AEAMESP

20ª Semana de Tecnologia Metroferroviária 2014

DESAFIOS DOS NOVOS SISTEMAS METROFERROVIÁRIOS BASEADOS EM SOFTWARES

Fábio Siqueira Netto

Gilmario Ribeiro

20ª SEMANA DE TECNOLOGIA METROFERROVIÁRIA

PRÊMIO TECNOLOGIA E DESENVOLVIMENTO METROFERROVIÁRIOS

FÁBIO SIQUEIRA NETTO

GILMARIO RIBEIRO

Desafios dos Novos Sistemas Metroferroviários Baseados em Softwares

São Paulo

2014

Sumário

Capítulo 1

Introdução

A evolução das tecnologias existentes produz uma série de benefícios para todos os setores da ciência e da sociedade, porém, em certos segmentos, tal evolução trouxe consigo a necessidade de se revisar, ou até mesmo de se adaptar, processos que já estavam consolidados e amadurecidos. Um exemplo deste cenário é a grande automatização de aplicações do setor metroferroviário, os quais são classificados como sistemas de Missão Crítica, termo este que será usado deste ponto em diante neste trabalho. Os requisitos vitais de segurança nessa categoria de sistemas devem ser garantidos por meio de rotinas de testes e de validações em seus softwares.

De forma geral, um sistema de Missão Crítica pode se referir a qualquer equipamento, processo ou programa, nos quais uma falha pode comprometer a sua operação, inviabilizando a missão para a qual tal sistema foi desenvolvido. São exemplos destes sistemas: o gerenciamento de operações bancárias, a automação de equipamentos hospitalares, o monitoramento de usinas nucleares e o controle de trens e aeronaves, dentre outros (Tanenbaum, 2004).

Nesses sistemas o conceito de segurança pode ser entendido de duas formas: a segurança de acesso ao sistema e aos seus dados, em Inglês *security*, ou a segurança da vida das pessoas que podem ser afetadas por uma falha, característica esta definida pela palavra em Inglês *safety*. As pesquisas apresentadas aqui se referem, principalmente, aos sistemas

preocupados com a garantia dos requisitos de *safety,* mas questões de *security* também são abordadas.

Para assegurar o atendimento a esses requisitos, existem diversas ferramentas disponíveis, porém a maioria das publicações que tratam deste assunto descreve que o processo V&V (*Validation and Verification*) é uma das mais eficientes para esta finalidade. O V&V corresponde a um processo de testes em sistemas, disciplina esta ligada à Engenharia de Software (Pressman, 2006).

No entanto, mesmo aplicando-se essa sistemática de verificação e de validação, ainda assim acontecem falhas em Sistemas de Missão Crítica que afetam a segurança de pessoas, tais como acidentes em aeronaves e em trens de passageiros (Parnas et al, 1990). Em virtude deste cenário, este trabalho procura ressaltar a necessidade de estratégias mais rígidas de testes e de validações destes sistemas, propondo o uso de rotinas adicionais ao processo que existe atualmente.

Para atingir tal objetivo, o conteúdo deste trabalho está distribuído em cinco capítulos. No Capítulo 1, é apresentada uma introdução que visa contextualizar a motivação e a estrutura do trabalho. No Capítulo 2 é apresentado o diagnóstico do problema tratado, explorando conceitos de Sistemas de Missão Crítica e a sua evolução e algumas áreas de atuação, mostrando a relação entre as rotinas de testes e validações e como são aplicadas aos sistemas de missão crítica. No capítulo 3 é apresentada uma breve análise, em forma de relatório, que compila o entendimento acerca do problema apresentado.

Por fim, o Capítulo 4 apresenta as conclusões preliminares obtidas neste trabalho, assim como uma série de possíveis pesquisas futuras que podem completar a discussão proposta aqui.

Capítulo 2

Diagnóstico

Como descrito, os sistemas metroferroviários atuais se aproveitam de tecnologias computacionais modernas já usadas em outros segmentos, tais como a medicina e a aeronáutica. No entanto, a maioria das operadoras de transporte sobre trilhos ainda mantém rotinas operacionais e processos técnicos baseados em modelos já ultrapassados. Este cenário merece atenção especial, pois pode acarretar uma série de problemas para as operadoras de transporte sobre trilhos, os quais vão desde o desperdício de recursos humanos e de materiais até a insegurança na operação dos seus sistemas.

