

Engenharia de Software I

Aula – 3



Prof. Salatiel Luz Marinho



salatiel.marinho@docente.unip.br

Prazo, Custo e Planejamento de Qualidade

É importante verificar que as tarefas de orçamento do projeto, cronograma do projeto e planejamento do projeto, são tarefas que devem ser realizadas para todos os projetos em questão e as mesmas devem ser realizadas pelos seus gerentes de projetos.

- O gerenciamento de projeto de software é uma parte essencial da engenharia de software. Um bom gerenciamento não pode garantir o sucesso do projeto, no entanto, um mau gerenciamento geralmente resulta em falhas do projeto: o software é entregue com atraso, custa mais do que foi estimado originalmente e falha ao atender os requisitos mínimos do cliente.
- Os gerentes de softwares são responsáveis pelo desenvolvimento de planos, cronogramas e estimativa de custos do projeto. Eles supervisionam o trabalho para garantir que ele esteja sendo realizado dentro dos padrões exigidos e monitoram o progresso para verificar se o desenvolvimento está no prazo e dentro do orçamento. O gerenciamento de projeto é necessário, pois a engenharia de software está sempre sujeita às restrições de orçamento e de cronograma da organização.

Prazo, Custo e Planejamento de Qualidade

Os gerentes de projetos fazem o mesmo trabalho que gerentes de outra engenharias, no entanto, a engenharia de software é diferente das outras engenharias de uma série de maneiras. Algumas das diferenças são:

- O produto é intangível: Ou seja, o software é intangível, não pode ser visto ou tocado, os gerentes de projeto de software não podem ver seu progresso. Eles contam e devem confiar em outras pessoas para produzir a documentação necessária para examinar o progresso.
- Não existe processo padrão em software: O processo de software varia drasticamente de uma empresa para outra, não existe um padrão na sua construção, uma linha de produção como produtos manufaturados.
- Projetos de software de grande porte são geralmente projetos “únicos”: Esses projetos de alguma forma são sempre diferentes dos anteriores, portanto, mesmo gerentes experientes podem considerar difícil prever problemas.

Atividades do Gerenciamento

A maioria dos gerentes assume a responsabilidade por algumas ou todas das seguintes tarefas:

- Elaboração de proposta
- Planejamento e Desenvolvimento do cronograma do projeto
- Custo do projeto
- Monitoração e revisão do projeto
- Seleção e avaliação de pessoal
- Elaboração de relatórios e apresentações

Atividades do Gerenciamento

A maioria dos gerentes assume a responsabilidade por algumas ou todas das seguintes tarefas:

- Elaboração de proposta
- Planejamento e Desenvolvimento do cronograma do projeto
- Custo do projeto
- Monitoração e revisão do projeto
- Seleção e avaliação de pessoal
- Elaboração de relatórios e apresentações

Planejamento de Projeto

O gerenciamento eficiente de um projeto de software depende de um planejamento minucioso do progresso do projeto. Os gerentes devem prever problemas que podem ocorrer e preparar soluções experimentais para esses problemas. Um plano elaborado no início de um projeto deve ser usado como guia. Esse plano inicial deve ser o melhor possível em face das informações disponíveis. Ele deve evoluir à medida que o projeto progride e melhores informações se tornem disponíveis.

O planejamento é um processo iterativo, apenas concluído quando o projeto é concluído. À medida que as informações se tornam disponíveis durante o projeto, o plano deve ser regularmente revisado. As metas da empresa constituem um importante fator que deve ser considerado na formulação do plano do projeto. À medida que essas metas mudam, as metas do projeto também mudam e, portanto, são necessárias mudanças no plano de projeto.

No início do processo de planejamento deve ser avaliado as restrições que afetam o projeto. Junto com isso deve ser estimado os parâmetros do projeto, como sua estrutura, tamanho e distribuição de funções.

