

# Planejamento de Desenvolvimento de Software

Everson Santos Araujo  
[everson@por.com.br](mailto:everson@por.com.br)

# Introdução

- O planejamento para o desenvolvimento de um software é fator preponderante para que se tenham resultados satisfatórios de produtividade e qualidade

# Análise de riscos

- Esta atividade está baseada na realização de quatro tarefas, conduzidas de forma seqüencial:
  - Identificação
  - Projeção
  - Avaliação
  - Administração

# Identificação dos Riscos

- O objetivo é que sejam levantados todos os eventuais riscos
  - De projeto
  - Técnicos
  - De produto

# Riscos de projeto

- Estão associados a problemas relacionados ao próprio processo de desenvolvimento
  - Orçamento
  - Cronograma
  - Pessoal

# Riscos técnicos

- Consistem dos problemas de projeto efetivamente
  - Implementação
  - Manutenção
  - Interfaces
  - Ferramentas

# Riscos do produto

- Relacionados aos problemas que são consequência da inserção do software como produto no mercado
  - Produto sem interesse de mercado
  - Produto ultrapassado
  - Inadequado à venda

# Projeção dos riscos

- Probabilidade de ocorrência de determinado risco
- Conseqüências dos problemas associados ao risco no caso de ocorrência

# Projeção de riscos

- Criação de uma escala para estimar a probabilidade de ocorrência
- Estabelecimento das conseqüências
- Estimativa do impacto sobre o projeto

# Avaliação dos riscos

- Processa as informações sobre o fator de risco, o impacto e a probabilidade de ocorrência
- Checar as informações obtidas, definindo prioridades
- Estabelecer um limite de risco como referência para decisão

# Administração e Monitoramento

- Tomar medidas para evitar a ocorrência dos riscos
- Definir ações que serão tomadas na eventualidade de uma ocorrência
- Geração de **Plano de Administração e Monitoramento de Riscos**

# Geração do Plano de Administração e monitoramento de riscos



# Cronograma

- Geralmente baseada na definição prévia de um prazo de entrega do software
- O planejamento deve ser feito de modo a distribuir os esforços ao longo do prazo estabelecido

# As relações pessoas-trabalho

- O número de pessoas é proporcional ao tamanho do projeto
- Quanto mais pessoas, maior o número de canais de comunicação
- É melhor realizar o desenvolvimento de um software com uma equipe com menos pessoas por um período maior de tempo do que o contrário

