Pontifícia Universidade Católica de São Paulo

Monografia de Conclusão de Curso MBIS

Master Business Information System

“Mapeamento entre PMBOK, CMM e RUP”

Maurício Nacib Pontuschka

Coordenadora Prof.ª Lavínia Napoleão

janeiro de 2003

Agradecimentos

*À **Professora Lavínia Napoleão**, minha coordenadora, que com seus conhecimentos e habilidades me conduziram, incentivaram e motivaram tornando possível a realização desta monografia.*

*Ao **Professor José Carlos Arruda Alves** pelas idéias, incentivo e confiança no meu trabalho.*

*Aos meus pais **Walter Maigon Pontuschka** e **Nídia Nacib Pontuschka** pelo eterno apoio e carinho que nem sempre consigo retribuir.*

Índice

1 Introdução

A atividade de desenvolver software vem sofrendo alterações em seu comportamento durante todos os anos de sua existência. Desde a configuração das chaves nos primeiros computadores até a disponibilização de sistemas via Internet muitos enfoques foram dados à esta atividade. Enfoques estes que partiram de uma simples codificação de chaves para se obter o resultado de processos muito simples e progrediram até a atual necessidade de integração com sistemas legados, garantia de qualidade no desenvolvimento, garantia da continuidade do sistema em futuras manutenções, portabilidade, acessibilidade, segurança entre outros fatores que se fazem indispensáveis em sistemas atuais.

Com a evolução dos computadores, a capacidade de processamento, memória e armazenamento aumentaram muito. Isto permitiu que os softwares pudessem processar mais e mais rápido, permitiu a elaboração de sistemas mais complexos e abrangentes.

Atualmente os programas possuem dimensões gigantescas se comparados com programas construídos a poucos anos atrás. Estes programas estão cada vez mais exigindo um planejamento melhor, um controle melhor da condução em seu desenvolvimento. E para isso várias metodologias de desenvolvimento apareceram, o paradoxo de objetos surgiu como uma melhor ferramenta para se entender e modelar software, surgiram metodologias criando formas mais adequadas para o desenvolvimento com base em experiências passadas.

Essa complexidade se transformou em dificuldades que por vezes eram suficientes para inviabilizar a construção do software. Infelizmente isto ocorreu em muitas empresas e ainda ocorre em muitos projetos. São interrompidos devido à estimativas mal realizadas, má utilização de recursos, má alocação de pessoal ou outros tantos motivos que por vezes vinham ocorrer durante o processo de desenvolvimento.

Outro fator importante é que os computadores da atualidade possuem muito mais recursos que os de antigamente, com isso são capazes de processar muito mais informações e executarem softwares muito mais complexos e maiores em tamanho. Dificilmente um software de grandes proporções é construído por apenas uma pessoa devido ao tempo de desenvolvimento, ou até devido a áreas de especialização que vem surgindo e cada vez mais um profissional de especializa em desenvolver trechos específicos do programa como é facilmente observado através dos profissionais e cargos das áreas da computação como: DBAs (Administradores de banco de dados); Integradores de sistemas; Especialistas em Sistemas Operacionais; entre outras tantas especialidades.

Dado este panorama, o desenvolvimento de software foi obrigado a criar formas organizadas de se trabalhar em equipe. Grupos de trabalho que possuem especialistas desenvolvendo as partes do software de seu conhecimento mas de forma integrada procurando diminuir problemas com o sincronismo das tarefas e eventuais problemas de integração das partes.

As equipes de desenvolvimento tomaram conhecimento da necessidade de planejar melhor e de efetivamente criar um processo de desenvolvimento bem definido e eficaz para o desenvolvimento de sistemas.

As novas armas dos desenvolvedores para conseguir com que os projetos caminhassem na direção do sucesso foram:

- Metodologias de Desenvolvimento de Software;
- Certificações de Níveis de Maturidade das Empresas;
- Metodologias de Gestão de Projetos.

As **Metodologias de Desenvolvimento de Software**, fornecem métodos, atividades, papéis e responsabilidades imprescindíveis para a construção de um software. Controlam as fases de desenvolvimento, as aprovações necessárias, toda a documentação a ser gerada e outros aspectos muito importantes para o bom desenvolvimento do sistema.

As **Certificações de Níveis de Maturidade das Empresas** permitem que as empresas sejam avaliadas periodicamente obtendo e publicando o seu nível de maturidade no desenvolvimento de software. Este nível é utilizado para que outras empresas saibam que determinadas práticas são efetivamente utilizadas na empresa em questão e que o desenvolvimento que será realizado poderá ser acompanhado, mensurado, estimado e provavelmente possuirá um nível de qualidade alto.

Quanto às **Metodologias de Gestão de Projetos**, estas permitem o controle completo desde a elaboração até a entrega de um projeto. Esta é uma habilidade que profissionais podem se certificar obtendo uma certificação PMP (Project Manager Professional) do PMI (Project Management Institute). Esta matéria trata de práticas consagradas pela experiência de muitos profissionais da área de gerência de projetos, todas registradas em uma publicação PMBOK (Project Management Body of Knowledge).

Este trabalho concentra-se na utilização em conjunto destas áreas o que servirá de subsídios para a elaboração de um documento único da empresa com definições dos processos, métodos, técnicas e padrões utilizados pela empresa no seu processo de desenvolvimento de software.

Muitas das características presentes em cada uma destas áreas são similares, o que permite que seja realizado um mapeamento de forma que um mesmo conjunto de características e práticas seja utilizado para obtenção de uma certificação CMM (Capability Maturity Model) da empresa e também para que o gerente do projeto que necessita de projetos que sigam as recomendações PMBOK para a obtenção da sua certificação PMP.

O Project Management Institute (PMI) é uma associação sem fins lucrativos fundada em 1969 e se tornou a organização escolhida por pessoas que trabalham ou são interessadas em gestão de projetos.

Esta organização possui três principais focos:

- definição de padrões;
- certificações;
- pesquisa.

O PMI desenvolve padrões para o ofício de gestão de projetos. O principal documento desenvolvido, mantido e atualizado pelo PMI com esta finalidade é o Project Management Body of Knowledge – PMBOK. O PMBOK é uma norma que descreve uma base de conhecimento do ofício de gerenciamento de projetos obtido de profissionais práticos e acadêmicos a partir das melhores práticas amplamente utilizadas.

Desde 1984 o PMI esteve dedicado em desenvolver e manter um programa rigoroso de certificação para reconhecer os níveis de competência alcançados por profissionais desta área. A certificação Project Management Professional – PMP, é a credencial de maior reconhecimento no âmbito desta profissão. Em 1999 o departamento de Programas de Certificação do PMI foi o primeiro a obter o reconhecimento ISO 9001.

O PMI realiza conferências internacionais bienais de pesquisa em gestão de projetos. Nestas conferências são realizadas pesquisas externas, é alimentado um banco de dados de pesquisas do PMI e são identificados novos tópicos de pesquisas a serem desenvolvidos. Estes novos conhecimentos coletados ou construídos são disseminados para toda a comunidade através de suas publicações.

O Capability Maturity Model (CMM) é um modelo de maturidade para desenvolver software. É um dos mais conhecidos produtos do Instituto de engenharia de Software (software engeneering Institute – SEI) da universidade Carnegie Mellon. O CMM representa a combinação do julgamento de centenas de engenheiros com muita experiência acumulada durante as suas carreiras profissionais.

Além do conjunto dos conhecimentos e melhores práticas, o CMM procura realmente fornecer uma classificação da empresa junto ao mercado a fim de permitir que os relacionamentos entre empresas possam se basear em algumas garantias quanto à forma que a empresa trabalha.

Por exemplo, se conhecermos uma empresa com CMM nível 2 que já realizou trabalhos de desenvolvimento com bons resultados em uma determinada área e em um determinado tipo de projeto, esta empresa deverá conseguir repetir o sucesso anteriormente obtido em novos projetos similares.

São cinco os níveis de certificação do CMM:

Nível 1: (qualquer empresa sem obtenção de um nível de maturidade).

Nível 2: repetível.

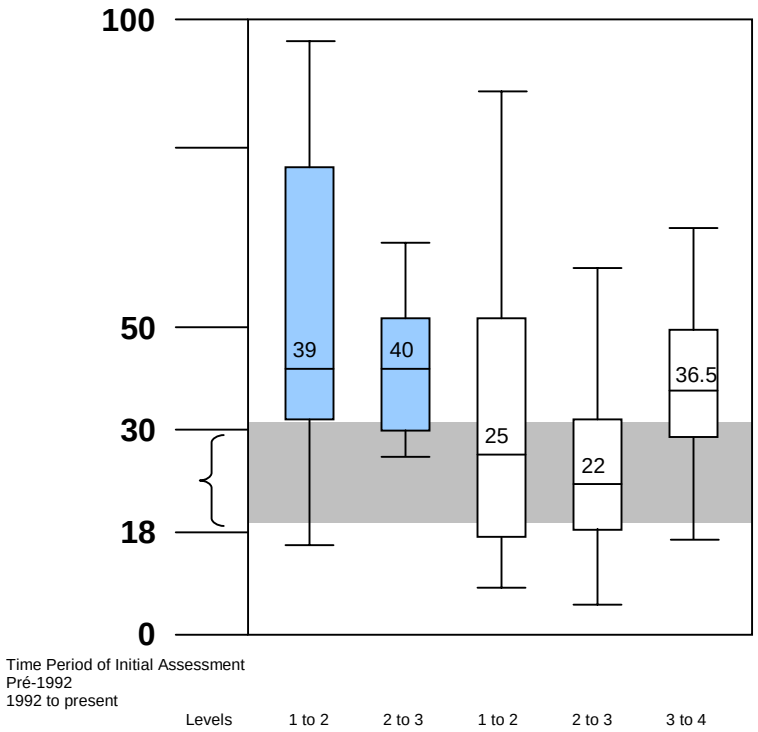
Nível 3: definido.

Nível 4: mensurável.

Nível 5: otimizado.

Toda esta experiência foi traduzida em um documento: CMM vers. 1.1. Este documento descreve o CMM que é dividido em 5 níveis de maturidade e cada nível descreve um conjunto de áreas de conhecimento. Cada nível de maturidade deve atender as áreas, para ele, descritas e também as áreas dos níveis inferiores.

A empresa se prepara para alcançar um determinado nível e então agenda uma verificação através de entidades credenciadas pela SEI para fornecer esta certificação. Estas verificações também podem identificar retrocessos no nível de maturidade. Na tentativa de alcançar maiores níveis a empresa deve estar preocupada em manter os níveis anteriormente atingidos, caso estes não estejam mais cobertos pelas práticas da empresa, o nível geral da empresa será reduzido. Para se ter uma idéia do número de certificações de empresas na classificação CMM, a seguir encontra-se um gráfico estatístico das certificações até o ano 2000.



A terceira área a ser integrada é representada pelas Metodologias de Desenvolvimento de Software. Existem várias metodologias que são utilizadas pelas mais variadas equipes de desenvolvimento de software.

O importante é que estas metodologias de desenvolvimento fornecem o aspecto operacional do desenvolvimento. A maioria destas metodologias fornece recursos informatizados para automatização dos processos de desenvolvimento. São produtos de software que ajudam na condução do processo de desenvolvimento.

