

Métricas de software / Estimativas

Engenharia de Software I

Introdução

- 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.

Como controlar estatisticamente o desempenho do projeto

Medição nas áreas de engenharia



Medição nas áreas de engenharia:

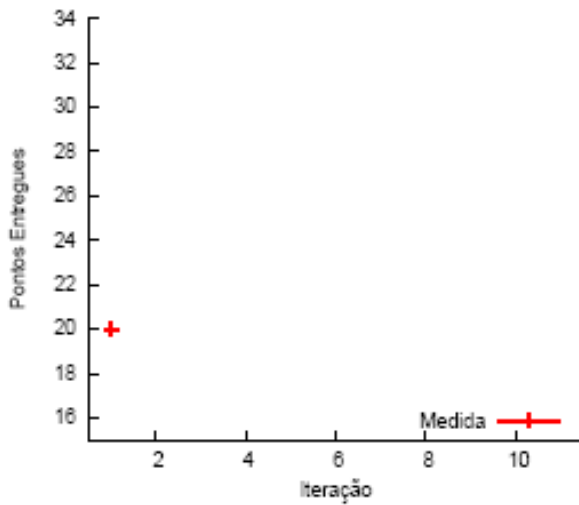
- A medição é algo comum no mundo da engenharia, existindo vários tipos de grandezas para medição:
 - peso, temperatura, dimensões, tensão, entre outras.

Medição na área de engenharia de software

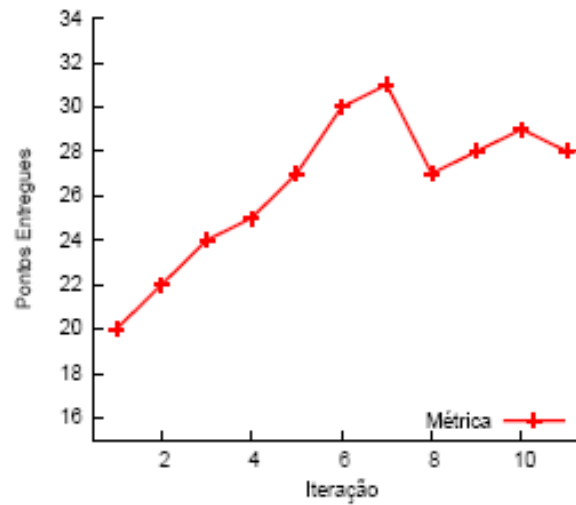


- Infelizmente a engenharia de software está longe de ter uma medição padrão amplamente aceita e sem nenhum fator subjetivo.
- Temos **dificuldade em concordar sobre o que medir e como avaliar o resultado** das medições obtidas.
- **Métricas de softwares** possibilitam realizar uma das atividades mais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

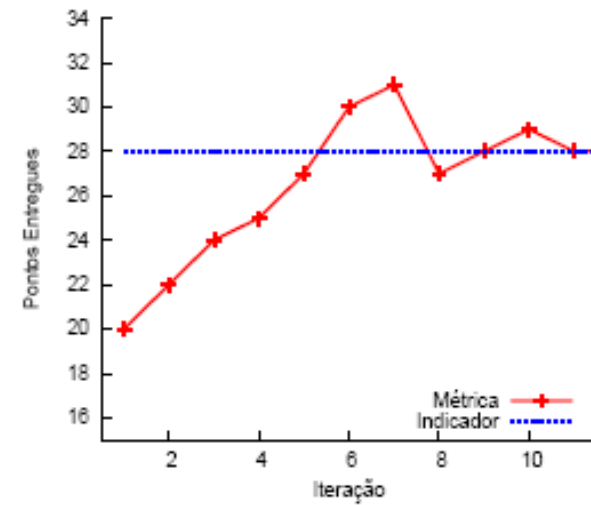
Conceitos iniciais: Medida, métrica, medição e indicador