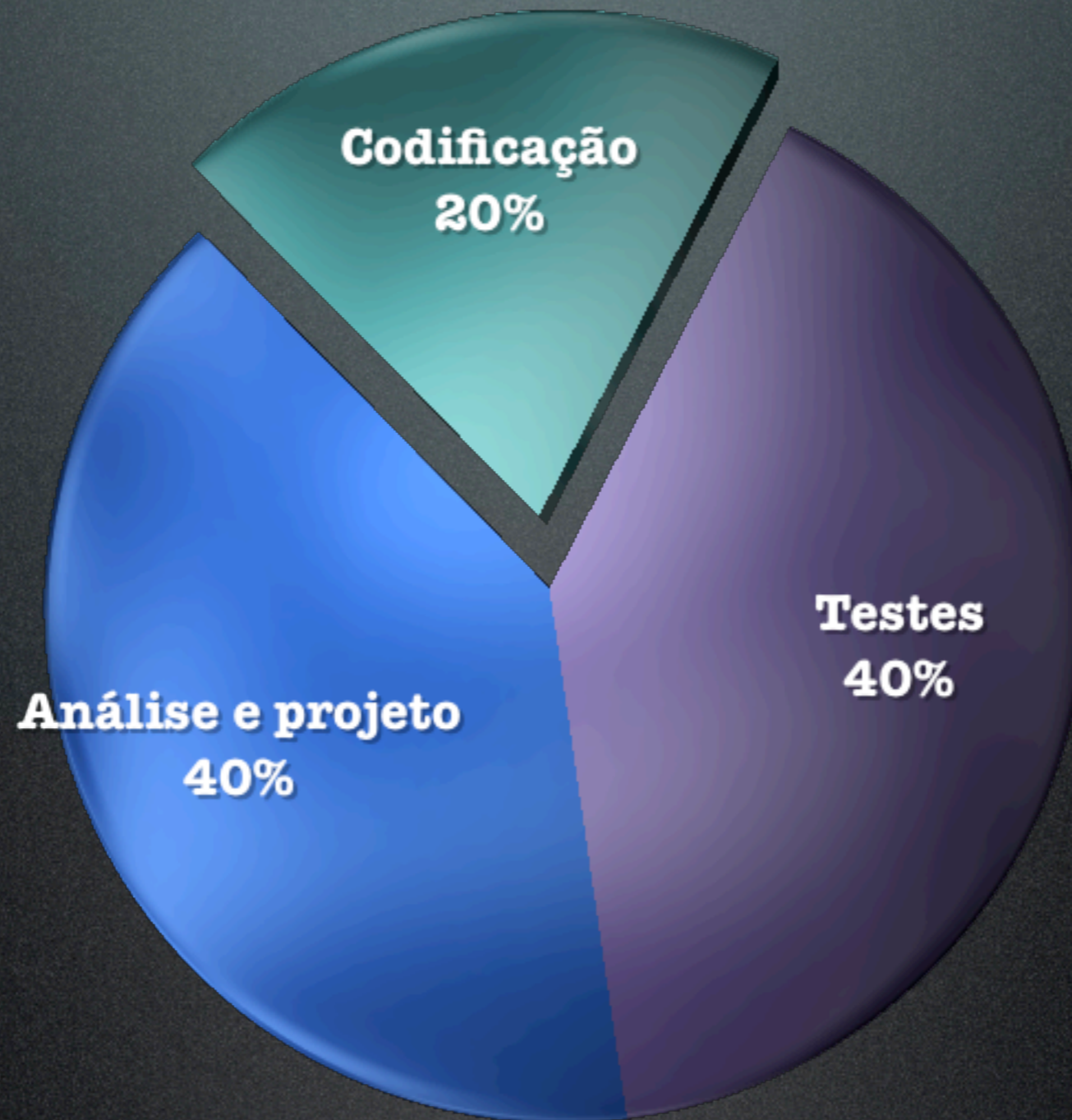
# Definição de tarefas

- Com projetos que possuam uma quantidade razoável de recursos humanos, é possível trabalhar em paralelo através da definição de tarefas e suas dependências de recursos

