

# Engenharia de Software I

- Métricas de software

# Métricas de software

---

- Um cliente ao contratar um determinado projeto de software tem interesse de saber:
  - Qual o custo?
  - Qual o prazo de entrega?
- A empresa quer saber:
  - Qual o “tamanho” do produto?
  - Qual o esforço?
  - A qualidade está conforme o esperado?
- As empresas necessitam **estimar o tamanho dos produtos de *software*** visando a **realização** de um melhor **planejamento** para a construção de produtos de *software* e, ainda, diminuir o risco da tomada de decisões erradas.



Plano de testes



Codificação



Testes de software



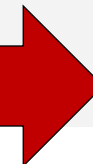
Implantação



Elicitação/  
Especificação  
de requisitos

Manutenção  
Evolução

GERÊNCIA DE PROJETOS



**Como controlar estatisticamente  
o desempenho do projeto?**

# Medição nas áreas de engenharia

---

## Medição nas áreas de engenharia:

- A medição é algo comum no mundo da engenharia, existindo vários tipos de grandezas para medição:
  - peso, temperatura, dimensões, tensão, entre outras.

# Medição nas áreas de Engenharia de Software

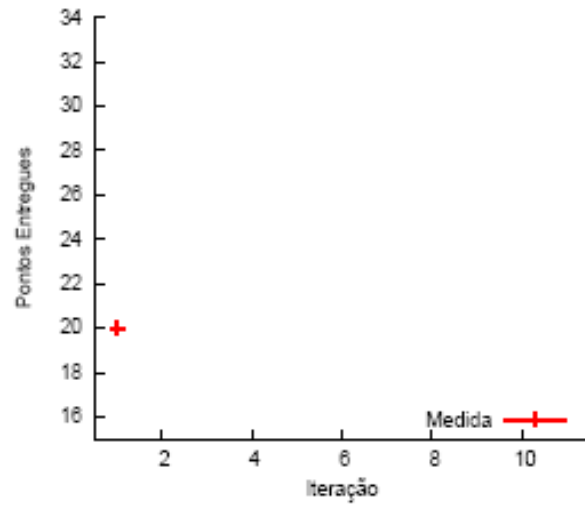
---

- Infelizmente a engenharia de software está longe de ter uma medição padrão amplamente aceita e sem nenhum fator subjetivo.
- Temos **dificuldade em concordar sobre o que medir e como avaliar o resultado** das medições obtidas.
- **Métricas de softwares** possibilitam realizar uma das atividades mais importantes do gerenciamento de projetos: o **planejamento**.

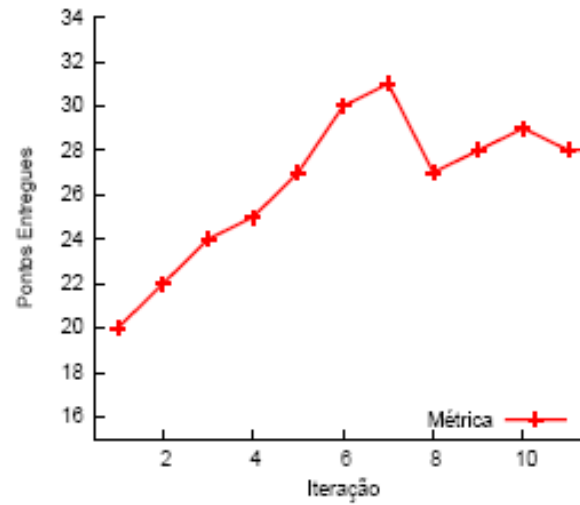
# Conceitos iniciais:

## Medida, métrica, medição e indicador

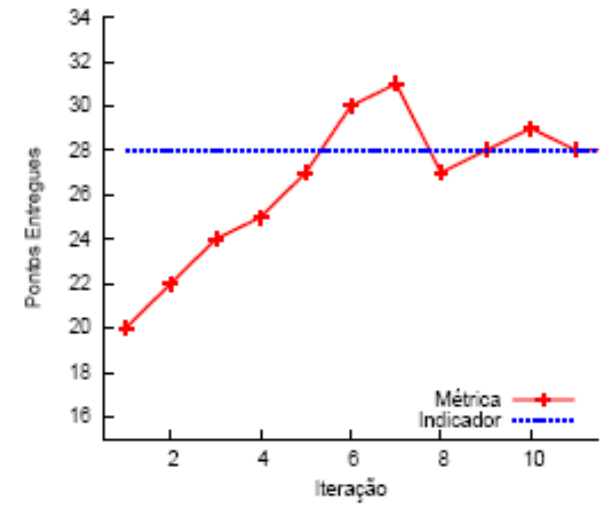
---



(a) Medida



(b) Métrica



(c) Indicador

# Conceitos iniciais

---

- **Medida:** fornece uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de algum atributo de um produto ou processo.
- **Medição:** ato de determinação de uma medida.
- **Métrica:** medida quantitativa do grau em que um sistema se encontra em relação a um determinado atributo
- **Indicadores:** métrica ou combinação de métricas que fornece uma compreensão de um processo, projeto, ou produto.



# Medição de Software

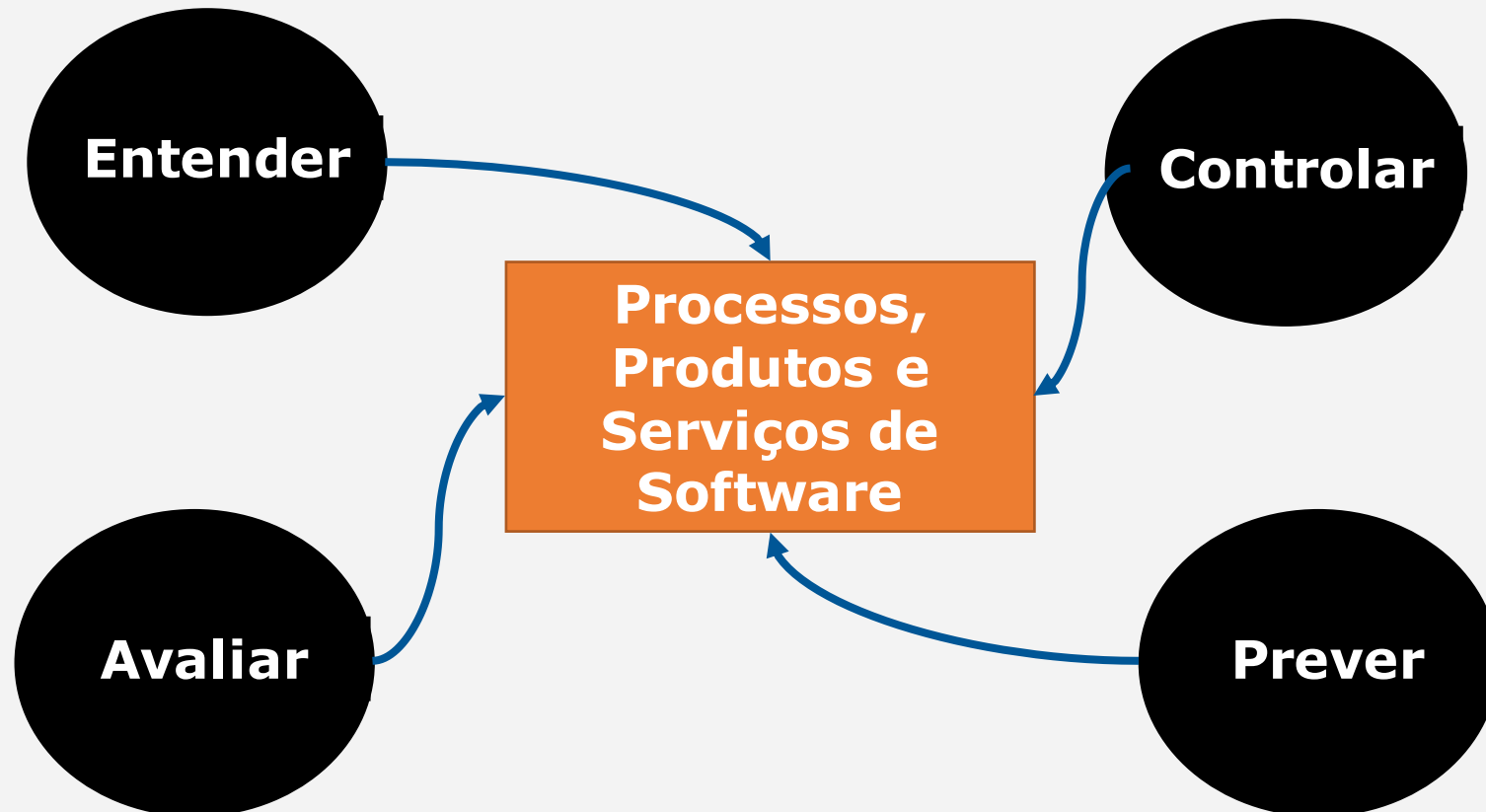
---

- Provê informação útil para que as organizações tomem decisões que impactam em seus objetivos de negócios.
- A medição de software é um dos **principais pilares da melhoria de processos de software.**

# Papéis da medição

---

- São quatro os principais papéis de Medições de Software:



# Objetivos da Medição de Software e utilidade das métricas

---

- **Entender:** ajudam a entender o comportamento e o funcionamento de produtos de software.
- **Controlar:** utilizadas para controlar processos, produtos e serviços de software.
- **Prever:** utilizadas para prever valores de atributos.
- **Avaliar:** utilizadas para determinar padrões, metas e critérios de aceitação.

