

# PLANEJAR O TEMPO

Dr. rer. nat. Christiane Gresse von Wangenheim, PMP



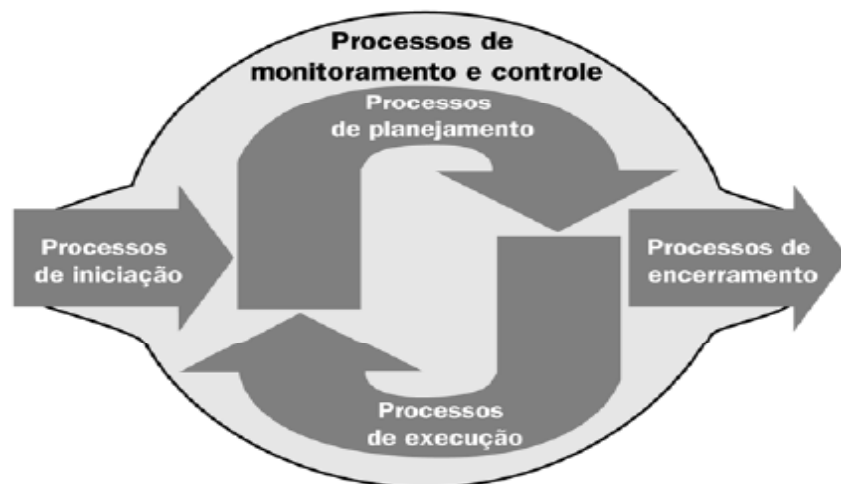
## Objetivo de aprendizagem desta aula

Ao final desta aula, você deverá ser capaz de:

- ☐ Descrever conceitos básicos e o processo de planejamento de tempo.
- ☐ Planejar o tempo de um projeto simples.



## Grupos de processos de gerenciamento de projetos



## Processos e áreas de conhecimento

	Iniciação	Planejamento	Execução	Monitoramento e Controle	Encerramento
<b>Integração</b>	Desenvolver o termo de abertura do projeto	Desenvolver o plano de gerenciamento do projeto	Orientar e gerenciar a execução do projeto	Monitorar e controlar o trabalho do projeto Realizar o controle integrado de mudança	Encerrar o projeto ou a fase
<b>Escopo</b>		Coletar os requisitos Definir o escopo Criar a EAP		Verificar o escopo Controlar o escopo	
<b>Tempo</b>		Definir as atividades Sequenciar as atividades Estimar os recursos das atividades Estimar a duração das atividades Desenvolver o cronograma		Controlar o cronograma	
<b>Custos</b>		Estimar os custos Determinar o orçamento		Controlar os custos	
<b>Qualidade</b>		Planejar a qualidade	Realizar a garantia da qualidade	Realizar o controle da qualidade	
<b>RH</b>		Desenvolver o plano de RH	Mobilizar a equipe do projeto Desenvolver a equipe do projeto Gerenciar a equipe do projeto	Gerenciar a equipe do projeto	
<b>Comunicações</b>	Identificar as partes interessadas	Planejar as Comunicações	Distribuir informações Gerenciar as expectativas das partes interessadas	Reportar o desempenho	
<b>Riscos</b>		Planejar gerenciamento dos riscos Identificar os riscos Realizar a análise qualitativa dos riscos Realizar a análise quantitativa dos riscos Planejar as respostas aos riscos		Monitorar e controlar os riscos	
<b>Aquisições</b>		Planejar as aquisições	Realizar as aquisições	Administrar as aquisições	Encerrar as aquisições

[PMBOK, 2009]

## Planejamento do tempo

### Definir as atividades

Seqüenciar as atividades

Estimar os recursos das atividades

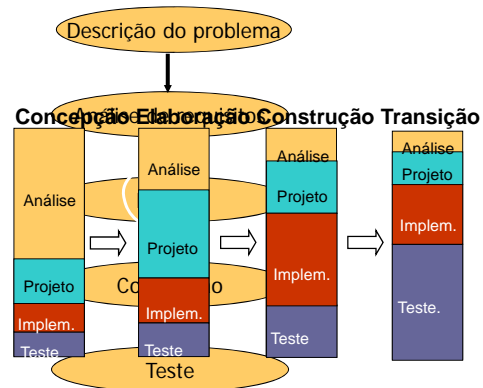
Estimar a duração das atividades

Desenvolver o cronograma

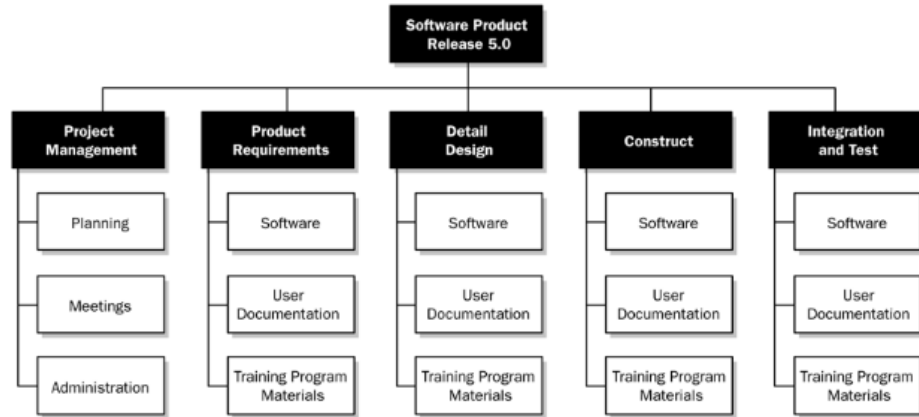
## Definir ciclo de vida

- ❑ **Ciclo de vida do projeto:** determina as fases ( e a sua seqüência) de acordo com o escopo de requisitos, as estimativas para os recursos do projeto e a natureza do projeto.

- ❑ Modelo cascata
- ❑ Modelo de prototipação
- ❑ Modelo iterativo/incremental
- ❑ Modelo espiral ...



## EAP baseado nas fases principais



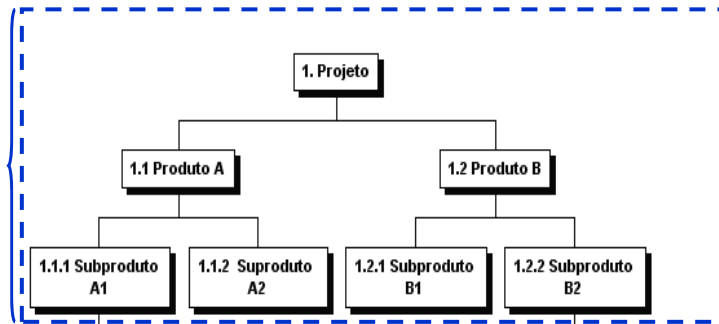
The WBS is illustrative only. It is not intended to represent the full project scope of any specific project, nor to imply that this is the only way to organize a WBS on this type of project.

## Definir as atividades

- ❑ Identificação das ações específicas a serem realizadas para produzir as entregas do projeto.
- ❑ Tipicamente os pacotes de trabalho são decompostos em componentes menores ⇒ **atividades**
- ❑ Atividades representam o trabalho necessário para completar o pacote de trabalho.
- ❑ As atividades e principalmente a sua seqüência depende do modelo de ciclo de vida.

## Definir as atividades

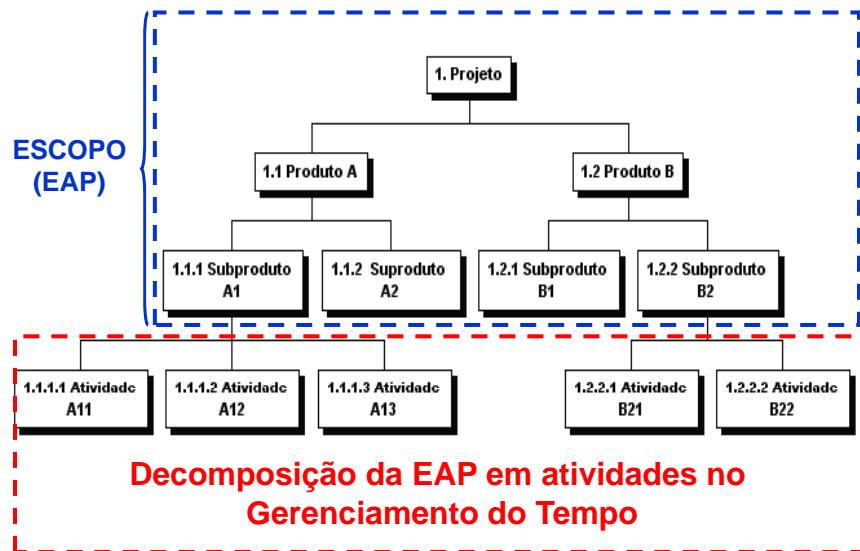
ESCOPO  
(EAP)



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Definir as atividades

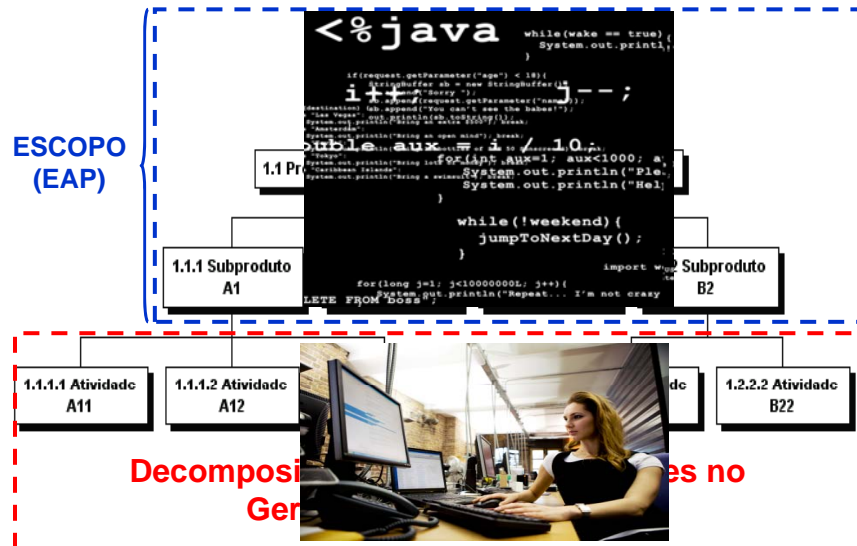
ESCOPO  
(EAP)



