

Documentação de teste baseado na Norma IEEE 829 – estudo de caso: “sistema de apoio a tomada de decisão”

Mariana Zanuzzio Blanco

Abstract: *The test process is crucial for a good software quality with the fewest possible errors. Therefore, it is necessary to plan the test cases for the entire development project and not just the final stage. Thus, some model documents are designed to implement unit, system and integration tests and allowing the organization and the errors corrected prior to customer delivery. This study to describe a test documentation based on the IEEE 829 and its applicability to a case study_Support System for decision making.*

Keywords: *software quality, tests, test documentation*

Resumo: O processo de teste é crucial para obter um software de boa qualidade com a menor quantidade de erros possíveis. Para isso, é necessário planejar os casos de testes durante todo o projeto de desenvolvimento e não apenas na fase final. Assim, alguns modelos de documentos são desenvolvidos para implementar os testes de unidade, integração e sistema permitindo a organização e a da correção dos erros antes da entrega ao cliente. Esse estudo descreve uma documentação de teste baseada na norma IEEE 829 e a sua aplicabilidade no estudo de caso “Sistema de Apoio a tomada de decisão”.

Palavras-chave: qualidade de software, testes, documentação de teste

I. INTRODUÇÃO

Como é notório, a tecnologia avança desde a revolução industrial à velocidade crescente. Novas tecnologias apresentam-se a todo momento. Estar atento a essas transformações e evoluções é fundamental. Aliado a esse processo contínuo de transformação é indispensável atender aos interesses econômicos com custos mais baixos e as novas tecnologias.

Essa é a busca incansável da engenharia de software. Embora o avanço seja evidente, ainda assim apresenta problemas. Muitos softwares apesar de serem submetidos a um longo processo de desenvolvimento apresentam erros quando se tornam operacionais. Para minimizá-los, a atividade de teste é introduzida durante todo o desenvolvimento de software, visando uma maior confiabilidade do produto.

Segundo Pressman (2006), o objetivo do teste é encontrar o maior número possível de erros com esforço controlado aplicado durante um intervalo de tempo previsto. Assim, apesar da resistência das empresas em aplicar uma metodologia de teste, devido ao custo agregado, essa relutância está sendo repensada, pois estão descobrindo que a qualidade do produto final e consequentemente a satisfação do cliente são alcançadas com esse processo.

Nesse contexto, esse trabalho tem como objetivo descrever um modelo de documentação de teste baseado na norma IEEE

829 visando auxiliar o desenvolvedor de software a organizar, implementar os métodos de testes e facilitar o processo de detecção e correção de erros do produto final.

Normalmente, a implementação do teste ocorre apenas no final do projeto e com prazos curtos, o que resulta em uma atividade de teste insatisfatória. Portanto, deve ser feito o planejamento e a geração de casos de teste durante todo o desenvolvimento do sistema. Sendo assim, é apresentado um estudo de caso para exemplificar a metodologia de teste.

Portanto, sabendo a importância do tema, o restante do artigo está organizado da seguinte forma: Na Seção II apresenta-se a fundamentação teórica dos testes em sistema. Na sequência, Seção III descreve-se a documentação de teste. Na Seção IV apresenta um estudo de caso no qual será implementada a metodologia de teste. A Seção V detalha os documentos utilizados na implementação da metodologia de teste no estudo de caso. Na Seção VI mostra alguns trabalhos relacionados a esse estudo. Finalmente, a Seção VII apresenta a conclusão do trabalho.

II. DEFINIÇÃO DE TESTE

Segundo Crespo et al. (2004): “*Teste de software é o processo de executar o software de uma maneira controlada com*

o objetivo de avaliar se o mesmo se comporta conforme o especificado.”

Infelizmente não é possível testar todas as entradas de dados e suas centenas ou milhares de combinações possíveis. Criar casos de teste para todas essas possibilidades é impraticável, pois levaria muito tempo e seria economicamente inviável (Myers, 2004).

Segundo Myers (2004), à medida que as fases do projeto são implementadas, os custos em descobrir e corrigir erros aumentam exponencialmente.

