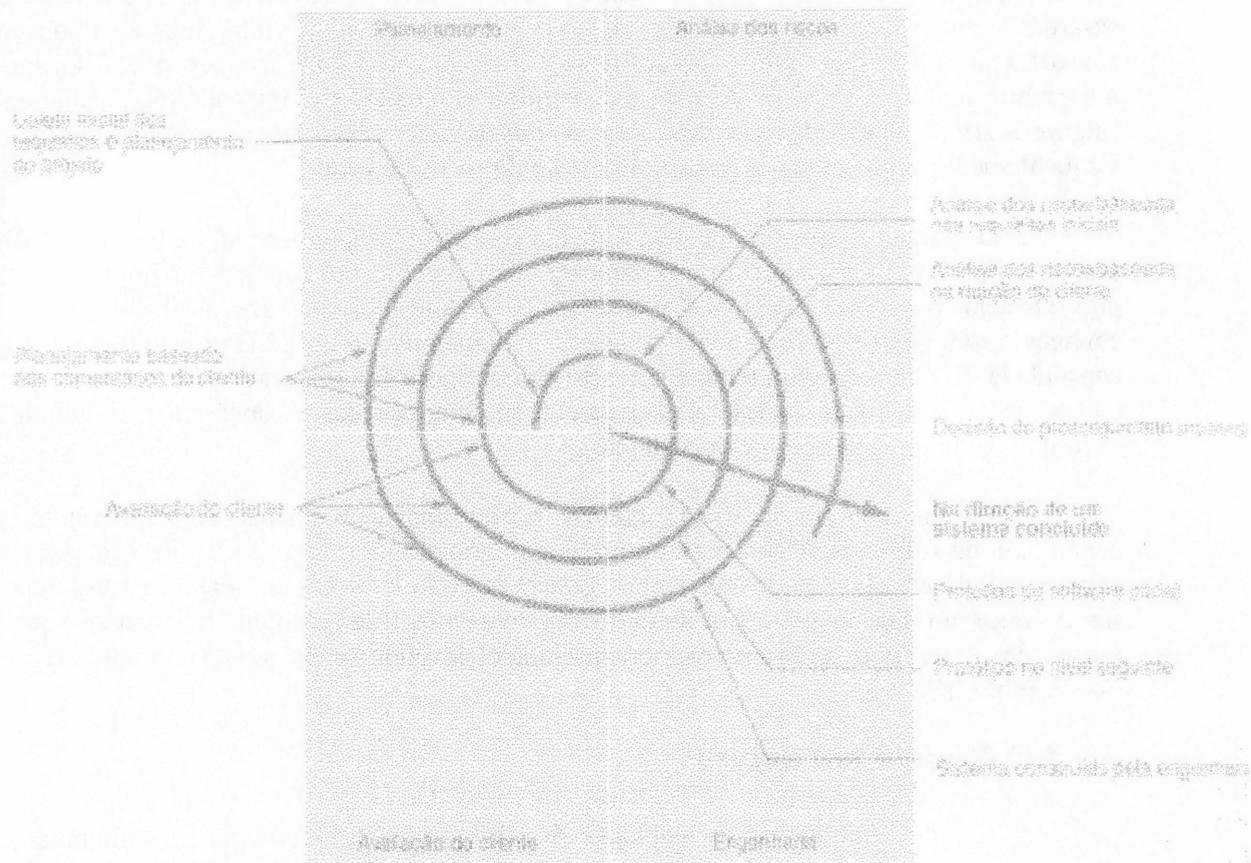


Figura 15

O cliente avalia o trabalho de engenharia (o quadrante de avaliação do cliente) e apresenta sugestões para modificações. Baseada na entrada do cliente, ocorre a fase seguinte de planejamento e análise dos riscos. Em cada arco da espiral, a conclusão da análise dos riscos resulta numa decisão de “prosseguir ou não prosseguir”. Se os riscos forem muito grandes, o projeto pode ser encerrado.

Na maioria dos casos, entretanto, o fluxo ao redor de uma trajetória espiral continua, com cada trajetória movimentando os desenvolvedores para fora na direção de um modelo mais completo do sistema e, em última análise, ao próprio sistema operacional. Cada giro ao redor da espiral requer um trabalho de engenharia (quadrante inferior direito) que possa ser levado a efeito usando-se ou a abordagem de ciclo de vida clássico ou prototipação. Deve-se notar que o número de atividades de desenvolvimento que ocorre no quadrante inferior direito eleva-se à medida que as atividades se afastam do centro da espiral.

O paradigma de modelo espiral para a engenharia de software atualmente é a abordagem mais realística para o desenvolvimento de sistemas e de software em grande escala. Ele usa uma abordagem “evolucionária” à engenharia de software, capacitando o desenvolvedor e o cliente a entender e reagir aos riscos em cada etapa evolutiva. O modelo

espiral usa a prototipação como um mecanismo de redução dos riscos, mas, o que é mais importante, possibilita que o desenvolvedor aplique a abordagem de prototipação em qualquer etapa da evolução do produto. Ele mantém a abordagem de passos sistemáticos sugerida pelo ciclo de vida clássico, mas incorpora-a numa estrutura iterativa que reflete mais realisticamente o mundo real. O modelo espiral exige uma consideração direta dos riscos técnicos em todas as etapas do projeto e, se adequadamente aplicado, deve reduzir os riscos antes que eles se tornem problemáticos.

Mas, como os demais paradigmas, o modelo espiral não é uma panacéia. Pode ser difícil convencer grandes clientes (particularmente em situações de contrato) de que a abordagem evolutiva é controlável. Ela exige considerável experiência na avaliação dos riscos e fia-se nessa experiência sucesso. Se um grande risco não for descoberto. Indubitavelmente problemas. Finalmente, o próprio modelo é relativamente novo e sido tão amplamente usado como o ciclo de vida clássico ou a prototipação Demorará muitos anos até que a eficácia desse importante novo paradigma possa ser determinada com certeza absoluta.

