Engenharia de Software

Natália Schots

16/junho/2015 – Aula 17

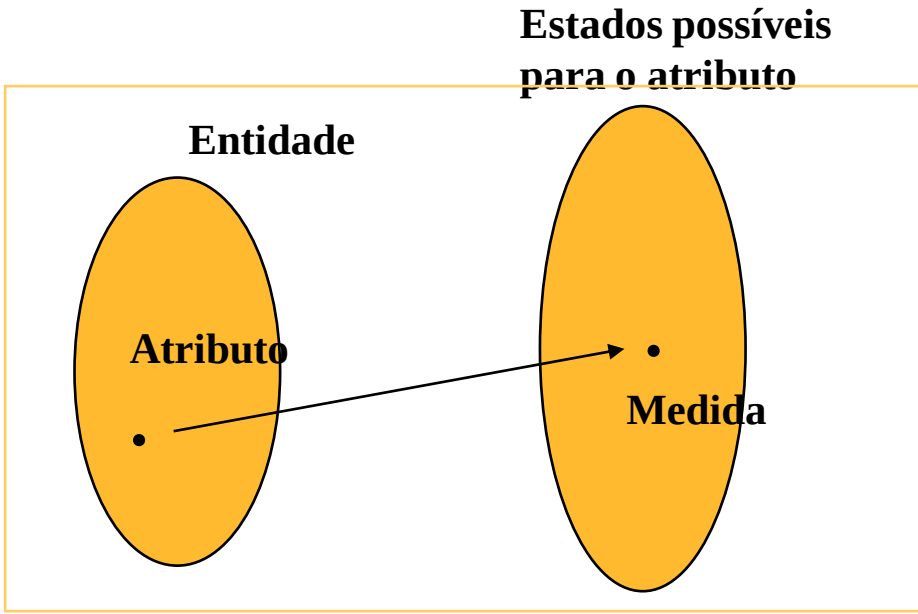
Agenda

- Medição de Software – Parte 2
  - Estimativas
  - Análise por Ponto de Função
  - COCOMO
  - *Planning Poker*

Na aula passada...