Para Ali et al. (2007), a programação orientada a objeto (OO) oferece vários benefícios, tais como encapsulamento, abstração e a reutilização de códigos com o objetivo de melhorar a qualidade de software. No entanto, além dessas funcionalidades da programação OO, também introduz desafios para as pessoas responsáveis pelo teste do sistema, pois as interações entre os objetos podem dar origem a erros sutis que podem ser difíceis de detectar.

A atividade de teste está dividida em três fases: teste de unidade, teste de integração e teste de sistema. O teste de unidade é o primeiro a ser realizado e analisa cada unidade do sistema (uma unidade é a menor parte do sistema que possa ser testada) separadamente na busca de erros. Logo após, vem o teste de integração que verifica a integração entre as unidades e, por último, o teste de sistema verifica-se o desempenho, o comportamento e as funcionalidades do sistema estão conforme a especificação.

As fases que serão aplicadas dependerão da estratégia utilizada pela empresa, ou seja, uma empresa poderá adotar a estratégia de fazer somente o teste de sistema e não aplicar teste de unidade e teste de integração (Crespo, 2004).

Para que possa implementar essas fases, existem basicamente duas técnicas: caixa-branca e caixa-preta. Teste caixa-branca, conhecido também como teste estrutural, tem o objetivo de gerar casos de teste baseado na estrutura interna do programa verificando as linhas de comandos, os loops, as variáveis e as estruturas condicionais. Enquanto que o teste caixa-preta ou teste funcional analisa se todos os requisitos solicitados foram aplicados no software e estão funcionando.

A arquitetura de software orientado a objetos (OO) resulta em vários subsistemas e classes. Cada um executa funções que ajudam a satisfazer os requisitos do sistema. É necessário testar um sistema OO em uma variedade de níveis diferentes para descobrir erros que possam ocorrer à medida que as classes colaboram uma com as outras e os subsistemas comunicam entre as camadas (Pressman, 2006).

III. DOCUMENTAÇÃO DE TESTE

A metodologia de teste aplicada será baseada nos artefatos sugeridos pela norma IEEE 829-1998 (IEEE, 1998).

Essa norma descreve oito documentos para as atividades de teste de um produto de software. Esses documentos serão usados na tarefa de planejamento, especificação e relato de testes.

O documento Plano de Teste é utilizado na tarefa de planejamento para execução do teste, no qual serão identificadas as funcionalidades a serem testadas com ênfase nas datas, pessoas

envolvidas e riscos. Esse documento é de responsabilidade da equipe de testes.

A tarefa de especificação de testes é composta pelos documentos Especificação de Projeto de Teste, Especificação de Caso de Teste e Especificação de Procedimento de Teste.

A Especificação de Projeto de Teste refina a abordagem apresentada no Plano de Teste, que além de identificar os casos e os procedimentos de teste, se existirem, apresenta os critérios de aprovação.

A Especificação de Caso de Teste identifica os casos de teste, com os dados de entrada, resultados esperados, ações e condições gerais para a execução do teste.

A Especificação de Procedimento de Teste identifica quais serão os passos para executar os casos de teste. A criação desse documento é de responsabilidade dos líderes de projetos, da equipe de teste e dos usuários.

Os relatórios de teste são compostos por quatro documentos: Diário de Teste, Relatório de Incidente de Teste, Relatório-Resumo de Teste e Relatório de Encaminhamento de Item de Teste.

O Diário de Teste exibe os registros cronológicos dos dados relevantes relacionados com a execução dos testes.

O Relatório de Incidente de Teste documenta qualquer evento que ocorra durante a atividade de teste e que necessite de análise posterior. No qual são relatados os erros que podem ocorrer durante a execução do teste. Esses erros podem ser causados tanto pela falha na construção do teste como por erros de programação. Após o registro do erro o relatório deverá ser encaminhado para o responsável pela correção.

O Relatório-Resumo de Teste mostra um resumo dos resultados das atividades de teste associadas com uma ou mais especificações de projeto de teste e provê avaliações baseadas nesses resultados.