Com base nesse cenário, este capítulo apresenta uma série de conceitos e informações que formam um conjunto de evidências que atestam as preocupações apontadas neste trabalho.

2.1 Sistemas de Missão Crítica

Assim como descrito na Introdução, os Sistemas de Missão Crítica correspondem a qualquer equipamento, dispositivo, processo, tarefa, programa, entre outros, nos quais a manutenção da missão do sistema é a característica mais importante, pois uma falha pode comprometer o negócio da empresa ou expor a integridade física das pessoas. As características principais desses sistemas são (Pressman, 2006):

- Tempo de resposta pré-definido;
- Alta confiabilidade;
- Tolerância a falhas;
- Segurança.

No que diz respeito ao requisito de tempo de resposta, em sistemas de missão crítica esse tempo é fator fundamental que deve ser considerado no projeto, pois atendê-lo implica menos riscos à sua operação. Pode-se citar, como exemplo, o controle de caldeiras de uma indústria siderúrgica, no qual o sistema deve agir rapidamente se receber uma informação de sobreaquecimento. Neste caso, há um tempo limite para que a caldeira seja desligada e os dispositivos de segurança sejam ativados, tempo este que, se não atendido pelo sistema, pode causar sérios acidentes.

Porém, é importante ressaltar que não é a velocidade da resposta do sistema que o caracteriza como crítico, mas sim que esse tempo seja estabelecido previamente e que seja respeitado na sua implantação, ou seja, não importa se a resposta é rápida ou lenta, ela deve ser conforme foi definida.

Já a confiabilidade consiste na probabilidade de um sistema não falhar em um determinado período de tempo. Tal estimativa é obtida por meio de cálculos matemáticos que consideram fatores como a degradação física do equipamento e a obsolescência do software. Ter alta confiabilidade significa falhar muito pouco durante a operação do sistema.

Outra característica importante dos sistemas de missão crítica é a tolerância a falhas. Diferentemente da confiabilidade, mas que afeta diretamente o seu cálculo, a tolerância a

falhas fornece ganhos de disponibilidade, isto é, o sistema permanece mais tempo disponível para a operação quando o surgimento de falhas não afeta o seu processamento. Algumas técnicas conhecidas para a tolerância a falhas são (Bloom, 2009):

- Módulos redundantes, sejam físicos ou lógicos, que funcionam em modo de espera, assumindo o processamento quando o dispositivo principal entre em falha;
- Rotinas de auto diagnósticos, as quais proporcionam requisitos de manutenções preventivas e preditivas, pois antecipam possíveis problemas futuros.

Por fim, o requisito do sistema de missão crítica mais relevante no contexto deste trabalho é o de segurança. Na língua portuguesa a palavra “segurança” tem um amplo espectro, podendo apresentar várias interpretações, porém na língua inglesa existem dois termos que melhor a definem: *security*, quando envolve a segurança dos dados e dos seus acessos e; *safety*, quando envolve a segurança da vida das pessoas. As pesquisas apresentadas aqui, conforme já declarado, correspondem principalmente aos aspectos de *safety*, sendo este termo o usado no decorrer do trabalho.

Segundo (Pressman, 2006), um projeto de um sistema de missão crítica é uma das tarefas mais desafiadoras para a engenharia de software, em virtude da sua complexidade e por ter muitas ligações com o mundo externo. Tal afirmação reforça a necessidade de serem estabelecidos modelos rígidos para a verificação e para a validação dos requisitos destes sistemas, modelos estes que são abordados e discutidos nas próximas sessões.

2.2 Sistemas de Missão Crítica no Setor Metroferroviário

Em se tratando do setor metroferroviário, os desafios dessa evolução se concentram em quatro grandes dimensões: requisitos, desenvolvimento, processos de validação e de testes e modelo de trabalho. Os problemas vão surgindo e se acumulando de etapa em etapa de cada uma dessas dimensões, desde a definição de um conjunto de requisitos elaborado sem a devida base conceitual necessária, até a reestruturação do modelo de trabalho. As faltas de conhecimento e de capacitação em cada uma dessas dimensões têm, como resultados, a entrega de sistemas que não atendem as exigências operacionais e a aplicação de novos modelos de trabalho que são incompatíveis com os processos internos e com a cultura previamente estabelecida na organização.