(a) Medida



(b) Métrica



(c) Indicador

Conceitos iniciais

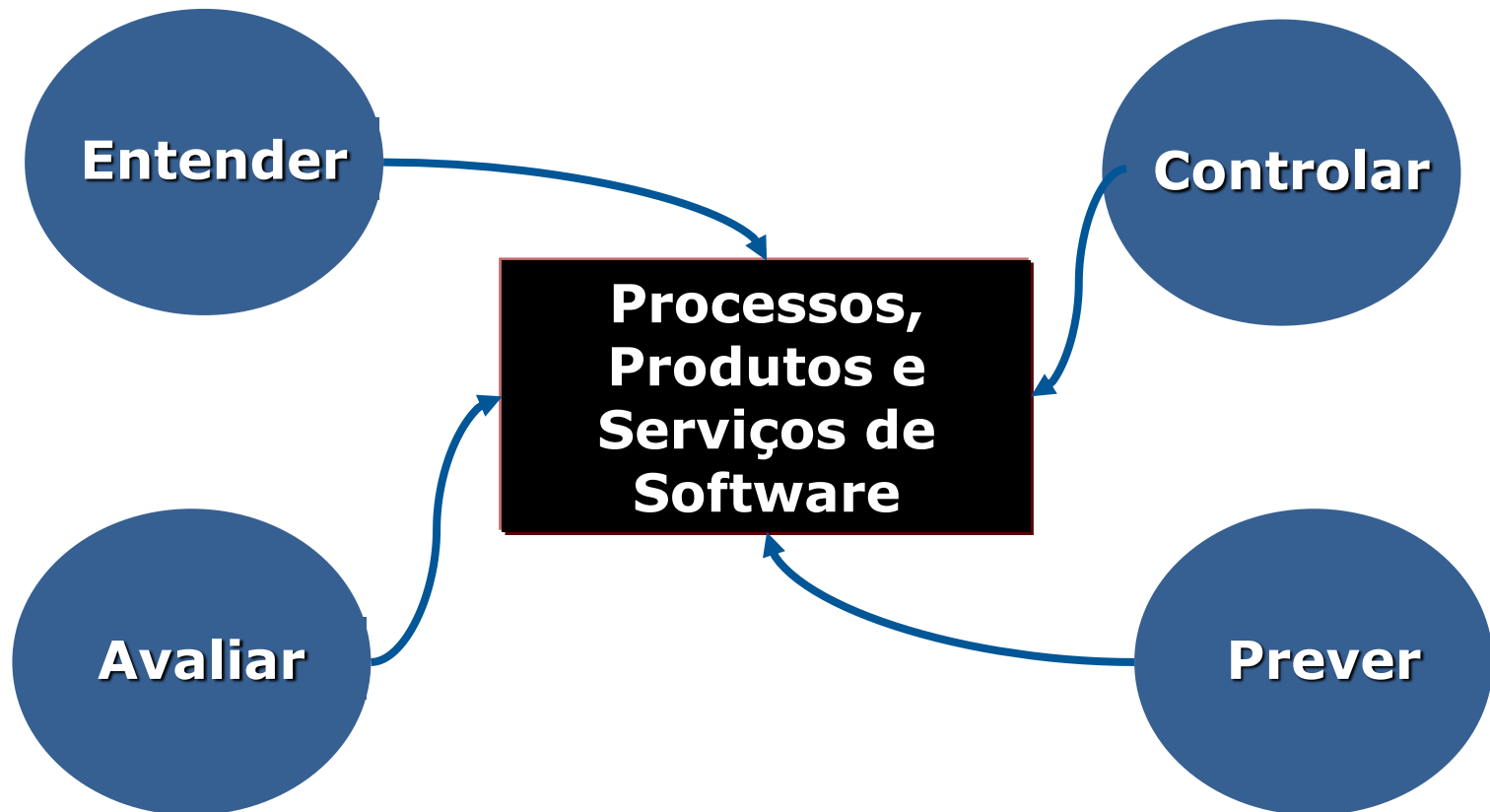
- **Medida:** fornece uma indicação quantitativa da extensão, quantidade, dimensão, capacidade ou tamanho de algum atributo de um produto ou processo.
- **Medição:** ato de determinação de uma medida.
- **Métrica:** medida quantitativa do grau em que um sistema se encontra em relação a um determinado 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?
- Onde estão os nossos principais problemas?
- 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?
- É possível cumprir o prazo, garantindo que o tempo que foi prometido?
- Qual a produtividade da atividade do processo?
- Qual a produtividade da equipe?
- Qual é a qualidade do código gerado?
- Como medir a qualidade do produto?
- Onde estão os nossos principais problemas?
- 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

