



## Suporte automatizado à rastreabilidade em um processo de teste de software baseado em documentação<sup>†</sup>

Jorge Luiz da Cruz<sup>1,3</sup>, Mario Jino<sup>2</sup>, Adalberto Nobiato Crespo<sup>1,4</sup>, Miguel Argollo<sup>1,3</sup>

<sup>1</sup>Centro de Pesquisas Renato Archer (CenPRA) – Campinas – SP – Brasil

<sup>2</sup>UNICAMP – Campinas – SP – Brasil

<sup>3</sup>Faculdades Hoyer – Hortolândia – SP – Brasil

<sup>4</sup>Universidade São Francisco – Itatiba – SP – Brasil

{jorge.cruz, adalberto.crespo, miguel.argollo}@cenpra.gov.br,  
jino@dca.fee.unicamp.br

**Abstract.** *Systematic software testing is not an easy activity. To achieve effectivity and efficacy in testing, in addition to product evaluation aspects, test must be well documented, all the involved artifacts must be up to date and consistent and the associations among artifacts must be traceable. This paper discusses the advantages of traceability for a software test process based on documentation and the need for automated support to traceability of test artifacts.*

**Resumo.** *Testar sistematicamente um software não é uma tarefa trivial. Para que o teste seja efetivo e eficaz, além dos aspectos relacionados à avaliação do produto, ele deve ser bem documentado, todos os artefatos envolvidos devem estar atualizados e consistentes e as associações entre esses artefatos devem ser rastreáveis. Este artigo apresenta as vantagens da rastreabilidade para um processo de teste de software baseado em documentação e a necessidade de suporte automatizado à rastreabilidade de artefatos de teste.*

### 1. Introdução

O teste de software é reconhecido como um processo fundamental para obtenção de software de boa qualidade [Chrissis 2003], [ISO/IEC 2003]. Entretanto, testar sistematicamente um software não é uma atividade trivial, pois envolve o gerenciamento de um conjunto de atividades complexas e de longa duração. Esse gerenciamento deve ser baseado em informações atualizadas e precisas: os documentos que registram ou fornecem essas informações devem estar atualizados e ser facilmente rastreáveis [Beizer 1990], [Craig e Jaskiel 2002], [Kaner 2001], [Pressman 2002].

Informações desatualizadas ou inconsistentes levam a tomada de decisões incorretas, implicando, entre outras coisas, em testes incompletos ou inapropriados no software [Pressman 2002], [Sommerville 2003]. Entretanto, dispor de informações atualizadas e consistentes a qualquer momento do processo de teste é difícil e dispendioso, já que, entre outras coisas, é necessário estabelecer e controlar diversos relacionamentos de dependência entre os vários elementos que representam as fontes

---

<sup>†</sup> Este trabalho contou com suporte financeiro parcial da HP-Brasil.



dessas informações [Domges e Pohl 1998], [Gotel e Finkelstein 1995], [Ramesh e Jarke 2001], [Cleland-Huang 2003].

Considerando o problema exposto, este artigo apresenta o conceito de rastreabilidade, seu emprego para manter o relacionamento entre os diversos artefatos do processo de desenvolvimento de forma geral e do processo de teste de forma específica, a importância da rastreabilidade na obtenção de um processo de teste de boa qualidade, e a necessidade de suporte automatizado. O suporte automatizado é fornecido por Prometeu, uma ferramenta de apoio ao rastreamento entre artefatos e informações relacionados a um processo de teste baseado em documentação. Entendem-se como 'artefatos' todas as informações relevantes que são produzidas, registradas e, eventualmente, modificadas como parte de um processo de software.

O restante deste artigo está organizado da seguinte forma: na Seção 2 define-se o que é rastreabilidade e como ela pode ser modelada; na Seção 3 apresentam-se alguns benefícios possíveis da rastreabilidade para o teste de software; na Seção 4 apresenta-se a rastreabilidade no contexto de um processo de teste de software baseado em documentação; na Seção 5 destaca-se a necessidade de suporte automatizado à rastreabilidade e apresenta-se brevemente a ferramenta Prometeu, com exemplos de algumas de suas funcionalidades, e na Seção 6 são apresentadas as conclusões deste artigo.

## 2. Rastreabilidade

O processo de desenvolvimento de software é bastante complexo e dinâmico, exigindo a produção de uma quantidade considerável de documentos. A cada etapa do desenvolvimento novos documentos são criados e documentos existentes podem ser alterados diversas vezes. Normalmente, uma modificação em um documento pode causar impacto em vários outros.

Manter a rastreabilidade entre os diversos documentos de um projeto é considerado imprescindível para obtenção de um produto de software de qualidade [Cleland-Huang 2003], [Ramesh e Jarke 2001], [Gotel e Finkelstein 1995]. Entre outras coisas, a rastreabilidade auxilia na avaliação do impacto no projeto que eventuais modificações em artefatos possam causar e também contribui para verificar se todas as partes envolvidas no projeto têm suas necessidades satisfeitas. Entende-se como 'partes envolvidas' os clientes, seus possíveis representantes, órgãos reguladores, desenvolvedores, testadores, enfim todos os envolvidos no projeto.

Algumas definições existentes para rastreabilidade são: a) a habilidade de descrever e acompanhar um requisito ao longo do seu ciclo de vida, tanto na direção das suas fontes (pessoas, idéias, artefatos) como na direção dos artefatos que o sucedem [Gotel e Finkelstein 1995], e b) a capacidade de cada elemento de um processo de software em justificar sua existência, ou o grau de relacionamento que pode ser estabelecido entre dois ou mais artefatos de um processo de software, especialmente artefatos que, entre si, possuam relacionamento do tipo predecessor-sucessor ou mestre-subordinado [IEEE 1991].

A rastreabilidade é complexa de implantar e cara de ser mantida, já que abrange o controle de um grande número de associações entre artefatos, de diversos tipos de informações e de várias partes envolvidas [Cleland-Huang 2003], [Ramesh e Jarke 2001], [Gotel e Finkelstein 1995].



