

**Qualidade de Software:
Uma Experiência em Desenvolvimento de
Sistemas de Software dentro do
Subprojeto de Engenharia de Software do
Veículo Lançador de Satélites**

Marcelo José Ruy Lemes ⁽¹⁾
Maria Augusta Soares Machado ⁽²⁾
Adilson Marques da Cunha ⁽³⁾

Sumário

Os níveis de complexidade em Sistemas de Software têm crescido vertiginosamente nos últimos anos. Este fato tem chamado a atenção dos responsáveis pelo desenvolvimento de software para diversos problemas relacionados com a Qualidade dos Softwares produzidos. O Projeto do Veículo Lançador de Satélites (VLS) tem no seu Software um de seus principais componentes. A preocupação com a Qualidade e a Confiabilidade desse Software motivou a Gerência do Sub-Projeto de Engenharia de Software do VLS a adotar diversos procedimentos, a fim de minimizar esses problemas de modo a evitar o comprometimento do Software tanto como processo quanto produto. O objetivo deste trabalho é descrever esses procedimentos.

(1) Gerente de Garantia da Qualidade de Software do Subprojeto de Engenharia de Software do VLS o CTA/IAE

(2) Pesquisadora do Instituto de Pesquisas da Marinha (IPqM)

(3) Professor Adjunto do ITA/Divisão de Ciência da Computação e Consultor Técnico do Subprojeto e Engenharia de Software do VLS no CTA/IAE

1. Introdução

Dentre as atividades criativas executadas pelo homem, sem dúvida, uma das mais complexas é a de Engenharia de Software. Embora largamente debatida, tem sido bastante difícil de se estabelecer um consenso sobre a origem dessa complexidade.

Fred Brooks, em seu trabalho *No Silver Bullet* [1], afirma que a complexidade de software pode ser dividida em essencial e acidental. A complexidade essencial está relacionada com a natureza do software como um elemento lógico fruto de uma atividade criativa. A complexidade acidental é aquela ligada aos problemas que podem ser contornados através do controle do desenvolvimento do software.

A ausência de ferramentas adequadas e a lenta evolução dos métodos para o desenvolvimento de software em relação ao hardware e os níveis de requisitos crescentes dos sistemas é o principal ponto abordado por Brad Cox [2] para o problema do controle da complexidade de software. Cox aponta o desenvolvimento orientado a objetos como sendo a solução para essa situação.

Os sistemas aeronáuticos (figura 1) ilustram de forma bastante clara o aumento de complexidade de software que, para esse tipo de aplicação em específico, praticamente quadruplicou suas necessidades em termos de memória embarcada em pouco mais de duas décadas.

Legenda de alguns tipos de aeronaves:

É interessante notar-se, entretanto, que mesmo para softwares que têm missões críticas por realizar, e sendo estes desenvolvidos sob rígidos padrões, não raro encontram-se documentado casos de defeitos de software que levaram a grandes prejuízos e até mesmo a perda de vidas humanas [4].

Nesse moto-contínuo de necessidades gerando aplicações, que por sua vez geram novas necessidades, torna-se imperativa a adoção de procedimentos concretos que, de alguma forma, possam controlar a complexidade dos sistemas e garantir níveis aceitáveis de qualidade e confiabilidade. É baseado nesses aspectos que se fundamentam as principais preocupações da Gerência de Garantia da Qualidade (GGQ) do Subprojeto de Engenharia de Software do Veículo Lançador de Satélites (VLS) que está sendo realizado pelo Instituto de Aeronáutica e Espaço (IAE) do Centro Técnico Aeroespacial (CTA).

1.1 O VLS

O VLS é um foguete multiestágios, que tem a missão de colocar em órbita terrestre Satélites de Coleta de Dados (SCD) e Satélites de Sensoriamento Remoto (SSR), ambos desenvolvidos pelo Instituto Nacional de Pesquisas Espaciais (INPE).

Durante a fase que antecede o lançamento e no decorrer de seu voo, o VLS deve apresentar diversas capacidades, tais como: capacidade de satelização, capacidade de rastreamento, capacidade de autocontrole e autodestruição etc. Dentre essas capacidades especificadas, várias delas são implementadas no Software do Computador de Bordo. Esse fato confere ao Software do VLS uma natureza crítica, pois qualquer falha na execução de uma de suas funções poderá comprometer o êxito de toda a missão.

