MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-10240-PRE/5758

TESTE DE CONFORMIDADE PARA SOFTWARE DE SISTEMAS ESPACIAIS

Ana Maria Ambrosio

Trabalho publicado nos Anais do Workshop de Teses e Dissertações São Paulo, realizado São Paulo-SP em outubro de 2003.

INPE
São José dos Campos
2004

[1]Teste de Conformidade para Software de Sistemas Espaciais

Ana Maria Ambrosio

Instituto Nacional de Pesquisas Espaciais (INPE)
Av. Dos Astronautas, 1758 - São Jose dos Campos -12227-010 - SP - Brasil
ana@dss.inpe.br

Resumo: Este artigo apresenta uma análise do ciclo de vida de verificação e validação de sistemas espaciais, conforme recomendado nos padrões da *European Cooperation for Space Standardization* – ECSS. O objetivo desta análise é identificar as fases nas quais os teste de conformidade de software, tal como especificado pelas normas ISO (ISO-9646 CTMF) para teste de protocolos, são aplicáveis. A análise fundamenta a proposta de um processo de teste de conformidade para software de aplicações espaciais, envolvendo teste de conformidade e técnicas de injeção de falhas para apoiar a garantia da qualidade dos sistemas especiais, no que se refere ao alto grau de confiança no funcionamento (*dependability*).

1. Introdução

O desenvolvimento de um sistema espacial está sujeito a riscos devido a situações como: (i) necessidade de alto nível de desempenho; (ii) número reduzido de produção, elevando o custo de desenvolvimento sem oportunidade de amortização; (iii) inabilidade de operar completamente o elemento espacial em condições realísticas antes do seu lançamento ao espaço; (iv) acesso limitado ao produto durante a validação e; (v) condições ambientais específicas do espaço.

Visando orientar a produção de sistemas espaciais de alta qualidade e a custos mais reduzidos, a Agência Espacial Européia (ESA) vem se empenhando para definir uma série de padrões chamados *European European Cooperation for Space Standardization* (ECSS), que possam ser usados tanto pelas agências espaciais governamentais como pela indústria aeroespacial. Este esforço é justificado por constantes falhas nas missões, bem como por atrasos devido a subestimados cálculos de custos e de cronograma. A tabela 1 ilustra alguns problemas ocorridos nos últimos anos, apresentado pelo Dr. Carlo Mazza [1].

O insucesso das missões exemplificadas na tabela 1 é atribuído a insuficiência na validação do software durante os testes de sistema. Esta ilustração e o fato de que as funções desempenhadas pelos satélites são cada vez mais delegadas ao software,

[1] Publicado nos anais doAnais do Workshop de Tese e Dissertações São Paulo, SP Outubro de 2003, Ed. Por A F.Zorzo, F. Brasileiro, I.E.S.J. Pôrto , São Paulo - EPUSP, 2003, p. 85-90. (ISBN- 85-86686-25-5).

apontam a necessidade de se enfatizar as atividades de teste de software nos projetos espaciais.

Tabela 1. Falhas em Sistemas Espaciais

PROJETO	INCIDENTE	CAUSA
ARIANE 501, Junho 1996	Falha no Lançamento, perda de 4 satélites CLUSTER	Insuficiência de Validação de software e Validação de sistema
Mars Climate Orbiter Set. 1998	Perda do *Orbiter*	Troca de unidades de medida. Insuficiência de validação de software e validação de sistema
Mars P. Lander Dez.1998	Perda do *Lander*	Erro de sw forçou desligamento prematuro da máquina de pouso: insuficiência de validação de software
Titan-4B Abril 1999	Falha ao colocar o satélite USAF Militar na sua órbita final	Falha na ignição do estágio superior do Centaur. Erro no ponto decimal no sistema de guiagem. Insuficiência de validação apropriada no software
Delta-3, Abril 1999	Lançamento abortado	Falha do SW de bordo na inicialização da máquina principal. Insuficiência de validação apropriada de software
PAS-9 Março 2000	Perda do satélite ICO *Global Communication* F-1	Erro na atualização do software de solo. Insuficiência de validação de software

A definição de uma metodologia e de uma arquitetura para testes de software de sistemas espaciais pode reduzir custos e torná-los cada vez mais confiáveis, uma vez que a própria confiança no funcionamento (*dependability*) da ferramenta de teste cresce.