## 2.1. Modelagem da rastreabilidade

Modelar a rastreabilidade em um projeto envolve considerar quais associações e quais informações de rastreabilidade devem ser controladas.

### 2.1.1. Associações (*links*) de rastreabilidade

Uma associação de rastreabilidade representa um relacionamento existente entre dois artefatos, com a definição de uma associação explícita (interconexão, parentesco, dependência) entre esses dois artefatos [Cleland-Huang 2003]. Os aspectos a serem considerados sobre as associações são seu tipo e orientação e sua semântica.

#### Tipo e orientação

Existem relacionamentos entre artefatos de tipos distintos e entre artefatos do mesmo tipo. Além disso, para cada artefato podem existir relacionamentos entre as partes que o compõem.

Para registrar a relação existente entre artefatos de tipos distintos, utiliza-se a rastreabilidade horizontal. Já a rastreabilidade vertical é utilizada para registrar as relações existentes entre artefatos do mesmo tipo ou as relações entre as partes que compõem um mesmo artefato [Pfleeger e Bohner 1990]. Na Figura 1, as setas tracejadas representam a rastreabilidade horizontal. Já as setas sólidas no documento “Especificação de Projeto de Teste”, entre as duas seções do documento e entre os casos de teste da seção ‘Casos de teste’, representam a rastreabilidade vertical.

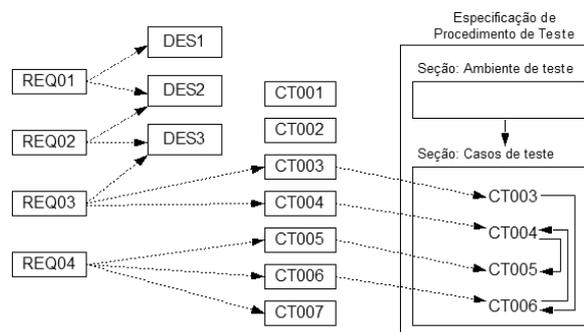


Figura 1 – Rastreabilidade horizontal e vertical entre artefatos.

Também é importante registrar a precedência entre artefatos. Uma associação do tipo *forward* liga um artefato a artefatos que o satisfazem ou implementam – geralmente artefatos que são criados posteriormente. Já uma associação do tipo *backward* liga um artefato a artefatos criados geralmente em fases anteriores [Gotel e Finkelstein 1995]. A Figura 2 ilustra a rastreabilidade *backward* e *forward* entre três artefatos.

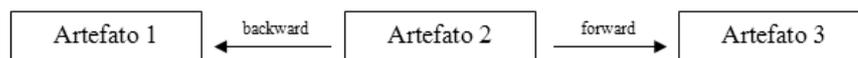


Figura 2 - Relacionamentos do tipo *backward* e *forward*.

Os mesmos conceitos são apresentados por outros autores como *traced-to* e *traced-from* [Leffingwell e Widrig 2000]. Uma associação do tipo *traced-to* indica, geralmente, quais artefatos criados em fases posteriores implementam ou detalham o artefato que originou o *link*. Na Figura 3, Artefato1 é implementado ou detalhado por Artefato2. Uma associação do tipo *traced-from* indica uma fonte, ou origem, de um artefato. Na Figura 3, a fonte, ou origem, de Artefato2 é Artefato1.

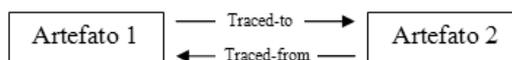


Figura 3 - Relacionamentos do tipo *traced-to* e *traced-from*.

### Semântica

Um *link* de rastreabilidade, para detalhar suas características, pode ser caracterizado por um ou mais atributos [Cleland-Huang 2003], [Leffingwell e Widrig 2000], [Ramesh e Jarke 2001]. Na Figura 4, apresenta-se uma situação na qual um artefato do tipo “requisito” está associado a um artefato do tipo “caso de teste” através de uma associação do tipo “*traced-to*”, ou “*forward*”. Para detalhar as características dessa associação, entre outros atributos, ela pode receber o rótulo “testado por”.

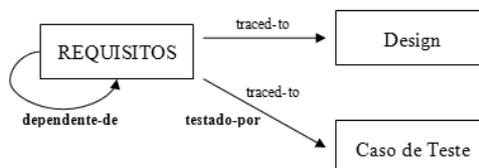


Figura 4 – Semântica dos *links* de rastreabilidade.

#### 2.1.2. Informações de rastreabilidade

Informações de rastreabilidade são informações contidas no próprio artefato. Elas podem ser atributos adicionais a serem rastreados, como justificativas e observações, além de atributos que permitem associações com outros artefatos ou atributos que façam referência às partes envolvidas.

Em um projeto deve-se considerar quais informações necessitam serem rastreadas, quais são as partes envolvidas e como essas partes podem influenciar o processo de software.

#### Informações a serem rastreadas

As informações que devem ser rastreadas podem ser organizadas da seguinte maneira [Ramesh e Jarke 2001]: a) o quê – refere-se a qual tipo de informação, incluindo seus atributos ou características. Exemplo: atributos (observações, justificativas) e artefatos (especificação de caso de teste); b) quem – agentes envolvidos no projeto e que desempenham diferentes papéis na criação, manutenção e uso das informações a serem rastreadas. Exemplo: gerente de projeto, analista de sistemas, representante do cliente; c) onde – quais as fontes (origem ou procedência) que documentam as informações de rastreabilidade que originaram o artefato, representando o elo entre o artefato e as razões básicas para sua proposição. Exemplo: pessoas ou documentos; d) como – qual o formato de representação das informações. Exemplo: somente texto? Combinação de texto e gráficos?; e) por que – registro do motivo da criação, modificação ou evolução de um determinado artefato; f) quando – quando uma determinada informação foi registrada ou modificada.