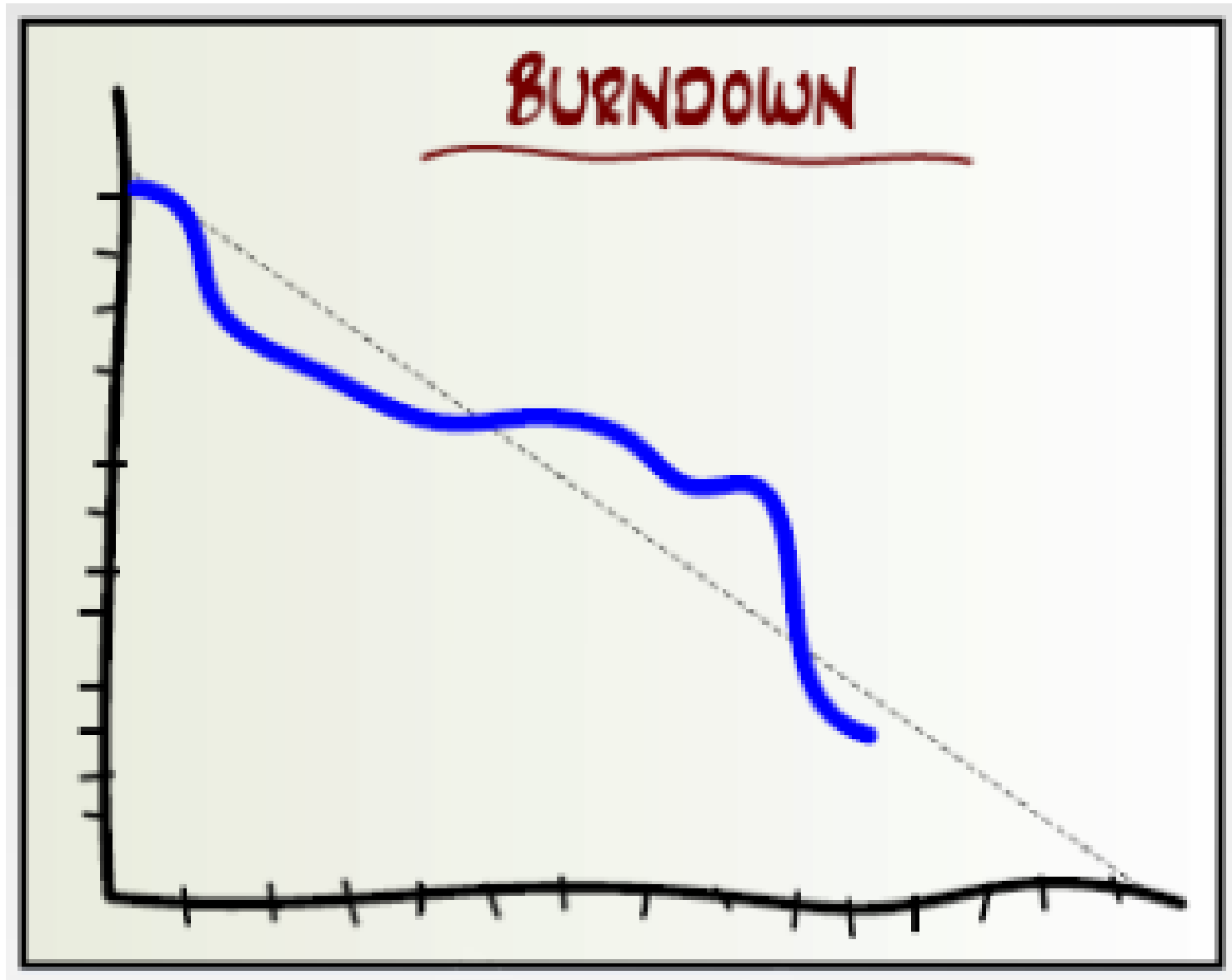


Gráfico Burndown da Sprint

- 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