É impossível ignorar o trabalho dos metodologistas Ivar Jacobson, Grandy Booch e James Rumbaugh que criaram a Unified Modelling Language – UML. Este trabalho levou em conta todas as mais utilizadas metodologias com o objetivo de criar uma forma única de representação dos modelos de classes, requisitos, seqüências, colaborações e outros elementos do processo de desenvolvimento de software. Esta linguagem já é um padrão e é amplamente utilizada por toda a comunidade de software.

Como a UML é uma linguagem e não uma metodologia, ela pode ser utilizada por vários profissionais nas mais variadas metodologias.

Os três criadores da UML também propuseram uma metodologia: o “Unified Process”. Esta metodologia é uma das mais utilizadas na atualidade, mas não teve o mesmo impacto que obteve a UML. Isto aconteceu devido ao fato de que cada forma de trabalho definida nas mais variadas metodologias oferece benefícios específicos em diferentes situações. Dependendo da situação uma metodologia é melhor que outra e isto pode inverter caso o contexto seja diferente.

Portanto é muito difícil afirmar que uma metodologia é melhor ou pior que outra, é necessário estudar o contexto para poder realizar a escolha da metodologia.

Para efeitos de exemplificação, foi necessário escolher uma metodologia em particular para a realização do mapeamento que é a proposta deste trabalho. A metodologia escolhida foi o Rational Unified Process em homenagem aos criadores da UML os quais também criaram o Ration Unified Process (RUP). Isto não quer dizer que esta metodologia seja melhor ou pior que qualquer outra. A escolha foi feita simplesmente para poder exemplificar a forma de trabalho proposta unindo as três áreas: Gestão de Projetos; Certificação de Maturidade e Metodologias de Desenvolvimento de Software.

Nos capítulos seguintes são estudados os três elementos do mapeamento que são: ***Project Mamagement Body of Knowledge***; ***Capability Maturity Model*** e ***Rational Unified Process***. Cada um destes será explicado para que seja possível a realização do mapeamento.

É importante salientar que o PMBOK é muito mais abrangente que os outros dois elementos. O PMBOK se propõe a gerenciar qualquer tipo de projeto. Podendo variar desde a realização de um pequeno evento até a construção de navios. O contexto deste trabalho é o desenvolvimento de software, portanto o mapeamento realizado irá apenas incluir elementos no que tange a projetos de software. Tanto o CMM quanto o RUP já se enquadram no contexto de software, portanto serão levados em conta de uma forma total.

2 Estudo do PMBOK (Project Management Body Of Knowledge) do PMI (Project Management Institute)

2.1 PMBOK

Para realizar o mapeamento entre PMI, CMM e RUP primeiramente realizarei uma descrição de cada um deles.

Como já foi dito o Project Management Institute é uma organização que se destina a organizar e manter uma base de conhecimento a respeito das melhores práticas no âmbito da atividade de gestão de conhecimento.

Este trabalho utilizará um produto desta organização, o PMBOK (Project Management Body Of Knowledge).

Algumas definições se fazem necessário para que não haja uma dupla interpretação de termos que utilizaremos no decorrer deste texto.

2.1.1 Projeto

As organizações realizam trabalhos, e estes trabalhos podem ser organizados de duas formas: por processos ou por projetos.

Tanto os processos quantos os projetos, são realizados por pessoas, restritos por limites de recursos, planejados, executados e controlados.

Os projetos normalmente representam formas de atingir objetivos descritos em planos estratégicos da empresa.

A principal diferença entre processos e projetos é que processos representam atividades contínuas e repetitivas dentro da empresa, enquanto os projetos são temporários e únicos.

Ao dizer que os projetos são temporários, entenda que os projetos possuem um começo, um meio e um fim muito bem determinados. O projeto é finalizado quando os objetivos estipulados para o projeto forem atingidos. Portanto estes objetivos devem ser tangíveis e mensuráveis.

O fato de que um projeto seja temporário, não implica que os produtos por ele gerados também o sejam. Projetos podem gerar produtos como: edifícios, máquinas, livros e outros mais, podendo ter uma durabilidade muito grande. Eventualmente, dependendo da situação, poderá haver a necessidade de se instalar “processos” de manutenção dos mesmos. Estas manutenções são classificadas como processos porque são atividades contínuas e repetitivas.

Outra característica de projetos é que eles possuem resultados únicos. Mesmo que uma empresa construa uma série de edifícios, cada um possui suas próprias características nos mais variados níveis, desde as empresas terceirizadas que participaram da construção até o tipo dos materiais utilizados. Por tanto estes trabalhos devem ser tratados como projetos e não como processos.

Os projetos possuem uma elaboração progressiva, isto é, ele é realizado em etapas evolutivas. Um dos grandes riscos na elaboração de projetos é o de nunca conseguir chegar ao final do projeto, ou seja, nunca atingir os objetivos

previamente estipulados devido a não se ter deixado claras as condições de término, ou o escopo do projeto não estar adequadamente fechado.

2.1.2 Gestão de Projetos

Gestão de projetos é a aplicação de conhecimento, padrões, ferramentas e técnicas para que as atividades de um projeto alcancem os objetivos do projeto. A gestão de projetos é realizada através da utilização de processos como: inicialização, planejamento, controle, execução e fechamento. Trata o universo gerencial produtivo.

2.1.3 Processo iterativo

O Processo iterativo para a realização de um projeto permite que a cada etapa do projeto (iteração), a equipe do projeto consiga gradativamente conhecer melhor o problema, uma vez que este está sendo estudado a cada iteração de uma abordagem ou ângulo de visão diferente.

2.1.4 Gerenciamento por Projetos X Gerenciamento por Processos

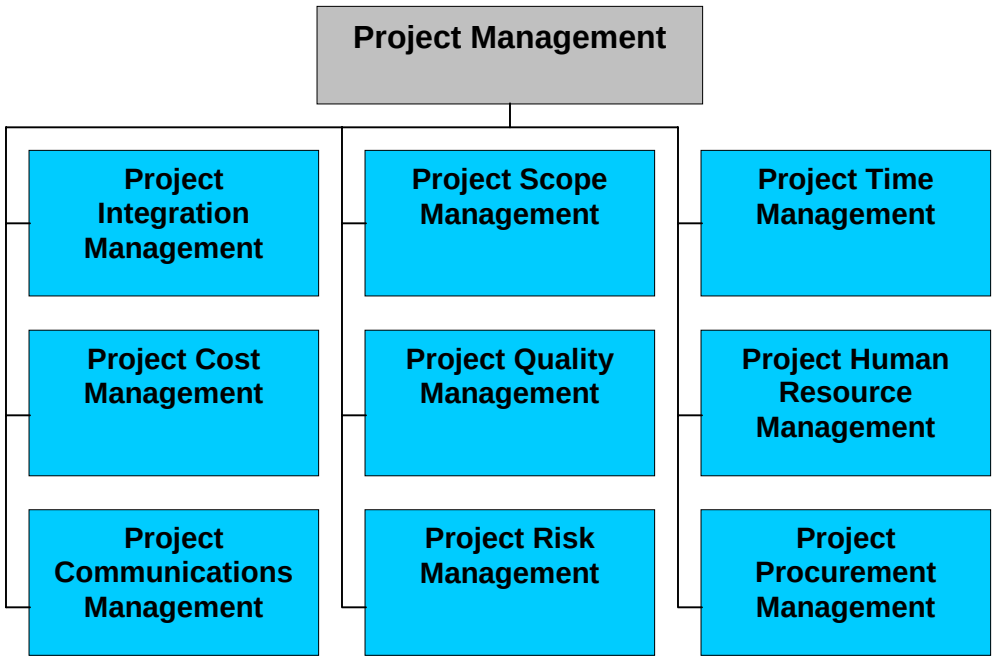
Existem duas formas de se gerenciar empresas, por projetos ou por processos. O gerenciamento por processos é caracterizado por existirem normas e procedimentos pré-estabelecidos e estas são aplicadas e reaplicadas para a execução das suas atividades de uma forma repetitiva e contínua. A gestão de projetos é caracterizada por encarar cada atividade como um projeto distinto com suas próprias características e em cada situação é elaborado um planejamento e este é seguido até a sua conclusão. Não existe uma forma certa ou uma forma

errada, apenas é importante que a empresa tenha consciência da sua natureza e atue de acordo.

2.1.5 Estrutura do PMBOK

Como já foi citado, o PMBOK procura registrar o conhecimento a respeito das práticas na área de gestão de projetos.

Este conhecimento está dividido em nove grandes áreas:



Project Integration Management: Descreve os processos necessários para garantir que os vários elementos do projeto estejam adequadamente coordenados.

Project Scope Management: Descreve os processos necessários para garantir que o projeto inclua todo o trabalho necessário, e somente o trabalho necessário para obter sucesso na conclusão do projeto.

Project Time Management: Descreve os processos necessários para garantir que o projeto seja concluído em tempo hábil

Project Cost Management: Descreve os processos necessários para garantir que o projeto seja concluído com o orçamento aprovado.

Project Quality Management: Descreve os processos necessários para garantir que o projeto irá satisfazer as necessidades que foram especificadas.

Project Human Resource Management: Descreve os processos necessários para realizar a utilização mais efetiva das pessoas envolvidas no projeto.

Project Communications Management: Descreve os processos necessários para garantir a geração, disseminação, coleção, armazenamento e disposição das informações do projeto em tempos adequados.

Project Risk Management: Descreve os processos centrados na a identificação, análise e controle dos riscos do projeto.

Project Procurement Management: Descreve os processos necessários para captar bens e serviços externos à organização.

2.2 Contexto de Gestão de Projetos

Os projetos e a gestão de projetos operam em um ambiente mais amplo que o projeto de software. A equipe do projeto deve entender este amplo contexto – gerenciar as tarefas pré-estabelecidas é necessário mas não suficiente. A fim de explorar melhor este contexto de realização do projeto os tópicos a seguir abordam questões de relacionamento do dia a dia do projeto com os seus relacionamentos externos ao projeto.

2.2.1 Fases do projeto e ciclo de vida do projeto

Por serem únicos, os projetos possuem um certo grau de incerteza. As empresas dividem o projeto em fases para que se possa haver um melhor controle de seu desenvolvimento.

Estas fases são criadas cada uma com objetivos previamente estabelecidos, e estes objetivos muitas vezes se repetem em vários projetos (como por exemplo uma fase de análise de requisitos presente em praticamente qualquer projeto de desenvolvimento de software).

Estas fases acabam sendo conhecidas e estudadas individualmente. É necessário um conjunto destas fases que são colocadas de forma coordenada para atender um projeto.