#### Partes envolvidas

Diferentes tipos de participantes de um processo de software possuem diferentes necessidades e expectativas, baseados em pontos de vista diferentes. Essas diferenças representam informações que devem ser rastreadas, pois exercem forte impacto sobre o gerenciamento de um projeto, podendo determinar seu sucesso ou falha [Gotel e Finkelstein 1995].



### 2.1.3. Usuários da rastreabilidade

O sucesso da implantação da rastreabilidade em uma organização é fortemente influenciado pela cultura desta organização. Os usuários e os interessados pelas informações de rastreabilidade podem ser divididos em dois tipos [Ramesh e Jarke 2001]: *low-end* e *high-end*.

Os usuários *low-end* percebem a rastreabilidade simplesmente como obrigação imposta pelos patrocinadores do projeto. Seu nível de experiência com rastreabilidade não passa de dois anos. Esses usuários percebem a aplicação da rastreabilidade em sistemas de até mil requisitos, onde seu uso se restringe a decomposição e alocação dos requisitos a outros artefatos e como apoio a verificação de conformidade e controle de modificações. Os usuários *high-end* percebem a rastreabilidade como um componente importante para o desenvolvimento de um sistema de qualidade. Seu nível de experiência com rastreabilidade ultrapassa os dois anos e eles percebem a rastreabilidade como importante apoio que aumenta a probabilidade de produzir um sistema que satisfaça todos os requisitos do cliente e que seja de fácil manutenção. Além disso, para este tipo de usuário, a rastreabilidade deve fazer parte de todo o ciclo de vida de um software, ao longo das dimensões do produto e do processo.

Pelo apresentado até este ponto, pode-se considerar que rastrear as dependências existentes entre os artefatos, entre os artefatos e suas fontes, entre as informações de rastreabilidade existentes em cada artefato, além de rastrear as contribuições das partes envolvidas, envolve gerenciar uma grande diversidade de informações e de relacionamentos entre artefatos. Entretanto, sendo bem planejada e implantada, a rastreabilidade desempenha papel importante na manutenção da integridade (sincronia, atualidade e correção) das informações utilizadas ao longo de um processo de software, fornecendo valioso auxílio ao seu planejamento, execução e acompanhamento.

## 3. Benefícios da rastreabilidade para o teste de software

Algumas contribuições da rastreabilidade para o teste de software são descritas a seguir.

a) Teste baseado nos requisitos: nos níveis de Teste de Sistema e de Teste de Aceitação, é importante garantir que o software seja validado de acordo com os requisitos especificados, com cada requisito sendo exercitado por um conjunto de casos de teste [Beizer 1990], [Craig e Jaskiel 2002]. A rastreabilidade entre requisitos e casos de teste permite, entre outras coisas, visualizar a quantidade de casos de teste alocada a um requisito e se algum requisito ainda não possui casos de teste associados.

b) Atualidade e consistência das informações: a documentação deve ser a mais completa possível, baseada em informações atualizadas e consistentes. Caso contrário, os testes, as correções e as atualizações necessárias poderão ser incompletas ou inapropriadas [Phillips 1998], [Hetzel 1987], [Gelperin 1982]. Quando qualquer um dos artefatos relacionados à documentação for alterado, deve ser possível rastrear rapidamente quais outros artefatos foram afetados e que devem ser revisados e, eventualmente, atualizados. As associações estabelecidas entre os artefatos evitam que seja esquecido qualquer documento. Por exemplo, quando um procedimento de teste for modificado, todos os artefatos associados devem ser revisados e, se necessário, atualizados.



c) Teste preventivo: sabe-se que o custo da remoção de um defeito aumenta exponencialmente à medida que ele é propagado das fases iniciais até as fases finais de um processo de software. Assim, prevenir a inserção e propagação dos defeitos, ao invés de meramente erradicar esses defeitos após sua manifestação, é fator de economia de recursos e de tempo no processo de teste. Exemplo: a associação direta entre casos de teste e os requisitos especificados faz com que os requisitos sejam revisados como consequência do desenho dos casos de teste criados para validá-los, tornando-os testáveis e reduzindo a ocorrência de requisitos vagos, mal especificados, que geralmente se transformam em defeitos [Craig e Jaskiel 2002].

d) Análise do impacto das modificações: o teste de um software é um processo caro, sendo muito importante compreender o impacto que eventuais modificações em quaisquer artefatos de teste podem causar no projeto. A rastreabilidade permite visualizar rapidamente quais artefatos, e quais profissionais, estão relacionados no momento em que uma modificação é solicitada, tornando possível avaliar o impacto no processo de teste com maior precisão. Uma situação deste tipo é uma modificação em um requisito. Essa modificação pode implicar em modificações em elementos da arquitetura<sup>†</sup>, em um grande número de casos de teste, além de ocasionar mudanças nos documentos de planejamento do teste.

e) Implementação de modificações: ao serem executadas correções e melhorias, deve-se assegurar que o sistema continue funcionando corretamente. As correções devem ser testadas para garantir que o defeito detectado foi corrigido e que novos defeitos não tenham sido introduzidos no software; as melhorias devem ser testadas para garantir que não possuem defeitos e que, também, novos defeitos não foram introduzidos no software. Em ambos os casos, a rastreabilidade facilita isolar rapidamente os artefatos associados, além das pessoas responsáveis pelos artefatos, bem como suposições e idéias que justificam a existência de cada artefato. No caso de correções, facilita encontrar possíveis causas da falha, como não-conformidades com as especificações, falhas na comunicação com o cliente e erros na arquitetura. No caso de melhorias, facilita identificar os impactos que modificações ou a inserção de novos requisitos trarão aos artefatos já existentes: novos artefatos serão necessários? Artefatos existentes poderão ser reutilizados? Quantos, e quais, deverão ser modificados?

f) Gerenciamento do processo de teste: o planejamento, projeto e acompanhamento do teste devem se basear em informações atualizadas, consistentes e precisas, aumentando a qualidade do processo. A rastreabilidade auxilia na obtenção de melhores estimativas de escopo, prazo e custo e melhora no mapeamento e satisfação das diferentes necessidades e expectativas das partes envolvidas. Alguns exemplos de informações que podem ser obtidas são: (1) dos casos de teste projetados, quantos já foram executados e quantos ainda faltam, (2) dos casos de teste já executados, em quais o sistema passou e em quais o sistema falhou, (3) o número de defeitos descobertos pelo cliente e descobertos pela equipe de teste, (4) em quais fases do desenvolvimento os defeitos mais ocorrem, (5) quais justificativas e suposições fundamentam a criação dos artefatos, (6) informações para manutenção da integridade dos artefatos.