O Relatório de Encaminhamento de Item de Teste identifica os itens encaminhados para teste no caso de equipes distintas serem responsáveis pelas tarefas de desenvolvimento e de teste.

A norma pode ser ajustada - para softwares de baixa complexidade - abreviando o conteúdo dos documentos ou agrupando alguns. Dessa forma os custos de produção serão reduzidos (Bruneli, 2006). Além disso, as pessoas responsáveis pelo teste poderão decidir aplicar um único plano que contemple todas as fases de teste ou um plano para cada fase: unidade, integração e sistema.

Portanto, utilizando essa norma podem-se implementar os testes na fase de planejamento e projeto e também na fase de teste propriamente dita, dessa forma evita-se iniciar os testes após a conclusão da codificação.

IV. ESTUDO DE CASO: SISTEMA DE APOIO A TOMADA DE DECISÃO

Sistema de Apoio a tomada de Decisão (SAD) é um sistema desenvolvido para empresas de pequeno e médio porte com a finalidade de obter informações referentes a algumas questões, como:

- Quais clientes estão deixando de comprar?
- Com que frequência ocorrem as compras de cada cliente?

- Quais são os clientes que compraram recentemente?

A partir do momento que os administradores das empresas obtêm as respostas para essas questões eles poderão desenvolver produtos que melhor atendam aos seus clientes, estudar campanhas de marketing, conhecer o perfil de seus clientes.

O objetivo do sistema é classificar e agrupar os dados da empresa de forma que em um mesmo gráfico seja possível traçar um perfil preciso e rápido sobre sua carteira de clientes.

O SAD foi desenvolvido usando o framework JSF (Java ServerFaces) e JPA como exibido na Figura 1. Analisando essa figura pode-se observar que a coluna banco de dados, possui dois bancos de dados distintos, o banco de dados do sistema SAD e o banco de dados da empresa que terão os dados classificados e agrupados identificado na figura por “BD externo”.

A metodologia de teste foi aplicada no SAD visando facilitar a documentação do teste e melhorar a qualidade do software.

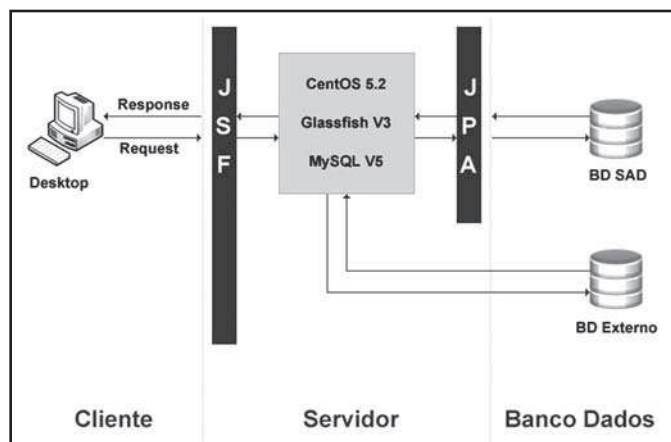


Figura 1. Arquitetura do sistema SAD

Tendo como base o documento de requisitos o sistema foi desenvolvido no padrão MVC (Model-View-Controller). A Figura 2 demonstra a alteração dos dados cadastrais do usuário no sistema SAD com o padrão MVC.

A escolha desse modelo MVC foi desvincular a interface com usuário, regras de negócio e o acesso ao banco de dados.

V. Aplicação da norma IEEE 829 no estudo de caso

Essa seção apresenta a aplicação da norma IEEE 829 no estudo de caso do sistema SAD. Devido ao fato desse sistema ter baixa complexidade alguns documentos foram compactados e outros agrupados, visando simplificar a implementação e reduzir os custos. Dessa forma os seguintes documentos são criados: plano de teste, casos de teste, relatório de incidente e relatório resumo de teste.

