

Introdução a Métricas de Software

Josino Rodrigues Neto

josinon@gmail.com

Objetivos

- Entender porque medição é importante para avaliação e garantia da qualidade de software
- Entender as abordagens principais de métricas e como elas são utilizadas
- Conhecer algumas métricas e suas aplicações
- Entender o que é um Plano de Métricas e como escrever um

Motivação

Um dos objetivos básicos da Engenharia de Software é: *a transformação da criação de sistemas software de uma maneira artística, indisciplinada e pouco entendível para uma forma devidamente **controlada, quantificada e previsível***

“Métricas de Software” é um assunto discutido há mais de 20 anos na engenharia de software ... e no entanto não é verificada sua utilização, na prática, pela grande maioria dos projetos de construção de software

Pesquisas realizadas em empresas de software indicam que mais da metade de grandes projetos de software se deparam com algum tipo de atraso, excesso de custo ou prazo ou algum fracasso na execução quando implantado

Falta de controle dos projetos

Motivação

- “Não se pode gerenciar o que não se pode medir”.

Tom De Marco

- “Se você não sabe para onde você quer ir, qualquer caminho você pode seguir. Se você não sabe onde você está, um mapa não vai ajudar!”.

Roger Pressman

O que são métricas de software?

- Uma métrica é a medição de um atributo (propriedades ou características) de uma determinada entidade (produto, processo ou recursos). Exemplos:
 - Tamanho do produto de software (ex: Número de Linhas de código)
 - Número de pessoas necessárias para implementar um caso de uso
 - Número de defeitos encontrados por fase de desenvolvimento
 - Esforço para a realização de uma tarefa
 - Tempo para a realização de uma tarefa
 - Custo para a realização de uma tarefa
 - Grau de satisfação do cliente (ex: adequação do produto ao propósito, conformidade do produto com a especificação)

Por que medir software?

- Entender e aperfeiçoar o processo de desenvolvimento
- Melhorar a gerência de projetos e o relacionamento com clientes
- Reduzir frustrações e pressões de cronograma
- Gerenciar contratos de software
- Indicar a qualidade de um produto de software
- Avaliar a produtividade do processo
- Avaliar os benefícios (em termos de produtividade e qualidade) de novos métodos e ferramentas de engenharia de software
- Avaliar retorno de investimento

Por que medir software?

- Identificar as melhores práticas de desenvolvimento de software
- Embasar solicitações de novas ferramentas e treinamento
- Avaliar o impacto da variação de um ou mais atributos do produto ou do processo na qualidade e/ou produtividade
- Formar uma *baseline* para estimativas
- Melhorar a exatidão das estimativas
- Oferecer dados qualitativos e quantitativos ao gerenciamento de desenvolvimento de software, de forma a realizar melhorias em todo o processo de desenvolvimento de software

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
- Passível de automação
- Repetível e independente do observador
- Sugere uma estratégia de melhoria

Em resumo...

- Uma métrica deve ser:
 - Válida: quantifica o que queremos medir
 - Confiável: produz os mesmos resultados dadas as mesmas condições
 - Prática: barata, fácil de computar e fácil de interpretar
- Dois contextos para medição de software
 - Processo: ex. produtividade
 - Produto: ex. qualidade

Categorização de Métricas

- Métricas diretas (fundamentais ou básicas)
 - Medida realizada em termos de atributos observados (usualmente determinada pela contagem)
 - Ex.: custo, esforço, no. linhas de código, capacidade de memória, no. páginas, no. diagramas, etc.
- Métricas indiretas (derivadas)
 - Medidas obtidas a partir de outras métricas
 - Ex.: complexidade, eficiência, confiabilidade, facilidade de manutenção

Categorização de Métricas

- Métricas orientadas a tamanho
 - São medidas diretas do tamanho dos artefatos de software associados ao processo por meio do qual o software é desenvolvido.
 - Ex.: esforço, custo, no. KLOC, no. páginas de documentação, no. erros
- Métricas orientadas por função
 - Consiste em um método para medição de software do ponto de vista do usuário, determinando de forma consistente o tamanho e a complexidade de um software.