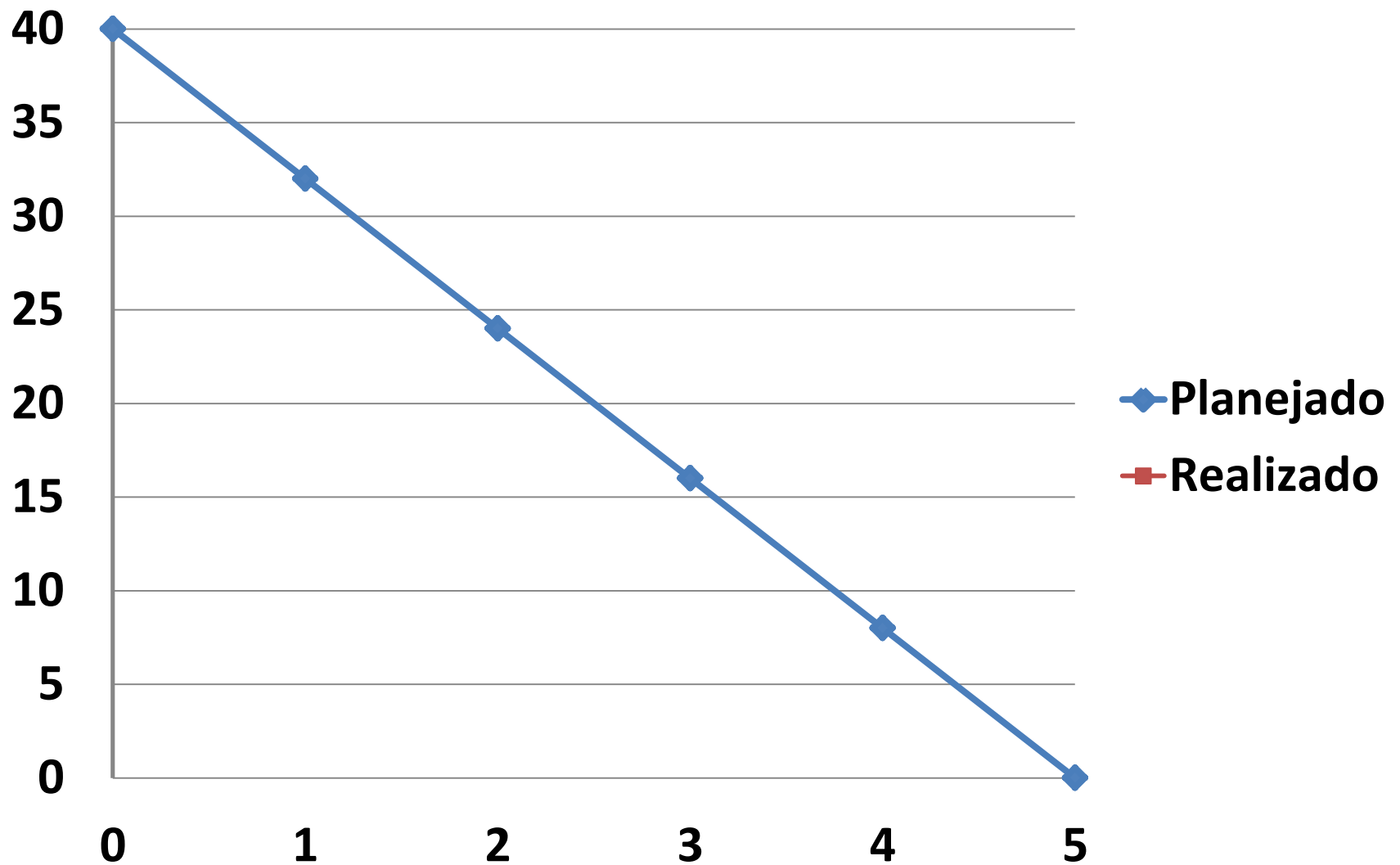
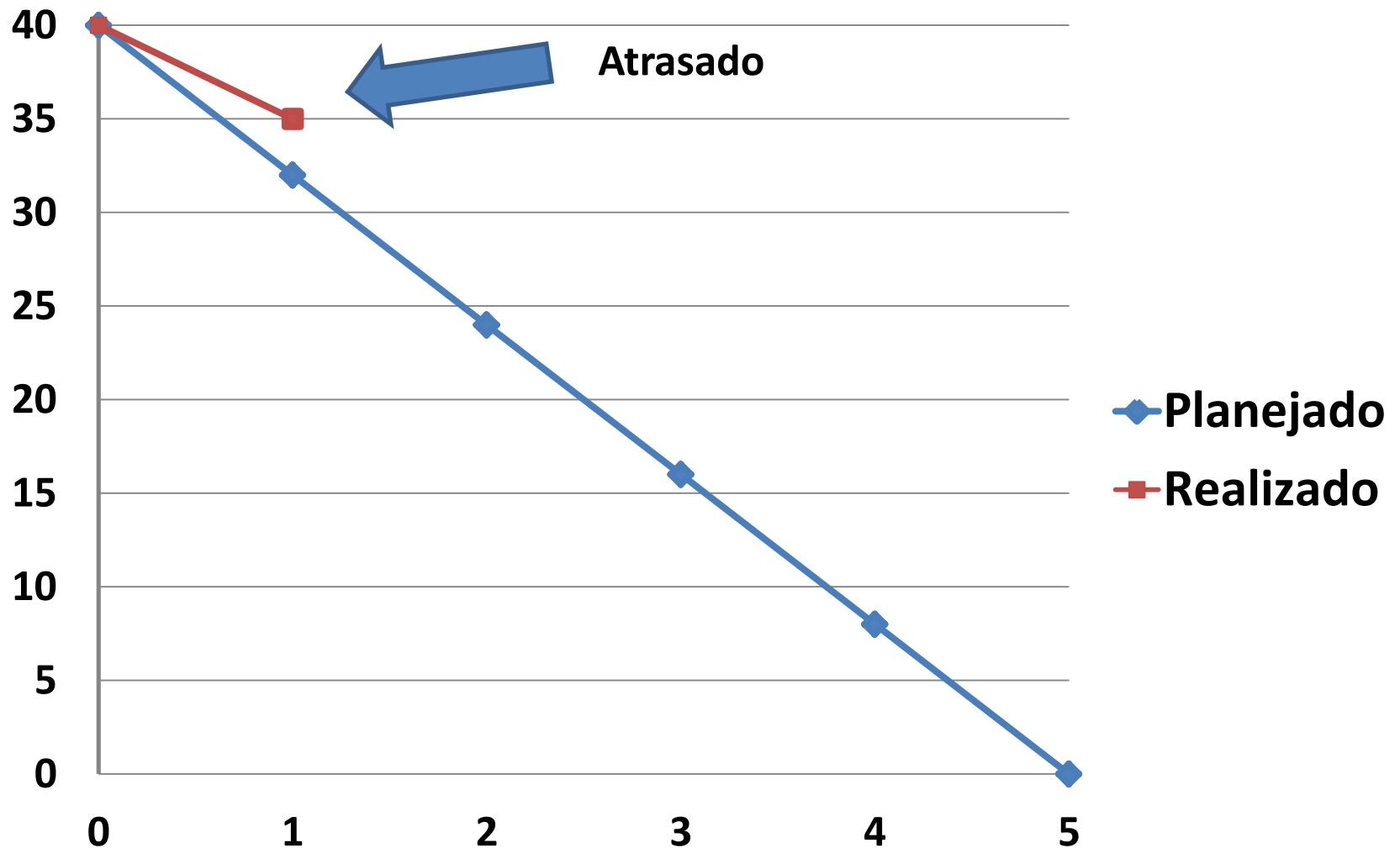


