

Engenharia de *Software*



Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

Material Teórico



Planejamento e gerenciamento de projetos

Responsável pelo Conteúdo:

Prof.^a Dr.^a Ana Paula do Carmo Marchetti Ferraz

Revisão Textual:

Prof.^a Me. Luciene Oliveira da Costa Santos



- Introdução
- Acompanhamento de Processo



Objetivo de APRENDIZADO

Nesta unidade apresentaremos a importância do planejamento e do gerenciamento para que um projeto de software seja bem-sucedido. Você conhecerá tarefas e o esforço necessário para a criação de um plano realístico e administrável, além da importância das atividades de levantamento e gerenciamento de requisitos.

Você também compreenderá quais atividades devem ser executadas para a melhor análise das necessidades do cliente. Além disso, conhecerá a problemática da comunicação entre as pessoas envolvidas nessas atividades.



Atenção

Para um bom aproveitamento da Disciplina, leia o material teórico atentamente antes de realizar as atividades. É importante também respeitar os prazos estabelecidos no cronograma.

Contextualização

Durante as duas unidades anteriores, você percebeu que o desenvolvimento de *software* envolve inúmeras tarefas. Essas tarefas precisam ser executadas para haver a produção de um *software* que atenda às necessidades do cliente.

À medida que se expandiram as aplicações para *software*, ele passou a ser utilizado nas mais variadas áreas. Um dos desafios da Engenharia de *Software* é aumentar a produtividade dos processos de desenvolvimento para que a indústria de *software* consiga suprir a crescente demanda por novos *softwares*. Porém, além de focar na velocidade da criação dos processos de *software*, é importante que os produtos entregues tenham a qualidade esperada.

Para que um projeto seja bem-sucedido, é importante que ele seja planejado e gerenciado. Isso vale para todos os tipos de projetos, inclusive para os de *software*.

Nesta unidade 3, apresentaremos alguns conceitos relacionados ao planejamento e gerenciamento de projetos.

Não existe projeto bem-sucedido, sem planejamento e gerenciamento e não existe planejamento sem informações. Sugerimos que assista aos seguintes vídeos, sugeridos para entender um pouco mais sobre a importância do planejamento e do gerenciamento e seu inter-relacionamento com diversas áreas do conhecimento:



Explore

- http://www.youtube.com/watch?v=_xustQAKn6I
- <http://www.youtube.com/watch?v=fVwk9WUNYuU>
- <http://www.youtube.com/watch?v=z9hqZPgktO8>

Nesta unidade, especificamente sugerimos pesquisas adicionais sobre métodos de estimativa de tempo para execução de determinadas atividades. Dentre eles, podemos citar o Método do Caminho Crítico (COM – *Critical Path Method*), que é bastante utilizado. No entanto, devido à carga horária de nossa disciplina, não teremos como abordá-lo.

Outro assunto que sugerimos pesquisas adicionais são *softwares* que possuem funcionalidades relacionadas à gestão de projetos. Alguns deles são *Excel* da Microsoft, *NetOffice*, *MS-Project* da Microsoft, *Planner*, *DotProject*, *GranttProject* e, se considerar necessário, faça o download (alguns *softwares* são free). É importante que você realize a pesquisa para conhecer a funcionalidade de ferramentas como essas e como se relacionam na gestão de projetos de *software*, principalmente na relação das ferramentas que utilizam o “Diagrama de Gantt” ou “Gráfico de barras de atividades” como recursos acoplados ou independentes e verifique as principais características.

Introdução



Sempre quando começamos qualquer coisa na vida, ou melhor, quando pensamos em executar qualquer atividade, seja uma viagem, arrumar o guarda-roupa, levar o carro para lavar, sair de casa etc., a primeira coisa a fazer é **Planejar** e, posteriormente, **Gerenciar** e avaliar se o que foi planejado foi executado dentro do esperado.

Construir um software não é diferente.

Já estamos familiarizados com os modelos de desenvolvimento de software e optar por cada um deles é considerar características específicas de funcionalidades e dos requisitos e escolher a forma como planejar o desenvolvimento – que inclui custo, tempo, investimento, pessoas etc.

Nesta unidade, entenderemos um pouco desse processo e conheceremos algumas técnicas que nos ajudarão.

Não importa o tamanho ou a funcionalidade do seu software – atividades de análise, planejamento e gerência nunca deverão ser ignoradas ou menosprezadas. Delas pode depender o sucesso do seu projeto.

O tempo é o bem mais valioso que está disponível a um Engenheiro de Software.

