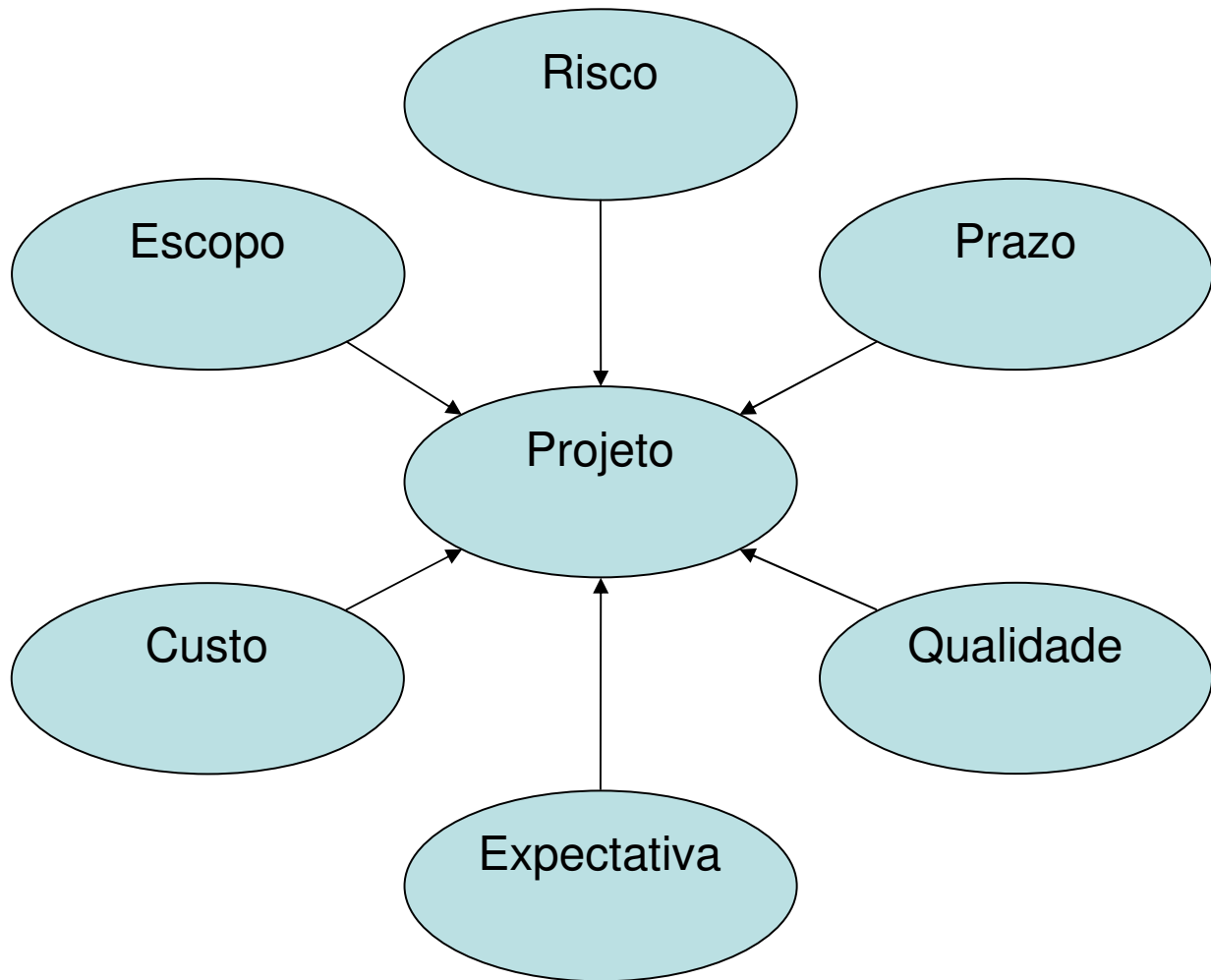


Gerenciamento de Projetos

Prof. Cláudio Luís Vieira Oliveira

clvoliveira@terra.com.br

Introdução



Sistemas de Informação

É conjunto de elementos ou componentes inter-relacionados que coletam (entrada), manipulam (processamento) e disseminam (saída) os dados e informação e fornecem um mecanismo de re-alimentação (feedback) para atender um objetivo.

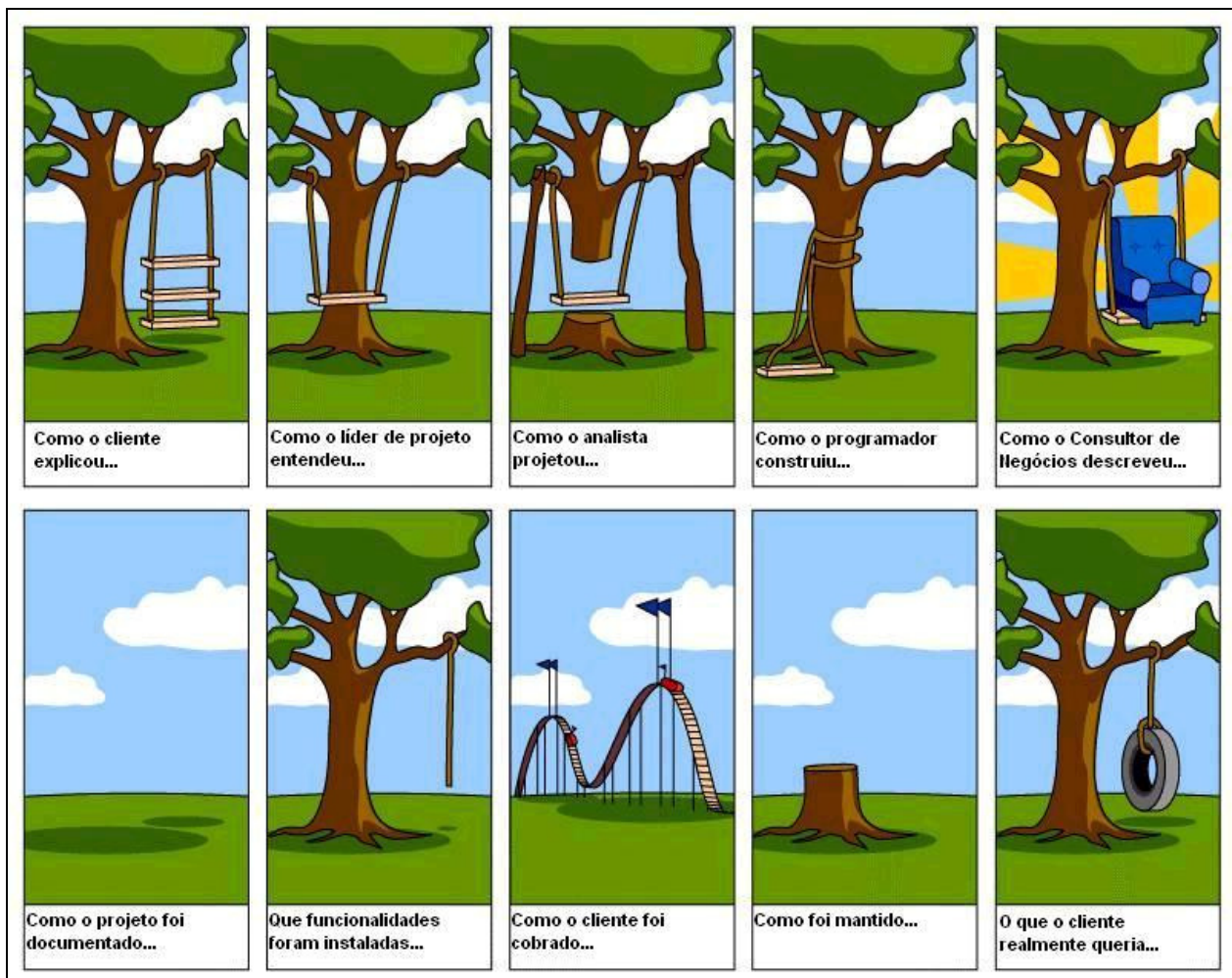
Desafios no Projeto de Sistemas de Informação

“Muitos desenvolvedores de software começam querendo construir prédios altos, como se estivessem fazendo uma casinha de cachorro.”

Grady Booch¹



Fases de um Projeto



¹ Booch, G.; Rumbaugh, J.; Jacobson, I. **UML: Guia do usuário**. Editora Campus, 2006.