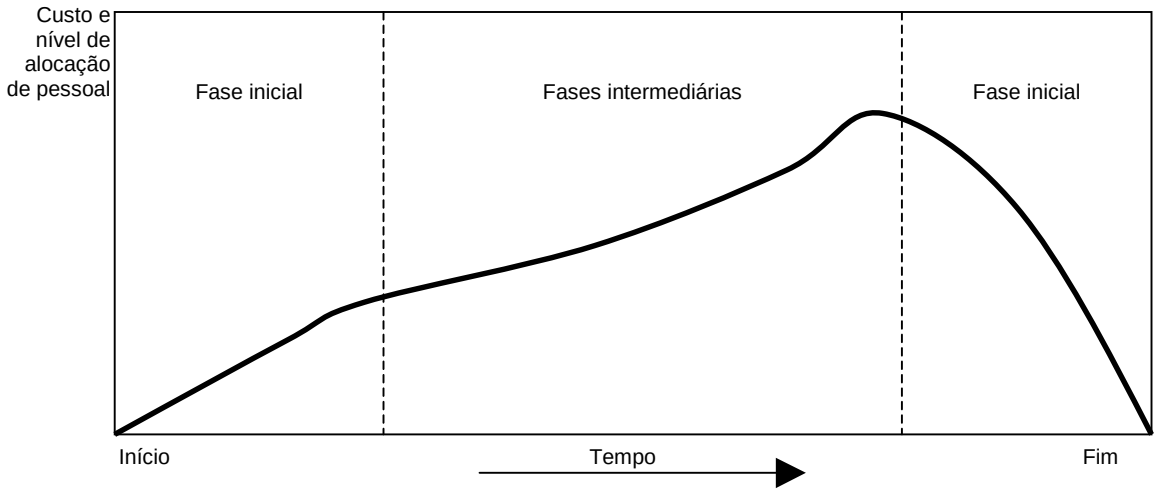
Não existe apenas uma possibilidade de arranjo destas fases para se obter sucesso na confecção de um projeto, existem muitas possibilidades. O conjunto destas fases é chamado de ciclo de vida do projeto.

Cada fase é marcada pela confecção de um ou mais produtos de trabalho. Um produto de trabalho deve ser tangível, verificável e normalmente ao final de cada fase há uma análise de performance de sua execução.

Com isso é possível determinar se o produto de software pode avançar para a próxima fase e também permite que erros sejam detectados de forma a não comprometer o projeto.

O ciclo de vida de um projeto normalmente determina as atividades técnicas que devem ser realizadas e também quem estará envolvido em cada fase.

A maioria das descrições de ciclo de vida dos projetos possuem algumas características similares. Possuem níveis de custo e alocação de pessoal baixos no seu início, alto ao se aproximar do final e cai rapidamente no que o projeto vai chegando a sua conclusão.



Sub-projetos podem possuir ciclos de vida distintos, dependendo da complexidade da fase, a implantação de um ciclo de vida para a sua realização oferecerá um

melhor controle e maiores garantias no que se diz respeito a cumprir custos e prazos previamente estipulados.

2.2.2 Stakeholders

Stakeholders são indivíduos ou organizações que estão ativamente envolvidos ao projeto ou que podem ser afetados positivamente ou negativamente pelo projeto ou pelos seus resultados.

Existem alguns stakeholders chave que normalmente estão presentes em todo projeto. Os stakeholders chave são:

- **Sponsor**: Indivíduo ou grupo interno ou externo a organização que provê recursos financeiros para o projeto.
- **Project Manager**: Indivíduo responsável por gerenciar o projeto;
- **Customer**: Indivíduo ou organização que irá utilizar os produtos gerados pelo projeto. Pode haver mais de uma camada de clientes. Por exemplo clientes de um novo produto farmacêutico podem ser: médicos que o prescrevem; pacientes que o utilizam e planos de saúde que pagam por ele. Dependendo do projeto, clientes e usuários são tratados como sinônimos, mas podem não ser. Pode haver situações em que clientes são os indivíduos ou organizações que realizam a compra do produto e usuários são os que efetivamente o utilizam;
- **Performance Organization**: Organização que mantém a maioria dos envolvidos diretamente ao projeto na realização de suas atividades ;

•**Project Team**: Grupo que realiza as atividades do projeto;

2.2.3 Influências Organizacionais

Os projetos estão sempre vinculados à uma organização a qual o abriga e proporciona a sua realização. Portanto este projeto estará sob forte influência desta organização. A maturidade desta organização a respeito dos sistemas de gestão de projetos, cultura, estilo, estrutura organizacional e o departamento de gestão de projetos, todos estes elementos podem influenciar o projeto.

A tabela abaixo apresenta dados a respeito dos níveis de influência esperados para diferentes características de empresas e projetos.

Estrutura organizacional / **Características do projeto**	**Funcional**	**Matricial**			**Projetada**
		Fraca	**Balanceada**	**Forte**	
Autoridade do gerente do projeto	Pouca ou nenhuma	Limitada	Pouca a moderada	Moderada a alta	Alta a quase total
Porcentagem do pessoal da empresa designados em período integral ao projeto	Virtualmente nenhuma	0-25%	15-60%	50-95%	85-100%
Papel do gerente de projeto	Meio período	Meio período	Período integral	Período integral	Período integral
Cargo comum para o papel de gerente do projeto	Coordenador de projeto / Líder de projeto	Coordenador de projeto / Líder de projeto	Gerente de projeto / Escritório de projetos	Gerente de projeto / Gerente de Programa	Gerente de projeto / Gerente de programa
Pessoal administrativo da gerência de projetos	Meio período	Meio período	Meio período	Período integral	Período Integral

2.2.4 Habilidades-chave de gestão

Gerenciamento em geral é um assunto muito amplo que negocia com cada aspecto de gerenciamento no ambiente em produção. Existem muitos tipos de

projeto e cada um com suas próprias habilidades-chave de gerenciamento, mas de uma maneira geral, a maioria dos projetos necessitam de habilidades comuns que são:

- Liderança
- Comunicação
- Negociação
- Resolução de problemas
- Influenciando a organização

2.3 Processos da Gestão de Projetos

2.3.1 Processos dos projetos

Um projeto é composto por processos, e cada processo é uma série de ações as quais produzem um resultado. Os processos de gestão de projetos são divididos em cinco grupos:

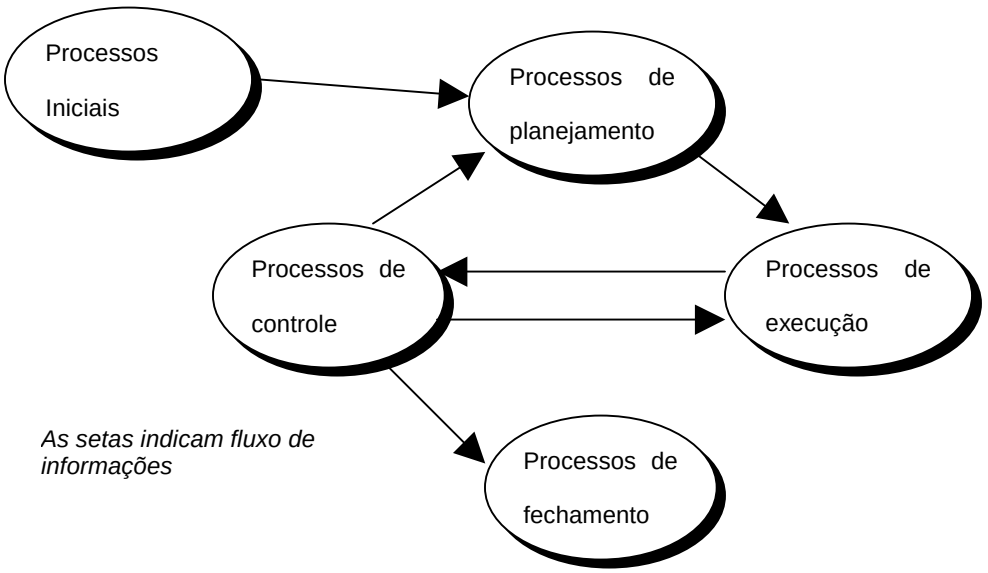
- **Processos de inicialização** ("*Initiating processes*" – autorização do projeto ou fase);
- **Processos de planejamento** ("*Planning processes*" – definição e refinamento de objetivos e seleção do melhor dos cursos alternativos de ações a fim de se atingir os objetivos que o projeto procura encontrar);
- **Processos de execução** ("*Executing processes*" – coordenação de pessoal e outros recursos para executar o plano);

- **Processos de controle** ("*Controlling processes*" – garantir que os objetivos do projeto sejam atingidos através de monitoramento e medições da regularidade do progresso identificando variâncias em relação ao plano ou fase realizando ações corretivas quando necessárias);

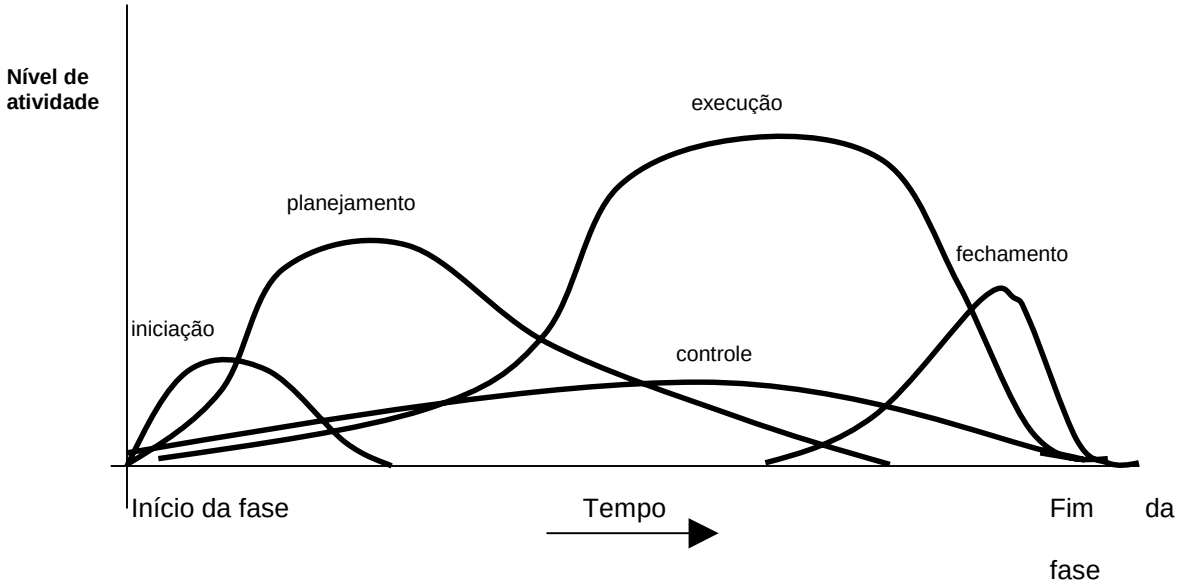
- **Processos de fechamento** ("*Closing processes*" – formalização do aceite do projeto ou fase).

2.3.2 Grupos dos processos

Os grupos de projeto estão ligados através dos produtos de software que produzem. Geralmente o resultado de um grupo alimenta o outro.