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Definir as atividades

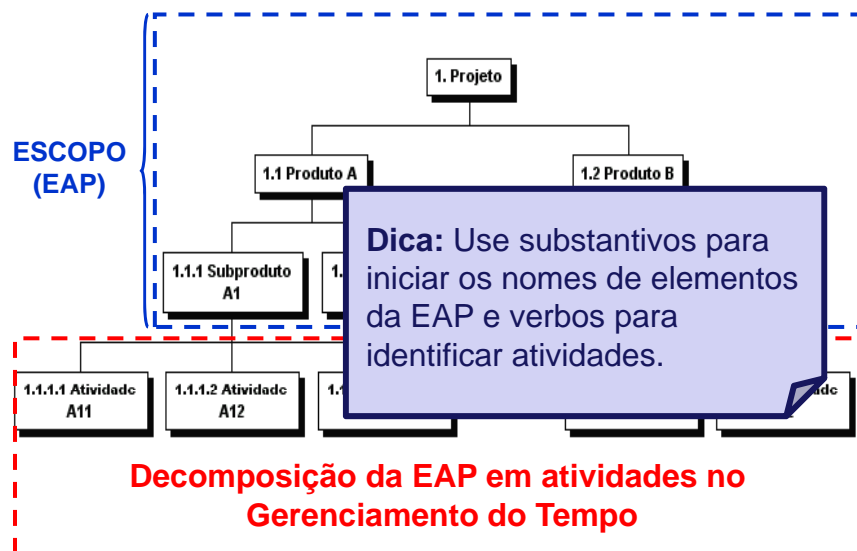
ESCOPO  
(EAP)



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Definir as atividades

ESCOPO  
(EAP)

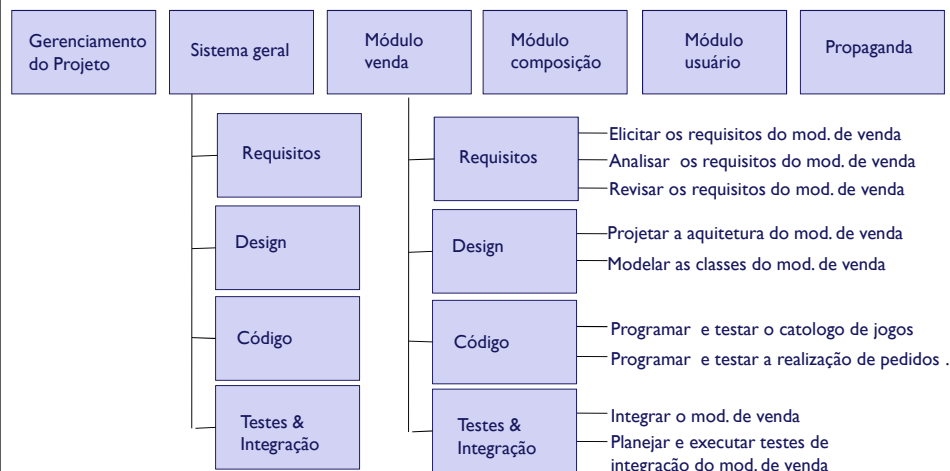


UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Resultados da definição das atividades

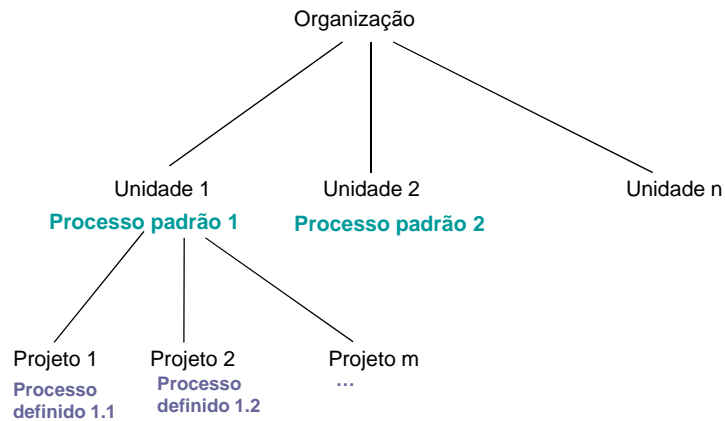
- ❑ **Lista das atividades:** inclui todas as atividades necessárias no projeto.
  - ❑ Identificador e descrição de cada atividade em detalhe suficiente para os membros da equipe entendem que trabalho precisa ser feito.
- ❑ **Lista dos marcos:** identifica todos os marcos do projeto.

## Exemplo simplificado: atividades



## Processo padrão

- ❑ Processo padrão serve como base para definir as atividades.



## Planejamento do tempo

Definir as atividades

Seqüenciar as atividades

Estimar os recursos das atividades

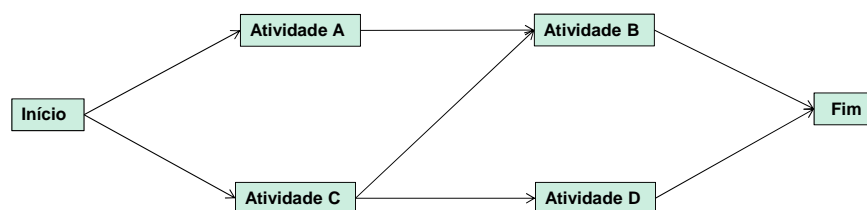
Estimar a duração das atividades

Desenvolver o cronograma



## Seqüenciar as atividades

- ❑ Identificação e documentação dos relacionamentos entre atividades do projeto identificando predecessores e sucessores.
  - ❑ **Método do diagrama de precedência (MDP):** construção de rede do cronograma do projeto mostrando as atividades nos nós.
  - ❑ **Diagramas de rede:** demonstração esquemática dos relacionamentos lógicos entre as atividades do cronograma do projeto.

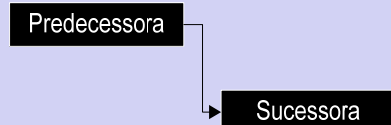


## Exemplo simplificado

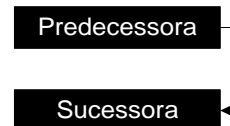
ID	Lista de atividades	Atividade(s) precedente(s)
3.1	Elicitar os requisitos do mod. de venda	---
3.2	Analisar os requisitos do mod. de venda	3.1
3.3	Revisar os requisitos do mod. de venda	3.2
3.4	Projetar arquitetura do mod. de venda	3.3
3.5	Modelar as classes do mod. de venda	3.4
3.6	Programar e testar o catalogo de jogos	3.5
3.7	Programar e testar a realização de pedidos	3.5
3.8	Integrar o mod. de venda	3.6, 3.7
3.9	Planejar e executar testes de integração do mod. de venda	3.8

## Tipos de dependências ou relações lógicas

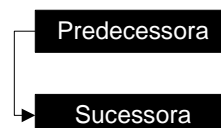
- ❑ **Término para início (TI):** início da atividade sucessora depende do término da atividade predecessora. (o mais comum)



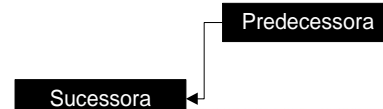
- ❑ **Término para término (TT):** término da atividade sucessora depende do término da atividade predecessora.



- ❑ **Início para início (II):** início da atividade sucessora depende do início da atividade predecessora.



- ❑ **Início para término (IT):** término da atividade sucessora depende do início da atividade predecessora.



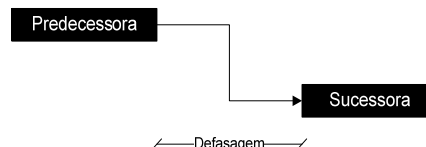
## Aplicando antecipação

- ❑ Mudar as dependências/ relações lógicas entre as atividades

- ❑ **Antecipação (Lead):** Modificação de um relacionamento lógico que permite uma antecipação da atividade sucessora.



- ❑ **Espera (Lag):** Modificação de um relacionamento lógico que gera um atraso na atividade sucessora.



## Planejamento do tempo

Definir as atividades

Seqüenciar as atividades

Estimar os recursos das atividades

Estimar a duração das atividades

Desenvolver o cronograma

## Estimar os recursos das atividades

Estimar **tipos** e **quantidades** de pessoas, material, equipamentos ou outros recursos que serão necessários para realizar cada atividade.



## Lista de recursos

Item	Quantidade
<b>Pessoal</b>	
Coordenador	1 (10 horas/mês por 12 meses)
Pesquisador sênior	2 (40 horas/mês por 12 meses)
<b>Equipamentos e Material permanente</b>	
Notebook	2
Servidor	4
Projeto multimídia	1
Serviços de terceiros (Design)	1
Papel (Caixa com 10 Resmas)	2
Toner	1
<b>Diárias</b>	
Diárias para evento nos EUA	5
<b>Passagens</b>	
Passagens para EUA	2

Que recursos tipicamente precisamos para projetos de software?

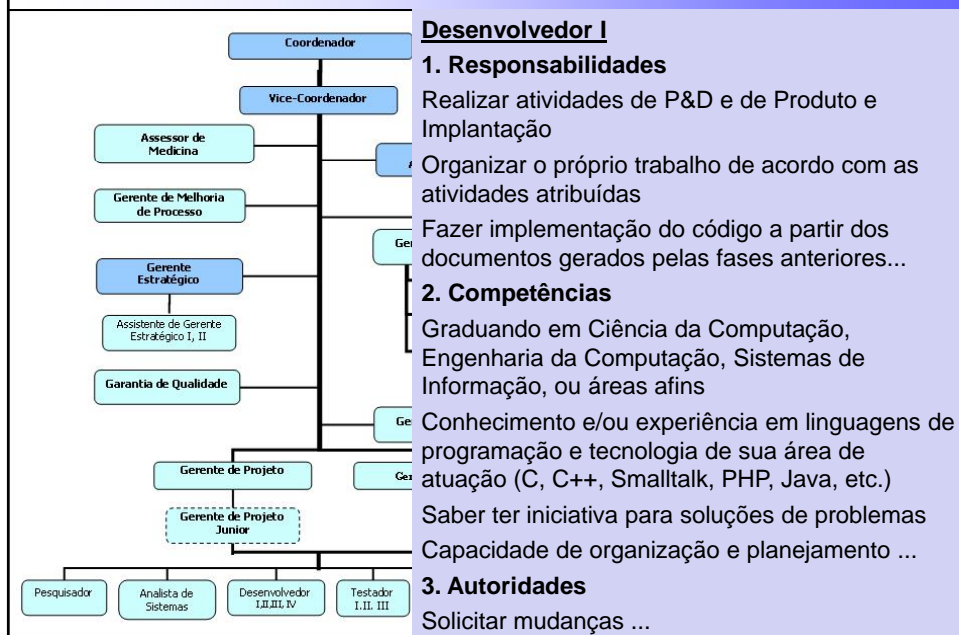
## Estimar os recursos das atividades

- ☐ Como fazer?
  - ☐ Opinião especializada
  - ☐ Análise de alternativas
  - ☐ Dados publicados para auxílio a estimativas
  - ☐ Estimativa bottom-up
- ☐ Entradas:
  - ☐ Organograma
  - ☐ Estrutura analítica de recursos

## Organograma

**Organograma:** representa a estrutura formal de uma organização mostrando tipicamente hierarquia de papéis, responsabilidades, competências e autoridades.

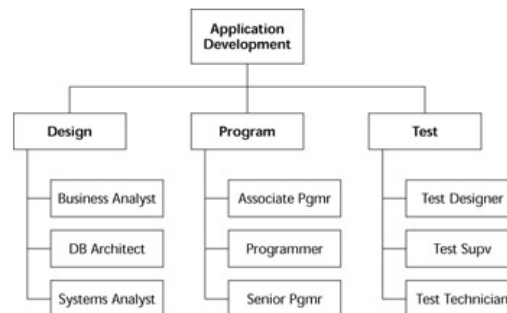
## Organograma do CYCLOPS/UFSC



## Estrutura analítica dos recursos

❑ **Estrutura analítica dos recursos:** estrutura hierárquica dos recursos identificados organizada por categoria e tipo de recursos.