Categorização de Métricas

- Métricas de produtividade
 - Concentram-se na saída do processo de engenharia de software.
 - Ex.: no. de casos de uso/iteração.
- Métricas de qualidade
 - Oferecem uma indicação de quanto o software se adequa às exigências implícitas e explícitas do cliente.
 - Ex.: erros/fase
- Métricas técnicas
 - Concentram-se nas características do software e não no processo por meio do qual o software foi desenvolvido.
 - Ex.: complexidade lógica e grau de manutenibilidade

Possíveis problemas com métricas

- Ex: Comparar a produtividade de engenheiros em termos de linha de código
 - Está sendo utilizado a mesma unidade de medida?
 - O que é uma linha de código válida?
 - O contexto considerado é o mesmo?
 - Todos os engenheiros são familiarizados com a linguagem de programação?
 - O que se quer realmente é o tamanho do código?
 - E a qualidade do código?
 - Como o resultado será interpretado?
 - Produtividade média de um engenheiro?
 - O que se quer com o resultado?
 - Comparar a produtividade do processo de software?

Teoria da Medição

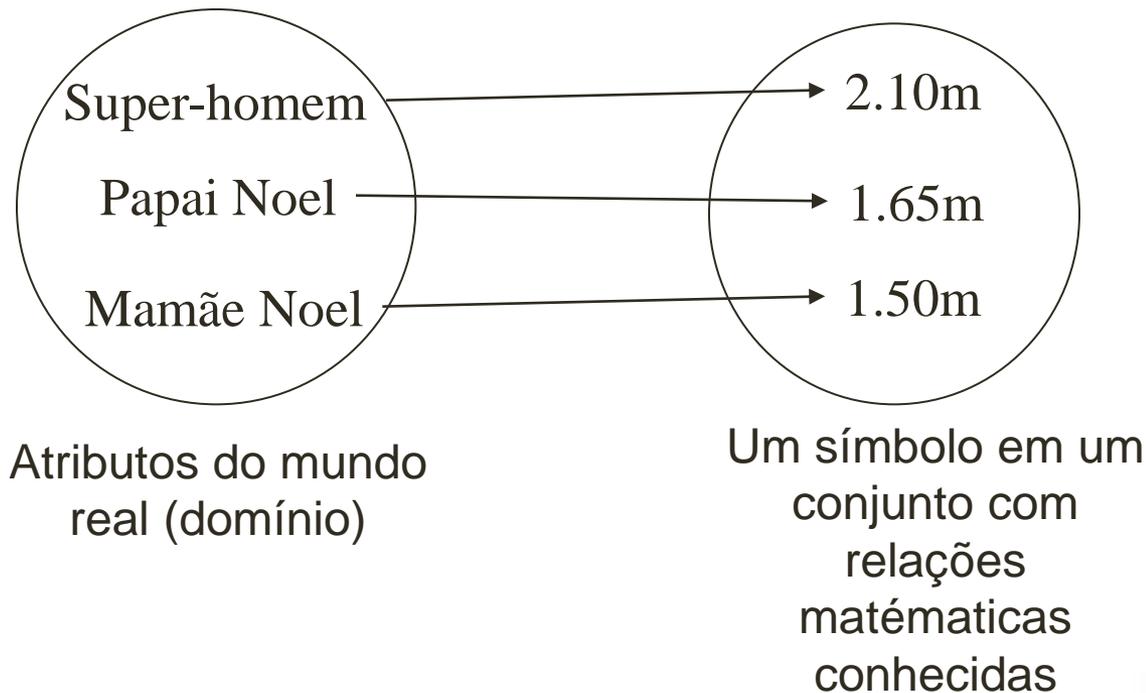
- Teoria sobre métricas pode ajudar a resolver estes problemas.