5. Qualidade e o Modelo RUP

5.1.1 A Concepção da excelência em RUP

Segundo Sf (98), o Rational Unified Process, ou comumente denominado RUP, é um processo de engenharia de software que tem por objetivo aumentar a qualidade e aderência dos sistemas gerados por uma organização. Da ênfase na criação e manutenção de *modelos* e foi idealizado com base revisões fundamentais. A primeira esta relacionada as atividades de desenvolvimento que são *orientadas por use case (casos de uso)* como parte funcional do sistema que agrega valor ao resultado do software e é *iterativo* com flexibilidade incremental, pequenos ciclos de projeto gerando as evoluções do produto. E por ultimo, e *"focado na arquitetura que é desenvolvida em paralelo aos use cases"*, englobando aspectos estáticos e dinâmicos do software.

5.1.2 As Fases do RUP

O Desenvolvimento de software por meio do ciclo de vida do RUP é dinâmico e mental, com marcos de progresso do processo definido com produtos e definidas passagens das fases.

Concepção ou iniciação

Esta fase é a que estabelece o caso de negócio para o projeto. Visa a compreensão da proposta e da tecnologia através da definição dos *Use cases* mais críticos. Inclui critérios de sucesso, avaliação de riscos, estimativas de recursos necessários e um plano de trabalho.

No final dessa fase, os objetivos do ciclo de vida são avaliados juntamente com o nível do produto, os riscos e a viabilidade do projeto, sem desviar a atenção da proposta de negócio empresarial.

Elaboração

Esta fase é que estabelece um plano de projeto e uma arquitetura sólida. Visa o detalhamento da arquitetura do software e a descrição dos requisitos do software gerados em use cases. No final dessa fase, são avaliados o escopo, o detalhamento dos requisitos funcionais (use cases), a arquitetura do software, os riscos, os custos, o plano de trabalho e o plano de construção.

Construção

Fase de desenvolvimento do sistema. Visa a construir o *software* e preparar os clientes do mesmo. Contempla a descrição dos critérios de aceitação dos usuários que são trabalhados em testes conjuntos. Também são produzidos os manuais documentação. No final dessa fase são avaliados os produtos do software, inclusive os manuais.

Transição

Fase que disponibiliza o sistema aos clientes. Visa também a fornecer treinamento aos usuários finais. Alguns ajustes podem ser elaborados. No final dessa fase são avaliados os resultados gerais do software, inclusive o nível de satisfação da equipe como um todo.

5.1.3 Workflows do Ciclo de Desenvolvimento

Segundo Sf (98), a passagem pelas fases e chamada de ciclo de desenvolvimento que gera um *software* que é adicionalmente dividido em iterações e finalizado com ponto a avaliação para analisar os objetivos e os resultados propostos.

A iteração é estruturada em *workflows* de processo ou fluxos de trabalho, realizados pelas pessoas da equipe do projeto. São nove componentes:

Modelagem de Negócio

Descreve a estrutura dinâmica da empresa. Objetiva o entendimento comum nos envolvidos sobre quais processos de negócio devem ser contemplados. Pode ser elaborada por meio dos *use cases* de negócio.

Requisitos

Descreve o método baseado em casos de uso para identificar requisitos. Visa entender os requisitos funcionais que serão contemplados pelo software. Sua ênfase está direcionada para as fases de concepção e elaboração, a fim de compreender e delimitar o escopo do software.

Análise e projeto

Descreve as várias visões da arquitetura do software. Convencionalmente essas fases eram separadas. Em conjunto, essas atividades visam a detalhar e compreender os *Use cases*

definidos nos requisitos funcionais do software. Geram os modelos dos sistemas focados na futura implementação adequada ao ambiente de funcionamento. Sua ênfase esta direcionada na fase de elaboração e posteriormente estará direcionada no início da fase de construção.

Implementação

Desenvolve, testa e integra o software em elaboração. Visa a estruturação e elaboração dos códigos de programação para implementar as classes e os objetos e seus componentes. Elabora também os primeiros testes dos componentes em termos de unidades. Realiza a integração dos códigos produzidos. Sua ênfase esta direcionada na fase de construção do software.

Teste

Descreve os casos de teste, procedimentos e medidas de acompanhamento de erros. Visa a avaliar se os requisitos funcionais foram atendidos com qualidade. Também analisa os eventuais defeitos que serão removidos antes da implantação. Os modelos de testes são criados para descrever como os testes serão realizados. Sua ênfase esta mais direcionada na fase final da construção e no início da fase de transição.

Entrega

Contempla a configuração do sistema a ser entregue. Visa a produzir resultados *releases* do produto e entregá-los aos clientes *e/ou* usuários finais. Também inclui, se necessário, as atividades de beta-teste, migração dos dados *e/ou* software existente. Abrange inclusive a aceitação formal com assinaturas dos envolvidos.

Gerenciamento da configuração

Também chamado de gerenciamento de mudanças. Esse *workflow* e considerado de apoio e visa suprir algumas das lacunas deixadas pelo RUP original. Controla as modificações e mantém a integridade dos artefatos e produtos de *software*. Visa a controlar e garantir a integridade das diversas versões dos produtos produzidos durante o projeto, assegurando que os resultados não conflitem entre si.

Gerenciamento de projeto

Descreve as varias estratégias para a gestão do trabalho como um processo interativo. Também e considerado de apoio para suprir algumas lacunas deixadas IX UP original. Contempla *um framework* para a gestão total do projeto, com recomendações para o planejamento de tarefas, alocação de recursos e gerência de riscos.

Ambiente

Contempla a infra-estrutura necessária para o desenvolvimento do sistema. Também é considerado de apoio para suprir algumas lacunas deixadas pelo UP original. Abrange

também a descrição das ferramentas para execução de todos os processos. do software.

