Projeto de Software: O Projeto de Software se inicia logo após a Análise de Requisitos ter sido levantada, analisada e modelada.Seu principal objetivo é: definir uma estrutura que possa ser implementada em um produto de software e que atenda aos requisitos especificados para ele na análise. Na Análise de Requisitos é levantado “o que” será implementado, no Projeto de Software é definido “como” será sua construção.

O projeto de software é um processo que possui as seguintes atividades: estrutura de dados, arquitetural, componentes e interface.Ele cria uma representação que fornece detalhes sobre: a arquitetura do software, as estruturas de dados,interfaces e componentes fundamentais para implementar um sistema.

Segundo Pressman:os modelos de requisitos representam os requisitos dos clientes,enquanto,os modelos de projeto oferecem uma especificação concreta para a construção do software.Pressman também fala que: o modelo de requisitos é usado para preencher a lacuna entre uma representação sistêmica que descreve o sistema (como o usuário imagina) como um todo ou a funcionalidade de negócio,o modelo de projeto de software - é usado para descrever a arquitetura da aplicação de software, a interface do usuário e a estrutura no nível de componentes.

Seguindo sua visão na fase de projeto,temos:O processo de projeto passa de uma visão macro do software para uma visão mais estreita que define os detalhes necessários para implementar um sistema. O processo começará concentrando-se na arquitetura e depois serão definidos subsistemas;são estabelecidos mecanismos de comunicação entre os subsistemas;identificação de componentes; e é desenvolvida uma descrição detalhada de cada componente. Além disso,serão projetadas interfaces externas,internas e para o usuário.

Segundo Falbo:o objetivo da fase de projeto é definir e especificar uma solução que será implementada para um problema.Assim,podemos dizer que é uma fase em que tomamos muitas decisões,pois temos ao nosso alcance muitas soluções que podem ser possíveis.O mesmo fala que projeto é um processo de refinamento. Assim,de maneira geral,ele descreve que um projeto deve:

- Considerar abordagens alternativas com base nos requisitos do problema;
- Restrições e conceitos de projeto;
- Ser rastreável a sua especificação;
- Não “reinventar a roda”,isto é,reutilizar soluções;
- Exibir uniformidade (estilo) e integração (interfaces bem definidas entre componentes da coisa a ser construída;

- Ser estruturado para acomodar mudanças;
- Ser passível de avaliação da qualidade;
- Ser revisado para minimizar erros;

E também segundo o mesmo, em geral,um modelo de projeto deve:

- Prover uma visão da totalidade do que deve ser construído;
- Decompor o todo em partes e prover diferentes visões;
- Refinar e descrever com mais detalhes cada parte ou visão do que deve ser construído,de modo a prover orientação para a construção de cada detalhe;

Sobre Planejamento de projetos:Uma coisa é exigir dos engenheiros de software estimativas de prazos,e cobrar o cumprimento dos prazos prometidos. Clientes e gerentes podem e devem fazê-lo.Outra coisa é os pressionar para que façam promessas que não podem ser cumpridas.Uma frase muito comum desta cultura é: “Não me interessa como você vai fazer,desde que entregue no prazo!”.Na realidade,o cliente ou gerente deve,no seu próprio interesse, ter algum meio de checar se o cronograma e orçamento propostos são realistas; se preciso,recorrendo aos serviços uma terceira parte.A cultura do prazo político é ruim para todos.Para os desenvolvedores significará:estresse e má qualidade de vida.Enquanto para os gerentes:perda de credibilidade e prejuízos.E para os clientes:produtos de má qualidade e mais caros do que deveriam.Ainda por cima, entregues fora do prazo

Implementação de Software: A implementação demanda grande parte do tempo no processo de desenvolvimento de um software, por ser uma das atividades mais trabalhosas e exigir grandes habilidades do profissional da área de informática.Conforme Sommerville a implementação de software é o processo de “conversão” de uma especificação do sistema em um sistema executável. A fase de implementação sempre começa quando a fase de projeto tiver sido encerrada.