Este trabalho apresenta as principais idéias que vem norteando o desenvolvimento do Software do Computador de Bordo do VLS com vistas a obtenção de um produto de software com níveis de qualidade e confiabilidade aceitáveis para os níveis dos requisitos estabelecidos.

2. A Qualidade de Software dentro do Projeto VLS

Nesta seção são apresentados alguns aspectos do Subprojeto de Engenharia de Software do VLS sob o ponto de vista das atividades desenvolvidas pela Gerência de Garantia da Qualidade (denominada daqui em diante por Grupo de Qualidade - GQ).

2.1 A Posição do Grupo de Qualidade no Subprojeto de Engenharia de Software do VLS

Geralmente encontra-se na literatura que o setor de qualidade de uma organização que deve possuir uma posição de independência em relação aos setores de desenvolvimento, a fim de evitar decisões influenciadas por relacionamentos pessoais, afinidades técnicas etc e permitir pareceres comprometidos somente com os objetivos estabelecidos para a qualidade. Entretanto, esse fato, é geralmente levado ao extremo e o grupo de qualidade passa a ser considerado somente como um agente de cobrança, uma entidade geradora de burocracias e retrabalhos.

O GQ está ligado funcionalmente à Gerência do Subprojeto de Engenharia de Software do VLS. Com isso satisfaz-se o requisito de independência para a tomada de decisões. Contudo, a política de atuação junto aos demais grupos do Subprojeto (Gerência Administrativa e Gerência de Desenvolvimento) tem sido de estreita cooperação, onde os elementos do GQ tem agido como provedores de meios para que os demais integrantes do Subprojeto possam produzir dentro dos níveis de qualidade desejados.

2.2 Atividades Grupo de Qualidade

Para direcionar as atividades de qualidade dentro do Subprojeto de Engenharia de Software do VLS, procurou-se inicialmente caracterizar o software como **processo e produto**. Esse tipo de visão permite que a seleção das ferramentas de apoio ao processo de desenvolvimento e a avaliação de produtos sejam melhor determinadas, segundo suas características.

Em relação ao processo de desenvolvimento do software, estão associadas as tarefas de normatização, revisões, geração da documentação etc. Quanto ao produto (ou produtos) de software, estão associadas as tarefas de definição e aplicação de um modelo para avaliação da qualidade.

2.2.1 O Software do VLS enquanto Processo

O desenvolvimento de software de aplicação em projetos espaciais é uma atividade recente em termos de Brasil. Como em toda inovação, existem riscos a assumir. Uma das preocupações do Grupo de Qualidade de Software do VLS tem sido quanto a esses riscos, uma vez que os requisitos da missão são rigorosos e o tempo para a sua realização é relativamente curto.

Visando minimizar os riscos da falta de experiência na construção desse tipo de software, foi adotada uma metodologia estruturada dirigida para sistemas de tempo real para apoiar o seu desenvolvimento [5,6]. Essa metodologia estruturada tem sido utilizada juntamente com técnicas de prototipação, principalmente durante a fase de levantamento de requisitos, que pressupõe a criação de um modelo que evolui conforme a evolução dos requisitos (figura 2).

No caso de um software com o nível de requisitos como o do VLS, a utilização de uma metodologia de desenvolvimento é necessária mas não suficiente para garantir níveis aceitáveis de qualidade e confiabilidade. O emprego de normas para apoiar o desenvolvimento de software, tem se tornado consenso mundial, principalmente quando o software envolvido pode ser um considerado crítico [8]. Para o desenvolvimento do software do VLS, estão sendo adotadas um conjunto de normas utilizadas pelo Departamento de Defesa Americano, conhecidas por DoD-Std - *Department of Defense Standard*. Dentro desse "pacote" de normas destaca-se a DoD-Std-2167A ou simplesmente 2167A, a qual tem fornecido as linhas mestras para o projeto [9].