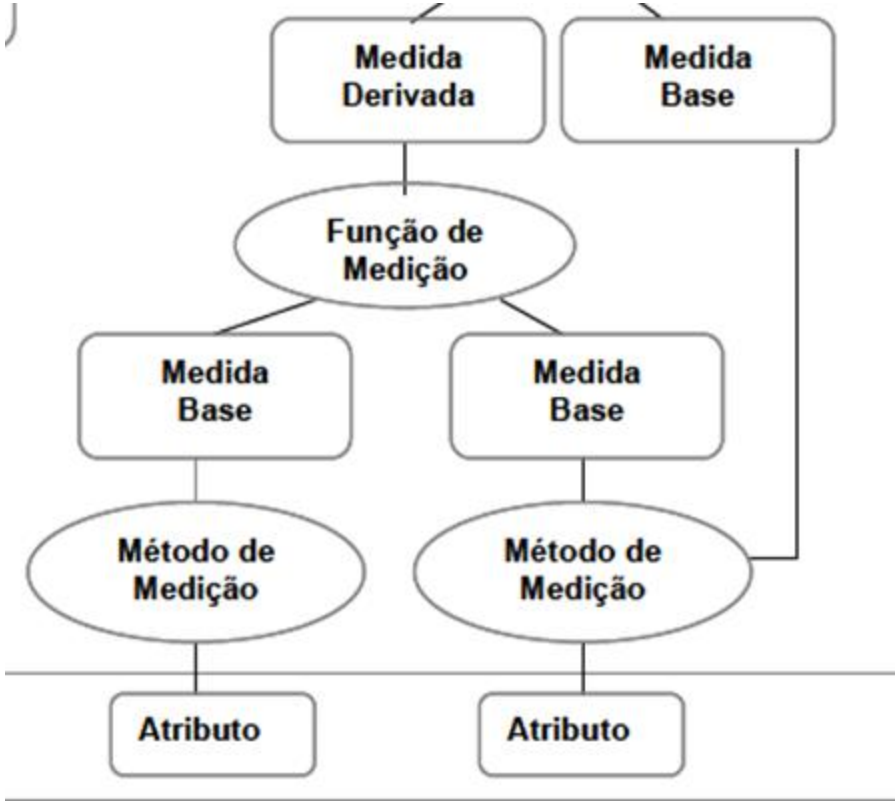
Medição de Software

- Medição
- Medida x Métrica



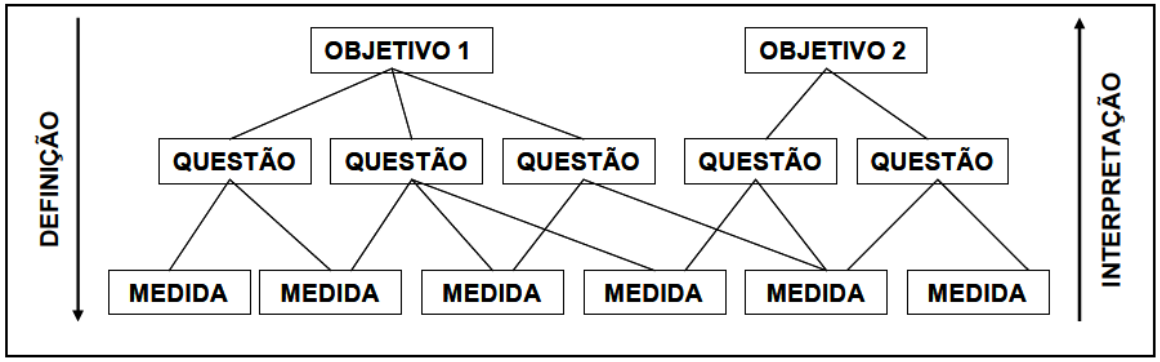
Medição de Software

- Medição
- Medida x Métrica
- Medida básica x derivada
- Definição operacional
- Plano de Medição
- Relatório de Medição



Abordagens

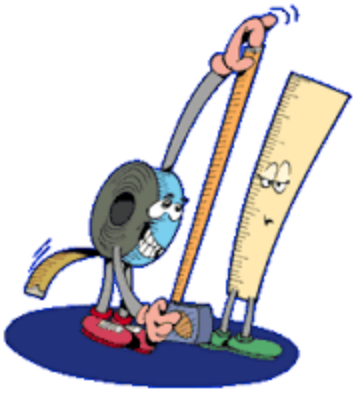
- GQM – *Goal Question Metric*



- PSM – *Practical Software Measure*

Five Core Metrics

- Tamanho
- Produtividade
- Esforço
- Qualidade
- Tempo



Estimativas

Dificuldades em estimar

- Natureza intangível do software
- Muitas variáveis impactam no desenvolvimento do software
- Falta de dados históricos
- Falta de caracterização adequada dos projetos

Tipos de estimativas (1/3)

- Estimativa análoga
  - Usa a duração real de uma atividade anterior semelhante como base
  - Usa informações históricas e opinião especializada
  - Geralmente é utilizada quando ainda não há informações detalhadas sobre o projeto
    - Em fases iniciais do projeto

Tipos de estimativas (2/3)

- Estimativa paramétrica
  - Determinada quantitativamente multiplicando a quantidade de trabalho a ser realizada pelo valor da produtividade
  - Dado de entrada mais importante é o **tamanho** do software
    - Ex: Análise por Pontos de Função, Pontos de Casos de Uso, COCOMO

Tipos de estimativas (3/3)

- Estimativa de três pontos
  - Média (ponderada) de três estimativas
    - Mais provável: estimativa obtida por meio da análise da disponibilidade dos recursos, produtividade, dependências etc.
    - Otimista: estimativa considerando o melhor cenário
    - Pessimista: estimativa considerando o pior cenário

COCOMO

O que é?

