

Gerenciamento de Projeto

Engenharia de Software: Conceitos e Práticas

Prof. Raul Sidnei Wazlawick

UFSC-CTC-INE

Elsevier, 2013

Gerência de projeto

- A gerência de projeto pode ser entendida como uma disciplina dentro de um processo de engenharia de software, usualmente exercida por um único indivíduo (o gerente de projeto) que tem como objetivo levar o projeto a produzir os objetivos previamente planejados, dentro dos prazos, custos e qualidade previstos.
- Para ter sucesso nisto o gerente deve manter os riscos do projeto no nível mais baixo possível de probabilidade e impacto, avaliando continuamente o progresso e tomando medidas proativas para redução destes riscos, ou medidas de correção se apesar de tudo problemas ocorrerem.

Particularidades da gerência de projetos de SOFTWARE

- Os clientes dificilmente percebem as reais complexidades envolvidas com o processo de desenvolvimento de software, especialmente as dificuldades relacionadas com as mudanças de requisitos.
- É praticamente inevitável que os próprios processos de engenharia de software acabem gerando a necessidade de introdução de novos requisitos ou de modificação dos requisitos existentes.
- Como resultado disso, o software é frequentemente construído por um processo de refinamento iterativo ao invés de uma sequência de atividades previamente bem definidas e programadas.
- A engenharia de software necessariamente incorpora aspectos de criatividade e disciplina. Manter um balanceamento apropriado entre estes dois aspectos frequentemente é uma tarefa difícil.
- O grau de novidade e de complexidade do software frequentemente é extremamente alto.
- A tecnologia subjacente ao software muda com muita frequência.

Subáreas do gerenciamento de projetos de software (SWEBOK)

- Inicialização e definição de escopo.
- Planejamento de projeto de software.
- Condução de projeto de software.
- Revisão e avaliação.
- Fechamento.
- Medição em engenharia de software.

Gerentes de Projeto

- *Têm o dom de prever.*
- *São organizados.*
- *Sabem liderar.*
- *São bons comunicadores.*
- *São pragmáticos.*
- *São empáticos.*

Gerentes de Projeto

- *Têm o dom de prever.*
 - Bons gerentes têm a capacidade de visualizar e antecipar problemas antes que eles ocorram, e tomar ações preventivas.
- *São organizados.*
- *Sabem liderar.*
- *São bons comunicadores.*
- *São pragmáticos.*
- *São empáticos.*

Gerentes de Projeto

- *Têm o dom de prever.*
- *São organizados.*
 - A organização tem vários aspectos, mas um que parece ser especialmente importante para o gerente de projeto é a capacidade de visualizar prioridades e passá-las para sua equipe.
 - Assim, em meio à complexidade de um projeto o gerente será capaz de enxergar o que realmente é importante e concentrar os esforços da equipe nisso.
- *Sabem liderar.*
- *São bons comunicadores.*
- *São pragmáticos.*
- *São empáticos.*

Gerentes de Projeto

- *Têm o dom de prever.*
- *São organizados.*
- *Sabem liderar.*
 - O gerente de projeto não é apenas um chefe.
 - Além da equipe, ele precisa interagir com pessoas que não estão sob seu comando (clientes, usuários, especialistas de domínio, por exemplo).
 - Assim, ele precisa ter o carisma de líder e ser capaz de motivar as pessoas a gastar seu tempo nas atividades que são necessárias para o projeto.
- *São bons comunicadores.*
- *São pragmáticos.*
- *São empáticos.*

Gerentes de Projeto

- *Têm o dom de prever.*
- *São organizados.*
- *Sabem liderar.*
- *São bons comunicadores.*
 - Eles são capazes de utilizar múltiplos meios de comunicação, como email, telefone, reuniões, apresentações, etc., como forma de obter e transmitir as informações necessárias.
 - Eles são efetivos, objetivos e pragmáticos quando se comunicam (não ficam fazendo rodeios).
 - Além disso, são capazes de ouvir.
 - Um gerente que não ouve sua equipe ou outros interessados poderá levar um projeto a fracassar, pois pequenos problemas não resolvidos de início muitas vezes acabam virando grandes problemas no final de um projeto de desenvolvimento de software.
- *São pragmáticos.*
- *São empáticos.*

Gerentes de Projeto

- *Têm o dom de prever.*
- *São organizados.*
- *Sabem liderar.*
- *São bons comunicadores.*
- *São pragmáticos.*
 - Existem dois extremos quando se pensa na forma de tomada de decisão de um gerente: os que adiam a decisão até conhecer todas as implicações das opções, e os que tomam decisões por impulso, sem pensar.
 - No equilíbrio entre estes dois extremos está o gerente pragmático: ele analisa os prós e contras da forma mais eficiente possível, e é capaz de avaliar rapidamente se está em condições de tomar uma decisão fundamentada ou não.
 - Caso a análise dos pros e contras, porém, tomar mais tempo do que tentar uma das opções, ele vai perceber isso e, neste caso, decidir no sentido de tentar um caminho e, se não der certo, tentar o outro.
- *São empáticos.*

Gerentes de Projeto

- *Têm o dom de prever.*
- *São organizados.*
- *Sabem liderar.*
- *São bons comunicadores.*
- *São pragmáticos.*
- *São empáticos.*
 - Um gerente não faz o trabalho da equipe, mas também não faz o seu trabalho sozinho.
 - Ele precisa se apoiar em outras pessoas.
 - Para obter essa colaboração ele precisa entender o que motiva as pessoas.
 - Empatia é a capacidade de colocar-se no lugar do outro e entender suas necessidades.
 - A partir disso, o gerente balizará suas ações de motivação.

O gerente equilibra os vértices do triângulo

- Tempo
- Custo
- Escopo



RESTRIÇÕES EM PROJETOS SEGUNDO O PMBOK



NO PASSADO, O FOCO ERA A
QUADRÚPLICE RESTRIÇÃO



ATUALMENTE, O PMBOK CONSIDERA 6 FATORES
QUE LIMITAM AS DECISÕES NO PROJETO
SE UM FATOR SOFRE MUDANÇA, UM OU MAIS
PODERÃO SER IMPACTADOS

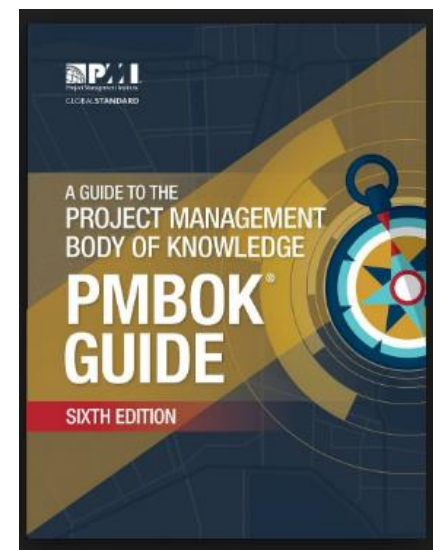
PMBOK – Project Management Book of Knowledge

Grupos de processos:

PMBOK	SWEBOK
Iniciação	Iniciação Definição de Escopo
Planejamento	Planejamento
Monitoramento e controle	Medição Revisão Avaliação
Encerramento	Fechamento

O Guia *PMBOK*®

- ❑ **PMBOK = Conjunto de Conhecimentos em Gerenciamento de Projetos**
 - ❑ é soma dos conhecimentos, técnicas e ferramentas intrínsecas à profissão de gerenciamento de projetos e pertence ao conjunto de profissionais e acadêmicos que o aplicam e desenvolvem.
-
- Publicado pelo PMI desde 1987
 - 6ª. Edição (atual - 2017)



O Guia *PMBOK*®

Identificar o subconjunto do Conjunto de conhecimentos em gerenciamento de projetos que é **amplamente reconhecido** como **boa prática**.

- ✓ **"Identificar"** = **fornecer uma visão geral**, e não uma descrição completa.
- ✓ **"Amplamente reconhecido"** = o conhecimento e as práticas descritas são aplicáveis à maioria dos projetos na maior parte do tempo, e que existe um **consenso geral em relação ao seu valor e sua utilidade**.
- ✓ **"Boa prática"** = existe um consenso de que a aplicação correta dessas habilidades, ferramentas e técnicas pode **aumentar as chances de sucesso** em uma ampla série de projetos diferentes.