Conceito de Projeto

Projeto é um conceito que é inerente ao ser humano, tudo aquilo que realizamos ao longo da nossa vida consiste, na verdade, de pequenos projetos.

“Um empreendimento temporário com o objetivo de criar um produto ou serviço único.”

(PMBOK²)

“Projeto é um empreendimento ou evento não repetitivo, caracterizado por uma sequência clara e lógica de eventos, com início, meio e fim, que se destina a atingir um objetivo claro e definido, sendo conduzido por pessoas dentro de parâmetros predefinidos de tempo, custo, recursos e qualidade.”

(VARGAS, 2000³)

Projetos X Operações de Rotina

Com o intuito de diferenciar esses dois conceitos, vamos imaginar um sistema de folha de pagamento que é realizado mensalmente através de uma planilha do Excel, a tarefa de alimentar a planilha com os dados do mês é um bom exemplo de “Operação de Rotina”. A característica principal deste tipo de operação é a continuidade.

Usando esse mesmo exemplo, poderíamos ter a necessidade de melhorar esse processo através da adoção de novas tecnologias e procedimentos abolindo, desta forma, o uso da planilha em Excel. Neste caso temos um projeto, sendo importante observar o seu caráter temporário que dará origem a uma solução que será única.

Evolução do Gerenciamento de Projetos

O histórico do gerenciamento de projeto é quase tão antigo quanto a própria humanidade. As grandes obras do passado como as pirâmides do Egito e a muralha da China entre outras, apenas foram possíveis devido as habilidades de coordenação e planejamento de um gerente de projetos.

O que é Gerenciamento de Projetos?

O Gerenciamento de Projetos pode ser entendido como a aplicação de conhecimentos, habilidades, ferramentas e técnicas nas atividades do projeto a fim de atender aos requisitos do mesmo.

O conceito é melhor entendido através da adoção de grupos de processos no desenvolvimento de um projeto: Iniciação, Planejamento, Execução, Monitoramento, Controle e Encerramento.

Atender aos requisitos do projeto envolve o balanceamento das seguintes demandas conflitantes:

- Escopo, tempo, custo, risco e qualidade do projeto;
- Satisfação de diferentes interessados (stakeholders) com necessidades e expectativas diferentes;
- Requisitos identificados (necessidades) e requisitos não identificados (expectativas).

O ponto principal do conceito de Gerenciamento de Projetos é o controle. Não é possível atender aos requisitos do cliente se não houver controle. O objetivo do Gerenciamento de Projetos é garantir o cumprimento do escopo, dos prazos, dos custos e da entrega dos produtos com a qualidade esperada pelo cliente.

² Project Management Body of Knowledge

³ VARGAS, R. V. **Gerenciamento de Projetos: estabelecendo diferenciais competitivos**. Rio de Janeiro: Brasport, 2000.

Elementos-chave do gerenciamento de projetos

Stakeholders:

Consistem nas pessoas e organizações envolvidas no projeto ou os indivíduos que serão afetados pelo projeto, seja de forma positiva ou negativa. O envolvimento pode ocorrer em qualquer uma das fases do ciclo de vida do projeto. A equipe de gerência do projeto deve identificar todos os envolvidos no projeto antes de iniciá-lo, conhecer suas necessidades e expectativas.

Alguns exemplos de stakeholders mais comuns nos projetos:

- Dono (sponsor) ou patrocinador;
- Cliente;
- Gerente do projeto;
- Organização executora;
- Membros da equipe do projeto.

Pessoas pensam de maneira diferente e possuem objetivos e expectativas diversos. Uma das tarefas mais desafiadoras de um gerente de projeto consiste em encontrar as soluções adequadas para as divergências e conflitos entre os envolvidos no projeto.

Áreas de Conhecimento: descrevem as competências-chave que os gerentes de projeto devem desenvolver. O PMBOK divide essas áreas em nove: gerenciamento da integração, gerenciamento do escopo, gerenciamento do tempo, gerenciamento do custo, gerenciamento da qualidade, gerenciamento dos recursos humanos, gerenciamento das comunicações, gerenciamento dos riscos e gerenciamento das aquisições do projeto.

Ferramentas e Técnicas de Gerenciamento: ajudam os gerentes de projetos e suas equipes a conduzir melhor o escopo, o tempo, os custos e a qualidade.

Fases e ciclo de vida dos projetos

Projetos são empreendimentos únicos, nunca realizados, desta forma, há um inerente grau de incerteza. Para um melhor controle gerencial dessas incertezas, a condução do projeto deve ocorrer em fases. O conjunto das fases do projeto é conhecido como ciclo de vida do projeto.

Em cada fase deve-se ter um conjunto de subprodutos que irão proporcionar um melhor controle do gerenciamento. Em projetos de TI, em especial o desenvolvimento de software, podem ser adotadas as seguintes fases: levantamento de requisitos, análise, implementação, codificação, testes, documentação, implantação, transição e suporte.

Como o projeto tem o fator tempo limitado, o ciclo de vida irá determinar o início e o fim do projeto, além da sequência das fases. Os ciclos também determinam o trabalho técnico a ser executado em cada uma das fases, as pessoas responsáveis, a necessidade de recursos e os custos envolvidos.

Influência das organizações

Os projetos fazem parte das organizações, sem as quais não existiriam. Os projetos estão sujeitos a sofrer algumas influências dependendo da forma como as organizações estão estruturadas. Os fatores relacionados a seguir influenciam diretamente a condução e o contexto do gerenciamento do projeto:

- Maturidade da organização em relação a metodologias de gerenciamento de projetos;
- Cultura;
- Estilo;
- Estrutura organizacional;

Existem organizações orientadas a projetos (dependem dos projetos para sobreviver) e organizações não-orientadas a projetos (são organizadas em departamentos, nos quais realizam-se as operações de rotina).

Principais habilidades do gerente de projetos

Para exercer o papel de gestor de projetos, a pessoa precisa de habilidades de administração geral, por exemplo:

- Liderança;
- Comunicação;
- Negociação;
- Solução de problemas;
- Influência na organização.

Influências socioeconômicas e ambientais

A gerência do projeto deve estar alerta aos aspectos e influências socioeconômicas, uma vez que as condições e tendências do mercado podem refletir diretamente no andamento dos projetos. Essas mudanças podem resultar, por exemplo, em mudanças no escopo, atrasos no cronograma ou alterações no custo. Deve-se acompanhar constantemente as notícias sobre as questões econômicas e políticas nas regiões nos quais o projeto é realizado.

Processos de Gerência de Projetos

Cada projeto é composto por uma série de processos, os quais interagem entre si. Quando se fala em processo, na verdade, estamos nos referindo a resultados pois, por definição:

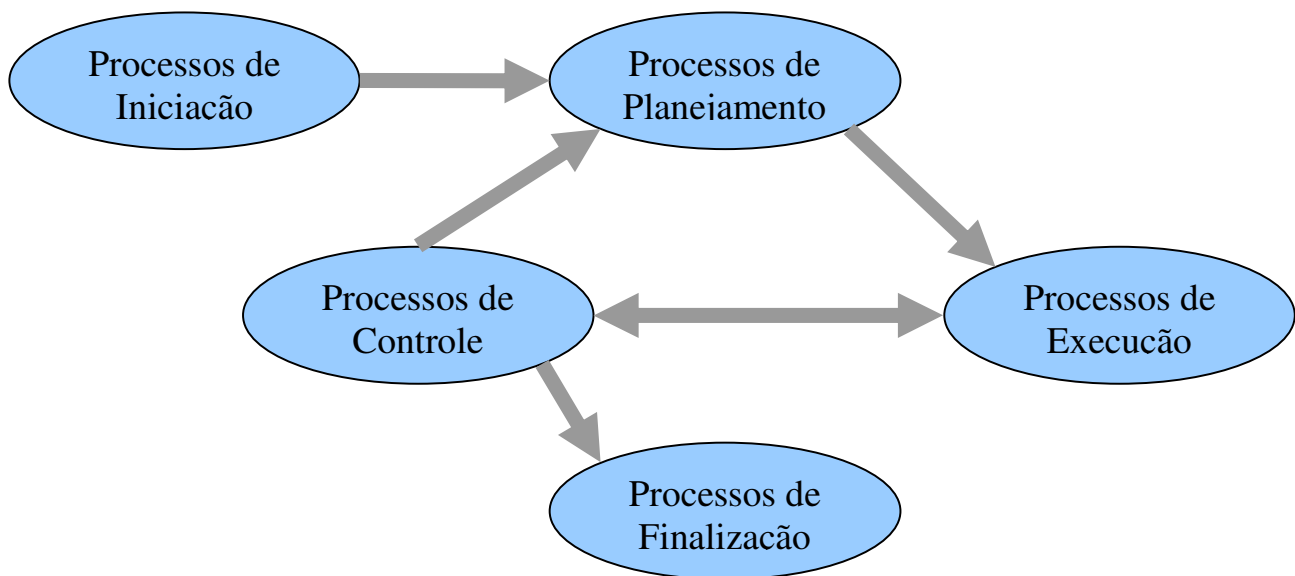
“Processos são uma série de ações que geram um resultado.”