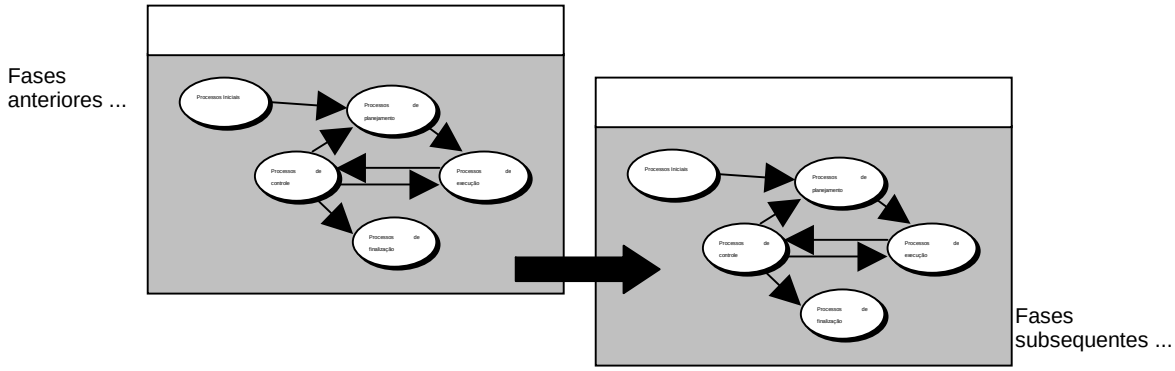


Além disso os grupos não são discretos, eles possuem sobreposições de suas atividades que ocorrem nos diversos níveis de intensidade em cada fase do projeto.



E finalmente, as interações entre grupos através das fases onde o resultado do fechamento de um é a entrada da inicialização do outro.

Grupos do processo Áreas de conhecimento	Inicialização	Planejamento	Execução	Controle	Fechamento
Integração		•Desenvolvimento do plano de projeto	•Execução do plano do projeto	•Controle de mudanças integrado	
Escopo	•Inicialização	•Planejamento de escopo •Definição de escopo		•Verificação de escopo •Controle de mudança de escopo	
Tempo		•Definição de atividade •Seqüencialização de atividades •Estimativas de duração das atividades •Desenvolvimento do cronograma		•Controle do cronograma	
Custo		•Planejamento de recursos •Estimativas de custo •Orçamento dos custos		•Controle do custo	
Qualidade		•Planejamento de qualidade	•Garantia de qualidade	•Controle de qualidade	
Recursos Humanos		•Planejamento organizacional •Recrutamento de pessoal	•Desenvolvimento da equipe		
Comunicações		•Planejamento de comunicações	•Distribuição de informações	•Relatórios de performance	•Fechamento administrativo
Risco		•Planejamento de gerência de riscos •Identificação dos riscos •Análise qualitativa dos riscos •Análise quantitativa dos riscos •Planejamento de responsabilidades pelos riscos		•Controle e monitoramento de riscos	
Terceirizações		•Planejamento de terceirizações •Planejamento de licitações	•Licitações •Seleção de fornecedores •Administração de contratos		•Fechamento de contratos



2.3.3 Interações dos processos

Como já foi dito um processo pode ser dividido em cinco grupos (inicialização, planejamento, controle, execução e fechamento). Para o caso de gestão de projetos, existem estas cinco grandes áreas, cada uma com as suas sub-divisões já incluindo as melhores práticas para melhor atendê-las. Este mapeamento está relacionado a seguir.

2.3.4 Inicialização

- **Inicialização**: a autorização do projeto ou de uma fase é parte do gerenciamento de escopo do projeto.

2.3.5 Planejamento

- **Desenvolvimento do plano de projeto**: Tomar os resultados de outros processos de planejamentos e colocando-os dentro de um documento coerente e consistente.

- **Planejamento de escopo**: Desenvolvimento de um documento escrito de escopo como base para futuras decisões de projeto.

- **Definição de escopo**: Subdivisão dos principais produtos de trabalho, a serem gerados pelo projeto, em componentes menores e mais fáceis de de gerenciar.

- **Definição de atividade**: Identificar atividades específicas que devem ser realizadas para produzir os vários produtos de trabalho do projeto.

- **Seqüencialização de atividades**: Identificar e documentar interações e dependências entre as atividades do projeto.

- **Estimativa de duração das atividades**: Estimativas de número de períodos de trabalho necessários para completar atividades individualmente.

- **Desenvolvimento do cronograma**: analisar as seqüências de atividades , duração das atividades e necessidades de recursos para criar o cronograma do projeto.

- **Planejamento da gerência de riscos**: Decidir como abordar e como se planejar para a gerência de riscos do projeto.

- **Planejamento de recursos**: Determinar quais recursos (pessoas, equipamentos, materiais, etc) e quais quantidades de cada um deverá ser utilizada para a execução das atividades do projeto.

- **Estimativa de custos**: Desenvolver uma aproximação (estimativa) dos custos os recursos necessários para completar as atividades do projeto.

- **Orçamento de custos**: estabelecer a estimativa completa de custos para os produtos individuais de trabalho.

- **Planejamento de qualidade**: Identificar quais os padrões de qualidade que são relevantes para o projeto e determinar como satisfazê-los.

- **Planejamento organizacional**: Identificar, documentar, atribuir papéis, responsabilidades e reportar relacionamentos.

- **Recrutamento de pessoal**: recrutar recursos humanos necessários associados ao projeto.

- **Planejamento de comunicações**: Determinar as informações e comunicações necessárias com os stakeholders quem necessita da informação, quando irão precisar e como serão entregues.

- **Identificação de riscos**: Determinar quais os riscos que podem afetar o projeto e documentar as características de cada uma.

- **Análise qualitativa dos riscos**: Realizar uma análise qualitativa dos riscos e condições para priorizar os efeitos aos objetivos do projeto.

- **Análise quantitativa dos riscos**: Medir a probabilidade e impacto dos riscos e estimar as implicações para com os objetivos do projeto.

- **Planejamento de responsabilidade pelos riscos**: Desenvolver procedimentos e técnicas para elevar as oportunidades e reduzir as ameaças dos riscos para com os objetivos do projeto.

- **Planejamento de terceirizações**: Determinar o que licitar, quanto e quando.

- **Planejamento de licitações**: Documentar as necessidades de produtos e identificar as fontes potenciais.

2.3.6 Execução

- **Execução do plano de projeto**: Realizar o plano do projeto através da execução das suas atividades.

- **Garantia de qualidade**: Avaliar a performance geral em bases regulares a fim de assegurar que o projeto irá satisfazer os padrões de qualidade relevantes.

- **Desenvolvimento de equipe**: Desenvolver competências em caráter individual e em grupo a fim de aumentar a performance do projeto.

- **Distribuição de informações**: Disponibilizar as informações necessárias para os stakeholders do projeto de uma maneira atualizada e periódica.

- **Licitações**: Realizar cotações, leilões, orçamentos ou propostas quando apropriado.

- **Seleção de Fornecedores**: Escolha dos potencias fornecedores.

- **Administração de contratos**: Gerenciamento dos relacionamentos com os fornecedores e vendedores.

2.3.7 Controle

- **Controle de mudanças integrado**: Coordenar as mudanças ocorridas durante todo o projeto.

- **Verificação de escopo**: Formalização do aceite do escopo do projeto.

- **Controle de alteração de escopo**: controlando de mudanças no escopo do projeto.

- **Controle do cronograma**: Controle de mudanças no cronograma do projeto.

- **Controle do custo**: Controlando mudanças no orçamento do projeto.

- **Controle de qualidade**: Monitoramento de resultados específicos do projeto a fim de determinar se eles estão de acordo com padrões relevantes de qualidade e identificar formas de eliminar causas de performances insatisfatórias.

- **Relatórios de performance**: Coleta e disseminação de informações de performance. Incluindo relatórios de estado do projeto, medidas de acompanhamento e previsões.

- **Controle e monitoração de riscos**: acompanhar os riscos identificados, monitorando riscos residuais e identificando novos, garantir a execução do planejamento de riscos e avaliar a sua eficiência reduzindo riscos.

2.3.8 Fechamento

- **Fechamento administrativo**: geração, reunião e disseminação de informações a fim de formalizar uma fase ou o projeto, incluindo avaliar o projeto, reunindo lições aprendidas para uso em futuros projetos ou fases.

- **Fechamento de contratos**: Finalização e pagamento dos contratos, incluindo a resolução de quaisquer itens em aberto.

3 CMM – Capability Maturity Model

Como foi dito anteriormente o CMM fornece um recurso de aferição da maturidade das empresas desenvolvedoras de software.

A satisfação dos clientes se tornou o ponto principal na competitividade entre empresas. A qualidade do software final bem como a qualidade do processo que o confeccionou hoje é verificada por parte dos clientes a fim de escolher seus fornecedores.

Muito tem se discutido, nos últimos anos, a respeito de qualidade de software e muitas vezes ações foram tomadas sem sucesso. As ações que conseguiram atingir um certo grau de sucesso foram ações baseadas na qualidade do processo e não somente do produto final.

Foi com base nestas informações que o CMM reuniu as melhores práticas do mercado sob o ponto de vista do processo de desenvolvimento de software. Estas práticas são verificadas e quando presentes na empresa, indicam que a mesma possui um certo grau de maturidade em seu processo de desenvolvimento.

Estas práticas foram divididas em 4 grandes níveis e cada conjunto de um determinado grau, abrange todas as práticas dos grupos de graus inferiores.

Instalar um processo contínuo se baseia em passar por muitos passos pequenos e evolutivos ao contrário de revoluções inovadoras. O CMM fornece um roteiro de trabalho para organizar estes passos evolutivos em cinco níveis de maturidade

que fornecem os fundamentos de uma forma sucessiva para a implantação de um processo contínuo.

Cada nível de maturidade abrange um conjunto de objetivos de processo que, quando satisfeitos, estabilizam um componentes importante do processo de software. Alcançar cada nível de maturidade através deste roteiro de trabalho é estabelecer um novo componente na maneira de trabalhar o que resulta no aumento da capacidade do processo de desenvolvimento de software da empresa.

Na realidade são definidos 5 níveis de maturidade.

O nível 1: **Nível inicial**

O processo de software é caracterizado por não possuir definição de processo. Ocasionalmente até caótico. O sucesso depende do heroísmo e no esforço individual de uma ou um grupo pequeno de pessoas.

A organização não possui um ambiente estável de desenvolvimento e manutenção de software. Sobrecarga de trabalho é uma característica deste nível e normalmente empresas deste nível possuem dificuldades de se comprometerem com projetos quanto a escopo, custo e prazos.

Dificilmente encontramos uma documentação adequada do produto, o cliente não consegue acompanhar o desenvolvimento, não há como garantir que o produto estará pronto no prazo e na maioria das situações o sistema está na memória do principal analista-programador o qual se torna indispensável para a manutenção do software. Esta dependência não é desejada pois este analista poderá se

desligar da empresa por qualquer que seja o motivo e então o software não terá mais como ser mantido.

Os softwares neste nível de maturidade possuem a característica de serem feitos por indivíduos e não por equipes.

O nível 2: **Nível Repetível**

Processos básicos de gerenciamento de projetos são estabelecidos para traçar custo, cronograma e escopo. A disciplina necessária é a de conseguir repetir sucessos anteriores em projetos de aplicações similares. Políticas para o gerenciamento do projeto de software e procedimentos para implementação destas políticas são estabelecidas. Planejar e gerenciar novos projetos são baseados em outros projetos similares já realizados.