Relações Empíricas

- Ajudam a observar as relações do tipo verdadeiro/falso entre entidades do mundo real
- Ex. Relações empíricas entre o atributo altura das pessoas
 - Binária: O Super-homem é mais alto do que papai Noel
 - Unária: O Super-homem é alto
 - Ternária: O Super-homem é mais alto do que papai Noel e mamãe Noel

Medida

- Medida é uma função de mapeamento



Medição

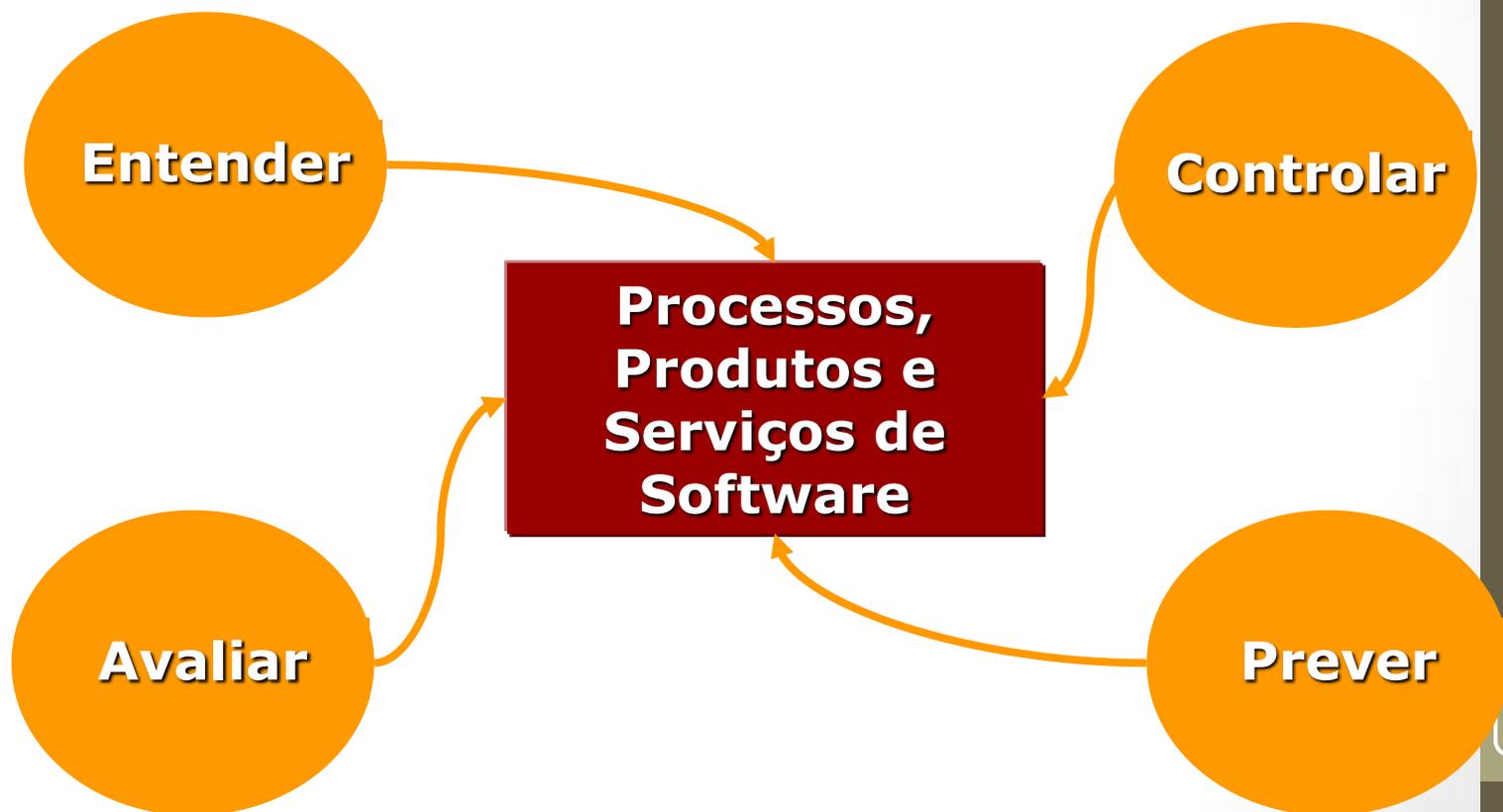
- É a atribuição de uma medida (através de um símbolo) a um atributo do mundo real
- Propósito: manipular símbolos na faixa => determinar conclusões sobre os atributos do domínio
- Para ser precisa, a medição deve especificar
 - Domínio: Será medido a largura ou altura das pessoas?
 - Faixa: A medida da altura foi feita em m ou cm?
 - Regras de mapeamento: Será permitido medir altura considerando pessoas calçadas?