Os processos, por sua vez, podem ser divididos em processos de gerência de projeto (apresentam relacionamento com a descrição e com o planejamento do escopo do projeto) e processos orientados ao produto (estão ligados aos requisitos, especificações e criação do produto do projeto).

Os processos orientados ao produto são definidos pelo ciclo de vida do projeto e podem ser divididos em cinco grupos:

- Iniciação: reconhecem formalmente e autorizam o início de um projeto ou fase. É muito importante, neste processo, obter o comprometimento da organização para a realização do projeto;
- Planejamento: definem e refinam os objetivos e as escolhas das melhores alternativas de ação para atingir os objetivos propostos pelo projeto;
- Execução: responsáveis pela coordenação das pessoas e dos outros recursos para realizar o que foi planejado;
- Controle: servem para monitorar e garantir que os objetivos do projeto sejam atingidos. Deve-se monitorar o progresso com frequência, indentificando qualquer alteração no que foi planejado. Caso essas alterações ocorram, ações proativas e corretivas devem ser tomadas.
- Encerramento: tem a função de formalizar a aceitação do projeto ou da fase e realizar o encerramento do modo organizado.

Interação entre os Grupos de Processos



Os grupos de processos são interligados através dos resultados que cada um deles produz. Além disso, os grupos de processos não podem ser vistos de maneira separada ou descontínua. O que determina uma forte interação entre os processos são as entradas, ferramentas, técnicas e saídas. As saídas são os documentos resultantes do processo.

Exercícios

A partir dos cenários descritos abaixo identificar o escopo, stakeholders e os riscos de cada um deles:

- 1) Uma vídeo locadora pretende informatizar suas operações. É sabido que os clientes podem realizar a reserva, locação e devolução de filmes, além de efetuar os pagamentos em dinheiro e cartão de crédito. Por outro lado, os atendentes dessa vídeo locadora são responsáveis pela manutenção do cadastro do cliente e também informá-lo sobre a ocorrência de atrasos na entrega dos filmes locados. Um filme deve possuir pelo menos um volume para locação. Para um volume deverá ser armazenado o seu código além de estar devidamente relacionado à um filme. É necessário armazenar o código, título, gênero e duração do cada filme.
- 2) Considerando um projeto de um sistema para clínica veterinária, e levando em contas as seguintes informações obtidas através de entrevista com o dono e os funcionários da clínica:
 - a. Um cliente pode possuir muitos animais, mas um animal pertence exclusivamente à um cliente. A clínica precisa de informações a respeito de cada cliente, como nome, endereço e telefone além de saber os animais que o mesmo possui;
 - b. Um animal pertence sempre a uma única espécie;
 - c. É necessário manter informações sobre cada animal, como nome, sexo, idade e a espécie à qual pertence;
 - d. Um animal pode receber muitos tratamentos, mas um tratamento sempre é realizado exclusivamente para um animal;
 - e. Cada tratamento deve possuir ao menos uma consulta. Uma consulta é exclusiva de um tratamento. Cada consulta deve armazenar informações sobre a data que foi realizada, o veterinário responsável e um resumo sobre a consulta.
 - f. Um médico veterinário pode realizar muitas consultas, porém uma consulta deve ser realizada sempre por um único veterinário;
 - g. Em uma consulta pode ocorrer a necessidade de serem marcados exames para o animal, o número de exames possíveis em uma consulta é ilimitado, mas todos devem registrar o nome do exame, a data, o veterinário, o motivo da solicitação do exame e um resumo sobre o resultado.

Importante: Adotar para os dois cenários as seguintes premissas:

- As empresas não possuem informatização;
- Está fora do escopo a especificação de hardware;
- Deve-se analisar alternativas de software livre na implementação.

Metodologias e Modelos

O que é metodologia? **É o estudo dos métodos, ou seja, das etapas a serem seguidas em um determinado processo.** Uma metodologia apresenta como finalidade:

- Captar e analisar as características dos vários métodos disponíveis;
- Avaliar suas capacidades, potencialidades, limitações ou distorções;
- Criticar os pressupostos ou as implicações de sua utilização.

O que é um modelo? **É a representação ou interpretação simplificada da realidade, ou uma interpretação de um fragmento de um sistema, segundo uma estrutura de conceitos.** Um modelo apresenta apenas uma visão ou cenário de um fragmento do todo.

Capability Maturity Model (CMM)

Amplamente difundido, o CMM é um modelo de maturidade que foi desenvolvido pelo SEI (Software Engineering Institute) da Universidade de Carnegie Mellon. O CMM é um modelo baseado em cinco níveis e ajuda na melhoria dos processos de desenvolvimento de software:

- Inicial: os processos de software são pouco definidos, há pouca ou nenhuma documentação e o sucesso dos projetos é dependente de esforços individuais;
- Repetível: os processos básicos de gerenciamento de projetos são estabelecidos para monitorar custos, cronograma e escopo. Deve existir uma disciplina de processos para repetir o sucesso em projetos semelhantes;
- Definido: os processos de software para as atividades de gerenciamento e engenharia é documentado, padronizado e integrado em um padrão para a organização. Todos os projetos seguem e utilizam esse padrão para as etapas de desenvolvimento e manutenção de software;
- Gerenciado: métricas detalhadas de processos de software e qualidade do produto são coletadas. Os projetos são quantitativamente compreendidos e controlados;
- Otimizado: a contínua melhoria dos processos é possível através do retorno quantitativo dos processos, de idéias inovativas e de tecnologias.

Project Management Body of Knowledge (PMBOK®)

O Project Management Body of Knowledge, também conhecido como PMBOK® é um conjunto de práticas em gerência de projetos levantado pelo Project Management Institute (PMI) e constituem a base da metodologia de gerência de projetos do PMI. Estas práticas são compiladas na forma de um guia, chamado de Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos, em resumo o Guia PMBOK consiste no detalhamento dos seguintes tópicos:

Grupo de processos de gerência de projetos:

- Iniciação;
- Planejamento;
- Execução;
- Controle e Monitoramento;
- Encerramento.

Áreas de conhecimento:

- Gerência de Integração do projeto;

- Gerência de Escopo do Projeto;
- Gerência de Tempo do Projeto;
- Gerência de Custos do Projeto;
- Gerência da Qualidade do Projeto;
- Gerência de Recursos Humanos do Projeto;
- Gerência de Comunicações do Projeto;
- Gerência de Riscos do Projeto;
- Gerência de Aquisições do Projeto.

Áreas de especialização:

- O Conjunto de Conhecimentos em Gerenciamento de Projetos;
- Conhecimento, normas e regulamentos da área de aplicação;
- Entendimento do ambiente do projeto;
- Conhecimentos e habilidades de gerenciamento geral;
- Habilidades Inter-pessoais.

Conhecimento em gerência de projeto consiste em:

- Definição de um ciclo de vida;
- Aplicação de 5 processos e 9 áreas de conhecimentos.

