Critérios para Apoiar a Decisão Sobre o Momento de Parada dos Testes de Software

Victor Vidigal Ribeiro

Guilherme Horta Travassos

{vidigal, ght}@cos.ufrj.br

Agenda

- Introdução
- Resultados da revisão
- Corpo de conhecimento
- Consulta ao corpo de conhecimento
- Exemplo de critério de parada
- Limitações dos critérios de parada
- Conclusões
- Referências

Introdução

- Testes de software
  - Técnica de verificação e validação (dinâmica) que consiste na execução do software com o objetivo de revelar falhas
    - Com a identificação das falhas do software, é possível identificar e corrigir as faltas que geraram as falhas através do processo de depuração
    - Com menos faltas, assume-se que o software falhe menos e, assim aumenta a confiança no software

Introdução

Testes contribuem para a qualidade do software

- \+ Impressão dos usuários
- \+ Relacionamento com clientes
- − Perdas financeiras
- − Perdas humanas

Introdução

- Limitações dos Testes de Software
  - **Atividade custosa:** atividades relacionadas ao teste de software consomem cerca de 50% do orçamento de um projeto
  - **Testes exaustivos são inviáveis:** não é viável executar todas as possíveis combinações de entrada e analisar as saídas geradas por essas entradas
  - Testes não provam que um software está correto

- Sempre pode haver uma bateria de testes que revelaria uma falha.
  - Qual o momento ideal para parar os testes e liberar o software para utilização?

Introdução

- Quais critérios têm sido utilizados para determinar o momento de parada dos testes de software?

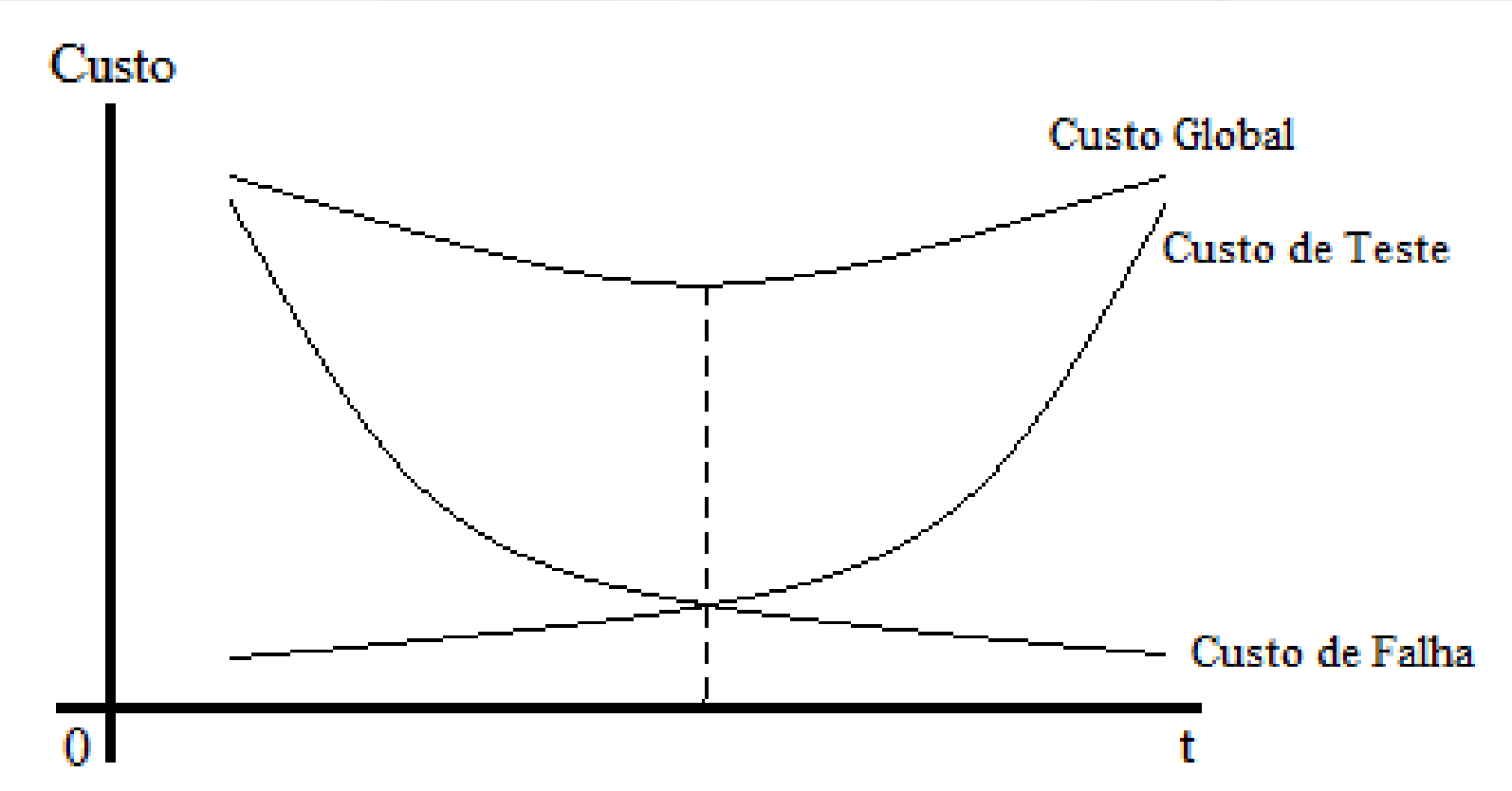
- Execução de uma *quasi*-Revisão Sistemática
  - Identificação de **74** critérios de parada!

