

Introdução

Por que medir software?

- Melhorar a gerência de projetos e o relacionamento com clientes;
- Reduzir frustrações e pressões de cronograma;
- Gerenciar contratos de software;

Medida

- É a quantificação de dados em um padrão e qualidade aceitáveis (exatidão, completude, consistência, temporalidade).
- É a avaliação de uma grandeza por meio da comparação com outra grandeza da mesma espécie tomada como unidade

Métrica

- Pode ser entendida como a relação entre duas medidas de grandezas iguais ou diferentes

Quanto a aplicação

1. **Métricas de processo:** Elas são usadas para medir a eficiência e eficácia de processos.
2. **Métricas de projeto:** Elas são usadas para quantificar defeitos, custos, cronograma, produtividade e estimativa para vários recursos e entregas de projeto.

Quanto a orientação

1. **Métricas orientadas ao tamanho:** São medidas diretas do software e do processo por meio do qual ele é desenvolvido. Alguns exemplos são: Linhas de código
2. **Métricas orientadas à função:** São medidas indiretas do software e do processo por meio do qual ele é desenvolvido. Concentram-se na funcionalidade ou utilidade do software. Alguns exemplos são: PF (Pontos por Função) e SOD (Speed of Delivery).

Indicador

- É uma representação de forma simples ou intuitiva de uma métrica ou medida para facilitar sua interpretação quando comparada a uma referência ou alvo.
 - Representam informações a partir das quais é possível avaliar uma situação e sua evolução histórica
1. **Indicadores de resultado (lagging indicators):** são utilizados para medir resultados passados. Monitoram o efeito e não permitem mais alterar um dado resultado.
 2. **Indicadores de desempenho (leading indicators):** são utilizados para indicar tendências futuras. Monitoram a causa antes do efeito e caracterizam-se pela possibilidade de alterar o curso para o alcance de um resultado.

Normalmente são especificados através de um conjunto de informações, como por exemplo:

1. Título do indicador
2. Sigla
3. Unidade de medida
4. Fonte dos dados
5. Período de coleta
6. Regra de cálculo
7. Forma de coleta
8. Meta (valores desejados)
9. Forma de interpretação

Exercícios

- 1) Qual a importância de medir software?
- 2) O que é uma medida?
- 3) O que é uma métrica?
- 4) Qual a diferença entre métricas de processo e de projeto?
- 5) O que é um indicador?
- 6) Qual a diferença entre indicadores de desempenho e de resultado?

Métricas em Projetos Tradicionais

Modelo Tradicional

- Também chamado de ciclo de vida clássico ou Cascata (Waterfall).
- É utilizado quando os requisitos são razoavelmente bem compreendidos, ou seja, o trabalho flui da comunicação até a implantação de um modelo razoavelmente linear

Etapas:

1. **Comunicação:** iniciação do projeto e levantamento de requisitos.
2. **Planejamento:** Estimativas e cronograma.
3. **Modelagem:** Análise e projeto do sistema.
4. **Construção:** Codificação e teste.
5. **Implantação:** Entrega, manutenção e feedback.

Tipos de métricas:

1. **Métricas de qualidade:** (Ex.: % de testes sem erros por mês)
2. **Métricas de desempenho:** (Ex.: % de andamento do cronograma)
3. **Métricas orientadas à função:** (Ex.: Pontos de função de um sistema)

Métricas de desempenho

- **EVM:** É o resultado da Análise de Valor Agregado de um projeto (**EVA**).
- **EVA:** É um método que permite ao gerente de projeto medir a quantidade de trabalho realmente executada em um projeto e pode ser utilizado na revisão do custo e prazo do cronograma.

O EVA é baseado em 3 medidas principais:

1. **PV:** É o planejamento de orçamento gasto em uma tarefa até a data atual.
2. **EV:** É o que foi produzido até a data atual.
3. **AC:** É o que foi gasto de orçamento até a data atual.

As medidas PV, EV e AC combinadas geram as seguintes métricas:

1. **CV:** É a diferença entre o que foi produzido(**EV**) e o custo real(**AC**), se for maior que zero, o custo do projeto está abaixo do orçamento.
2. **CPI:** É a variação entre o que foi produzido(**EV**) e o custo real(**AC**), se for maior que um, o custo do projeto está abaixo do orçamento.
3. **SV:** É a diferença entre o que foi produzido(**EV**) e o custo planejado(**PV**), se for maior que zero, o cronograma do projeto está adiantado.
4. **SPI:** É a variação entre o que foi produzido(**EV**) e o custo planejado(**PV**), se for maior que um, o cronograma do projeto está adiantado.

Exercícios:

- 1) Quais são as etapas do modelo tradicional de desenvolvimento de sistemas?
- 2) Quais são os principais tipos de métricas associadas a projetos?
- 3) O que é EVM?
- 4) O que é EVA?
- 5) Quais são as 3 medidas principais do EVA?