O nível 3: **Definido**

O processo de desenvolvimento de software para gerenciamento e engenharia das atividades é documentado padronizado e integrado em um processo de desenvolvimento de software padrão da empresa.

O nível 4: **Gerenciado**

Medições detalhadas do processo de software e da qualidade do produto são realizadas. Tanto o processo de desenvolvimento de software quanto os produtos são quantitativamente entendidos controlados.

O nível 5: **Otimizado**

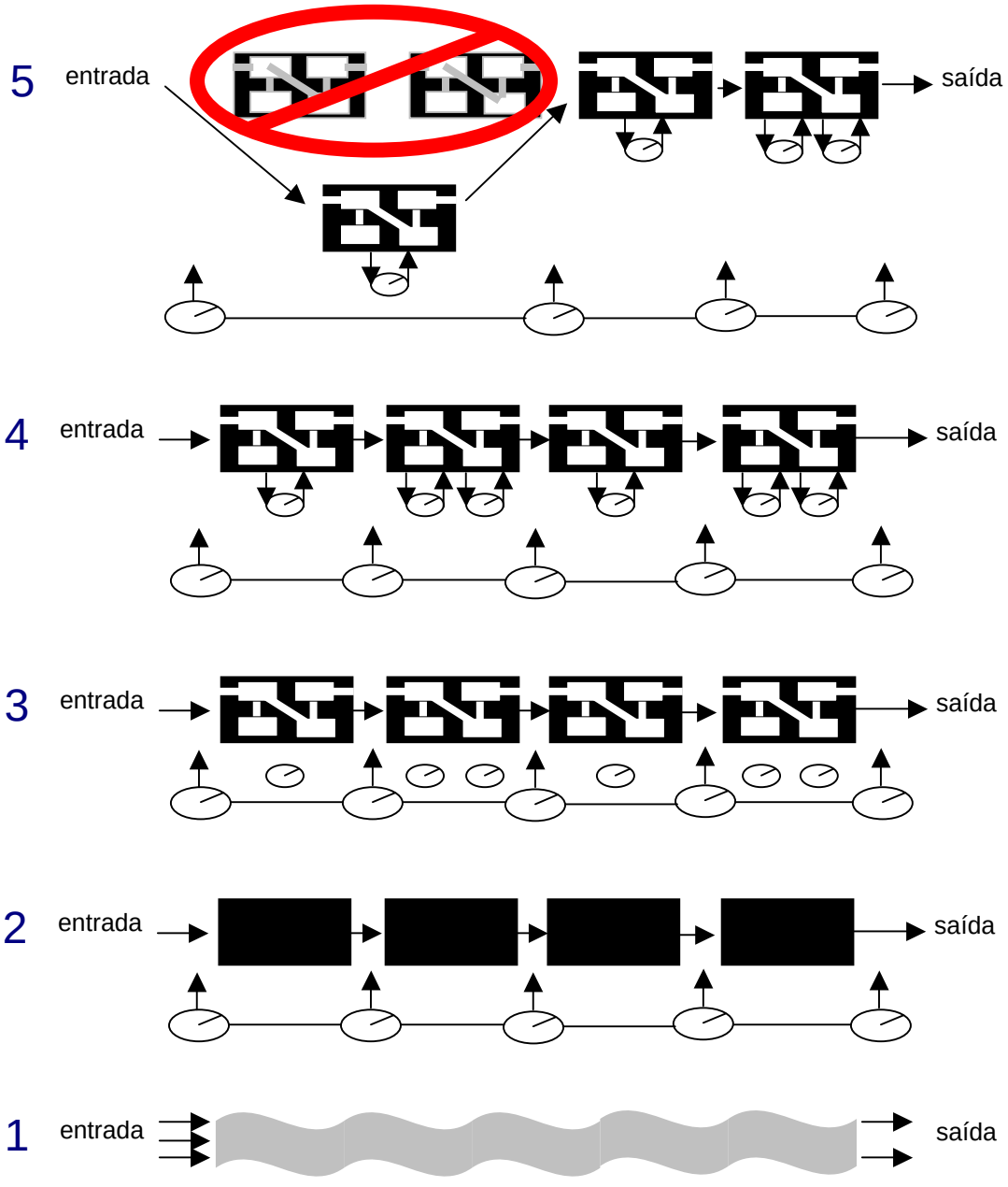
A evolução do processo contínuo é possibilitada através de retornos quantitativos de processos e da aplicação de novas idéias e tecnologias. O processo de desenvolvimento estará sendo aprimorado e revisado a cada projeto que é realizado. Isto permite que o processo acompanhe tendências de mercado e esteja sempre atual, impedindo que a empresa fique amarrada a um processo antiquado de desenvolvimento e comece a perder produtividade em relação a outras empresas.

Estes cinco níveis refletem o fato de que o CMM é um modelo para aumentar a capacidade do processo de desenvolvimento de software das organizações.

3.1 Visibilidade do Processo de Desenvolvimento de Software

A cada nível de maturidade do CMM a visibilidade do processo de desenvolvimento de software, tanto no aspecto gerencial quanto no da engenharia, é melhorada. No nível 1 o processo de software é uma grande incógnita que inicia-se no pedido do software e conclui na entrega do mesmo. Já no nível 2 o processo de software é visto como uma seqüência de “caixas pretas” cada uma representando uma etapa do processo estas etapas controladas e acompanhadas pelo gerente do projeto. No nível 3 a estrutura das caixas pretas são visíveis. Esta estrutura interna representa a forma como o processo de software padrão da organização foi aplicado em projetos específicos. No nível 4 o processo de software definido é instrumentalizado e controlado quantitativamente. Gerentes podem medir o progresso e verificar problemas. No nível 5 novas formas

de construir software são experimentadas continuamente de uma forma controlada a fim de aumentar a produtividade e qualidade.



3.2 Key Process Areas

O CMM escolheu um conjunto de processos-chave para serem tratados no desenvolvimento de software.

Com exceção do nível de maturidade 1, todos os outros quatro estão divididos em processos-chave. Cada conjunto de processos de um determinado nível somados aos processos dos níveis inferiores é responsável por atingir os objetivos do nível em questão. Por exemplo, para se atingir os objetivos de maturidade 4, é necessário abranger os processos dos níveis 1, 2, 3 e 4.

Os processos-chave de cada nível são:

Nível 1 - inicial (0 processos):

Nenhum

Nível 2 - Repetível (6 processos):

RM - Requirement Management

SPP - Software Project Planning

SPTO - Software Project Tracking and Oversight

SSM - Software Subcontract Management

SCM - Software Configuration Management

Nível 3 - Definido (7 processos):

OPF - Organization Process Focus

OPD - Organization Process Definition

TP - Training Program

ISM - Integrated Software Management

SPE - Software Product Engeneering

IC - Intergroup Coordination

PR - Peer Reviews

Nível 4 Gerenciado (2 processos):

QPM - Quantitative Process Management

SQM - Software Quality Management

Nível 5 - Otimizado (3 processos):

DP - Defect Prevention

TCM - Technology Change Management

PCM - Process Change Management

3.3 Descrição dos Processos-Chave do CMM

A seguir estão descritos todos os processos do CMM separados pelos níveis de maturidades aos quais pertencem.

Nível 2

3.3.1 RM - Requirement Management

Este processo procura estabelecer um entendimento único do projeto de desenvolvimento de software entre o cliente e a equipe do projeto. Este acordo é a base para o planejamento e gerenciamento do projeto.

É necessário descrever os requisitos de sistema que, quando implementados, irão satisfazer as necessidades do cliente. Sempre que este acordo precisar ser alterado por qualquer que seja o motivo, será necessário rever as condições, prazos e custos anteriormente estabelecidos e o documento alterado deverá ser novamente submetido à apreciação do cliente e da equipe de desenvolvimento do software a fim de estabelecer um novo acordo com a nova situação do projeto.

3.3.2 SPP - Software Project Planning

Estabelece planos viáveis para a execução do projeto no aspecto da engenharia e da gerência dos trabalhos.

Os planos são baseados em estimativas realizadas estabelecendo os recursos necessários para a execução do projeto. Este planejamento inclui passos para

estimar o tamanho do software e recursos necessários para produzir um cronograma, identificar e tratar riscos e negociar recursos.

3.3.3 SPTO - Software Project Tracking and Oversight

Fornece uma boa visibilidade da execução do software permitindo acompanhar a evolução dos trabalhos. Os eventuais problemas que surgem no decorrer do projeto devem ser registrados e acompanhados até que seja encontrada a sua solução.

Este acompanhamento deve ser realizado com base nos planos gerados no processo SPP. O processo acompanha os resultados e os confronta com o plano SPP tomando ações corretivas quando necessário. Nestas ações incluem-se revisões do plano a fim de melhorar a produtividade do projeto.

3.3.4 SSM - Software Subcontract Management

O objetivo deste processo é de selecionar terceiros qualificados em desempenhar atividades de desenvolvimento de software e gerenciá-los eficientemente.

A seleção dos terceiros é baseada na habilidade de executar trabalhos, mas outros fatores contribuem para esta decisão. Terceiros também devem ser selecionados baseado em alianças estratégicas de negócio, da mesma forma como considerações de capacidade e conhecimentos técnicos.

O trabalho a ser realizado pelos terceiros e o plano para o trabalho são documentados, e o contratante monitora a performance de acordo com estes planos.

3.3.5 SQA - Software Quality Assurance

Procura um gerenciamento com uma visibilidade apropriada dentro do processo sendo usado e pelos produtos sendo construídos.

Esta visibilidade é alcançada através de revisões e auditorias nos produtos de software e atividades a fim de verificar se estão de acordo com os padrões estipulados e procedimentos adequados. Questões de conformidade são definidas no projeto de software, o SQA as confere e registra as não conformidades para uma solução apropriada.

3.3.6 SCM - Software Configuration Management

A proposta do SCM é estabelecer e manter a integridade dos produtos de software através do ciclo de vida definido no projeto.

Este integridade é alcançada através da identificação e configuração dos software em dados momentos, sistematicamente controlando mudanças de configuração e mantendo a integridade e rastreabilidade desta configuração através do ciclo de vida de desenvolvimento. As linhas de base (base lines) são mantidas em uma biblioteca de linhas de base de acordo com o seu desenvolvimento. Mudanças a estas linhas de base e as versões dos produtos de software construídas são sistematicamente controladas através de controle de mudanças e funções de auditorias de configuração do SCM.

Nível 3

3.3.7 OPF - Organization Process Focus

Estabelece uma responsabilidade organizacional para as atividades do processo de desenvolvimento de software. As pessoas que implementam o processo devem estar intimamente envolvidas com suas definições e comprometidos com o processo.

3.3.8 OPD - Organization Process Definition

Desenvolver e manter um conjunto usável de processos de desenvolvimento de ativos de software a fim de aprimorar a performance através dos projetos e fornecer bases para definir dados significantes para um gerenciamento quantitativo do processo. Estes ativos fornecem fundamentos estáveis que podem ser institucionalizados através de mecanismos como treinamento.