Resultados da Revisão

- Critérios de parada começaram a ser propostos na década de 70
- Geralmente baseados em modelos de confiabilidade
- Objetivam encontrar o ponto de equilíbrio entre o custo com testes e o custo de falha[1]

[1]*Custo da falha: custo de encontrar falhas na fase de produção*

Resultados da Revisão



- **Custo de Teste:** custo total dispendido com testes
- **Custo da Falha:** custo de encontrar falhas na fase de produção
- **Custo Global:** custo total do processo de desenvolvimento

Corpo de Conhecimento

- Repositório de critérios de parada caracterizados e gerados a partir da *quasi*-revisão sistemática

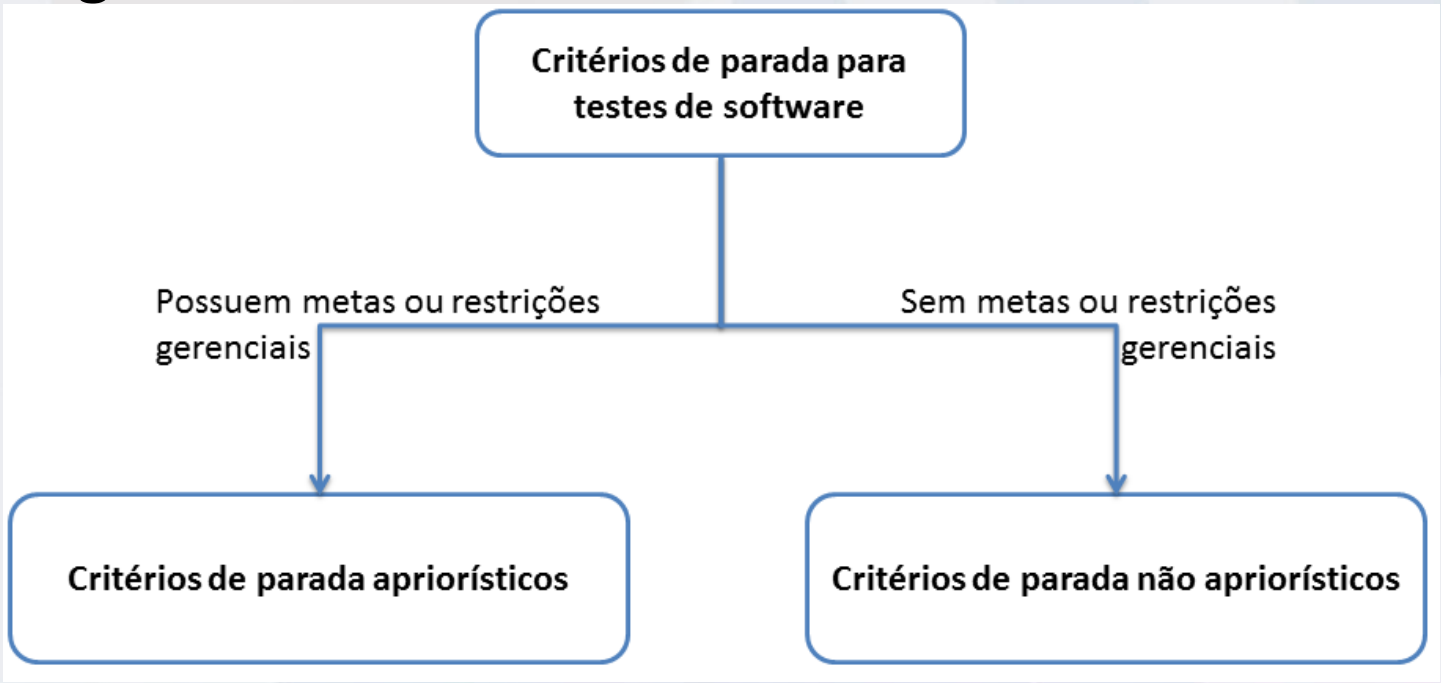
- Atributos de caracterização visando apoiar o processo de escolha de critérios de parada