A integração dos grupos de processos com as áreas de conhecimento permite determinar a seguinte tabela de processos:

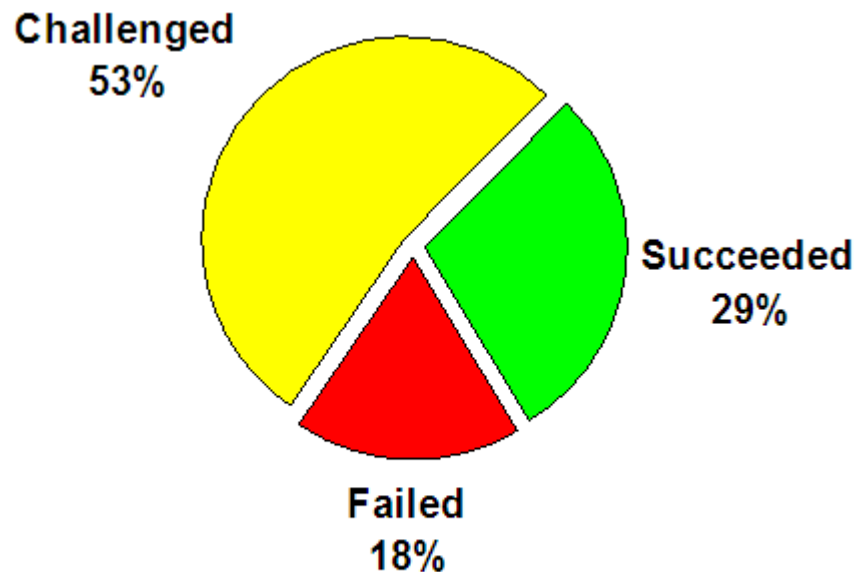
| | Iniciação | Planejamento | Execução | Monitoramento e Controle | Encerramento |
|-------------|-----------|--------------|----------|--------------------------|--------------|
| Integração | ● | ● | ● | ● | ● |
| Escopo | | ● | | ● | |
| Tempo | | ● | | ● | |
| Custos | | ● | | ● | |
| Qualidade | | ● | ● | ● | |
| RH | | ● | ● | ● | |
| Comunicação | | ● | ● | ● | |
| Riscos | | ● | | ● | |
| Aquisições | | ● | ● | ● | ● |

Conhecimento da gestão de projetos baseado no PMBOK:

| | Planejamento | Execução | Controle |
|------------------|--------------------------------------|------------------------------|-----------------------------------|
| Integração | Desenvolvimento Plano do Projeto | Execução do Plano do Projeto | Controle Integrado de Mudanças |
| Escopo | Planejamento do Escopo | | Verificação do Escopo |
| | Detalhamento do Escopo | | Controle de Mudança do Escopo |
| Tempo | Definição das Atividades | | Controle do Cronograma |
| | Seqüenciamento das Atividades | | |
| | Estimativa da Duração das Atividades | | |
| | Desenvolvimento do Cronograma | | |
| Custo | Planejamento dos Recursos | | Controle de Custo |
| | Estimativa dos Custos | | |
| | Orçamentação | | |
| Qualidade | Planejamento da Qualidade | Garantia da Qualidade | Controle da Qualidade |
| Recursos Humanos | Planejamento Organizacional | Desenvolvimento da Equipe | |
| | Montagem da Equipe | | |
| Comunicações | Planejamento das Comunicações | Distribuição das Informações | Relato de Desempenho |
| Risco | Planejamento dos Riscos | | Controle e Monitoração dos Riscos |
| | Identificação dos Riscos | | |
| | Análise Qualitativa dos Riscos | | |
| | Análise Quantitativa dos Riscos | | |
| | Planejamento de Respostas a Riscos | | |
| Aquisições | Planejamento das Aquisições | Pedido de Propostas | |
| | Preparação das Aquisições | Seleção de Fornecedores | |
| | | Administração dos Contratos | |

Gestão de Riscos

Nos últimos anos, vem crescendo muito a adoção do gerenciamento de projetos como uma maneira das organizações atingirem os seus objetivos. Porém, o fato de se usar o gerenciamento de projetos por si só não garante o sucesso dos projetos. Garantir o sucesso de um projeto, ou seja, entregar dentro do prazo e do orçamento e com satisfação do cliente, continua sendo extremamente difícil. O causa principal de tantas dificuldades são os riscos envolvidos no empreendimento.



Copyright © 2006 The Standish Group International, Inc.,

Segundo uma pesquisa realizada pelo The Standish Group (<http://www.standishgroup.com>) que publica um relatório chamado The Chaos Report e mostra o cenário da indústria de software: - aproximadamente 18% dos projetos são cancelados por atrasos e orçamento estourados e 53% dos projetos estouram o orçamento e/ou o prazo enquanto apenas 29% de todos os projetos de TI atingem seus objetivos dentro de prazo e custo estimados.

É impossível eliminar os riscos por completo, porém é necessário um esforço para reduzir a probabilidade deles ocorrerem, ou pelo menos, para reduzir as suas consequências negativas. O objetivo da gestão de riscos é identificar, analisar, avaliar e tratar os riscos que podem afetar o projeto.

O estágio de identificação dos riscos é de responsabilidade de todos os envolvidos no projeto, desde os diretores e donos da organização até os membros da equipe. Ao se identificar os riscos elimina-se o fator surpresa, aumentando assim o sucesso no tratamento das consequências. O resultado deste estágio consiste em uma descrição detalhada de todos os riscos identificados.

Na análise e avaliação, os riscos identificados são elencados conforme a probabilidade de ocorrência dos eventos associados ao risco e o impacto que terão caso ocorram. Como, na maioria dos projetos, os riscos são numerosos é necessários priorizá-los considerando as três principais dimensões: escopo (qualidade), prazos (tempo) e recursos (custo).

No tratamento dos riscos serão implantadas medidas para evitar ou reduzir os riscos. Entre elas estão incluídas os planos de contingência, as ações para evitar os riscos, as medidas para minimizar os riscos e a aceitação dos riscos.

O produto final dos quatro estágios consiste em um registro priorizado dos riscos, com atribuição de responsabilidades e definição de planos de ação.

Ferramentas para a Gestão de Projetos

As ferramentas para gerenciamento de projetos tem o objetivo de auxiliar e prover de forma rápida e eficiente as informações necessárias para o correto controle e acompanhamento do trabalho. Ao escolhermos uma ferramenta é importante observar se a mesma apresenta as seguintes funcionalidades:

- Controle das tarefas, recursos e custos do projeto;
- Controle do progresso do projeto;
- Gráficos comparativos de acompanhamento dos projetos.

O site <http://www.project-management-software.org/> pode ser utilizado como fonte de referência para identificar as principais ferramentas para gerenciamento de projetos disponíveis no mercado, dentre as quais podemos destacar:

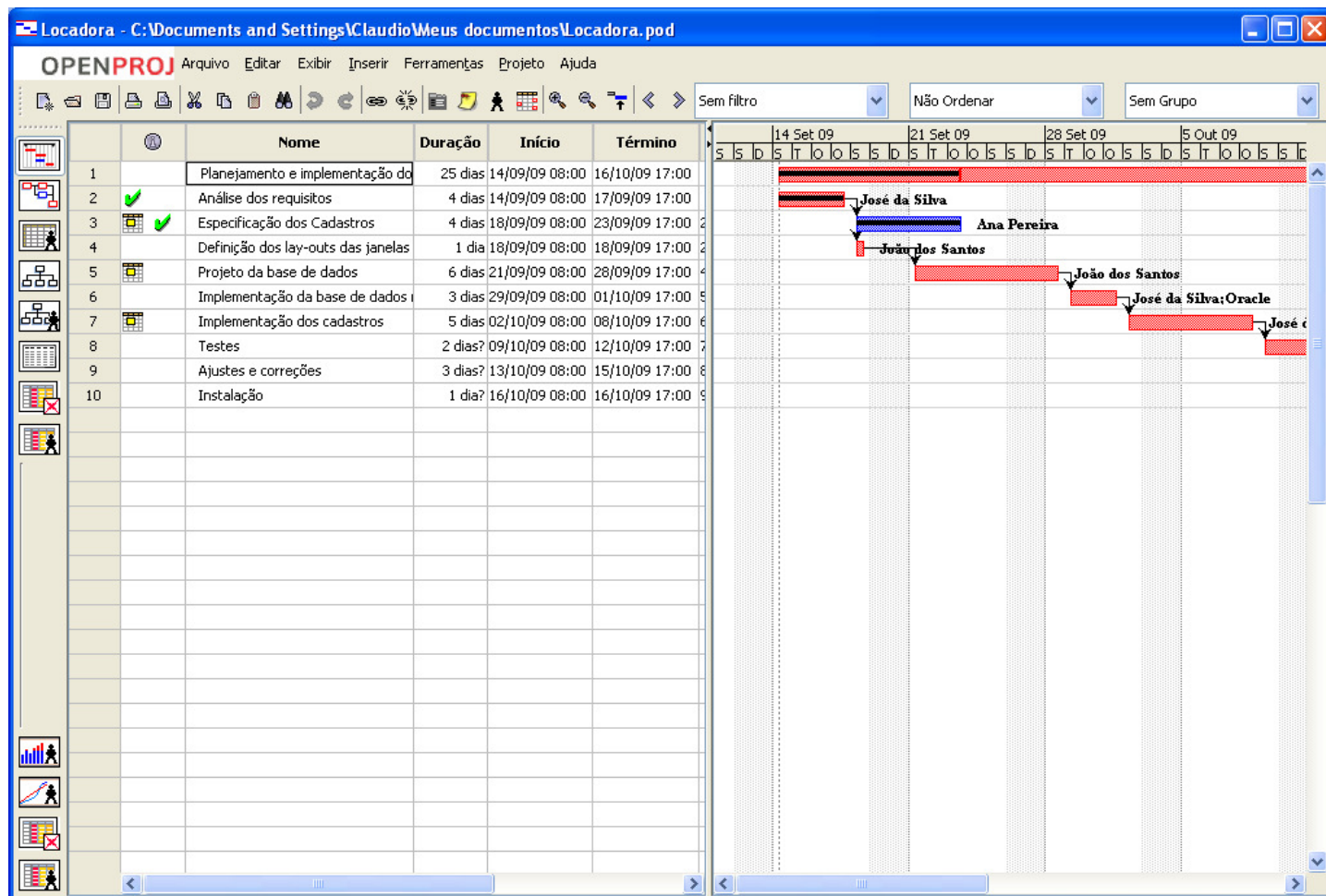
- **Microsoft Project** (<http://office.microsoft.com/project/>): Mantenha-se informado, controlando o trabalho, a agenda e as finanças do projeto e transmitindo os dados associados de forma eficiente;
- **OpenProj** (<http://openproj.org/>): Gerenciador de projetos desenvolvido como alternativa ao Microsoft Project. Com código aberto, o programa oferece várias opções para a inclusão e administração de todas as atividades co-relacionadas ao desenvolvimento das atividades, apresentando os resultados, em especial, sob o formato de Gráficos de Gantt;
- **dotProject** (<http://www.dotproject.net/>): É um sistema de gerência de projetos em software livre de fácil utilização, com um conjunto de funcionalidades e características que o tornam indicado para implementação em ambientes corporativos, pois atende a diversas necessidades de gerentes e escritórios de projetos.
- **ProjectPier** (<http://www.projectpier.org/>): Realizado em PHP para gerenciar tarefas, projetos e equipes num entorno muito intuitivo. Facilita a comunicação entre os membros da equipe e está programado para facilitar a criação de novos módulos.