5.1.4 Artefatos e Modelos

Cada atividade do RUP tem artefatos associados ou exigidos como uma entrada gerados como saídas. Uns com finalidade de direcionar a entrada para atividade subsequentes mantidos como recursos de referencia sobre o projeto e outros gerados em formatos de entregas contratuais ou produtos do software

Segundo Sf (98), os modelos são o tipo mais importante de artefato do RUP. Um modelo é simplificação da realidade, criado para proporcionar um melhor entendimento do software em elaboração.

São nove os modelos que em conjunto abrangem todas as decisões importantes para visualização, especificação, construção e documentação de um software.

O modelo de negocio contempla uma abstração da empresa. O modelo de domínio contempla o contexto do sistema. O modelo de caso de "SO contempla os requisitos funcionais do sistema". O modelo de análise contempla a idéia do projeto do software. O modelo de projeto contempla o vocabulário do problema e a sua solução. O modelo de processo contempla os mecanismos de concorrência e de sincronização do sistema. O modelo de implantação contempla a topologia do hardware necessária para executar o sistema. O modelo de implementação contempla as partes utilizadas para executar o sistema do ponto de vista físico. E o modelo de teste contempla os caminhos de verificação e validação do software.

A arquitetura de um sistema no RUP e percebida em cinco visões: projeto, processo, implantação, implementação e caso de uso.

Os artefatos são categorizados como de gerenciamento ou técnicos. Os artefatos técnicos do RUP são os conjuntos de: requisitos, projeto, implementação e implantação: O primeiro descreve o que o software deve fazer. O segundo descreve como o software será elaborado do ponto de vista lógico. O terceiro descreve como o software será elaborado do ponto de vista físico. E o quarto descreve sobre o empacotamento, entrega, instalação e execução no ambiente do cliente ou usuário final de destino.

6. Estudo de caso “Migração de Tecnologia”

Atenção, o relato e a descrição deste projeto são meramente pedagógicos, sem nenhuma veracidade sobre as hipóteses, e ou conjectura apresentadas, porém sendo focado como forma de condução para o aprofundamento dos temas a que este trabalho se ressalva.

6.1.1 Breve Relato

O estudo de caso aqui proposto como “Migração de Tecnologia” embasa-se em um breve projeto que está situado em um banco cujo qual se referenciará como “NETBANK”. O Netbank possui dentre os seus diversos departamentos, o departamento de seguros de vida, de automóveis e demais ramos. Para este estudo de caso somente será abordado e

mencionado fatores relevantes ao seguro de auto (*Automóveis*) com a finalidade de otimizar os objetivos deste projeto, para sua melhor aplicação e elaboração.

Sobre o contexto descrito acima, o Netbank possui uma *carteira* (*entende-se por carteira como: um conjunto de clientes que usufruem dos serviços de seguros prestado pelo banco, "NetBank"*), de clientes que são membros do exército do estado do Acre.

A um seletor grupo de clientes, que fazem parte da carteira de seguros do Netbank se denomina como *estipulante*, as pessoas a qual possuem um vínculo com a seguradora do banco. O objetivo maior deste projeto é migrar a antiga tecnologia de controle de seguro de auto do *mainframe*, (*entenda-se aqui para fins explicativos que: mainframe são computadores de grande capacidade de processamento*) para plataforma em *Unix* (*entenda-se aqui para fins explicativos que, a plataforma em Unix é como um sistema operacional de computadores, como o Microsoft Windows*)

6.1.2 Histórico

Este projeto foi iniciado para contemplar melhorias nos sistemas de emissão e cadastro da carteira de Automóvel, visando atender características diferenciadas para clientes de grande porte, em especial os seguros do pessoal do Exército do Acre, através do estipulante AMA – Abrigo Militar do Acre.

O processo de emissão atual para as apólices AMA foi desenvolvido no ambiente Mainframe, o qual se encontra tecnologicamente desatualizado. Existem dificuldades de manutenção e novas necessidades foram identificadas o qual este projeto se propõe a atendê-las da melhor forma possível dentro das normas de compliance e políticas de segurança do NETBANK Seguros.

Por isso, o objetivo principal deste projeto é viabilizar a migração dos seguros emitidos no sistema Mainframe para os sistemas desenvolvidos no ambiente Unix.

Migrar para o ambiente Unix significa integração dos processos de comercialização, gerenciamento de produção, sinistro, renovação, cadastro de apontamentos, central de bônus, vistoria prévia, consultas e outros aplicativos que foram desenvolvidos ou que venham a ser desenvolvidos nesta plataforma.

6.1.3 Objetivos do Negócio

O objetivo principal deste projeto é viabilizar a migração dos seguros emitidos no sistema Mainframe para os sistemas desenvolvidos no ambiente Unix. Migrar para o ambiente Unix significa integração dos processos de comercialização, gerenciamento de produção, sinistro, renovação, vistoria prévia, consultas técnica e outros processos que forem desenvolvidos ou que venham a ser desenvolvidos nesta plataforma e que viabilizem as propostas de seguro de auto.

O direcionamento dado pela área de Negócios da Seguradora foi viabilizado em função de agilidade e estratégia de negócios, permitindo em menor tempo e menos desgaste,

implementar as solicitações que tornarão o produto mais rentável e ágil. Objetiva-se implementar as funcionalidades descritas em apenso.

Pelo fato de o Netbank, ser responsável pela carteira de seguro de autos da AMA (Abrigo Militar do Acre) as entradas de vendas das apólices de seguros são processadas em sua central no (AMA), e posteriormente transmitidas via web, através de uma tecnologia denominada *omnidata* (*tipo de tecnologia de transmissão de dados*) a seguradora responsável do Netbank, que posteriormente irá processar as apólices recepcionadas. Uma das metas deste projeto é gerar um conversor de dados que estará convertendo novas funcionalidades sistêmicas, requeridas neste projeto para a plataforma Unix, onde veremos a seguir mais detalhado, logo abaixo, onde o aplicativo de seguros, com as otimizações necessárias para administrar o produto Auto.