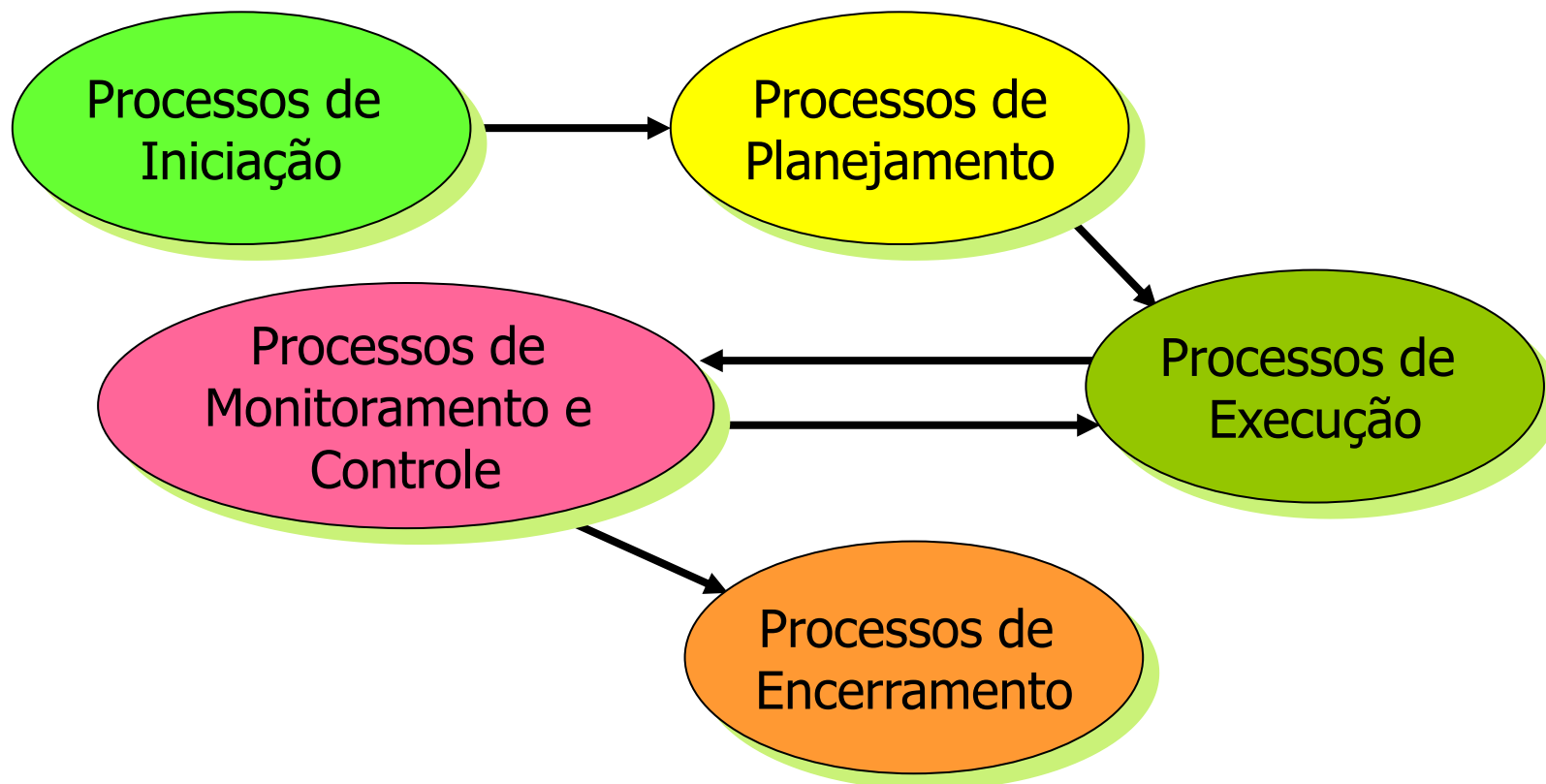


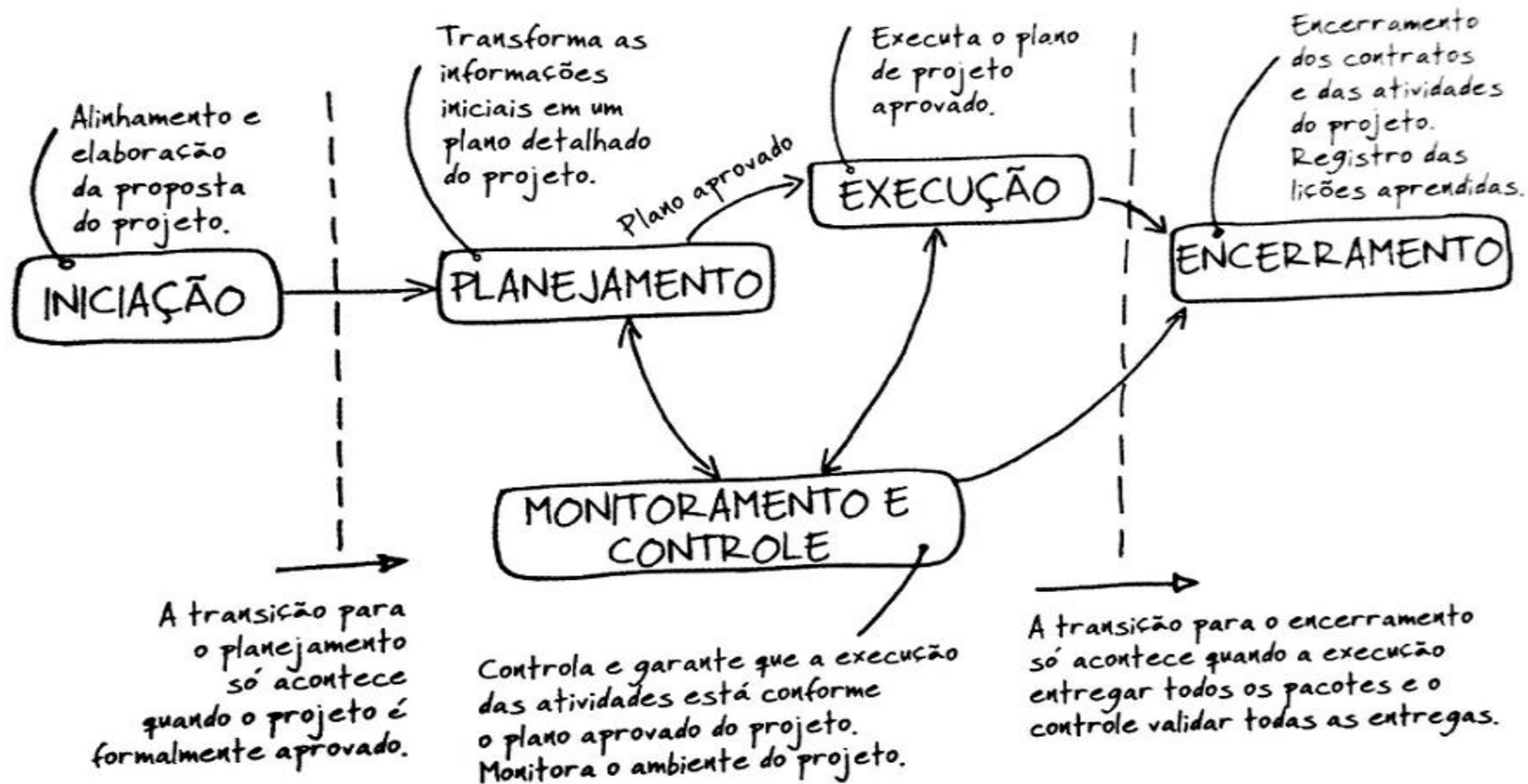
ÁREAS DE CONHECIMENTO

OS PROCESSOS SÃO CLASSIFICADOS POR ÁREAS DE CONHECIMENTO. AS ÁREAS DE CONHECIMENTO SE REFEREM A ATIVIDADES ESPECÍFICAS PARA CADA PROCESSO.



5 Grupos de processos





Condução de Projeto de Software

- Um projeto, mesmo bem planejado pode falhar por vários motivos:
 - erros da equipe,
 - erros no próprio projeto,
 - erros na concepção do processo ou ainda
 - fatores imprevistos.
- Isso decorre principalmente do fato de que os executores do projeto são pessoas, e não máquinas, e que um projeto não é um programa de computador que roda de forma previsível.
- Muitos fatores de incerteza estão envolvidos mesmo nos projetos mais bem planejados e gerenciados.

- Staa (2003) indica que um dos maiores desafios que o gerente de projeto enfrenta no momento do acompanhamento da execução de um projeto é a *indisciplina*.
- Membros da equipe poderão não seguir os padrões, prazos ou prioridades estabelecidos.
- O folclore da ciência da computação apresenta os “gênios” como super-desenvolvedores que não seguem regras, não tem horários preestabelecidos, que são desorganizados e muitas vezes com problemas de higiene pessoal.
- Se não houver disciplina no ambiente de trabalho, estes “heróis” poderão facilmente ressurgir e o projeto será refém deles.

- É responsabilidade do gerente de projeto, então, identificar se há algum tipo de desvio nocivo e motivar os desenvolvedores, conscientizando-os da necessidade de trabalhar em harmonia, lembrando que *harmonia* não significa cada um fazer o que quer, significa todos seguirem as regras que o próprio grupo estabeleceu.

Perfis de gerência

- *Ditador.*
 - É o mais danoso dos três tipos.
 - Ele faz todo o planejamento e estimativas.
 - Ele sozinho determina quem faz o que e quando.
 - Ele gera belíssimos diagramas Gantt que ninguém leva a sério, e usa de força e ameaça para fazer projetos que não estão indo bem voltar para os trilhos, mas nem sempre consegue.
- *Coordenador.*
- *Facilitador.*

Perfis de gerência

- *Ditador.*
- *Coordenador.*
 - Ele ouve os outros e faz as previsões e planejamento em conjunto com a equipe.
 - Ele faz revisões do planejamento periodicamente.
 - Seus diagramas Gantt são levados mais a sério, pois representam um compromisso realista da equipe e não apenas uma determinação do gerente.
 - Seus projetos têm maior chance de terminar no prazo.
 - Este tipo de gestão usualmente funciona bem com métodos prescritivos.
- *Facilitador.*

Perfis de gerência

- *Ditador.*
- *Coordenador.*
- *Facilitador.*
 - Ele apenas agiliza e facilita do trabalho da equipe, mas não toma as decisões.
 - A própria equipe define prazos e o planejamento.
 - Este tipo de gerência funciona bem com os métodos ágeis.