Planejamento de Projeto

As maiores dos planos devem incluir as seguintes seções:

Introdução: Descreve brevemente os objetivos do projeto e estabelece as restrições que afetam o gerenciamento do projeto.

Organização do projeto: Descreve o modo como a equipe de desenvolvimento está organizada, as pessoas envolvidas e seus papéis na equipe.

Análise de riscos: Descreve os possíveis riscos do projeto, e as estratégias adotadas se os mesmos ocorrerem.

Requisitos de recursos de hardware e software: Especifica os recursos necessários para o desenvolvimento.

Estrutura analítica: Especifica a estrutura analítica de um projeto

Cronograma do projeto: Apresenta as dependências entre as atividades, o prazo estimado necessário para atingir cada marco e a alocação de pessoas nas atividades.

Mecanismo de monitoração e relatório: Definem os relatórios de gerenciamento que devem ser produzidos, quando devem ser produzidos e o mecanismo de monitoração utilizado.

Cronograma do Projeto

O desenvolvimento do cronograma do projeto é um dos trabalhos mais difíceis para um gerente de projeto. Os gerentes estimam o tempo e recurso necessários, organizando-as em uma sequência coerente. A menos que o projeto cuja o cronograma esteja sendo desenvolvido seja similar a um projeto anterior, as estimativas anteriores constituem uma base incerta para o desenvolvimento do cronograma do novo projeto. A estimativa de cronograma é mais complicada pelo fato de que projetos diferentes podem usar métodos e linguagens de implementações diferentes.

Os cronogramas devem ser constantemente atualizados à medida que mais informações sobre o progresso se tornem disponíveis.

O desenvolvimento do cronograma do projeto envolve a divisão do trabalho total de um projeto em atividades separadas e a avaliação do tempo necessário para completar essas atividades. Em geral algumas das atividades podem ser realizadas simultaneamente.


Ao estimar o cronograma não podemos desconsiderar que todos os estágios do cronograma estarão livres de problemas, afinal de contas trabalhamos com pessoas e elas podem adoecer. Além do calendário o gerente de projeto deve prestar atenção também nos recursos necessários para realizar uma tarefa sendo que o principal dele é o esforço humano necessário.

Custo do Projeto

Existem três parâmetros envolvidos no cálculo do custo total de um projeto de software:

- Custo de hardware e software, incluindo a manutenção
- Custo de viagens e treinamentos
- Custo de esforços

O custo mais representativo nos projetos são os custos dos esforços, pois os custos dos esforços não são apenas salários dos engenheiros de software que estão envolvidos no projeto. As organizações calculam os custos dos esforços em termos indiretos nos quais são considerados o custo total de operação da organização e este é dividido pelo número de pessoal produtivo. Portanto os seguintes custos são parte do custo total de esforços:



Custo do Projeto

- Custo de subsistência, aquecimento e iluminação no espaço de escritório
- Custo de pessoal de apoio como contadores, administradores, gerentes de sistemas, faxineiros e técnicos.
- Custo de operações de rede de comunicações
- Custo de instalações centrais, como de biblioteca ou recreação
- Custo de seguridade social e benefícios dos empregados, como pensões e seguros

Uma vez que o projeto esteja em andamento, os gerentes de projetos devem atualizar regularmente suas estimativas de custo. Isso ajuda no processo de planejamento e uso eficiente de recursos. Se as despesas reais são significativamente maiores do que as estimadas, o gerente de projeto deve tomar alguma providência. Isso pode envolver a aplicação de recursos adicionais para o projeto ou a modificação do trabalho a ser realizado.

Qualidade de Software

A qualidade de software tem se aprimorado significativamente nos últimos 15 anos. Uma razão para isso é o fato de as empresas terem adotado novas técnicas e tecnologias, como o uso do desenvolvimento orientado a objeto e de ferramentas de apoio CASE, e também teve uma crescente conscientização da importância da qualidade de software.