# Objetivos da medição na ES

---

## **Perguntas que normalmente se busca responder com o uso de métricas:**

- Quanto tempo vou demorar para desenvolver esse projeto?
- Estou avançando de forma controlada, garantindo que o software seja entregue no tempo que foi prometido?
- Qual é o custo de cada atividade do processo?
- Qual é a produtividade da equipe?
- Qual é a qualidade do código gerado?
- Como medir a qualidade do produto?
- O cliente está satisfeito com nosso produto?

# Objetivos da medição na ES

Perguntas que normalmente se busca responder com o uso de métricas:

- Quanto tempo vou levar para desenvolver o projeto?
- Estou avançando mais rápido do que o planejado?
- Estou produzindo o que foi prometido?
- Qual a produtividade do processo?
- Qual a produtividade da equipe?
- Qual é a qualidade do código gerado?
- Como medir a qualidade do produto?
- O cliente está satisfeito com nosso produto?

**Como responder essas perguntas?**

# Métricas de software

---

- As análises baseadas em métricas são mais eficientes do que as que utilizam informações subjetivas.
- Utilizar dados históricos das métricas.

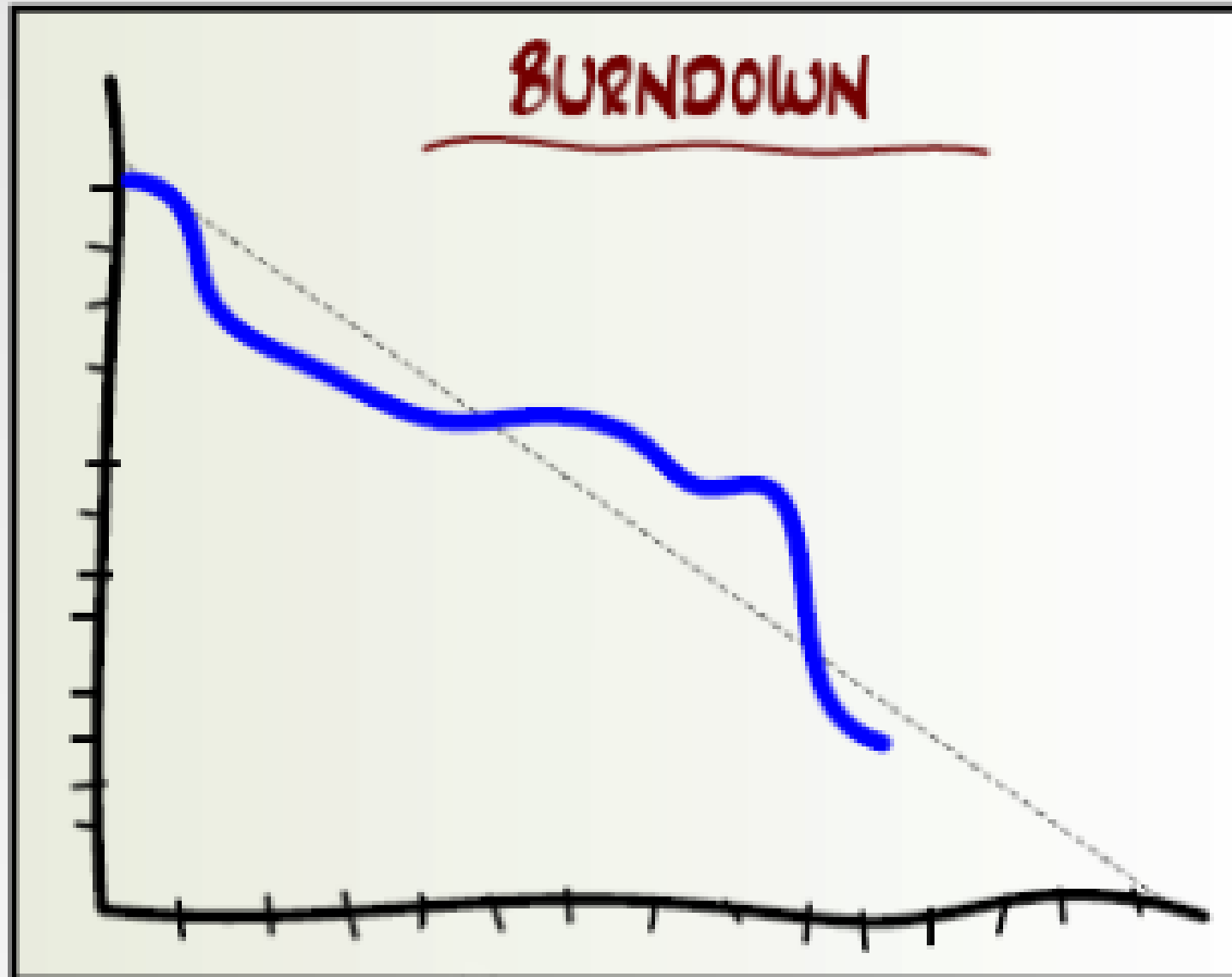
# Exemplos da definição formal métricas

---

## **Lembrando da definição de Métrica:**

- É uma medida quantitativa (fórmula de cálculo) do grau em que um sistema se encontra em relação a um determinado atributo.
- Ou seja, é uma comparação de medias/valores em períodos de tempo diferentes.

# Exemplo: Gráfico Burndown





# Gráfico Burndown da Sprint

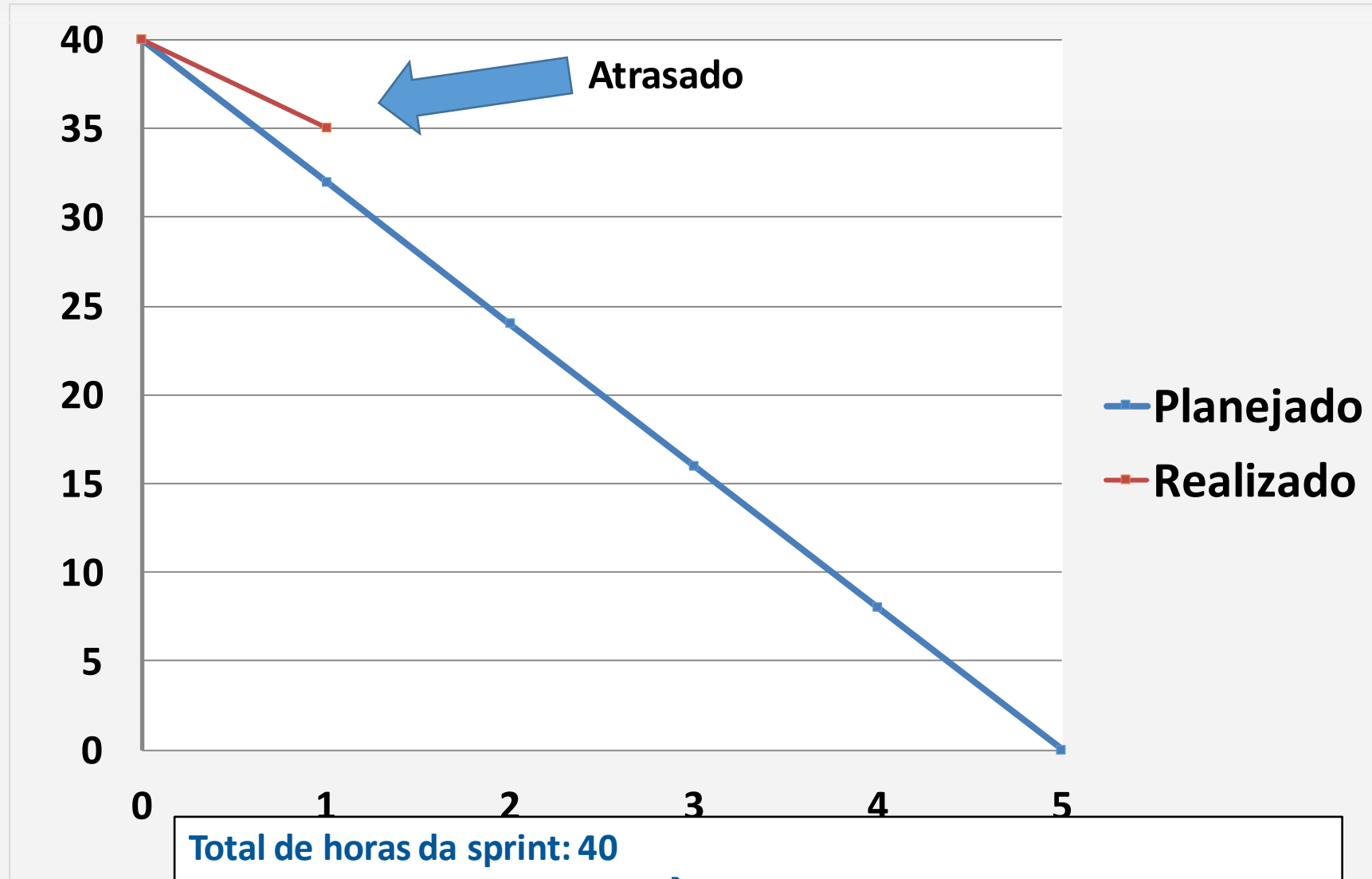
---

- O gráfico *burndown* é utilizado para medir o progresso das tarefas realizadas por uma equipe
- Apresenta uma diferença entre o número de pontos (ou horas) planejados e o número de pontos (ou horas) realizados, até um determinado momento da sprint
- no eixo horizontal é apresentado o número de dias da iteração e no eixo vertical é apresentado o número de pontos (ou horas) a serem desenvolvidos
- O gráfico apresenta duas linhas uma contínua que representa o desempenho ideal (planejado), sendo que no final não irá restar nenhum trabalho a ser concluído; e outra linha que apresenta o desempenho real da equipe ao longo da iteração