- Franquia Majorada;
- Agrupamento de Regiões de Risco;
- Perfil do Condutor;
- Bônus diferenciado;
- VMR, VD;
- Desconto Fidelidade;
- Cobertura – incêndio;
- Assistência 24 horas;
- Carro Reserva;
- Comercialização de Garantia de Vidros;
- Valor de Zero até seis Meses;
- Parcelamento do Prêmio em até 10 vezes;
- Registro dos Segurados e agregados (Cenário do Consignante);
- Adicional de fracionamento;
- Isenção de custo na emissão da Apólice;
- Cadastro de apontamentos
- Central de Bônus
- Identificar Pró-labore não dedutível;
- Emitir segunda via do contrato com pós-venda personalizado;
- Gerar arquivo de propostas aceitas para o sistema corretor;
- Disponibilizar informações do serviço de assistência 24 horas a Mondial;
- Desconto para pagamento à vista;
- Desconto para segundo veículo (frota);
- Automatização de regras para ajuste de taxas diferenciadas

6.1.4 Controles de Auditoria

Os processos para extração de informações e os controles necessários para trilhas de auditoria serão os mesmos já implementados nos sistemas e aplicativos integrados ao Unix, compreendendo:

- Transações on-line estão fortemente baseados no processo de autenticação existente no sistema de login, utilizado pela seguradora;

- Registro da data em que as propostas foram transmitidas pelo sistema;
- Registro da matrícula do funcionário que analisou a pendência da proposta, bem como a data e o motivo da liberação/recusa efetivada pelo analista do risco;
- Registro da data em que as propostas foram transmitidas do corretor para emissão;
- Backup e logs de atualização do banco de dados;
- Arquivos fontes de programas;
- Modelos de dados do sistema, disponíveis do System Architect (Sistema de Documentação padronizado pelo Netbank);
- Documentação do sistema no System Architect (fluxos de serviço, diagramas de contexto, diagramas de navegação, modelos de funções);
- Documentação gerada pelo uso da metodologia do Netbank.

6.1.5 Impactos no negócio

Com a adoção do processo de emissão através do sistema kit seguros e Comercialização de Seguros, sistemas internos de seguro, o negócio terá como característica a transparência e adequação aos padrões estabelecidos para a carteira de seguros automóvel na cia. Com essa adequação, os impactos no negócio poderão ser assim percebidos:

- Desativação do processo no sistema Mainframe após o direcionamento para o Unix de todas as apólices vigentes naquele sistema – a consequência será a redução de esforços de manutenção em IT;
- Disponibilidade de informações confiáveis no Unix, facilitando a regulação de sinistros - a consequência será a redução de esforços da área de indenizações na busca de informações com o corretor;
- Acompanhamento gerencial da conta – a consequência será a agilização na tomada de decisões para corrigir rumos de resultado da conta, permitindo evitar perdas financeiras no negócio;

6.1.6 Impactos no Ambiente Operacional

O processo de “migração” vai requerer alterações nos sistemas de Comercialização de Seguros já existentes no Netbank em um “**Impacto no Ambiente Operacional**”, sobre os sistemas atuais do banco tais como: Kit seguros, Gerador de Produtos de Seguros, Consulta Técnica, Cobrança, Cadastro de Clientes, (Produção, Estatística e Rentabilidade), (Pós-Venda personalizado), Renovação, Indenizações, Reservas, Estipulantes, Vistoria Prévia.

Outro impacto será a convivência entre os dois sistemas (Unix e Mainframe). Isto será necessário, pois a solução não prevê migração automática das bases de dados de um

sistema para o outro. A convivência se dará até que sejam emitidos no Unix todos os itens vigentes atualmente no Mainframe. Para suportar esse período de convivência as diversas áreas envolvidas no processo deverão estar aptas para atuar nos dois sistemas.

As áreas de produção e arrecadação do Netbank serão aquelas com maior impacto em função da necessidade de controle e acompanhamento proporcionados pela nova forma de receber os dados do estipulante.

As áreas de Logística/Expedição terão aumentada a quantidade de documentação para separar, montar o “kit” do segurado e enviar. Isto poderá demandar a necessidade de contratação de pelo menos mais dois recursos.

Por ocasião da preparação das especificações funcionais do Desenho Externo, deverá contemplar espaço para reuniões e debates com os gestores de todas as áreas e sistemas que sofrerão algum impacto com a mudança do processo.

6.1.7 Descrição Geral

Atendendo os diferenciais exigidos em contrato através de licitação, a solução até então adotada foi a utilização do processo de emissão no sistema Mainframe, assumindo-se a aceitação, sem restrição, dos riscos emitidos pelo corretor da conta.

Com a criação do novo sistema de comercialização para o seguro de Auto, onde os módulos, formados por uma série de unidades, estão conectados através de uma rede corporativa aos servidores centrais, utilizando o gerenciador de banco de dados.

As unidades utilizam um conjunto de aplicativos INBR, desenvolvidos na linguagem Java, que são executados em cada PC que recebem nova versão dos aplicativos periodicamente. Além de aplicativos desenvolvidos em banco de dados, existem aplicativos que executam nos Servidores, desenvolvidos em “C”.

Com base na plataforma Unix, os processos do Mainframe ficaram defasados e desatualizados, inviabilizando quaisquer controles gerenciais do respectivo negócio, desde a contratação até o recolhimento do prêmio. Enfim, os riscos de perdas financeiras com significativo impacto no lucro da empresa podem ser levantados como possibilidades de hipóteses.