Se houver tempo disponível, um software pode ser adequadamente analisado, uma solução pode ser comprehensivamente planejada, o código-fonte cuidadosamente implementado e testado. O problema é que **nunca há tempo suficiente.**(Autor desconhecido)

Acompanhamento do Processo



A atividade de gerenciamento e planejamento de um sistema pode ser vista como uma *atividade guarda-chuva*, pois abrange todo o **processo** de desenvolvimento. Ela possibilita **compreender** o escopo do trabalho, riscos, recursos exigidos, tarefas a executar, programação (cronograma) a ser seguido, esforço despendido, dentre outras características.

Segundo Pfleeger (2004, p. 63), “[...] **um software somente é útil se realizar uma função desejada ou fornecer um serviço necessário**”. Por isso um projeto geralmente começa quando um cliente faz um pedido de um sistema, ou quando, a partir da percepção visionária de alguém, começa a ser idealizado, pensado, projetado.

Imagine que alguém lhe solicite a construção de um software para efetuar a matrícula de alunos de uma escola que vai do Maternal ao Ensino Médio e que também os pais, por meio da web, possam ter acesso a algumas informações específicas dos seus filhos (matrículas, boletins, boletos etc.).

Você provavelmente, ao final da reunião sobre essa proposta, teria que responder a algumas questões para o cliente, dentre elas:

- » Você entendeu o que eu quero?
- » Você pode projetar esse sistema?
- » Quanto tempo você levaria para projetá-lo?
- » Quanto vai custar o desenvolvimento disso?

Responder às duas últimas perguntas pode ser a coisa mais complicada do sistema e requer um cronograma de projeto bem planejado. Muitas vezes, devido à ausência de respostas claras e objetivas sobre esses dois itens é que acontecem falhas como, por exemplo, sistemas não entregues no prazo previsto porque sua estimativa de tempo de desenvolvimento e esforço foi subdimensionada.

Um cronograma de projeto descreve o ciclo de desenvolvimento de software para um projeto específico, enumerando as etapas ou estágios e dividindo cada um deles em tarefas ou atividades a serem realizadas. O cronograma também retrata as interações entre essas atividades e estima o tempo necessário para a realização de cada tarefa ou atividade. Portanto, o cronograma é uma linha de tempo que mostra quando as atividades começarão e terminarão, e quando estarão prontos os produtos relacionados ao desenvolvimento. (PFLEINGER, 2004, p. 66)

Se até as tarefas mais simples – rotina diária – precisam de planejamento, podemos dizer que tudo nasce de um projeto (formal ou não) e, a partir da definição do que esperamos como resultados, são previstos tempo e valores.

Segundo Sommerville (2011), os critérios do sucesso de um sistema alteram-se de um para outro. Entretanto, questões como fornecer o software ao cliente no prazo estabelecido, manter os custos gerais dentro do orçamento, entregar o produto que atenda à expectativa do cliente e manter a equipe de desenvolvimento satisfeita, bem e feliz são pontos comuns do reconhecimento positivo de qualquer software.

Se você analisar com cuidado o parágrafo anterior, perceberá que esses itens não são exclusivos da Engenharia de Software (ES), mas de todo projeto de Engenharia. Entretanto, o que diferencia o gerenciamento de um produto de software de outros é o fato de que o produto de software é intangível. Os grandes projetos de software geralmente são únicos, pois todos eles, de uma forma ou de outra, possuirão especificações diferentes para clientes diferentes. Mesmo com questões relacionadas a padrões existentes na área de ES, os produtos de software são mudados de uma organização para outra.

Sendo assim, todo produto precisa levar em conta as fases de análise dos recursos e objetivos e, a partir deles, considerar a definição de atividades que serão executadas e de marcos.

Mas o que é realmente **atividade de marco** na ES?

Você já ouviu a expressão “isso foi um marco na minha vida”? Consegue perceber, quando alguém fala isso, o que significa?

Para entendermos, temos que definir primeiro o que é **atividade**. Ela é uma parte do projeto que acontece ao longo de determinado período, enquanto **marco** é a finalização de uma atividade – um momento específico no tempo que pode ou não marcar o início de outra atividade. Por exemplo, temos a atividade de entrevista do cliente; após esta atividade, podemos finalizá-la (marco) e iniciamos o desenvolvimento.