Corpo de Conhecimento

- Atributos de caracterização do Corpo de Conhecimento
  - **Confiabilidade pré-determinada**
  - **Orçamento para testes pré-determinado**
  - **Tempo para testes pré-determinado**
  - Contexto de aplicação
  - Caracterização do software
  - Objetivo do critério de parada
  - Níveis de testes
  - Modelos Utilizados
  - Habilidades necessárias

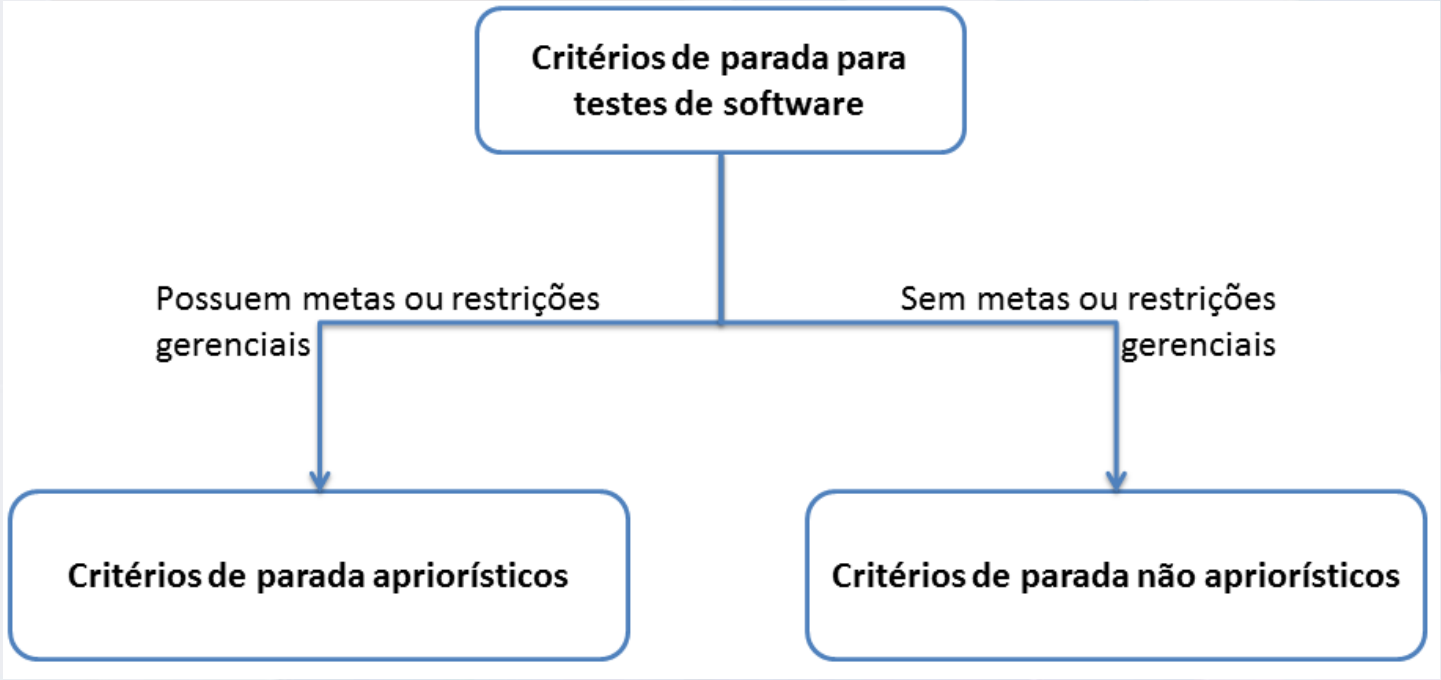
Corpo de Conhecimento

- Apriorísticos
  - Auxiliam a atingir uma meta para a fase de testes determinada a priori e/ou avaliar se esta meta foi atingida



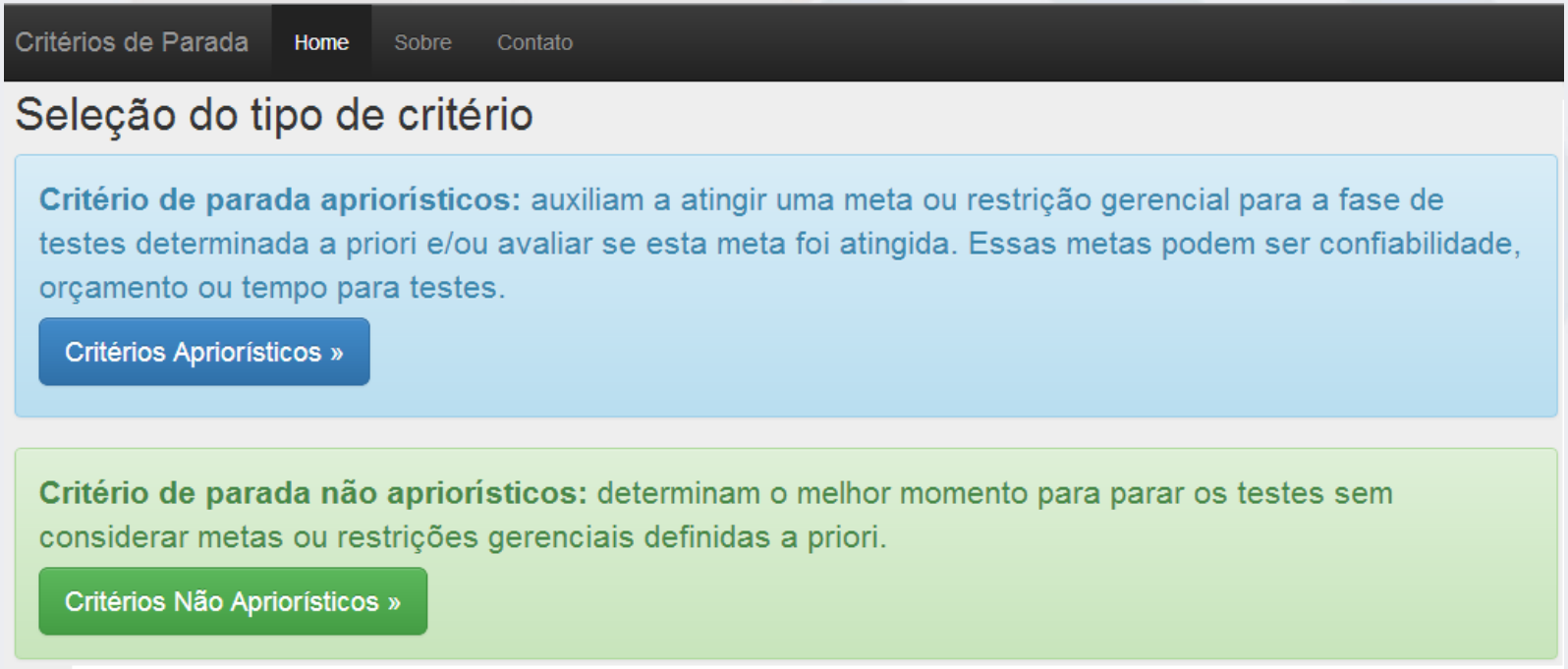
Corpo de Conhecimento

- Não Apriorísticos
  - Determinam o melhor momento para parar os testes sem considerar metas definidas a priori