O OpenProj é um programa de gestão de projetos, com código aberto e distribuição gratuita, sendo uma alternativa em substituição ao Microsoft Project e outros programas similares. É uma poderosa ferramenta, apresentando todas as funções que se pode esperar de um programa desta categoria.

Guia do Usuário

Criação de um novo projeto: Após escolher no menu as opções Arquivo e Novo Projeto a seguinte janela será exibida:



Após entrar na opção do Gráfico de Gantt torna-se necessário apenas cadastrar as tarefas, utilize a tabela abaixo como exemplo, e o gráfico será automaticamente construído.

Tarefas:

| Nº | Nome | Duração | Início | Término | Predecessoras | Recursos |
|----|---|---------|----------------|----------------|---------------|---|
| 1 | Planejamento e implementação do processo de locação | 25 dias | 14/09/09 08:00 | 16/10/09 17:00 | | |
| 2 | Análise dos requisitos | 4 dias | 14/09/09 08:00 | 17/09/09 17:00 | | José da Silva |
| 3 | Especificação dos Cadastros | 4 dias | 18/09/09 08:00 | 23/09/09 17:00 | 2 | Ana Pereira |
| 4 | Definição dos lay-outs das janelas | 1 dia | 18/09/09 08:00 | 18/09/09 17:00 | 2 | João dos Santos |
| 5 | Projeto da base de dados | 6 dias | 21/09/09 08:00 | 28/09/09 17:00 | 4 | João dos Santos |
| 6 | Implementação da base de dados no Oracle | 3 dias | 29/09/09 08:00 | 01/10/09 17:00 | 5 | José da Silva; Oracle |
| 7 | Implementação dos cadastros | 5 dias | 02/10/09 08:00 | 08/10/09 17:00 | 6 | José da Silva; Ana Pereira; João dos Santos; Java; Oracle; Provedor Internet |
| 8 | Testes | 2 dias | 09/10/09 | 12/10/09 | 7 | Ana Pereira; |

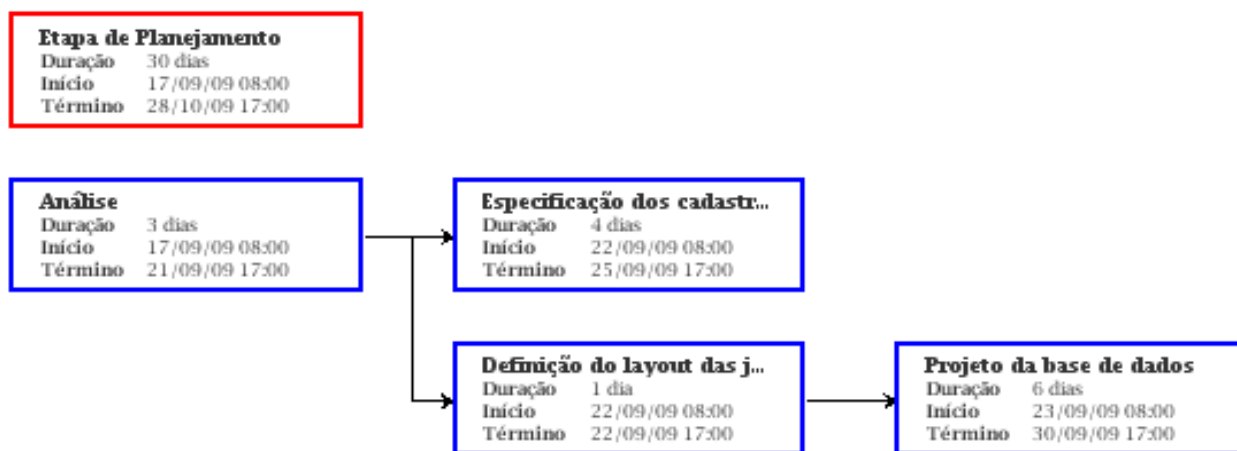
| | | | | | | |
|----|---------------------|--------|-------------------|-------------------|---|--|
| | | | 08:00 | 17:00 | | Oracle; Java |
| 9 | Ajustes e correções | 3 dias | 13/10/09 08:00 | 15/10/09 17:00 | 8 | José da Silva; Oracle; Java |
| 10 | Instalação | 1 dia | 16/10/09 08:00 | 16/10/09 17:00 | 9 | João dos Santos; Provedor Internet |

Quando se realiza o cadastro das tarefas é possível acessar os demais gráficos que são providos pelo sistema, ou seja:

Gráfico de Gantt: É um gráfico de barras que ilustra o cronograma de um projeto. O gráfico ilustra as datas de início e fim de níveis sumários (**Estrutura Analítica de Projeto - EAP**) e atividades que compõem o cronograma. Alguns Gráficos de Gantt mostram os relacionamentos de dependência entre atividades, isto é, a rede de precedência. Também podem ser usados para mostrar a situação atual do desenvolvimento de um projeto.



Diagrama de Rede: Também conhecido como gráfico de Pert, possibilita uma melhor visualização da sequência de tarefas, a estimativa de tempo e as tarefas dependentes, sejam elas sucessoras ou antecessoras.