Ao pensarmos num projeto como um todo, podemos (e devemos) definir que seu desenvolvimento seja o resultado de uma sucessão de fases, todas com etapas e com uma ou várias atividades inseridas, como mostra a Figura 1:

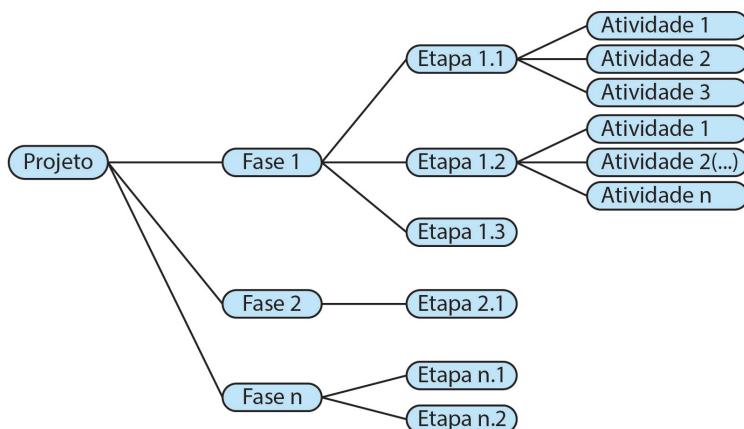


Figura 1 – Esquema de um projeto dividido em fases e etapas.

Para ilustrarmos esta Figura, pense no desenvolvimento do projeto solicitado no início deste Material teórico – um sistema escolar. Ele deverá ser dividido em grandes fases como, por exemplo, análise, codificação e teste.

Na análise, podemos definir algumas etapas – falar com o cliente, analisar os dados e requisitos coletados, modelar o sistema, dentre outras.

Se pensarmos na etapa falar com o cliente, podemos definir algumas atividades como: marcar a reunião, definir as perguntas que nortearão ações, tomar nota das solicitações dos clientes etc.

Essas atividades e os marcos gerados a partir delas funcionarão como referências para acompanhar o desenvolvimento ou a manutenção do projeto.

Em cada atividade, quatro parâmetros devem ser definidos: o precursor, a duração, a data prevista e o encerramento. Podemos dizer que um **precursor** é um evento ou um conjunto de eventos que deve ocorrer antes da atividade começar. Ele descreve uma série de condições necessárias para que a atividade tenha início. A **duração** é o tempo necessário para completar a atividade. A **data prevista** é aquela em que a atividade deve ser concluída, em geral determinadas por prazos contratuais. O **encerramento** é um marco ou um requisito que foi concluído.

A partir disso, é possível começarmos a definir uma **estrutura de divisão do trabalho**. É nesse momento em que são montadas as equipes de análise, de programação, do setor financeiro, as equipes de teste e até de manutenção **para determinado projeto**.

Cada projeto possuirá equipes diferentes, de acordo com as especificidades requeridas e competências de cada pessoa que trabalha na empresa.

Essas equipes podem ser alocadas para participar durante todo o projeto ou apenas em algumas fases, por exemplo, a equipe de análise pode participar apenas da fase inicial do projeto na qual são definidos os requisitos e modelado o sistema, depois, ao final dessas funções, talvez possa se desligar do projeto “A” e ser realocada para o projeto “B”.

Os itens precursor, data, divisão do trabalho etc., ao serem planejados, darão origem à documentação do projeto que deverá servir de “mapa” para o gerenciamento do desenvolvimento.



É importante conhecermos as técnicas de determinação de estimativa de tempo para execução de determinadas atividades. Dentre elas, podemos citar Método do Caminho Crítico (COM – *Critical Path Method*) que é bastante usado. Entretanto, devido ao conteúdo delimitado e nosso prazo, não teremos como abordá-lo.

Gerenciar significa cuidar de:

- » **Pessoa:** Quem participa do Processo;
- » **Produto:** O que será desenvolvido;
- » **Processo:** Como será feito;
- » **Projeto:** Das fases, etapas e atividades.



Pense

O gerente que esquece que o trabalho da ES está intensamente relacionado com as pessoas nunca terá sucesso na gerência de projetos.



Atenção

O gerente de sistemas ou engenheiro de software que não incentivar uma compreensiva comunicação com o cliente, logo no início do projeto, corre o risco de equívocos na tentativa de solucionar problemas.

Ferramenta para Acompanhar o Processo

Existem inúmeras ferramentas que podemos utilizar para acompanhar o projeto, algumas mais simples como *Excel* e outras mais complexas, como o *MS-Project*, da *Microsoft*.

Para verificar qual delas pode ser utilizada, uma boa dica é analisar a estrutura do trabalho, dividir tarefas e trabalhos e, daí, optar por uma ferramenta ou uma combinação de diferentes ferramentas.

Para ilustrarmos essa ideia de ferramenta, vamos utilizar um exemplo de Pfleeger (2004, p. 71-72).