No entanto, existem alternativas que podem ajudar a evitar que um cenário deste tipo se configure. Essas alternativas sempre se baseiam em um planejamento em longo prazo, no qual deve estar previsto a troca experiências com setores empresariais diferentes, de forma a se evitar que erros já superados em outros segmentos se repitam em projetos do setor metroferroviário, assim como a definição e estruturação das rotinas de testes em software, assunto este abordado com maior profundidade neste trabalho.

Outro fator crítico de sucesso em projetos que envolvam novas tecnologias é a capacitação do corpo técnico e operativo, pois com base nos novos conhecimentos adquiridos é possível readequar eficazmente o modelo de trabalho e preparar a organização para tirar o maior proveito dos recursos que as novas tecnologias oferecem.

2.3 Processo de Testes e Validação de Software

As atividades de testes de software permeiam quase todo o ciclo de desenvolvimento de um sistema e tem como objetivo encontrar erros ou problemas não identificados anteriormente. Estas rotinas buscam assegurar a qualidade do software desenvolvido, fazendo com que ele atenda os requisitos especificados em projeto. A Figura 2.1 apresenta o modelo clássico em cascata do ciclo de via do software, no qual é possível constatar a existência de rotinas de testes em boa parte das etapas do seu desenvolvimento (Rocha et al, 2006).

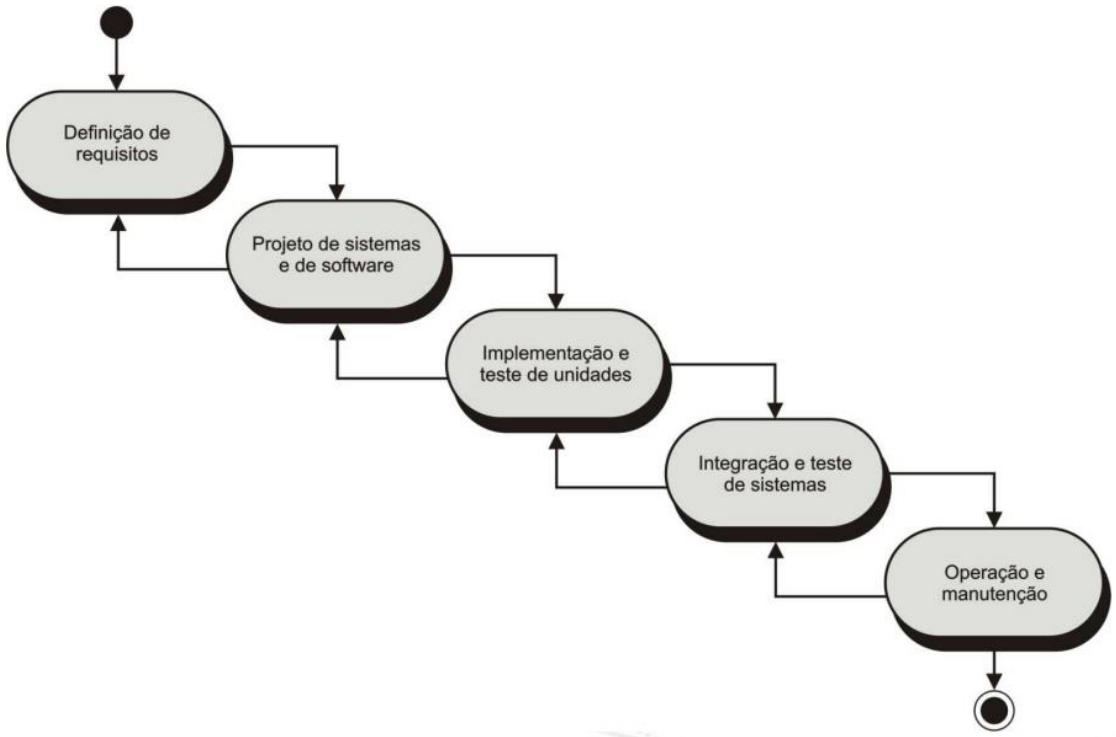

Figura 2.1: Modelo de ciclo de vida em cascata.