Objetivos do acompanhamento de projetos

- Acompanhar os resultados e desempenhos reais confrontando-os com o plano de desenvolvimento de software.
- Tomar ações corretivas e gerenciá-las até sua conclusão, sempre que resultados ou desempenhos reais desviarem significativamente do que foi estabelecido (estimado) no plano de desenvolvimento de software.
- Assegurar que as alterações nos compromissos de software se deem através de acordo entre as pessoas e os grupos envolvidos.
- Acompanhar processos e meta-processos obtendo indicadores quanto à sua eficácia em instanciar planos e processos.

Técnicas de gerenciamento

- Folha de tempo (timesheet)
- Acompanhamento de problemas
- Registro de artefatos

Folha de tempo

- É uma forma de manter o registro das ações dos desenvolvedores ao longo do tempo, visando estudo para acompanhamento de desempenho e reavaliação de estimativas e do processo em si.
- Deve conter as ações nas quais o desenvolvedor se envolveu ao longo de um dia, como, escrever um método, revisar um diagrama, entrevistar usuário, participar de reunião, resolver problemas pessoais, etc.
 - Não se deve confundir aqui essas ações com a atividade estruturada de um processo, a qual tem artefatos de entrada e saída bem definidos.

Exemplo de folha de tempo

Projeto:						
Atividade de projeto:						
Funcionário:						
Data:						
Resumo das ações do dia:						
Lista de problemas encontrados:						
Ações						
Início	Fim	Natureza	Descrição	Artefato alterado	Artefatos consultados	FAP

Ficha de acompanhamento de problemas

Acompanhamento de problemas

- Uma das grandes diferenças entre o processo de desenvolvimento de software e outros processos de engenharia é que ele usualmente não pode ser totalmente definido a priori.
- Durante este processo, riscos e dúvidas surgirão e com eles, *problemas*, que precisam ser resolvidos e rastreados.
- Além da garantia de solução dos problemas, um controle eficiente de problemas pode ser útil também para identificar a causa dos problemas, desta forma eles podem passar a ser evitados no futuro.
- Além disso, essa análise também poderá identificar problemas no próprio processo, que então poderá ser ajustado.

A identificação inicial de um problema normalmente tem duas fontes: os desenvolvedores e os usuários. Quando o problema é apontado pelo usuário, existem as seguintes possibilidades:

- *Trata-se de um problema urgente que deve ser resolvido de forma emergencial.*
 - Neste caso, ele é imediatamente repassado a uma equipe, que vai parar o que estiver fazendo para achar uma solução para o problema (esta não pode ser a regra para tratar todos os problemas, porém).
- *Trata-se de um problema real, mas não urgente.*
 - Neste caso o problema vai gerar uma entrada na lista de problemas conhecidos da versão corrente, e também na lista de modificações solicitadas. Uma próxima versão do artefato vai possivelmente resolver o problema.
- *Trata-se de um problema real, porém já resolvido numa versão mais atual do que a que o usuário possui.*
 - Neste caso, o usuário deve ser orientado a atualizar sua versão do software.
- *Não se trata de um problema real.*
 - Neste caso o usuário deve ser orientado sobre como proceder ou como compreender o sistema.

FAP - *Folha de Acompanhamento de Problema*

- É um documento ou entrada em um sistema que vai indicar um problema identificado, sua origem (cliente ou desenvolvedores), sua localização (quem ou qual setor da empresa está com o problema sob sua responsabilidade) e seu estado.
- Usualmente, os estados possíveis de um problema são:
 - Aguardando análise.
 - Em análise.
 - Aguardando solução.
 - Em solução.
 - Aguardando informação complementar.
 - Aguardando aprovação.
 - Aprovado/resolvido.

Registro de Artefatos

- O processo de acompanhamento de projeto e controle de problemas depende de forma crucial do sistema gerenciador de configuração, ou controle de versões.
- O gerente deve se certificar de que todos os membros da equipe sigam corretamente a política de controle de versão estabelecida na empresa.
- Uma política típica é que os desenvolvedores somente tenham acesso a um artefato quando liberado pelo seu proprietário.
- Eles podem trabalhar na cópia do artefato até realizar a tarefa a que se propuseram.
- Depois, submetem o artefato modificado ao controle de qualidade, usualmente na forma de testes.
- Se aprovado, salvam uma nova versão do artefato no sistema de gerenciamento de configuração.

Medição em Engenharia de Software

- Um processo de gerência, para ser mais eficaz, precisa se basear em medições.
- Como saber se a tarefa não está sendo feita se não se tem uma medida para chegar a essa conclusão?
- Como saber que a qualidade é inaceitável?
- Então, de uma maneira ou de outra, o gerente de projeto vai acabar se envolvendo com a atividade de medição de software, para a qual terá que aplicar uma ou mais métricas.

Termos usados

- *Medida:*
 - é um valor obtido para alguma dimensão do software.
- *Métrica:*
 - é a escala na qual os valores de uma medida são tomados.
- *Medição:*
 - é o processo de obtenção de medidas.

Qualidades de uma boa métrica

- Ser ***simples***, ou seja, ter uma definição curta e fácil de ser compreendida.
- Ser a mais ***objetiva*** possível, isto é, não deve depender de opiniões e, se duas pessoas avaliarem o mesmo produto usando essa métrica, devem obter o mesmo resultado.
- Ser ***facilmente obtida***, isto é, o custo para efetuar a medição deve ser razoavelmente baixo.
- Ser ***válida***, isto é, a métrica deve indicar um valor que seja útil e que seja efetivamente representativo da grandeza que se pretende medir.
- Ser ***robusta***, isto é, pequenas mudanças no produto devem produzir mudanças proporcionais na medida; apenas grandes mudanças no produto podem produzir grandes mudanças na medida.

Métricas diretas

- Custo financeiro.
- Esforço em desenvolvedor-hora.
- Linhas de código (SLOC).
- Velocidade de execução em segundos.
- Memória em megabytes.
- Número de defeitos localizados (total ou relativo ao número de KSLOC).
- Complexidade ciclomática.

Métricas indiretas (dependem de definição operacional)

- Funcionalidade.
- Qualidade.
- Complexidade.
- Eficácia.
- Confiabilidade.
- Manutenibilidade.
- Usabilidade.

Agrupamento das métricas em função de sua utilidade para o gerente

- *Métricas de produtividade,*
 - como custo, esforço (em AFP ou UCP) e KSLOC, que podem ser usadas para verificar o andamento do projeto e possíveis desvios.
- *Métricas de qualidade,*
 - como número de defeitos, qualidade, eficiência, confiabilidade e manutenibilidade. São usadas para avaliar se o produto satisfaz critérios de aceitação para uso por parte do cliente e também critérios internos, que afetam a eficiência da equipe.
- *Métricas técnicas.*
 - São outros aspectos ligados ao produto, não necessariamente ligadas nem a qualidade, nem a produtividade, mas a aspectos inerentes do sistema, como complexidade ciclomática, modularidade, paralelismo, distribuição, etc.

Revisão e Avaliação

- É importante que as reuniões de revisão e avaliação, para que sejam efetivas, sejam planejadas com antecedência.
- As pessoas envolvidas devem ser comunicadas, local adequado reservado, e mais importante, os objetivos da reunião devem ser claramente comunicados.
- Quando se fala em objetivos, aqui, deve-se entender isso no sentido de definir quais serão os *artefatos de saída* da reunião.
- Pode-se ver então que a reunião de revisão e avaliação é também vista como uma atividade de projeto.