Gráfico Burndown - Dia 1



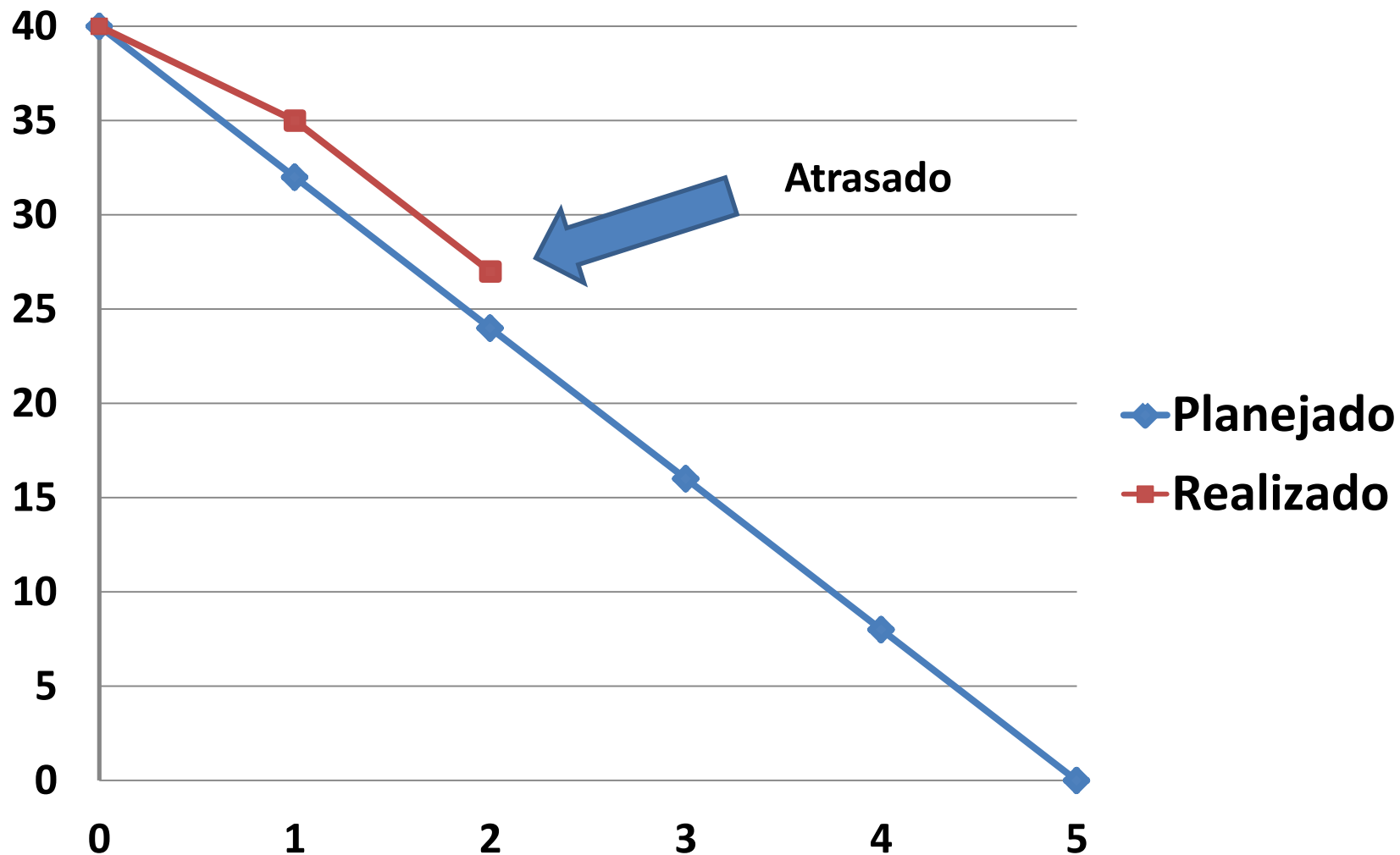
Total de horas da sprint: 40

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



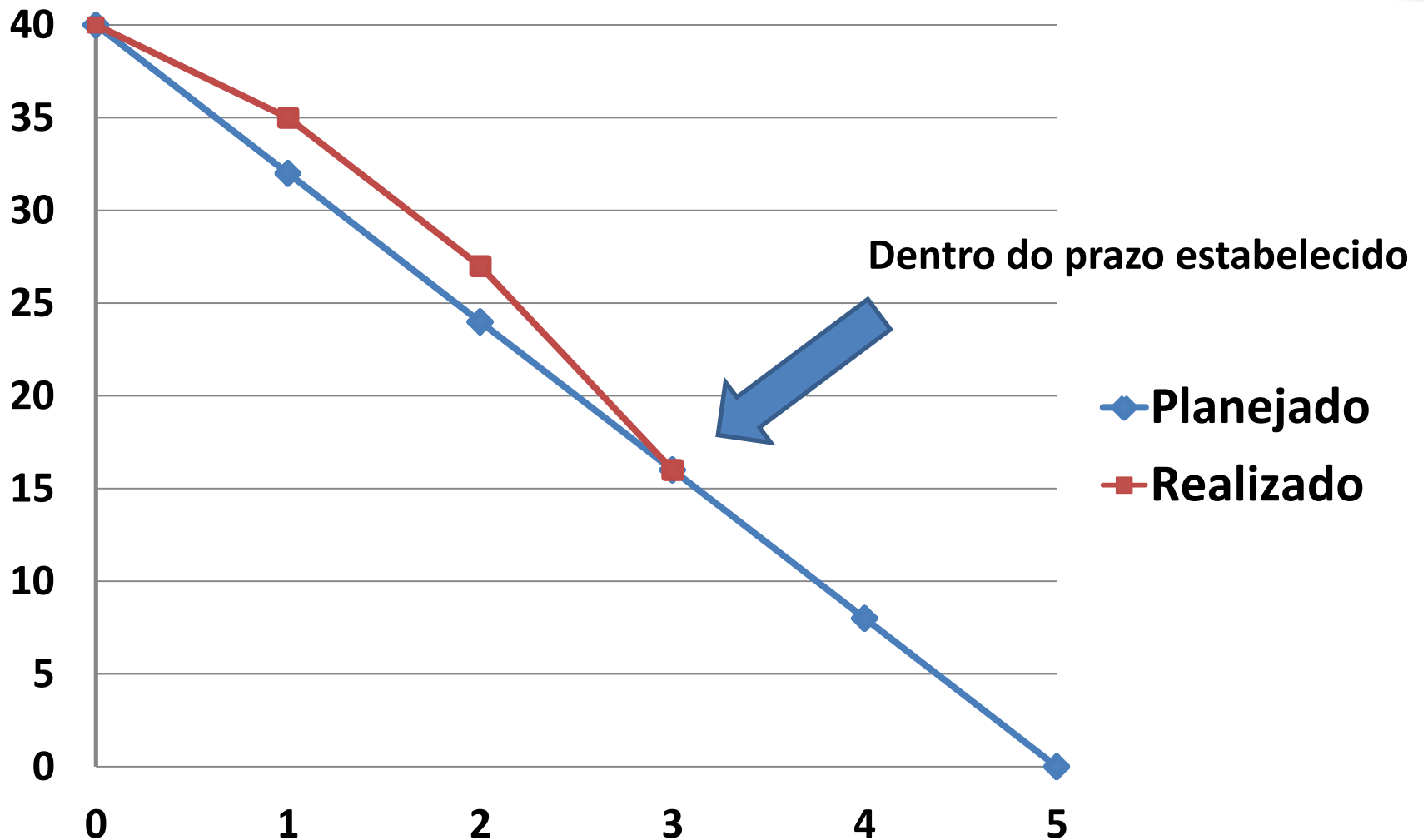
Total de horas da sprint: 40

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



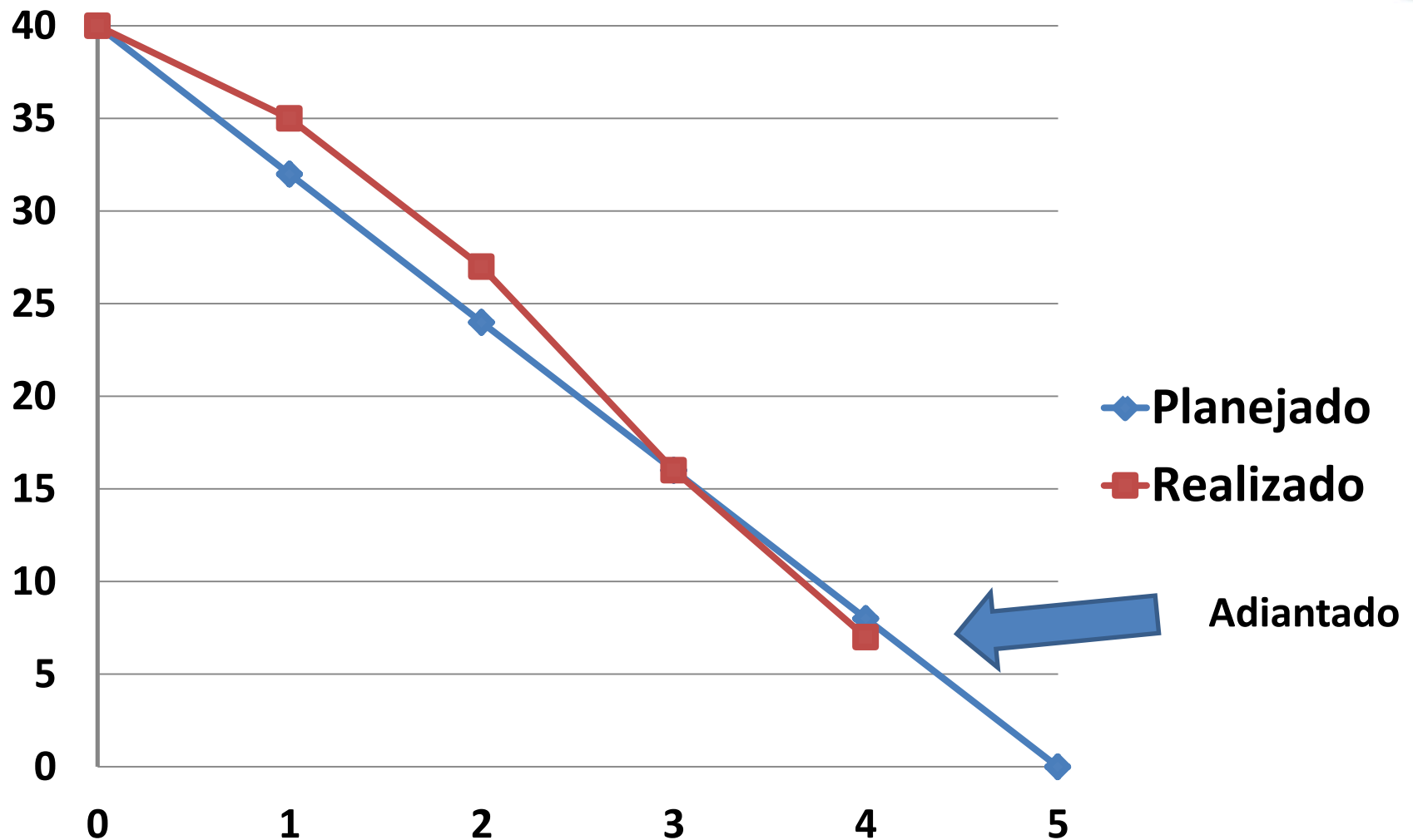
Total de horas da sprint: 40

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



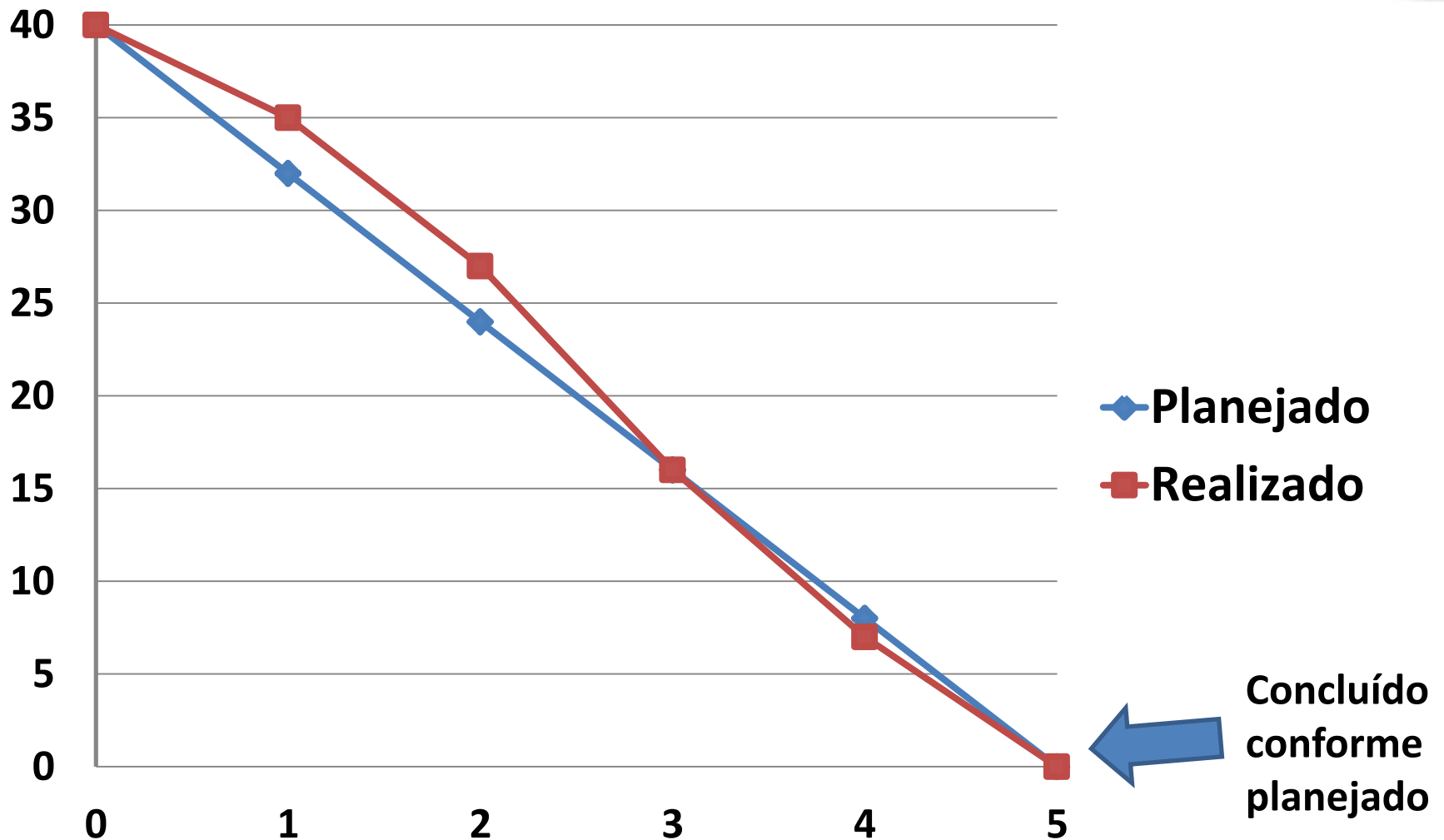
Total de horas da sprint: 40

Pontos previstos no 4º dia: 8 → acumulado previsto: $40 - 8 = 32$

Soma do trabalho realizado no 4º dia: 9

Ponto do gráfico burndown no 4º dia: $32 - 9 = 23$

Gráfico Burndown - Dia 5



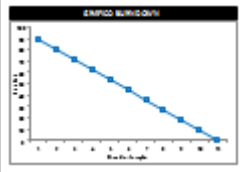
Total de horas da sprint: 40

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></div> <p>(a) Gráfico com linha estimativa de pontos/dia</p> $(b) TT_PTS_R_d = \sum_{k=1}^n PTS_R_{d,k}$ <p>Onde:</p> <p>n = Número de tarefas</p> <p>$PTS_R_{d,k}$ = Número de pontos realizados na tarefa k do dia d</p> <p>$TT_PTS_R_d$ = Total de pontos realizados no dia d</p> $(c) Bw_R_{i,d} = (TT_PTS_R_{i,d-1} - TT_PTS_R_{i,d})$ <p>Onde:</p> <p>$Bw_R_{i,d}$ = 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