## Gráfico Burndown – Dia 0



# Gráfico Burndown – Dia 1



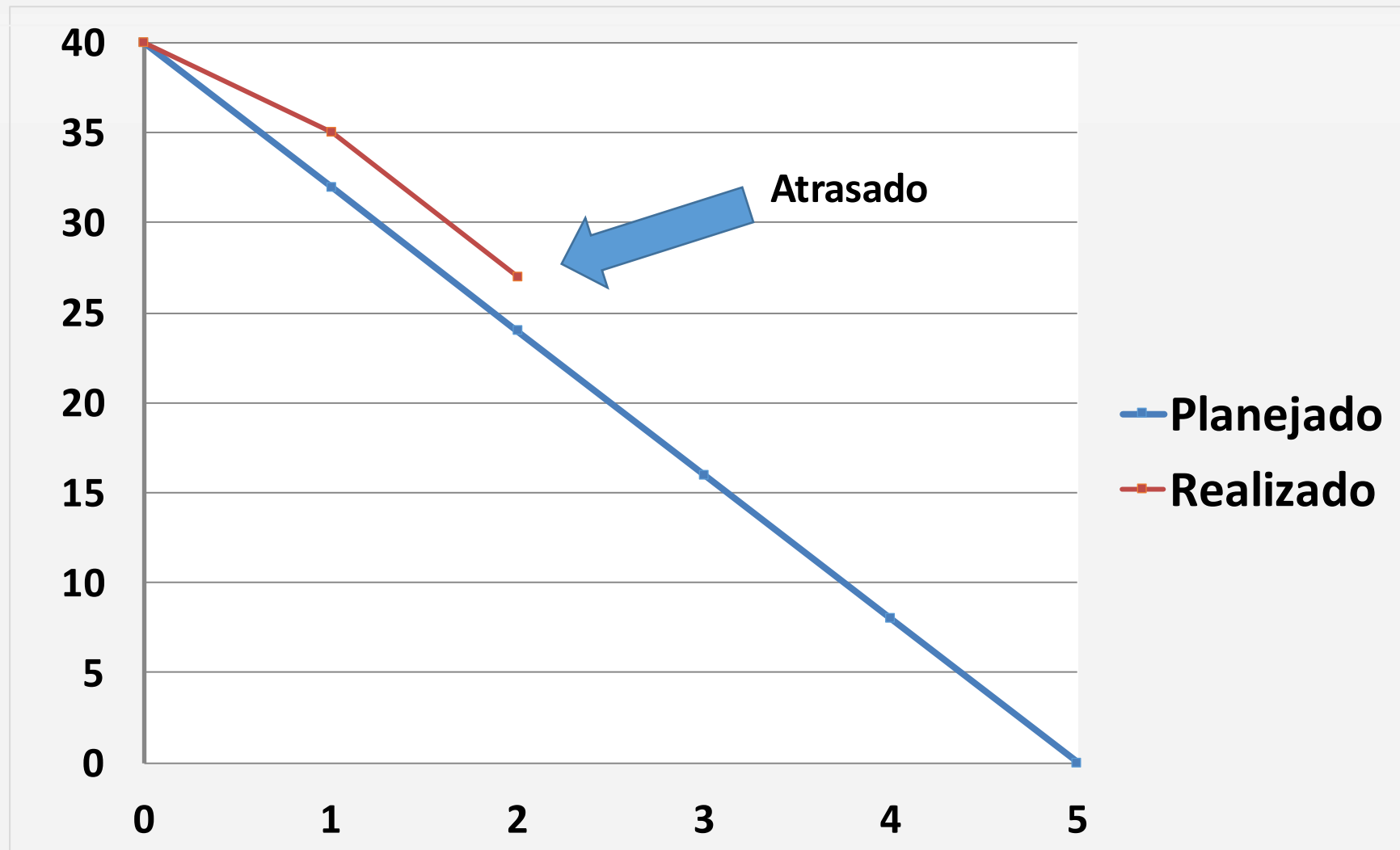
Total de horas da sprint: 40

Pontos previstos no 1º dia: 8 → acumulado: 32

Soma do trabalho realizado no 1º dia: 5

Ponto do gráfico burndown no 1º dia:  $40 - 5 = 35$

## Gráfico Burndown – Dia 2



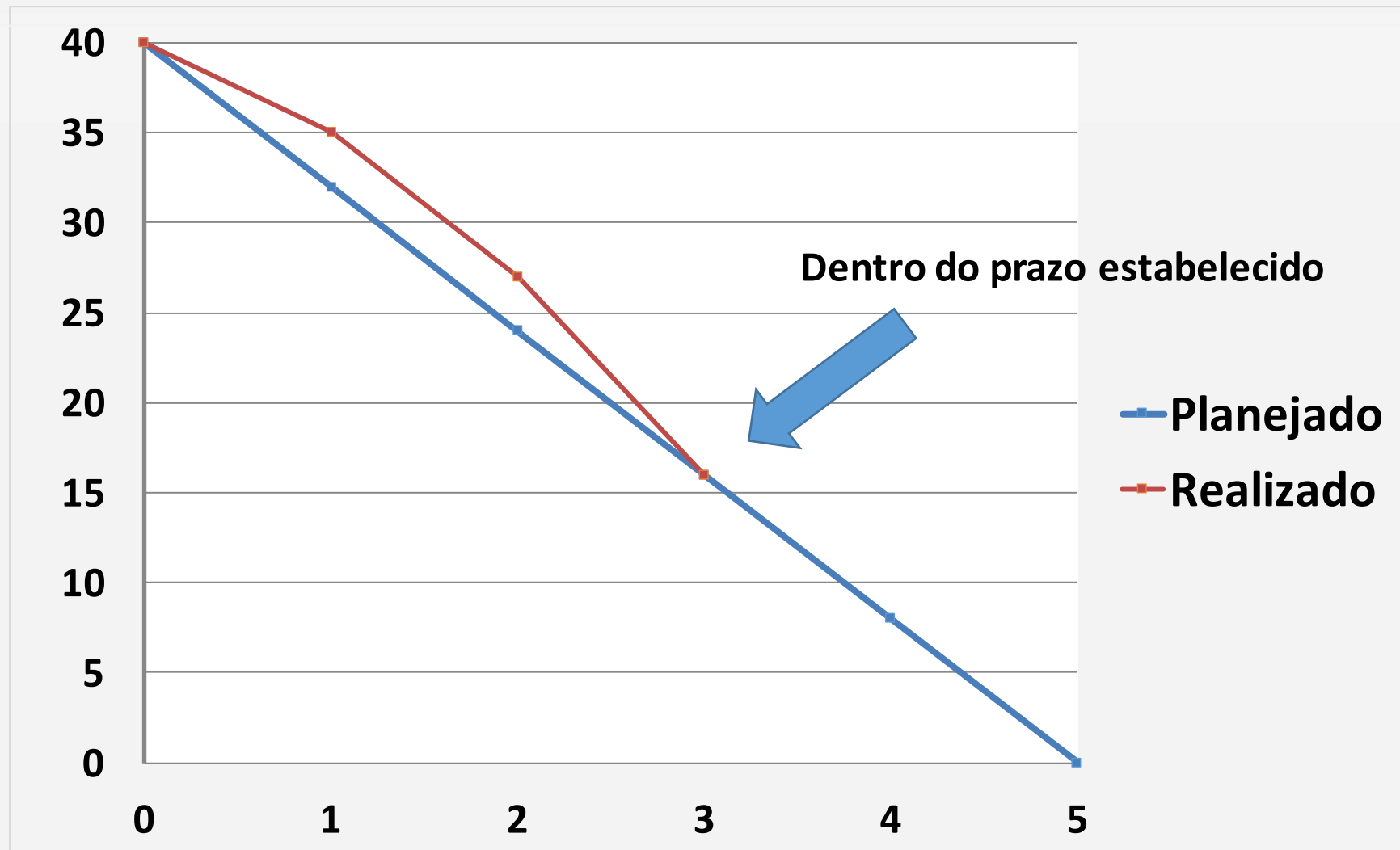
Total de horas da sprint: 40

Pontos previstos no 2º dia: 8 → acumulado previsto:  $32 - 8 = 24$

Soma do trabalho realizado no 2º dia: 8

Ponto do gráfico burndown no 2º dia:  $35 - 8 = 27$

## Gráfico Burndown – Dia 3



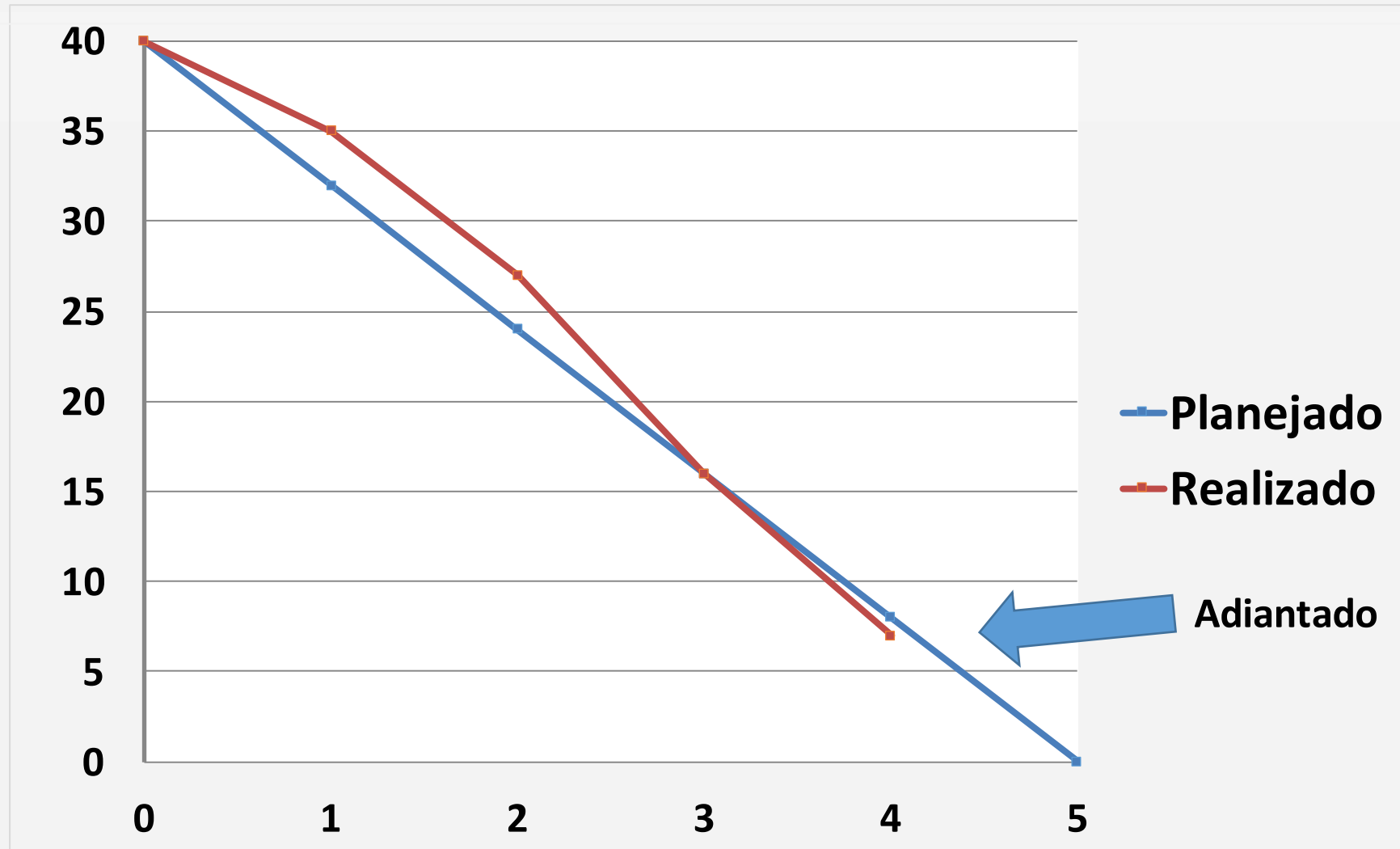
**Total de horas da sprint: 40**

**Pontos previstos no 3º dia: 8 → acumulado previsto:  $24 - 8 = 16$**

**Soma do trabalho realizado no 3º dia: 11**

**Ponto do gráfico burndown no 3º dia:  $27 - 11 = 16$**

## Gráfico Burndown – Dia 4



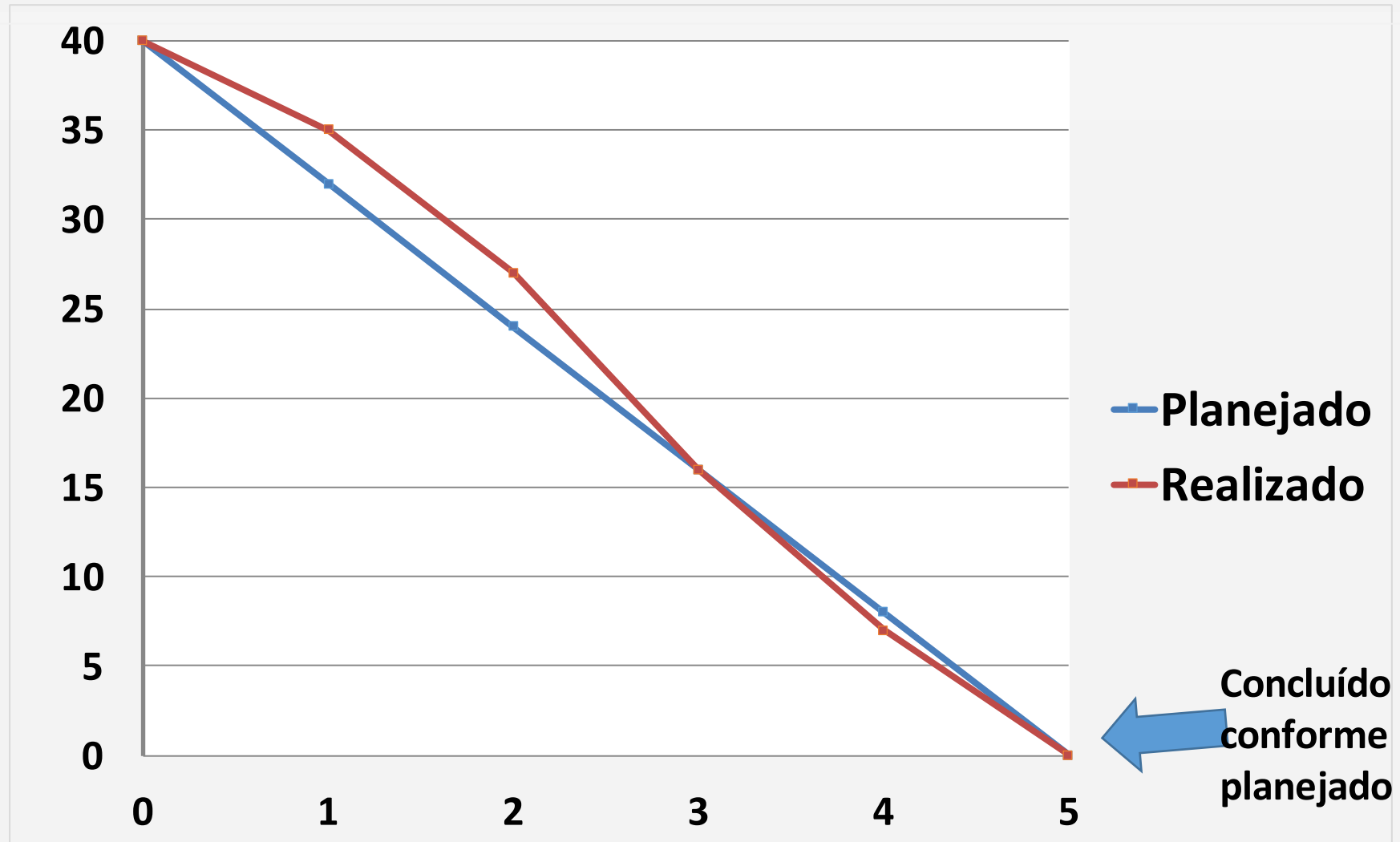
Total de horas da sprint: 40

Pontos previstos no 4º dia: 8 → acumulado previsto:  $16 - 8 = 8$

Soma do trabalho realizado no 4º dia: 9

Ponto do gráfico burndown no 4º dia:  $16 - 9 = 7$

## Gráfico Burndown – Dia 5



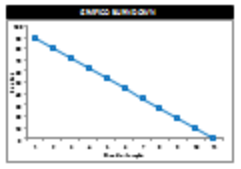
Total de horas da sprint: 40

Pontos previstos no 5º dia: 8 → acumulado previsto:  $8-8=0$

Soma do trabalho realizado no 5º dia: 7

Ponto do gráfico burndown no 5º dia:  $7-7=0$

# Nome da métrica: BURNDOWN DA ITERERAÇÃO

Categoria	Processo																						
Objetivo(s) de medição	Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos. Visualizar o progresso do projeto.																						
Equação	<div><p>Gráfico de Burndown da Iteração</p><p>O gráfico mostra a evolução dos pontos restantes ao longo de uma iteração de 10 dias. O eixo Y representa o 'Pontos Restantes' (0 a 10) e o eixo X representa o 'Dia da Iteração' (1 a 10). Uma linha azul desce linearmente de 10 pontos no dia 1 até 0 pontos no dia 10, representando o plano ideal de conclusão das tarefas.