- *COnstructive COst MOdel*
  - Desenvolvido por Barry Boehm no início da década de 1980

- Modelo empírico para **esforço e tempo** baseado no tamanho do projeto, estimado em número de linhas de código

- Dividido em três níveis, aplicados quando novas informações sobre o projeto estão disponíveis

Níveis do Modelo

- COCOMO básico
  - Calcula o esforço e o tempo de desenvolvimento de um projeto de acordo com seu tamanho em linhas de código
- COCOMO intermediário
  - Considera alguns “geradores de custos” (*cost drivers*) no cálculo do esforço de desenvolvimento
- COCOMO avançado
  - Avalia o impacto dos geradores de custos nas diversas etapas do processo de desenvolvimento (análise, projeto, ...)

Categorias de Projetos

- Orgânicos (*organic*)
  - Projetos pequenos, com requisitos bem definidos e desenvolvidos por equipes experientes no domínio da aplicação
- Médios (*semi-detached*)
  - Projetos de médio porte, com requisitos mais instáveis e desenvolvidos por equipes maiores e com diversos níveis de experiência
- Complexos (*embedded*)
  - Projetos sujeitos a restrições técnicas (hardware, desempenho, ...) ou projetos grandes em um domínio de aplicação pouco conhecido

Parâmetros – COCOMO Básico

$$\text{Esforço} = a \cdot \text{KLOC}^{\,b}$$

$$\text{Tempo} = c \cdot \text{Esforço}^{\,d}$$

Tipo de Projeto	a	b	c	d
Orgânico	2.4	1.05	2.5	0.38
Médio	3.0	1.12	2.5	0.35
Complexo	3.6	1.20	2.5	0.32

Fatores de Ajuste

- Atributos do produto
  - Confiabilidade, complexidade, tamanho do BD
- Atributos do hardware
  - Restrições de memória, desempenho, ...
- Atributos de recursos humanos
  - Capacidade do analista, capacidade dos desenvolvedores, experiência no domínio, experiência com a linguagem, ...
- Atributos de projeto
  - Utilização de ferramentas, utilização de métodos de ES, cronograma de desenvolvimento do projeto

COCOMO Intermediário e Avançado

- Ajuste da medido inicial
  - Cada fator de ajuste é avaliado em uma escala ordinal de 0 e 5, variando de “pouco” até “muito”
  - Pesos publicados em tabelas são associados a cada fator de ajuste, de acordo com sua avaliação
  - O esforço calculado pelo COCOMO básico é multiplicado pelo produto dos pesos de cada fator de ajuste
  - A nova medida de esforço é utilizada para reavaliar o tempo de desenvolvimento

Vantagens x Desvantagens

- Por utilizar o número de linhas de código como base:
  - Vantagens
    - Possibilidade de automação do processo de medição
    - Existência de dados históricos na literatura
    - Relativa simplicidade de coleta
  - Desvantagens
    - Dependência de linguagem de programação
    - Desconsidera programas mais curtos e inteligentes
    - Nível de detalhes difícil de estimar no início do projeto

COCOMO II

- Atualização do COCOMO
- Possui nova classificação dos projetos
- Incorpora métodos por linhas de código e variantes de pontos por função
  - Pontos por função são recomendados no início do projeto

Análise por Pontos de Função (APF)

O que é?

- É uma técnica para medição de projetos, visando estabelecer uma medida de **tamanho**
- Focaliza na funcionalidade do software
- É uma medida do ponto de vista do usuário

Processo de contagem (1/2)

1. Cinco informações básicas são estimadas
   - Número de arquivos internos
   - Número de interfaces externas

   } Função de dados

   - Número de dados de entrada
   - Número de dados de saída
   - Número de consultas