Os testes devem ser planejados antecipadamente, assim como também devem ser realizados sistematicamente. Existe uma grande variedade de técnicas usadas para este fim, tornando a escolha do modelo certo de teste uma questão crítica.

Não existe uma rotina de teste de uso geral, elas devem ser definidas a partir do sistema que está em desenvolvimento. No entanto, independentemente do modelo utilizado, ele deve apresentar as seguintes características (Pressman, 2006):

- A atividade de teste se inicia no nível do módulo e prossegue em direção a integração do sistema;
- Os testes devem ser executados pela equipe de desenvolvimento em conjunto com uma equipe independente;
- As atividades de testes e de depuração são diferentes, mas as atividades de testes devem suportar processos de depuração quando for necessário.

A atividade de testes é um elemento de um conjunto mais amplo chamado de Verificação e Validação (V&V – *Validation and Verification*). Em virtude da sua importância no contexto deste trabalho, tal processo é detalhado na sessão seguinte.

Processo de Verificação e Validação – V&V

O modelo V&V corresponde a um conceito amplo de técnicas e métodos aplicados para assegurar que o sistema em desenvolvimento cumpra as suas especificações e atenda

as expectativas dos seus usuários. Neste processo, é importante diferenciar os conceitos de verificação e de validação.

A etapa de Verificação procura avaliar se o software está sendo construído corretamente, ou seja, preocupa-se com o processo de desenvolvimento. Além disso, é nesta fase que se verifica se o sistema está atendendo as especificações definidas em projeto. Já a Validação foca o produto final, avaliando se está em conformidade com as necessidades dos usuários.

Existe uma norma editada pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE - *Institute of Electrical and Electronic Engineers*) de número 1012 e de título *"Software Verification and Validation"* a qual visa estabelecer critérios para que o processo V&V seja realizado de forma organizada e padronizado. Essa norma possui três princípios básicos (IEEE, 2005):

1. Prover um padrão mínimo de requisitos que faça parte do escopo do Plano de Verificação e Validação do Software - SVVPs (*Software Verification and Validation Plans*);
2. Definir especificações mínimas de atividades de V&V, incluindo os requisitos de entrada e saída, as quais devem ser estar nos SVVPs;
3. Sugerir atividades de V&V opcionais para serem usadas sob medida nos SVVPs.

A norma IEEE-1012 recomenda a execução de vários tipos de testes: em unidade, de integração, em sistema e de aceitação, assim como deve ser elaborada a documentação dos resultados obtidos e dos defeitos detectados na execução destas atividades. A Tabela 3.1

apresenta algumas atividade e tarefas de V&V recomendadas por esta norma (IEEE, 2005). Na próxima sessão estes conceitos de testes e do processo V&V são usados para apresentar a proposta deste trabalho.

Tabela 3.1: Atividades V&V conforme norma IEE-1012.

Atividade	Tarefas
Gerência de Software	Planejamento; Monitoração; Avaliação dos resultados da monitoraão e o impacto dessas mudanças; Relatórios Gerenciais.
Requisitos de Software	Elaboração da documentação de especificação de Requisitos do Software; Análise de Impacto; Análise da Interface; Planejamento dos Testes de Sistema.
Projeto do Software	Análise de Impacto; Desenvolvimento do Projeto do Software Análise da Interface; Planejamento dos Testes de Unidade; Planejamento dos Testes de Integração.
Codificação	Desenvolvimento do Código; Análise da Interface; Complementação dos Testes de Unidade.
Teste de Unidade	Execução dos Testes de Unidade; Registros dos testes executados.
Teste de Integração	Finalização da preparação dos Testes de Integração; Execução dos Testes de Integração; Registros dos testes executados.
Teste de Sistema	Complementeção da preparação dos Testes de Sistema; Execução dos Testes de Sistema; Registros dos testes executados.
Teste de Aceitação	Execução dos Testes de Aceitação; Registros dos testes executados.
Operação e Manutenção do Software	Análise de impacto das mudanças; Repetir as atividades de "Gerência de Software"; Repetir das atividades técnicas V&V anteriores.