</p><table border="1"><thead><tr><th>Dia da Iteração</th><th>Pontos Restantes (Ideal)</th></tr></thead><tbody><tr><td>1</td><td>10</td></tr><tr><td>2</td><td>9</td></tr><tr><td>3</td><td>8</td></tr><tr><td>4</td><td>7</td></tr><tr><td>5</td><td>6</td></tr><tr><td>6</td><td>5</td></tr><tr><td>7</td><td>4</td></tr><tr><td>8</td><td>3</td></tr><tr><td>9</td><td>2</td></tr><tr><td>10</td><td>1</td></tr></tbody></table></div> <p>(a) Gráfico com linha estimativa de pontos/dia</p> <p>Onde:</p> <p>Número de tarefas = Número de pontos realizados na tarefa k do dia d Total de pontos realizados no dia d</p> <p>Onde:</p> <p>Ponto da linha de desempenho real no gráfico <i>burndown</i> que apresenta os pontos do dia d na iteração i</p>	Dia da Iteração	Pontos Restantes (Ideal)	1	10	2	9	3	8	4	7	5	6	6	5	7	4	8	3	9	2	10	1
Dia da Iteração	Pontos Restantes (Ideal)																						
1	10																						
2	9																						
3	8																						
4	7																						
5	6																						
6	5																						
7	4																						
8	3																						
9	2																						
10	1																						
Comportamento da métrica	Alvo-é-melhor																						
Fonte dos dados	Quadro de tarefas																						



**Exemplo:**  
**Grau de atendimento aos requisitos funcionais**

## Nome da métrica: Grau de atendimento aos requisitos funcionais

<b>Categoria</b>	Qualidade
<b>Objetivo(s) de medição</b>	Avaliar a qualidade do produto após sua homologação
<b>Equação</b>	Onde: Grau de atendimento aos requisitos funcionais na iteração i Número total de requisitos funcionais (histórias) da iteração i Número de requisitos funcionais (histórias) aceitos pelo cliente na iteração i
<b>Comportamento da métrica</b>	Maior-é-melhor
<b>Fonte dos dados</b>	Requisitos entregues ao cliente