---

<sup>†</sup> O termo ‘elementos da arquitetura’ será usado para referenciar artefatos da fase de *Design* de um processo de software.



#### **4. Rastreabilidade em um processo de teste baseado em documentação**

Documentar o processo de teste é um aspecto crítico para seu sucesso, sendo necessário um projeto detalhado e o mais completo possível. Uma documentação criteriosa, bem estruturada, baseada em informações atualizadas e consistentes, melhora a visibilidade das várias fases do teste, contribui efetivamente para o controle gerencial e técnico da sua execução e pode ser utilizada como mecanismo de comunicação entre as partes interessadas e envolvidas [Gelperin 1982], [Hetzel 1987], [Phillips 1998].

Em todos os trabalhos pesquisados a rastreabilidade é tratada com foco nos requisitos. Não foram encontradas referências que enfocassem diretamente a rastreabilidade no processo de teste de software: nem com ferramentas, nem com modelos.

##### **4.1. A Norma IEEE-Std-829 - um modelo para documentação do processo de teste**

Mesmo reconhecendo a necessidade de documentação, é difícil definir quais informações devem ser registradas e controladas. Esta questão pode ser resolvida com a utilização da norma IEEE-829 pode ser empregada como referência e guia para a documentação de um processo de teste de software [IEEE 1998]. A norma descreve um conjunto de oito documentos para documentação de um processo de teste de um software. Os documentos preconizados pela norma abordam as atividades de planejamento, projeto e registro do teste, e serão apresentados a seguir.

O documento seguinte aborda a tarefa de planejamento do teste.

Plano de Teste – apresenta o planejamento para execução do teste, incluindo a abrangência, abordagem, recursos e cronograma das atividades de teste. Identifica os itens e as funcionalidades a serem testados, as tarefas a serem realizadas e os riscos associados com a atividade de teste.

Os três documentos seguintes abordam a tarefa de projeto dos testes.

Especificação de Projeto de Teste – refina a abordagem apresentada no Plano de Teste e identifica as funcionalidades e características a serem testadas pelo projeto e por seus testes associados. Este documento também identifica os casos e os procedimentos de teste, se existirem, e apresenta os critérios de aprovação.

Especificação de Caso de Teste – define um caso de teste, incluindo dados de entrada, resultados esperados, ações e condições gerais para a execução do teste.

Especificação de Procedimento de Teste – especifica os passos para executar um determinado conjunto de casos de teste.

Os quatro documentos seguintes abordam o registro das atividades de teste.

Diário de Teste - apresenta registros cronológicos dos detalhes relevantes relacionados com a execução dos testes.

Relatório de Incidente de Teste - documenta qualquer evento que ocorra durante a atividade de teste e que requer análise posterior.

Relatório-Resumo de Teste – apresenta de forma resumida os resultados das atividades de teste associadas com uma ou mais especificações de projeto de teste e provê avaliações baseadas nesses resultados.



Relatório de Encaminhamento de Item de Teste – identifica os itens encaminhados para teste no caso de existirem equipes distintas para tarefas de desenvolvimento e de teste.

A Figura 5 ilustra esses oito documentos e os relacionamentos entre eles.

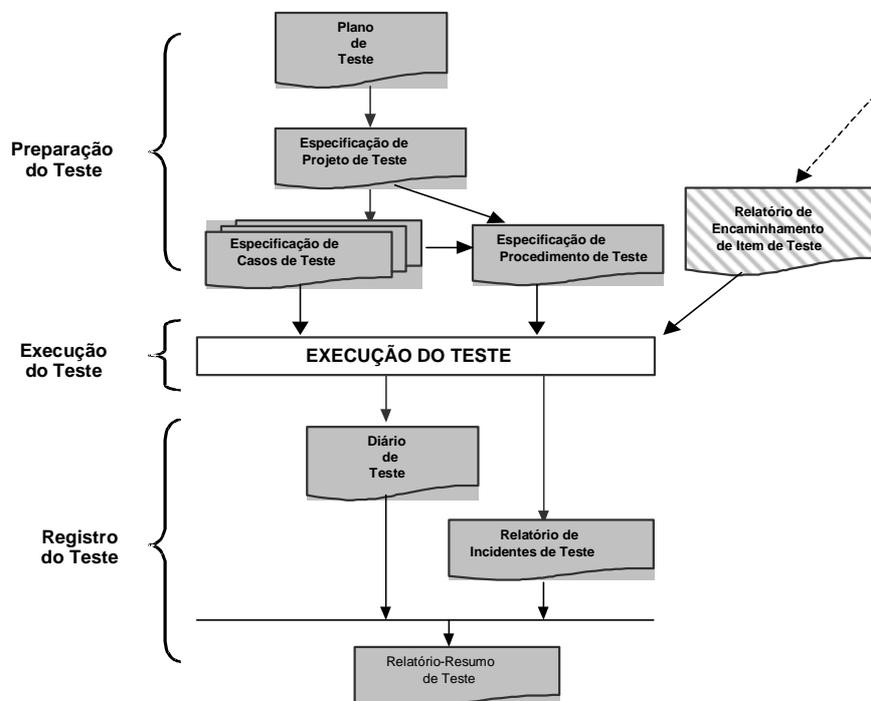


Figura 5 - Relacionamento entre os documentos de teste.