Escala

- Representa os símbolos na faixa de uma medida mais as manipulações permitidas
- Ex. de manipulações:
 - Mapeamento: transformar símbolos em um conjunto em outros símbolos em outro conjunto.
 - $\{\text{verdadeiro, falso}\} \rightarrow \{1, 0\}$

Os Quatros papéis de Medição

- Segundo Humphrey, são quatro os principais papéis de Medições de Software:



Os Quatros papéis de Medição

- Entender
 - Métricas ajudam a entender o comportamento e funcionamento de processos, produtos e serviços de software
- Avaliar
 - Métricas podem ser utilizadas para tomar decisões e determinar o estabelecimento de padrões, metas e critérios de aceitação
- Controlar
 - Métricas podem ser utilizadas para controlar processos, produtos e serviços de software
- Prever
 - Métricas podem ser utilizadas para prever valores de atributos

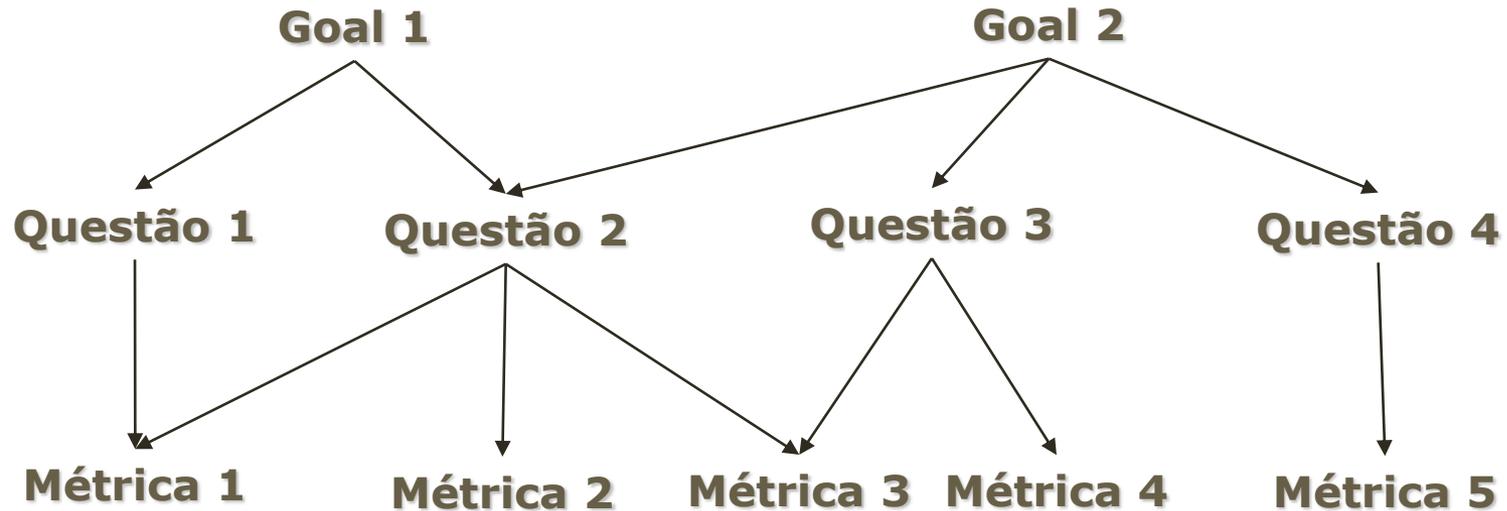
O Paradigma Goal Question Metrics (GQM)