Sugestões de itens de pauta para reunião de revisão e avaliação

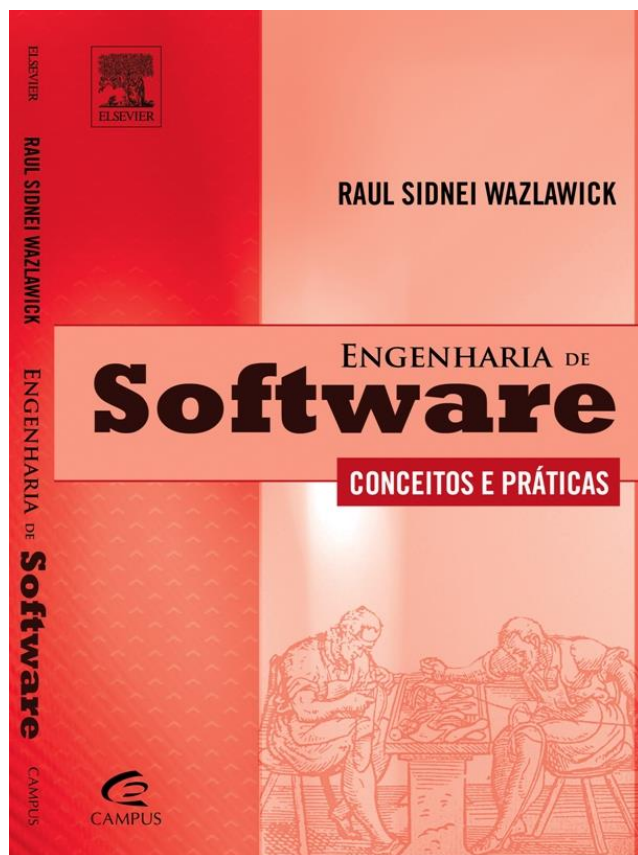
- Os principais marcos de projeto (*milestones*) alcançados.
- Os desvios que eventualmente tenham ocorrido em relação ao plano de desenvolvimento do projeto.
- Modificações significativas na alocação de esforço, ou seja, se mais (ou menos) tempo do que o previsto foi dedicado a alguma atividade.
- Modificações significativas em termos de despesas, ou seja, se as atividades consumiram mais (ou menos) itens de orçamento do que o previsto.
- Mudanças no escopo estimado de trabalho, ou seja, se houve alguma modificação em termos das funcionalidades ou outros aspectos que inicialmente consistiam em um objetivo da fase ou ciclo e que foram mudados, removidos ou acrescentados.
- Mudanças na métrica de qualidade, ou seja, se por alguma razão a forma de medir a qualidade do trabalho foi mudada.
- *Status* dos riscos de projeto, ou seja, quais os riscos que continuam sendo muito importantes, quais riscos tiveram sua importância alterada ao longo da fase ou ciclo (especialmente caso a alteração tenha sido no sentido de um risco ficar mais importante do que era anteriormente).
- Novos riscos identificados.
- Problemas relevantes que tenham surgido e ainda não tenham sido resolvidos.
- Andamento de eventuais ações que tenham sido determinadas em reuniões anteriores.
- Lições aprendidas para projetos futuros.

Fechamento

- O fechamento ou encerramento de um projeto (ou de uma fase de um projeto) tem como objetivo formalizar a entrega do produto e a sua aprovação pelo cliente.
- Será uma reunião ainda mais formal do que as reuniões periódicas de avaliação do projeto e, possivelmente, deverá envolver um número maior de interessados.

Pontos a observar em uma reunião de fechamento

- Se todos os objetivos iniciais do projeto foram alcançados.
- Se houve algum desvio importante ao longo do projeto em relação ao seu plano.
- Se houve alguma mudança significativa em relação ao esforço previsto e executado (o que poderá servir de entrada para calibragem de métricas de esforço).
- Se houve alguma mudança significativa em relação aos recursos alocados e consumidos.
- Se houve mudanças de escopo importantes.
- Os resultados finais das métricas de qualidade e, eventualmente, sua evolução.
- Riscos de operação, caso tenham sido identificados.
- Problemas relevantes que tenham surgido e ainda não tenham sido resolvidos.
- Lições aprendidas para projetos futuros.



Planejamento de Projeto

Engenharia de Software: Conceitos e Práticas

Prof. Raul Sidnei Wazlawick

UFSC-CTC-INE

Elsevier, 2013

Seleção de Projetos

- Normalmente existe mais de uma possibilidade de projeto e nem sempre todos os projetos podem ser desenvolvidos.
- Questões:
 - A empresa tem competência para desenvolver esse tipo de produto?
 - A empresa está dando conta dos projetos atuais, ou seja, tem folga operacional para assumir um novo projeto?
 - O cliente é conhecido e confiável?
 - O produto dará um bom retorno financeiro?

Do ponto de vista do cliente

- Ponderações:
 - Retorno financeiro em relação ao investimento.
 - Grau de incremento da participação da empresa no mercado.
 - Melhoria da imagem da empresa.
 - Utilização de capacidade ociosa.
 - Aquisição de novas tecnologias.
 - Etc.

Termo de Abertura (*project charter*)

- Objetivo e justificativa do projeto.
- Descrição em alto nível do projeto.
- Requisitos de alto nível que satisfazem os principais interessados.
- Nomeação do gerente de projeto e definição do nível de autoridade conferida.
 - Pode usar os recursos sem aprovação superior? Pode contratar pessoal?
- Cronograma de marcos (*milestones*) resumido.
- Definição dos papéis e responsabilidades das partes interessadas.
- Organização funcional do projeto.
- Premissas ou hipóteses.
 - São perguntas para as quais ainda não se tem resposta, mas que são aceitas, a princípio, para iniciar o projeto. Por exemplo, haverá um especialista disponível na tecnologia X?
- Restrições.
- Estudo de viabilidade (*business case*) indicando o retorno financeiro e não financeiro previsto.
- Orçamento previsto em linhas gerais.

Declaração de Escopo

- Inicialmente o planejador de um projeto deve estabelecer quais são os objetivos finais do projeto.
 - O produto nem sempre é apenas o software funcionando.
 - Outros elementos são usualmente necessários e desejáveis.
- Sem definir claramente onde o projeto vai chegar é muito difícil estabelecer um bom plano.
- O objetivo de um projeto (e também das iterações) deve ser sempre um conjunto de **artefatos**.

Elementos de uma declaração de escopo

- *Descrição do produto do projeto.*
 - Embora o termo de abertura já contenha uma definição do produto em alto nível, a declaração de escopo deverá refinar esta descrição.
- *Principais entregas do projeto.*
 - Devem ser definidas as principais entregas do projeto, ou seja, os momentos em que o cliente estará recebendo algum tipo de *deliverable* dos desenvolvedores.
 - Normalmente trata-se de versões implementadas do sistema, mas essa lista poderá incluir também outros itens como projeto, manuais, discos de instalação, treinamento, etc.
- *Objetivos do projeto.*
 - São os itens quantificáveis que serão usados para determinar se o projeto foi um sucesso ou não.
 - Os objetivos do projeto devem incluir pelo menos métricas relacionadas ao prazo, custo e qualidade do produto.
 - O estabelecimento de objetivos não quantificáveis (por exemplo, “cliente satisfeito”, ou “sistema fácil de usar”) representa um fator de alto risco para a determinação do sucesso do projeto.
- *Critérios de aceitação do produto.*
 - Define-se o processo e critérios para que o produto, como um todo, seja aceito e o projeto finalizado.

Planejamento de Projeto com Ciclos Iterativos

- O objetivo do planejamento de projeto é criar um *plano* para o projeto como um todo.
- Entre outras coisas, é importante que o responsável por este planejamento utilize as melhores ferramentas possíveis para avaliar a quantidade de esforço a ser despendido no projeto.
- Tal estimativa poderá dar origem tanto ao cronograma geral do projeto quanto à estimativa de custo total do projeto.

Atividades do planejamento geral

- Estimar o *esforço total* para realizar o projeto.
- Em função do esforço total, calcular o *tempo linear* necessário e *tamanho médio da equipe*.
- Estimar a *duração e esforço nas diferentes fases do projeto*.
- Estimar a *duração e número dos ciclos iterativos*.

Estimação da Duração e Esforço nas Diferentes Fases do Projeto

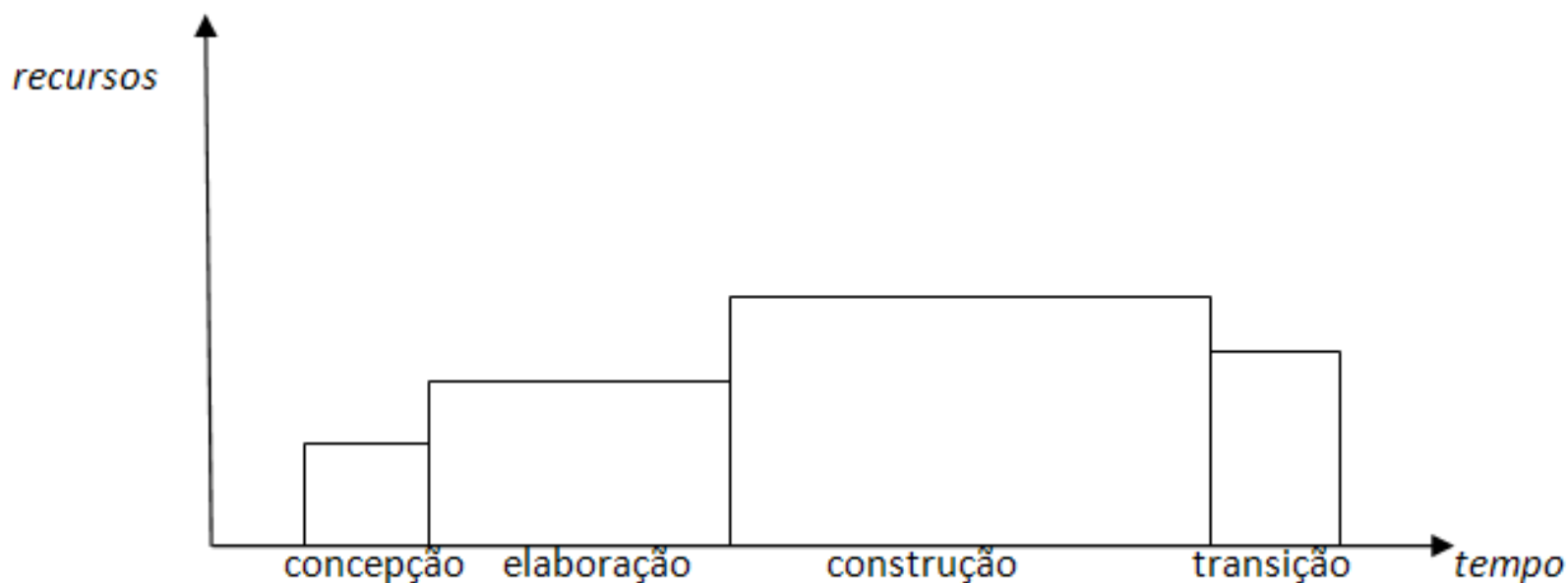


Figura 6-1: Perfil de duração e esforço típicos para um projeto usando UP.