Vamos considerar a seguinte estrutura de divisão de trabalho:

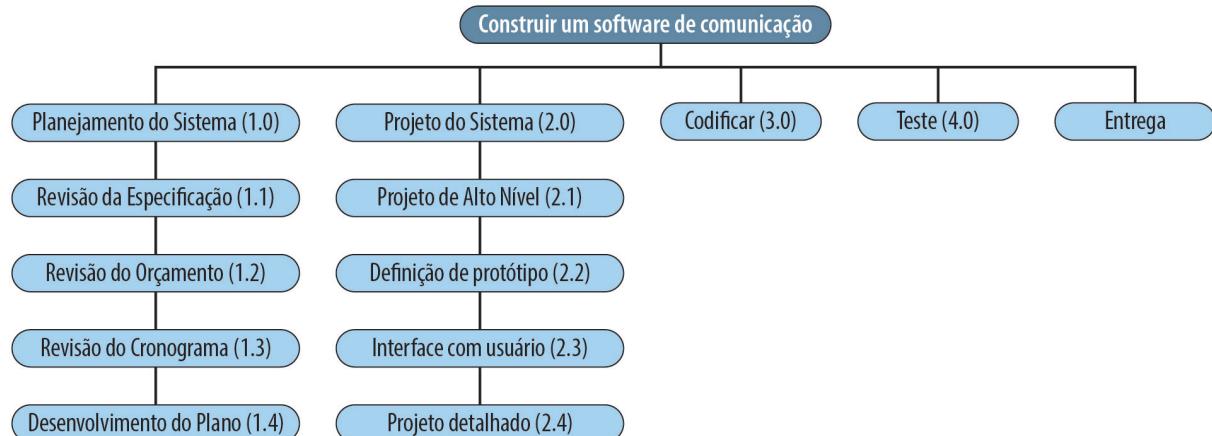


Figura 2 – Exemplo de uma divisão do trabalho, adaptado de Pfleeger (2004).

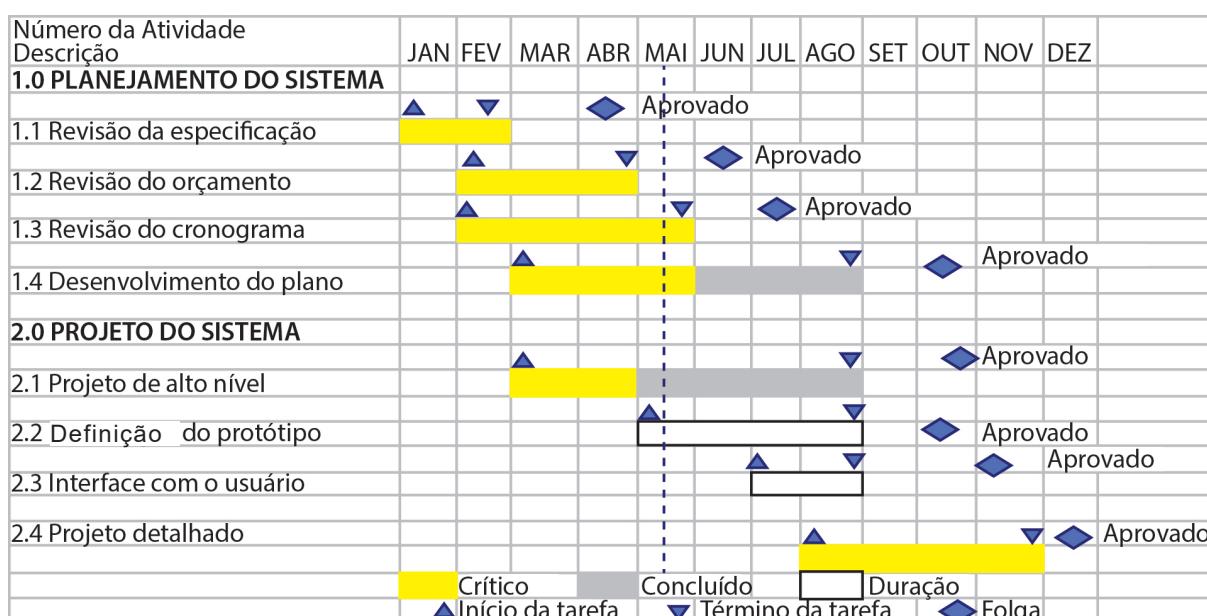


Figura 3 – Diagrama de Gantt para o exemplo da Figura 2; adaptada de Pfleeger (2004)

A partir da divisão do trabalho, podemos discutir, com mais precisão, tempo e gerenciamento do desenvolvimento do projeto

Algumas ferramentas conduzem o gerente por meio de um **Diagrama de Barra de Atividades**. Nele, as atividades são mostradas em paralelo, com grau de conclusão indicado por uma cor ou um ícone. Esse diagrama ajuda o gerente a perceber quais atividades podem ser executadas simultaneamente e quais são críticas para o cumprimento das etapas.

A utilização correta de uma ferramenta que produza um diagrama de barra de atividade pode ser fundamental para o cumprimento de etapas, fases e atividades dentro do prazo e custo estipulados.