# **Métricas do Kanban**

# Kanban

---

É um método ágil baseado nos princípios Lean Production – Produção Enxuta (Sistema Toyota de produção)

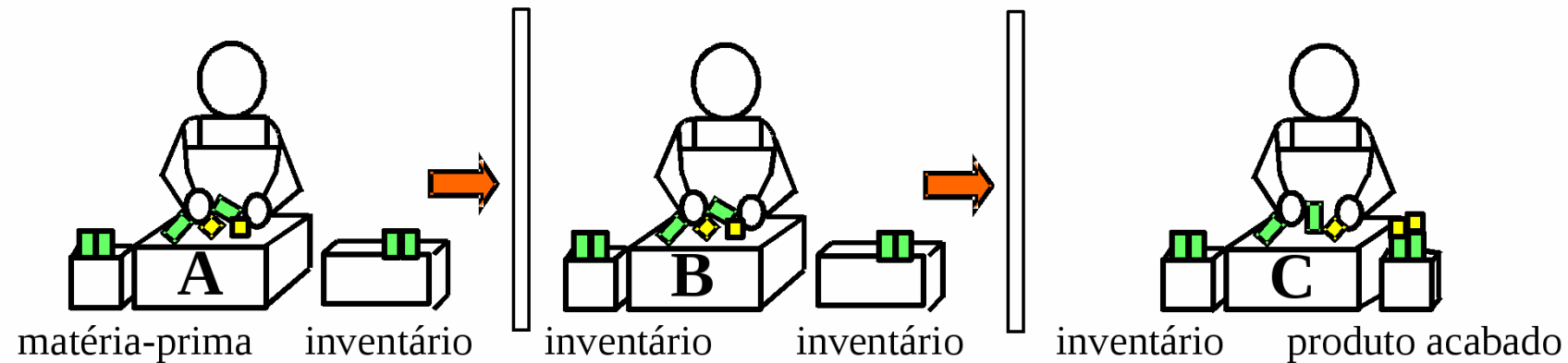
Palavra Kanban:

- Kan: visual
- Ban: cartão
- Kanban quer dizer sinalização visual

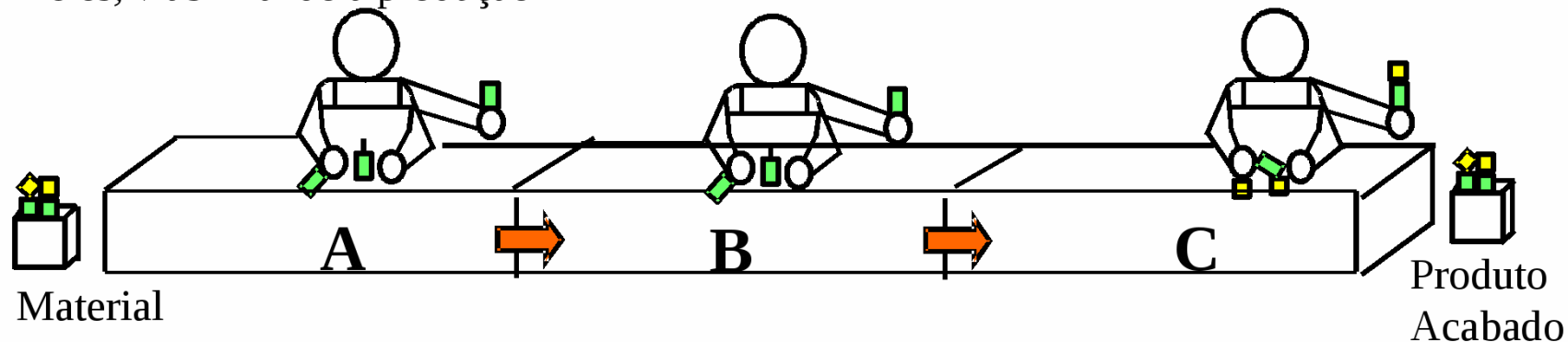
**Foco no trabalho em processo, apresentando a evolução do trabalho de forma visual**

# Produção puxada versus produção empurrada

Tradicional (Tipo funcional) - Os trabalhadores estão separados



Fluxo contínuo: Elimina as verdadeiras “estagnações” de trabalho em cada processo e entre eles, viabilizando a produção  $1 \times 1$



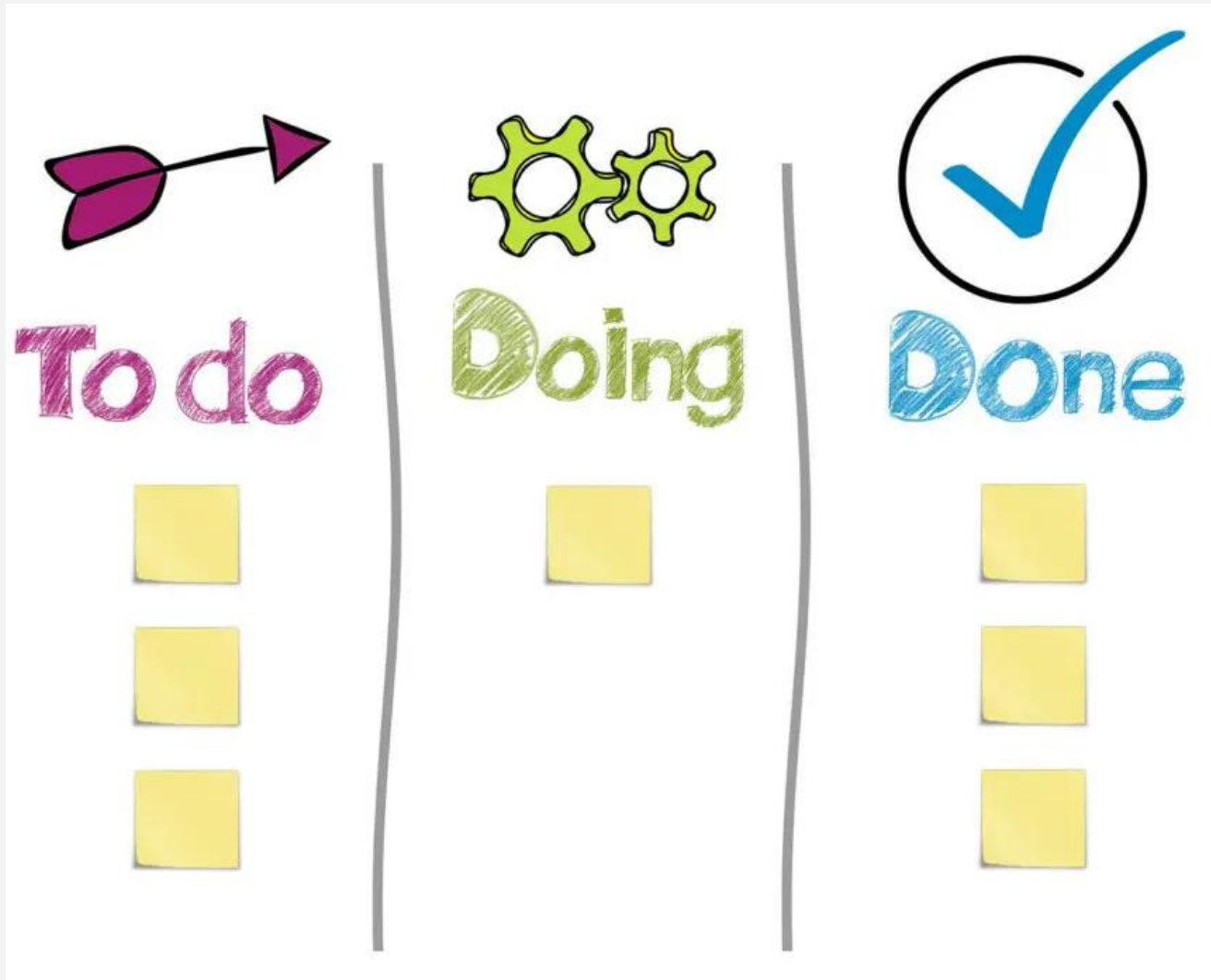


**É necessário:**

- 1) Mapear o fluxo de atividades (fluxo de valor)
- 2) Ajustar as equipes para conseguirem trabalhar de forma sincronizada e coordenada
- 3) Eliminar gargalos de produção
- 4) Criar mecanismos de resolução rápida de problemas