- Usado para definir o conjunto de métricas a ser coletado
- Proposto por:
 - Basili and Rombach's, Goal-Question-Metrics Paradigm, IEEE Transactions on Software Engineering, 1988.
- Baseia-se no fato de que deve existir uma necessidade clara associada a cada métrica

O Paradigma Goal Question Metrics (GQM)

- Inicia-se com a identificação dos interessados na medição.
- Com base nos interessados, estabelecem-se os principais objetivos da medição para a organização, o projeto ou uma tarefa específica. Ex: reduzir defeitos, aumentar produtividade, etc.
- A partir dos objetivos, geram-se perguntas cujas respostas dirão se os objetivos foram ou não alcançados (ex: Qual a taxa de defeito atual? Qual a taxa de defeito após a implantação do novo processo?)
- A partir das perguntas, definem-se métricas: que dados serão necessários? Quais os formatos? Como coletar (fórmula e processo)? Onde armazenar e como utilizar?

O Paradigma Goal Question Metrics (GQM)



Exemplo do uso do GQM

- Objetivo: Assegurar que todos os defeitos são corrigidos antes do software ser liberado para uso.
- Perguntas:
 - Quantos defeitos temos atualmente?
 - Qual o status de cada defeito?
 - Qual a cobertura dos testes?
- Métricas:
 - Número de defeitos
 - Número de defeitos por status
 - Número de casos de testes planejados x executados
 - Número de requisitos testados

Selecionando Objetivos

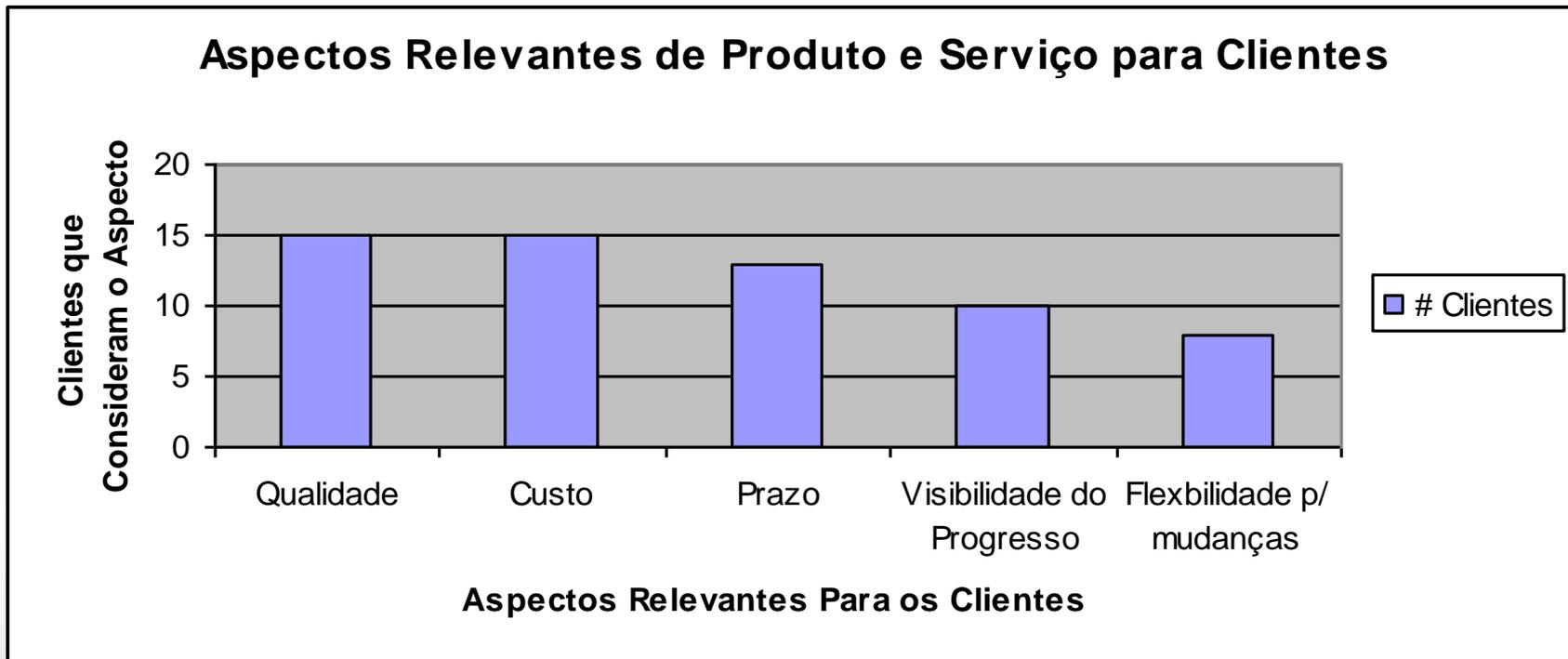
- Devem estar associados a um período de tempo
 - Aumentar a produtividade em 20% no prazo de 12 meses
 - Facilita o acompanhamento e a tomada de ações para viabilizar objetivo → pois existe um prazo!!!
- Estudos indicam que objetivos muito complexos e de longo prazo podem causar impacto na motivação
 - Objetivos menores, a curto prazo, permitem que as pessoas visualizem o progresso e alcancem sucessos
 - Com o tempo e com a maturidade da organização, os objetivos devem se tornar mais complexos e mais desafiadores