Embora a utilização da norma seja criticada [Kaner 2001], a experiência no trabalho com pequenas empresas demonstra que ela pode ser usada com sucesso [Crespo 2002], [Crespo 2004]. A norma não pode ser encarada, nem aplicada, sem um estudo prévio: ela define o formato e os itens de informação necessários para cada tipo de documento, sem determinar quais tipos de documentos deverão efetivamente ser utilizados. Os oito documentos podem ser utilizados como proposto pela norma ou com seus itens de informação combinados em um número menor de documentos. Quantos, quais e em que arranjo usar esses documentos depende de cada empresa em particular, depende do processo e do escopo do projeto. Um arranjo possível é Plano de Teste, Especificação de Procedimento de Teste e Relatório-Resumo de Teste. Estes três documentos registram o esforço do teste que se pretende realizar, os testes realizados e os incidentes ocorridos [Hetzl 1987], [Phillips 1998].

#### 4.2. Definindo o que deve ser rastreado

Implementar rastreabilidade envolve manipular um volume muito grande de informações. Negligenciar ou capturar informações de rastreabilidade em quantidade insuficiente impede a obtenção dos benefícios da rastreabilidade. Por outro lado, capturar e gerenciar todas as informações possíveis de acordo com os diversos modelos disponíveis, não é nem desejável, nem factível em termos de custo e tempo, podendo levar a uma quantidade muito grande de informações desestruturadas e sem utilidade [Domges e Pohl 1998].



Em um processo de teste baseado em documentação, a necessidade de documentação direciona a definição de vários artefatos que devem ser rastreados: além daqueles relacionados à documentação, deve ser rastreado um conjunto mínimo de tipos de artefatos, como requisitos e elementos da arquitetura, e informações que servirão de apoio à documentação do teste.

Para o gerenciamento da grande quantidade de informações de rastreabilidade é imprescindível o devido suporte automatizado [Beizer 1990], [Cleland-Huang 2003], [Gotel e Finkelstein 1995], [Ramesh e Jarke 2001]. A seguir será apresentada a ferramenta Prometeu e os artefatos e informações por ela rastreados.

## 5. A ferramenta Prometeu

Prometeu é uma ferramenta de apoio à documentação e ao rastreamento entre artefatos e elementos de informação relacionados a um processo de teste. A ferramenta está sendo desenvolvida no contexto de uma metodologia para a introdução ou melhoria do processo de teste de software em empresas e baseada na norma IEEE-829 [IEEE 1998], [Crespo 2002], [Crespo 2004]; não existem ainda resultados de sua aplicação.

As seções seguintes descrevem a ferramenta de maneira resumida. São citados os artefatos gerenciados, e o motivo para sua utilização, a estratégia de rastreabilidade adotada e alguns dos seus recursos de suporte à documentação e à rastreabilidade.

### 5.1. Artefatos gerenciados

Os artefatos e informações controlados pela ferramenta são: a origem ou fonte de um artefato, as especificações de requisitos, as especificações de projeto, os planos de teste, as especificações de projeto de teste, as especificações de procedimento de teste, as especificações de casos de teste, os registros de incidentes, e os diários de teste.

**Origem ou fonte:** a fonte de um artefato representa informações importantes que devem ser rastreadas [Gotel e Finkelstein 1995], [Ramesh e Jarke 2001]. Uma fonte pode ser um indivíduo, um grupo de indivíduos ou outro artefato. Uma fonte fornece observações e justificativas que fundamentam a existência e evolução de um artefato, além de fornecer informações adicionais que podem ser utilizadas no projeto de outros artefatos associados ou na manutenção do software.

**Especificações de requisitos:** requisitos são artefatos importantes no planejamento do teste de software [Beizer 1990], [Craig e Jaskiel 2002], [Leffingwell e Widrig 2000], [Robertson e Robertson 1999]. Na ferramenta, os requisitos são representados, no nível mais alto, como funcionalidades. Cada funcionalidade é detalhada por um ou mais requisitos do tipo Funcional ou Não-Funcional [Cockburn 2005], [Leffingwell e Widrig 2000], [Robertson e Robertson 1999]. Sobre os requisitos são registradas informações que possam auxiliar as atividades do teste, como sua origem, prioridade do requisito para o teste, motivação, observações, o identificador do requisito. Considera-se que os requisitos podem ser armazenados tanto em arquivos separados (um arquivo para cada requisito) como em um único arquivo, sendo necessário que cada requisito possua um identificador único, que será utilizado para associá-lo a um projeto de teste e permitir que ele seja rastreado.

**Elementos da Arquitetura:** também são artefatos importantes no planejamento do teste de software [Craig e Jaskiel 2002]. Como os requisitos, considera-se que são



armazenados em arquivos externos. Na ferramenta são registrados somente informações que possam auxiliar as atividades do teste.

**Documentos do teste:** os tipos de documentos descritos neste tópico são o Plano de Teste, a Especificação de Projeto de Teste, a Especificação de Procedimento de Teste, a Especificação de Caso de Teste, o Relatório de Incidente de Teste e o Diário de Teste. Cada documento é composto por um conjunto de seções, conforme descrito na norma IEEE-829 [IEEE 1998]. Cada seção pode ser preenchida com um ou mais itens de informação, que podem ser estáticos (texto digitado pelo usuário) ou dinâmicos (campos importados de registros de artefatos controlados pela ferramenta). Itens dinâmicos estabelecem associações entre o documento e outros artefatos.

Documentos do tipo “Plano de Teste” possuem duas seções com itens dinâmicos: “Funcionalidades que Serão Testadas” e “Funcionalidades que Não Serão Testadas”. No tipo “Especificação de Projeto de Teste”, as seções com itens de informação dinâmicos, são: “Funcionalidades e Características Tratadas”, “Procedimentos Associados” e “Casos de Teste Associados”. No tipo “Especificação de Procedimento de Teste”, a seção com itens dinâmicos, é “Casos de Teste Associados”. A Figura 6 ilustra a seção “Funcionalidades que Serão Testadas” de um documento do tipo “Plano de Teste”. Podem ser vistos todos os requisitos registrados na ferramenta e que podem ser associados ao documento no momento da sua criação ou edição.