2.4 Validação e Testes em Sistemas de Missão Crítica

A preocupação em se melhorar os processos de testes e validações em sistemas de missão crítica têm crescido exponencialmente nos últimos anos, pois a complexidade de tais sistemas só aumenta com o tempo. O processo V&V é o mais utilizado para assegurar os requisitos de missão crítica, mas ele por si só não é suficiente (Alves, 2011; Parnas et al, 1990).

Neste capítulo são apresentados critérios adicionais que podem incorporar melhorias nesses sistemas, principalmente aqueles que exigem requisitos de *safety,* tais como, por exemplo, a operação de trens de alta velocidade e a controle de ônibus espaciais. A Figura 4.1 mostra estes exemplos.

(a) (b)

Figura 4.1: Exemplos de Sistemas de Missão Crítica: (a) Trem de Alta Velocidade; (b) Ônibus espacial da NASA.

Espera-se que neste tipo de sistema as rotinas de teste consigam detectar quaisquer problemas que os deixaria em uma situação de risco. Teoricamente, esse risco é uma função matemática calculada por meio do tempo médio para a ocorrência de uma falha insegura, o MTTUF (*Mean Time To Unsafe Failure*). Este indicador, para sistemas como o e controle de trens, normalmente aponta para uma falha insegura a cada 200 ou 300 anos (Smith et al, 2000).

No entanto, muitos sistemas de missão crítica apresentam falhas que levam a ocorrência de acidentes, como é o caso da queda do avião Airbus da empresa TAM no ano de 2007. Este acidente, amplamente divulgado pelas mídias nacionais e internacionais, pode ter acontecido por uma falha do software de automatismo da aeronave, conforme apresentado pelo Jornal da TV Gazeta no dia primeiro de agosto de 2007.

Se de fato houve um problema nesse sistema, será muito difícil saber, pois a tecnologia envolvida é considerada segredo industrial e é pouco provável que será revelada. Porém, acidentes como este levantam questões do tipo: “Este acidente poderia ter sido evitado?” ou, “O que faltou ser visto para que o problema não ocorresse?”.

A proposta deste trabalho não é responder questões como estas, mas fornecer rotinas complementares ao processo V&V para que elas não sejam feitas. Estas rotinas são detalhadas a seguir.

Equipe independente, treinada e capacitada no sistema

A independência da equipe de testes em relação à equipe de desenvolvimento é condição normal da Engenharia de Software, porém nem sempre essa equipe está treinada e

tem conhecimento sobre o sistema em testes. Tal conhecimento proporciona uma visão sistêmica ampliada, fazendo com que possíveis problemas possam ser detectados independentemente deles estarem nos casos de testes.

Procedimentos claros e objetivos, com parâmetros pré-definidos

Os procedimentos de testes, ou casos e cenários de avaliação, devem estar claros, com parâmetros que a equipe conheça e saiba da sua importância. A clareza evita interpretações dúbias e o reconhecimento de eventuais desvios.

Ferramentas adequadas e aferidas

A equipe de testes, estando capacitada e tendo em mãos procedimentos claros e objetivos, também precisa dispor de todas as ferramentas necessárias à execução das suas atividades. O uso de ferramentas inadequadas pode apontar resultados incoerentes com o verdadeiro estado do sistema, incorrendo no seu mau funcionamento quando este já estiver em operação.

Manter registros dos testes executados

Todas as atividades executadas devem ter os resultados apontados em registros, os quais devem permanecer armazenados para futuras consultas. Esses registros guardam a informação de como o sistema estava no momento de sua implantação, assim como servem de referência caso algum problema ocorra durante a sua operação.

Estas rotinas não garantem a inexistência de uma falha insegura, porém elas podem ajudar a identificar uma gama maior de problemas e erros em sistemas de missão crítica, auxiliando o processo V&V quando aplicado a esses sistemas.

Capítulo 3

Análise dos Resultados

Outra ocorrência que ilustra a preocupação do uso intenso de softwares em sistemas de missão crítica é o acidente com um avião da empresa Spainair em 2008, que vitimou 154 pessoas no voo JK5022 em decorrência de uma anomalia (vírus) no sistema de controle de falhas das aeronaves. Esta situação demonstra a importância de uma abordagem completa sobre os softwares que estão em todas as cadeias de segurança dos sistemas, não devendo ser concentrada apenas nos módulos embarcados, mas também nas aplicações de apoio.