Selecionando Métricas

- Seja realista e prático
- Considere o processo e o ambiente de desenvolvimento atual
 - Não selecione métricas em que os dados sejam difíceis de serem coletados na sua realidade
- Comece com o que for possível
- A equipe não deve ser muito impactada
- Utilize a abordagem incremental
 - Com o tempo, com os benefícios, mais dados estarão disponíveis...

Selecionando Métricas

- Objetivo: Aumentar satisfação do cliente
 - Que atributos dos nossos produtos e serviços são mais importantes para os nossos clientes?



O processo de medição

- É um processo cíclico que envolve:
 - Planejar
 - Medir
 - Analisar os dados
 - Tomar decisões baseadas na análise
 - Implementar as decisões
 - Voltar a planejar e medir

Princípios de um Processo de Medição

- Um processo de medição deve:
 - Fornecer uma base para melhoria contínua do processo
 - Quantificar a qualidade e produtividade
 - Estar integrado com o ciclo de vida de desenvolvimento
 - Medir o impacto de vários métodos, ferramentas, e técnicas de melhorias

Princípios de um Processo de Medição

- Medições devem ser usadas para medir processos, não pessoas
- O processo de medição deve ter objetivos claros e bem-definidos
- O processo de medição deve ser fortemente acoplado com o processo de gerência da qualidade e integrado dentro de planos e orçamentos

Princípios de um Processo de Medição

- O processo de coleta de dados deve ser simples, e ferramentas automáticas para extração de dados devem ser usadas
- O processo de medição é contínuo e sujeito a melhoria

Características de um programa efetivo de medição

- Escolha um conjunto adequado de métricas
- Relacione as métricas ao processo de tomada de decisão (suportado pela alta administração)
- Avalie processos e não pessoas (explique os objetivos da medição)
- Não use as métricas para punir
- Envolver várias pessoas na seleção e formulação das métricas
- Estabeleça alta prioridade (recursos, ferramentas, etc.)
- Integre o programa ao desenvolvimento de software
- Alinhe aos objetivos de negócio
- Padronize e documente
- Compartilhe as métricas obtidas
- Institucionalize como parte da cultura da organização
- Integre com o programa de melhorias (ilustre o progresso e as melhorias obtidos a partir do programa)
- Ofereça planos de ação