   } Funções transacionais

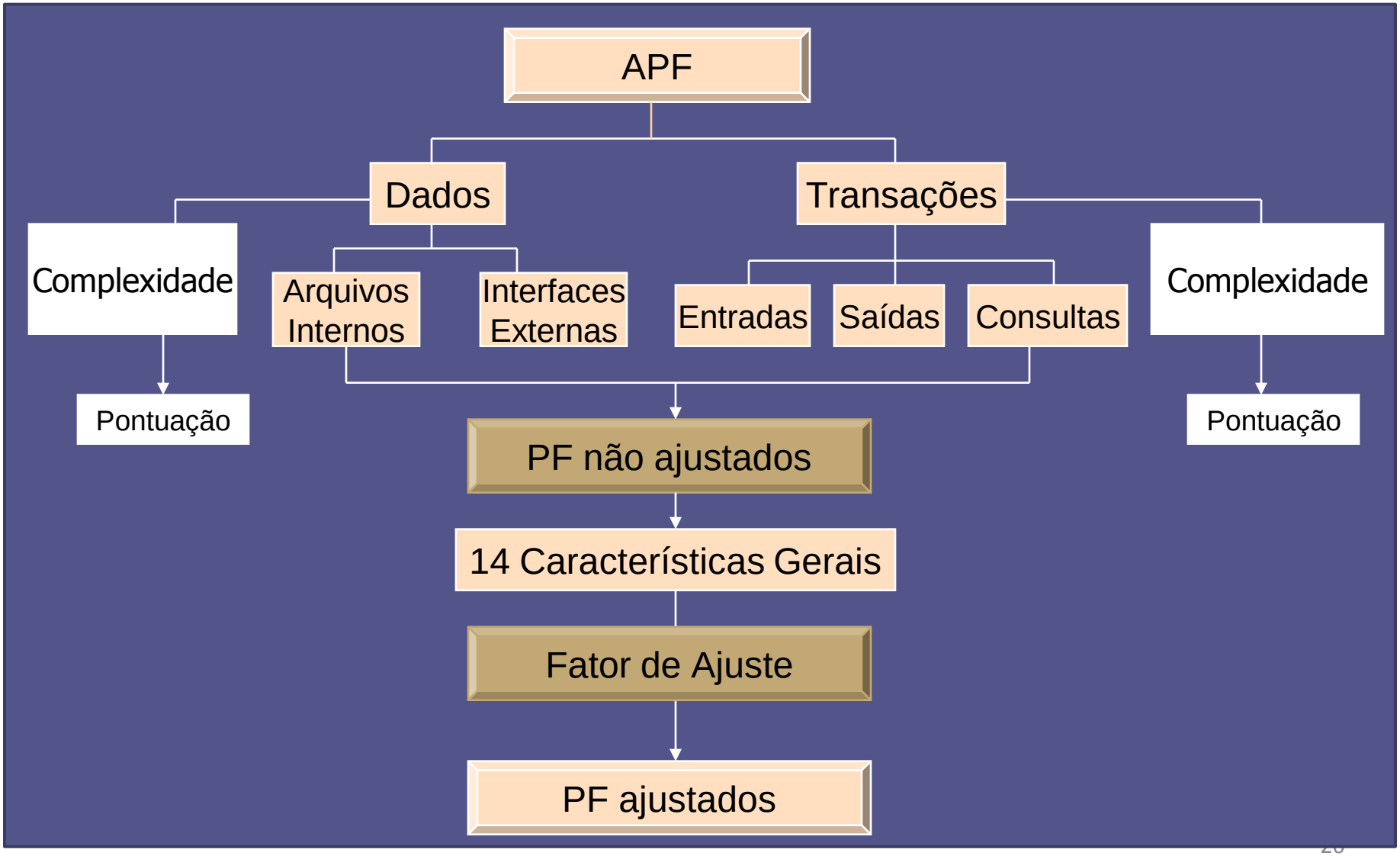
2. Um fator de complexidade é associado a cada informação
   - Baixa, Média, Alta

Processo de contagem (2/2)

3. As informações são ponderadas e agregadas (somadas)
   - Resultado: número de pontos por função

4. Fator de ajuste é calculado de acordo com as características da organização e é aplicado ao número de pontos por função anterior
   - Resultado: número de pontos por função ajustado