# Quadro Kanban simples

---



# Quadro Kanban de software – Fluxo geral completo





# Kanban - Quadro físico X Quadro Digital



# Benefícios do quadro Kanban

---

Visibilidade

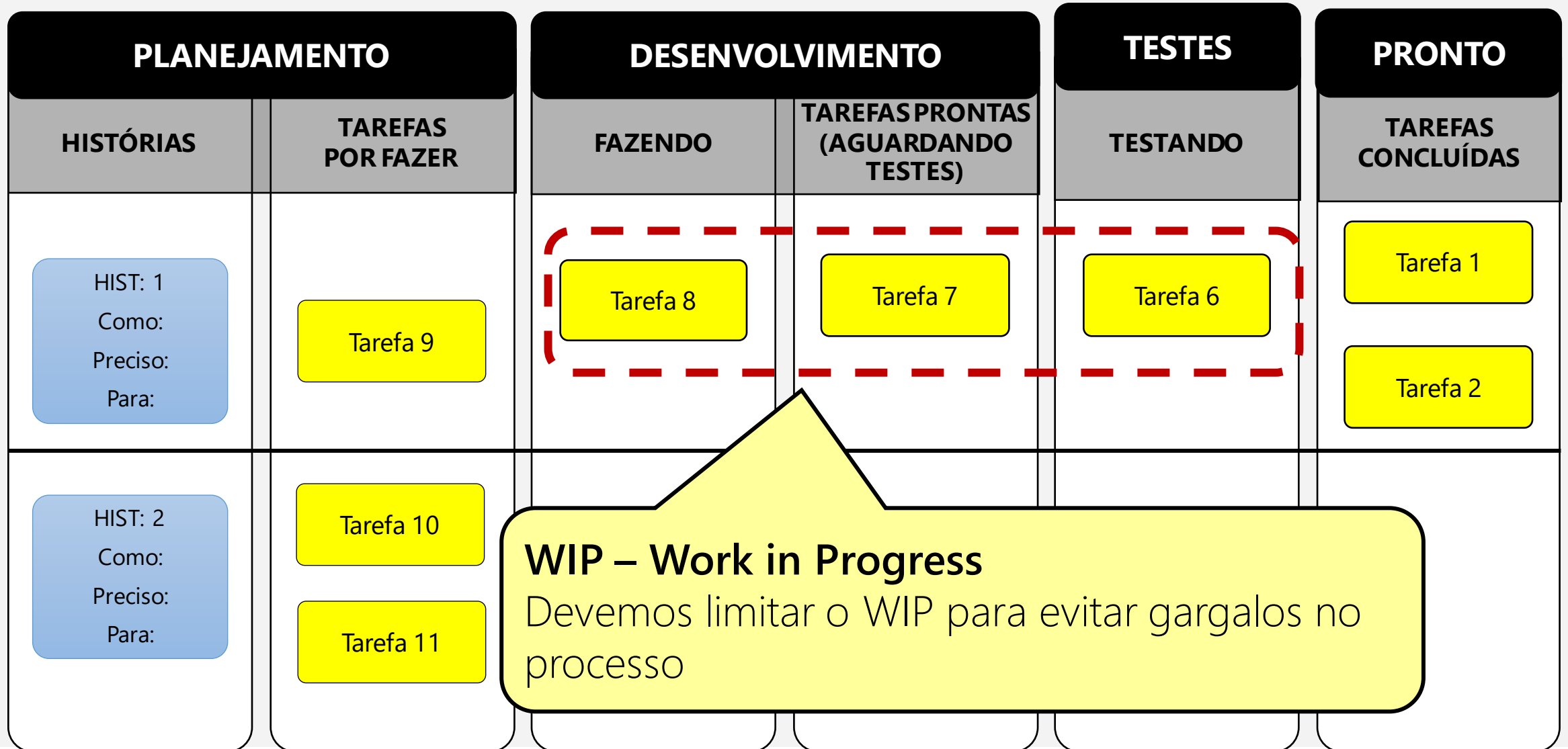
Transparência

Visualizar gargalos das atividades

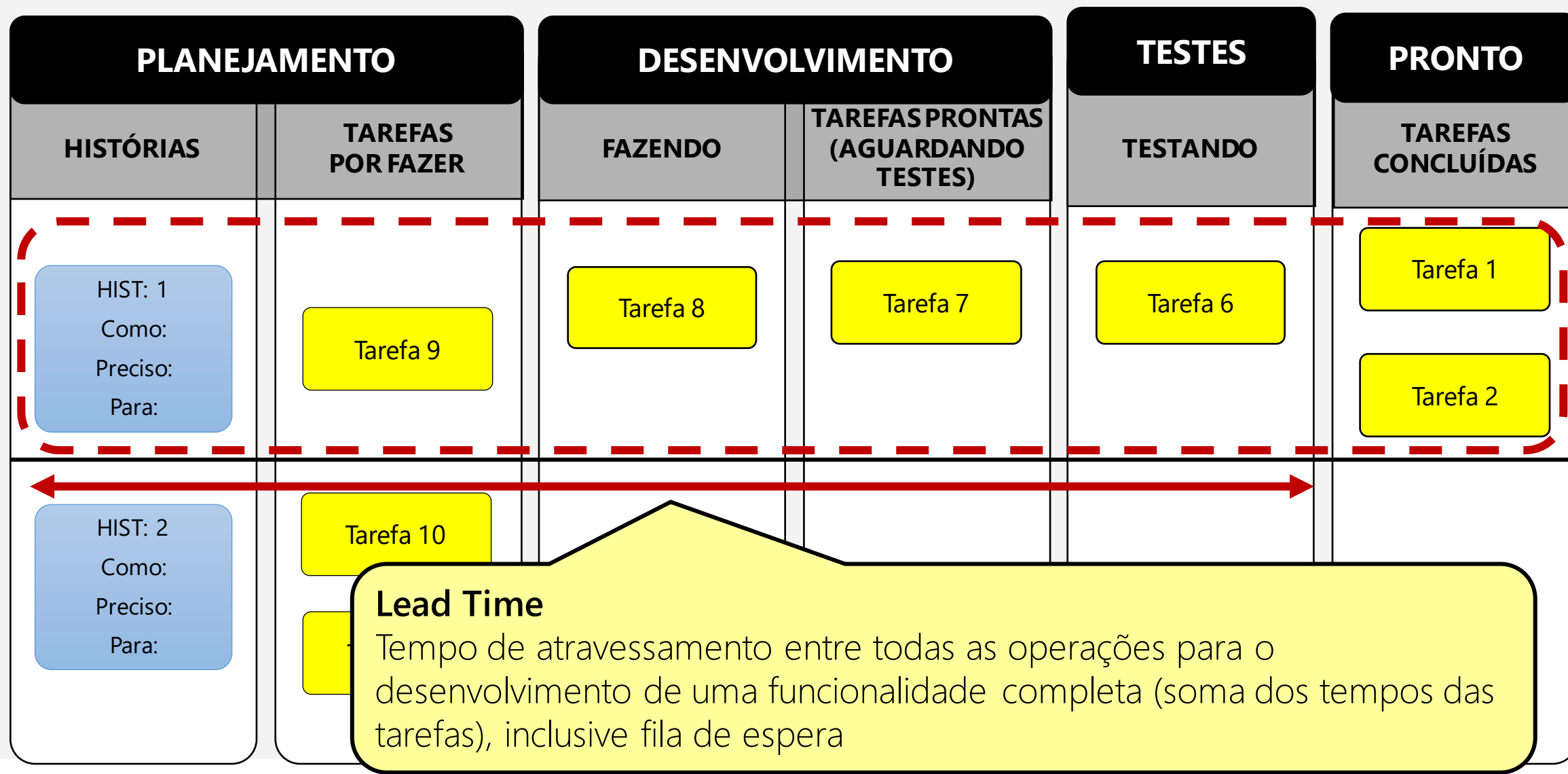
Visualizar o fluxo de trabalho

Visualizar os impedimentos

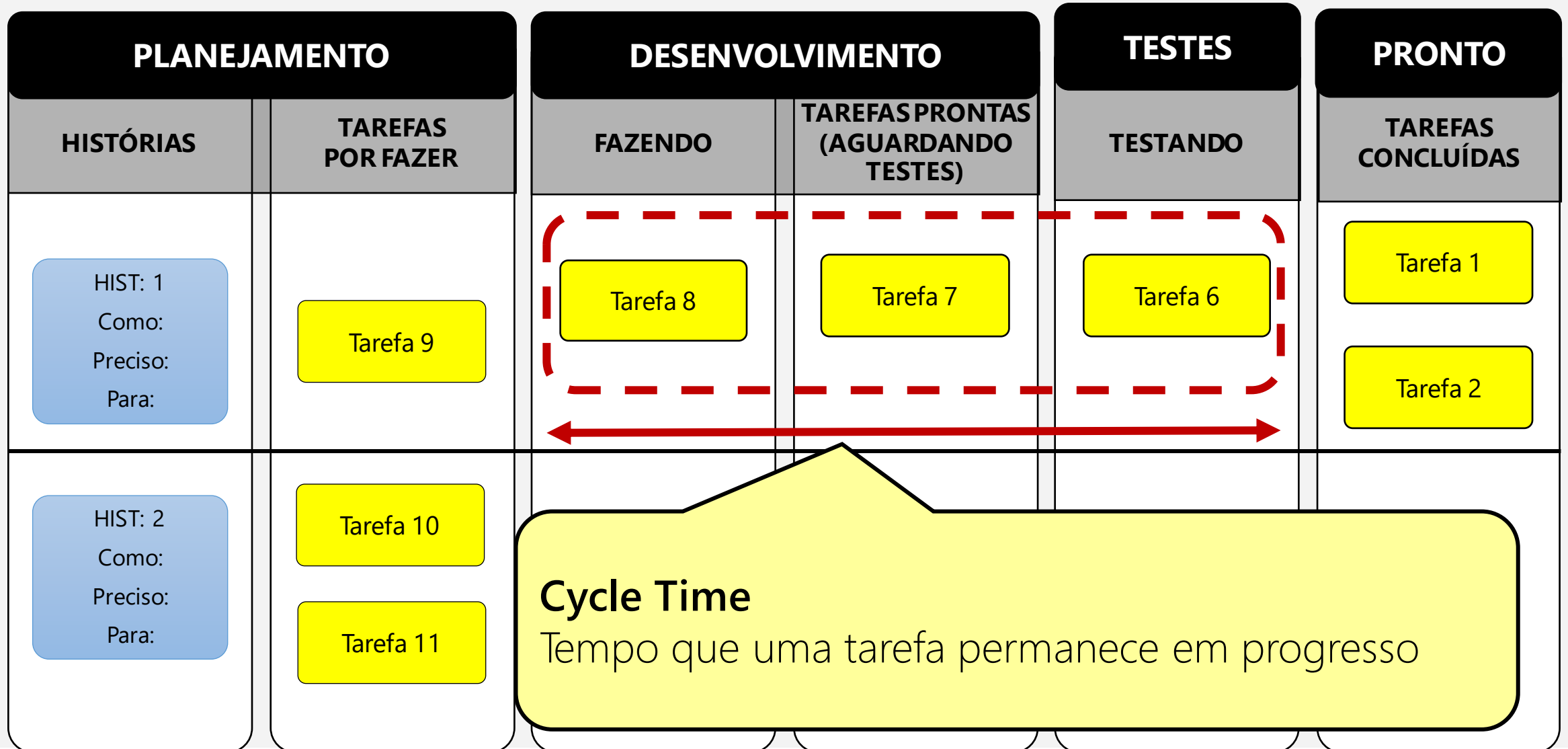
# Kanban de uma sprint – métrica: Work in Progress