Na Figura 2 você pode ver o diagrama de barras de atividades, a partir da divisão do trabalho.

Fazer uma pesquisa sobre as ferramentas que utilizam o diagrama de barra de atividades e verificar as características de seus recursos.

Estimativas

Estimar um cronograma é uma das tarefas mais difíceis de realizar na etapa de planejamento do software.

São vários os aspectos a serem considerados: a disponibilidade de recursos no momento da execução de uma dada tarefa, as interdependências das diferentes tarefas, a ocorrência de possíveis estrangulamentos do processo de desenvolvimento e as operações necessárias para agilizar o processo, identificação das principais atividades, revisões e indicadores de progresso do processo de desenvolvimento etc.

Uma das formas de se garantir que os cronogramas sejam cumpridos é pensar no projeto como um todo, as interdependências que existem e como o documento pode trabalhar com o Diagrama de Barra de Atividades, pois a partir dele é possível verificar quais atividades são dependentes, simultâneas e o tempo total para a execução.

O cronograma deve apresentar as atividades importantes do desenvolvimento e os indicadores de progresso, ou seja, deve prever os marcos do projeto. É importante que os indicadores de progresso sejam percebidos por resultados concretos que devem estar documentados.

A disponibilidade de recursos deve ser representada ao longo do cronograma. O impacto da indisponibilidade dos recursos no momento em que eles são necessários deve também ser representado, se possível, no cronograma.

Fixar prazos do projeto não é tarefa fácil, entretanto, quando realizadas, algumas questões devem ser respondidas:

- » Como relacionar o tempo cronológico com o esforço humano? Quantas pessoas serão necessárias para realização do projeto?
- » Que tarefas e que grau de paralelismo podem ser obtidos?
- » Como podemos medir o progresso do processo de desenvolvimento, ou seja, quais os indicadores de progresso que devemos monitorar?
- » Como o esforço pode ser distribuído ao longo do processo?
- » Que métodos estão disponíveis para a determinação de prazos?
- » Como representar fisicamente o cronograma e como acompanhar o progresso a partir do início do projeto?

Outro ponto crucial no planejamento e gerenciamento de um projeto é a **Estimativa de Custo**.

Um custo mal dimensionado acarreta problemas: se superestimado, pode provocar uma rejeição do cliente na hora de fechar o negócio; uma estimativa de preço abaixo do real pode levar a equipe a trabalhar muito além do tempo necessário sem o devido retorno financeiro.

Uma boa estimativa de custo faz com que o projeto se desenvolva de forma adequada, auxilia na definição das etapas, da quantidade de pessoas no desenvolvimento do projeto, além da competência e do perfil necessário de cada integrante da equipe antecipadamente.

Existem inúmeras razões para estimativas imprecisas e Pfeleger (2004, p. 81) define algumas:

- » Frequentes solicitações de mudança pelos usuários.
- » Tarefas negligenciadas.
- » Falta de entendimento do usuário sobre suas próprias exigências.
- » Análise insuficiente no desenvolvimento de uma estimativa.
- » Falta de coordenação do desenvolvimento do sistema, dos serviços técnicos, das operações, do gerenciamento de dados e de outras funções durante o desenvolvimento.
- » Complexidade do sistema proposto.
- » Necessidade de integração com os sistemas existentes.
- » Complexidade dos programas no sistema.
- » Capacidade dos membros da equipe de projeto.
- » Experiência da equipe do projeto com a aplicação.
- » Quantidade de padrões de programas e documentação.
- » Disponibilidade de ferramentas, tais como geradores de aplicação.
- » Experiência da equipe com o hardware.

Claro que essas não são as únicas razões que podem provocar uma estimativa mal feita. Porém, ao observarmos esses pontos, estaremos conduzindo o projeto em direção a uma estimativa de custo adequada.

Quando falamos em custo ou orçamento de projeto, estamos falando de vários tipos: instalações, pessoal, métodos e ferramentas.

É importante pensarmos em todos os custos diretos e indiretos na hora de dimensionarmos o valor. Custo de instalações, por exemplo, englobam hardware, espaço, mobília, telefones, modem, sistemas de aquecimento ou ar-condicionado, cabos, discos, papel, caneta, copiadores e todos os outros itens que fazem parte do ambiente no qual será desenvolvido o projeto.

É preciso pensarmos também em custos ocultos, que não são visíveis aos gerentes e desenvolvedores como, por exemplo, espaço físico para trabalhar, silêncio do ambiente etc.



Pense

Qual tipo de custo oculto você poderia citar?

Outro tipo de estimativa que deve ser considerada é a **Estimativa de Esforço**.

Um dos itens importantes no desenvolvimento de um projeto é o esforço a ser despendido.

