

Uso de métricas de software na quantificação dos custos de erros encontrados em um projeto real

Thiago Pelizoni
André Terceiro
Rodrigo Raminelli
Pedro Antonio Saraiva Júnior

2013

Resumo

O custo de defeitos em software tende a crescer exponencialmente a medida que as etapas do desenvolvimento de software avançam. O valor pode ser alto ao ponto de custar entre 90 a 880 vezes o valor do levantamento deste requisito na sua fase inicial.[NISL 2002] A Engenharia de Software utiliza métricas de produto afim de construir um software de alta qualidade. Desta forma, este trabalho tem o intuito de demonstrar a aplicação das técnicas de métricas de software no projeto de inscrição dos participantes da Corrida de São Silvestre afim de quantificar os erros encontrados, seja na fase de testes, homologação ou produção e calcular o custo destes erros, evidenciando o quanto a empresa economizaria em uma melhoria no processo.

1 Introdução

São comuns abordagens a respeito de qualidade de software, que é necessário seguir boas práticas para se obter um software de alta qualidade, no entanto, como pode ser definida a qualidade? Em um sentido geral, qualidade de software é a satisfação de requisitos funcionais e de desempenho explicitamente declarados, normas de software explicitamente documentadas e características implícitas que são esperadas em todo o software desenvolvido profissionalmente. A partir desta definição, podem ser enfatizados três pontos importantes:

1. Requisitos de software são a fundação a partir da qual a qualidade é medida. A falta de conformidade com os requisitos é falta de qualidade.
2. Normas especificadas definem um conjunto de critérios de desenvolvimento que guiam o modo pelo qual o software é construído. Se os critérios não são seguidos, quase sempre, irá resultar em falta de qualidade.
3. Há um conjunto de requisitos implícitos que frequentemente não são mencionados, por exemplo, o desejo de facilidade de uso (usabilidade). Se o software satisfaz os requisitos explícitos, mas deixa de satisfazer requisitos implícitos, a qualidade do software é suspeita.

[Roger S Pressman 2006]

1.1 Métricas de Software

A qualidade de um produto é função de quanto ele muda o mundo para melhor

Tom DeMarco[Roger S Pressman 2006]

Um elemento-chave que a engenharia utiliza em seu processo é a medição, de modo a ter um melhor entendimento dos atributos dos modelos criados afim de avaliar a qualidade seus produtos ou sistemas desenvolvidos.

De acordo com [Roger S Pressman 2006], a medição é um processo pelo qual números ou símbolos são associados aos atributos de entidades do mundo real, de modo que os determinem de acordo com regras claramente definidas. Nas ciências físicas, medicina, economia e mais recentemente nas ciências sociais, atualmente é possível medir atributos que anteriormente não eram mensuráveis. Obviamente, tais medições em engenharia de software não são tão refinadas quanto muitas medições nas ciências físicas mas, elas existem e decisões importantes são tomadas com base nelas. A obrigação de tentar medir o não mensurável afim de melhorar o entendimento de entidades particulares é tão potente em engenharia de software como em qualquer outra disciplina.

1.1.1 Medidas, Métricas e Indicadores

Uma ciência é tão madura quanto seus instrumentos de medição

Louis Pasteur[Roger S Pressman 2006]

Em engenharia de software, uma medida fornece uma indicação quantitativa da extensão, qualidade, dimensão, capacidade ou tamanho de algum produto ou processo, deste modo, podemos definir como *medição* o ato de determinar uma medida.[Roger S Pressman 2006]

Quando um ponto de dados é coletado, por exemplo, o número de erros descoberto em um componente de software, uma medida é estabelecida. Neste caso, a medição ocorre como o resultado da coleção de um ou mais pontos de dados, por exemplo, um certo número de revisões de componentes e testes de unidade são investigados afim de coletar medidas do número de erros de cada um.

Uma métrica de software visa relacionar as medidas individuais de algum modo, por exemplo, o número de erros encontrados por revisão ou o número médio de erros encontrados por teste de unidade. Com posse dessas informações, um engenheiro de software desenvolve métricas de modo que os indicadores possam ser obtidos. Um *indicador* consiste em um *métrica* ou em uma combinação de *métricas* cujo intuito é fornecer uma profundidade na visão do processo de software, projeto de software ou o produto em si.[Roger S Pressman 2006]

1.1.2 Princípios de Medição

De acordo com [Roger S Pressman 2006] um processo de medição pode ser caracterizado por cinco atividades:

1. *Formulação*: A derivação de medidas e métricas de software adequadas para a representação do software que está sendo considerado.
2. *Coleta*: Mecanismo usado para acumular os dados necessários para derivar as métricas formuladas.
3. *Análise*: Cálculo de métricas e aplicação das ferramentas matemáticas.
4. *Interpretação*: Avaliação das métricas em um esforço para ganhar profundidade na visão da qualidade da representação.
5. *Realimentação*: Recomendações derivadas da interpretação das métricas de produto transmitidas à equipe de software.

1.2 Objetivo deste trabalho

Tendo por base as informações anteriormente citadas, o objetivo deste trabalho demonstrar a aplicação das técnicas de métricas no projeto de inscrição dos participantes da Corrida de São Silvestre, afim de quantificar os erros encontrados, sejam estes na fase de testes, homologação ou produção; calcular o custo destes erros afim evidenciar o quanto a empresa economizaria em uma melhoria no processo, detectando os erros em fases do processo de desenvolvimento em que tal detecção é mais barata.

2 Materiais e métodos

Materiais e métodos, como foi feito?

3 Resultados

Resultados, o que foi observado

4 Conclusão

Considerações finais, o que se pode concluir dos resultados?

Referências

[Ian Sommerville 2003] Ian Sommerville. *Engenharia de Software*. [S.l.]: Pearson Education do Brasil, 2003.

[NISL 2002] NISL. *The Economic Impacts of Inadequate Infrastructure for Software Testing - Final Report*. 2002. 37,118 p. URL: <http://www.nist.gov/director/planning/upload/report02-3.pdf> (Acessado em 05/2013).

[Roger S Pressman 2006] Roger S Pressman. *Engenharia de Software*. [S.l.]: Mcgraw-hill Interamericana, 2006. 348,349,350,352 p.