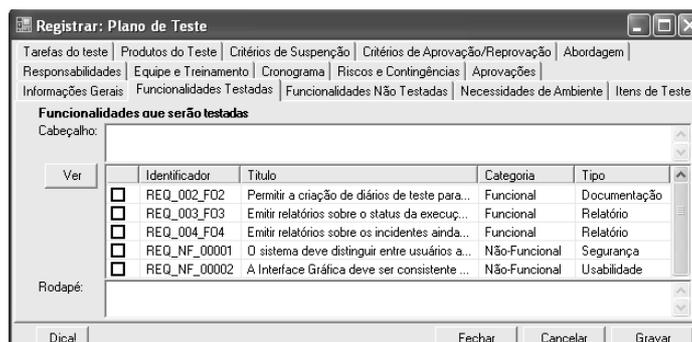


Figura 6 – Seção de um documento do tipo “Plano de Teste”.

Documentos do tipo “Casos de Teste” registram dados de entrada, resultados esperados, ações, condições gerais para sua execução e sua eventual associação com requisitos ou elementos da arquitetura. Essa associação é estabelecida a partir da seção “Artefatos Associados”, conforme ilustrado na Figura 7. Cada artefato pode ser consultado a partir da lista exibida.

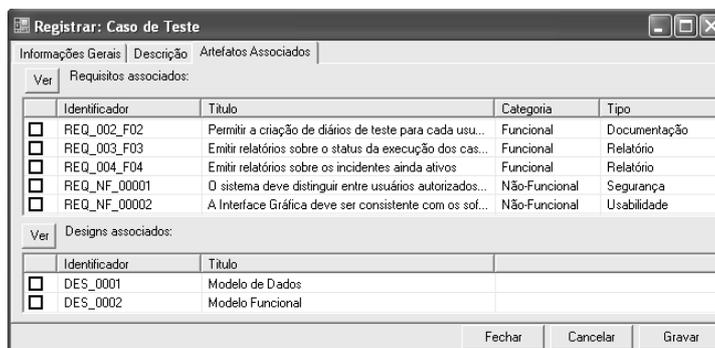


Figura 7 – Associação de um caso de teste a requisitos e elementos da arquitetura.



Documentos do tipo “Relatório de Incidente de Teste” registram qualquer evento que ocorra durante o processo de teste e que requeira análise posterior. A partir do registro de um incidente é possível consultar qualquer artefato a ele relacionado.

Documentos do tipo “Diário de Teste” registram tarefas relevantes relacionadas com a execução dos testes.

## 5.2. Estratégia de rastreabilidade

### 5.2.1. *Links* de rastreabilidade

Considerando o foco em usuários do tipo *low-end*, Prometeu possui *links* de rastreabilidade pré-definidos, liberando os envolvidos com o teste da responsabilidade de defini-los e mantê-los. Assim, o aprendizado da ferramenta é facilitado permitindo uma compreensão mais rápida da importância da rastreabilidade e de seus benefícios. Atualmente, os *links* de rastreabilidade não possuem propriedades adicionais além daquelas definidas pela própria ferramenta, e são armazenados e recuperados como tabelas e relações em um gerenciador de banco de dados relacional.

A ferramenta suporta rastreabilidade horizontal entre os artefatos Requisitos, Elementos da Arquitetura, Caso de Teste, Plano de Teste, Especificação de Projeto de Teste, Especificação de Procedimento de Teste e Incidentes. A rastreabilidade vertical existe nos artefatos Requisitos, Plano de Teste, Especificação de Projeto de Teste, Especificação de Procedimento de Teste e Caso de Teste.

### 5.2.2. Informações de rastreabilidade

Para possibilitar o rastreamento das informações de rastreabilidade, nas tabelas relacionadas à representação de um artefato são armazenados dados sobre os seis tipos de informações sugeridos na seção 2.1.2. Esses atributos permitem o rastreamento, por exemplo: (a) das partes envolvidas no processo e seus papéis, como gerente de projeto, analista de sistemas, representante do cliente, (b) de observações e justificativas sobre a evolução de um artefato, (c) de quais fontes - pessoas ou artefatos - originaram um artefato, (d) de quando uma determinada informação foi capturada e modificada.

## 5.3. Alguns recursos de suporte à documentação e à rastreabilidade

A seguir são apresentadas algumas funcionalidades da ferramenta Prometeu.

### 5.3.1. Visualização das associações, e seu status, entre diversos artefatos

A Figura 8 ilustra a rastreabilidade entre os artefatos casos de teste, requisitos, elementos da arquitetura e incidentes na forma de uma matriz. A montagem da matriz é possível através da rastreabilidade horizontal existentes entre os artefatos. Além do formato de matriz, a ferramenta também permite a visualização das mesmas informações na forma de árvore e em HTML. Todos os artefatos listados podem ser consultados e visualizados. As colunas cujo conteúdo é “???” indicam situações nas quais estão faltando associações entre os artefatos. Na Figura 8 os casos de teste CT0001\_PRJ2005\_06 e CT0001\_PRJ2005\_07 não possuem nenhum requisito ou elemento da arquitetura associados.



Caso de Teste	Requisito	Arquitetura	Incidente
CT0001_PRJ2005_01	REQ_002_JORGE	DES_0001 ; DES_0002	INCIDENTE00001
CT0001_PRJ2005_02	REQ_003_JORGE		INCIDENTE00002
CT0001_PRJ2005_03	REQ_003_JORGE	DES_0001	INCIDENTE00003
CT0001_PRJ2005_04	REQ_001_JORGE		INC2233344
CT0001_PRJ2005_06	???	???	
CT0001_PRJ2005_07	???	???	

Figura 8 – Rastreabilidade entre casos de teste, requisitos e elementos da arquitetura.