Porém, o mau funcionamento computacional não é a única fonte de problemas para os sistemas modernos, também devem ser destacadas as possibilidades de invasões lógicas, as quais são ameaças comuns no novo contexto de tecnologias interligadas por meio de redes de comunicação de dados.

A exemplo disso, cita-se a criação do CDCiber - Centro de Defesa Cibernética, um órgão do Exército Brasileiro responsável pela defesa cibernética do país, onde a proteção às Redes Públicas passa a ser considerada como assunto de Segurança Nacional, haja vista o Brasil se encontrar entre os 10 países com maiores números de ataques (média de 220 mil/mês).

Com base no que foi apresentado até este ponto, é possível deduzir que a evolução tecnológica em todos os setores da ciência é inevitável, porém cada área possui um ritmo evolutivo próprio, o qual depende de questões culturais, estruturais, políticas, organizacionais, técnicas, entre outras.

Capítulo 4

Conclusões Preliminares

A evolução tecnológica faz com que os sistemas computacionais, cada vez mais, ganhem vida própria. Quando esses sistemas são de Missão Crítica, os processos que visam a garantia dos requisitos de projeto despertam preocupação nas mais diversas áreas, pois falhas podem implicar perdas materiais e de vidas humanas.

Atualmente, os processos chamados de V&V, Verificação e Validação, são os mais usados durante a etapa de testes e de aceitação desses sistemas, porém, conforme mostrado no decorrer das pesquisas apresentadas aqui, eles não são suficientes para garantir o MTTUF de projeto em sistemas com requisitos de *safety* incorporados.

Para ajudar a suprir eventuais lacunas no processo V&V aplicado aos sistemas de missão crítica, este trabalho propõe rotinas adicionais que podem melhorar a qualidade do software desenvolvido para missões críticas. Acredita-se que tais rotinas podem, de fato, apoiar as atividades de testes e de validações.

Porém, por se tratar de uma trabalho preliminar, se propõe como pesquisas futuras os itens abaixo, os quais podem completar as ideias apresentadas até este ponto:

- Definição de métricas que considerem as rotinas adicionais e avaliem o seu ganho no processo V&V;
- Estudar criteriosamente o processo V&V para localizar eventuais pontos de melhoria e a compatibilidade com as rotinas propostas neste trabalho.

Referências

[Alves,, 2011] ALVES, M. C. B. . *V&V of the Brazilian Satellite Launcher Control System. Practical UML based Specification, Validation, and Verification of Mission Critical Software*. 1ª edição. Indianapolis: Dog Ear Publishing, 2011, v. , p. 130-141.

[Bloom, 2009] HAHN, G. & SHAPIRO, S. *Reliability Centered Maintenance: implementation made simple*. Nova Iorque: Mcgraw Hill, 2009.

[IEEE, 2005] Institute of Electrical and E. Engineers. *IEEE 1012-2004 - ieee standard for software verification and validation.* IEEE.2005, pp. 0–110, revisão da IEEE Std 1012, 1998.

[Parnas et al, 1990] PARNAS, D.L., SCHOUWEN, A. J. V., KWAN, S. P. *Evaluation of safety-critical software*. Communications of the ACM, 33(6):636–648, 1990.

[Pressman, 2006] PRESSMAN, R. S. *Engenharia de Software.* 6ª Edição. São Paulo: Mcgraw Hill, 2004.

[Rocha et al, 2006] ROCHA, A. R. C., MALDONADO, VJ. C., WEBER, K. C. *Qualidade de Software: Teoria e Prática.* São Paulo: Prentice Hall, 2001.

[Smith et al, 2000] SMITH, D.T.; DELONG, T.A.; JOHNSON, B.W.; GIRAS, T.C.. *High Assurance Systems Engineering, 2000, Fifth IEEE International Symposim on. HASE 2000.* , p. 17-24.

[Tanenbaum, 2004] TANENBAUM, A. S. *Sistemas Operacionais Modernos.* 2ª Edição. São Paulo: Pearson, 2004.