Duração e esforço no UP

- Concepção: 10% do tempo e 5% do esforço.
- Elaboração: 30% do tempo e 20% do esforço.
- Construção: 50% do tempo e 65% do esforço.
- Transição: 10% do tempo e 10% do esforço.

Exemplo

- Esforço total: 40 desenvolvedor-mês.
- Duração linear: 8,5 meses.
- Duração das fases:
 - Concepção: 10% de 8,5, ou seja, cerca de 0,85 meses.
 - Elaboração: 30% de 8,5, ou seja, cerca de 2,55 meses.
 - Construção: 50% de 8,5, ou seja, cerca de 4,25 meses.
 - Transição: 10% de 8,5, ou seja, cerca de 0,85 meses.

Exemplo

- Esforço total: 40 desenvolvedor-mês.
- Duração linear: 8,5 meses.
- Duração das fases:
 - Concepção: 0,85 meses.
 - Elaboração: 2,55 meses.
 - Construção: 4,25 meses.
 - Transição: 0,85 meses.
- Tamanho da equipe nas fases:
 - Concepção: 5% de 40, ou seja, 2 desenvolvedor-mês, o que dividido por 0,85 meses dá cerca de 2,35 desenvolvedores em média na fase.
 - Elaboração: 20% de 40, ou seja, 8 desenvolvedor-mês, o que dividido por 2,55 meses dá cerca de 3,13 desenvolvedores em média na fase.
 - Construção: 65% de 40, ou seja, 26 desenvolvedor-mês, o que dividido por 4,25 meses dá cerca de 6,11 desenvolvedores em média na fase.
 - Transição: 10% de 40, ou seja, 4 desenvolvedor-mês, o que dividido por 0,85 meses dá cerca de 4,7 desenvolvedores em média na fase.

Alterações do perfil típico

- Se for necessário um tempo mais longo para estabelecer o projeto, achar financiadores, fazer pesquisa de mercado ou construir provas de conceito, a fase de concepção deve ser tornada mais longa.
- Se houver altos riscos técnicos ou de pessoal ou se houver restrições de desempenho importantes e nenhuma arquitetura prévia definida, então a fase de elaboração deve ser alongada, porque serão necessários mais ciclos de elaboração para definir a arquitetura e/ou mitigar os riscos conhecidos.
- Se essa não for a primeira geração do produto (pode ser um ciclo de evolução) e se não serão feitas maiores alterações na arquitetura então as fases de concepção e elaboração podem ser encolhidas.
- Se o objetivo for atingir o mercado rapidamente por causa de concorrentes ou porque se está criando o mercado, então a fase de construção pode ser encolhida e a fase de transição aumentada. Assim, versões executáveis serão liberadas mais cedo e gradativamente no mercado.
- Se houver necessidade de uma transição complicada, como por exemplo, substituir um sistema em funcionamento sem interromper os serviços, ou no caso de domínios que exigem certificações ou regulamentos a serem avaliados (medicina, aeronáutica, etc.), a fase de transição deve ser aumentada.

Estimação da Duração e Número dos Ciclos Iterativos

- Equipes pequenas com até 5 pessoas poderão fazer o planejamento juntos numa manhã de segunda-feira, executar o trabalho ao longo da semana e gerar uma *release* na sexta.
- Equipes com mais de 20 pessoas precisarão de mais tempo para distribuir atividades e sincronizar as atividades, até porque a carga de trabalho naturalmente será bem maior. Além disso, a geração da *release* tomará mais tempo, pois haverá um volume maior de partes a serem integradas. Assim, neste caso, uma iteração de 3 a 4 semanas seria mais recomendável.
- Equipes com mais de 40 pessoas precisarão trabalhar em um ambiente muito mais formal e com mais documentação intermediária de forma que o fluxo de informação será naturalmente mais lento. Desta forma, um ciclo de 6 a 8 semanas seria recomendável neste caso.

Outros fatores que afetam a duração de um ciclo

- Quanto mais automatização no processo e geração de código e no ambiente de desenvolvimento em geral, mais curtos poderão ser os ciclos.
- Quanto mais familiaridade a equipe tiver com o Processo Unificado e com as técnicas de análise e *design*, mais curtos poderão ser os ciclos.
- Quanto mais crítico for o fator “qualidade” no desenvolvimento, quanto mais críticas as revisões e testes que precisam ser feitas, mais longos deverão ser os ciclos.

Número de iterações

- O número de iterações de um projeto dependerá do tempo linear a ser despendido, especialmente nas fases de elaboração e construção, dividido pelo tamanho planejado para as iterações.
- Por exemplo, um projeto com iterações de duas semanas, cujas fases de elaboração e construção devem durar 6 meses no total (24 semanas) terá 12 ciclos de elaboração e construção.

Definição dos Marcos ou Entregas

- Uma vez definido o tamanho das iterações, tamanho da equipe em cada fase e a duração de cada fase (em número de ciclos), o planejador deverá retomar a declaração de escopo para definir os marcos de projeto e as datas de entregas.
- O Processo Unificado já estabelece marcos padrão ao final de cada fase, mas convém que no plano de projeto esses marcos, bem como outros momentos importantes do projeto sejam claramente identificados.