Na Implementação de Software serão detalhados os componentes que foram descritos na fase de projeto,como:os códigos-fonte usados na linguagem de programação,lembrando que devem ser conforme as tecnologias que foram informadas.Na implementação é definida a linguagem de programação,que

podem ser:Java,C#,PHP,C++ ou qualquer outra, que possa desenvolver o que foi modelado na fase de projeto.O código é escrito por programadores e é muito importante que haja uma organização na escrita das instruções.Toda essa organização pode ser definida com a criação de padrões a serem seguidos pela equipe de programação.Exemplo de padrões que podem ser definidos: declaração de nomes de variáveis,formato de cabeçalhos, comentários dos códigos e como documentar o código.Nessa fase, construímos e codificamos os programas e os módulos que envolvem o software são integrados.

Pressman afirma que:os problemas de confiabilidade de um software podem quase sempre serem associados a defeitos de projeto ou de implementação.Ele afirma que todas as falhas que um software possuem estão associadas aos problemas na fase de projeto e de implementação.Segundo o autor:as falhas são de ambos, tanto do cliente,quanto de quem deselvolveu o software.

A fase de implementação é uma maneira de formalizar ou mostrar,utilizando uma linguagem de programação,das análises e dos modelos levantados nas fases de requisito e de projeto,e assim gerando um sistema que possa executar as tarefas que foram descritas pelo usuário.O mesmo afirma que: podemos definir a implementação como sendo a fase de programação que transformará o projeto em um sistema com forma computacional mais palpável pelo usuário.Na fase de implementação,também podem ser iniciados alguns testes,por exemplo:os testes de depuração de erros que são executados durante a programação e que podem ser executados pelos próprios programadores.É importante que nessa fase as “versões” do sistema que estão sendo implementadas sejam controladas e gerenciadas, para que se tenha um controle de tudo o que está sendo codificado e alterado.

Sendo assim, antes de se iniciar a etapa de implementação de um software, é necessário escolher o ambiente de programação e tratar outras questões que possam influenciar direta ou indiretamente no bom desempenho dessa atividade.Além da escolha do ambiente de programação,existem também boas práticas a serem seguidas para facilitar,principalmente,a manutenção do software e de alguns problemas a serem solucionados relativos à: documentação,rotinas de teste,integração da equipe de desenvolvimento e a composição de arquivos de configuração da aplicação.No caso de ser um ambiente orientado a objetos,outros problemas surgirão,como,por exemplo: controle de instâncias e relacionamentos entre objetos e persistência de objetos.

Teste de Software: O software tornou-se um componente importante e de sucesso para várias empresas desenvolvedoras, fazendo que haja uma crescente busca pela qualidade do seu produto final,o software.Segundo Weber:a qualidade de um software é determinada pela qualidade dos processos que serão usados durante a fase de desenvolvimento do mesmo.A

qualidade de software é o resultado de atividades que foram realizadas no processo de desenvolvimento desse software.Quando se fala em qualidade de software,é necessário lembrar que:o projeto do software,processo de desenvolvimento e o produto final têm que ter qualidade também.

Segundo Pressman:a atividade de teste de software é uma garantia de qualidade de software e ela é a última fase que representa a revisão do que foi especificado nas fases de projeto e implementação.

Segundo Sommerville:os objetivos da fase de teste de software podem ser expressos,de forma mais clara por:

- Atividade de Teste: processo de executar um programa com a intenção de localizar um defeito/erro;

- Caso de teste bom:apresenta uma elevada probabilidade de revelar um defeito/erro ainda não descoberto;

Conforme Molinari:o teste é uma atividade que deve ocorrer paralela ao desenvolvimento e conduzida nas diversas fases do processo de desenvolvimento de software.E,para isso,o teste deve ser planejado, controlado e supervisionado por profissionais experientes.A equipe de teste deve identificar e minimizar os erros no software e executar atividades em paralelo ao teste:como documentação e relatórios.