Entretanto, qualidade de software é um conceito complexo que não é diretamente comparável com a qualidade na manufatura. Na manufatura, a noção de qualidade tem sido aquela de que o produto desenvolvido deve atender às especificações. Em um mundo ideal essas definições deveriam ser aplicadas a todos os produtos, mas, para sistemas de software há problemas com isso, embora existam algumas medidas de programas.

Bons gerentes de qualidade têm por objetivo desenvolver uma cultura de qualidade na qual todos os responsáveis por desenvolvimento de produto estão comprometidos em atingir um alto nível de qualidade de produto. Eles encorajam as equipes a assumirem a responsabilidade pela qualidade de seu trabalho e a desenvolverem novas abordagens para aprimoramento da qualidade.

Qualidade de Software

O gerenciamento de qualidade formalizada é particularmente importante para as equipes que estão desenvolvendo sistemas amplos e complexos. A documentação de qualidade é um registro do que tem sido feito por cada grupo no projeto. Isso ajuda as pessoas a verificarem se tarefas importantes não foram esquecidas ou se uma parte da equipe não fez suposições incorretas sobre o que outras equipes fizeram. A documentação de qualidade é também um meio de comunicação ao longo da existência de um sistema. Para sistemas menores a gerencia de qualidade é ainda importante, mas a mesma pode ser adotada de uma maneira mais informal.

No desenvolvimento de software a qualidade de projeto abrange os requisitos, as especificações e o projeto do sistema.

Satisfação do usuário = produto adequado + máxima qualidade + entrega no prazo dentro do orçamento

Para a qualidade de um projeto existem três tarefas de suma importância:

Controle de Qualidade

O controle da variação pode ser equiparado ao controle da qualidade. O controle da qualidade envolve a série de inspeções, revisões e testes usada ao longo do processo de software para garantir que cada produto de trabalho satisfaça os requisitos para ele estabelecidos.

Um conceito chave do controle de qualidade é que todos os produtos de trabalho tem especificações mensuráveis e definidas com as quais nós podemos comparar os resultados de cada processo.

Garantia da Qualidade


Garantia da qualidade consiste em um conjunto de funções para auditar e relatar que avalia a efetividade e completeza das atividades de controle de qualidade. A meta da garantia da qualidade é fornecer a gerencia os dados necessários para que fique informada sobre a qualidade do produto, ganhando assim compreensão e confiança de que a qualidade do produto esta satisfazendo as suas metas.

Custo da Qualidade

O custo da qualidade inclui todos os custos decorrentes da busca da qualidade ou da execução das atividades relacionadas à qualidade. Estudos de custo de qualidade são conduzidos para obter um referencial para o custo de qualidade. Esse custo de qualidade pode ser dividido em custos associados com a prevenção, com a avaliação e com as falhas. Os custos de prevenção incluem planejamento da qualidade, revisões técnicas formais, equipamentos de teste e treinamentos. Os custos de avaliação incluem atividades para obter entendimento da condição do produto na primeira execução de cada projeto.

Os custos de falhas são aqueles que desapareceriam se nenhum defeito aparecesse antes de se entregar um produto ao cliente. Os custos de falhas podem ser subdivididos em custos de falhas internas e custo de falhas externas. Sendo o custo de falhas internas aqueles que ocorrem quando detectamos um defeito no nosso produto antes do embarque. Os custos de falhas internas incluem refazer, reparar e analisar o modo como a falha ocorreu. O custo de falha externa é associado com os defeitos encontrados depois que o produto já esta com o cliente, geralmente são os mais caros.

Modelos de Qualidade de Software

- **CMMI;**
 - **MPSBr;**
 - **ISO**
- 

CMMI

Antes de surgir o modelo CMMI, surgiu o modelo SW – CMM (Capability Maturity Model for software) é um modelo de capacitação de processos que foi patrocinado pelo departamento de defesa dos EUA, para avaliação da capacidade dos seus fornecedores de software. O CMM baseou-se em algumas idéias importantes do movimento de qualidade industrial.