Estrutura Analítica de Projetos (EAP): Também conhecida como **work breakdown structure (WBS)** é uma ferramenta de decomposição do trabalho do projeto em partes manejáveis. Consiste em uma estrutura hierárquica, partindo de tarefas e entregáveis mais gerais para os mais específicos. O objetivo da EAP é identificar elementos terminais, ou seja, os itens reais a serem feitos em um projeto. Desta forma, a EAP serve como base para a maior parte do planejamento do projeto.

| | | | | |
|--|---|---|---|--|
| Etapla de Planejam... Custo R\$ 0,00 Orçam... | Análise Custo R\$ 1600,00 Orçam... | Especificação dos ... Custo R\$ 1920,00 Orçam... | Definição do layo... Custo R\$ 400,00 Orçam... | Projeto da base de ... Custo R\$ 2400,00 Orçam... |
|--|---|---|---|--|

Resource Breakdown Structure (RBS): Do mesmo modo que é necessário detalhar as tarefas a serem efetuadas deve-se também detalhar os recursos que serão utilizados. Na RBS as listas estão organizadas com base na função e hierarquia.

| | |
|----------------------|-------------|
| José da Silva | |
| Custo | R\$ 4400,00 |
| Orçame... | R\$ 0,00 |

| | |
|-------------------------|-------------|
| Maria dos Santos | |
| Custo | R\$ 1920,00 |
| Orçame... | R\$ 0,00 |

| | |
|------------------------|----------|
| Paulo Gonçalves | |
| Custo | R\$ 0,00 |
| Orçame... | R\$ 0,00 |

Orçamento

Um dos fatores mais complexos na elaboração de um projeto é realizar o levantamento dos custos envolvidos. Em linhas gerais, os seguintes aspectos fazem diferença dentro de um projeto de software:

- A **dimensão** do produto, ou seja, o número absoluto de requisitos. Quando se está na fase de projeto representa o número de elementos projetados. A dimensão final de um produto é mostrada, por exemplo, em linhas de código ou classes existentes no programa.
- O **esforço** requerido para o desenvolvimento é medido em homens/horas. O projeto deve estimar quantos homens/horas é necessário para se construir o produto dentro do prazo determinado.
- A **qualidade** do produto é medida em função da taxa de erros encontrados e solucionados. Tais erros podem ser achados em qualquer uma das fases do projeto, não apenas na fase de teste.
- O **cronograma** descreve ao longo do tempo como as atividades planejadas serão executadas, indicando sempre quando cada atividade começa e termina.

Além dos aspectos já relacionados os seguintes itens devem ser considerados nas estimativas de custos:

- Custos relativos ao esforço empregado pelos membros da equipe no projeto;
- Custos de hardware e software, incluindo as necessidades de manutenção e atualização;
- Outros custos relacionados ao projeto, tais como custos de viagens e treinamentos necessários ao projeto;
- Despesas gerais, incluindo gastos com água, luz, telefone, pessoal de apoio administrativo e equipe de suporte entre outras.

Para a maioria dos projetos, o custo dominante é o que se refere ao esforço empregado, juntamente com as despesas gerais. De modo geral, os custos relacionados com as despesas gerais correspondem a um valor equivalente aos custos relativos ao esforço empregado pelos membros da equipe no projeto. Assim, para efeitos de estimativas de custos é possível considerar esses dois itens como sendo um único, computado em dobro.

Os custos de hardware e software, ainda que menos influentes, não devem ser desconsiderados, sob pena de provocarem prejuízos para o projeto. Uma forma de tratar esses custos é considerar a depreciação com base na vida útil do equipamento ou da versão do software utilizada e as necessidades de manutenção e atualização.

Quando o custo do projeto estiver sendo calculado como parte de uma proposta para o cliente, então será preciso definir o preço cotado. Uma abordagem para definição do preço pode ser considerá-lo como o custo total do projeto mais o lucro. Entretanto, a relação entre o custo do projeto e o preço cotado para o cliente, normalmente, não é tão simples assim.

Qualidade de Software

Software de qualidade é fácil de usar, funciona corretamente, é de fácil manutenção e mantém a integridade dos dados em falhas do ambiente ou outras fora do seu controle. No entanto, em sua grande maioria, os softwares requerem conhecimentos técnicos especiais na sua utilização, são difíceis de alterar para modificar função existente ou implementar novos processos ou facilidades e, para desespero de seus usuários, apresentam falhas sem aviso prévio e não preservam a integridade dos dados.

Pouco se fala a respeito dos custos resultantes dos defeitos ou erros provocados por falha de softwares, tanto para produtores quanto para usuários. O bug do milênio, causado pelos erros que os computadores teriam ao confundir o ano 2000 com o ano 1900, consumiu bilhões de dólares para evitar um colapso mundial. Bancos poderiam perder milhões, clientes veriam o saldo de suas contas sumir de repente, telefones poderiam não funcionar, aviões poderiam ter sua rota desviada, e outros problemas bem mais graves poderiam ocorrer. Este é um exemplo recente e dimensiona o quanto dependemos das máquinas e de seus softwares. Com o uso maciço das tecnologias de informação e comunicação em todos os níveis da atividade humana, os problemas de qualidade de software tendem a adquirir a cada dia maior importância.

A Qualidade de Software é uma área de conhecimento da Engenharia de Software que objetiva garantir a qualidade do software através da definição e normatização de processos de desenvolvimento.

Apesar dos modelos aplicados na garantia da qualidade de software atuarem principalmente no processo, o principal objetivo é garantir um produto final que satisfaça às expectativas do cliente, dentro daquilo que foi acordado inicialmente.

O que é Qualidade?

A definição possui uma grande diversidade de interpretações:

“A qualidade é relativa. O que é qualidade para uma pessoa pode ser falta de qualidade para outra”. (G. Weinberg)

“Qualidade é a conformidade do produto com os requisitos ou especificações estabelecidos”. (Crosby)

“Qualidade é tudo aquilo que melhora o produto do ponto de vista do cliente”. (Deming)

A norma internacional ISO/IEC 9126, publicada em 1991 e que na versão brasileira de agosto de 1996 recebeu o número NBR 13596, define qualidade de software como:

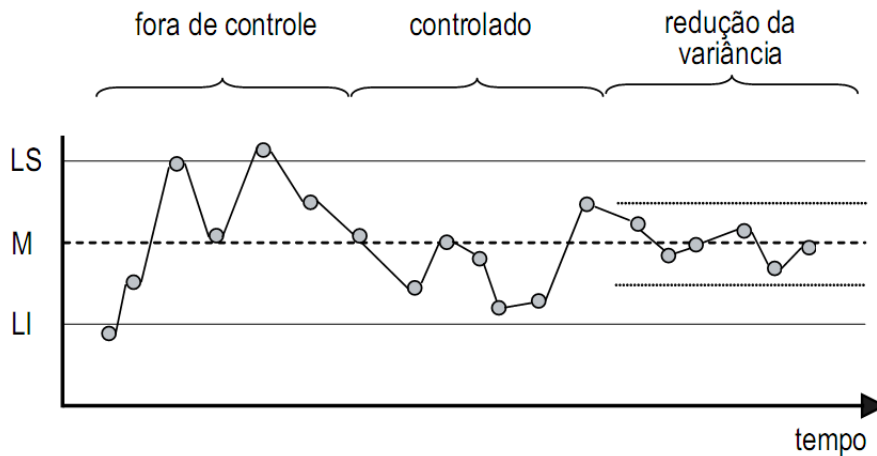
“A totalidade de características de um produto de software que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas”.

Necessidades explícitas são as condições e objetivos propostos por aqueles que produzem o software. Desta forma, são fatores relativos à qualidade do processo de desenvolvimento do produto e são percebidos somente pelas pessoas que trabalharam no seu desenvolvimento. As necessidades implícitas são necessidades subjetivas dos usuários (inclusive operadores, destinatários dos resultados do software e os mantenedores do produto), são também chamadas de fatores externos e podem ser percebidas tanto pelos desenvolvedores quanto pelos usuários. As necessidades implícitas são também chamadas de qualidade em uso e devem permitir aos usuários atingir metas com efetividade, produtividade, segurança e satisfação em um contexto de uso especificado.

Breve Histórico

Os egípcios já utilizavam padronização há +- 4 mil anos, conseguindo precisão da ordem de 0,05% na construção das pirâmides.

Em 1920 com a Revolução Industrial surge o Controle Estatístico da Produção.



LS = Limite superior; M = Média; LI = Limite inferior.

Em 1940 surgem vários organismos ligados a qualidade tais como:

- ASQC - American Society for Quality Control;
- ABNT - Associação Brasileira de Normas Técnicas;
- ISO - International Standardization Organization.

Crise de Software

A Engenharia de Software, como aceita por muitos, iniciou-se em 1968, quando aproximadamente 50 especialistas em computação de 11 países se reuniram na Alemanha, para discutir os problemas de desenvolvimento de software da época. Durante a conferência houve uma grande quantidade de debate sobre o tema que os participantes escolheram chamar de “Software Crisis” ou “Software Gap”.

Entre os principais problemas, relacionados a este tema, que foram descritos durante a conferência estavam:

- Projetos realizados acima do orçamento;
- Projetos finalizados acima do tempo esperado;
- Produtos de software de baixa qualidade;
- Produtos de software sem atender aos requisitos do cliente;
- Projetos ingerenciáveis e com código de difícil manutenção.