Estrutura da contagem

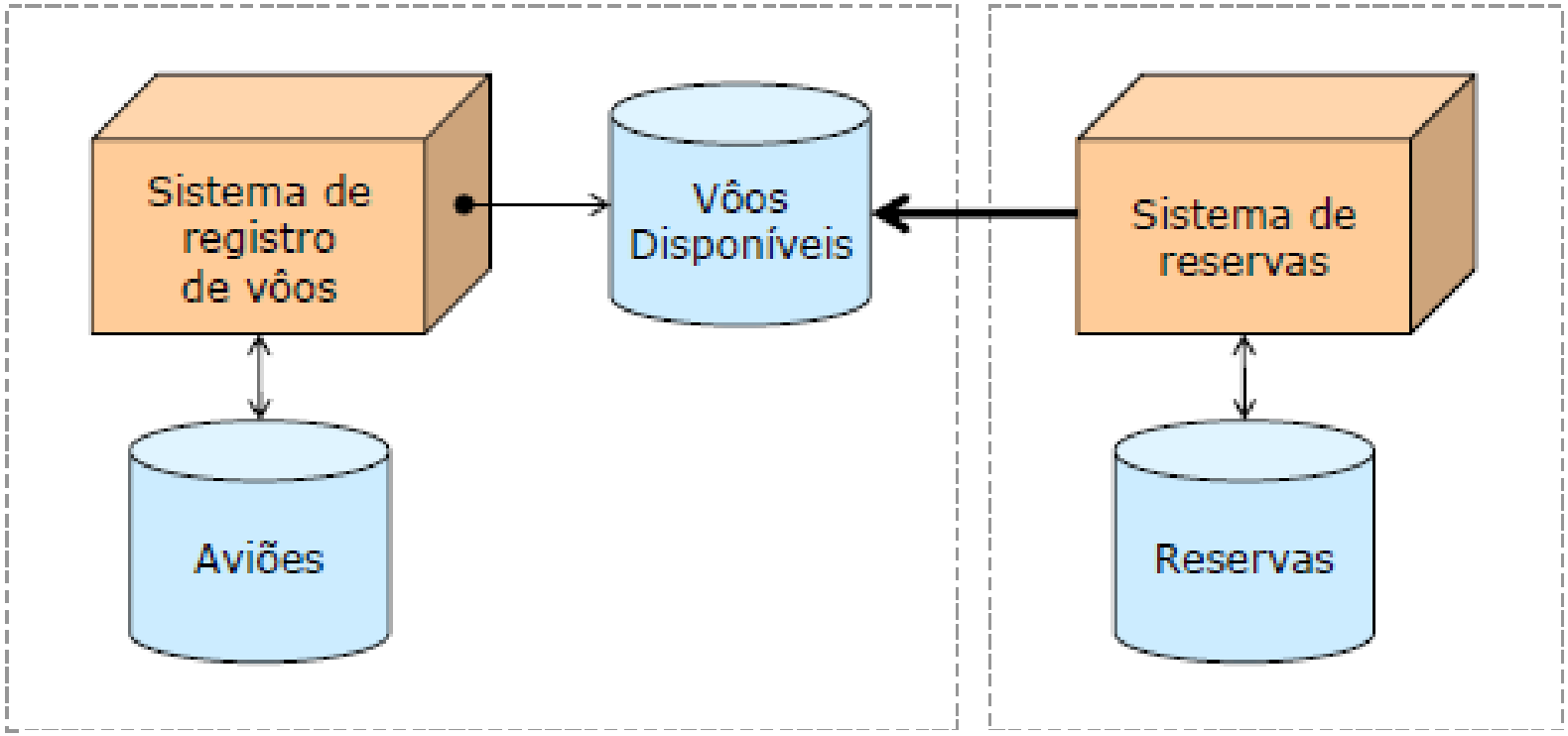


Funções de dados (1/2)

- Arquivo Interno (ILF –*Internal Logical File*)
  - É um grupo de dados logicamente relacionados entre si e mantidos dentro do escopo da aplicação
  - Deve ser identificado no nível do requisitos do sistema
- Interface Externa (EIF –*External Interface File*)
  - É um grupo de dados logicamente relacionados entre si e referenciados pela aplicação, embora sejam mantidos por outra aplicação
  - Deve ser identificado no nível do requisitos do sistema
- Para a identificação do ILF e EIF, o desenvolvedor deve construir um “modelo de dados” conceitual simplificado da aplicação

Funções de dados (2/2)

- Exemplo: sistema de controle de reservas de passagens aéreas



Funções Transacionais (1/3)