- ❑ Categorias: mão de obra, material, equipamento, ...
- ❑ Tipos: nível de habilidade, etc.



## Exemplo simplificado

ID	Lista de atividades	Atividade(s) precedente(s)	Recursos humanos
3.1	Elicitar os requisitos do mod. de venda	---	Analista sênior, Diretor
3.2	Analisar os requisitos do mod. de venda	3.1	Analista sênior e Analista junior
3.3	Revisar os requisitos do mod. de venda	3.2	Analista sênior, Diretor
3.4	Projetar arquitetura do mod. de venda	3.3	Projetista
3.5	Modelar as classes do mod. de venda	3.4	Projetista, Programador sênior
3.6	Programar e testar o catalogo de jogos	3.5	2 Programadores junior
3.7	Programar e testar a realização de pedidos	3.5	Programador sênior, Programador junior
3.8	Integrar o mod. de venda	3.6, 3.7	Programador sênior
3.9	Planejar e executar testes de integração do mod. de venda	3.8	Testador

## Planejamento do tempo

Definir as atividades  
Seqüenciar as atividades  
Estimar os recursos das atividades  
Estimar a duração das atividades  
Desenvolver o cronograma

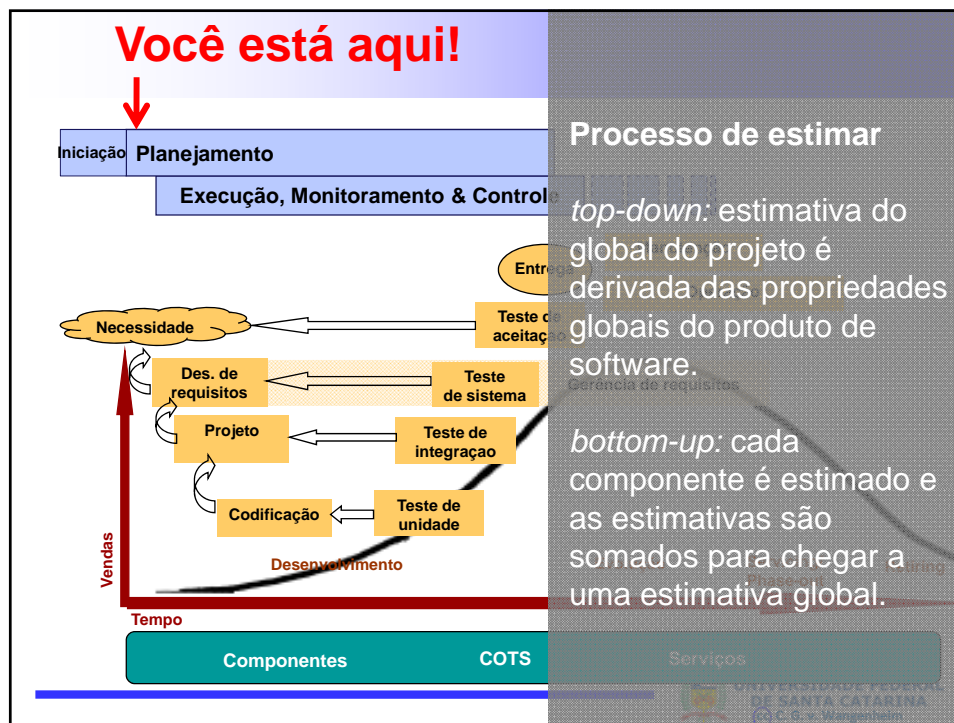
**Estimativa:** Um julgamento aproximado ou cálculo, a partir do valor, quantidade, tempo, tamanho ou peso de algo.

## O que estimar?

- I. **Tamanho/complexidade** do sistema, serviços ou resultado a ser desenvolvido (LOC, PF, PCU, ...)
- II. **Esforço da atividade**: quantidade de trabalho necessário para realizar uma atividade (pessoas-hora, pessoa-mês, ...)
- III. **Duração da atividade**: Quantidade de tempo que decorre entre o início e o término da atividade (horas, meses, ...)

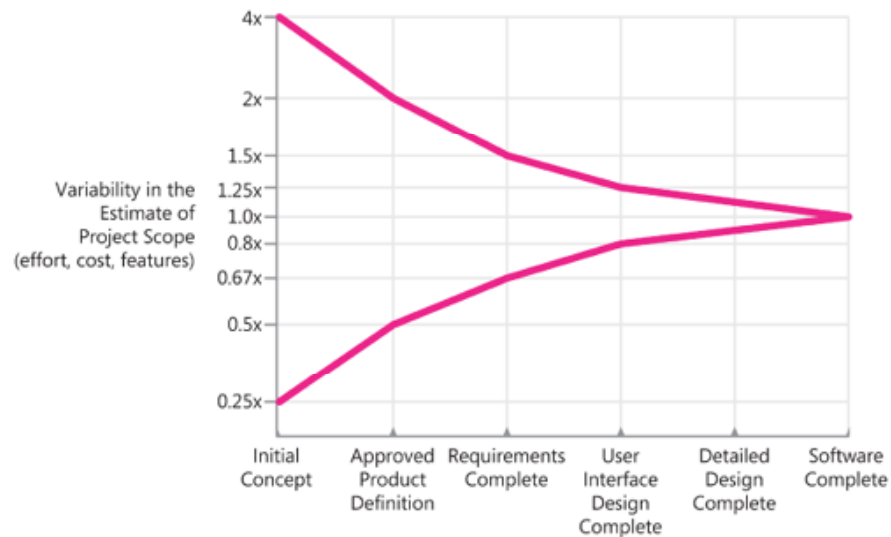
### Esforço X Duração

- Esforço estimado: 16 pessoa-horas
- 2 funcionários alocados 4h/dia -> Duração: 2 dias





## Cone da incerteza



[S. McConnell. Software Estimation: Demystifying the Black Art. Microsoft Press, 2006]



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Técnicas de estimativa

- ❑ **Analogia:** baseado em dados históricos da organização comparando o projeto com projetos semelhantes.
- ❑ **Baseado em modelos algorítmicos/ paramétricos:** baseado em funções matemáticas de atributos de produto, projeto e processo .
  - ❑ Modelos genéricos (p.ex. COCOMO), PF, PCU
  - ❑ Modelos específicos com base em dados históricos da própria organização
- ❑ **Opinião Especializada:** especialistas utilizam sua experiência e intuição para estimar.
  - ❑ *Planning Poker*
  - ❑ *Wideband delphi*



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

### Estimar o **tamanho** em o que?

- Linhas de Código (LOC)  
... mas dependem de vários fatores e não podem ser medidos no começo do projeto.
- Entidades atômicas:
  - Número de páginas
  - Número de objetos, métodos, etc.
- Pontos de função
- Pontos de caso de uso
- ...

### Pontos de função

- ❑ Idéia: software é melhor mensurado em termos de número e complexidade das funções que são executadas.
- ❑ **Pontos de função (PF):** medida de dimensionamento de software através da funcionalidade implementada em um sistema sob o ponto de vista do usuário.

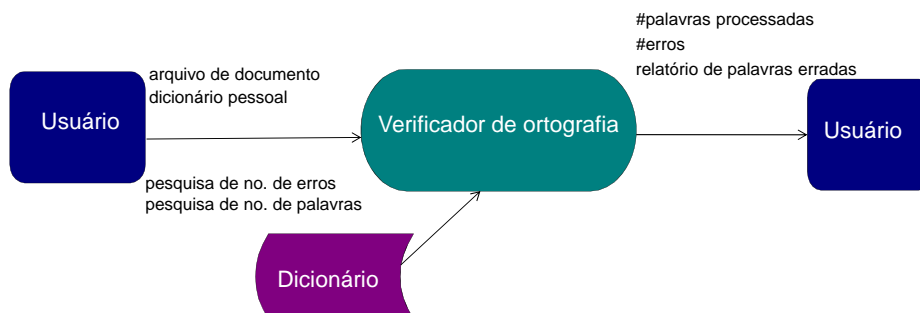
## Pontos de função (*Function Points*)

### ❑ Como fazer?

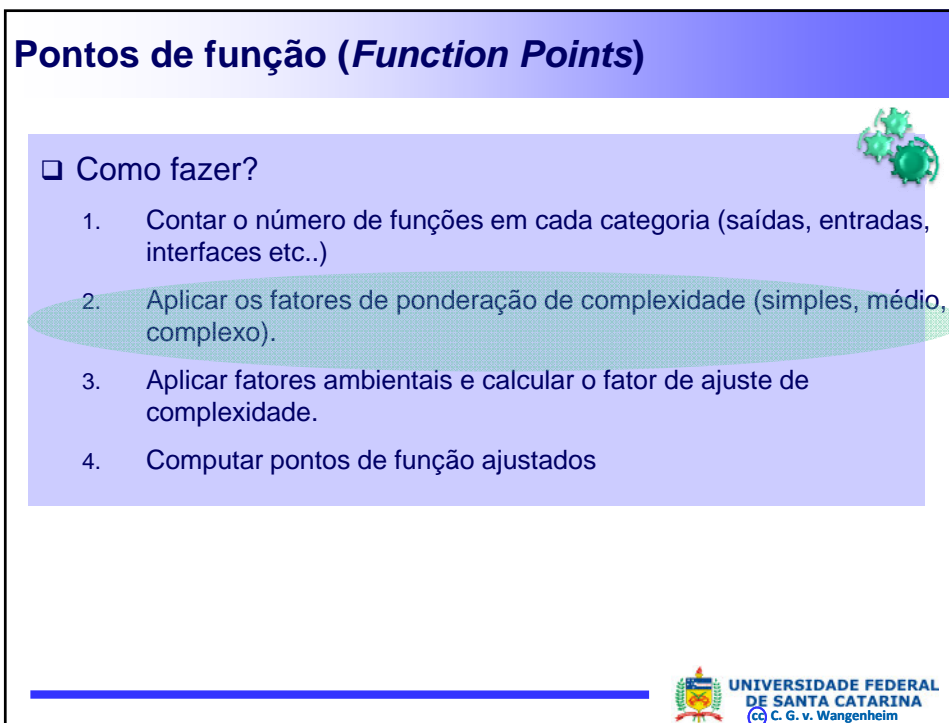
1. Contar o número de funções em cada categoria (saídas, entradas, interfaces etc..)
2. Aplicar os fatores de ponderação de complexidade (simples, médio, complexo).
3. Aplicar fatores ambientais e calcular o fator de ajuste de complexidade.
4. Computar pontos de função ajustados

## Exemplo: Pontos de função

- ❑ **Verificação de Ortografia:** O verificador aceita como entrada um arquivo de documento e um arquivo de dicionário pessoal opcional. O usuário pode examinar o número de palavras processadas e o número de erros de ortografia achados em qualquer fase durante o processamento.