Consulta ao Corpo de Conhecimento

- Desenvolvimento de uma ferramenta de consulta ao corpo de conhecimento
  - http://lens-ese.cos.ufrj.br/criteriosparada/



Consulta ao Corpo de Conhecimento

- Características dos critérios VS Características do projeto de software

Pesquisar

Autor	Ano Publicação	Título	Confiabilidade pré-determinada	Duração testes pré-determinada	Custo testes pré-determinado	Embarcado	Missão Crítica	Depuração perfeita	Descrição	Formulário
Okumoto, K. and Goel, A.L.	1979	Optimum release time for software systems	S	N	N	N	N	NI		
Koch, H.S. and Kubat, P.	1983	Optimal Release Time of Computer Software	N	S	N	N	N	S		
Yamada, Shigeru and Osaki, Shunji	1985	COST-RELIABILITY OPTIMAL RELEASE POLICIES FOR SOFTWARE SYSTEMS	S	N	N	N	N	NI		
Ross, S.M.	1985	Software Reliability The Stopping Rule Problem	S	N	N	N	N	S		
Yamada, S.a , Osaki, S.b	1987	Optimal software release policies with simultaneous cost and reliability requirements	S	N	N	N	N	NI		

1/9

Exibir 5 registros

Exemplo de Critério de Parada

$$T_0 = \frac{1}{b} \ell n \frac{ab(c_2 - c_1)}{c_3}$$

Onde,
- **a**: número estimado total de defeitos
- **b**: risco de uma falha ocorrer
- $\mathbf{c_1}$: custo de correção de um defeito durante a fase de testes
- $\mathbf{c_2}$: custo de correção de um defeito com o software em produção ($c_1 < c_2$)
- $\mathbf{c_3}$: custo por unidade de tempo devido ao atrazo na entrega do software

Exemplo de Critério de Parada

Parâmetros utilizados para cálculo
- **a**: 1348
- **b**: 0,124
- $\mathbf{c_1}$: \$1 por defeito
- $\mathbf{c_2}$: \$5 por defeito
- $\mathbf{c_3}$: \$100 por defeito
- **t**: 100 semanas

Aplicando fórmula demonstrada na Figura 2-4:

$$T_0 = \frac{1}{0,124} \ell n \frac{1348 \cdot 0,124 (5 - 1)}{100}$$

Resolvendo a euqação, o tempo de teste ótimo para o sistema é:

$$T_0 = 15,3\ Semanas$$

Limitações dos Critérios de Parada

- Iniciativas independentes
  - A quantidade de critérios encontrados é um indício da importância do tema
  - Cada iniciativa cria um critério de parada que pode funcionar apenas no contexto em que ele foi proposto
- Falta de caracterização do software
  - A maioria dos artigos não caracteriza o software em que o critério de parada pode ser aplicado
  - Caracterizações são imprecisas (“sistemas de software em larga escala”)
  - Não apresentam informações detalhadas sobre o contexto de aplicação