Plano de Métricas

- Para cada objetivo técnico o plano contém informação sobre:
 - POR QUE as métricas satisfazem o objetivo
 - QUE métricas serão coletadas, como elas serão definidas, e como serão analisadas
 - QUEM fará a coleta, quem fará a análise, e quem verá os resultados
 - COMO será feito: que ferramentas, técnicas e práticas serão usadas para apoiar a coleta e análise das métricas
 - QUANDO no processo e com que frequência as métricas serão coletadas e analisadas
 - ONDE os dados serão armazenados

Após todo o planejamento...

Executar as atividades com base no planejamento realizado

Utilizar o plano de medição como base!!

Comunicar os resultados ao público alvo de cada métrica

Tomar ações com base nos resultados

Acompanhar os itens de ação

Ajustar o processo com melhorias a partir dos resultados de sua execução:

Inicialmente vai ser difícil definir todos esses procedimentos da melhor forma

Eles devem ser melhorados a medida em que o processo é executado

Novos objetivos e métricas surgem...

Melhores forma de coleta são identificadas

As orientações para realização da análise vão sendo refinadas a medida que conhecemos melhor os dados

Após todo o planejamento... (2)

- Armazenar os resultados
 - Tanto os dados, como os resultados, as ações tomadas, tudo que for relevante
 - Toda informação que contextualize a métrica ou que forneça alguma informação adicional

Dados históricos não são apenas números

Estimativas de Software

Por que é tão difícil estimar?

- É difícil conhecer se é possível desenvolver o produto desejado pelo cliente antes de conhecer os detalhes do projeto.



Por que é tão difícil estimar?

- Desenvolvimento é um processo gradual de refinamento
 - Incerteza da natureza do produto contribui para a incerteza da estimativa
 - Requisitos e escopo mudam
 - Defeitos são encontrados e demandam retrabalho
 - Produtividade varia

O Processo de Estimativas

1. Estimar o tamanho do produto
2. Estimar o esforço
3. Estimar o prazo
4. Fornecer estimativas dentro de uma faixa permitida e refinar essa faixa à medida que o projeto progride

Tipos de Estimativas

- Tamanho
 - Quantidade de software a ser produzida
 - Ex. no. linhas de código, no. pontos de função, n.o de requisitos, pontos de casos de uso
- Esforço
 - Derivado da estimativa de tamanho
 - Ex. dividindo a estimativa de tamanho por produtividade produz-se o esforço

Métricas de Software: Resumo

- As atividades de medição devem ser guiadas por objetivos
- Plano de Métricas detalham como criar programas de medição para atender a objetivos técnicos específicos
- Tendências recentes: evolução de métricas ou modelos específicos para amplos programas organizacionais de métricas

Referências

- Fenton NE and Pfleeger SL, 'Software Metrics: A Rigorous & Practical Approach' (2nd Ed.), PWS, 1998.
- www.csr.city.ac.uk/people/norman.fenton
- www.softwaremetrics.com
- IFPUG International Function Point Users Group:
www.ifpug.org
- Total Metrics: www.totalmetrics.com

Referências

- Chou, Tim. The Hidden Cost of Software. Maio 29, 2003. Url: <http://itmanagement.earthweb.com/entdev/print.php/2214031>.
- Negulescu, Radu. Software Engineering Practice – Software Metrics II. McGill University, 2002.
- Métricas de Software. Url: <http://www.internext.com.br/mssa/medidas.html>
- Haufe, Maria Isabel. Produtividade no Desenvolvimento de Software. Url: <http://www.inf.ufgrs.br/pos/SemanaAcademica/Semana99/mariaisabel/mariaisabel.html>
- Métricas e Estimativas de Software – O início de um rally de regularidade. Url: <http://www.apinfo.com/artigo44.htm>
- Pressman, Roger. S. Engenharia de Software. Makron Books, 1995.