[N. E. Fenton & S. L. Pfleeger. Software Metrics: A Rigorous and Practical Approach. Course Technology, 1998]



## Exemplo: Pontos de função

### ❑ Contador de pontos por função não ajustado (UFC)

Item	Número	Fator de Ponderação			Total
		simples	média	complexo	
Entrada externa	2 x	3	4	6	= 8
Saída externa	3 x	4	5	7	= 15
Consulta externa	2 x	3	4	6	= 8
Arquivos externos	2 x	7	10	15	= 20
Arquivos internos	1 x	5	7	10	= 7
<b>UFC = <math>\Sigma(\text{número de itens } i) \times \text{fator de ponderação } i</math></b>					<b>= 58</b>

## Pontos de função (*Function Points*)

### ❑ Como fazer?

1. Contar o número de funções em cada categoria (saídas, entradas, interfaces etc..)
2. Aplicar os fatores de ponderação de complexidade (simples, médio, complexo).
3. Aplicar fatores ambientais e calcular o fator de ajuste de complexidade.
4. Computar pontos de função ajustados

## Cálculo de ponto por função ajustado

- ❑ **Fator de complexidade técnica (TCF)** inclui Fatores de ajuste da complexidade ( $F_i$ )
  - ❑ *backup* e recuperação confiável
  - ❑ comunicação de dados
  - ❑ funções distribuídas
  - ❑ desempenho
  - ❑ ambiente operacional
  - ❑ intensivamente utilizado
  - ❑ entrada de dados *on-line*
  - ❑ facilitação de operação
  - ❑ atualização *on-line*
  - ❑ interfaces complexas
  - ❑ processamento complexo
  - ❑ reutilização
  - ❑ facilitação de instalação
  - ❑ múltiplas instalações
  - ❑ facilitação de mudanças
- ❑ A influência de cada fator é pontuado numa escala de 0 - sem influência até 5 – Essencial
- ❑  $TCF = 0.65 + 0.01 \sum F_i$



## Exemplo: Pontos de função

- ❑ Fatores de ajuste da complexidade ( $F_i$ )
 

❑ <i>backup</i> e recuperação confiável	3
❑ comunicação de dados	3
❑ desempenho	5
❑ entrada de dados <i>on-line</i>	3
❑ facilitação de operação	3
❑ atualização <i>on-line</i>	3
❑ processamento complexo	5
❑ facilitação de mudanças	3
❑ outros	0

❑  $TCF = 0.65 + 0.01 (18 + 10) = 0.93$



## Pontos de função (*Function Points*)

### ❑ Como fazer?

1. Contar o número de funções em cada categoria (saídas, entradas, interfaces etc..)
2. Aplicar os fatores de ponderação de complexidade (simples, médio, complexo).
3. Aplicar fatores ambientais e calcular o fator de ajuste de complexidade.
4. Computar pontos de função ajustados



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Pontos de função

### ❑ Ponto de função (PF) =

contador de pontos por função não ajustado (UFC) x fator de complexidade técnica (TCF)

### ❑ Exemplo

$$\square PF = UFC \times TCF = 58 \times 0.93 = 53.94$$



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Estimar esforço com base em PF

Ambiente/Linguagem	Produtividade (horas/PF)		
	Baixa	Média	Alta
Visual Basic	8,8 h	6,8 h	5,7 h
Delphi	8,8 h	6,8 h	5,7 h
C	26,4 h	17,6 h	13,2 h
C ++	13,2 h	8,8 h	6,6 h
Access	8,8 h	6,8 h	5,7 h
ORACLE	13,2 h	8,8 h	6,6 h
SQL	8,3 h	5,9 h	4,3 h
UNIX Shell Scripts	8,8 h	6,8 h	5,7 h
WEB/Documentos			
VB–Script	12 h	12 h	12 h
ASP	12 h	12 h	12 h
HTML	8,3 h	5,9 h	4,3 h
Java	13,2 h	8,8 h	6,6 h
DT2	17 h	17 h	17 h
Lotus Notes	5,5	3,9	3,1

Fonte: [HAZAN]

## Mais informações sobre pontos de função



- ❑ **BFPUG - Brazilian Function Point Users Group**  
[www.bfpug.com.br](http://www.bfpug.com.br)
- ❑ **IFPUG – International Function Point Users Group**  
[www.ifpug.org](http://www.ifpug.org)



## Pontos de Caso de Uso (Use Case Points - UCP)

- ❑ Considera:
  - ❑ complexidade das ações requeridas por cada tipo de usuário (**atores**)
  - ❑ avalia as atividades que devem ser executadas em cada tarefa (**casos de uso**)
- ❑ Para projetos orientados a objeto baseado em casos de uso

### Como fazer?

1. Classificar os atores
2. Classificar os casos de uso
3. Definir os pontos de caso de uso não ajustados
4. Determinar os fatores de ajuste (técnico e de ambiente)
5. Calcular o total ajustado dos pontos de caso de uso

## Classificar os atores

Classificação dos atores	Peso	Quantidade
Ator simples: um sistema externo (p.ex. com uma API definida)	1	
Ator médio: um sistema externo onde a comunicação é feita por meio de um protocolo de comunicação (p.ex. TCP/IP) ou um sistema de hardware, onde a interface de comunicação é também, de alguma forma, padronizada	2	
Ator complexo: um ator humano	3	
<b>Total ponderado não ajustado dos atores (UAW) = <math>\Sigma</math></b>	<b>peso *</b>	<b>x</b>

## Pontos de Caso de Uso

Como fazer?

1. Classificar os atores
2. Classificar os casos de uso
3. Definir os pontos de caso de uso não ajustados
4. Determinar os fatores de ajuste (técnico e de ambiente)
5. Calcular o total ajustado dos pontos de caso de uso

## Classificar os casos de uso

Complexidade do caso de uso	Interface com usuário	Manipulação de entidades de DB	No. de passos do fluxo base	No. de classes	Peso	Quantidade
Simple	Simple	Uma única entidade	1 a 3	1 a 5	5	
Médio	Mais elaborado	2 ou mais entidades	4 a 7	6 a 10	10	
Complexo	Complexo	3 ou mais entidades	8 ou mais	11 ou mais	15	
Total ponderado não ajustado dos casos de uso (UUCW) = $\Sigma$					peso *	x

## Pontos de Caso de Uso

Como fazer?

1. Classificar os atores
2. Classificar os casos de uso
3. Definir os pontos de caso de uso não ajustados
4. Determinar os fatores de ajuste (técnico e de ambiente)
5. Calcular o total ajustado dos pontos de caso de uso



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Total não ajustado de pontos de caso de uso

**Total não ajustado de pontos de caso de uso (UUCP) =  
UAW + UUCW**



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Pontos de Caso de Uso

Como fazer?

1. Classificar os atores
2. Classificar os casos de uso
3. Definir os pontos de caso de uso não ajustados
4. Determinar os fatores de ajuste (técnico e de ambiente)
5. Calcular o total ajustado dos pontos de caso de uso

## Fatores de complexidade técnica (TCF)

Fator		Peso	Influência (0 –nenhuma a 5- forte)
T1	Sistema distribuído	2	
T2	Tempo de resposta (desempenho)	1	
T3	Eficiência para o usuário final	1	
T4	Complexidade do processamento interno	1	Quanto mais alto o peso, <u>maior</u> o esforço necessário para terminar um projeto.
T5	Código reutilizável em outras aplicações	1	
T6	Facilidade de instalação	0.5	
T7	Facilidade de uso	0.5	
T8	Portabilidade	2	
T9	Facilidade de alteração	1	
T10	Necessidade de concorrência	1	
T11	Aspectos especiais de segurança	1	
T12	Conexão com outros sistemas	1	
T13	Necessidade de treinamento especial para o usuário	1	
TCF = 0,6 + (0,01 * Σ (		Peso *	X))

## Fatores de ambiente (EF)

Fator	Peso	Influência (0 –nenhuma a 5- forte)
F1 A equipe é familiar com o processo formal de desenvolvimento que será utilizado	1.5	
F2 Experiência da equipe com o domínio do problema	0.5	Quanto mais alto o peso, menor o esforço necessário para terminar um projeto.
F3 Experiência da equipe com orientação a objetos	1	
F4 Capacitação do analista chefe	0.5	
F5 Motivação da equipe	1	
F6 Estabilidade dos requisitos	2	
F7 Estagiários ou funcionários em tempo parcial	-1	
F8 Domínio da tecnologia e configuração do ambiente	-1	
<b>EF = 1,4 + (-0,03 * Σ (</b>		<b>Peso * X))</b>



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Pontos de Caso de Uso

Como fazer?

1. Classificar os atores
2. Classificar os casos de uso
3. Definir os pontos de caso de uso não ajustados
4. Determinar os fatores de ajuste (técnico e de ambiente)
5. Calcular o total ajustado dos pontos de caso de uso



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

### Total ajustado de pontos de caso de uso

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{EF}$$

### Estimar esforço com base no UCP

- ☐ Inicialmente, a técnica UCP propõe uma média de 20 pessoas-hora/UCP.
- ☐ Entretanto, experiências de campo têm mostrado que o esforço pode variar entre 15 a 30 pessoas-hora/UCP.

## Mais informações sobre UCP



- ❑ Jacobson, I., G. Booch, and J. Rumbaugh. The Objectory Development Process. Addison-Wesley, 1998.
- ❑ R. K. Clemmons. Project Estimation With Use Case Points. CrossTalk, Feb 2006.  
([www.stsc.hill.af.mil/crosstalk/2006/02/0602Clemmons.html](http://www.stsc.hill.af.mil/crosstalk/2006/02/0602Clemmons.html))



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## COCOMO (COConstructive COst Model)

- ❑ Estimativa paramétrica
- ❑ Modelo que se baseia nos resultados de um grande número de projetos em toda a indústria.