A. Plano de Teste

As informações presentes nesse plano de teste são: nome do projeto, nome das pessoas que participarão dos testes e suas respectivas responsabilidades, cronograma com datas e

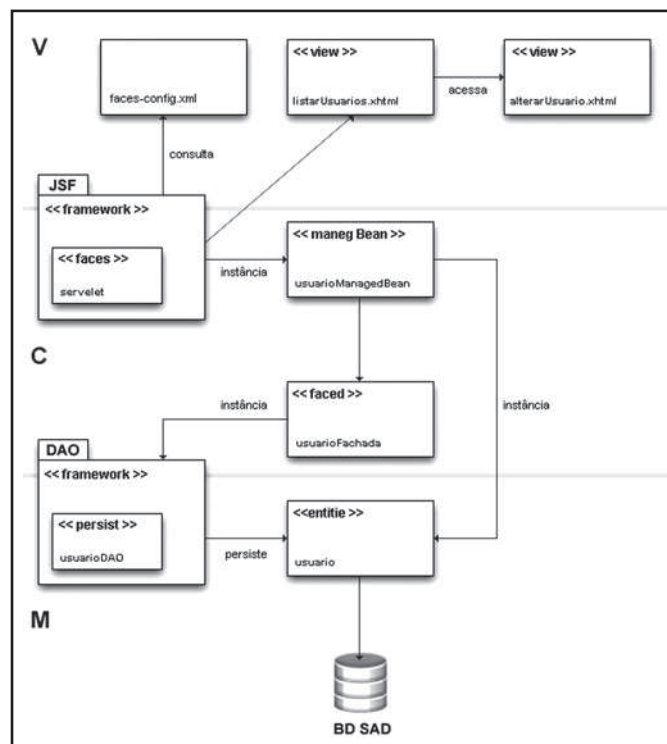


Figura 2. Arquitetura MVC do sistema SAD

horários para os testes, critérios para considerar o teste como finalizado, funcionalidades e módulos que serão testados, equipamentos e/ou softwares necessários, forma de escalar problemas - através de email direto aos envolvidos de cada área ou aos superiores acima e nome do responsável em decidir o término do teste.

A Tabela 1 mostra um exemplo do plano de teste baseado no estudo de caso SAD com todas as informações escritas no parágrafo anterior.

Esse documento será escrito apenas uma vez no projeto de teste e sempre ao iniciar o projeto de planejamento de teste.

B. Casos de Teste

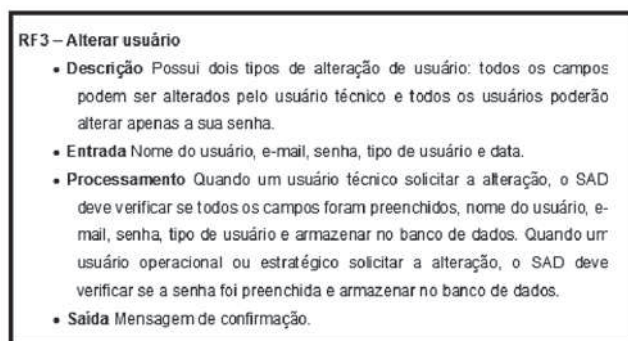
Visando facilitar a criação do documento na etapa de Especificação, foi feita a união dos documentos da Especificação do Projeto de Teste, Especificação dos Casos de Teste e Especificação dos Procedimentos de Teste.

As informações neste documento são: nome do projeto, número identificador do teste que deverá ser único, módulo ou rotina a ser testada, descrição sucinta do teste, roteiro de teste contendo a descrição do teste e passos a serem seguidos, resultado esperado, resultado final do teste obtido pelo desenvolvedor constando data e nome de quem testou e resultado final do teste obtido pelo usuário constando data e nome de quem testou.

Os casos de testes criados no sistema SAD analisam as informações desde o documento de requisito até a implementação do sistema. Desse modo, é possível verificar se as regras de negócios requisitadas estão implementadas.