Diante do acima exposto, a adequação do referido negócio ao processo adotado atualmente pela Seguradora é entendida como urgente e muito importante. Será considerado neste projeto, estudo para instalação da solução para procedimentos de cálculo e emissão de propostas em ambientes que atendem a contas similares às do AMA.

Serão desenvolvidos processos que permitam a recepção de propostas da AMA Corretora em seus escritórios, consistência, respostas de processamento, emissão de propostas e sistemas de gerenciamento e acompanhamento dos negócios relacionados ao AMA.

Serão desenvolvidos processos que permitam a recepção de propostas das Corretoras localizadas no AMA, em seus escritórios, consistência, respostas de processamento, emissão de propostas e sistemas de gerenciamento e acompanhamento dos negócios relacionados ao AMA.

Em resumo, o escopo compreenderá:

- Realizar alterações e melhorias nos sistemas Comercialização de Seguros, e afins, melhorias em Sistemas e Aplicativos, de forma a atender os requisitos do negócio;
- Disponibilizar acesso do corretor às informações referentes ao seu negócio específico no ambiente do corretor,
- Necessidades de Acessos e Informações ao Corretor e Estipulante. Como definição o corretor não poderá ter acesso a outras informações senão aquelas vinculadas aos estipulantes AMA ou ao seu código de produtor na Seguradora;
- Enviar arquivos ao corretor com informações para controle e gerenciamento da conta junto ao estipulante. Os arquivos deverão conter as informações existentes nos: Contrato, Risco, Garantias, Cláusulas, Prêmios, Parcelas, Acessórios e Endossos. Os arquivos devem permitir que a Corretora promova junto ao AMA a implantação dos “descontos em folha” sempre que o meio/forma de pagamento assim requerer.
- Promover a criação e desenvolvimento de pós-venda personalizado para o estipulante. As capas do pós-venda deverão conter o símbolo do AMA. No item “contrato” deverão ser acrescentadas as Condições Especiais pactuadas com o AMA.
- Habilitar com treinamento e recursos tecnológicos os escritórios do corretor para efetivar as emissões de propostas e endossos diretamente para os sistemas da Seguradora.

Nos escritórios do exército será mantido o sistema utilizado atualmente para emissões de propostas de seguros, segue emissão nos distritos militares, centralizando no 1º Distrito Militar, sendo então transmitidos ao Netbank via caixa postal, sistema Ominidata, devem ser convertidos e então consistidos de acordo com as regras de negócio do sistema de seguros. As irregularidades serão sinalizadas para intervenção e regularização da seguradora quando for o caso de solicitar.

A conversão será necessária para alimentação do Sistema Comercialização de Seguros. E a consistência será com base nas regras cadastradas no sistema de seguros da seguradora.

Semanalmente o AMA, enviará um arquivo que conterà as novas inclusões de apólices de seguro e alterações (endossos) nas propostas já existentes. Neste arquivo recepcionado pela seguradora, deverá ser realizado alterações para contemplar informações necessárias ao processamento das propostas de seguro Auto em consonância com o layout atual do sistema de seguros.

6.1.8 Processo Operacional

No Unix, os sistemas Comercialização Automóvel, Comercialização de Seguros e são, atualmente, as principais ferramentas para efetivação da contratação e emissão das apólices de seguros. Para tanto, são cadastradas as regras de aceitação de riscos, tabelas de taxas, geração de pós-venda, e várias outras funcionalidades, que atualmente existe, no sistema de seguros da seguradora do Netbank.

O Exército, atualmente, não participa deste processo e o Mainframe continua ativo para contemplar, basicamente, o negócio com o AMA.

No entanto, independente do sistema adotado para o negócio de seguros, são requeridos padrões operacionais que envolvem, entre outros: cotação, vistoria prévia, proposta, análise de risco, regras de aceitação, alçadas de liberação, condições contratuais, gerenciamento de resultados, etc. Com as deficiências do Mainframe, nem todos os padrões são possíveis no estipulante AMA.

Dessa forma, ao possibilitar que o estipulante AMA faça parte do Unix, as propostas serão processadas e consistidas pelo sistema de Comercialização de Seguros e estarão sujeitas as regras cadastradas no sistema de seguros. O gerenciamento do processo de emissão, aceitação do risco e pós-venda ocorrerá da mesma forma como ocorre atualmente para os contratos de estipulantes convencionais, observando-se as particularidades contratuais com o AMA o qual é objetivo desta proposta.

As implementações realizadas nos aplicativos que gerenciam as apólices de automóvel servirão para integrar o abrigo militar do Acre, no contexto atual existente e servirá de diferencial para comercializar novos produtos, tornando flexível e abrangente o leque de produtos ofertados.

6.1.9 Visão da arquitetura

O ambiente operacional INBR, onde está inserido o atual sistema de seguro, considerando todos os seus módulos, é formado por uma série de unidades, as quais estão conectadas através de uma rede corporativa aos servidores centrais, que estão localizados nos Centros Administrativos do Netbank. São servidores HP com sistema operacional UNIX, utilizando o sybase como gerenciador de banco de dados.

As unidades utilizam um conjunto de aplicativos desenvolvidos na linguagem Java, que são executados em cada PC que recebem nova versão dos aplicativos periodicamente.

Além de aplicativos desenvolvidos em Java, existem aplicativos que executam nos Servidores, desenvolvidos em "C".

Para o cliente, AMA, será desenvolvido novo módulo denominado Conversor. Este módulo será desenvolvido na linguagem Java para conversão e auditoria dos dados com posterior implementação na base de dados, as etapas de apresentação serão customizadas para aceitar novas funcionalidades aos moldes atuais.

Os dados referentes aos produtos de Auto, ramo 31 e 53, hoje combinados e tratados com o 10 nos arquivos de envio do exército, serão reformulados e implementados para atender o cliente AMA, dando maior agilidade e evolução na carteira, com significativo aumento nas emissões e renovações de apólices.