Barry Boehm

- ❑ Esforço em meses =  $A * (\text{tamanho em KLOC})^B * I(x)$
- ❑ Duração em meses =  $A * (\text{esforço em meses})^B$
- ❑ Tamanho da equipe = esforço/duração



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Constantes

Esforço em meses =  $A * (\text{tamanho em KLOC})^B * I(x)$

Duração em meses =  $A * (\text{esforço em meses})^B$

	Níveis de predictibilidade			
	Básico		Intermediário	
Tipo de projeto	A	B	A	B
Orgânico	2.4	1.05	3.2	1.05
Semi-destacado	3.0	1.12	3.0	1.12
Embutido	3.6	1.20	2.8	1.20

- ❑ Orgânico: simples, relativamente pequeno, requisitos menos rígidos
- ❑ Semi-destacado: intermediário
- ❑ Embutido: grande, complexidade alta, requisitos rígidos

## Parâmetro do nível de predictibilidade

- ❑ Níveis de predictibilidade
  - ❑ Básico: outras variáveis não são consideradas
  - ❑ Intermediário: 15 variáveis independentes são consideradas além do tamanho do código
  - ❑ Detalhado: intermediário aplicado para fases diferentes do projeto
- ❑  $I(x) = \Pi$  (Pontuação das variáveis de muito baixo para muito elevado)



## COCOMO: Variáveis

- X1. Confiabilidade exigida  
 X2. Tamanho do banco de dados da aplicação  
 X3. Complexidade  
 X4. Restrições de programação  
 X5. Restrições de hardware  
 X6. Volatilidade  
 X7. Tempo disponível  
 X8. Capacidade de desenvolvimento  
 X9. Capacidade de teste  
 X10. Experiência em linguagem de programação  
 X11. Experiência em engenharia de software  
 X12. Experiência em ferramentas de software  
 X13. Uso de ferramentas de software  
 X14. Aplicação de métodos de engenharia de software  
 X15. Cronograma de atividades de desenvolvimento exigido

	Variável			
Classificação	X1	X2	...	X15
Muito baixo	0.75	1.24		1.23
Baixo	0.88	1.10		1.08
Médio	1.00	1.00		1.00
Elevado	1.15	0.91		1.10
Extremamente elevado	1.40	0.82		1.10



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## COCOMO II

- ☐ É um modelo de custo mais abrangente, para calcular o esforço, custo e tempo a partir do tamanho do programa.
- ☐ É evoluído do COCOMO.
- ☐ É também um modelo baseado em fases, mas o tamanho do projeto pode ser medido em LOC e PF, ou até mesmo com objetos, como telas, relatórios.
- ☐ Reconhece diferentes abordagens de desenvolvimento de software, tais como prototipagem, desenvolvimento através da composição de componentes e uso de programação de banco de dados e também reutilização.
- ☐ [http://sunset.usc.edu/csse/research/COCOMOII/cocomo\\_main.html](http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html)



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## E, estes modelos funcionam?

- ❑ Existem diversos modelos:
  - ❑ COCOMO (projetos semi-destacados):  $\text{Esforço} = 3.0 * (\text{tamanho em KLOC})^{1.12}$
  - ❑ Modelo Walston/Felix (IBM):  $\text{Esforço} = 5.2 * (\text{tamanho em KLOC})^{0.91}$
  - ❑ Modelo Bailey/Basili (NASA-SEL):  $\text{Esforço} = 5.2 + 0.73 (\text{tamanho em KLOC})^{1.61}$
- ❑ Pesquisa realizada mostra uma taxa média de erro de estimativa de 85 até 772%.

[C. F. Kemerer. Na empirical validation of software cost estimation models. Communications of the ACM, vol. 30, no. 5, May 1987]

- ❑ A variação provavelmente depende do grau em que os ambientes onde os modelos foram criados coincidem com o ambiente em que estão sendo utilizados.
- ❑ Organizações que pretendem utilizar modelos algorítmicos de estimativas precisam coletar os dados históricos dos seus projetos para calibrar-los às condições locais.



## Opinião de especialistas

- ❑ Consultando um ou mais especialistas



Copyright © 1994 United Feature Syndicate, Inc.  
Redistribution in whole or in part prohibited.

- ❑ **Consenso em grupo:** experiência de uso de várias pessoas para chegar a uma estimativa
  - ❑ *Planning Poker*
  - ❑ *Wideband Delphi*





**Planning Poker**

- Técnica ágil de estimativa.
- Várias iterações para chegar num consenso.
- Evita influência de uma estimativa em outra.

[M. Cohn. Agile Estimating and Planning. Prentice Hall PT, 2005]

## Planning poker

### Como fazer?

1. O desenvolvedor com maior conhecimento sobre uma determinada funcionalidade/história dá uma breve explicação da funcionalidade. Para a equipe é dada a oportunidade de fazer perguntas e discutir para clarificar premissas e riscos.
2. Cada indivíduo dispõe de uma carta virada para baixo que representa sua estimativa. Unidades utilizadas variam, pode ser dias de duração, dias ideais ou pontos de história.
3. Todos viram as cartas simultaneamente.
4. Pessoas com estimativas muito altas e baixas justificam as suas estimativas e a discussão continua.
5. O processo de estimação é repetido até um consenso é alcançado.

## História de usuários (*User stories*)

**História de usuários:** definição alto-nível de requisitos, contendo apenas informação suficiente para que os especialistas possam chegar a uma estimativa razoável do esforço para implementá-la.

**Como usuário, eu quero ser capaz de ter alguns, mas nem todos os itens no meu carrinho embrulhados para presente.**

## Planning poker

Como fazer?

1. O desenvolvedor com maior conhecimento sobre uma determinada funcionalidade/história dá uma breve explicação da funcionalidade. Para a equipe é dada a oportunidade de fazer perguntas e discutir para clarificar premissas e riscos.
2. Cada indivíduo dispõe de uma carta virada para baixo que representa sua estimativa. Unidades utilizadas variam, pode ser dias de duração, dias ideais ou pontos de história.
3. Todos viram as cartas simultaneamente.
4. Pessoas com estimativas muito altas e baixas justificam as suas estimativas e a discussão continua.
5. O processo de estimação é repetido até um consenso ser alcançado.

## Unidades de estimativa

### ❑ *Dias ideais*

- ❑ Ideal dias são dias de desenvolvimento ideal de esforço ininterruptos, concentrados na tarefa e planejamento produtivo.
- ❑ Exemplo: tempo ideal de um filme geralmente é 120 minutos. Considerando os intervalos de propaganda o tempo total é muito maior.

### ❑ *Pontos de história de usuários*

- ❑ Medidas dificilmente baseiam-se na combinação de tamanho e complexidade de trabalho.
- ❑ Estimativas **sem dimensões**, mas unidades **proporcionais**.
  - ❑ Uma história de usuário de 10 pontos é esperado de levar duas vezes mais tempo do que uma história de usuário de 5 pontos.

## Cartas do *planning poker*

- ❑ Se baseiam parcialmente na sequência de números fibonacci



Mas isso já está pronto ...



Não entendi o que é para fazer ...



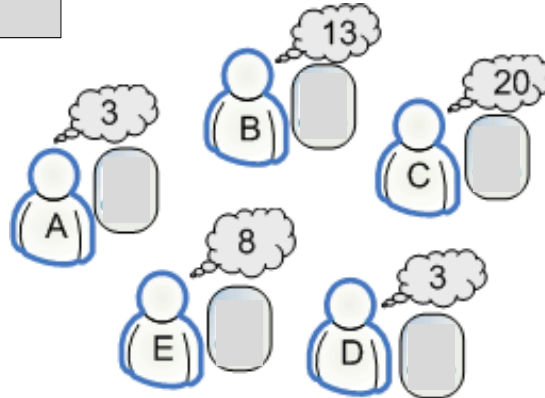
Não agüento mais !!! Vamos parar um pouco?!

## Planning poker

Ok gente, quantos dias ideais demorará para implementar esta história?



Como usuário, eu quero ser capaz de ter alguns, mas nem todos os itens no meu carrinho embrulhados para presente.



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Planning poker

Como fazer?

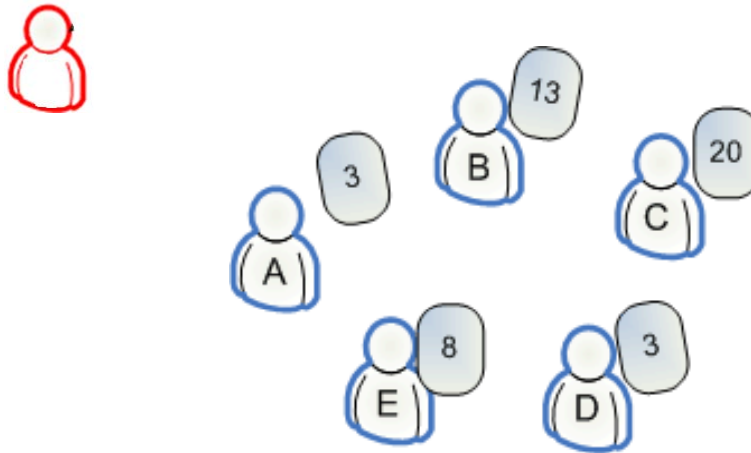


1. O desenvolvedor com maior conhecimento sobre uma determinada funcionalidade/história dá uma breve explicação da funcionalidade. Para a equipe é dada a oportunidade de fazer perguntas e discutir para clarificar premissas e riscos.
2. Cada indivíduo dispõe de uma carta virada para baixo que representa sua estimativa. Unidades utilizadas variam, pode ser dias de duração, dias ideais ou pontos de história.
3. Todos viram as cartas simultaneamente.
4. Pessoas com estimativas muito altas e baixas justificam as suas estimativas e a discussão continua.
5. O processo de estimação é repetido até um consenso é alcançado.



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Planning poker



## Planning poker

### Como fazer?

1. O desenvolvedor com maior conhecimento sobre uma determinada funcionalidade/história dá uma breve explicação da funcionalidade. Para a equipe é dada a oportunidade de fazer perguntas e discutir para clarificar premissas e riscos.
2. Cada indivíduo dispõe de uma carta virada para baixo que representa sua estimativa. Unidades utilizadas variam, pode ser dias de duração, dias ideais ou pontos de história.
3. Todos viram as cartas simultaneamente.
4. Pessoas com estimativas muito altas e baixas justificam as suas estimativas e a discussão continua.
5. O processo de estimação é repetido até um consenso é alcançado.



## Planning poker

Ah, mas você pensou em todas as possíveis alternativas? Tem ...

Acredito que seria bem fácil de implementar isto, porque é só a parte de ...

Talvez usar um cronometro (~3 min.) e não vira-lo mais de 2 vezes!

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
C. G. v. Wangenheim

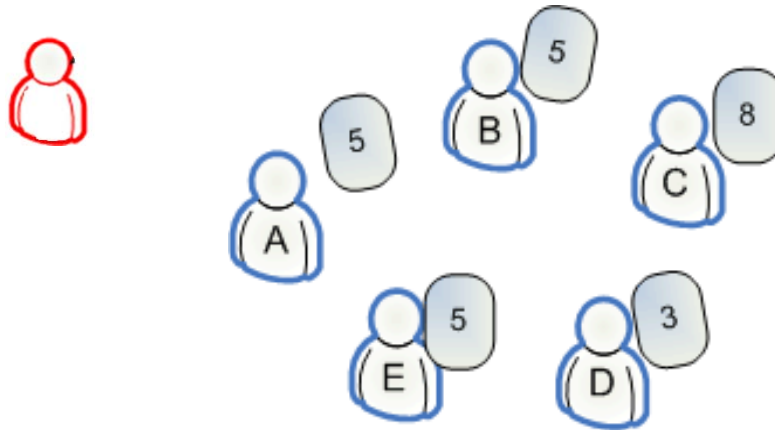
## Planning poker

### Como fazer?

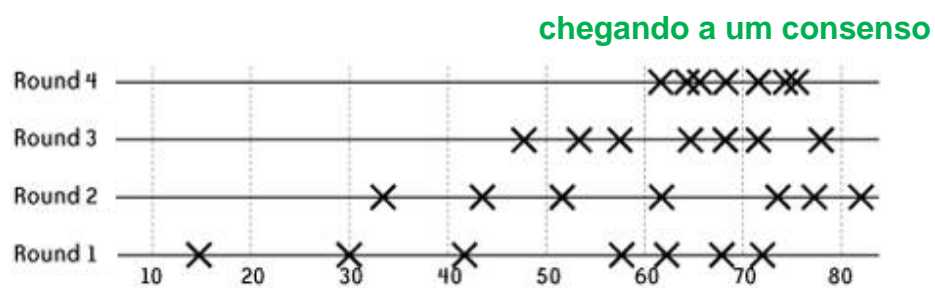
1. O desenvolvedor com maior conhecimento sobre uma determinada funcionalidade/história dá uma breve explicação da funcionalidade. Para a equipe é dada a oportunidade de fazer perguntas e discutir para clarificar premissas e riscos.
2. Cada indivíduo dispõe de uma carta virada para baixo que representa sua estimativa. Unidades utilizadas variam, pode ser dias de duração, dias ideais ou pontos de história.
3. Todos viram as cartas simultaneamente.
4. Pessoas com estimativas muito altas e baixas justificam as suas estimativas e a discussão continua.
5. O processo de estimação é repetido até um consenso é alcançado.