Bastos fala que:quanto menos defeitos forem deixados no software nas fases iniciais,menos custos terá a sua manutenção depois do software estiver pronto,ou seja,mais rentabilidade para empresa em questão.

Sobre o teste,Pressman afirma:O teste muitas vezes requer mais trabalho de projeto do que qualquer outra ação da engenharia de software.Se for feito casualmente,perde-se tempo,fazem-se esforços desnecessários,e,ainda pior, erros passam sem ser detectados.Portanto,é razoável estabelecer uma estratégia sistemática para teste de software.

De acordo com Molinari:todo software que se destina ao público e/ou ao mercado deve sofrer um nível mínimo de teste.Assim,quanto maior o nível de complexidade do software,mais testes e técnicas de testes se tornam necessários para a obtenção da sua qualidade.Sem os testes,não se consegue garantir que o software irá se comportar conforme o esperado ou conforme as solicitações do cliente, e isso pode ser negativo para a empresa que o desenvolveu.Mas caso na empresa não se tenha uma análise de requisitos ou uma documentação do software detalhada,a equipe de desenvolvimento e a equipe de teste não saberão se o que está sendo construído é o que o cliente espera.Pensando nisso,Molinari destacou axiomas e conceitos que podem ser usados no processo de teste,e que em muitos casos são considerados como verdades no mundo dos testes.Podem ser listados como:

- Não é possível testar um programa completamente;

- Teste de software é um exercício baseado em risco;

- Teste não mostra que bugs não existem, mas sim, o contrário;

- Quanto mais bugs são encontrados, mais bugs poderão aparecer;

Rios denuncia que os testadores querem destruir o software,nocauteando através da busca de falhas e indicando seus erros!!! pois conforme o autor,uma vez que se indicam os defeitos de um software,ele pode ser corrigido e com isso,se torna muito melhor.Conforme Rios e Moreira “se não se podem descobrir todos os defeitos de um programa e em decorrência disso nunca se pode afirmar que ele está 100% correto, por que testar? Porque o propósito dos testes é descobrir e corrigir os problemas e,com isso melhorar a sua qualidade.

O autor ainda acrescenta,que “na prática, não se pode testar um programa por completo e garantir que ele ficará livre de bugs”.É quase impossível testar todas as possibilidades de formas e alternativas de entrada de dados,bem como testar as diversas possibilidades e condições criadas pela lógica do programador”.

Segundo Pressman “muitas estratégias de teste de software já foram propostas na literatura”.Todas elas fornecem um modelo para o teste e todas têm características genéricas”,e elas são:

- Executar um teste eficaz, proceder revisões técnicas eficazes;

- Teste começa no nível de componente e progride em direção à integração do sistema;

- Diferentes técnicas de teste são apropriadas para diferentes abordagens.

- O teste é feito pelo desenvolvedor do software e (para grandes projetos) por um grupo independente de teste;

- O teste e a depuração são atividades diferentes,mas a depuração deve ser associada com alguma estratégia de teste;

Segundo o mesmo:o teste dificilmente chega ao fim.O que vai acontecer é uma transferência do desenvolvedor para o seu cliente,ou seja, toda vez que um cliente usa o sistema,um teste está sendo realizado.

Por fim,o testador deve saber exatamente o seu nível de competência que é medido pela sua experiência e pelos cursos que fez. Não tentando testar um software para o qual você não se tem conhecimento técnico suficiente.Testar softwares embarcados não será a mesma coisa que testar softwares comerciais.A graduação como testador deve sempre dizer até onde você

poderá chegar com o seu trabalho,logo não ultrapasse limites,pois nessas faixas é que poderão aparecer os seus maiores defeitos.Aquele testador que registra um defeito dando palpites técnicos sobre como o software deveria ser desenvolvido,com toda a certeza está ultrapassando os seus limites.O certo é conhecer a sua área de atuação e os limites que demarcam os seus conhecimentos daqueles inerentes aos do desenvolvedor,pois teoria e prática são segmentos bem distintos.