Essa estimativa está relacionada ao número de pessoas/dia para desenvolver o projeto dentro do cronograma estipulado.

O esforço é o componente de custo de maior grau de incerteza, pois capacitação, experiência, interesse e treinamento podem influir no tempo de execução de uma tarefa (PLEEDGER, 2004).

Estimativas de custo, de cronograma e de esforços devem ser feitas o quanto antes, durante o ciclo de vida do projeto, pois elas afetam a distribuição de recursos e a viabilidade do projeto.

Análise de Riscos

A análise dos riscos é uma das atividades essenciais para o bom encaminhamento de um projeto de software e, consequentemente, da sua gerência.

Essa atividade está baseada na realização de quatro tarefas, conduzidas de forma sequencial:

- a) Identificação dos riscos.
- b) Projeção dos riscos.
- c) Avaliação dos riscos.
- d) Administração e monitoramento dos riscos.

Identificação dos Riscos

Na primeira etapa, o objetivo é que sejam levantados, pelos responsáveis gerentes e analistas, os eventuais riscos aos quais o projeto poderá ser submetido.

É importante salientar que é difícil identificar todos os riscos, mas é possível, a partir da experiência existente, identificar uma quantidade significativa e trabalhar num projeto mais coeso.

Podemos dividi-los em 3 categorias, de acordo com a sua natureza:

- **Riscos de projeto:** estão associados a problemas relacionados ao próprio processo de desenvolvimento (orçamento, cronograma, pessoal).
- **Riscos técnicos:** aqueles relacionados ao projeto nas fases de implementação, manutenção, interfaces, plataformas de implementação.
- **Riscos de produto:** estão mais relacionados aos problemas que surgirão a partir e para a inclusão do software como produto no mercado como, por exemplo, oferecer um produto que ninguém está interessado, produto ultrapassado ou inadequado à venda.

A categorização dos riscos, apesar de interessante, não garante a obtenção de resultados satisfatórios, uma vez que nem todos os riscos podem ser identificados facilmente. Uma boa técnica para conduzir a identificação dos riscos de forma sistemática é o estabelecimento de um conjunto de questões (checklist) relacionado a algum fator de risco.

Projeção dos Riscos

A projeção ou estimativa de riscos permite definir basicamente duas questões:

Qual a probabilidade de que o risco ocorra durante o projeto? Quais as consequências dos problemas associados ao risco no caso de ocorrência do mesmo?

As respostas a essas duas questões podem ser obtidas basicamente a partir de quatro atividades:

- » Estabelecimento de uma escala que reflita a probabilidade estimada de ocorrência de um risco.
- » Estabelecimento das consequências do risco.
- » Estimativa do impacto do risco sobre o projeto e sobre o software (produtividade e qualidade).
- » Anotação da precisão global da projeção de riscos.

A natureza do risco permite indicar os problemas prováveis, como casos de risco técnico – má definição de interface entre o software e o hardware –, que conduzirá certamente a problemas de teste e integração.

Avaliação dos Riscos

O objetivo da atividade de avaliação dos riscos é processar as informações sobre o fator de risco, o impacto do risco e a probabilidade de ocorrência.

Nesta avaliação, serão verificadas todas as informações coletadas na fase de projeção de riscos, a fim de priorizá-las e definir quais as melhores formas de controlá-los ou evitar sua ocorrência.

Para tornar a avaliação eficiente, deve ser definido um **nível de risco referente**. Exemplos de níveis referentes típicos são: custo, prazo e desempenho. Isso significa analisar se poderá haver excesso de custo, ultrapassagem de prazo de entrega e degradação do desempenho, ou qualquer combinação dos três em quaisquer níveis (profundidade).

Dessa forma, se os problemas observados ou a combinação dos riscos provocarem a superação de alguns dos níveis definidos, o projeto poderá ser suspenso. Normalmente, é possível estabelecer um limite, denominado ponto referente (*breakpoint*) onde tanto a decisão de continuar o projeto ou de abandoná-lo pode ser avaliada.

Administração e Monitoração dos Riscos

Uma vez avaliados os riscos de desenvolvimento, é importante que medidas sejam tomadas para evitar que eles realmente ocorram, ou planejar ações para realizá-las caso não tenhamos como evitá-los. Esse é o objetivo da tarefa de administração e monitoração dos riscos. Para isso, as informações mais importantes são aquelas obtidas na tarefa anterior, avaliação dos riscos.

Por exemplo, vamos imaginar que, numa determinada empresa, os fatores de riscos são dimensionados de zero a um. Especificamente no tocante à função de desenvolvedores, existe alta rotatividade de pessoal. Com base em dados de projetos passados, obtém-se que a probabilidade de ocorrência desse risco é de 0,70 (muito elevada) e que a sua ocorrência pode aumentar o prazo do projeto em 15% e o seu custo global em 12% (PRESSMAN, 2006, p. 234).