## Planning poker



## Estimativa por consenso

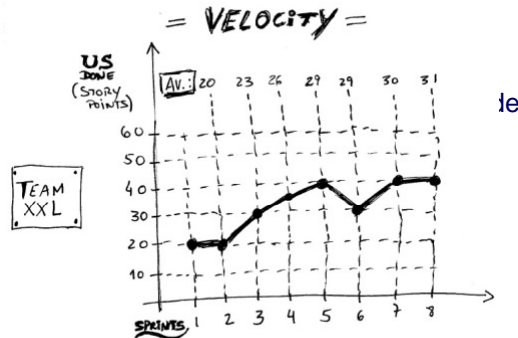


## Como estimar então o esforço?

- ❑ Esforço é estimado com base na velocidade.

### ❑ Velocidade

- ❑ Uma medida de quantos pontos de trabalho são concluídos por uma equipe em uma sprint.
- ❑ A mesma equipe pode alcançar diferentes velocidades dependendo do tipo de trabalho.
- ❑ O nosso melhor palpite é a velocidade média da equipe (considerando o "clima do trabalho" ou "weather").



## Exercício



- ❑ Exercício em pequenos grupos de 4 – 6 alunos.
- ❑ Instruções
  - ❑ Estimem o tamanho dos pacotes de trabalho a serem desenvolvidas usando a técnica de *planning poker*.

- ❑ Tempo: 15 min

## Wideband delphi

Como fazer?

1. Coordenador apresenta às especialistas o escopo e um formulário para estimativas. Estimadores esclarecem dúvidas com o coordenador.
2. Cada estimador faz individualmente as estimativas e entrega para o coordenador de forma anônima.
3. Coordenador analisa as estimativas individuais.
4. Estimadores recebem como *feedback* um resumo dos resultados.
5. Estimadores se reúnem para discutir diferenças.
6. Repetido até chegar a um consenso.

[A. Stellman, J. Greene. Applied Software Project Management. O'Reilly, 2005]



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Wideband delphi

Como fazer?

1. Coordenador apresenta às especialistas o escopo e um formulário para estimativas. Estimadores esclarecem dúvidas com o coordenador.
2. Cada estimador faz individualmente as estimativas e entrega para o coordenador de forma anônima.
3. Coordenador analisa as estimativas individuais.
4. Estimadores recebem como *feedback* um resumo dos resultados.
5. Estimadores se reúnem para discutir diferenças.
6. Repetido até chegar a um consenso.

[A. Stellman, J. Greene. Applied Software Project Management. O'Reilly, 2005]



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Wideband delphi

Como fazer?

1. Coordenador apresenta às especialistas o escopo e um formulário para estimativas. Estimadores esclarecem dúvidas com o coordenador.
2. Cada estimador faz individualmente as estimativas e entrega para o coordenador de forma anônima.
3. Coordenador analisa as estimativas individuais.
4. Estimadores recebem feedback e reavaliam.
5. Estimadores se reúnem para discutir discrepâncias.
6. Repetido até chegar a uma estimativa consensual.

[A. Stellman, J. Greene. Applied Software Engineering]

Goal statement To estimate the time to develop prototype for customers A & B									
Estimators Mike, Quentin, Jill, Sophie								Units days	
Shaded items must be discussed									
WBS# or priority	Task name	M.	Q.	J.	S.	Best-case	Worst-case	Avg. hi & lo	Notes
1	Interview customers (A+B)	6	9	3	3	3	6	3.5	
2	Develop requirements docs	5	10	2	5	2	10	5	Discrepancy between Q. and J.
3	Inspect requirements docs	7	5	6	5	5	7	5.5	
4	Do rework	8	7	9	7	7	9	7.5	
5	Prototype design	28	23	31	25	23	31	26.5	
6	Test design	9	7	6	6	6	9	6.5	
Total		63	56	57	51	46	72	54.5	

## Exemplo simplificado

ID	Lista de atividades	Atividade(s) precedente(s)	Estimativa de esforço (pessoas-horas)
3.1	Elicitar os requisitos do mod. de venda	---	40
3.2	Analisar os requisitos do mod. de venda	3.1	100
3.3	Revisar os requisitos do mod. de venda	3.2	40
3.4	Projetar arquitetura do mod. de venda	3.3	40
3.5	Modelar as classes do mod. de venda	3.4	80
3.6	Programar e testar o catalogo de jogos	3.5	200
3.7	Programar e testar a realização de pedidos	3.5	200
3.8	Integrar o mod. de venda	3.6, 3.7	20
3.9	Planejar e executar testes de integração do mod. de venda	3.8	40

## Nossas estimativas são tendenciosas?



Imagine que você está solicitando uma estimativa de um de seus funcionários, e em seguida pergunta: "Qual é a probabilidade de terminar 20% mais tarde do que sua estimativa?" Neste caso, você receberá algum tipo de número razoável, mostrando que o estimador é uma pessoa do mundo real levando em conta os fatos tristes da vida. Agora pergunta: "E, qual é a probabilidade de terminar 20% mais cedo do que a sua estimativa?" Muito provavelmente, seu estimador olhará para você de forma confusa, como se você tivesse

chegado recentemente de outro planeta.

A estimativa deve ser uma previsão, que tem a mesma probabilidade de estar acima ou abaixo do resultado real.

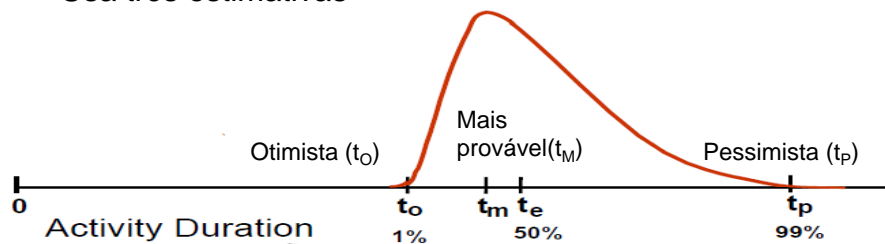
Tom Demarco



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Estimativas de três pontos

- Para melhorar a precisão das estimativas considerando as incertezas e riscos.
- Origem na Técnica de Revisão e Avaliação de Programa (PERT)
- Usa três estimativas



Cálculo da duração esperada ( $t_e$ ) usando a média ponderada

$$t_e = \frac{t_p + 4t_m + t_o}{6}$$



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim

## Exercício



- ☐ Trabalho em grupo
- ☐ Instruções
  - ☐ Calcule a duração esperada dado as seguintes estimativas:

Atividade	$t_O$	$t_M$	$t_P$
A	14	27	47
B	41	60	89

- ☐ Tempo: 5 min



## Estimar as durações das atividades

- ☐ Se a estimativa for esforço ou outra unidade, deve ser convertida em duração.
- ☐ Duração = Esforço / Recursos (às vezes)
- ☐ Levando em consideração:
  - ☐ Disponibilidade de recursos / calendários
  - ☐ Competência
  - ☐ Produtividade
- ☐ Análise de reservas: incluindo reservas para contingências (*buffers*) para considerar incertezas.
  - ☐ Pode ser uma porcentagem da duração estimada da atividade
  - ☐ Número fixo de períodos de trabalho
- ☐ Para mostrar incertezas podem incluir indicações da faixa de resultados:
  - ☐ 2 semanas  $\pm$  2 dias
  - ☐ 15% probabilidade de exceder 3 semanas

*"É difícil fazer previsões,  
especialmente sobre o  
futuro."*

Yogi Berra



## Planejamento do tempo

Definir as atividades  
Seqüenciar as atividades  
Estimar os recursos das atividades  
Estimar a duração das atividades  
Desenvolver o cronograma

## Desenvolver o cronograma

**Cronograma do projeto:** as datas planejadas para realizar as atividades do cronograma e para atingir os marcos do cronograma.

- ❑ Processo de análise de seqüências das atividades, suas durações, recursos necessários e restrições ao cronograma visando criar o cronograma do projeto.
  - ❑ Análise da rede do cronograma
  - ❑ Método do caminho critico
  - ❑ Nivelamento de recursos
  - ❑ Método de corrente critica
  - ❑ Análise do cenário “E – se”



## Análise da rede do cronograma

- ❑ **Caminho de ida:** Cálculo das datas de início mais cedo e de término mais cedo para as partes incompletas de todas as atividades da rede.
- ❑ Indica duração do projeto.

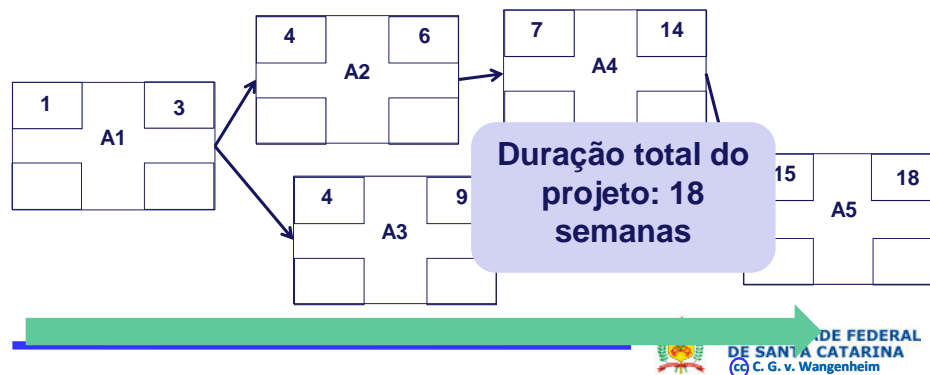




## Caminho de ida

Atividade	Predecessora	Estimativa de duração
A1	-	3 semanas
A2	A1	3 semanas
A3	A1	6 semanas
A4	A2	8 semanas
A5	A3, A4	4 semanas

Início mais cedo		Término mais cedo
<b>Atividade</b>		
Início mais tarde		Término mais tarde



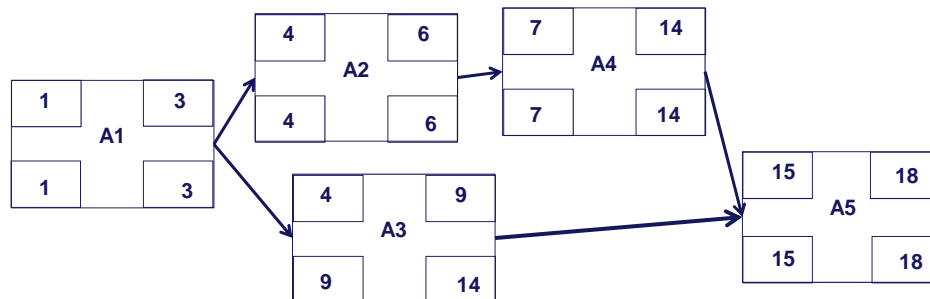
## Análise da rede do cronograma

- ❑ Caminho de volta: Cálculo das datas de término mais tarde e datas de início mais tarde para as partes incompletas de todas as atividades do cronograma.