Com o sucesso do SW-CMM, outros modelos foram sendo criados, o P-CMM (People CMM para recursos humanos), o SA-CMM (para aquisição de software) e o SE-CMM para engenharia de sistemas, com o surgimento de todas essas vertentes de modelos começou a ser causada uma certa confusão de quando usar um e outro, para isso foi criado então o CMMI que integrou todos esses e pode ser aplicado para a empresa como um todo.

O objetivo do CMMI é servir de guia para melhoria de processos na organização e também na habilidade dos profissionais em gerenciar o desenvolvimento, a aquisição e manutenção de produtos e serviços.

CMMI

O CMMI um pouco diferente do seu originário SW-CMM, pode ser abordado por níveis ou então em melhoria contínua, atuando com níveis de capacidade.

Os níveis do CMMI são:

- Inicial – Ad hoc
 - Nível Gerenciado – Processos disciplinados
 - Nível Definido – Processos padronizados
 - Nível gerenciado Quantativamente – Processos medidos e controlados
 - Nível Otimizado – Processos melhorados continuamente.
- 

MPSBr

O MPSBr foi criado por pesquisadores brasileiros para melhoria de processos de desenvolvimento de softwares em empresas brasileiras. É um modelo recente e ainda em desenvolvimento.

O MPSBr atende à necessidade de implantar os princípios de engenharia de software de forma adequada ao contexto das empresas brasileiras, seguindo as principais abordagens internacionais para definição, avaliação e melhoria de software.

A definição desse processo baseia-se em três guias: Guia Geral, Guia de aquisição e Guia de Avaliação.

MPSBr

O MPSBr assim como o CMMI também define níveis de maturidade, porém esses níveis são sete compreendidos de A a G sendo:

A – Em otimização

B – Gerenciado Quantitativamente

C – Definido

D – Largamente Definido

E - Parcialmente Definido

F – Gerenciado

G – Parcialmente Gerenciado

ISO

As normas ISO 9000 é uma referencia em sistemas de qualidade, tendo influência em diversas outras normas e metodologias.

A ISO9000 tornou-se sinônimo de preocupação com qualidade em todo o mundo, a sigla é conhecida por consumidores dos mais diversos produtos e funciona muitas vezes como um apelo de marketing muito eficiente. A série teve origem na norma britânica BS570, baseada em padrões militares ainda mais antigos.

A ISO9000 é, na realidade, uma família de normas e tem um caráter genérico. Serve aos propósitos de qualquer organização, em qualquer ramo de atividade, que queria realizar o controle de qualidade dos produtos ou serviços oferecidos.

ISO


A norma ISO/IEC 15504 guarda estreita relação com o modelo CMMI. Essa norma apresenta estrutura para a realização de avaliações de processos em organizações, pode ser aplicada em situações como:

- Uma empresa que busca melhorias internas;

Avaliação de terceiros ao realizarem contratos de prestação de serviços ou fornecimento de produtos.

Modelos de Processo Pessoal e de Equipe

O melhor processo de software é aquele que se aproxima do pessoal que fará o serviço. Se um modelo de processo de software tiver sido desenvolvido no nível da empresa ou organização, ele pode ser efetivo apenas se for suscetível a adaptações significativas de modo a satisfazer às necessidades da equipe de projeto que esta realmente fazendo o trabalho de engenharia de software.



PSP (Personal Software Process – Processo Pessoal de Software)

Todo desenvolvedor usa algum processo para construir software de computador. O processo pode ser aleatório ou ad hoc, pode modificar-se diariamente, pode não ser eficiente, efetivo ou mesmo, bem-sucedido, mas existe um processo.

O PSP enfatiza a medição pessoal tanto do produto do trabalho que é produzido quanto a qualidade resultante do produto do trabalho. Além disso, o PSP torna o profissional responsável pelo planejamento do projeto e dá poder ao profissional para controlar a qualidade de todos os produtos do trabalho de software, por ele desenvolvido.