Após quase quatro décadas a Engenharia de Software evoluiu bastante. Durante todo este tempo foram desenvolvidos muitos métodos, técnicas e teorias com o objetivo de resolver ou amenizar os problemas reportados durante a conferência. Grande parte destas evoluções veio como resultado do aprendizado produzido em projetos reais e fizeram com que a Engenharia de Software amadurecesse como área da Ciência da Computação.

Qualidade e Requisitos

A preocupação com a qualidade na produção de software inicialmente centrou foco na qualidade do produto, porém, ao longo do tempo, esta visão evoluiu e atualmente a preocupação com qualidade envolve tanto o processo de produção, o ciclo de desenvolvimento do software, quanto o resultado final que consiste no produto ou software gerado.

O ciclo de desenvolvimento de um produto de software, de acordo com as metodologias clássicas de desenvolvimento, tem início pelo processo de requisitos. Nesta fase são determinados os requisitos que o software em construção deverá atender. Este processo gera um conjunto de artefatos que constituem a base para o registro e o acompanhamento da evolução dos requisitos ao longo do ciclo de desenvolvimento, possibilitando um efetivo gerenciamento de requisitos.

Estudos recentes indicam que aproximadamente 50% das falhas detectadas na fase de testes são causadas por defeitos em requisitos, e que a correção dessas falhas implica em re-trabalho muitas vezes evitável. Os defeitos em requisitos, isoladamente, é a falha com maior frequência em projetos de software.

Para minimizar esse problema deve-se adotar métricas que irão fornecer subsídios importantes para o gerenciamento do projeto, e podem ser aplicadas em diferentes etapas do processo de desenvolvimento. É possível desenvolver casos de teste ainda no processo de requisitos; dificuldades na geração indicam problemas na definição dos requisitos. As inspeções ajudam a encontrar defeitos em artefatos, antes que se passe à fase seguinte do processo; no cerne do processo de inspeção estão a identificação das informações a serem checadas e a técnica a ser utilizada para identificar defeitos nessas informações.

Os problemas de definir e avaliar qualidade ainda não são completamente resolvidos apenas pelos requisitos. Mas já é um grande passo para a solução relacionar os dois conceitos:

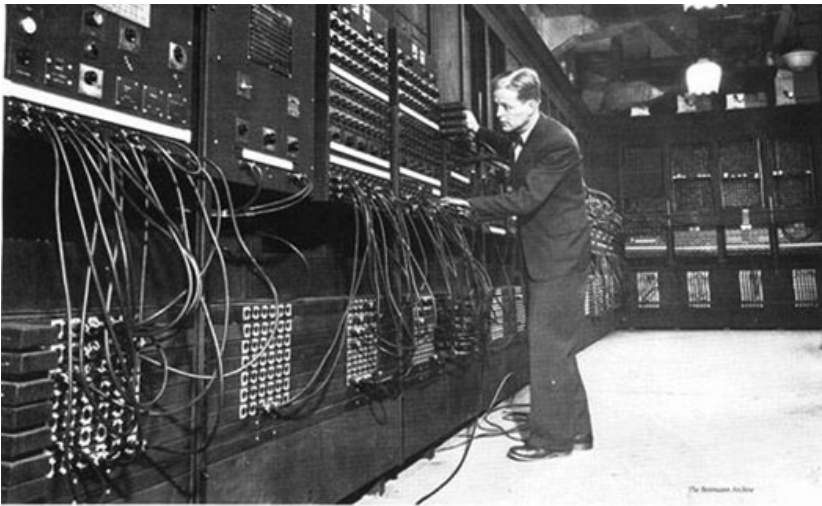
$$Qualidade = f(requisitos)$$

$$Qualidade = f(observado - especificado)$$

$$Qualidade = f(observado - especificado + \epsilon)$$

onde ϵ representa um erro de medição que não é possível controlar.

Bugs



Com o surgimento do primeiro computador...

9/9


0800 Antan started
 1000 " stopped - antan ✓

1300 (032) MP-MC { 1.2700 9.037 847 025
 2.130476415 (033) PRO 2 2.130476415 9.037 846 995 connect
 connect 2.130676415 4.615925059(-2)

Relays 6-2 in 033 failed special speed test
 in relay .. 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)
 1525 Started Multi-Adder Test.

1545  Relay #70 Panel F
 (moth) in relay.

First actual case of bug being found.

1630 Antan started.
 1700 closed down.

Relay 2145
 Relay 3376

... ocorrem os primeiros "Bugs".

Qualidade e Bugs

A verificação da qualidade de um software depende de se decidir o que significa qualidade!

Não é um assunto que possa ser tratado com dogmas: “*Não cometerás erros de programação*”

É preciso considerar diversos fatores que afetam a construção do produto:

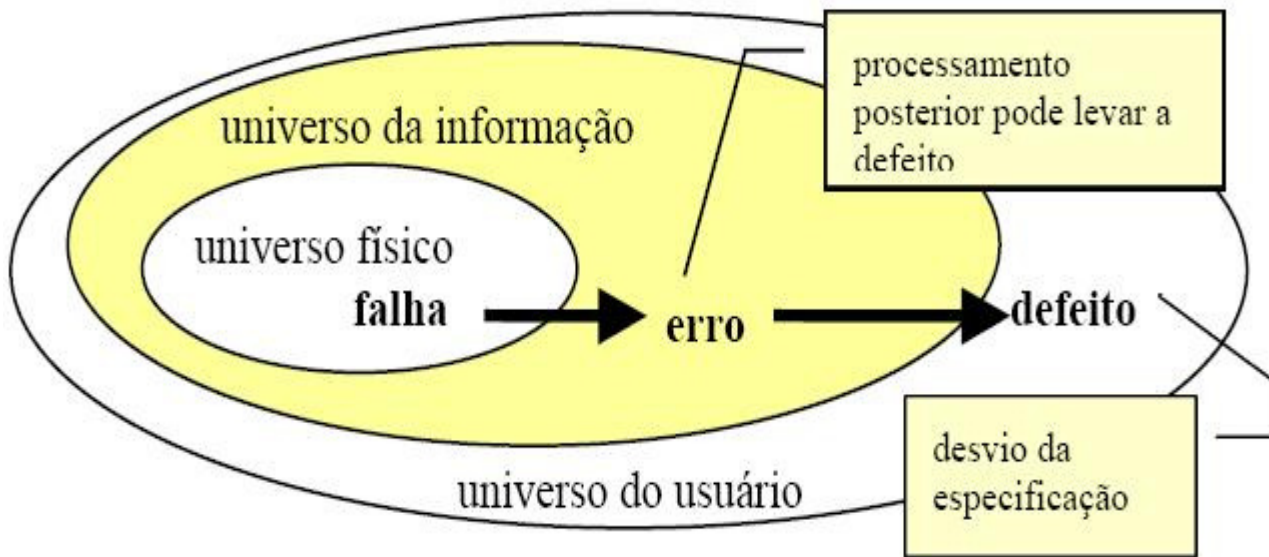
- Tamanho e complexidade do software;
- Número de pessoas envolvidas no projeto;
- Ferramentas utilizadas;
- Custos associados à existência de erros;
- Custos associados à detecção e à remoção de erros.

Falha, Erro ou Defeito?

Falha (*fault*) ocorre no nível mais baixo. Uma falha poderá gerar um erro.

Erro (*error*) é a representação da falha no universo da informação.

Defeito (*failure*) é um desvio na especificação, e ocorre em consequência de um erro.



Como Obter Qualidade?

Adoção de metodologias - apresenta como finalidade captar e analisar as características dos vários métodos disponíveis, avaliar suas capacidades, potencialidades, limitações ou distorções e criticar os pressupostos ou as implicações de sua utilização.



É possível medir a qualidade de um software?

O principal problema com que se defronta a engenharia de software é a dificuldade de se medir a qualidade de software. A qualidade de um dispositivo mecânico é frequentemente medida em termos de tempo médio entre suas falhas, que é uma medida da capacidade de o dispositivo suportar desgaste. O software não se desgasta, portanto tal método de medição de qualidade não pode ser aproveitado.