## Caminho de volta

Atividade	Sucessora	Estimativa de duração
A1	-	3 semanas
A2	A1	3 semanas
A3	A1	6 semanas
A4	A2	8 semanas
A5	A3, A4	4 semanas

Início mais cedo		Término mais cedo
<b>Atividade</b>		
Início mais tarde		Término mais tarde



## Método do caminho crítico

### ❑ Folga

- ❑ Atraso total permitido para a data de início mais cedo de uma atividade do cronograma sem atrasar a data de término do projeto ou violar uma restrição do cronograma.
- ❑  $\text{Folga} = \text{Início mais tarde} - \text{Início mais cedo}$

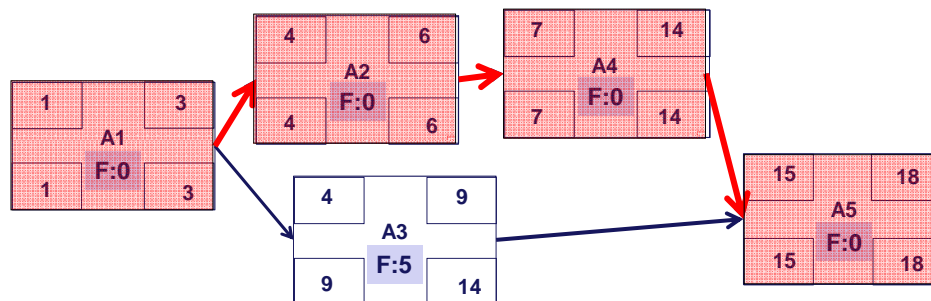
### ❑ Caminho crítico

- ❑ Caminho mais longo através da rede de diagrama do projeto, que indica o menor tempo para completar o projeto.
- ❑ Qualquer atraso nas atividades do caminho crítico atrasarão diretamente o projeto.

## Folgas e caminho crítico

Atividade	Sucessora	Estimativa de duração
A1	-	3 semanas
A2	A1	3 semanas
A3	A1	6 semanas
A4	A2	8 semanas
A5	A3, A4	4 semanas

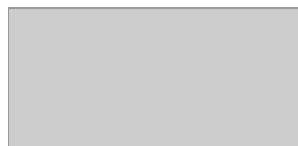
Início mais cedo		Término mais cedo
<b>Atividade</b>		
Início mais tarde		Término mais tarde



## Exercício



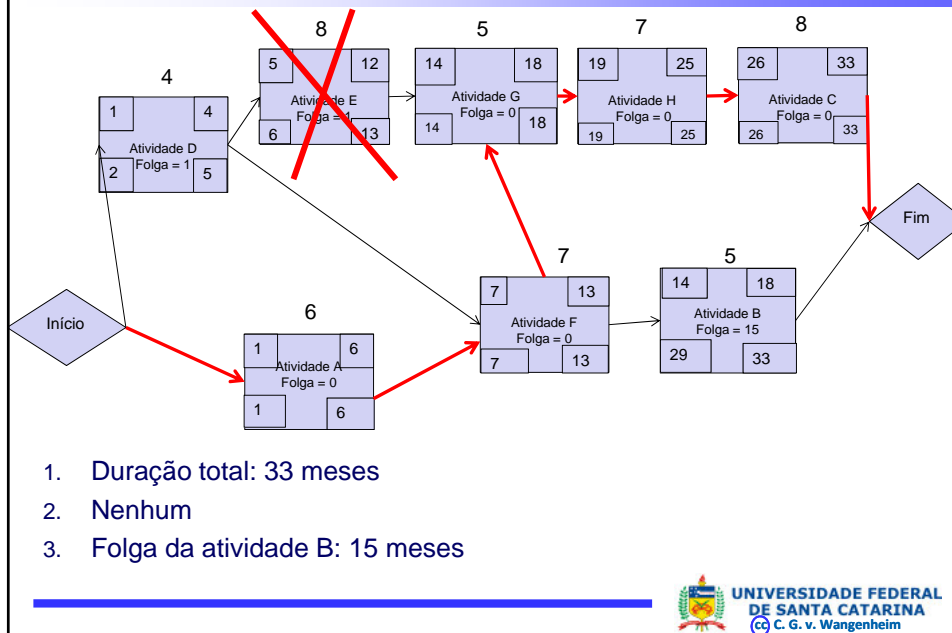
- ☐ Trabalho em grupo.
- ☐ Instruções: Responda as seguintes questões:
  - ☐ 1. Qual a duração do caminho crítico?
  - ☐ 2. Para encurtar a duração do projeto, o patrocinador propôs retirar o trabalho da atividade E do projeto, tornando a atividade D a antecessora de atividades G e F. Qual será o efeito dessa alteração na duração total?
  - ☐ 3. Qual é a folga da atividade B?



Tempo: 10 minutos

Activity	Preceding Activity	Estimate In Months
Start		0
D	Start	4
A	Start	6
F	D, A	7
E	D	8
G	F, E	5
B	F	5
H	G	7
C	H	8
End	C, B	0

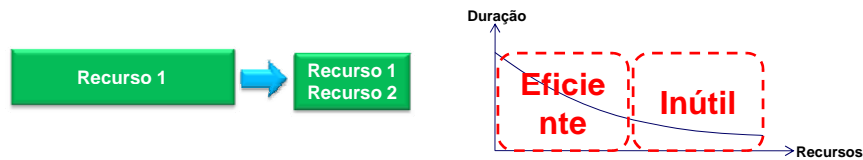
## Exercicio



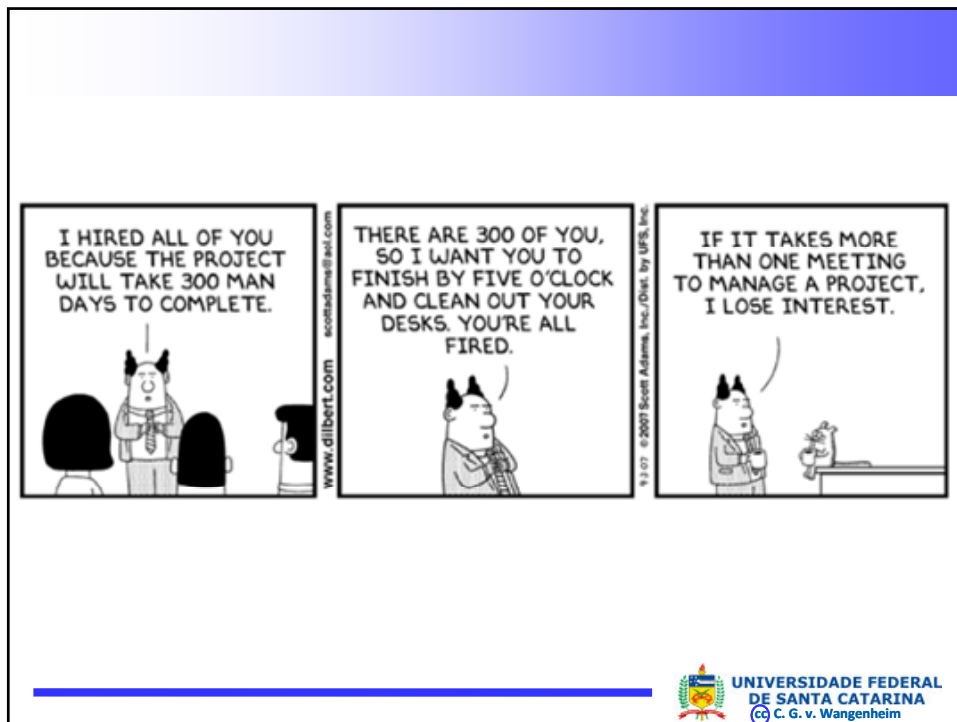
## Como reduzir a duração do projeto?

Objetivo: encurtar o cronograma sem mudar o escopo para respeitar restrições do cronograma.

- ☐ **Compressão (*crashing*):** analisar compensações entre custo e cronograma para determinar como obter a maior quantidade de compressão com o mínimo incremento de custo
  - ☐ Horas extras
  - ☐ Recursos adicionais
  - ☐ Pagamento para a aceleração da entrega de atividades no caminho crítico



- ☐ Só funciona para atividades onde recursos adicionais encurtarão a sua duração.
  - ☐ Nem sempre viável e pode resultar num maior risco e/ou custo.
- UNIVERSIDADE FEDERAL DE SANTA CATARINA  
C. G. v. Wangenheim



## Como reduzir a duração do projeto?

- ❑ **Paralelismo (*fast tracking*)**: fases ou atividades normalmente executadas em sequência são executadas em paralelo.