6.2.0 Visão do Negócio

Atual

Atualmente o processo de cotação e emissão de proposta do exército é realizado e imputado via Mainframe. Existe dificuldades de manutenção em processos como; garantias, taxas, descontos comerciais, agilidades nos processo de emissão, integração, de informações recebidas do produtor externo, agilização do pós-venda, regulação de sinistro.

Devido às dificuldades encontradas para gerenciar a carteira do exército e a falta de produtos a oferecer entende-se que customizando este processo, o Netbank Seguradora poderia alavancar em 2/3 a mais a atual carteira existente, sem contar os produtos recorrentes que podem agregar a carteira Banco.

Novo

Será viabilizado a migração das propostas de seguros emitidos no sistema Mainframe para os sistemas desenvolvidos em ambiente Unix, isto é, pode esta iniciativa partir no momento de renovar a apólice e nas futuras emissões de propostas, iniciando por sub-estipulante (*uma categoria de estipulante*) e evoluindo mensalmente até atingindo todos os estipulantes do AMA.

Implementar agregados ao corpo técnico do exército e Civis funcionários, dando maior volume de emissões de propostas, personalizando o pós-venda, otimizando algumas funções de sistemas hoje inexistente ao corretor do exército, disponibilizar maior funcionalidade na consulta realizada no site do Netbank, na opção de parceiros, corretores de seguros, onde o nível de acesso hoje utilizado pelos corretores é muito baixo, sendo que este volume deveria ser bem aquém de 2.500.000 consultas/mês em função de utilizarem o endereço web (www.Netbank.com.br), para verificar possíveis renovações, vigências das apólices e início de regulação de sinistro, pode haver a necessidade de treinamento na ferramenta e hipoteticamente, necessite ser adicionado ferramentas mais voltadas ao cliente do exército, agilizando seu processo de gerenciamento de apólices.

6.2.1 Estratégia de Implantação

Com o indicativo de implementar a proposta de conversão de dados nesta primeira fase e posteriormente no ambiente do cliente AMA (Abrigo Militar do Acre), foi dado ênfase a versão que causa menos impacto de alterações nos aplicativos (AMA) e que demande maior agilidade nos seus processos de emissão e gerenciamento das propostas e/ou apólices, conseguindo retorno rápido dos resultados esperados pelo exército e pelo Netbank, devendo considerar que as consultas necessárias no dia a dia do exército serão realizadas através de módulo de consulta técnica disponível na internet, através do site do Netbank no endereço web (www.Netbank.com.br), opção Corretores, parceiros.

A implementação do Conversor das apólices de seguro, regras de cálculos, funcionalidades novas e parametrizações ocorrerão no aplicativo de seguro do Netbank,

onde se viabilizará de forma sistemática e gradual devendo contemplar os itens abaixo relacionados:

Definição e Detalhamento de Escopo do Projeto - AMA

Transmissão de dados de novas propostas.

Esta funcionalidade diz respeito ao envio das apólices de seguros do AMA, a seguradora do Netbank, onde serão customizados a performance de envio de dados, e a devida conferência das apólices enviadas.

Deverá ser referenciada a definição necessária para alterar o layout hoje transmitido ao NETBANK Mainframe, que doravante deverá ser transmitido ao UNIX, resultando em mais informações para regulagem do risco proposto em cada proposta de seguro Auto.

Transmissão de dados de endossos de propostas.

As informações de endossos de propostas deverão obedecer a regras estabelecidas pelo ambiente UNIX no que tange o Conversor e o Auditor, será realizada a importação dos dados do Mainframe (Par o caso de não ser enviado todos os dados da proposta de seguro Auto) juntamente com os dados enviados de endosso e posteriormente enviado ao Conversor de apólices de seguro, para as devidas consistências e conversões de layout.

Transmissão de dados de propostas inconsistentes.

Elaborar parâmetros de layout de retorno das inconsistências para devidos ajustes e reenvio, os casos serão analisados pelo Nebank - Seguros via on-line em tela projetada, hoje existente, para liberar ou aceitar as inconsistências ocorridas no processamento das propostas do exército. A transmissão continua sendo realizada via Ominidata.

Deverá ser apresentado e acordado com o exército as definições apresentadas e necessárias para implementar layout necessário à transmissão com informações que norteiem os ajustes e alterações que se façam necessárias para retransmitir as propostas de seguro Auto para o Netbank realizar o aceite e emissão das mesmas.

Disponibilizar arquivos de:

- Registros inconsistentes (Primeira verificação).
- Registros pendentes / recusados pela seguradora com motivo da recusa (Mensagem explicita).

Alinhar Informações adicionais ao layout do exército.

Repassar o exército em tempo hábil, novas informações a serem adicionadas ao layout de envio de informações de propostas do exército, face às novas adaptações e funcionalidades disponibilizadas, envio de técnico de IT ao 1º Distrito Militar do Acre.

Adaptação de layout do Mainframe para o Unix.

Estudo de caso para qualificar layout de dados do Mainframe para o Unix, consistências hoje utilizadas devem ser substituídas pelas regras de negócio utilizadas no ambiente Unix. Para atender a migração destes itens do Mainframe para o Unix em Sybase, com Java.

Processo de consistências das Apólices de seguro.

Depois de realizado a conversão de dados e inserção destas informações no aplicativo da seguradora, com base no cadastro de regras, os métodos de consistências serão aplicados e recálculos serão realizados para preenchimento da planilha de prêmios. Caso se verifique novamente inconsistência, deverá retornar arquivo indicando pendência.

Para especificarmos como e quais consistências devem ser feitas, foi sugerido que se analisasse o procedimento de consistência realizado nas propostas oriundas do Corretor.

Conceito de tratamento de clientes segurados do exército.