Exemplo:
Grau de atendimento aos requisitos
funcionais

Nome da métrica: Grau de atendimento aos requisitos funcionais

Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto após sua homologação
Equação	$G_{RF_i} = \frac{RF_{ac_i}}{TT_{RF_i}} * 100$ <p>Onde: G_{RF_i} = Grau de atendimento aos requisitos funcionais na iteração i TT_{RF_i} = Número total de requisitos funcionais (histórias) da iteração i RF_{ac_i} = Número de requisitos funcionais (histórias) aceitos pelo cliente na iteração i</p>
Comportamento da métrica	Maior-é-melhor
Fonte dos dados	Requisitos entregues ao cliente

Classificação das métricas

Ponto de vista da medição

- As métricas de software, do ponto de vista de medição, podem ser divididas em duas categorias:
 - Diretas
 - Indiretas

Métricas diretas

- Podemos considerar como de **métricas diretas**:
 - Gráfico burndown
 - Grau de atendimento aos requisitos funcionais
 - Custo do projeto;
 - Esforço (soma do tempo) aplicado para o desenvolvimento ou manutenção;
 - Quantidade de linhas de código produzidas (LOC);
 - Velocidade de execução (número de requisitos entregues num período de tempo);
 - Número de requisitos entregue/ soma do número de defeitos;
 - Entre outras métricas mensuráveis.

Métricas indiretas

- Porém existem critérios difíceis de serem avaliados, e só é possível medir de **forma indireta**, como por exemplo:
 - Qualidade;
 - Complexidade;
 - Eficiência;
 - Confiabilidade;
 - Entre outros.

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

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?

Paradigma GQM - Goals Question Metrics

GQM

- 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

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.

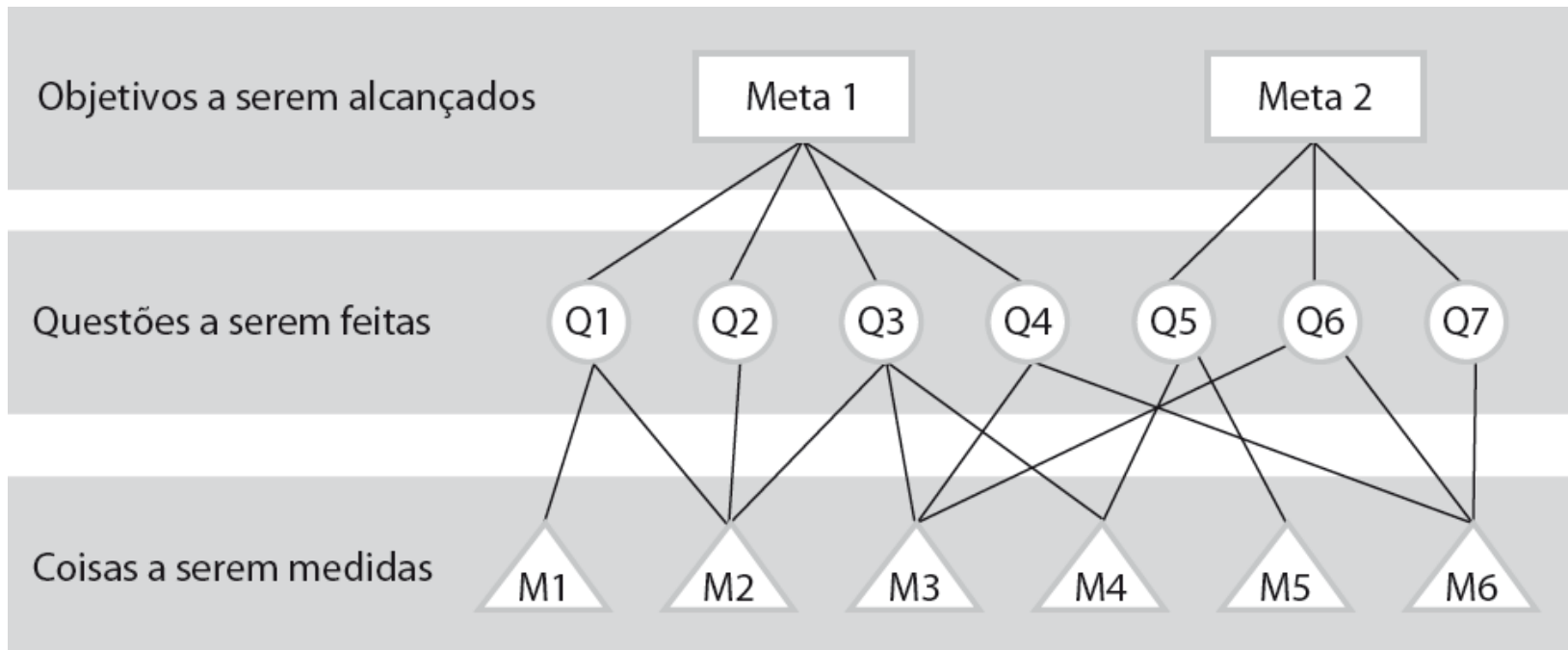
QUESTIONS (questões, perguntas)