- ❑ Só funciona se as atividades puderem ser sobrepostas
- ❑ Pode resultar na repetição de trabalho e aumento de risco

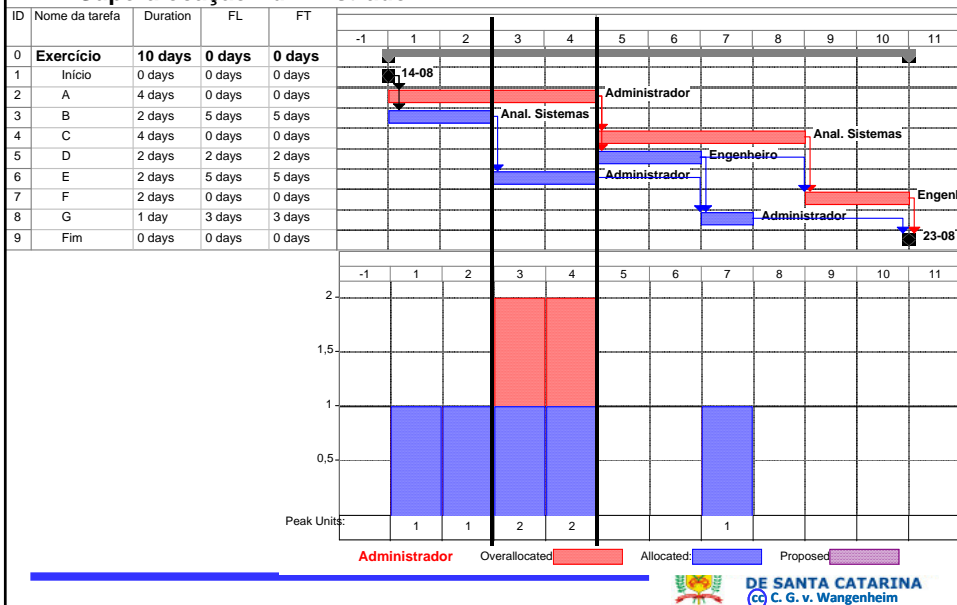
## Nivelamento dos recursos

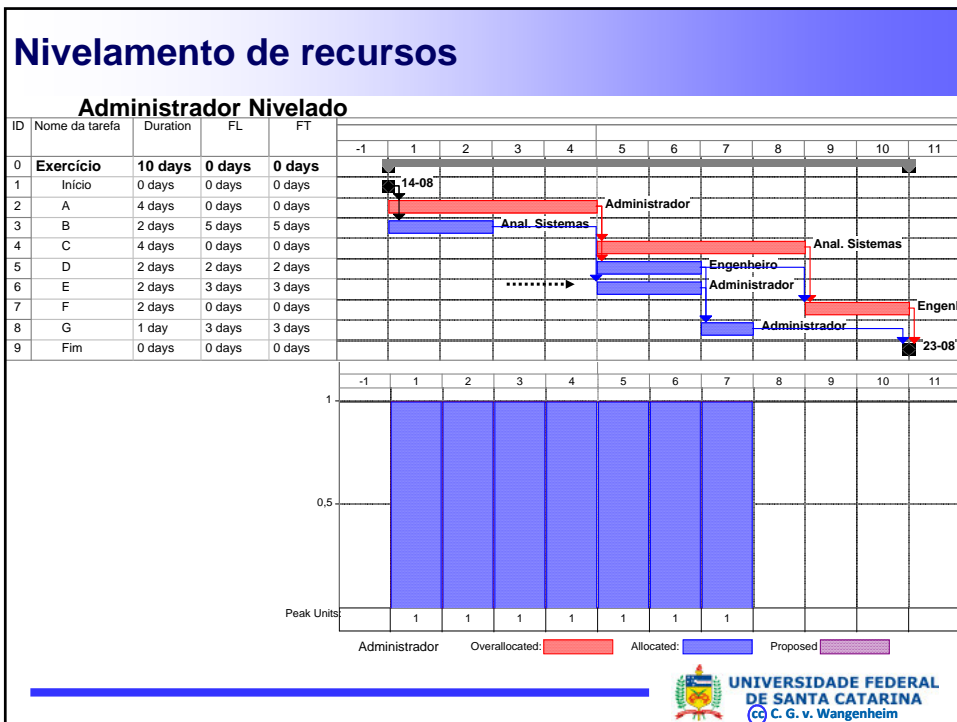
- ❑ Efetuar mudanças no cronograma em função da disponibilidade dos recursos com o objetivo de diminuir os custos do projeto.
- ❑ Usado para produzir um cronograma de recursos limitados.
- ❑ Buscar o equilíbrio no uso dos recursos atenuando “picos” e os “vales” de utilização, minimizando respectivamente a necessidade de recursos adicionais e a ociosidade de recursos alocados.
- ❑ Soluções mais comuns:
  - ❑ Atraso em atividades causadoras de super-alocação
  - ❑ Substituição de recursos
  - ❑ Hora-extra



## Nivelamento de recursos

### Superalocação Administrador



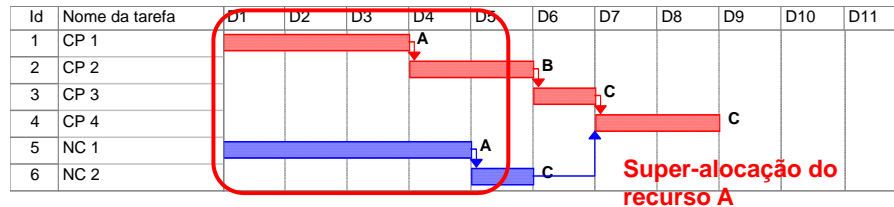


### Método da corrente crítica

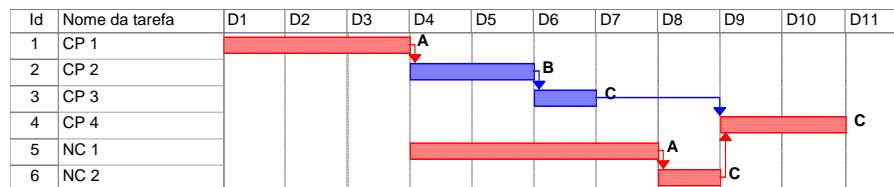
- ☐ Análise de rede do cronograma que modifica o cronograma do projeto para que se leve em conta a limitação de recursos.
- ☐ **Corrente crítica:** caminho crítico restrito por recursos.
- ☐ Adiciona *buffers* de duração, que são atividades sem trabalho do cronograma para gerenciar as incertezas.
- ☐ O foco de gestão passa da folga total para os *buffers* de duração e recursos limitados.
- ☐ Ajustar antecipações e esperas pode ajudar na viabilidade do cronograma do projeto.

## Caminho crítico vs. Corrente crítica

No gerenciamento tradicional de projetos o Caminho Crítico é definido antes do nivelamento de recursos, ou seja, a alocação dos recursos disponíveis.



A Corrente Crítica é definida depois do nivelamento de recursos, quando os tempos previstos reais estão definidos.



## Análise de cenários

❑ E se tal cenário ocorrer, o que ocorre com o projeto?

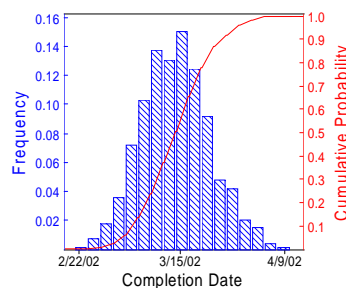
❑ Simulação de Monte Carlo:

❑ Calcula a duração do projeto usando valores diferentes de entrada, que são obtidos de uma distribuição de probabilidade.

❑ No final possíveis cenários são exibidos.

Date: 1/29/02 10:29:03 AM  
Samples: 1000  
Unique ID: 0  
Name: Module XYZ

Completion Std Deviation: 5.42 days  
95% Confidence Interval: 0.34 days  
Each bar represents 2 days



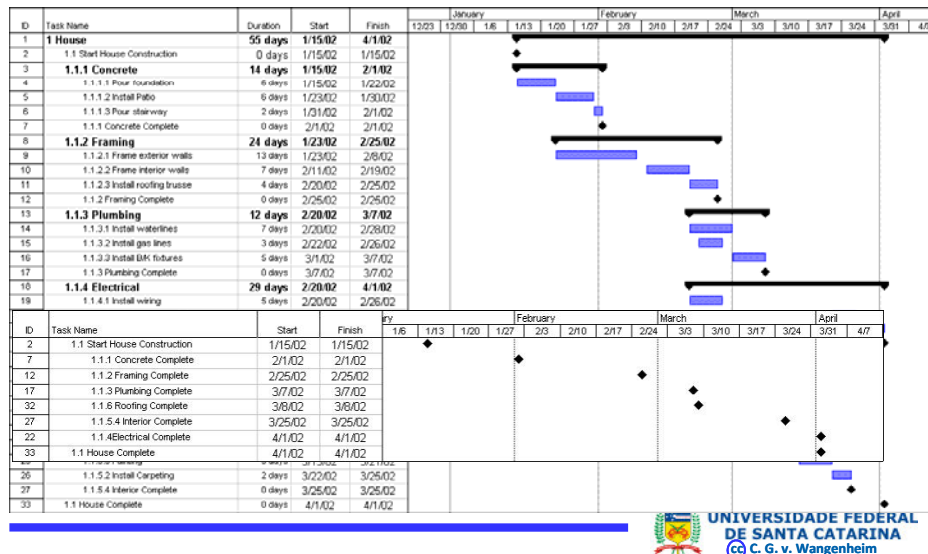
Completion Probability Table

Prob	Date	Prob	Date
0.05	3/5/02	0.55	3/18/02
0.10	3/6/02	0.60	3/19/02
0.15	3/8/02	0.65	3/19/02
0.20	3/11/02	0.70	3/20/02
0.25	3/12/02	0.75	3/21/02
0.30	3/12/02	0.80	3/22/02
0.35	3/13/02	0.85	3/25/02
0.40	3/14/02	0.90	3/27/02
0.45	3/15/02	0.95	3/29/02
0.50	3/15/02	1.00	4/9/02



## Resultados do desenvolvimento de cronograma

### ❑ Cronograma do projeto



## Resumo

- ❑ Desenvolvimento de um cronograma aceitável é geralmente um processo iterativo.
- ❑ Processo composto de:
  - ❑ Definir as atividades
  - ❑ Seqüenciar as atividades
  - ❑ Estimar os recursos das atividades
  - ❑ Estimar a duração das atividades
  - ❑ Desenvolver o cronograma
- ❑ Estimar é crucial para o sucesso dos projetos.
  - ❑ No geral, nós ainda não somos bons nisso.
  - ❑ Às vezes, nos estimamos e temos sorte :-)
  - ❑ Mas estimar alguma coisa é melhor do que não estimar nada!

## Leia mais sobre este tema ...



- ❑ PMI. Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK). 2000-2009. [\[UFSC BU Acervo 204532\]](#)
- ❑ H. Kerzner. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Wiley, 2003-2009. [\[UFSC BU Acervo 204553\]](#)
- ❑ A. Acuña, X. Ferre, M. López, L. Mate. The Software Process: Modeling, Evaluation and Improvement. Argentina: World Scientific Publishing Company, 2000.


## A1. Planejamento do tempo



- ❑ Trabalho em grupo
- ❑ Instruções:
  - ❑ Com base nos passos anteriores do trabalho A1, planeje o tempo para o seu projeto.
    - ❑ Definição das atividades e o seqüenciamento das atividades
    - ❑ Estimativa de recursos
    - ❑ Estimativa de tamanho de pacotes de trabalho da EAP
    - ❑ Estimativa de duração de atividades
    - ❑ Diagrama de rede do cronograma
    - ❑ Diagrama GANTT
  - ❑ Documente os resultados utilizando o *template* definido e integra ao A1. Plano de Projeto.
  - ❑ Termine esta atividade antes do próximo passo.



Instruções do trabalho no moodle



**Atribuição-Uso Não-Comercial-Compartilhamento pela Licença 2.5 Brasil**

**Você pode:**

- copiar, distribuir, exibir e executar a obra
- criar obras derivadas


**Sob as seguintes condições:**

**Atribuição** — Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.

**Uso Não-Comercial** — Você não pode utilizar esta obra com finalidades comerciais.

**Compartilhamento pela mesma Licença** — Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.

Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/br/> ou mande uma carta para Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.



UNIVERSIDADE FEDERAL  
DE SANTA CATARINA  
C. G. v. Wangenheim