A definição de processos envolve desenvolver e manter padrões do processo de desenvolvimento de software da organização, juntamente com os ativos de software, como descrições do ciclo de vida do processo de desenvolvimento de software, linhas gerais e critérios para o acompanhamento do processo, o banco de dados do processo de software da organização e uma biblioteca de documentação de software relacionados ao processo. Estes ativos devem ser coletados de várias formas; por exemplo, as descrições a respeito do ciclo de vida do processo pode ser parte integral dos padrões de desenvolvimento de software da empresa.

3.3.9 TP - Training Program

Desenvolver as bases e o conhecimento dos indivíduos de forma que possam desempenhar seus papéis eficientemente e eficazmente. O treinamento é uma responsabilidade da organização, mas os projetos de software são responsáveis por identificar as necessidades básicas e fornecer o treinamento necessário quando a necessidade do projeto for única.

Um programa de treinamento começa através da identificação de necessidades de treinamento da organização, e indivíduos, então desenvolver ou contratar treinamento para atender estar necessidades. Estas necessidades devem ser específicas ao projeto ou a indivíduos em um momento bem específico, mas o treinamento necessário pode ser identificado baseado nos papéis e responsabilidades específicas dentro dos padrões do processo de desenvolvimento de software da organização. Algumas necessidades de treinamento são atendidas eficientemente e eficazmente através de veículos de treinamento informal, por exemplo: mentorização. Outras necessitam de um treinamento mais formal como treinamento em sala de aula.

3.3.10 ISM - Integrated Software Management

Este processo se propõe a gerar uma integração coerente e bem definida entre a engenharia de software e o gerenciamento do projeto. Este processo-chave é uma evolução do SPP e SPTO definidas no nível 2 a fim de aproveitar a infra-estrutura organizacional estabelecida no nível 3. Satisfazer o ISM significa que o software é

planejado e gerenciado de acordo com um processo bem definido nos ativos do processo de desenvolvimento de software da organização.

3.3.11 SPE - Software Product Engeneering

A proposta deste processo-chave é executar consistentemente um processo de engenharia bem definido que integra todas as atividades de engenharia para produzir um produto de software correto, consistente de forma eficiente e eficaz. Produtos de software de engenharia descrevem atividades técnicas para o projeto, por exemplo, requisitos, análise, projeto, codificação e testes.

Estes processos de engenharia envolvem documentar os produtos de software e manter a rastreabilidade e consistência entre eles. Isto é necessário para garantir uma transição controlada entre os estágios do ciclo de vida e para fornecer produtos de software de alta qualidade para o cliente.

3.3.12 IC - Intergroup Coordination

Este processo-chave se propõe a estabelecer a forma como a equipe de engenharia do projeto irá se relacionar com outras equipes de engenharia. Normalmente se trata de projetos sendo realizados em parcerias de desenvolvimento ou desenvolvimento concorrente. De qualquer forma, as interações entre as equipes devem ser planejadas e acompanhadas a fim de garantir a qualidade e integridade do sistema como um todo. Todas as equipes deve estar a par do atual estágio e dos planos para o desenvolvimento de todas as equipes.

3.3.13 PR - Peer Reviews

A proposta deste processo-chave é remover defeitos dos produtos de trabalho de software de forma rápida e eficiente.

O peer review é um importante método implementável através de inspeções, walkthoughs ou outros métodos de revisão pertinentes.

Nível 4

3.3.14 QPM - Quantitative Process Management

A proposta deste processo-chave é de controlara performance do processo do projeto de desenvolvimento de software de maneira quantitativa.

A performance do processo de desenvolvimento de software representa os resultados atuais atingidos ao seguir o processo de desenvolvimento de software. Cada execução de projetos possui uma performance que, mesmo com um processo bem definido e uma equipe muito bem treinada, pode variar um pouco. Esta variação também pode ser conhecida e portanto é possível estabelecer uma faixa aceitável para a performance dos projetos. Portanto, com um processo estável, a performance está normalmente estará em uma faixa conhecida, e quando esta performance cai e sai desta faixa existe a necessidade de identificar a "razão especial" desta variação, e quando apropriado, corrigir as circunstâncias que levaram o projeto a ter esta variação fora do normal. O resultado esperado deste processo-chave é que o processo se mantenha estável e quantitativamente previsível.

3.3.15 SQM - Software Quality Management

O objetivo deste processo-chave é desenvolver um entendimento quantitativo da qualidade dos produtos de software do projeto e alcançar metas específicas de qualidade.

Objetivos quantitativos são estabelecidos para os produtos de software baseados nas necessidades da organização, dos clientes, e usuários finais. SQM é focado em produto, enquanto QPM é focado no processo.

Nível 5

3.3.16 DP - Defect Prevention

A proposta deste processo-chave é identificar as causas dos defeitos e prevenir que estes reapareçam.

O projeto de desenvolvimento de software analisa o defeito identifica as suas causas e toma atitudes para prevenir que estes reapareçam. É comum que ao tomar estas atitudes o processo de desenvolvimento de software tenha que ser alterado. Isto poderá até acarretar mudanças nos padrões da organização. DP é um mecanismo para melhorar incrementalmente o processo de desenvolvimento de software de uma forma evolutiva.

3.3.17 TCM - Technology Change Management

A proposta do TCM é identificar novas tecnologias que possam estar sendo utilizadas pela organização e transferi-la para a organização de uma maneira controlada e organizada.

Foco em transferência de tecnologia implica em identificar, selecionar, avaliar novas tecnologias e incorporá-las de forma eficiente dentro da organização. O objetivo é melhorar a qualidade do software gerado, aumentar a produtividade, diminuir o tempo no ciclo de vida para o desenvolvimento do produto. O resultado esperado para o TCM é de executar inovações de forma eficiente em um mundo em constante mudança. O TCM é melhorar o processo de desenvolvimento de software de uma maneira revolucionária.

3.3.18 PCM - Process Change Management

A proposta do PCM é melhor continuamente o processo de desenvolvimento de software utilizado na organização com a intenção de melhorar a qualidade, aumentar a produtividade e diminuir o tempo no ciclo de vida de desenvolvimento do produto. Um processo contínuo envolve definir objetivos de melhoria no processo e com um gerente sênior, proativamente e sistematicamente identificar, avaliar e implementar melhoria aos padrões da organização e do processo de desenvolvimento de software. O PCM se concentra em desenvolver, tanto mudanças evolutivas incrementais quanto mudanças revolucionárias e inivadoras.

4 RUP Rational Unified Process

O RUP é uma das muitas metodologias de engenharia de software orientadas a objeto que concorrem entre si no mercado. Ele foi desenvolvido pela Rational Software Corporation e é composto de ferramentas on-line para desenvolvimento de software e padrões de documentação (templates).

Ele fornece uma abordagem disciplinada para designar tarefas e responsabilidades em uma organização que se propõe a desenvolver software. O seu objetivo é garantir a produção de software de alta qualidade que atenda as necessidades de seus usuários finais com um cronograma e orçamento previsíveis.

4.1 Arquitetura do RUP

O RUP está dividido em 4 fases distintas que percorrem todo o desenvolvimento de software.

O primeira fase é a **Inception** que se propõe a especificar a visão do produto de acordo com seus usuário finais.

A segunda fase é a **Elaboration** que planeja as atividades necessárias; os recursos necessários; especificando os recursos e desenhando a arquitetura do software.

A terceira fase é a **Construction** que se concentra na confecção do produto baseada na visão do produto e nos planos de atividades.

A quarta e última fase do RUP é a Transition que se concentra na transição do produto para os usuários finais. Nesta fase incluem-se as atividades de empacotamento, entrega, treinamento, suporte e manutenção do software.

Estas quatro fases constituem o ciclo de vida RUP. Cada uma destas fases possui uma seqüência de micro-fases que podem ser repetidas gerando uma ou mais iterações dentro da mesma fase. Estas micro fases são:

- **Modelagem de Negócio** (Business Modeling);
- **Requisitos** (Requirements);
- **Análise e Projeto** (Analysis and Design);
- **Implementação** (Implementation);
- **Teste** (Test);
- **Entrega** (Deployment);
- **Gestão de Configuração** (Configuration Management);
- **Gestão de Projetos** (Project Management) e
- **Ambiente** (Environment).

*4.1.1 Modelagem de Negócio **(Business Modeling)***

O objetivo deste processo é entender a estrutura e a dinâmica da organização de uma forma única para clientes, usuários-finais e desenvolvedores a qual o sistema está destinado. Também se propõe a entender os problemas atuais, identificar

melhorias potenciais e delinear os requisitos necessários para dar suporte a organização alvo.

4.1.2 Requisitos ***(Requirements)***

Este processo se propõe a coletar os requisitos do software a ser desenvolvido. Para isso o processo deve estabelecer e manter um acordo com os clientes e outros stakeholders no que o software deve ou não deve conter. Fornece um melhor entendimento dos requisitos do software. Define o escopo do software. Fornece a base para planejar o conteúdo técnico das iterações. Fornece a base para estimar custo e tempo para desenvolver o software. Definir uma interface para o software focando as necessidades do usuário-final.

4.1.3 Análise e Projeto ***(Analysis and Design)***

Este processo se propõe a traduzir os requisitos em um especificação que descreva como implementar o software. Para realizar esta tradução, é necessário entender os requisitos e transformá-los em um projeto de software através da escolha da melhor estratégia de implementação. É necessário primeiramente estabelecer uma arquitetura de forma que o software seja facilmente entendido e construído. Então deve-se ajustar o projeto a fim de combinar com o ambiente de forma a conseguir performance, robustez, escalabilidade e testabilitade entre outras qualidades.

4.1.4 Implementação ***(Implementation)***

Este processo se propõe a definir a organização do código em termos de sub-sistemas organizados em camadas. Implementação de classes e objetos em termos de componentes (código-fonte, executáveis e outros). Realizar o teste unitário das componentes de código geradas. Integrar os resultados produzidos por implementações individuais ou de equipes em um sistema executável único.

4.1.5 Teste ***(Test)***

A proposta deste processo é a de verificar as interações entre os componentes de software, verificar se todos os requisitos foram implementados corretamente, identificar e garantir que todos os defeitos encontrados e registrados anteriormente foram resolvidos.

4.1.6 Entrega ***(Deployment)***

A proposta deste processo é testar o software no ambiente final (Beta Teste); empacotar o software para entrega; distribuir o software; instalar o software; treinar os usuários-finais e a equipe de vendas; migrar os softwares existentes ou converter banco de dados.

4.1.7 Gestão de Configuração ***(Configuration Management)***

A proposta deste processo é acompanhar e manter a integridade dos ativos do projeto. Durante o desenvolvimento, muitos produtos intermediários são gerados e devem ser acompanhados e mantidos íntegros para um eventual reúso.

4.1.8 *Gestão de Projetos* ***(Project Management)***

Fornecer um roteiro de trabalho para o gerenciamento de projetos de desenvolvimento de software; fornecer diretivas práticas para planejar, contratar =, executar e monitorar projetos; fornecer um roteiro de trabalho para acompanhamento de riscos. Não é coberto o gerenciamento de pessoas, de orçamento e de subcontratados, o foco está no planejamento iterativo através do ciclo de vida de desenvolvimento, controle de riscos e monitoramento do progresso do desenvolvimento através de métricas.