### 5.3.2. Visualização do impacto causado por uma modificação em um artefato

Antes de gravar uma alteração em um artefato, a ferramenta alerta sobre todos os artefatos a ele associados. Todas as associações podem ser revisadas e, se necessário, os artefatos podem ser atualizados. Isso é possível através da existência de associações de rastreabilidade horizontal entre os artefatos. A Figura 9 ilustra esta situação durante a alteração de um caso de teste.

Identificador	Título	Tipo	Categoria
REQ_002_F02	Permitir a criação de diários de teste para ...	Requisito	Funcional
REQ_003_F03	Emittir relatórios sobre o status da execuçã...	Requisito	Funcional
REQ_004_F04	Emittir relatórios sobre os incidentes ainda ...	Requisito	Funcional
PROCEDIM_001	Procedimento para regras 011 até 020	Especificação de Procedimento de Teste	
PROCEDIM_002	Procedimento para regras 001 até 010	Especificação de Procedimento de Teste	
PROJETO_002	Projeto de teste 02	Especificação de Projeto de Teste	

Figura 9 – Alerta de artefatos associados no momento de uma alteração.

### 5.3.3. Identificação de problemas de atualidade e consistência entre artefatos

Uma das maneiras de descobrir possíveis problemas de inconsistência entre artefatos é no momento que um documento for consultado. Todos os artefatos associados, cujas informações de rastreabilidade estiverem desatualizadas, serão exibidos em uma lista, permitindo que os artefatos e as associações sejam revisados. Isso é possível através da rastreabilidade horizontal existente entre os artefatos. A Figura 10 ilustra um exemplo desta situação para um documento do tipo Especificação de Projeto de Teste.

Identificador	Seção do Documento	Tipo
REQ_003_F03	Funcionalidades Testadas	Requisito Funcional
REQ_004_F04	Funcionalidades Testadas	Requisito Funcional
CT0001_PRJ2005_10	Casos de Teste Associados	Caso de Teste
CT0001_PRJ2005_11	Casos de Teste Associados	Caso de Teste
CT0001_PRJ2005_12	Casos de Teste Associados	Caso de Teste
PROCEDIM_02	Procedimentos de Teste Associados	Procedimento de Teste

Figura 10 – Identificando problemas de atualidade e consistência entre artefatos.



### 5.3.4. Geração de um documento personalizado

Documentos personalizados podem ser gerados através da combinação de seções específicas de um ou mais documentos previamente registrados. O documento resultante, que será exibido em formato HTML, não será armazenado no banco de dados. A Figura 11 ilustra a tela para seleção das seções que comporão o documento.

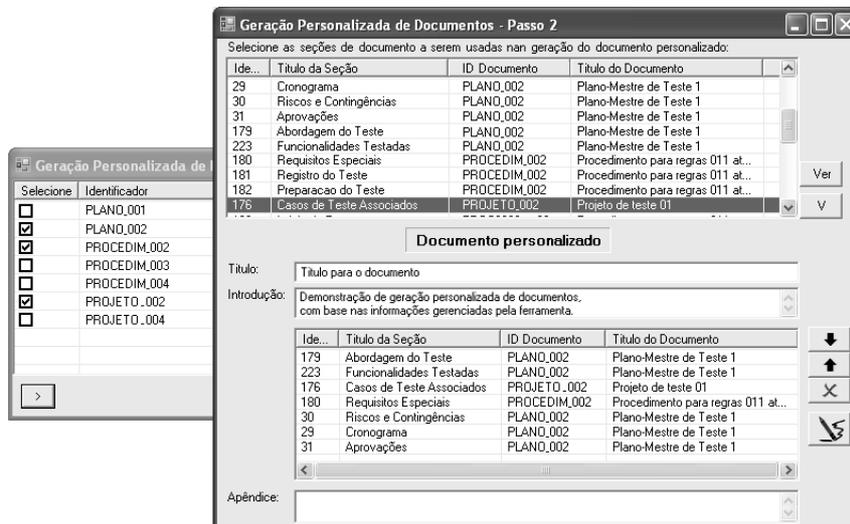


Figura 11 – Geração de um documento personalizado.

### 5.3.5. Identificação das partes envolvidas e dos artefatos relacionados à correção de um defeito

Artefatos associados a um incidente, assim como as partes envolvidas na sua criação e manutenção, são fontes de informações que podem contribuir para descobrir a causa do incidente, facilitando sua correção. A partir de cada incidente, pode-se navegar rapidamente até os artefatos associados e consultá-los; pode-se, também, consultar as partes envolvidas na criação e manutenção de cada artefato associado ao incidente. Isso é possível através da rastreabilidade horizontal e através do uso das informações de rastreabilidade existentes no conjunto de artefatos relacionados. A Figura 12 ilustra os artefatos relacionados a um incidente.

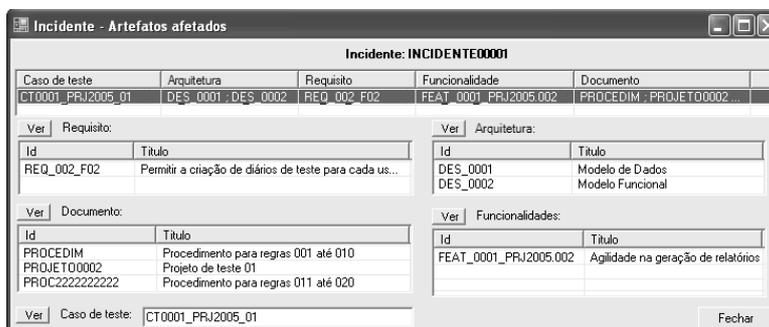


Figura 12 – Artefatos associados a um incidente de teste.

## 5.4. Comparação com outras ferramentas

Para posicionar Prometeu em relação a outras ferramentas de teste, é interessante fazer uma comparação resumida entre suas funcionalidades. É feita uma comparação básica entre Prometeu, ferramentas da IBM Rational e da Mercury.