Tabela 1. Plano de teste do sistema *SAD*

Plano de Teste
Nome do Projeto: Sistema de Apoio a Tomada de Decisão
Pessoas Envolvidas / Responsabilidade
Usuário2 – Criação de casos de testes e execução dos testes
Usuário1 – Criação de casos de testes e execução dos testes
Funcionalidades ou Módulos
Cadastro de cliente, classificação dos dados e relatórios
Equipamentos / Softwares
O sistema deve funcionar em um servidor Web com acesso via browser em desktop e dispositivos móveis.
Cronograma
Data de Início e Fim do Projeto: 01/08/2009 – 01/10/2010
Data de Início e Fim do Teste: 01/02/2010 a 01/10/2010
Local dos Testes:
O sistema não possui um local específico para testes. Os mesmos serão executados nas máquinas das pessoas envolvidas em locais aleatórios
Critérios para considerar o teste finalizado
O teste será considerado como finalizado ao preencher a coluna resultado do teste do documento de casos de teste como “Executado com sucesso”
Observações
O relatório de incidente será enviado para todos os desenvolvedores por e-mail assim que alguma alteração tenha sido feita. Serão criados os casos de testes, os relatórios de incidentes e o relatório resumo de teste

Figura 3. Fragmento Documento de Requisito do sistema *SAD*

Sendo assim, na Figura 3 é exibido um fragmento do documento de requisito que demonstra a alteração de dados cadastrais dos usuários. Nesse documento constam algumas validações de campos, além dos campos que poderão ser alterados.

Após a criação da arquitetura do sistema, o documento de requisito foi novamente utilizado para desenvolver o layout da tela, Figura 4, examinando se todos os campos do documento estão representado na tela, além das regras de negócios que devem estar implementadas.

Figura 4. Alteração de dados dos usuários – Sistema *SAD*

Durante todo o processo de desenvolvimento são listados e armazenados os casos de testes que deverão ser analisados posteriormente a procura de inconsistência de dados, erro de codificação do sistema, erro ao implementar as regras de negócio, como é exemplificado na Tabela 2.

Como podemos observar na tabela de casos de teste no módulo de Alteração de Usuário, ID (identificação) 1, o desenvolvedor afirmou que a alteração tinha sido executada com sucesso, mas durante o teste foi encontrado um erro, como é verificado na coluna resultado de teste no dia 25/02/2010 pelo Usuário1. Após essa detecção de erro o desenvolvedor deverá corrigir e adicionar uma linha no relatório de incidente, Tabela

3, com o mesmo número de identificação (ID) encontrado no caso de teste.

C. Relatório de Incidente

O relatório de incidentes é a união dos documentos Diários de Teste e Relatório de Incidente, sendo ilustrado pela Tabela 3 baseado no estudo de caso.

As informações contidas neste documento são: número do caso de teste, status, nome da pessoa responsável pela correção, prioridade para correção do erro pelo desenvolvedor, descrição detalhada do erro podendo conter a mensagem de erro exibida na tela e nome e data de quem fez a correção.

O desenvolvedor ao verificar que foi encontrado um erro na Tabela 2, caso de teste, irá corrigir o erro e adicionar uma linha na Tabela 3. Como é mostrado na Tabela 3, o campo ID 1, foi corrigido pelo Usuário2 no dia 01/03/2010. Esse erro foi encontrado pelo Usuário1 no dia 25/02/2010 como pode ser visto na Tabela 2, id 1. É importante lembrar que o número de identificação da Tabela 2 é correspondente ao ID da Tabela 3.

Pode-se observar que Tabela 3 possui uma coluna que descreve a prioridade de correção do erro, podendo variar entre alta e baixa.

Após a correção do erro e a inserção no relatório de incidente, o módulo é novamente analisado a procura de erros. E consequentemente, a coluna resultado de teste da tabela de casos de teste é alterada com a informação que o teste foi executado com sucesso ou com a data do teste e a descrição do erro.

D. Relatório Resumo de Teste

Esse é o documento final a ser preenchido pela equipe de teste com as seguintes informações: nome do projeto; data do início e fim do teste; descrição detalhada de como foi o teste; pessoas envolvidas; os principais números dos testes, como por exemplo, número de casos de teste criados antes e durante a execução dos testes, e o número de execuções com sucesso; tipos de status: completado com sucesso, completado com restrições, e não completado; percentual de casos de teste executados; percentual de casos com sucesso e erros e percentual de casos de testes corrigidos pelo desenvolvedor.