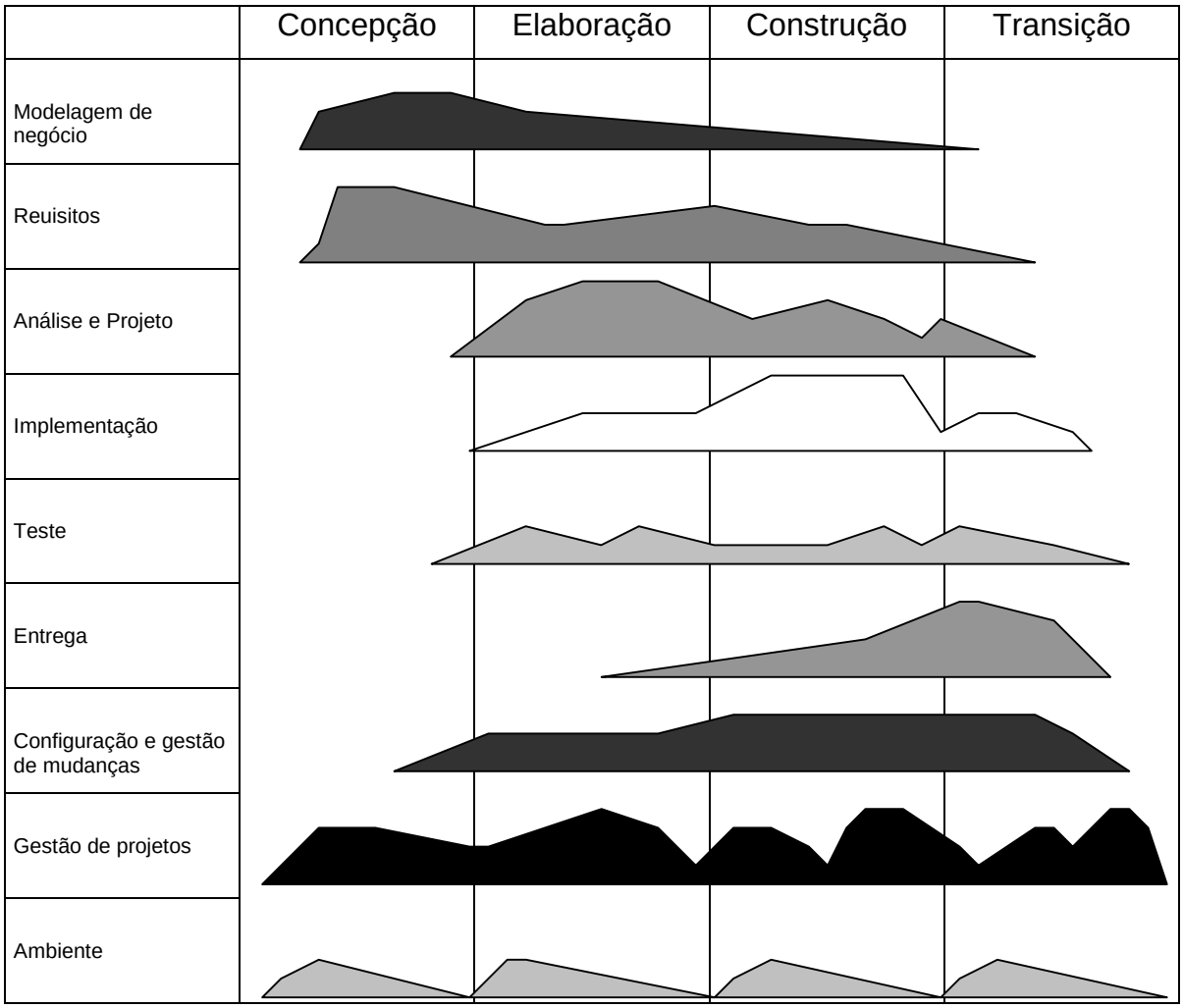
4.1.9 *Ambiente* ***(Environment)***

A proposta deste processo é a de escolha e aquisição de ferramentas, configuração das ferramentas de forma adequada para uso na organização, processo de configuração, processo de melhorias, Serviços técnicos para suportar o processo; Infra-estrutura para tecnologia da informação; administração, backup entre outros.

4.2 Ciclo de vida RUP

O ciclo de vida do RUP é iterativo e incremental. Cada uma das fases do processo de desenvolvimento RUP poderá ser repetida a fim de incrementar o seu desenvolvimento. Cada execução é chamada de iteração, portanto, em cada fase poderemos ter mais de uma iteração. Estatisticamente temos uma iteração na fase de Concepção (Inception), duas iterações na fase de Elaboração (Elaboration), três na fase de Construção (Construction) e duas na fase de Transição (Transition).

Todas as 9 sub-fases podem estar presentes em cada iteração. O que ocorre é que cada fase possui um objetivo diferente e por isso é natural que algumas sub-fases estejam sendo mais focadas em uma fase do que nas outras. Por exemplo, A sub-fase Requisitos está muito mais ligada a fase de Concepção e Elaboração do que a Construção e Transição. O gráfico a seguir ilustra as atividades de cada sub-fase no decorrer das fases do RUP. Note que existe a concentração de esforço das atividades em fases específicas.



4.3 Definindo o processo de desenvolvimento da organização baseado no cilco de vida RUP.

O RUP fornece todos os recursos para que a organização defina o seu processo de desenvolvimento de software.

A seguir está um exemplo utilizando um número reduzido de iterações o qual poderia ser um possível processo de desenvolvimento baseado no RUP.

4.3.1 Exemplo reduzido de um processo baseado no RUP

Fase I - Concepção (1 iteração)

1ª Iteração

Requisitos

Atividade: Reunião inicial com o cliente

Atividade: Levantamento de requisitos vagos

Análise

Atividade: Elaboração do plano preliminar de projeto

Atividade: Elaboração do Business Case inicial (proposta)

Atividade: Análise inicial de riscos

Implementação

Atividade: Elaboração da visão do produto

Atividade: Elaboração de um protótipo de telas inicial

Gestão de Projetos:

Atividade: Planejamento da próxima iteração

Fase II - Elaboração (2 iterações)

2ª Iteração

Requisitos

Atividade: Levantamento de requisitos

Atividade: Refinamento dos requisitos

Atividade: Elaboração do protótipo de navegação das telas (não funcional)

Análise

Atividade: Identificação das classes

Atividade: Revisão do Business Case inicial

Projeto

Atividade: Construção do diagrama de classes

Atividade: Construção da dinâmica dos objetos

Atividade: Detalhamento das classes

Atividade: Geração de código

Implementação

Atividade: Implementação dos métodos

Atividade: Construção do protótipo funcional do software

Testes:

Atividade: Testes de unidade do protótipo

Entrega:

Atividade: Configuração do servidor para abrigar o software

Gestão de Projetos:

Atividade: Planejamento da próxima iteração

3ª Iteração

Requisitos

Atividade: Levantamento de novos requisitos

Atividade: Refinamento dos requisitos

Atividade: Elaboração do novo protótipo de navegação das telas (não funcional)

Análise

Atividade: Identificação de novas classes

Projeto

Atividade: Construção do diagrama de classes

Atividade: Construção da dinâmica dos objetos

Atividade: Detalhamento das classes

Atividade: Geração de código

Implementação

Atividade: Implementação dos métodos

Atividade: Construção da versão Alfa do software

Testes:

Atividade: Testes da versão Alfa

Entrega:

Atividade: Configuração do servidor para abrigar o software

Atividade: Instalação da versão Alfa no Servidor

Gestão de Projetos:

Atividade: Planejamento da próxima iteração

Fase III - Construção (1 iteração)

4ª Iteração

Requisitos:

Atividade: Walkthrough

Atividade: Identificação de bugs

Análise:

Atividade: Analisar impacto dos bugs encontrados.

Projeto:

Atividade: Alterar modelo de classes

Atividade: Elaboração dos manuais de usuário.

Implementação:

Atividade: Implementação dos bugs identificados

Atividade: Finalização da implementação do software

Testes:

Atividade: Testes com a participação do usuário final

Entrega:

Atividade: Confecção dos manuais

Atividade: Levantamento do ambiente necessário para a instalação

Gestão de Projetos:

Atividade: Planejamento da próxima iteração

Fase IV - Transição

5ª Iteração

Requisitos:

Atividade: Levantamento do hardware e software necessário para instalação do software no cliente.

Análise:

Atividade: Estudo da configuração necessária dos equipamentos.

Projeto:

Atividade: Planejamento da configuração e da instalação dos equipamentos.

Implementação:

Atividade: Instalação e configuração dos equipamentos.

Testes:

Atividade: Testes do funcionamento dos equipamentos

Entrega:

Atividade: Instalação do produto.

Atividade: Testes de funcionamento do produto

Atividade: Treinamento dos usuários

Gestão de Projetos:

Atividade: Finalização do projeto

Todas as atividades descritas neste processo de desenvolvimento possuem artefatos sendo gerados. Para que o processo esteja completo, estes artefatos precisam estar descritos atividade por atividade a fim de se definir a sua forma padrão de documentação. Os padrões para estes documentos devem estar previamente elaborados e disponíveis para a realização dos projetos (templates). Uma cópia devidamente identificada deve estar anexada na documentação do processo. Além da identificação dos artefatos gerados, deve ser realizada uma descrição explicativa de cada atividade. Estas explicações devem estar disponíveis para a equipe de desenvolvimento a fim de esclarecer dúvidas no momento da confecção dos artefatos de desenvolvimento. Veja um exemplo de uma definição de atividade:

ATIVIDADE IR1 Reunião Inicial com o Cliente

Esta atividade representa o primeiro contato com o cliente e visa levantar rapidamente uma noção do escopo inicial do projeto a ser realizado coletando informações suficientes para planejar as sessões "JAD Plan" as quais efetivamente levantarão os requisitos vagos do sistema. Esta atividade gera um artefato "IR1_A1 - Designação de Responsabilidades" e o "IR1-A2 - Ata da Reunião Inicial". O artefato IR1-A1 identificará os principais papeis e suas respectivas responsabilidades para o andamento do projeto, e o artefato IR1-A2 deverá relatar a reunião anotando as decisões tomadas, as pendências encontradas e os responsáveis por acompanhá-las ou resolvê-las.

Artefatos necessários para esta atividade:

nenhum

Artefatos gerados por esta atividade:

[IR1-A1] - Designação de Responsáveis

[IR1-A2] - Ata da Reunião Inicial

Após definir cada uma das atividades o processo estará definido. Este trabalho não leva em conta os fluxos de trabalho do RUP. Estes devem ser levados em conta no momento de se definir as atividades do processo de desenvolvimento.

A tabela abaixo mostra os artefato gerados em cada uma das fases do RUP.