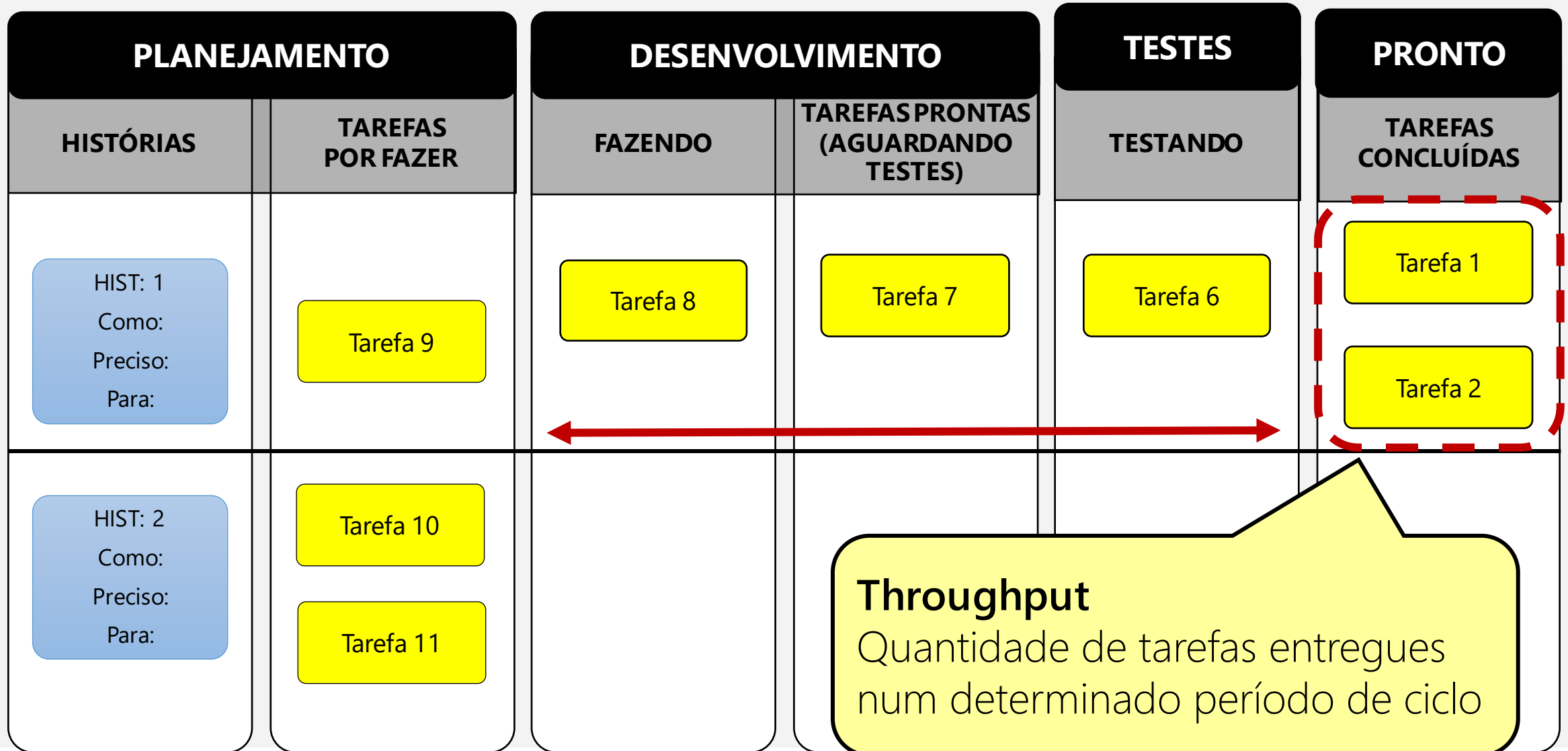
# Kanban de uma sprint – métrica: Lead Time



# Kanban de uma sprint – métrica: Cycle Time



# Kanban de uma sprint – métrica: Throughput



# Paradigma GQM - Goals Question Metrics

OK, eu entendi a importância de usar métricas, mas agora é necessário decidir quais métricas serão adotadas...

**Como vocês fariam para decidir quais métricas devem ser implantadas na sua empresa?**



# GQM - Goals Question Metrics

---

- Usado para definir o conjunto de métricas a ser utilizado pela empresa de software
- Baseia-se no fato de que deve existir uma necessidade clara associada a cada métrica

# GQM - Goals Question Metrics

---

Inicia-se com a identificação dos interessados na medição.

## **GOALS (objetivos)**

- Com base nos interessados, estabelecem-se os principais objetivos da medição para a organização, o projeto ou uma tarefa específica.
  - Reduzir defeitos
  - Aumentar produtividade, etc.

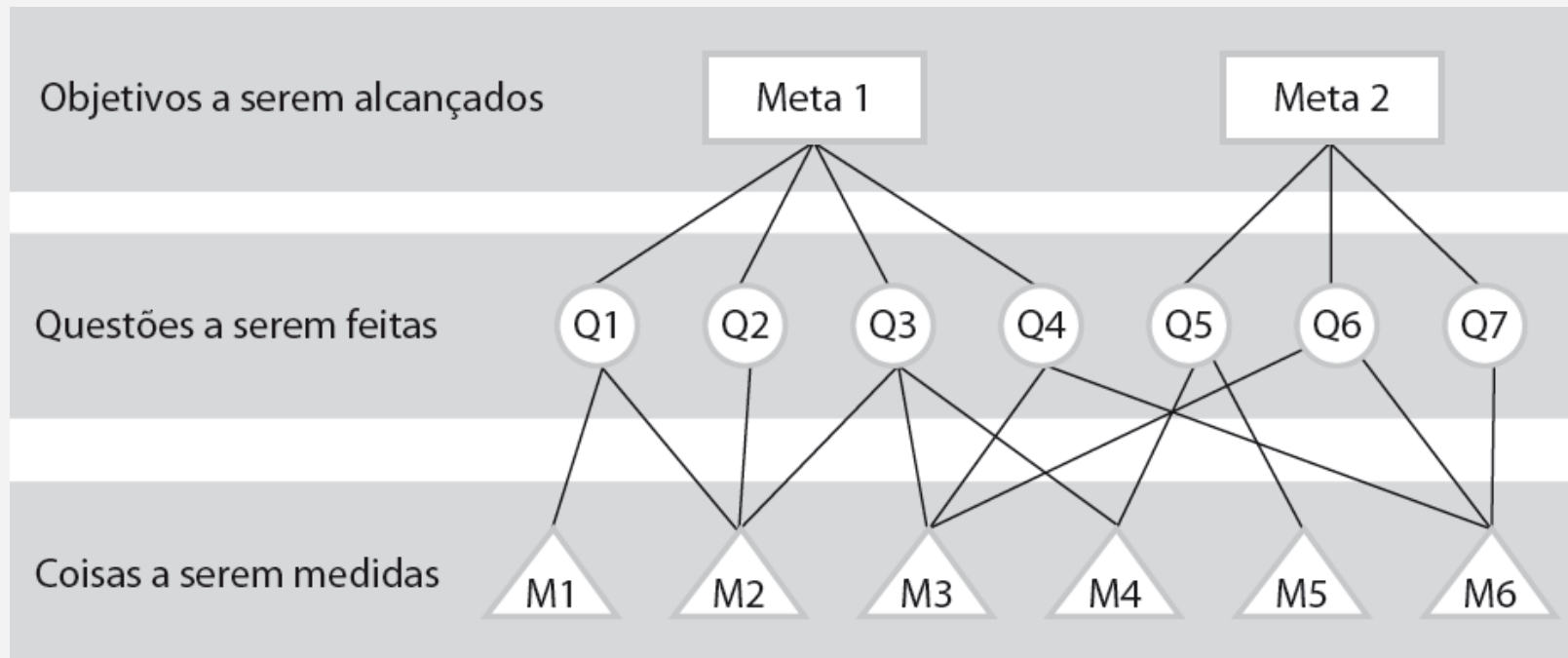
## **QUESTIONS (questões, perguntas)**

- A partir dos objetivos, geram-se perguntas cujas respostas dirão se os objetivos foram ou não alcançados
  - Qual a taxa de defeito atual?
  - Qual a taxa de defeito após a implantação do novo processo?

## **METRICS (métricas)**

- A partir das perguntas são definidas as métricas que permitem responder cada uma das perguntas: Que dados serão necessários? Quais os formatos? Como coletar (fórmula e processo)? Onde armazenar? Como utilizar?
  - Número defeitos por produto
  - Número de defeitos por status

# GQM - Goals Question Metrics



# Exemplo

---

- **OBJETIVO (GOALS):**

Garantir que os projetos sejam entregues no prazo prometido ao cliente

- **Perguntas (QUESTIONS):**

?

- **Métricas (METRICS):**

?

