

# **PLANEJAMENTO DE PROJETO**

**INE 5419 – Engenharia de Software II**

**Prof. Raul Sidnei Wazlawick**

**UFSC-CTC-INE**

**2012.1**

# CONTEÚDO

- Seleção de projetos
- Termo de abertura
- Declaração de escopo
- Planejamento de projeto com ciclos iterativos
  - Duração e esforço das fases
  - Duração e número de ciclos iterativos
  - Número de iterações
  - Marcos ou entregas
- Planejamento de iteração
  - WBS – estrutura analítica
  - Responsáveis
  - Recursos
  - Dependências
  - Cronograma



## 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

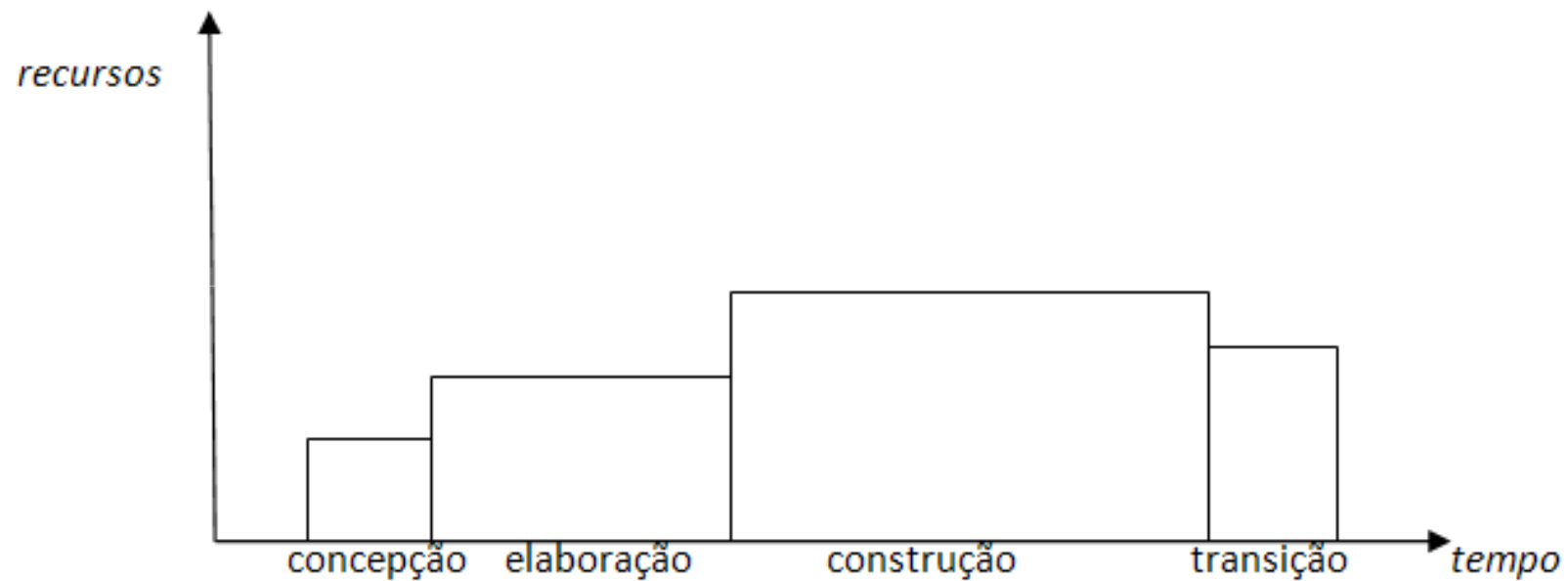


Figura 6-1: Perfil de duração e esforço típicos para um projeto usando UP.



## 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.



## RETOMANDO O EXEMPLO

Tabela 6-1: Esforço e duração de um projeto típico por fase do UP.

Fase	Duração (semanas)	Duração arredondada	Número médio de desenvolvedores	Número de desenvolvedores arredondado
Concepção	3,5	3	2,35	3
Elaboração	10,2	10	3,13	3
Construção	17	18	6,11	6
Transição	3,5	3	4,7	5
<b>Total</b>	<b>34,2</b>	<b>34</b>		



Tabela 6-2: Um exemplo de plano de projeto simplificado com definição de entregas.

Fase	Prazo (semana)	Desenvolvedores	Entregas
Concepção	3	3	Modelo de casos de uso preliminar para revisão.
	5	3	20% dos casos de uso de maior risco incorporados na arquitetura.
	7	3	40% dos casos de uso de maior risco incorporados na arquitetura.
	9	3	60% dos casos de uso de maior risco incorporados na arquitetura.
	11	3	80% dos casos de uso de maior risco incorporados na arquitetura.
	13	3	100% dos casos de uso de maior risco incorporados na arquitetura.
Construção	15	6	11% dos demais casos de uso incorporados na arquitetura.
	17	6	22% dos demais casos de uso incorporados na arquitetura.
	19	6	33% dos demais casos de uso incorporados na arquitetura.
	21	6	44% dos demais casos de uso incorporados na arquitetura.
	23	6	55% dos demais casos de uso incorporados na arquitetura.
	25	6	66% dos demais casos de uso incorporados na arquitetura.
	27	6	77% dos demais casos de uso incorporados na arquitetura.
	29	6	88% dos demais casos de uso incorporados na arquitetura.
	31	6	100% dos demais casos de uso incorporados na arquitetura.
Transição	34	5	Sistema instalado. Migração de dados concluída.