A Tabela 4 apresenta o relatório resumo de teste baseado no estudo de caso.

VI. TRABALHOS RELACIONADOS

O trabalho realizado por Oliveira et al. (2008) foi ressaltado a importância da documentação como forma de padronizar e organizar a execução de teste, além de mostrar o método MITs (The Most Important Tests), o qual identifica os riscos e falhas no sistema com o objetivo de testar apenas o que é mais importante.

O estudo feito por Souza et al. (2008) sugere uma técnica baseada em riscos (RBT - Risk-based Testing) e criação do modelo RBTPProcess para identificar as principais atividades no gerenciamento de risco e o processo de teste.

Tabela 2. Casos de testes do sistema SAD

Casos de testes						
Nome Projeto: Sistema de Apoio a Tomada de Decisão						
ID	Módulo	Descrição	Roteiro	Resultado esperado	Resultado do desenvolvedor	Resultado do teste
1	Alteração Usuário	Alterar dados cadastrais dos usuário	1) Escolher a opção Listar Usuário 2) Clicar em alterar na lista	Exibir na lista de usuários cadastrados	Usuário2 – 15/02/2010. Executado com sucesso	Usuário1 – 25/02/2010. Mostra um erro ao alterar usuário
2	Alteração Usuário	Verificar a validação dos campos	1) Escolher a opção Inserir Usuário. 2) Deixar os campos nome, e-mail e senha em branco	Mostrar a mensagem “Campo de preenchimento obrigatório” para os campos nome, senha e e-mail.	Usuário2 – 15/02/2010. Executado com sucesso	Usuário1 – 25/02/2010. Executado com sucesso
3	Alteração Usuário	Verificar a validação dos campos	1) Escolher a opção Inserir Usuário. 2) Inserir uma senha com menos de 2 caracteres ou mais de 32 caracteres	Mostrar a mensagem “O valor fornecido para este campo não pode ter menos que 6 e mais que 32 caracteres” para o campo senha	Usuário2 – 15/02/2010. Executado com sucesso	Usuário1 – 25/02/2010. O campo senha aceita menos de 6 caracteres
4	Alteração Usuário	Verificar a validação dos campos	1) Escolher a opção Inserir Usuário. 2) Digitar um e-mail inválido	Mostrar a mensagem “O e-mail informado não é válido”	Usuário2 – 15/02/2010. Executado com sucesso	Usuário1 – 25/02/2010. Executado com sucesso

Tabela 3. Relatório de incidente de teste do sistema SAD

Relatório de Incidente					
Nome Projeto: Sistema de Apoio a Tomada de Decisão					
ID	Status	Responsável pela correção	Prioridade de correção	Descrição do erro	Data e Nome de quem corrigiu
1	Pronto para testar novamente	Usuário2	Alta	Mostra um erro ao alterar usuário	01/03/2010 –Usuário2
3	Pronto para testar novamente	Usuário2	Alta	O campo senha aceita menos de 6 caracteres	14/03/2010-Usuário2
5	Pronto para testar novamente	Usuário1	Baixa	No cabeçalho o nome do relatório está errado	10/03/2010 - Usuário1
6	Aberto	Usuário2	Alta	Erro ao classificar os dados	

Tabela 4. Relatório de resumo de teste do sistema SAD

Relatório Resumo de Teste	
Nome Projeto:	Sistema de Apoio a Tomada de Decisão
Data início teste:	01/02/2010
Data fim teste:	01/10/2010
Descrição teste	
Com a criação dos casos de teste previamente facilitou a detecção e a correção dos erros. O relatório de incidência de teste demonstrou de forma clara a correção do programador e o momento certo para o retorno para equipe de teste, na qual foi realizado o teste novamente.	
Pessoas envolvidas	
Usuário1, Usuario2, Usuário3	
Números do teste	
Casos de testes criados antes do teste	30
Casos de testes criados durante o teste	3
Casos de testes executados	33
Casos de teste com sucesso	20
Casos de teste com erro	13
Casos de testes enviados para correção	13
Percentual	
Casos de testes executados	100%
Casos de testes executados com sucesso	66,67%
Casos de testes com incidência de erro	43,33
Casos de testes corrigidos pelo desenvolvedor	100%

Essa técnica utiliza o modelo de documentação de teste proposta nesse trabalho.