Deve-se analisar os impactos relacionados no ambiente do cliente quanto a formatos e tratamentos diferenciados hoje utilizados na recepção do Mainframe. Com a passagem destas informações do ambiente Mainframe para ambiente Unix, deveremos observar os atributos de campo, consistências realizadas, e armazenamento das mesmas.

Desconto para pagamento a vista, cliente AMA.

Para mantermos o conceito da planilha de prêmios atualmente utilizada por todos os produtos emitidos pelo Netbank na plataforma UNIX, considerar que o AMA do exército fornece desconto para pagamento à vista do seguro Auto.

Sendo importante manter o conceito de cada um dos prêmios da planilha, consideramos o fato de que o exército concede desconto para pagamento à vista.

Concluimos que o juro já está embutido no prêmio líquido calculado pelo sistema do exército. Portanto, a diferença entre o prêmio calculado e o prêmio concedido é o adicional de juros.

Para dirimir esta diferença, será necessário alterar o layout atual a fim de obter os prêmios por garantia antes e depois do desconto por pagamento à vista. A intenção é de que os demais prêmios que compõe o cálculo serão deduzidos por fórmulas que constituem estes prêmios, pelo utilitário conversor a ser desenvolvido.

Gerar prévia de faturamento mensal o exército.

Antes de emissão da fatura mensal o exército, gerar com antecedência de 07 dias ou sob solicitação, prévia do que será faturado mensalmente. O exército utilizará esta prévia de faturamento para realizar ajustes nas propostas no que tange a novas propostas ou endossos de substituição, cancelamentos ou renovações.

Vale ressaltar que o faturamento deve ser segmentado com base em estipulantes e sub-estipulantes separados por meio de pagamento, o que eliminaria o processo manual desta baixas de pagamento e os possíveis impactos que o processo manual pode originar.

Diversificar conceito de Carro reserva.

Parametrizar novas prospecções para carro reserva. Necessita alteração no Sistema de seguros e no Comercialização para exibição do descritivo do padrão do veículo com as respectivas alternativas de diárias.

Certificado, proposta de alteração.

Substituir o certificado atualmente utilizado pelo exército por proposta de seguro individual Auto, substituir apólice pelo conjunto; Estipulante, sub-estipulante, posto, segurado. Esta proposta passando pelo aceite do exército será confeccionada pela seguradora do Netbank.

Procedimento para implantação e testes.

Deverá ser priorizado a implantação por seguros novos que venha a ser emitido pelo 1º Distrito Militar do Acre e possíveis endossos de renovações, já os endossos de cancelamento devem ser realizados no Mainframe e para os endossos de substituição de bens poderemos estar executando no Mainframe ou implantando no Unix como emissão nova, mas demandará de alterações via programas corretivos que não os usais em função do tempo de vigência da apólice e demais itens.

6.2.2 Descrição Geral do Processo

Consoante o explicitado no Estudo de Viabilidade e de acordo com as propostas elencadas pela área de Negócios, adaptaremos a plataforma Unix para a comercialização das apólices novas de Auto e migração das apólices a serem renovadas atualmente no Mainframe de acordo com a nova diretriz de aceite entre exército e Netbank.

Os processos, na sua maioria, serão implementados paulatinamente, devendo priorizar a parametrização do sistema de seguros quanto a funcionalidades possíveis de disponibilizar no primeiro momento.

No processamento diário do sistema de seguro Auto, serão tratados as propostas de emissão de seguro auto comercializadas pelo canal do pessoal do exército do acre, a partir das novas emissões de propostas e no momento das renovações existentes no Mainframe, os valores dos prêmios de seguro dos consignantes e agregados serão convertidos para implementarem o Unix através de conversor e seguir regras de negócios, através de módulo auditor que seguirá regras hoje instanciadas através do canal de negócios existente na plataforma distribuída, a metodologia a ser utilizada, será a atual existente na plataforma Unix.

Porém, para cada emissão de proposta, utilizaremos o recálculo através das regras de negócio hoje vigentes nas cotações realizadas através do sistema de seguros, mas com a diferença de executar, após a chegada das transmissões via caixa postal (ominidata) do exército para o Netbank.

O tratamento em casos de contingência, exceções e ajustes, nos valores calculados de prêmio de seguro Auto, ocorrerão pela informação repassada pela área de negócios, da

seguradora do Netbank, de qual será o desvio padrão aceitável para não declinar as propostas de seguro.

Área de negócios deve determinar em que momento será gerada a prévia da fatura mensal e qual será a data para emissão da fatura mensal definitiva, após ajustes técnicos realizados entre a prévia e a fatura propriamente dita pelo cliente AMA

A parametrização necessária para completar o processo de conversão de informações proveniente do exército será implementado no sistema de seguros, onde será criado o produto "Auto Corporate", com garantias próprias, taxas e ajustes técnicos, onde será parametrizado e cadastrados vidros, incêndio, carro reserva, franquia majorada, Vmr (Valor referencial de mercado) e Vd (Valor determinado), região de risco diferenciada, perfil de condutor, bônus, valor de 0 Km até 6 meses, devendo as demais alterações serem implementadas com customização de telas e programas, sendo direcionado da melhor forma para conciliar o produto conforme o acordado com as expectativas do cliente entre AMA (Abrigo Militar do Acre) e a Seguradora do Netbank.

6.2.3 Processo aliado à Qualidade

O escopo e sua definição detalha, apresentada acima, é embasada dentro das melhores práticas de qualidade e conformidade estipuladas pelo Netbank Seguros, contemplando os pré-requisitos do cliente AMA, alinhando as boas práticas e foco voltado a qualidade e excelência em Gerenciamento de projetos voltados a TI, dos quais foram apresentadas nos itens anteriores (dois. Gerenciamento de projetos três. Escopo em Projetos, quatro. Qualidade em Projetos) e incorporadas intrinsecamente no decorrer deste projeto tornando foco e chave para o sucesso do estudo de caso do Netbank seguros, do projeto "Migração de Tecnologia" para seu cliente AMA (Abrigo Militar do Acre).