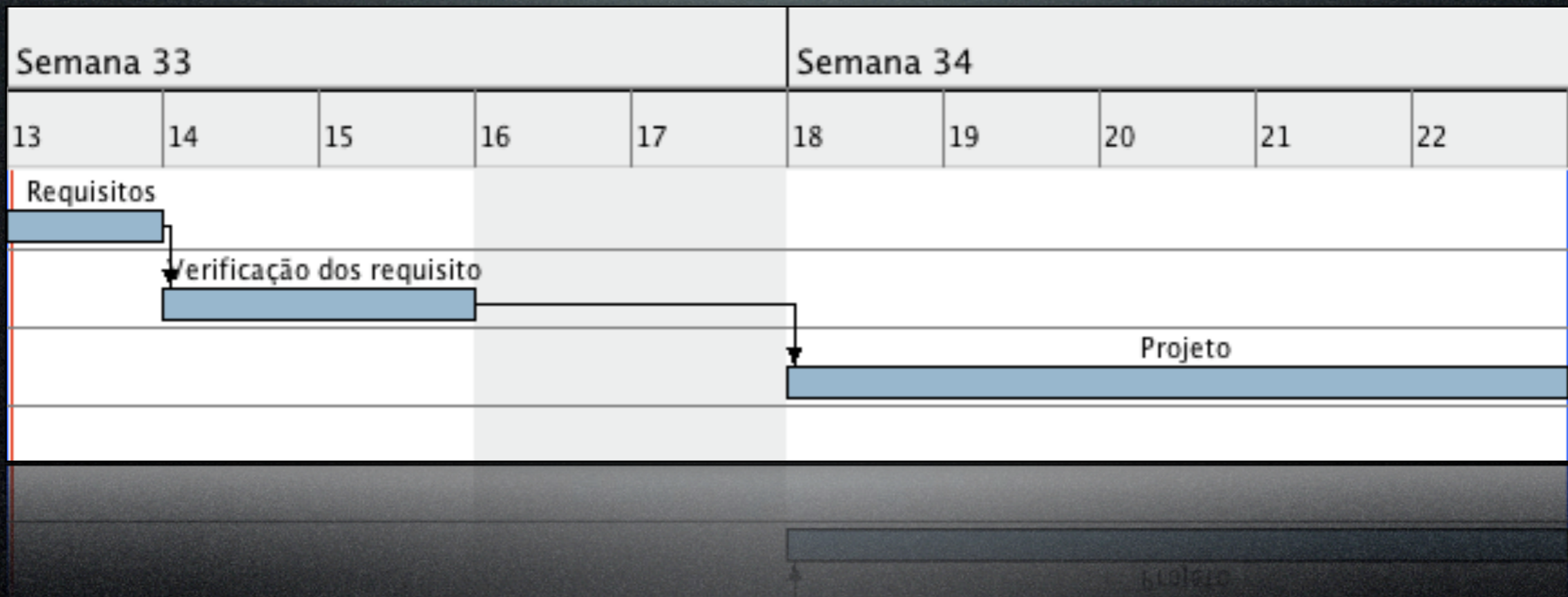
# Distribuição de esforço

- Definição do esforço em termos do número de homens-mês
- As características de cada projeto influenciam na parcela de esforço a ser dedicada em cada etapa
- O esforço dedicado à codificação é minimizado quando se utilizam boas regras de Engenharia de Software

# Regra 40-20-40



# Representação do cronograma



# Aquisição de Software

- Adquirir (ou licenciado) um sistema que atenda às especificações
- Adquirir um sistema e modificá-lo de forma a que o novo software atenda às especificações
- Encomendar o software a terceiros para que este atenda às especificações

# Porque comprar

- A data de entrega do produto precede a data de finalização caso seja desenvolvido internamente
- O custo de aquisição mais o custo de modificação é inferior ao custo de desenvolvimento

# Reengenharia

- Feito em sistemas legados que são fundamentais à realização de negócios de uma empresa e que oferecem grandes dificuldades de manutenção
- Os custos de manutenção de tais sistemas não justificam mais os benefícios

# Reengenharia

- Estimar o custo anual de manutenção
- Organizar, por ordem de prioridade, os sistemas legados
- Estimar os custos para realizar a reengenharia dos sistemas e estimar os custos de manutenção após realizada a reengenharia

# Reengenharia

- Calcular o tempo necessário para o retorno deste investimento
- Levar em consideração, melhorias em:
  - Confiabilidade
  - Desempenho
  - Interface

# Planejamento organizacional

- Escolher como a equipe de desenvolvimento será organizada
  - Decisão centralizada
  - Equipes informais
  - Equipes formais

# Decisão centralizada

- **n** indivíduos alocados a **m** tarefas
  - Pequeno grau de interação
  - Coordenação da equipe com o gerente de projeto

# Equipes informais

- **n** indivíduos alocados a **m** tarefas, com  **$m \leq n$** , formando equipes informais de desenvolvimento
  - Responsável ad-hoc de cada equipe
  - Coordenação entre as equipes é de responsabilidade do gerente do projeto

# Equipes formais

- **n** indivíduos são organizados em **k** equipes, cada equipe alocada para uma ou mais tarefas
  - Organização decidida por equipe
  - Coordenação a cargo da equipe e do gerente do projeto.

# Equipe Adicional

- Especialistas: Telecomunicação, Bancos de dados, Interface homem-máquina
- Apoio: Secretárias, Editores, Desenhistas, Bibliotecário

# Plano de Software

- Apresenta as informações levantadas quanto a custo e cronograma de desenvolvimento do software
- Consiste de um documento relativamente breve, e que deve ser encaminhado às diversas pessoas envolvidas no desenvolvimento do software

# Plano de Software

- Destaca-se:
  - Contexto e recursos necessários ao gerenciamento do projeto
  - Definição de custos e cronograma
  - Visão global do processo de desenvolvimento do software

# Plano de Software

## 1. Contexto

1.1. Objetivos

1.2. Funções principais

1.3. Desempenho

1.4. Restrições

## 2. Estimativas

2.1. Dados utilizados

2.2. Técnicas de Estimativa

2.3. Estimativas

## 3. Riscos do Projeto

3.1. Análise dos Riscos

3.2. Administração dos Riscos

# Plano de Software

## 4. Cronograma

4.1. Divisão do esforço no projeto

4.2. Timeline

4.3. Tabela de recursos

## 5. Recursos necessários

5.1. Pessoal

5.2. Software e Hardware

5.3. Outros recursos

6. Organização do Pessoal

7. Mecanismos de Acompanhamento

8. Apêndices