Os testes fazem parte das atividades de V&V. A cada fase do ciclo de vida do desenvolvimento do software, corresponde uma fase de teste: teste de unidade, teste de integração, teste de sistema e teste de aceitação [3].

Os testes de unidades dependem da linguagem de implementação e estão intimamente ligados às responsabilidades do cliente (contratado), assim também, os testes de integração. Já os testes de sistemas e os de aceitação, são do tipo caixa-preta [3] e devem ser definidos pela contratante do produto. Neste tipo de teste, o objetivo é verificar a conformidade da implementação com relação a sua especificação. A norma IS-9646 [6] define conceitos, uma metodologia e um arcabouço para *teste de conformidade* de protocolos, cujas idéias principais são apresentadas na seção 3.

Neste artigo é discutida uma proposta do processo de testes de conformidade ajustado às necessidades e características do software de sistemas espaciais. Este processo, combinará definições consagradas em testes de conformidade com técnicas de injeção de falhas, a fim de proporcionar facilidades para testar o software sujeito a problemas causados por condições ambientais específicas do espaço, como radiação e falhas de comunicação. A seção 4, apresenta as principais idéias do processo de teste e a seção 5, conclui este artigo.

2. Normas ECSS

2.1 Organização Geral

As normas ECSS, cobrem as áreas de um programa espacial, tais como: desempenho técnico e qualidade; orçamento, usuários relacionados, política; custo de contratos, cronograma, operação e segurança no funcionamento (*dependability*).

O conjunto de normas ECSS está organizado em 3 ramos: (i) Gerenciamento de Projeto; (ii) Garantia do Produto Espacial (atividades de V&V: revisões, análises, testes) e; (iii) Engenharia Espacial (atividades de realização de testes de software).

Uma descrição da seqüência de atividades de V&V e as recomendações do que fazer em cada fase do processo de desenvolvimento de um sistema espacial, para se atingir a qualidade desejada, fazem parte da ECSS.

2.2 Ciclo de vida de Verificação e Validação

As fases de V&V, ilustradas na figura 2, refletem o ciclo de vida de desenvolvimento recomendado pelas normas ECSS. As atividades de verificação iniciam-se já nas fases de definição da missão espacial [2], [4], com a preparação do Plano de Validação Operacional.

Dentre as atividades de V&V verifica-se que os testes são planejados em cada etapa do desenvolvimento e executados após a conclusão das primeiras unidades do código do software. Os testes de unidades, assim como os testes de integração, são do tipo estrutural, portanto utiliza-se técnicas caixa-branca para sua especificação. Analisando-se as normas no que concerne a Qualificação Técnica (ou validação funcional) e a Validação Operacional, constata-se que as atividades são compostas de teste do tipo caixa-preta [3]. Desta forma, pode-se contribuir com definições de teste de conformidade, cujo objetivo é responder se o produto implementado corresponde exatamente ao produto especificado.

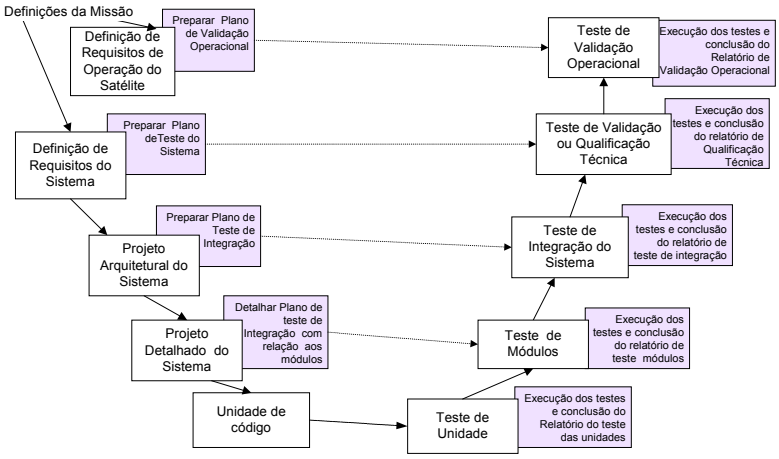

Figura 2. Ciclo de vida de um sistema espacial.