Em nível gerencial podemos concluir que os processos de levantamento de requisitos, planejamento de ações, auditorias, controle interno de desenvolvimento das soluções propostas, cumprimento dos cronogramas estipulados como metas, são fundamentais para a qualidade assegurada deste projeto

Ainda se ressalva a salientar que o processo de desenvolvimento de software padronizado e homologado com uma política da qualidade definida e seguida por todos os integrantes do projeto revela uma considerável maturidade no modelo de CMM, para a organização na condução de seus projetos aqui retratados como estudo de caso: Migração de Tecnologia.

6.2.4 Resumo Qualitativo

Este projeto desenvolvido e implementado dentro das normas de qualidade e conformidades dentro das mais atuais políticas e práticas de definição de escopo, gestão da qualidade, tais como: ISO9000, CMM dois, e tendência para o desenvolvimento de software obedecendo aos preceitos de *software development* em RUP, reflete a importância da teoria embasada para uma prática voltada a excelência do projeto em sua conclusão.

Todos os estágios do projeto, assim como cada fase de aprimoramento são reflexos de sucessivas fases de maturação e desenvolvimento das necessidades que o projeto

demonstrou. Cada estágio em seu ciclo natural de crescimento se revelou em uma instância mais intrínseca através de suas funcionalidades abordadas e aqui comentadas tais como apresentadas no item detalhamento de escopo e estratégia de implementação.

A qualidade de software development, não pode ser atribuída única e exclusivamente a adequação de um projeto através de um produto final, mas também é fundamental observarmos como foi adotada e estabelecidas condutas e padronizações, para que se obtenha a almejada qualidade como foco de excelência no projeto.

7. Conclusão

Com o crescimento das organizações, e a evolução na forma de trabalhar e conduzir projetos, o comportamento na execução de projetos se tornou de uma forma dinâmica, otimizada e orientada a resultados práticos e acessíveis, nos quesitos de necessidades das grandes corporações.

No entanto estas mudanças trouxeram marcos de evolução para as indústrias, companhias, e empresas em geral. A real necessidade de estar sempre em desenvolvimento nos quesitos de produção, qualidade, expectativas do cliente, e atuante em um ambiente globalizado e cada vez mais competitivo, são fatores os quais consolidam a constante transformação das organizações em seus objetivos. Esta constante transformação é consequentemente um ente para a evolução de boas práticas e a procura para um aperfeiçoamento contínuo dos processos e projetos, dos quais fazem parte no contexto das grandes corporações.

Tais transformações se tornaram objetos de estudo de anos, obtendo um embasamento teórico, pesquisas, procedimentos de condutas, e demais fatores que os projetaram atualmente através de suas aplicações e práticas tais como as normas da ISO, os modelos da CMM, a trilogia de Juran, os modelos RUP entre outros os quais são vastamente consagrados.

As técnicas de qualidade são ferramentas importantes para o sucesso dos projetos em TI, no entanto as pessoas chaves devem estar integradas com o desenvolvimento de suas tarefas e devidamente orientadas pelo seu gerente de projetos. Este intuito deve ser estrategicamente planejado afim de se obter uma exatidão no escopo e demanda das necessidade a serem supridas pelo conjunto de esforços realizado por todas as pessoas envolvidas no projeto.

O PMBOK é uma significativa conquista para a gerência de projetos, no entanto cada projeto segue sua peculiaridade, sendo único em sua concepção, construção e conclusão. A área de tecnologia da informação não é exceção, as práticas e adoções por uma conduta planejada e estruturada dentro das organizações, é de extrema valia para o sucesso dos projetos por ela realizados.

O significativo empreendimento do projeto, observando suas reais necessidades com um escopo coerente e aderente aos requisitos em todas as suas fases de execução, resulta em uma equipe orientada e devidamente focada em proporcionar os resultados almejados pelo projeto.

O planejamento estratégico se torna um forte aliado aos projetos de TI, quando se faz o uso correto e devidamente inserido no contexto de trabalho, de adoções como o acompanhamento do ciclo de vida do projeto em software, a tomada de decisões devidamente fundamentadas, o planejamento de um escopo plausível, e a utilização de padrões de qualidade tais como ISO, CMM, RUP e entre outras que colaboram a agregar valores de excelência para o sucesso na conclusão dos projetos.

O sucesso do gerenciamento de projetos, além da integração entre pessoas gerentes, departamentos está associado à habilidade em que o gerente de projetos, equipe e demais envolvidos tem em conduzir as tarefas aliadas a procedimentos e políticas de qualidade de uma forma padronizada, clara e objetiva para se alcançar as metas estipuladas pela instituição.

Este instrumento demonstrou de forma clara, objetiva e fundamental a importância da definição de escopo aliado ao gerenciamento de práticas e metodologias voltadas a qualidade de projetos, com ênfase na sua aplicação prática como forma de excelência na condução e conclusão de projetos voltados a tecnologia da informação.

Referencias:

Bibliografia citada:

- Gerenciamento Estratégico e Administração por Projetos (00), Dalton Valeriano – Dv (00), pg. 07 a 12.
- Gerencia em Projetos, Dalton Valeriano (95) - Dv (95) pg. 18, 20 a 26.
- Engenharia de software, Roger Presmann Rp(95), pg. 14 a 17, 32 a 34, 59 a 64.
- Engenharia de software com CMM (98), Soeli Fiorini, Renan Martins Baptista, Sf (98), pg. 44 a 54.
- PMI.(02).PMBOK v 1.0 português, pg. 10 a 14, 27 a 32.
- Juran na liderança pela qualidade Jj (90), Juran Joseph M., pg.35 a 42.
- Gestão de Projetos, Victor Roldão Vr (04), pg. 07 a 10, 19.