O trabalho de Crespo et al. (2004) - Centro de Pesquisas Renato Archer - descreve a criação de uma metodologia de teste para atender as necessidades das empresas desenvolvedoras de softwares que pretendem aprimorar a atividade de teste. Assim esta metodologia foi dividida em três componentes: Treinamento, Processo de Teste e Suporte para Geração de Documentos. O terceiro componente utiliza a documentação de teste baseado na norma IEEE 829.

Sendo assim, podemos constatar que, independentemente da técnica de como listar os casos de testes, todos os documentos pesquisados utilizam a norma IEEE 829 para gerar o modelo de documento.

VII. CONCLUSÃO

A cada dia encontramos clientes que exigem produtos de melhor qualidade. Sendo assim, foi apresentada um estudo de caso de aplicação de uma metodologia de teste baseada na norma IEEE 829 objetivando clientes mais satisfeitos devido a redução dos erros, com a aplicação no sistema de tomada de decisão (SAD).

Essa metodologia, para a aplicação no SAD, propôs a geração de quatro documentos fáceis de implementar. Desses, dois são escritos apenas uma vez por projeto - plano de teste escrito no início do projeto e relatório de resumo de teste escrito no final. Enquanto que os demais, casos de teste e o relatório incidente, têm manutenção maior, pois precisam ser alterados a cada realização do teste.

A utilização dos documentos de teste facilitou a padronização e organização da execução do processo, além de auxiliar no trabalho do testador. Sendo assim, o relatório de casos de testes pode ser utilizado como material de referência pelas pessoas responsáveis pelo teste, enquanto que os relatórios de incidente auxiliam os programadores na correção dos erros encontrados.

Com isso, podemos constatar que ao aplicar essa metodologia, o teste é planejado desde o início do projeto e aplicado

durante toda a sua duração e não apenas no final. Desse modo, o tempo gasto em refazer pequenas partes do projeto é menor do que se tivesse que refazer uma grande parte no final.

Mesmo assim, algumas empresas ainda demonstram resistência em organizar uma equipe de teste e documentar esse processo. Porém, isso está mudando, pois a satisfação dos clientes esta cada vez mais em evidência.

REFERÊNCIAS

- Ali Sh., Briand L., Rehman M.J., Asghar H., Zafar Z., Nadeem A., “A State-based Approach to Integration Testing for Object-Oriented Programs”, *Journal of Information and Software Technology* (Elsevier), 49 (2007).
- Bruneli M. V. Q. A utilização de uma metodologia de teste no processo da melhoria da qualidade do software”, *Mestrado Profissional UNICAMP, Campinas*, 2006.
- Crespo, A. N.; Silva, O. J.; Borges, C. A.; Salviano, C. F., Argolo, M. T., Jino, M.. Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo. III Simpósio Brasileiro de Qualidade de Software (SBQS), Brasília, 2004.
- IEEE Computer Society; IEEE Std 829: Standard for Software Test Documentation; September, 1998.
- Myers G. J.. *The Art of Software Testing*. Wiley, New York, 2nd edition, 2004.
- Oliveira, R. S.; Rosso R. S. U. J.; Murakami, E. Qtest: Uma Metodologia de Teste baseada no método MITs e na norma IEEE 829, III EBTS – Encontro Brasileiro Teste Software, 2008.
- Pressman R. S. *Engenharia de Software*. McGraw-Hill, sexta edição, 2006.
- Souza, E.; Gusmão C.; Rocha H., RBTPProcess - Proposta de Modelo de Processo de Teste de Software baseado em Riscos, III EBTS – Encontro Brasileiro Teste Software, 2008.