Sendo assim, podemos propor as seguintes ações para administrar os fatores de riscos:

- » Reuniões com os membros da equipe para determinar as causas mais importantes da rotatividade de pessoal.
- » Perceber questões relacionadas às condições de trabalho, salários, oportunidade de mercado de trabalho.
- » Criar planos de ações/providências para eliminar ou reduzir as causas “controláveis” desse processo antes de iniciar o projeto.
- » Prever, desde o início, quais trocas de pessoal poderão ocorrer. Prever a possibilidade de substituição dessas pessoas quando elas deixarem a equipe, criar equipe suplente a partir das competências exigidas para a função, principalmente para as funções mais críticas.
- » Organizar equipes de projeto de forma que as informações sobre cada atividade sejam compartilhadas, gerando um maior senso de equipe, principalmente no tocante a revisões dos trabalhos executados para que todos tenham ciência do processo.
- » Definir padrões para que a documentação seja construída de forma adequada.

É importante observar que a implementação dessas ações pode afetar também os prazos e o custo do projeto, por isso é fundamental avaliar o quanto tais ações poderão influenciar no tempo e valor final.

Todas essas ações não são e nem devem ser feitas empiricamente. Existem softwares que dimensionam e montam equipes (podem ser baseadas em experiências prévias, disponibilidade, habilidade técnica etc.), esforços e riscos. Calculam, matematicamente, esses valores e fornecem apoio às decisões.

O resultado do trabalho realizado, nesse momento, deverá ser registrado num documento denominado **Plano de Administração e Monitoração de Riscos**, o qual será utilizado posteriormente pelo gerente de projetos para a definição do Plano de Projeto, que é gerado ao final da etapa de planejamento que ajudará no gerenciamento.

Antes de tratarmos de Plano de Projeto, devemos conhecer o documento que engloba todos os assuntos vistos até o momento: **Plano de Software**.

Plano de software

Ao final da etapa de estimativa, de discussão sobre riscos e equipe está prevista a criação de um documento chamado de **Plano de Software (PS)**, o qual deverá ser revisto para servir de referência às etapas posteriores.

É importante mencionar que ele servirá como plano de ação para as próximas etapas e poderá ser atualizado sempre que necessário.

O PS apresentará as informações iniciais de custo e cronograma que vão direcionar todo o processo de desenvolvimento do produto de software.

PS não deve ser um documento longo, pois deverá ser enviado a várias pessoas envolvidas no projeto.

Todo PS deve, dentre outras, conter as seguintes informações:

- O contexto e a descrição dos recursos necessários para a efetiva gestão do projeto.
- A definição de custos e cronograma que serão acompanhados na gestão do projeto.
- A visão global do processo de desenvolvimento do software.

O Plano de Software não deve ser um documento extenso e complexo. O seu objetivo é auxiliar a análise de viabilidade dos esforços de desenvolvimento. Esse documento está associado aos conceitos de “o que”, “quanto” e “quão longo” ligados ao desenvolvimento. As etapas mais à frente vão estar associadas ao conceito de “como” (PRESSMAN, 1995).

Plano do projeto

Para comunicarmos a análise e o gerenciamento de riscos, a estimativa de custo, tempo e esforço, assim como a equipe e as diretrizes de teste, manutenção, garantia de qualidade do sistema etc., é necessário criarmos um **Plano de Projeto (PP)**.

O plano descreve as necessidades do cliente, assim como está sendo idealizada a execução dessas atividades.

Quando o sistema ainda está em análise, a documentação é chamada de PLANO; ao final do projeto, esse documento deverá ser atualizado e se tornará a documentação oficial do projeto que servirá como referência para novas atualizações, manual de usuário etc.

Um bom plano de projeto inclui os seguintes itens:

1. **Escopo do projeto:** descrição geral do projeto.
2. **Plano de Software:** informações que servirão de documentação para as etapas posteriores.
3. **Descrição técnica, ferramentas e procedimentos propostos para o projeto:** definição de requisitos e modelagem.
4. **Plano de garantia da qualidade:** definição das estratégias para garantia da qualidade do software.
5. **Plano de gerência de configurações:** definição das estratégias para gerenciar as alterações de software.
6. **Plano de documentação.**
7. **Plano de gerência de dados:** plano de gerenciamento de dados e integração de Bancos de Dados (BD).
8. **Plano de gerência de recursos:** plano de gerenciamento de recursos de desenvolvimento (pessoal, técnico etc.).
9. **Plano de testes.**
10. **Plano de treinamento.**
11. **Plano de segurança.**
12. **Plano de gerência de riscos.**
13. **Plano de manutenção.**



Explore

Plano de projeto. Disponível em: <<http://www.devmmedia.com.br/artigo-engenharia-de-software-3-plano-de-projeto/9527>>. Acesso em: 15 nov. de 2013.