Métricas em Projetos Ágeis

Seguem algumas ferramentas para medição do progresso ou tempo das entregas:

1. Burndown
2. Burnup
3. Velocity Chart
4. Cumulative flows (CDF)
5. Lead time
6. Lead time breakdown
7. Throughput

- **Burndown:** Apresenta dois eixos, onde o eixo horizontal X representa o tempo de uma sprint, e o eixo vertical Y representa o esforço em pontos de história (Story Points) necessário para completar a sprint. A redução das Story Points concluídas são representadas ao longo do tempo
- **Burnup:** Apresenta dois eixos, onde o eixo horizontal X representa o tempo de uma sprint, e o eixo vertical Y representa o esforço em pontos de história (Story Points) necessário para completar a sprint. O acúmulo das Story Points concluídas são representadas ao longo do tempo
- **Velocity Chart:** Apresenta a quantidade de pontos de história (Story Points) concluídos em cada Sprint. A velocidade da equipe é calculada pela média das Story Points concluídas em um determinado período, e pode ser utilizada para prever a quantidade de Story Points que poderão ser concluídas em uma nova Sprint.
- **Cumulative flows (CDF):** Apresenta dois eixos, onde o eixo horizontal representa a unidade de tempo (ex: dias, semanas, sprints, etc.), e o eixo vertical representa o número de itens (ex: total de histórias de usuário) a serem completados. As áreas pintadas representam a evolução das etapas do processo ao longo do tempo.
- **Lead time:** O Lead Time é o Tempo que um item demora para percorrer todo o fluxo até chegar no estado de concluído. O gráfico apresenta no eixo horizontal as atividades concluídas (Ex: User Stories) e o eixo vertical apresenta a quantidade de tempo (ex: dias) que levou para concluí-las. O gráfico demonstra o ritmo de trabalho da equipe.
- **Lead time breakdown:** No eixo horizontal apresenta as atividades concluídas (Ex: User Stories) e o eixo vertical apresenta a quantidade de tempo (ex: dias) que levou para concluí-las. O gráfico também detalha para cada atividade a proporção de tempo gasto em cada uma das etapas de construção. É útil para identificar quais são as principais razões para determinada atividade estar demorando para ser entregue.
- **Throughput:** Apresenta no eixo horizontal a unidade de tempo (Ex: Semanas) e o eixo vertical apresenta a quantidade de itens (Ex: User Stories) concluídos na unidade de tempo.

Análise de Pontos de Função

Objetivos:

1. Medir a funcionalidade que o usuário solicita e recebe.
2. Medir o desenvolvimento e a manutenção de software de forma independente da tecnologia utilizada para sua implementação.

Padronização:

- O processo de medição de pontos de função é padronizado pela **IFPUG**.
- O IFPUG é representado no Brasil pelo BFPUG.
- Outro padrão de medição de pontos de função é o NESMA.

Processo de medição:

- Deve ser simples o suficiente para minimizar o trabalho adicional envolvido no processo de medição.
- Deve ser uma medida consistente entre vários projetos e organizações, ou seja, pessoas diferentes medindo o mesmo projeto devem encontrar resultados similares.

Usuário:

- o conceito de usuário não está restrito apenas à pessoa física que usa o software. Usuário é qualquer pessoa ou coisa que interaja (envia ou recebe dados) com a aplicação.

Benefícios:

- Um meio de estimar custo e recursos para o desenvolvimento e manutenção de software.
- Uma ferramenta para fundamentar a negociação de contratos.
- Apoiar os gerentes de projeto no gerenciamento de escopo de projetos.

Processo da contagem de pontos de função

- **Reunir a documentação disponível, a documentação ideal deve:**
 1. Descrever a funcionalidade entregue pelo software;
 2. Descrever a funcionalidade que é impactada pelo projeto de software medido.
- **Exemplos:**
 1. Modelos de dados/objetos;
 2. Diagramas de classe;
 3. Diagramas de fluxo de dados;