Limitações dos Critérios de Parada

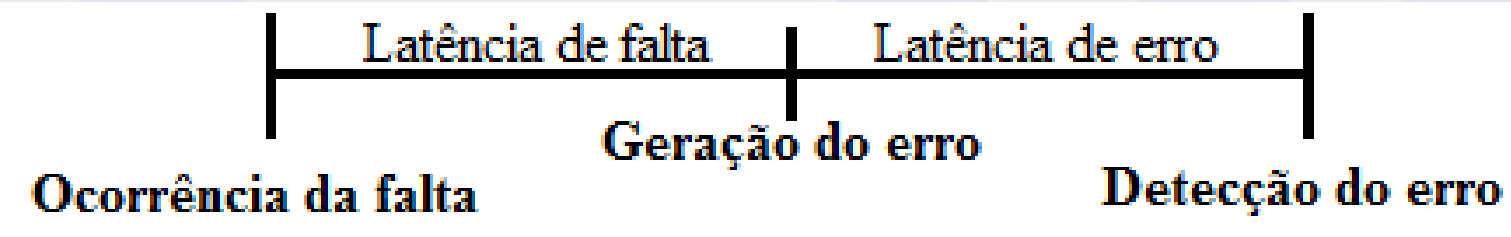
- Critérios baseados em modelo de confiabilidade
  - Não entendemos completamente o processo de desenvolvimento para gerar um modelo confiável
- Depuração Perfeita
  - Diversos critérios consideram depuração perfeita
- Falta de Avaliação Experimental

Limitações dos Critérios de Parada

- Não considera processo de testes dividido em fases
  - Quando parar cada fase de teste para prosseguir para fase posterior?
- Não consideram a forma como os testes são conduzidos
  - Casos de testes não eficientes podem antecipar o momento de parada

Limitações dos Critérios de Parada

- Latência de Falta
  - Tempo entre a ocorrência da falta e a geração do erro
- Latência de Erro
  - Tempo da geração do erro até sua detecção



Conclusões

- 74 critérios de parada encontrados e catalogados
- Sugestão de um método para seleção dos critérios de parada
- Apresentação de limitações dos critérios
  - Falta de avaliação formal
  - Não existe um critério de parada que possa ser utilizado em qualquer contexto!

Conclusões

- Utilizar um critério de parada VS parar os testes aleatoriamente

- Resultados da pesquisa utilizados como insumo para aprimorar a tomada de decisão na escolha de critérios de parada

Referências

BOLDRINI, J. L., COSTA, S. R., FIGUEIREDO, V. L., et al. (1980), —Álgebra Linear||, 3 ed., capítulo 8, Harper & Row do Brasil.

CAMUFFO, M., MAIOCCHI, M., MORSELLI, M., (1990), "Automatic software test generation". Information Software. Technology. 32 (5), 337–346.

CHÁVEZ, T., (2000) "A Decision-Analytic Stopping Rule for Validation of Commercial Software Systems", IEEE Transactions on Software Engineering.

DIAS NETO, A. C., (2009) "Seleção de Técnicas de Teste Baseado em Modelos", Tese de D.Sc., COPPE/UFRJ, Novembro.

DIJKSTRA, E. W. (1972). "The Humble Programmer". Communications of the ACM 15 (10): 859–866. doi:10.1145/355604.361591. (EWD340) PDF, ACM Turing Award lecture

GARG, M., LAI, R., HUANG JEN, S., (2010), "When to Stop Testing: A Study from the Perspective of Software Reliability Models", IET Software, Vol. 5, pp. 263-273

MATHUR A. P., (2008), "Foundations of Software Testing", 1ª Edition. Addison-Wesley Professional.

Referências

MASUDA, Y., MIYAWAKI, N., SUMITA, U. e YOKOYAMA, S., (1989), “A statistical approach for determining release Time of Software System with Modular Structure Reliability”, IEEE Transactions on, 365 -372

OKUMOTO, K. e GOEL, A., (1979b), “Optimum Release Time for Software Systems”, Computer Software and Applications Conference. Proceedings. COMPSAC 79. The IEEE Computer Society's Third International, 500 - 503

PAI, M. McCULLOCH, M. GORMAN, J.D. et al. (2004) “Systematic Reviews and meta-analyses: An illustrated, step-by-step guide”, The National Medical Journal of India, vol. 17, n.2.

ROCHA, A. R. C., MALDONADO, J. C., WEBER, K. C. (2001), “Qualidade de software - Teoria e prática ||”, Prentice Hall, São Paulo.

YAMADA, S. e OSAKI, S., (1985) “Cost-reliability optimal release policies for software systems”. IEEE Transactions on Reliability, R-34, 422 – 424.

Critérios para Apoiar a Decisão Sobre o Momento de Parada dos Testes de Software

Victor Vidigal Ribeiro

Guilherme Horta Travassos

{vidigal, ght}@cos.ufrj.br