- Uma transação é a menor unidade de funcionalidade que satisfaz as seguintes condições:
  - É representativa para os usuários
  - Representa uma tarefa completa do ponto de vista do sistema
  - Deixa o sistema em um estado consistente

- Por exemplo, o requisito funcional “Manter Clientes” deve ser dividido nas seguintes transações:
  - Adicionar cliente
  - Alterar cliente
  - Remover cliente
  - Consultar cliente

Funções Transacionais (2/3)

- Dado de entrada (EI –*External Input*)
  - É um processo elementar que coleta e processa dados e informações de controle que são recebidos pela aplicação
  - Seu objetivo é manter o conjunto de informações registradas em um ou mais arquivos internos

- Dado de saída (EO –*External Output*)
  - É um processo elementar que processa e envia dados ou informações de controle para fora da aplicação
  - Seu objetivo é apresentar informações processadas para o usuário (deve haver pelo menos uma transformação na informação original)

Funções Transacionais (3/3)

- Consulta (EQ –*External Query*)
  - É um processo elementar que envia dados ou informações de controle para fora do contexto da aplicação
  - Seu objetivo é apresentar informações para o usuário sem que haja um processamento nem alteração no conteúdo de arquivos internos

Fatores de Ajuste

- Mecanismos de comunicação são necessários ?
- Processamento distribuído é necessário ?
- O nível de desempenho é crítico ?
- O sistema deve atender a transações de alta frequência?
- O sistema exige entrada de dados on-line?
- A interface com o usuário é complexa ?
- Os arquivos sofrem atualizações on-line?
- O processamento interno é complexo ?
- O código desenvolvido deve ser reutilizável ?
- ...

Vantagens x Desvantagens

- Vantagens
  - Independe da linguagem de programação
  - Dados que podem ser determinados no início do projeto
  - Existência de um processo padronizado para contagem
- Desvantagens
  - Medida parcialmente subjetiva
  - Dificuldades na automação da medida
  - Dados não podem ser interpretados fisicamente

Planning Poker

O que é? (1/2)

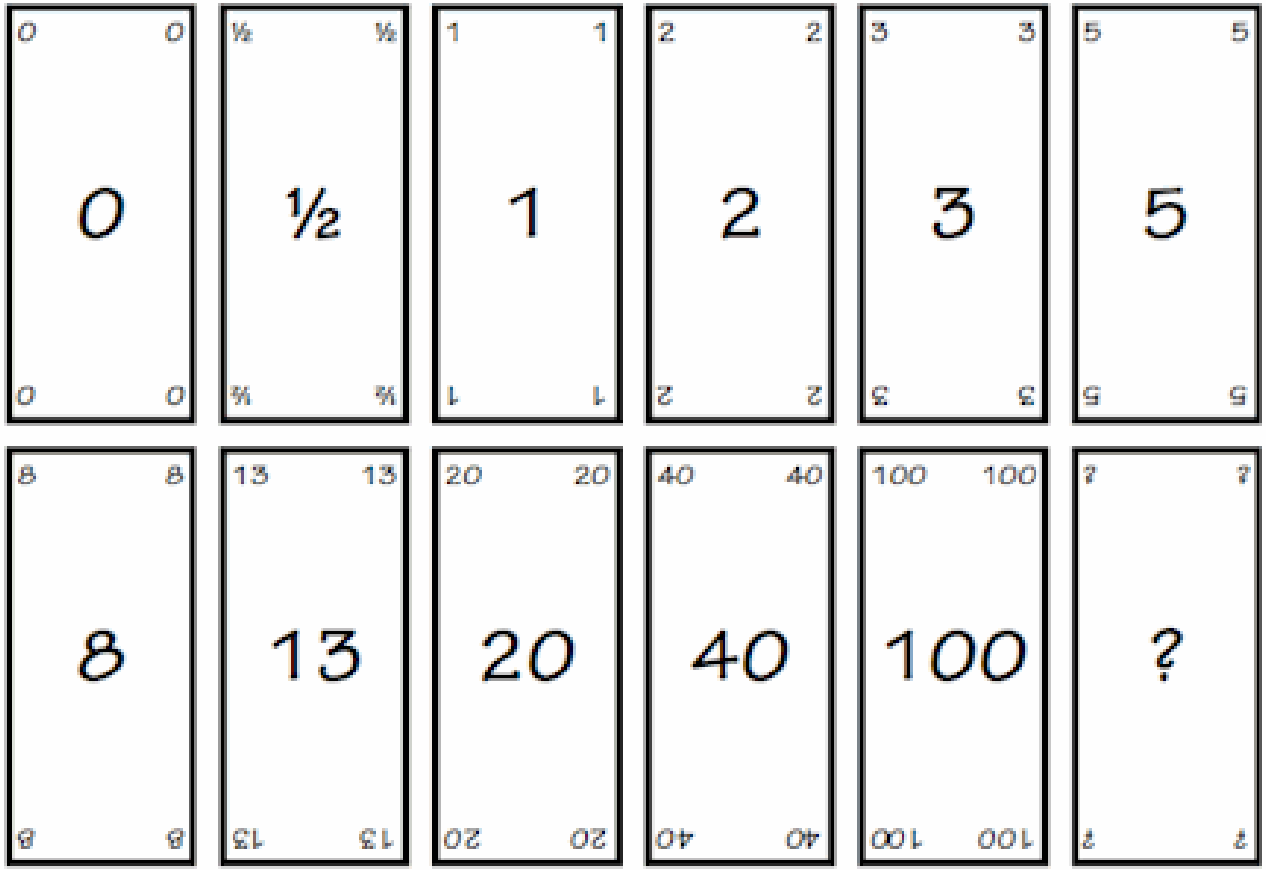
- Técnica de estimativa de **esforço** utilizada nas metodologias ágeis, principalmente no Scrum