# Exemplo

---

- **OBJETIVO (GOALS):**

Garantir que os projetos sejam entregues no prazo prometido ao cliente

- **Perguntas (QUESTIONS):**

- Qual a nossa produtividade atual?
- Qual o nosso acerto de estimativa das tarefas em relação ao tempo previsto e o tempo realizado?
- Qual a variação de produtividade no decorrer na semana de trabalho?

- **Métricas (METRICS):**

?

# Exemplo

---

- **OBJETIVO (GOALS):**

Garantir que os projetos sejam entregues no prazo prometido ao cliente

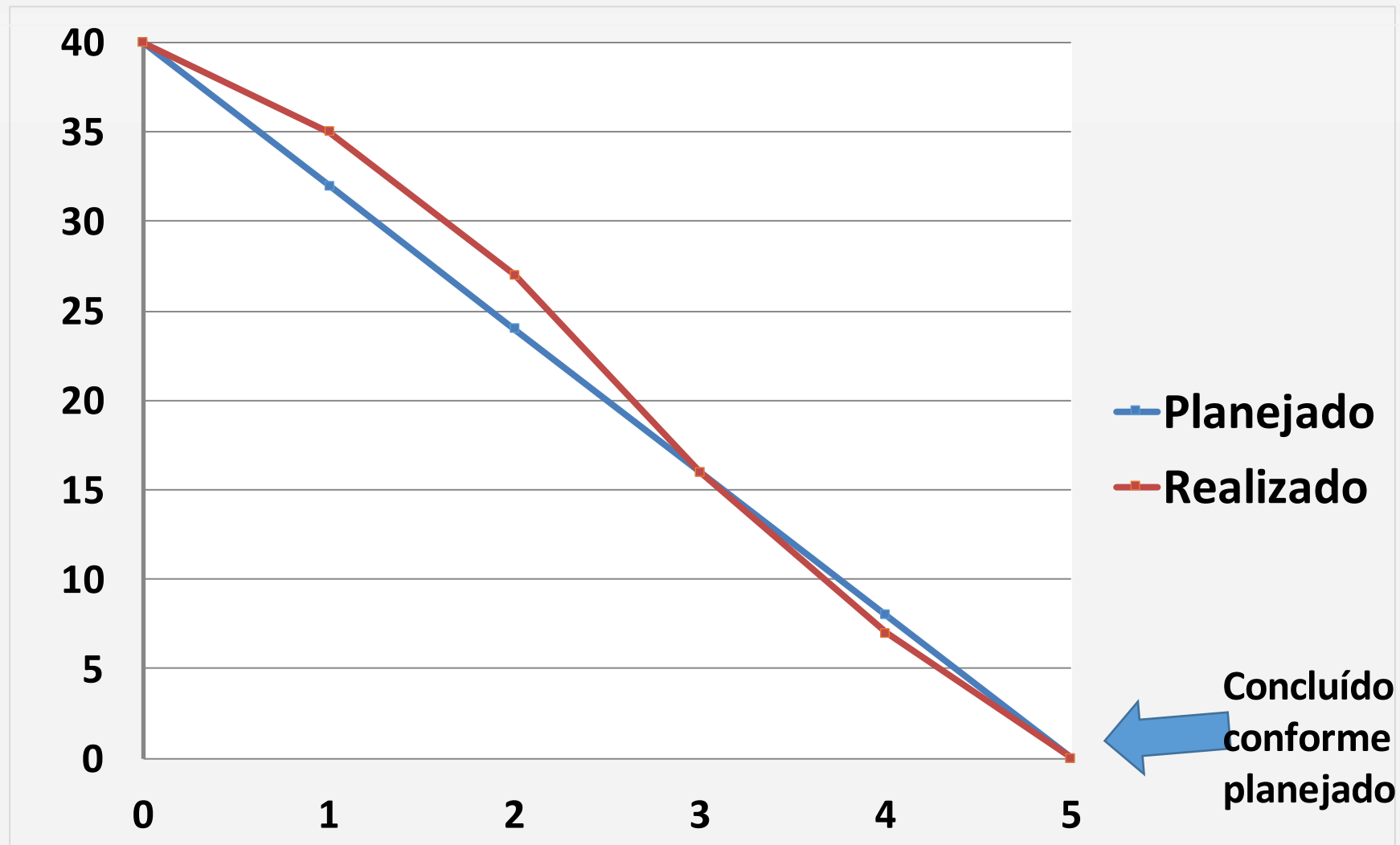
- **Perguntas (QUESTIONS):**

- Qual a nossa produtividade atual?
- Qual o nosso erro de estimativa das tarefas em relação ao tempo previsto e o tempo realizado?
- Qual a variação de produtividade no decorrer na semana de trabalho?

- **Métricas (METRICS):**

- Gráfico Burndown
- Taxa de acerto das estimativas das tarefas

## Gráfico Burndown – Dia 5



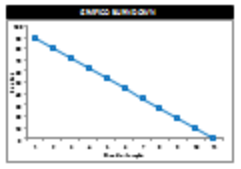
Total de horas da sprint: 40

Pontos previstos no 5º dia: 8 → acumulado previsto:  $8-8=0$

Soma do trabalho realizado no 5º dia: 7

Ponto do gráfico burndown no 5º dia:  $7-7=0$

# Nome da métrica: BURNDOWN DA ITERERAÇÃO

Categoria	Processo
Objetivo(s) de medição	Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos. Visualizar o progresso do projeto.
Equação	<div>A imagem mostra um gráfico de Burndown da Iteração. O eixo horizontal representa o tempo em dias (de 1 a 10), e o eixo vertical representa o número de pontos restantes (de 0 a 10). Uma linha azul desce linearmente de 10 pontos no dia 1 até 0 pontos no dia 10, representando o plano ideal de conclusão das tarefas. Pontos azuis representam o progresso real, que segue a linha ideal.</div> <p>(a) Gráfico com linha estimativa de pontos/dia</p> <p>Onde:</p> <p>Número de tarefas = Número de pontos realizados na tarefa k do dia d Total de pontos realizados no dia d</p> <p>Onde:</p> <p>Ponto da linha de desempenho real no gráfico <i>burndown</i> que apresenta os pontos do dia d na iteração i</p>
Comportamento da métrica	Alvo-é-melhor
Fonte dos dados	Quadro de tarefas



## Nome da métrica: TAXA DE ACERTO NA ESTIMATIVA DAS TAREFAS

<b>Categoria</b>	Processo
<b>Objetivo(s) de medição</b>	Identificar distorções entre o tempo de duração previsto para a execução das tarefas e o tempo de duração realizado. Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos.
<b>Equação</b>	Onde: Taxa de acerto na estimativa das tarefas na iteração $i$ Número de tarefas desenvolvidas no prazo estimado na iteração $i$ Número de tarefas da iteração $i$
<b>Comportamento da métrica</b>	Maior-é-melhor
<b>Fonte dos dados</b>	Quadro de tarefas

## Exercício: Utilizando a abordagem GQM, definir as questões e métricas para os seguintes objetivos

- **Objetivo (Goals):**
  - Melhorar (diminuir) o tempo gasto no processo de correção de defeitos
- Perguntas (Questions):  
?
- Métricas (Metrics):  
?

**OBJETIVO (GOALS):**  
**Melhorar (diminuir) o tempo gasto no processo de correção de defeitos**

**Pergunta 1:** Qual é a velocidade atual de correção de um defeito?

**Métrica 1:**

- Tempo médio de para correção de defeitos
- Fórmula

**(Após realizar ações para melhorar este objetivo)**

**Pergunta 2:** O tempo de correção de defeitos está melhorando?

**Métrica 2:**

- Tempo médio de para correção de defeitos em relação ao tempo estabelecido
- Fórmula =  $(\text{Tempo médio ciclo atual} / \text{Tempo médio ciclodesejado}) * 100$ 
  - ex1:  $(5/4)*100 = 125\%$ , ou seja, acima do esperado (resultado ruim)
  - ex2:  $(4/4)*100=100\%$ , ou seja, está dentro do esperado (resultado bom)
  - ex3:  $(3/4)*100=75\%$ , ou seja, abaixo do esperado (resultado muito bom)

**Pergunta 3:** Qual o % de casos acima do limite máximo definido como meta?

**Métrica 3:**

- Fórmula =  $(\text{Número defeitos acima tempo} / \text{Número total defeitos corrigidos}) / 100$ 
  - Ex1:  $(1/10)*100 = 10\%$  acima do tempo

**Será que é fácil implantar e  
utilizar métricas na área de desenvolvimento de  
software?**

# Principais Barreiras

---

- Medir custa caro
- Os maiores benefícios vêm a longo prazo
- Má utilização das métricas
- Grande mudança cultural necessária
- Dificuldade de estabelecer medições apropriadas e úteis
- Interpretações dos dados realizadas de forma incorreta
- Obter o comprometimento de todos os envolvidos e impactados
- Estabelecer inicialmente um programa de medições é fácil, o difícil é manter a longo prazo

# Propriedades desejáveis de uma métrica

---

- Facilmente calculada, entendida e testada
- Passível de estudos estatísticos
- Expressa em alguma unidade
- Obtida o mais cedo possível no ciclo de vida do software
- Assegurar uma estratégia de melhoria

**Quais seriam bons motivos para se utilizar métricas em empresas de desenvolvimento de software?**

# Porque utilizar métricas de software

---

- Entender e aperfeiçoar o processo de desenvolvimento
- Melhorar a gerência de projetos e o relacionamento com clientes
- Gerenciar contratos de software
- Indicar a qualidade de um produto de software
- Avaliar a produtividade das equipes
- Avaliar retorno de investimento
- Oferecer dados quantitativos e qualitativos para a equipe e aos gerentes de projetos, para que possam traçar estratégias para melhorar o desempenho