3. Teste de Conformidade - IS-9646

A norma IS-9646 [6], define uma metodologia, a terminologia correspondente, uma linguagem de especificação de teste, define procedimentos para serem seguidos durante os testes, os quais podem ser realizados por laboratórios distintos e ainda prove um arcabouço (*framework*) para especificação de casos de testes de conformidade (*test suite*). Esta norma orienta a execução dos testes de conformidade para protocolos comuns dos sistemas *Open System Interconnection* (OSI). Entretanto, dada a aplicabilidade de seus conceitos, ela tem sido usada para teste de outros sistemas de comunicação, não apenas os protocolos do padrão OSI [5].

O processo de teste de conformidade possui três grandes fases: (i) especificação de uma *seqüência abstrata de teste*[2] (conjunto de testes que independem da implementação) gerada a partir dos *propósitos de teste* (descrição informal de uma propriedade do protocolo); (ii) implementação dos testes (definição dos meios para se executar os testes e tradução dos *testes abstratos* nos *testes executáveis* (aqueles que podem ser executados ou interpretados pelo equipamento/sistema real de teste); (iii) execução dos testes com uma *implementação em teste* (IUT, do inglês, *implementation under test*). O comportamento da IUT é observado e levado em consideração para a declaração do *veredicto* de conformidade da implementação com relação à especificação.

4. Método de teste de conformidade associado à injeção de falhas

[2] A linguagem recomendada para escrever os testes, proposta pela CTMF, é a TTCN (*Tree and Tabular Combined Notation*)

A combinação dos conceitos de injeção de falhas com testes de conformidade, proporcionam ao sistema em teste a possibilidade de verificação do seu comportamento em presença de falhas externas, como as que um software de sistema espacial pode encontrar em seu ambiente de vôo. Um esboço dos passos do processo de teste de conformidade associado à técnica de injeção de falhas, é apresentado na figura 4.

Nas definições de [6] o conceito de *propósitos de testes*, corresponde ao requisito que deve ser exercitado durante os testes. Se os testes gerados não atenderem aos requisitos de teste, diz-se que os testes não tiveram sucesso e a etapa de validação não pode ser concluída.

No processo proposto, os casos de teste podem ser gerados automaticamente a partir de uma especificação formal que define seu comportamento; e podem ser complementados com *casos de falhas,* cujo tratamento pode ou não estar explicitamente especificado. Com base em pesquisas anteriores no contexto do projeto ATIFS [7], não apenas a geração de casos de teste pode ser automatizada, mas também, todo o processo ilustrado na figura 4.

Na primeira fase do teste de conformidade (*especificação da seqüência de testes abstratos),* a especificação do sistema é traduzida para uma linguagem formal baseada em EFSM, no caso a linguagem de especificação de protocolos (LEP) e a derivação dos testes de conformidade são feitas através de um algoritmo de busca em grafo, no qual, cada caminho corresponde a um caso de teste a ser executado [7]. A derivação de casos de falhas pode ser previamente programada, ou, por escolhas interativas do condutor dos testes. Há opções para mensagens ausentes, duplicadas e corrompidas; falhas intermitentes, repetitivas e transientes.

Na segunda fase (*implementação dos testes,)* a seqüência abstrata é traduzida em um programa em *Tool Command Language* (TCL), podendo assim, ser executada pelo sistema de teste.

Na terceira fase (*execução dos testes),* a execução é auxiliada por computador e o *log de teste* (contendo o traço observado durante a execução dos testes) é gerado automaticamente. O traço (seqüência de saídas) esperado é gerado em separado com base na especificação do comportamento da IUT. Posteriormente, o traço observado e o traço esperado são comparados e a declaração do veredicto do teste de conformidade é obtida automaticamente. Para as situações não previstas na especificação, um relatório é gerado para análise.