Em Prometeu, são registradas informações sobre requisitos e elementos da arquitetura que sejam pertinentes ao teste, e a documentação é feita de acordo com a norma IEEE-829. É possível a rastreabilidade entre partes envolvidas, requisitos, elementos da arquitetura, e todos os artefatos prescritos pela norma IEEE-829 (entre eles, casos de teste e defeitos). Também é possível a geração de documentos personalizados a partir da seleção de informações previamente registradas em documentos de tipos diferentes.

Nas ferramentas da IBM Rational, RequisitePro é usada para gerenciar os requisitos e permitir rastreabilidade entre eles, TestManager é usada para gerenciar o teste e ClearQuest é usada para gerenciar os defeitos descobertos. A rastreabilidade entre requisitos e casos de teste, entre elementos da arquitetura e casos de teste e entre defeitos e casos de teste é obtida após a integração dessas três ferramentas. Já a Mercury possui o TestDirector, que permite o registro e rastreamento de requisitos, casos de teste e defeitos descobertos.

Nem a TestManager tampouco a TestDirector baseiam-se em artefatos de teste prescritos por uma norma como a IEEE-829.

## 6. Conclusões

Este artigo apresentou o conceito de rastreabilidade e seu emprego em um processo de teste de software baseado em documentação, com o necessário suporte automatizado. A rastreabilidade desempenha importante papel na obtenção de um teste de boa qualidade, ao contribuir para que sua gerência e execução sejam baseadas em informações atualizadas e consistentes. Entretanto, a implantação e manutenção da rastreabilidade são complexas devido ao grande número de informações a serem manipuladas: os maiores benefícios apregoados sobre a rastreabilidade só podem ser obtidos com o apoio de uma ferramenta de software. Prometeu é uma ferramenta, ainda em desenvolvimento, que oferece suporte à rastreabilidade e à documentação em um processo de teste. Em relação à documentação, utilizou-se a norma IEEE-829 como base para a definição de um conjunto de artefatos a serem documentados e rastreados. Foram apresentadas resumidamente algumas funcionalidades da ferramenta que mostram seu suporte à documentação e alguns dos benefícios da rastreabilidade em um processo de teste. Futuras melhorias na ferramenta incluem (a) aprimoramento da sua interface gráfica, melhorando a navegação entre os *links* de rastreabilidade e facilitando o acesso às informações de rastreabilidade, (b) implementar suporte à rastreabilidade entre diversos ciclos de teste, (c) fornecer suporte à rastreabilidade de tarefas relacionadas ao processo, (d) implementar suporte à definição manual de *links* de rastreabilidade e sua semântica.

## 7. Referências

- Beizer, B. (1990), *Black-Box Testing: Techniques for Functional Testing of Software and System*, Van Nostrand Reinhold.
- Chrissis, M. B. et. al. (2003), *CMMI – Guidelines for Process Integration and Product Improvement*, Addison-Wesley.
- Cleland-Huang, J. et. al. (2003) “Event-based traceability for managing evolutionary change”, *IEEE Transactions on Software Engineering*, Vol. 29, No 9, Setembro, p. 796 – 810



- Cockburn, A. (2005), *Escrevendo Casos de Uso Eficazes – Um Guia Para Desenvolvedores de Software*, Bookman Editora.
- Craig, R. D. e Jaskiel, S. P. (2002), *Systematic Software Testing*, Artech House.
- Crespo, A. N. et. al. (2002) “Application of the IEEE 829 Standard as a Basis for Structuring the Testing Process”, *The Journal of Software Testing Professionals*, Vol. 3, No. 3, Dezembro.
- Crespo, A. N. et. Al. (2004) “Uma Metodologia para Teste de Software no Contexto da Melhoria de Processo”, *Simpósio Brasileiro de Qualidade de Software*, Maio, p. 271-285.
- Domges, R. e Pohl, K. (1998) “Adapting traceability environments to project-specific needs”, *Communications of the ACM*, Vol. 41, No. 12, Dezembro, p. 54-62.
- Gelperin, D. (1982) “A Software Test Documentation Standard”. *ACM Proceedings of the 1st annual International Conference on Systems Documentation*, p. 61-63.
- Gotel, O. e Finkelstein, A. (1995) “Contribution Structures”, *Proceedings of 2nd International Symposium on Requirements Engineering*. IEEE Computer Society Press, pp. 100-107.
- Hetzel, W. (1987), *Guia Completo ao Teste de Software*, Campus.
- IEEE (1991), The Institute of Electrical and Electronics Engineers, “IEEE Std 610: Standard Computer Dictionary”, IEEE Computer Society.
- IEEE (1998), The Institute of Electrical and Electronics Engineers, “IEEE Std 829: Standard for Software Test Documentation”, IEEE Computer Society, Setembro.
- ISO/IEC (2003), The International Organization for Standardization and the International Electrotechnical Commission, “ISO/IEC 15504 – Information Technology – Process Assessment – Part 2”.
- Kaner, C. et. Al. (2001), *Lessons Learned in Software Testing*, Wiley.
- Leffingwell, D. e Widrig, D. (2000), *Managing Software Requirements – A Unified Approach*. Addison-Wesley.
- Pfleeger, S. L. e Bohner, S. A. (1990) “A framework for software maintenance metrics”, *IEEE Conference on Software Maintenance*, Novembro.
- Phillips, D. (1998), *The Software Project Manager’s Handbook*. IEEE Computer Society.
- Pressman, R. S. (2002), *Engenharia de Software*, McGraw-Hill.
- Ramesh, B. e Jarke, M. (2001) “Toward reference models for requirements traceability”, *IEEE Transactions on Software Engineering*, Vol. 27, Issue 1, Janeiro, p. 58 – 93
- Robertson, S. e Robertson, J. (1999), *Mastering the Requirements Process*, Addison-Wesley.
- Sommerville, I. (2003), *Engenharia de Software*, Pearson Education do Brasil.