- 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

Abordagem GQM



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

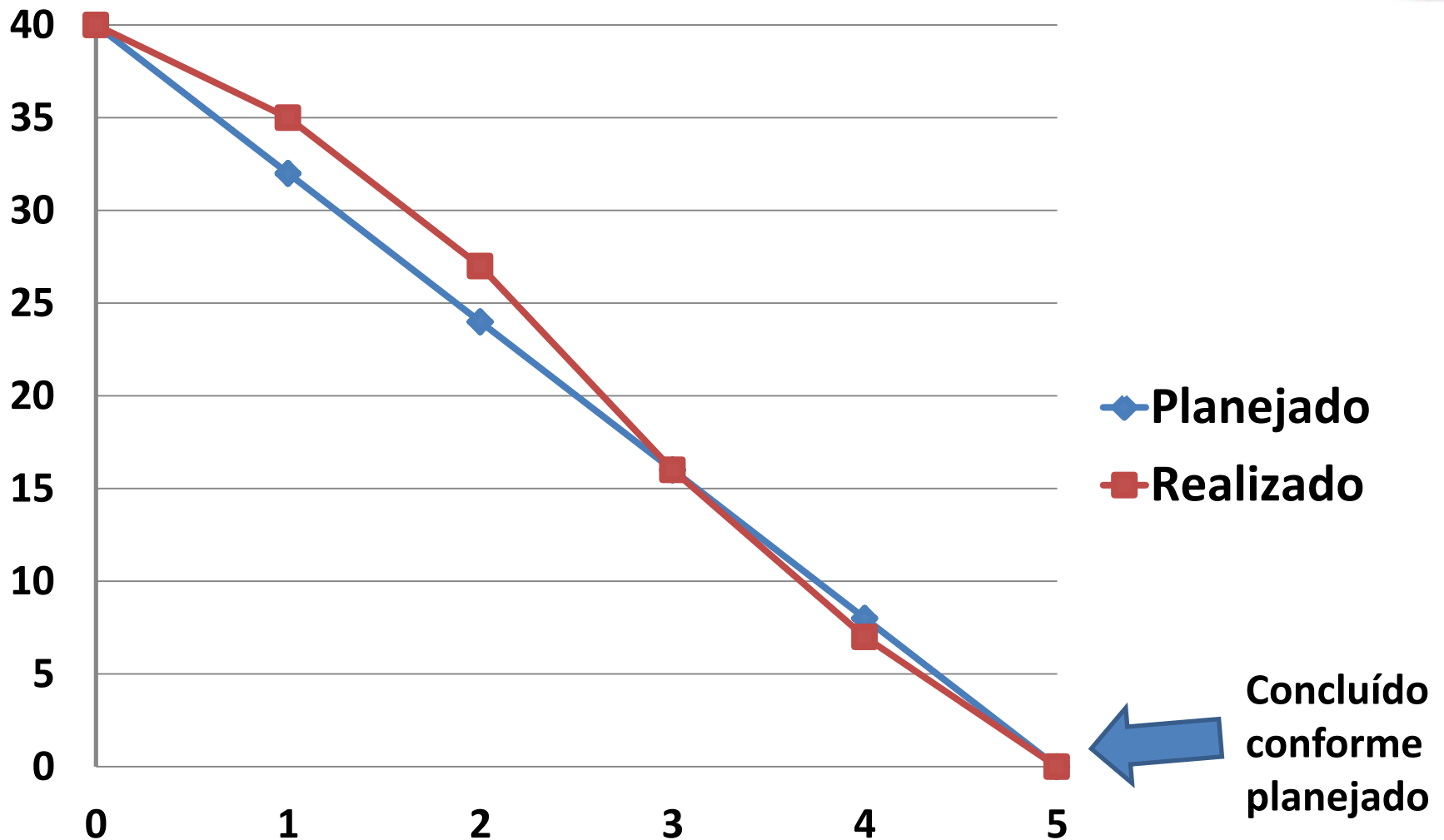
- **Perguntas (QUESTIONS):**

- 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



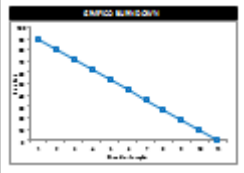
Total de horas da sprint: 40

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></div> <p>(a) Gráfico com linha estimativa de pontos/dia</p> <p>(b) $TT_PTS_R_d = \sum_{k=1}^n PTS_R_{d,k}$</p> <p>Onde:</p> <p>$n$ = Número de tarefas</p> <p>$PTS_R_{d,k}$ = Número de pontos realizados na tarefa k do dia d</p> <p>$TT_PTS_R_d$ = Total de pontos realizados no dia d</p> <p>(c) $Bw_R_{i,d} = (TT_PTS_R_{i,d-1} - TT_PTS_R_{i,d})$</p> <p>Onde:</p> <p>$Bw_R_{i,d}$ = 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

Categoria	Processo
Objetivo(s) de medição	<p>Identificar distorções entre o tempo de duração previsto para a execução das tarefas e o tempo de duração realizado.</p> <p>Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos.</p>
Equação	$Tx_{TS_i} = \frac{TS_{prazo_i}}{TS_i} * 100$ <p>Onde:</p> <p>Tx_{TS_i} = Taxa de acerto na estimativa das tarefas na iteração i</p> <p>TS_{prazo_i} = Número de tarefas desenvolvidas no prazo estimado na iteração i</p> <p>TS_i = Número de tarefas da iteração i</p>
Comportamento da métrica	Maior-é-melhor
Fonte dos dados	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):
 - ?

Solução do exercício

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 =
$$\frac{\sum_{Df=1}^n \text{Tempo de correção do defeito}_{Df}}{\text{Número de defeitos}}$$

(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 = (Tempo médio ciclo atual/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 = (Número defeitos acima tempo / 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

Exercício: Utilizar a abordagem GQM para criar métricas para avaliação dos testes

- 1º) Utilizando o conteúdo anterior sobre testes defina:
 - Objetivo(s) organizacional(ais) relevante(s)
 - Perguntas que revelam informações para formulação das métricas
 - A definição da(s) métrica(s) para possam auxiliar para alcançar o(s) objetivo(s) organizacional(is)
- 2º) Executar os casos de testes que você elaborou na aula anterior
- 3º) Mostrar o valor obtido da(s) métricas) após a execução dos testes