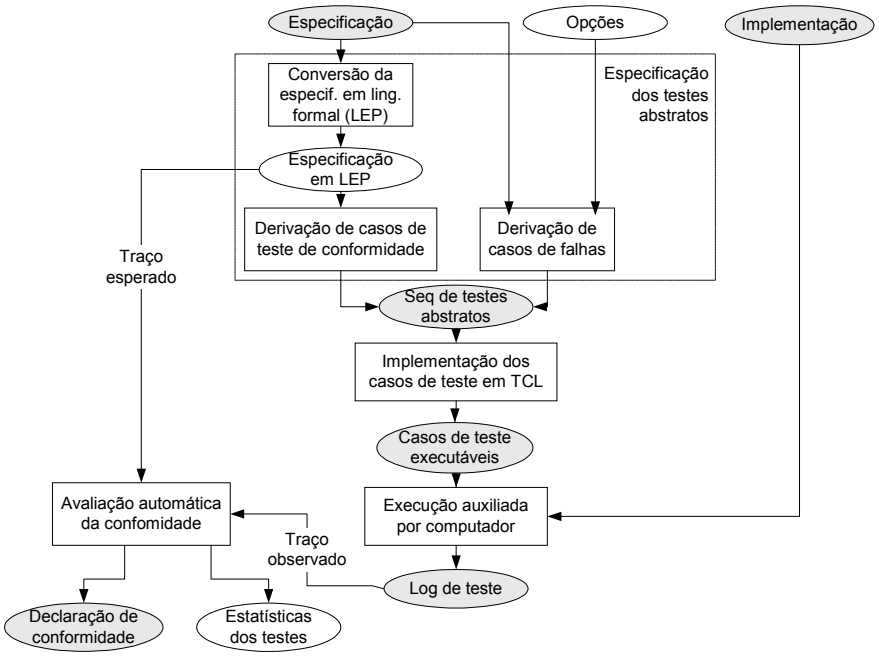

Figura 4. Processo de Teste de Conformidade com Injeção de Falhas

A geração automática da seqüência abstrata de testes, com base em métodos de varredura de grafos, pode levar ao problema de explosão do número de casos de testes. No método proposto, este problema é minimizado uma vez que os casos de exceção e os casos de tratamento de falhas são excluídos da especificação de conformidade e deixados para a execução por injeção de falhas.

5. Trabalhos futuros

Os sistemas que vão a bordo de satélites, estão sujeitos a eventos causados por radiação e falhas de comunicação e podem reagir da forma especificada ou não. Para cobrir tais situações, propõe-se um processo de teste que associa a técnica de injeção de falhas ao processo de testes de conformidade e ainda leve em conta as definições das normas ECSS no que concerne às necessidades das etapas de Qualificação Técnica e Validação Operacional.

A formalização para identificar requisitos de testes de conformidade e requisitos de falhas ainda está sendo estudada.

A especificação do sistema a ser testado, a partir da qual propósitos de teste são derivados, constam de documentos bem definidos para os protocolos da ISO [6]. Nos trabalhos da ISO/FMCT (*Formal Methods in Conformance Testing*) e de pesquisas de geração automática de teste a especificação deve estar em uma linguagem formal, seja MSC, SDL, Estelle, LOTOS.

Pretende-se explorar os diagramas em UML para obter os requisitos de teste, levando-se em conta a organização dos documentos, definida nas normas ECSS, uma vez que constam de um padrão de reconhecimento internacional para a área espacial.

Um outro aspecto a ser considerado na definição do processo de teste de conformidade é a definição de uma arquitetura de teste, a qual possa ser reutilizada em várias missões para reduzir custos e tornar os testes cada vez mais confiáveis.

Referências

1. Mazza, C. Standards: the foundation for Space I. T. In: *Workshop* Space Information Technology in the 21th Century. European Space Operations, Darmstadt, Germany, Setembro 2000. Disponível em: www.esoc.esa.de/pr/documents/workshops/it_2000/it_in_future/ESA_C_Mazza.ppt. Acesso em 07 julho 2003.

2. ESA PSS-05-0 –European Space Agency -ESA Software Engineering Standards – Issue 2, February 1991

3. Pressman, R.S. – Software Engineering – A Practioner´s Approach – McGraw-Hill Co., Inc. Fourth Edition, 1997.

4. ECSS-E-40B – European Cooperation for Space Standardization – Space Engineering – Software - Draft 1, 29 May 2002

5. Baumgarten, B.; Giessler, A. – OSI Conformance Testing Methodology and TTCN - Elsevier, 1994.

6. International Organization for Standardization/International Electrotechnical Commission – "CTMF- Conformance Testing Methodology and Framework", International Standard IS-9646. ISO, Geneve, 1991. Também: CCITT X.290-X.296.

7. Martins,E; Ambrosio, A.M; Mattiello-Francisco M.F. – ATIFS: a testing toolset with software fault injection. Workshop SofTest: UK Testing Research II. University of York: 4-5 September 2003.