## PLANEJAMENTO DE ITERAÇÃO

- Depende fortemente do processo escolhido



## OBJETIVOS POSSÍVEIS NO UP

- Implementar total ou parcialmente um ou mais *casos de uso*.
- Mitigar um *risco* conhecido, gerando ou executando um plano de redução de probabilidade, redução de impacto ou ainda de recuperação de desastre.
- Implementar total ou parcialmente uma ou mais *modificações* solicitadas.



## PRIORIZAR

- Casos de uso que representam os processos de negócio mais críticos para a organização, por exemplo, aqueles através dos quais a organização realiza seus objetivos, como por exemplo, obtenção de lucros.
- Riscos de alta importância, ou seja, com alto impacto e alta probabilidade de ocorrer.
- Modificações urgentes, como refatorações da arquitetura.



## WBS - ESTRUTURA ANALÍTICA DA ITERAÇÃO

- A estrutura analítica da iteração (WBS ou *Work Breakdown Structure*) é uma estrutura que apresenta as atividades que 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.

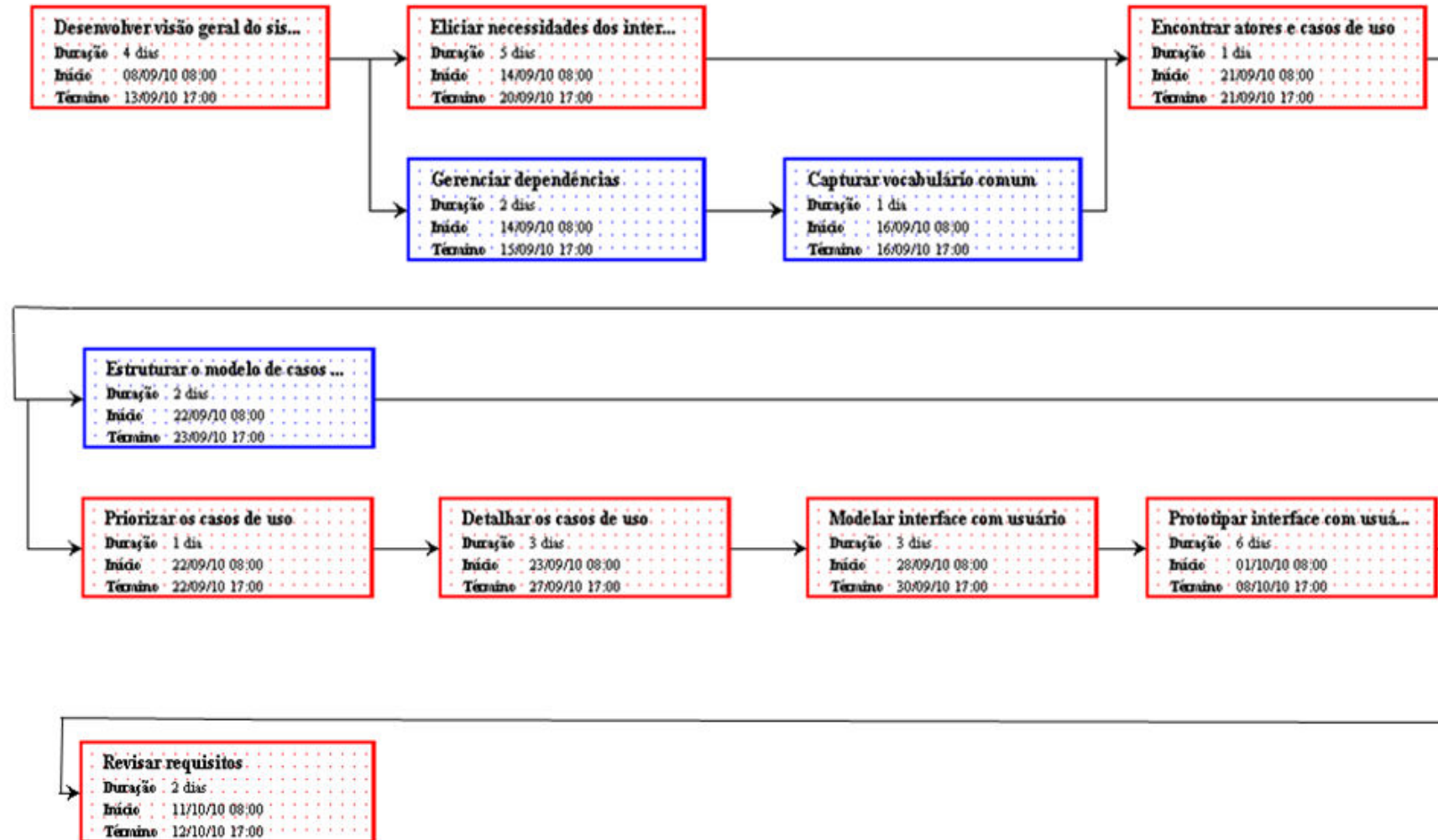


## EXEMPLO DE WBS

	Ⓜ	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