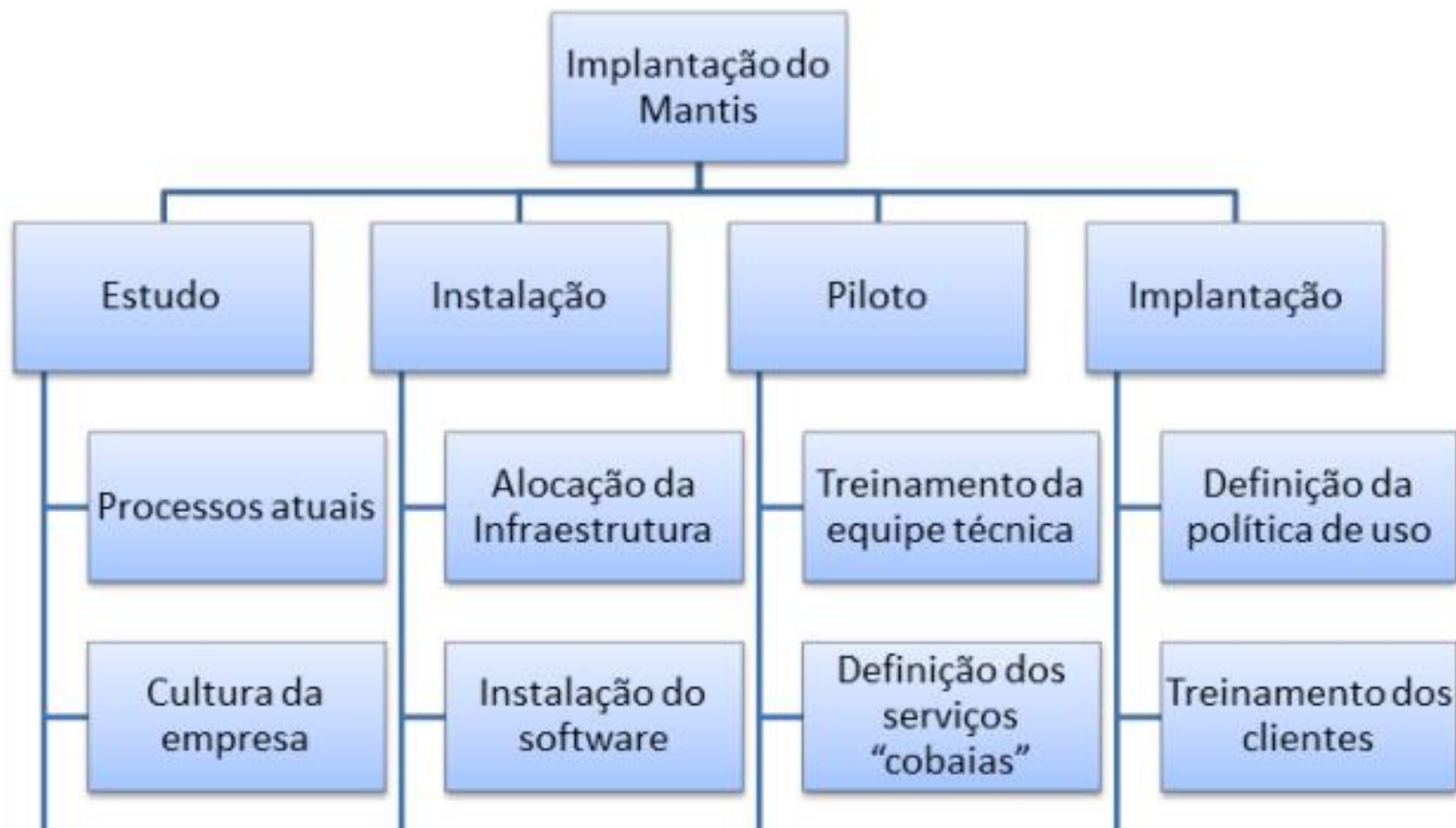
WBS - Estrutura Analítica do Projeto

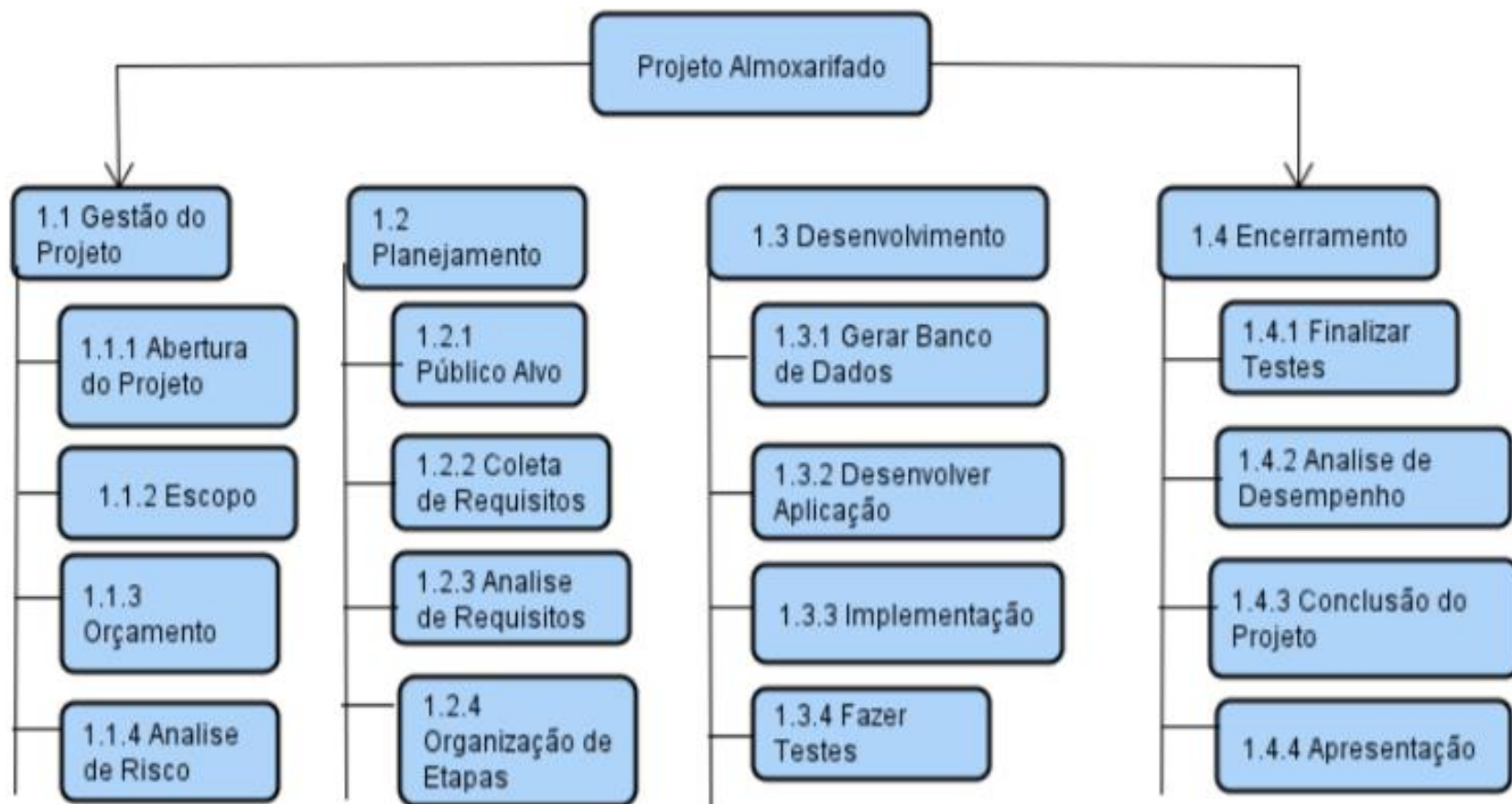
- A estrutura analítica do projeto (WBS ou *Work Breakdown Structure*) é uma estrutura que apresenta as atividades / ou entregas devem ser executadas para atingir os objetivos determinados.
- Se for usado um ciclo de vida prescritivo, os *workflows* vão indicar quais as atividades a serem executadas e quais as dependências entre elas.
 - Dependendo do processo adotado o *workflow* poderá até indicar formas de estimação de esforço para cada atividade.
- Se for usado um método ágil, recomenda-se que a equipe decida sobre as atividades a serem desenvolvidas.
 - Isso não impede que equipes usando métodos ágeis se baseiem em *workflows* existentes, se o grupo entender que isso poderá ser útil ao projeto.

- A WBS é uma estrutura exaustiva, ou seja, ela deve incluir todas as atividades necessárias para a execução do projeto ou iteração.
- A WBS poderá ser estruturada com uma árvore, sendo que atividades podem ser aglutinadas ou detalhadas e estabelecendo uma árvore de decomposição entre elas.
- É muito importante que o planejador do projeto planeje artefatos de saída e não meramente ações.
- Estilos de WBS que preveem diferentes estágios de um artefato (por exemplo, versão inicial, versão intermediária e versão final), devem caracterizar exatamente o que esperam de cada uma destas versões.

Regra 8-80

- **Nenhuma atividade terminal deve durar mais de 80 horas (duas semanas, ou dez dias de trabalho ideais), nem menos de 8 horas (um dia de trabalho ideal).**
- Métodos como XP são ainda mais restritivos em relação ao tamanho das atividades, pois exigem que sua duração seja de 1 a 3 dias ideais de trabalho, ou seja, 8 a 24 horas.
- A WBS deve ser organizada, precisa e pequena o suficiente para que ela possa servir de base para a gerência do projeto durante a iteração sem ser um estorvo.





Regra dos níveis

- A estruturação de uma boa WBS não deve ter mais de 3 ou 4 níveis de decomposição de atividades.

Regra do número de atividades

- Ainda mais, o número total de pacotes de trabalho em uma WBS não deve ultrapassar o limite de 200 elementos, embora 100 já seja considerado um número muito alto.

Regra dos 100%

- A regra dos 100% estabelece que uma WBS deve incluir 100% de todo o trabalho que deve ser feito na iteração.
 - Nenhum artefato será produzido se não estiver definido como saída de alguma das atividades da WBS e nenhuma atividade deixará de produzir algum artefato de saída.
- A regra dos 100% vale em todos os níveis da hierarquia de decomposição da WBS.
 - Além disso, quando uma atividade se decompõe em subatividades, o trabalho definido pela atividade será exatamente igual a 100% do trabalho definido nas subatividades (sempre em termos de artefatos de saída).

Os 10 mandamentos da WBS

- Cobiçarás a WBS do próximo.
- Explicitarás todos os subprodutos, inclusive os necessários ao gerenciamento do projeto.
- Não usarás os nomes em vão.
- Guardarás a descrição dos pacotes de trabalho no dicionário da WBS.
- Decomporás as atividades até o nível de detalhe que permita o planejamento e controle do trabalho necessários para a entrega do produto.
- Não decomporás em demasia, de forma que o custo/tempo de planejamento e controle não traga o benefício correspondente.
- Honrarás o pai.
- Decomporás de forma que a soma dos subprodutos das atividades filho seja igual a 100% do subproduto da atividade pai.
- Não decomporás somente um subproduto.
- Não repetirás o mesmo elemento como componente de mais de um subproduto.

Identificação dos Responsáveis por cada Atividade

- Um *workflow* usualmente define que o responsável por uma atividade é um papel, ou seja, uma pessoa com uma ou mais habilidades desejáveis.
 - Quando uma iteração for planejada a partir deste *workflow* deve-se então atribuir as atividades a pessoas reais que atendam a este papel desejado sempre que possível.
- Cada atividade da WBS deverá ser atribuída a um ou mais responsáveis.
 - Essas atribuições poderão ter efeito sobre o cronograma de projeto, pois embora certas atividades possam ser executadas em paralelo, não é possível fazê-lo assim caso estejam atribuídas ao mesmo responsável.

Identificação dos Recursos Necessários e Custo

- É possível que a maioria das atividades a serem executadas, além de recursos humanos (responsáveis e participantes) também tenham recursos físicos consumíveis ou não consumíveis a serem alocados.
- No momento do planejamento da iteração é necessário prever e alocar o uso destes recursos.
- O custo de uma atividade individual será então o custo das pessoas que se dedicam a ela somado ao custo dos recursos alocados.


Identificação das Dependências entre Atividades

- São dadas em função do *workflow* ou identificadas caso a caso pela equipe de planejamento.
 - Usualmente estas dependências existem porque as entradas de uma atividade são as saídas de outra atividade.
- A partir da estruturação das atividades o planejador do projeto deverá estimar os tempos necessários para a execução de cada atividade.
 - Usualmente é difícil estimar tempos com grande precisão.
 - Trabalha-se então com o *timeboxing* da iteração.
- Deve-se determinar as sequências de atividades mais difíceis primeiro, e alocar desenvolvedores a elas.
- Depois se distribui o tempo restante para as outras atividades. Eventuais erros para mais ou para menos nas estimativas podem compensar-se mutuamente.