O modelo PSP define cinco atividades: planejamento, projeto de alto nível, revisão do projeto de alto nível, desenvolvimento e pós-conclusão.

PSP (Personal Software Process – Processo Pessoal de Software)

Planejamento: Essa atividade, isola os requisitos e, com base neles, desenvolve estimativas tanto de tamanho quanto de recurso. Toda métrica é registrada em planilhas e gabaritos. Finalmente, as tarefas de desenvolvimento são identificadas e um cronograma de projeto é criado.

Projeto de alto nível: São desenvolvidas especificações externas para cada componente a ser construído e é criado um projeto dos componentes, Protótipos são construídos quando existe incerteza. Todos os itens são registrados e monitorados.

Revisão do Projeto de alto nível: Métodos de verificação formal são aplicados para descobrir erros no projeto.


Desenvolvimento: O projeto em nível de componentes é refinado e revisado. O código é, revisado, compilado e testado.

Pós- conclusão: Usando as medidas e as métricas coletadas, a efetividade do processo é determinada.

PSP (Personal Software Process – Processo Pessoal de Software)

O PSP enfatiza a necessidade de cada engenheiro de software identificar logo os erros, e igualmente importante, entender os tipos de erros que ele tende a fazer. Isso é conseguido por meio de uma atividade de avaliação rigorosa desenvolvida em todos os produtos de trabalho produzidos pelo Engenheiro de Software.

O PSP representa uma abordagem disciplinada, baseada na métrica, da engenharia de software que pode levar a um choque de cultura em muitos profissionais, quando o mesmo é usado pela engenharia de software o aperfeiçoamento resultante na produtividade e qualidade é significativo.



TSP (Team Process Software – Processo de Equipe de Software)

Como muitos projetos de software de nível industrial são desenvolvidos por uma equipe de profissionais, Watts Humphrey estendeu as lições aprendidas com a introdução do PSP e propôs o TSP. O objetivo do TSP é construir uma equipe de projeto “autodirigida” que se organize para produzir um software de alta qualidade. Para isso foi definido os seguintes objetivos:

Construir equipes autodirigidas que planejem e monitorem seu trabalho, estabelecem metas, e possuam seus próprios processos e planos.

Mostrar aos gerentes como acompanhar e motivar suas equipes, e como ajudá-las a manter seu desempenho de pico,

Acelerar o aperfeiçoamento do processo de software tornando o comportamento de nível 5 do CMMI normal e esperado,

Facilitar o ensino as habilidades de equipe de nível industrial.

TSP (Team Process Software – Processo de Equipe de Software)

O TSP define as seguintes atividades: lançamento, projeto de alto nível, implementação, integração e teste, pós-conclusão.

Essas atividades permitem que a equipe planeje, projete e construa softwares de modo disciplinado e, ao mesmo tempo, meça quantitativamente o processo e o produto. A pós-conclusão prepara o cenário para aperfeiçoamento do processo.

O TSP usa uma grande variedade de documentos, formulários e normas que servem para guiar os membros da equipe em seus trabalhos.



TSP (Team Process Software – Processo de Equipe de Software)

O TSP reconhece que as melhores equipes de software são as autogeridas. Os membros da equipe estabelecem objetivos do projeto, adaptam o processo para satisfazer às suas necessidades, têm controle sobre o cronograma e, por meio de medições e análise da métrica coletada, trabalham continuamente para melhorar a abordagem da equipe do ponto de vista da Engenharia de software.

O TSP é uma abordagem rigorosa de Engenharia e fornece benefícios distintos e quantificáveis em produtividade e qualidade. A equipe deve estabelecer um total comprometimento com o processo e deve passar por treinamentos para garantir que a abordagem seja propriamente aplicada.

Dúvidas