- Obtém-se a estimativa por meio de um jogo de cartas, no qual todos os membros da equipe de desenvolvimento devem participar

- A partir deste jogo de cartas, chega-se um consenso sobre o esforço e tempo de uma estória (requisito)

Jogo de cartas (1/3)

- Baralho de 12 cartas em uma sequência (similar aos número de Fibonacci), cada um representa o esforço (horas ou dias)



Jogo de cartas (2/3)

1. O Product Owner descreve para a equipe Scrum a estória e fornece informações a respeito do seu valor de negócio. A seguir, o coordenador vai perguntar o esforço necessário para concluir a história aos membros da equipe

2. Cada membro da equipe deve pensar a respeito do tempo e esforço necessário para se implementar a estória lida. Os membros da equipe devem considerar todas as tarefas envolvidas: desenvolver, testar, criar design, etc. Então, deve escolher uma carta no baralho correspondente ao valor desta estimativa e coloca-a virada para baixo. Quando todos os membros fizerem o procedimento acima, então devem revelar as cartas escolhidas simultaneamente.

Jogo de cartas (3/3)

3. Todos avaliam os resultados e verificam se houve convergência entre as cartas mostradas, ou seja, todas as estimativas possuam valores aproximados para a mesma estória

4. Caso contrário, o Scrum Master solicita aos membros, que mostraram o menor e o maior valor estimado, que expliquem o motivo que os levaram a tal estimativa. Então, uma nova rodada é realizada até que as estimativas de esforço cheguem a uma convergência

5. A estimativa final da estória será o valor que tiver maior ocorrência ou a média entre as estimativas informadas

Vantagens x Desvantagens

- Vantagens
  - Enfatiza a importância da opinião de todos os membros da equipe
  - Equipe mais comprometida
  - Divertida
- Desvantagens
  - Algumas pessoas podem ser influenciadas por outras
  - Pouco objetivo

Referências

- Pressman, R.S.; “Engenharia de Software”; 6ª edição, Ed. McGraw-Hill, 2006
- Slides do professor Márcio Barros, 2013, “Gerência de Estimativas”
- Devmedia, “Planning Poker e Ideal Day: Técnicas de Abordagem de Estimativa Ágil”, disponível em: http://www.devmedia.com.br/planning-poker-e-ideal-day-tecnicas-de-abordagem-de-estimativa-agil/31220