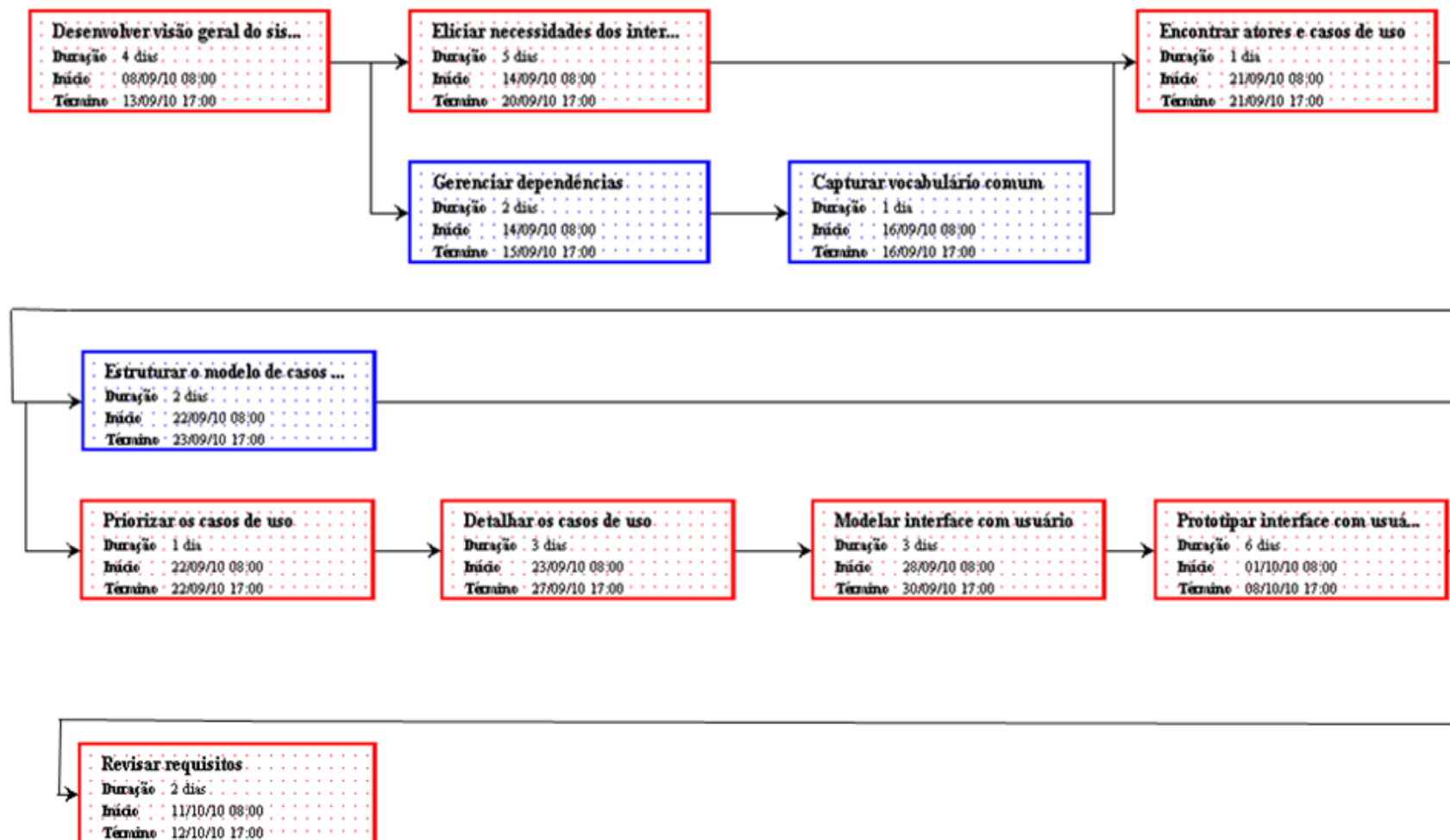
Rede PERT

- O grafo de dependências entre atividades com a duração prevista para cada atividade constitui-se na *rede PERT* do projeto ou iteração.

Exemplo de WBS

		Nome	Duração	Início	Término	Predecessoras
1		Desenvolver visão geral do sistema	4 dias	08/09/10 08:00	13/09/10 17:00	
2		Eliciar necessidades dos interessados	5 dias	14/09/10 08:00	20/09/10 17:00	1
3		Gerenciar dependências	2 dias	14/09/10 08:00	15/09/10 17:00	1
4		Capturar vocabulário comum	1 dia	16/09/10 08:00	16/09/10 17:00	3
5		Encontrar atores e casos de uso	1 dia	21/09/10 08:00	21/09/10 17:00	4;2
6		Estruturar o modelo de casos de uso	2 dias	22/09/10 08:00	23/09/10 17:00	5
7		Priorizar os casos de uso	1 dia	22/09/10 08:00	22/09/10 17:00	5
8		Detalhar os casos de uso	3 dias	23/09/10 08:00	27/09/10 17:00	7
9		Modelar interface com usuário	3 dias	28/09/10 08:00	30/09/10 17:00	8
10		Prototipar interface com usuário	6 dias	01/10/10 08:00	08/10/10 17:00	9
11		Revisar requisitos	2 dias	11/10/10 08:00	12/10/10 17:00	6;10

Rede Pert



Caminho Crítico

- Um conceito importante no diagrama PERT é o **caminho crítico**, que consiste no mais longo caminho que leva do início ao fim do projeto ou iteração.
- Esse caminho crítico é importante porque se qualquer atividade prevista nele atrasar, por algum motivo, todo o projeto irá atrasar.
- Este é o caminho sem folga.

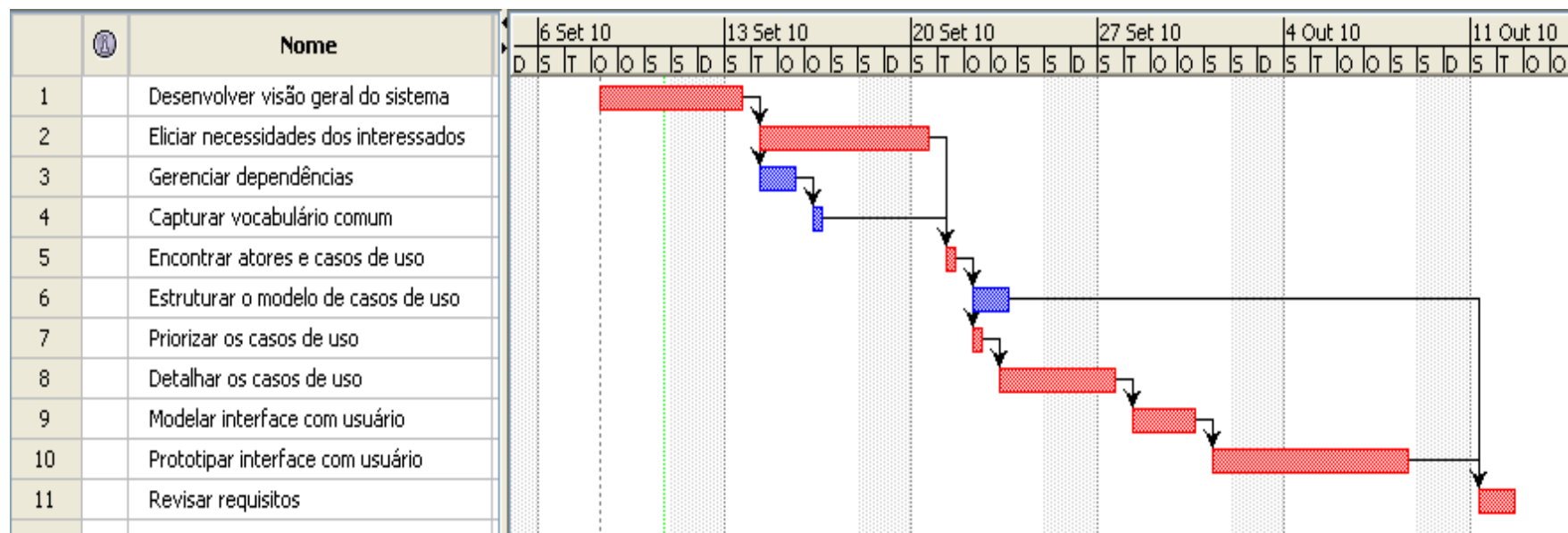
Como recuperar um atraso no caminho critico

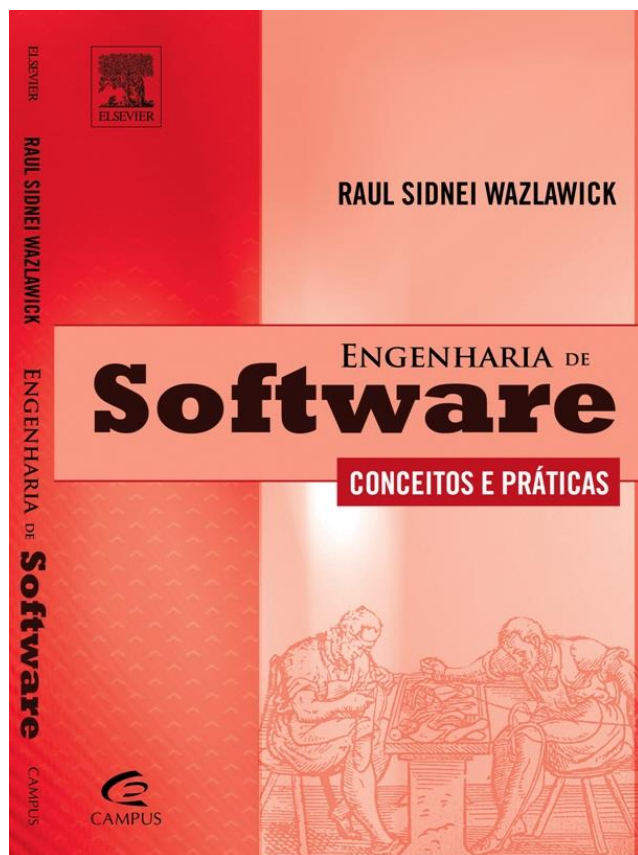
- Aumentar a jornada da equipe (o que não pode se transformar em rotina).
- Aumentar o tamanho da equipe (o que pode causar transtornos de gerência em função da colocação de pessoas novas no projeto, possivelmente com menos experiência).
- Eliminar alguns objetivos (artefatos) ou características de artefatos da iteração.
 - Por exemplo, ao invés de implementar 3 casos de uso, caso haja atrasos, implementa-se apenas 2, deixando para outra iteração a implementação do terceiro).