A 2167A, apesar de rigorosa quanto a sua aplicação a projetos de software, tem se mostrado um excelente instrumento de fomento à produtividade. Um ponto que ilustra essa característica e que a princípio parece paradoxal é o seu próprio rigor. Por exemplo, para a finalização de cada etapa do processo de desenvolvimento existe um grande número de requisitos a serem atendidos, mas que conseqüentemente reduz bastante o retrabalho esperado nas etapas subseqüentes.

Um dos pontos preconizados pela 2167A é a classificação do software em questão em categorias. Dentre as categorias propostas pela norma, o software do VLS foi enquadrado em 2 delas:

- (a) Software Embarcado; e
- (b) Software de Apoio.

Por software embarcado, foi considerado todo item de software carregado a bordo do VLS, e por software de apoio todo o software desenvolvido para suportar o desenvolvimento da parte embarcada, como por exemplo os programas de testes e ferramentas para o controle de configuração. O principal objetivo dessa divisão é o estabelecimento de diferentes níveis de requisitos quanto a documentação a ser gerada, exigências de revisões formais, controle de configuração etc.

A partir dessa classificação, é então sugerido uma adequação - *Tayloring* - da norma às características do software. Essa adequação deve ocorrer em 3 níveis:

- (a) Adequação da norma propriamente dita;
- (b) Seleção dos documentos preconizados na norma - *Data Item Description (DID)*; e
- (c) Seleção dos itens de cada documento selecionado, segundo sua aplicação ao projeto.

O processo de adequação atribui a 2167A a capacidade de vestir exatamente as necessidades de cada projeto. A 2167A está sendo adotada para atender, principalmente, às necessidades de desenvolvimento do Software do VLS. A Garantia da Qualidade de Software, além de beneficiar-se dessas características, possui uma norma própria, a IEEE 983/86 que é voltada para as necessidades específicas desse tipo de atividade.

É através da IEEE 983/86 que vem sendo elaborado o Plano de Garantia da Qualidade de Software do VLS (PGQS/VLS). Este plano deverá conter todos os procedimentos relativos à qualidade de software adotados para o projeto.

Como foi mencionado acima, o tempo para a construção do software do VLS é relativamente curto. Entretanto, a aplicação das metodologias selecionadas bem como a obediência a normas, tem exigido um determinado tempo para a sua compreensão, tempo este insuficiente para o cumprimento do atual cronograma do projeto. A solução adotada então foi a adoção da técnica de Treinamento Durante a Realização do Trabalho (TDRT) - *On The Job Training*. Com certeza esta não é a melhor estratégia a ser seguida, contudo, dentro das restrições impostas de prazos e recursos disponíveis, a aplicação da técnica de TDRT no âmbito do Subprojeto de Engenharia de Software do VLS tem apresentado resultados satisfatórios.

2.2.2 O Software do VLS enquanto Produto

O enfoque de produto de software que vem sendo adotado no Sub-Projeto de Engenharia de Software do VLS não se restringe somente ao produto final. Mesmo sob a orientação da 2167A, diversos produtos intermediários devem ser apresentados e criteriosamente avaliados antes do prosseguimento do desenvolvimento como, por exemplo, os documentos gerados em cada fase do ciclo de vida. Estas avaliações visam manter a aderência do software produzido com relação aos requisitos iniciais do projeto.

Um conjunto de revisões estão sendo estabelecidas. Cada uma dessas revisões tem um objetivo bem determinado, como por exemplo avaliar um documento de projeto ou código fonte. Este tipo de procedimento tem sido adotado visando antecipar, o quanto antes possível, a identificação de problemas potenciais.

Os testes de código são condição necessária a obtenção de programas confiáveis, entretanto, têm sido considerados como insuficientes para garantir a qualidade do produto, principalmente tratando-se de sistemas embarcados. Esse posicionamento se dá principalmente pela incapacidade dos testes em assegurar a inexistência de erros no código.

O testes podem ser tornar mais eficazes à medida que o código a ser testado possua baixa complexidade. Para tanto, tem-se adotado duas medidas de software, com finalidade de servir como indicativo do nível de complexidade do código produzido. Essas métricas foram derivadas diretamente dos modelos propostos por McCabe [10] e Halstead [11].