O modelo proposto pela ISO/IEC 9126 (NBR 13596) tem por objetivo servir de referência básica na avaliação de produto de software. Além de ter força de norma internacional, ela cobre os aspectos mais importantes para qualquer produto de software.

| CARACTERÍSTICAS | SUB-CARACTERÍSTICAS | SIGNIFICADO |
|---|---------------------------------------|--|
| Funcionalidade O conjunto de funções satisfaz as necessidades explícitas e implícitas para a finalidade a que se destina o produto? | Adequação | Propõe-se a fazer o que é apropriado? |
| | Acurácia | Gera resultados corretos ou conforme acordados? |
| | Interoperabilidade | É capaz de interagir com os sistemas especificados? |
| | Segurança de acesso | Evita acesso não autorizado, acidental ou deliberado a programas e dados? |
| | Conformidade | Está de acordo com normas e convenções previstas em leis e descrições similares? |
| Confiabilidade O desempenho se mantém ao longo do tempo e em condições estabelecidas? | Maturidade | Com que frequência apresenta falhas? |
| | Tolerância a falhas | Ocorrendo falhas como ele reage? |
| | Recuperabilidade | É capaz de recuperar dados após uma falha? |
| Usabilidade É fácil utilizar o software? | Inteligibilidade | É fácil entender os conceitos utilizados? |
| | Apreensibilidade | É fácil aprender a usar? |
| | Operacionalidade | É fácil de operar e controlar a operação? |
| Eficiência Os recursos e os tempos utilizados são compatíveis com o nível de desempenho requerido para o produto? | Comportamento em relação ao tempo | Qual é o tempo de resposta e de processamento? |
| | Comportamento em relação aos recursos | Quanto recurso utiliza? |
| Manutenibilidade Há facilidade para correções, atualizações e alterações? | Analísabilidade | É fácil encontrar uma falha quando ocorre? |
| | Modificabilidade | É fácil modificar e remover defeitos? |
| | Estabilidade | Há grandes riscos de <i>bugs</i> quando se faz alterações? |
| | Testabilidade | É fácil testar quando se faz alterações? |
| Portabilidade É possível utilizar o produto em diversas plataformas com pequeno esforço de adaptação? | Adaptabilidade | É fácil adaptar a outros ambientes sem aplicar outras ações ou meios além dos fornecidos para esta finalidade no software considerado? |
| | Capacidade para ser instalado | É fácil instalar em outros ambientes? |
| | Capacidade para substituir | É fácil substituir por outro software? |
| | Conformidade | Está de acordo com padrões ou convenções de portabilidade? |

A norma ISO/IEC 12119, publicada em 1994, define que cada pacote de software tenha uma descrição do produto e uma documentação do usuário, e estabelece alguns requisitos de qualidade:

- a) Descrição do produto compreensível e completa para ajudar o usuário ou comprador em potencial na avaliação da adequação do produto a sua realidade e fornecer informações comerciais;
- b) Documentação do usuário de fácil compreensão, permitindo uma visão geral do produto e de todas as suas funções, identificando conhecimento necessário para uso da aplicação;
- c) Identificação do tipo de interface com o usuário: interface gráfica, web, linha de comando, menu de comandos e janelas entre outras;
- d) Instruções detalhadas sobre como instalar o produto, caso a instalação possa ser conduzida pelo usuário;
- e) Possibilidade de verificar se a instalação foi bem sucedida;
- f) Especificação de valores-limite para quantidade de registros e dados de entrada, como, por exemplo, precisão de casa decimal;
- g) Operação normal, mesmo quando os dados informados estão fora dos limites especificados;
- h) Consistência de vocabulário entre as mensagens e a documentação;
- i) Função de auxílio (ajuda) sensível ao contexto;
- j) Mensagens de erro com informações necessárias para solucionar o problema;
- k) Diferenciação de tipos de mensagem: confirmação, consulta, advertência e erro;
- l) Clareza e padronização nos formatos de telas de entrada, relatórios e outras entradas e saídas;
- m) Capacidade de reverter funções de efeito drástico;
- n) Capacidade de recuperar dados após uma falha de hardware ou software, queda de energia ou erro fatal;
- o) Alertas claros para o usuário das conseqüências de uma determinada confirmação;
- p) Identificação dos arquivos utilizados pelo programa;
- q) Identificação da função do programa que está sendo executada no momento;
- r) Capacidade de interromper um processamento demorado.

Teste de Software

O teste do software é a investigação do software a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar. Isso inclui o processo de utilizar o produto para encontrar seus defeitos.

Fases

Uma prática comum é testar o software após uma funcionalidade ser desenvolvida, e antes dela ser implantada no cliente, por um grupo de profissionais diferente da implementação. Essa prática pode resultar na fase de teste ser usada para compensar atrasos do projeto, comprometendo o tempo devotado ao teste. Outra prática é começar o teste no mesmo momento que o projeto, num processo contínuo até o fim do projeto.

Teste de unidade: também conhecido como teste unitário ou teste de módulo, é a fase em que se testam as menores unidades de software desenvolvidas. O universo alvo desse tipo de teste são classes, subrotinas ou mesmo pequenos trechos de código. Assim, o objetivo é o de encontrar falhas de funcionamento dentro de uma pequena parte do sistema funcionando independentemente do todo.

Teste de integração: o objetivo é encontrar falhas provenientes da integração interna dos componentes de um sistema. Geralmente os tipos de falhas encontradas estão relacionada à troca de dados entre os componentes. Por exemplo, um componente A pode estar aguardando o retorno de um valor X ao executar um método do componente B. Porém, B pode retornar um valor Y, gerando uma falha. Não faz parte do escopo dessa fase de teste o tratamento de interfaces com outros sistemas (integração entre sistemas). Essas interfaces são testadas na fase de teste de sistema, apesar de, a critério do gerente de projeto, estas interfaces podem ser testadas mesmo antes de o sistema estar plenamente construído.

Teste de sistema: nesta fase, o objetivo é executar o sistema sob ponto de vista de seu usuário final, varrendo as funcionalidades em busca de falhas em relação aos objetivos originais. Os testes são executados em condições similares de ambiente, interfaces e base de dados em relação àquelas que o usuário utilizará no futuro.

Teste de aceitação: geralmente são realizados por um grupo restrito de usuários finais do sistema, que simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado. Consiste em um teste formal conduzido para determinar se um sistema satisfaz ou não seus critérios de aceitação e para permitir ao cliente determinar se aceita ou não o sistema. Nesta fase é possível incluir testes funcionais, de configuração, de recuperação de falhas, de segurança e de desempenho.

Teste de operação: o teste é conduzido pelos administradores do ambiente final em que o sistema ou software entrará em ambiente produtivo. Vale ressaltar que essa fase é aplicável somente a sistemas de informação próprios de uma organização, cujo acesso pode ser feito interna ou externamente a essa organização. Nessa fase de teste devem ser feitas simulações para garantir que a entrada em produção do sistema será bem sucedida. Envolve testes de instalação, simulações com cópia de segurança dos bancos de dados. Em alguns casos um sistema entrará em produção para substituir outro e é necessário garantir que o novo sistema continuará garantindo o suporte ao negócio.

Artefatos

O processo de teste de software pode produzir diversos artefatos. O caso de teste geralmente consiste de uma referência a um identificador ou requisito de uma especificação, pré-condições, eventos, uma série de passos a se seguir, uma entrada de dados, uma saída de dados, resultado esperado e resultado obtido. A série de passos pode estar contida em um procedimento separado, para que possa ser compartilhada por mais de um caso de teste.

Abaixo é mostrado um exemplo de documento de caso de teste:

| Caso de Teste | |
|----------------------|---|
| Nome: | Nome utilizado para identificar de modo único um caso de teste. |
| Pré-condições: | Deve indicar a condição inicial do sistema para permitir a execução do caso de teste. |
| Procedimentos: | Consiste na relação dos passos necessários para se chegar ao resultado esperado. |
| Resultado Esperado: | Resposta esperada pelo sistema, caso o caso de teste seja bem sucedido. |
| Dados de Entrada: | Dados que são necessário para a realização do caso de teste. |
| Ambiente: | Configuração do ambiente que é necessária para a execução do caso de teste. |
| Iteração: | Indica quantas vezes o referido caso de teste foi executado. |
| Implementação: | Manual ou automatizada. |

Um script de teste é a combinação de um caso de teste, um procedimento de teste e os dados do teste. Uma coleção de casos de teste é chamada de suite de teste. Geralmente, ela também contém instruções detalhadas ou objetivos para cada coleção de casos de teste, além de uma seção para descrição da configuração do sistema usado.

O plano de teste é um documento com uma abordagem sistemática para o teste de sistemas como hardware ou software. Ele geralmente consiste numa modelagem detalhada do fluxo de trabalho durante o processo.