Cronograma

- O cronograma do projeto usualmente é mostrado em um diagrama Gantt, que consiste em uma visualização do tempo linear transcorrido e a ocorrência das diferentes atividades ao longo deste tempo.

Exemplo





Gerenciamento de Riscos

Engenharia de Software: Conceitos e Práticas

Prof. Ral Sidnei Wazlawick

UFSC-CTC-INE

Elsevier, 2013

Risco

- Todo projeto de desenvolvimento de software apresenta um conjunto de **incertezas** em diferentes graus que podem causar **problemas**.
- Um planejador que não esteja atento aos riscos do projeto não terá **planos** para tratar situações que podem vir a prejudicar ou até inviabilizar todo um projeto.



Modelo de gerenciamento de riscos (SEI)

- *Identificação.*
 - Antes que os riscos possam ser tratados, eles precisam ser identificados.
- *Análise.*
 - Transforma a lista de riscos potenciais em um documento no qual os riscos são priorizados.
- *Planejamento.*
 - O planejamento frente aos riscos permite ao gerente prevenir problemas.
- *Rastreamento ou monitoramento.*
 - Consiste em avaliar, ao longo do projeto, as propriedades do risco baseando-se em métricas.
- *Controle.*
 - Em função de mudanças no *status* de um risco, planos podem ter que ser executados.
- *Comunicação.*
 - É um processo fundamental ao longo de todo um projeto de software, especialmente em relação à prevenção e tratamento de riscos.

Plano de Gerência de Riscos

- Quais são os riscos identificados.
- Uma análise qualitativa ou quantitativa de cada risco, por exemplo, indicando a probabilidade de sua ocorrência e seu provável impacto sobre o projeto, caso ocorra.
- Como a probabilidade do risco ocorrer pode ser reduzida.
- Como o impacto do risco, caso ocorra, pode ser reduzido.
- O que fazer se o risco ocorrer.
- Como monitorar os riscos.

Classificação dos riscos em relação ao conhecimento que se tem sobre eles

- *Riscos conhecidos.*  tratar
 - São aqueles já identificados e para os quais a equipe possivelmente está preparada.
- *Riscos desconhecidos.*  minimizar
 - São aqueles que, se as medidas de identificação adequadas tivessem sido tomadas poderiam ter sido descobertos, mas não foram.
- *Riscos impossíveis de prever.*
 - São aqueles que, mesmo com as melhores técnicas de identificação não teriam sido identificados.

Identificação de Riscos

- Elementos que compõe um risco:
 - Uma *causa*,
 - na forma de uma condição incerta.
 - Um *problema*
 - que pode ocorrer em função da causa, o qual vai provocar um *efeito* ou impacto em um ou mais objetivos do projeto ou iteração.

Riscos nas diferentes fases do UP

- *Concepção:*
 - riscos de requisitos e de negócio.
- *Elaboração:*
 - riscos de tecnologia e arquitetura de sistema.
- *Construção:*
 - riscos de programação e teste de sistema.
- *Transição:*
 - riscos de utilização do sistema no ambiente final.

Técnicas para identificação de riscos

- Uso de ***checklists*** pré-definidos com possíveis riscos.
- Reuniões e ***brainstormings*** com gerente e equipe de projeto com experiência em outros projetos.
- **Análise de cenários** e lições aprendidas em projetos anteriores com contexto semelhante.

Fontes de riscos

- Tecnologia (hardware e software).
- Pessoas (cliente, equipe, mercado, etc.).
- Projeto (atrasos, custos exagerados, etc.).

Riscos tecnológicos

- Estão relacionados a todas as incertezas referentes a como a equipe será capaz de lidar com a tecnologia necessária para realizar o projeto.
 - Quanto menos experiência nessas tecnologias, maiores serão os riscos.
- Projetos que envolvam diferentes sistemas de software e hardware também frequentemente enfrentarão problemas de compatibilidade.
- Outro ponto que pode oferecer risco tecnológico a um projeto é a questão da obsolescência.
 - Quão rápido as tecnologias usadas ou produzidas serão suplantadas por outras mais eficientes?

Riscos de pessoas

- *Riscos de pessoal.*
 - Perder uma pessoa da equipe de forma permanente ou temporária pode ter um impacto grande na medida em que essa pessoa for insubstituível.
- *Riscos de cliente.*
 - Até que ponto o cliente se manterá interessado no projeto?
 - O cliente estará disponível para esclarecer requisitos e realizar testes?
- *Riscos de negócio.*
 - A empresa poderá não ter a habilidade necessária para vender o produto, ou ainda, ter essa habilidade, mas o produto não ter efetivamente apelo comercial.
- *Riscos legais.*
 - Existem problemas ou possibilidade de litígio?
 - Uso de material protegido por direitos autorais?
 - Necessidade de celebração de contrato com terceiros?
 - Normas e leis específicas em outros estados ou países?

Riscos de projeto

- *Riscos de requisitos.*
 - A equipe, por ser inexperiente, pode não ter sido capaz de identificar corretamente os requisitos do projeto, o que causará problemas no decorrer do mesmo.
 - Requisitos poderão ser insuficientes, excessivos ou incorretos.
 - Requisitos podem ser naturalmente instáveis devido a características do próprio projeto.
- *Riscos de processo.*
 - O modelo de processo escolhido é adequado às características do projeto?
 - A equipe tem experiência com o processo?
 - O gerente tem experiência em projetos anteriores?
- *Riscos de orçamento.*
 - A verba necessária para o projeto está garantida até que ponto?
 - Os custos foram corretamente previstos em projetos passados?
- *Riscos de cronograma.*
 - É possível que prazos sejam alterados?
 - É possível que a ordem em que as funcionalidades são entregues possa mudar?
 - O planejador também deve saber em que grau a equipe mostrou-se capaz de ater-se ao cronograma em projetos passados.

Comunicação de Riscos

- A comunicação na área de gerenciamento de riscos é fundamental, inicialmente nas atividades de identificação.
- Usualmente o pessoal técnico já tem consciência dos riscos que um projeto vai enfrentar, mas se não forem incentivados a comunicar essa informação, não o farão.
- Assim, uma reunião de planejamento inicial com toda a equipe é fundamental para que riscos sejam levantados e avaliados pela equipe como um todo.

- Poucas crises, estatisticamente, ocorrem de surpresa.
- Normalmente os problemas vão crescendo lentamente até que se transformem em uma crise.
 - Isso não é diferente em projetos de software.
- O que vai fazer a diferença entre uma crise que cresce nas sombras até atingir um tamanho que coloque o projeto a perder e um problema que pode ser mitigado em seu início, é a capacidade da equipe de perceber e comunicar justamente esse início.

Problemas de comunicação que podem levar a crises

- Declaração inadequada.
- Opção discutível.
- Boatos.
- Vazamento de informação.
- Política interna.
- Falta de integração.
- Atendimento frágil.
- Questões pendentes.
- Falta de recursos.
- *Briefing* obscuro.
- Dados conflitantes.
- Postergar decisões.
- Funcionários insatisfeitos.
- Falta de orientação interna.

- Um dos fatores em comunicação de risco que deve estar sob atenção do gerente é que diferentes interessados terão diferentes percepções sobre o risco.
- É atribuída a Einstein a frase “o maior problema da comunicação é a ilusão de que ela ocorreu”.
- Assim, não basta apresentar a lista de riscos aos interessados e esperar que todos entendam o que devem fazer.
- É necessário, em muitos casos, enfatizar as perdas que podem ocorrer caso o risco não seja adequadamente prevenido.
 - Os desenvolvedores estarão mais interessados nos riscos técnicos que podem dificultar seu trabalho, causando frustração.
 - Os clientes estarão mais interessados nos riscos que poderão atrasar o cronograma ou afetar os custos do projeto.
 - Os usuários estarão interessados em riscos que envolvem a qualidade do sistema.
- Assim, comunicar os riscos corretamente às partes interessadas é também um processo que deve ser cuidadosamente pensado e executado pelo gerente de projeto.