MASIEIRO, P. Plano de Projeto. Disponível em: <http://www.univasf.edu.br/~ricardo.aramos/disciplinas/ESI2009_2/PlanejamentoProjeto_Masiero.pdf>. Acesso em: 15 nov. de 2013.

A partir de agora, podemos abordar os assuntos relacionados à definição de requisitos que direcionará todo o desenvolvimento.

Definir requisitos significa definir o que é necessário para o projeto atingir seus objetivos e por isso é uma fase de crucial importância na Engenharia de Software e dará origem à documentação de requisitos que servirá de Norte para os desenvolvedores.

Material Complementar



Explore

O objetivo do material complementar é lhe ajudar a entender, sob uma ótica diferente daquela da professora conteudista, assuntos abordados nas unidades teóricas.

É fundamental a leitura deste material para o melhor entendimento sobre o assunto.

Como nesta unidade abordamos os conceitos gerais da Engenharia de Software, nossa sugestão de material complementar é o capítulo 23, intitulado **Planejamento de projeto**, no seguinte livro:

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011. (p. 431-442).

Referências

Bibliografia Fundamental

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

Bibliografia Básica

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação**. 4. ed. Rio de Janeiro: LTC, 1999.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informação Gerenciais**. Administrando a empresa digital. 5. ed. São Paulo: Pearson Education do Brasil, 2006.

PFLEEGER, S. L. **Engenharia de Software**: teoria e prática. São Paulo: Prentice Hall, 2004.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Makron Books, 2006.

SOMMERVILLE, I. **Engenharia de Software**. 6. ed. São Paulo: Pearson Addison Wesley, 2005.

Bibliografia Complementar

ALCADE, E.; GARCIA, M.; PENUELAS, S. **Informática Básica**. São Paulo: Makron Books, 1991.

FAIRLEY, R. E. **Software engineering concepts**. New York: McGraw-Hill, 1987.

IEEE. **Software Engineering Standards**. 2013. Disponível em: <http://www.ieee.org/portal/innovate/products/standard/ieee_soft_eng.html>. Acesso em: 10 dez. 2013.

LUKOSEVICIUS, A. P.; CAMPOS FILHO, A. N.; COSTA, H. G. **Maturidade em gerenciamento de projetos e desempenho dos projetos**. Disponível em: <www.producao.uff.br/conteudo/rpep/.../RelPesq_V7_2007_07.doc>. Acesso em: 12 nov. 2013.

MAFFEO, B. **Engenharia de Software e especialização de sistemas**. Rio de Janeiro: Campus, 1992.

MICHAELIS. **Moderno dicionário da língua portuguesa**. São Paulo: Cia. Melhoramentos, 1998.

PARREIRA JÚNIOR, W. M. **Apostila de Engenharia de Software**. Disponível em: <http://www.waltenomartins.com.br/ap_es_v1.pdf>. Acesso em: 13 nov. 2013.

PAULA FILHO, W. P. **Engenharia de Software**: fundamentos, métodos e padrões. 2. ed. Rio de Janeiro: LTC, 2001.

Revista Engenharia de Software. Disponível em: <<http://www.devmmedia.com.br/revista-engenharia-de-software-magazine>>. Acesso em: 12 nov. 2013.

VONSTA., A. **Engenharia de Programas.** Rio de Janeiro: LTC, 1983.

WIENNER, R.; SINCOVEC, R. **Software engineering with Modula 2 and ADA.** New York: Wiley, 1984.

WIKIPEDIA (2007a). **ISO 9000.** Disponível em: <http://pt.wikipedia.org/wiki/ISO_9000>. Acesso em: 22 jun. 2007.

WIKIPEDIA (2007b). **Melhoria de processo de software brasileiro.** Disponível em: <<http://pt.wikipedia.org/wiki/MPS.BR>>. Acesso em: 22 jun. 2007.

WIKIPEDIA (2007c). **CMMI.** Disponível em: <<http://pt.wikipedia.org/wiki/Cmmi>>. Acesso em: 22 jun. 2007.

WIKIPEDIA (2007d). **Engenharia de Software.** Disponível em: <http://pt.wikipedia.org/wiki/Engenharia_de_software>. Acesso em: 1 fev. 2007.

Anotações





Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo SP Brasil
Tel: (55 11) 3385-3000



Universidade
Cruzeiro do Sul



UNICID
Universidade
Cidade de S. Paulo



UNIFRAN
Universidade
de Franca



UDF
Centro
Universitário



Módulo
Centro
Universitário