

Data: June de Sunday de 2010 @ 22:45:53 UTC  
Tópico: Artigos Publicados

## **Teste Unitário e de Cobertura para Java Script com JsUnit e JsCovarage**

### **Teste Unitário e de Cobertura para Java Script com JsUnit e JsCovarage:** *Uma Abordagem Prática*

Artigo intitulado "Teste Unitário e de Cobertura para Java Script com JsUnit e JsCovarage" foi publicado na revista [Engenharia de Software Magazine](#) do [Grupo DevMedia](#), na edição 25.

#### **Autores:**

Jenifer Vieira Toledo  
Elessandro Rodrigues Marques  
Marcelo Daibert  
Marco Antônio Pereira Araújo

A revista pode ser adquirida através do site:  
[Engenharia de Software Magazine - Edição 25.](#)

#### **Resumo:**

O conceito de teste de software surgiu no final dos anos 50 como uma atividade isolada da Engenharia de Software. Era aplicado de forma manual pelos desenvolvedores com o objetivo de revisar seus códigos fonte. Com o passar dos anos, com esta abordagem sendo aplicada, os projetos passaram a apresentar melhora na qualidade do produto final quando comparados a projetos de software que não aplicavam nenhuma etapa de testes. Mesmo assim, a tarefa de efetuar testes em software foi considerada secundária por algum tempo, chegando a ser vista até mesmo como uma punição para os programadores, ou como uma tarefa onde não se deveria gastar muito tempo e despender grandes investimentos. Além disso, testar software para descobrir os defeitos que desqualificam os produtos para os consumidores era visto como um trabalho tedioso e cansativo, a ser evitado pelos mais artificiosos dos especialistas em tecnologia. Entretanto, a responsabilidade com a qualidade do produto final desenvolvido deve ser levada em consideração e justamente por isso o tema teste de software está em evidência. Nos dias atuais, teste de software é visto como um subitem dentro da Qualidade de Software, sendo que para conseguir tal qualidade é necessário definir padrões de trabalho, métodos e melhorar o processo de desenvolvimento, assegurando de fato que os defeitos sejam identificados o mais cedo possível durante o processo de desenvolvimento de software. O custo de um defeito em um software não deve ser avaliado somente pelo aspecto financeiro. Os softwares são escritos para controlar equipamentos com propósitos mais variados possíveis, como dosagens de medicamentos, exames médicos, construção e controle de aviões, aeronaves espaciais, satélites, entre outros vários sistemas. E, em muitos casos, um defeito pode causar a destruição total de um caro equipamento ou até mesmo em perda de vidas humanas.

Esta Notícia é proveniente do Daibert.pro Portal System:  
<http://daibert.pro/Portal>

A URL para a Notícia é:  
<http://daibert.pro/Portal/index.php?op=NEArticle&sid=112>