Fase	Artefatos	Critérios de Sucesso
Concepção	• Documento de visão ou declaração de escopo • Casos de uso iniciais (aproximadamente 20%) • Business case inicial • Avaliação inicial de riscos • Plano preliminar de projeto • Um ou mais protótipos	• Concordância dos stakeholders do escopo do projeto • Entendimento documentado dos requisitos • Estimativas de custo e cronograma • Profundidade e abrangência do protótipo • Despesas atuais x planejadas
Elaboração	• Casos de uso revisados (aproximadamente 80%) • Plano de desenvolvimento de software (plano de gerenciamento, plano de contratação, planejamento de fases, planejamento de próximas iterações e planejamento de testes) • Avaliação atualizada dos riscos • Business case revisado • Descrição da arquitetura do software • Linha base executável da arquitetura	• Documento da visão de linha de base • Critério de avaliação mensurável para avaliação inicial das iterações durante a construção • Estimativas de custos e cronograma • Despesas atuais x planejadas
Construção	• Planos individuais de iteração • Documento de descrição de versão • Resultados dos casos de teste • Plano de entrega • Documentação do usuário • Produto incremental	• Produto estável e pronto para ser versionado • Stakeholder pronto para a transição • Despesas atuais x planejadas
Transição	• Versão final do produto • Documentação do produto atualizada • Análise das lições aprendidas	• Aceite do cliente • Satisfação do usuário • Despesas atuais x planejadas

O próximo capítulo mostra uma forma de se realizar um mapeamento entre RUP, CMM e PMBOK.

5 Mapeamento RUP, CMM e PMBOK

Cada organização possui a sua forma de trabalho, suas atividades e seu "knowhow" próprio.

Portanto não é interessante descrever atividade por atividade de um projeto específico o qual não teria muita serventia a não ser como um exemplo não aplicável. Portanto este trabalho se preocupa em mostrar como realizar este mapeamento.

Primeiramente faremos algumas considerações importantes a respeito dos assuntos até aqui discutidos.

O PMBOK trata-se de gestão de projetos de qualquer natureza e define áreas que deve ser contempladas para a gerência de projetos.

O CMM define processos-chave que devem ser contemplados para o desenvolvimento de software ele não é aplicável em projetos de natureza diferentes de desenvolvimento de software.

O RUP é um processo mais prático que define uma seqüência de fases, sub-fases e atividades para a confecção de softwares de qualidade. Ele não aborda questões a respeito de gerenciamento de pessoas ou acompanhamento de orçamento.

Como foi visto o RUP está dividido em quatro grandes fases Concepção, Elaboração, Construção e Transição. Também visto anteriormente temos o PMP com cinco grandes grupos de processo: Inicialização, Planejamento, Execução,

Controle e Fechamento. O CMM não define um processo seqüencial de desenvolvimento ele busca introduzir melhores práticas para o desenvolvimento de software utilizando o ciclo de vida definido pela própria organização. No caso deste trabalho o iterativo incremental. O CMM se propõe a garantir que determinados aspectos são cobertos pelo processo de desenvolvimento de software da própria organização. Aspectos estes que estão divididos em quatro grandes blocos: Processo, Pessoas, Tecnologia e Métricas.

Como o RUP e o PMBOK definem uma macro-seqüência para o processo de desenvolvimento de software o mapeamento realizado neste trabalho primeiramente apenas os levará em conta, e o CMM será inserido posteriormente.

5.1 Comparação RUP e PMBOK

As fases de Concepção do RUP e a fase de Inicialização do PMBOK possuem exatamente o mesmo objetivo divergindo apenas em pequenos pontos. Tanto do RUP quanto no PMBOK neste momento as equipes estão definido o projeto a fim de estabelecer um acordo com os clientes e usuários-finais a respeito do projeto a ser desenvolvido. Questões como escopo, cronograma e custo são definidas nestas etapas. O PMP não fornece padrões de documentação, portanto os padrões do RUP poderão ser utilizados (é importante lembrar que existem outros padrões para documentação utilizando o PMBOK um exemplo disso é o grupo TenStep (www.tenstep.com) que possui uma biblioteca on-line via Internet de padrões de documentação que podem ser utilizados para este fim).

Portanto na fase de Concepção e Inicialização temos realmente um mapeamento praticamente um para um. O controle realizado pelo RUP está adequado ao controle descrito pelo PMBOK.

Na fase de Planejamento do PMBOK o objetivo é preparar uma solução técnica viável para satisfazer os requisitos dos clientes. Esta solução descreve como os objetivos do projeto serão acompanhados como novas tecnologias serão abordadas e como estas serão assimiladas pela equipe de desenvolvimento do produto.

Os artefatos gerados nesta fase são similares aos artefatos gerados na fase de Elaboração do RUP. Os dois processos exigem que o plano do projeto seja aprovado pelo usuário para poder prosseguir para a próxima fase. Mas mesmo assim as informações e detalhamento em cada um dos processos pode ser diferenciada. No RUP o plano do projeto contém a idéia do projeto todo, porém, apenas para a primeira iteração do plano é que ele define com um grau maior de detalhes. No caso do PMBOK o projeto como um todo deve ser definido.

A partir deste ponto os dois processos se diferenciam na ênfase, o RUP passa a se preocupar com a execução o o PMBOK dá ênfase no gerenciamento administrativo. O PMBOK neste momento possui um foco maior no controle do desenvolvimento, garantir que o projeto será concluído no prazo e que tudo o que foi planejado seja efetivamente desenvolvido dentro no prazo e custo pré-determinados.

Na fase de Execução no PMBOK é comparável com as fases de Construção e Transição do RUP. A ênfase está no controle do desenvolvimento, prazos e custos.

Na fase de Fechamento a ênfase está em atividades de documentação, aprovações dos clientes e usuários-finais, pagamentos de terceiros e documentação das lições aprendidas no processo. Estas atividades com exceção dos pagamentos a terceiros, se encontram na fase de Transição do RUP.

Portanto a idéia é garantir que todos os aspectos do PMBOK estejam contemplados no ciclo de vida do RUP, na questão em que uma fase do RUP esteja contemplando duas do PMBOK, deveremos apenas definir as atividades destas fases identificando quais as fases do RUP e fases do PMBOK a que estão vinculadas.

Com isso podemos criar projetos utilizando o RUP dentro dos padrões do PMBOK.

5.2 Incluindo o CMM no processo RUP+PMBOK

O CMM determina áreas de conhecimento e estipula processo que deve ser elaborados para cada situação.

Os processos são construídos tendo a organização como base. Como por definição o processo utilizado por esta empresa é o RUP+PMBOK, devemos vincular este processo às áreas de conhecimento do CMM.

Os processos do CMM já definidos anteriormente são:

2 RM, SPP, SPTO, SCM, SQA, SSM

3 ISM, IC, OPF, OPD, TP, SPE, PR

4 QPM, SQM

5 PCM, TCM, DP

Cada atividade do processo de desenvolvimento pode estar utilizando mais de uma destas áreas de conhecimento. É importante então para a realização deste mapeamento identificar as áreas que estão vinculadas a cada uma das atividades e especificar qual o tipo de relacionamento que a atividade tem com cada uma das áreas de conhecimento do CMM. A seguir encontra-se um exemplo de uma atividade com as respectivas anotações de mapeamento. Note que a atividade possui referência ao PMBOK ao CMM e ao RUP.

ATIVIDADE IR1 Reunião Inicial com o Cliente
RUP - Concepção ***PMP - Planejamento***
Esta atividade representa o primeiro contato com o cliente e visa levantar rapidamente uma noção do escopo inicial do projeto a ser realizado coletando informações suficientes para planejar as sessões "JAD Plan" as quais efetivamente levantarão os requisitos vagos do sistema. Esta atividade gera um artefato "IR1_A1 - Designação de Responsabilidades" e o "IR1-A2 - Ata da Reunião Inicial". O artefato IR1-A1 identificará os principais papeis e suas respectivas responsabilidades para o andamento do projeto, e o artefato IR1-A2 deverá relatar a reunião anotando as decisões tomadas, as pendências encontradas e os responsáveis por acompanhá-las ou resolvê-las.
Artefatos necessários para esta atividade: *nenhum*
Artefatos gerados por esta atividade:
[IR1-A1] - Designação de Responsáveis
[IR1-A2] - Ata da Reunião Inicial

Nível CMM	**Processos do CMM**						
2	RM **realiza**	SPP -	SPTO -	SCM **registra**	SQA **confere**	SSM -	
3	ISM -	IC -	OPF -	OFD -	TP -	SPE -	PR -
4	QPM -	SQM -					
5	TCM -	PCM -	DP -				

Ao definir todas as atividades do processo de desenvolvimento de software da organização desta maneira, é possível garantir a conformidade com os padrões do RUP, PMP e CMM. Lógico que documentos de definição destes padrões terão de ser confeccionados de acordo com cada um dos processos.

Não basta realizar este mapeamento, mas também é necessário realizar uma auditoria em três etapas para este processo. Uma para cada um dos processos que a compõe. Primeiramente com o RUP pois é o processo que define as atividades em mais baixo nível, depois com o PMBOK pois é um padrão que trata a gerência de projetos definindo uma macro-seqüência que pode ser confrontada diretamente com o RUP, e por fim o CMM que define as áreas sem definir uma ordem. Cada área deve atender as atividades do processo.

6 Conclusão

As organizações são formadas por um número muito grande de processos, Não é difícil termos atividades guiadas por mais de um processo simultaneamente. É importante ter consciência dos processos que regem a organização a fim de poder sincronizá-los e controlá-los de forma adequada. Podem acontecer problemas de objetivos conflitantes ou artefatos que para uns processos são importante e para outro não o que pode gerar uma carga muito grande de trabalho e de controle diminuindo a capacidade de produzir efetivamente. É importante que o controle não passe a ser a atividade mais importante superando até o produto ou a sua qualidade. Os objetivos da organização são satisfazer seus clientes fornecendo software de alta qualidade com um cronograma e custos adequados.

O fator da certificação profissional e da empresa tem tomado cada vez mais a atenção das organizações. Cada vez mais a certificação está sendo sinônimo de qualidade e garantias de obter resultados satisfatórios. Empresas em cada vez mais adotado certificações como um critério de seleção de fornecedores e funcionários. Portanto não podemos ignorar este fato e a importância de produzir com qualidade e obter certificados de relevância internacional para a realização de nossas atividades.

Este trabalho é o início de uma discussão a respeito da integração destas certificações e como cada vez mais as organizações podem se aprimorar no desenvolvimento de software para ganhar competitividade e qualidade.

7 Bibliografia

PAULK, Mark C.; **WEBER**, Charles V.; **CURTIS**, Bill; **CHRISSIS**, Mary Beth – "*The Capability Maturity Model: Guidelines for Improving the Software Process*" – Ed. Addison Wesley – 1995.

BOOCH, Grandy; **RUMBAUGH**, James; **JABOBSON**, Ivar – "*The Unified Modeling Language User Guide*" – Ed. Addison Wesley – 1998.

JACOBSON, Ivar; **BOOCH**, Grandy; **RUMBAUGH**, James – "*The Unified Software Development Process*" – Ed. Addison Wesley – 1998.

KRUCHTEN, Philippe - "*The Rational Unified Process - An Introducton*" - 2ª edição - Ed. Addison Wesley - 2000.

PMI – Project Management Institute – "*A guide to the project management body of knowledge (PMBOK guide)*" – PMI – 2000.

LAMBERTSEN, Larry - "*Project Management in a Rational Unified Process (RUP) Environment*" - Proceedings of the Project Management Institute Annual Seminars & Symposium - October 3, 2002 - San Antonio, Texas USA - 2002.