

## INTEGRAÇÃO CONTÍNUA COM APLICAÇÃO DE TESTES DE REGRESSÃO

**GIANNECCHINI, Nilo**

Graduação em Tecnologia da Informação, FT - Unicamp, Limeira, São Paulo, Nilo.gianneccchini@gmail.com.

**RESUMO** - Quando há um aumento de complexidade do projeto, a necessidade de integrar código de diferentes membros de uma equipe cresce. Um dos pontos de maior risco no ciclo de desenvolvimento de software é a integração de mudanças independentes feitas por diferentes membros de uma equipe. Este trabalho apresenta uma metodologia para a execução de testes em um ambiente de integração contínua (CI). O objetivo é o de propor uma solução simples de aplicar uma bateria de vários tipos de ensaios. Neste ambiente, que por sua vez é capaz de identificar, de forma contínua, quando há mudanças no sistema. Como resultado, uma melhoria na detecção de falhas devido a mudanças feitas por desenvolvedores foi alcançada. Além disso, espera-se a fornecer uma orientação de boas práticas que podem reduzir a dificuldades de execução, adaptação e manutenção deste ambiente no futuro.

**Palavras-chave:** sistema de integração contínua; automação de teste; ambiente de integração de software; testes de regressão.

### INTRODUÇÃO

Esperar o fim do projeto para fazer a integração do sistema leva a todos os tipos de problemas de qualidade, como falhas na integração de componentes, que podem trazer custos adicionais e atrasos para o plano do projeto. Para evitar este problema, a integração deve ser continuamente realizada, tornando os riscos a serem geridos de forma mais rápida. Se ocorrer um problema, a equipe será alertada prontamente. De acordo com a Gartner [1], as empresas investem cerca de 40% de seus orçamentos de TI (Tecnologia da Informação) em integração de projetos. Quando a tarefa é implementada com sucesso, existe uma redução de custos e tempo de desenvolvimento, bem como um retorno mais rápido do investimento (ROI) e maior produtividade. Além disso, esta tarefa ainda colabora para alcançar os principais objetivos de negócios e melhora o relacionamento com clientes, parceiros e funcionários.

Observa-se que as empresas, especialmente pequenas e médias empresas, têm dificuldade para associar metodologias de teste de software para os seus processos de desenvolvimento. A dificuldade é muitas vezes a escassez de recursos financeiros e o pessoal disponível. Assim, a seguinte pergunta surgiu: uma metodologia de teste de software pode ajudar na organização de processos de negócios internos, para que se reflita na qualidade e satisfação do produto por parte dos clientes e destacar no mercado competitivo? O uso do IC pode reduzir os custos de manutenção e de retrabalho, que conduz a minimizar as falhas no produto entregue ao cliente. Isso reflete em um maior nível de confiabilidade do sistema. Com mais tempo disponível, a equipe pode investir em melhorias e novos planos [2].

## **METODOLOGIA**

A metodologia faz uso de técnicas, procedimentos e ferramentas para melhorar o processo de teste, que permita desenvolver produtos com maior qualidade. A metodologia de amostragem é baseada em três componentes:

- Formação: formação em conceitos básicos de teste, teste de processo e testes técnicos;
- Processo de teste: a realização do planejamento, projeto, implementação e acompanhamento dos testes;
- Suporte para geração de documentos: aplicação técnica para a criação de documentos baseados em padrões (por exemplo, o padrão IEEE).

A metodologia de teste proposto se baseia em alguns pressupostos básicos:

- Sistema e testes de aceitação são concebidos e executados sob a responsabilidade da equipe de teste;
- Plano de Teste: apresenta o planejamento, incluindo o escopo, abordagem, recursos e cronograma. Identifica os itens e recursos a serem testados e os riscos associados;
- Relatórios de teste: identifica os itens enviados para análises considerando as diferentes equipes que são responsáveis pelo desenvolvimento e teste; apresenta registros cronológicos dos detalhes relevantes relacionados com os testes; documenta qualquer evento que ocorra durante a atividade de teste e exigindo uma análise mais aprofundada.

## **RESULTADOS OBTIDOS**

Uma primeira validação da proposta foi realizada num sistema JavaScript. Normalmente, a cada novo componente do sistema entregue a equipe de teste recebe uma média de vinte falhas após o teste de aceitação feito pelos usuários.

Os testes de unidade criam os parâmetros, testes integrados realizam testes de parametrização e os testes de regressão executam todas as etapas anteriores para cada nova transação adicionada. Durante a nova campanha de testes foram utilizados o GruntJS para executar todos os testes de desempenho de uma bateria de testes em todas as transações existentes.

Como resultado, o usuário relatou uma diminuição na quantidade de falhas encontradas nos testes de aceitação.

## **CONCLUSÕES**

Considerando os desafios para manter a construção de sistemas à prova de falhas, os desenvolvedores de empresas olham para automação na execução de testes. Este trabalho tem como objetivo oferecer uma alternativa que irá contribuir para a melhoria dos ambientes de teste visando à qualidade de software.

Para trabalhos futuros pretendemos planejar casos de teste mais complexos para um grande conjunto de sistemas que devem ser submetidos ao ambiente CI em paralelo com as suas cópias no ambiente real.

## **REFERÊNCIAS**

- [1] Gartner - Integração de Aplicativos e Serviços na Web. Abril 2004, Rio de Janeiro
- [2] Rocha, A. R. C., Maldonado, J. C., Weber, K. C. Qualidade de software – Teoria e prática. São Paulo: Prentice Hall, 2001.