4. Casos de uso;
 5. Descrições procedurais;
 6. Layout de relatório e telas;
 7. Manuais de usuário.
- **Identificar o propósito da contagem, pode ser dividida em:**
 1. **Determinar o tipo de contagem, divididos em 3 tipos:**
 - **Contagem de um projeto de desenvolvimento:** Mede a funcionalidade fornecida aos usuários finais do software quando da sua primeira instalação, além de possíveis migrações de dados.
 - **Contagem de um projeto de melhoria:** Mede as funções adicionadas, modificadas ou excluídas do sistema pelo projeto de melhoria, além de possíveis migrações de dados.
 - **Contagem de uma aplicação:** Mede a funcionalidade fornecida aos usuários por uma aplicação instalada
 2. **Determinar o escopo da contagem**
 - O escopo define quais funções serão incluídas na contagem, se ela abrangerá um ou mais sistemas ou apenas parte de um sistema. Assim, o escopo pode abranger:
 - Todas as funcionalidades disponíveis;
 - Apenas as funcionalidades efetivamente utilizadas pelo usuário;
 - Apenas algumas funcionalidades específicas (relatório, transações, etc.).
 3. **Determinar a fronteira da aplicação**
 - A fronteira da aplicação é a interface conceitual que delimita o software que será medido e o mundo exterior (seus usuários).
 - A fronteira deve ser determinada com base no ponto de vista do usuário.
 - A fronteira deve ser baseada na separação das funções conforme estabelecido pelos processos de negócio
 - Em projetos de melhoria, a fronteira estabelecida no início do projeto deve estar de acordo com a fronteira já estabelecida para a aplicação sendo modificada.

Medir funções de dados

- As funções do tipo dado representam as funcionalidades fornecidas pelo sistema ao usuário para atender a suas necessidades de armazenamento de dados. São classificadas em:
 - **Arquivo Lógico Interno (ALI):** É um grupo logicamente relacionado de dados, reconhecido pelo usuário, mantido dentro da fronteira da aplicação sendo contada. Sua principal intenção é armazenar dados mantidos através de uma ou mais transações da aplicação sendo contada. Ex: dados de entrada e saída de um trabalhador.
 - **Arquivo de Interface Externa (AIE):** É um grupo logicamente relacionado de dados, reconhecido pelo usuário, mantidos fora da fronteira da aplicação sendo contada. Sua principal intenção é armazenar dados referenciados através de uma ou mais transações da aplicação sendo contada. Ex: dados do trabalhador contido no sistema de controle de ponto.

Medir funções de Transação

- As funções do tipo transação representam os requisitos de processamento fornecidos pelo sistema ao usuário. São classificadas em:
 - **Entrada Externa (EE):** É uma transação que processa dados ou informações de controle originados de fora da fronteira da aplicação. Ex: incluir cliente, alterar cliente, excluir cliente.
 - **Consulta Externa (CE):** É uma transação que envia dados ou informações de controle para fora da fronteira da aplicação. Sem realização de cálculos ou manipulação nos dados.
 - **Saída Externa (SE):** É uma transação que apresenta dado ao usuário ou envia dados ou informações de controle para fora da fronteira da aplicação. Com realização de cálculo ou manipulação de dados.

Calcular o tamanho funcional

- A complexidade das funções pode ser baixa, média ou alta e do tipo de dados (**ALI ou AIE**) é determinada pela quantidade de dois parâmetros:
 - **Tipos de Dados (TD):** campo único, reconhecido pelo usuário e não repetido, seria cada atributo de uma tabela.
 - **Tipos de Registros (TR):** subgrupos de dados, reconhecido pelo usuário, componente de um ALI ou AIE. a tabela reconhecida pelo usuário e as demais se tornam tipos de registro.

- A complexidade das funções do tipo transação (**EE, SE e CE**) é determinada pela quantidade de dois parâmetros:
 - **Tipos de Dados (TD)**: campo único, reconhecido pelo usuário e não repetido, seria cada campo da tela, grupo de botões, mensagens.
 - **Arquivos Referenciados (AR)**: Quantidade de ALI que a transação se comunica

Fluxo para cálculo de dados e transações

- **Contagem detalhada:**
 1. Identificar se é ALI ou AIE / EE, CE, SE.
 2. Identificar a quantidade TD e TR / TD e AR.
 3. Identificar a complexidade baseado na quantidade de TD e TR / TD e AR.
 4. Verificar quantos PF valem a complexidade dos dados e transações.
- **Contagem por estimativa:**
 1. Determinar todas as funções de todos os tipos (ALI, AIE, EE, SE, CE).
 2. Para cada função do tipo dado (ALI, AIE) ter a sua complexidade funcional avaliada como Baixa
 3. cada função transação (EE, SE, CE) deve ser avaliada como complexidade Média.
 4. Verificar quantos PF valem a complexidade dos dados e transações.
- **Contagem indicativa:**
 1. Calcular de acordo com a Fórmula = $35 \times (ALI) + 15 \times (AIE) = PF$

Documentar e reportar o resultado

- O resultado final da contagem deve ser documentado, incluindo:
 1. A documentação utilizada na contagem
 2. O propósito da contagem
 3. O tipo da contagem
 4. O escopo da contagem
 5. A fronteira da aplicação
 6. A data e os participantes da contagem
 7. Identificação das funções de dados e transações com suas respectivas complexidades
 8. O total de pontos de função identificados para cada função de dados e transações