Apesar da complexidade do software ser um item de especial importância dentro do contexto do projeto VLS, este é apenas um dos itens que vêm sendo avaliados. O modelo de avaliação que vem sendo utilizado (figura 3) baseia-se nos trabalhos de Boehm [12], McCall [13] e na norma ISO-9126 [14], que também é derivada dos trabalhos dos referidos autores. A aplicação desse modelo e qualquer outro procedimento referente a qualidade de software é regida pelo Plano de Garantia da Qualidade de Software (PQGS).

A referência a características e sub-características de software segue a nomenclatura adotada proposta pela norma ISO-9126. Contudo, esses termos podem ser facilmente mapeados para os modelos de Boehm e McCall onde são referenciados como fatores e critérios.

Dentre as diversas características de software que representam os objetivos da qualidade a serem alcançados, a Confiabilidade de Software possui uma conotação especial. A próxima seção desse trabalho, descreve as principais atividades que se pretende estabelecer para este quesito.

3. Metodologia a ser Utilizada para a Análise de Confiabilidade do Software do VLS

Devido à própria natureza do software que está sendo produzido, a confiabilidade do software do VLS torna-se crítica para o êxito da missão.

Para a determinação da Confiabilidade do Software, pretende-se estabelecer as seguintes atividades:

Serão determinados os caminhos críticos dos submódulos dos processos funcionais do software do VLS, através da técnica de Análise por Árvores de Falhas (FTA) [15];

Serão ajustadas as distribuições de Weibull, características de cada um dos eventos, individualmente aos eventos básicos considerados;

Também será realizada uma análise dos modos de falha relevantes, bem como suas causas e efeitos sobre o sistema como um todo e para as funções principais, onde não se admite nenhuma degradação (aquelas cujo efeito da falha seja catastrófico);

A partir desse tipo de estudo, serão determinadas as medidas compensatórias (ou redundantes) necessárias ao software; e finalmente

Serão identificadas as necessidades de medidas preventivas (sinais tipo "alarmes") para os casos de se evitar os possíveis modos de falha que causam efeitos catastróficos.

4. Conclusões

O domínio das técnicas de construção de software complexos, em termos de Brasil, apresenta-se como questão estratégica não só em relação à competitividade entre empresas mas também para se estabelecer a independência tecnológica do país nesse setor. As experiências relatadas, de caráter puramente brasileiro, tem sido poucas nesse setor, o que, de certa forma tem dificultado o intercâmbio entre pesquisadores.

O Subprojeto do Software do VLS, por sua complexidade inerente, apresenta-se como um grande desafio e, por isso mesmo, tem sido cercado de grande cuidado. As idéias aqui apresentadas representam um primeiro esforço do GQ para fornecer meios adequados e dirigidos à qualidade aos integrantes do Subprojeto de Engenharia de Software do VLS responsáveis pelo desenvolvimento dos sistemas de software do veículo lançador de satélites.

Quanto à metodologia utilizada para a análise de confiabilidade de software, esta será validada após a coleta de dados realizada durante os testes dos sistemas de software do VLS.

Abstract

Os níveis de complexidade em sistemas de software têm crescido vertiginosamente nos últimos anos. Este fato tem trazido problemas relacionados com a qualidade dos softwares produzidos. O Projeto do Veículo Lançador de Satélites (VLS) tem no seu software de bordo um de seus principais componentes.

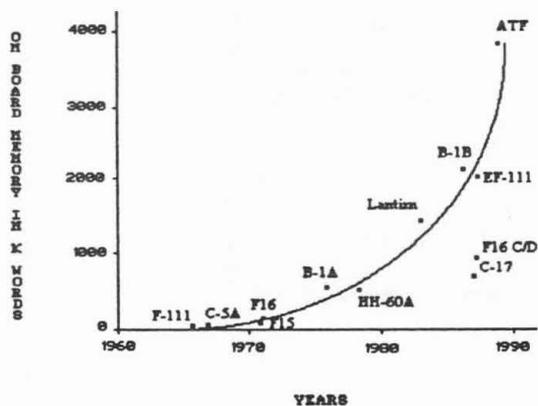
A preocupação com a qualidade e confiabilidade desse software motivou a Gerência do Subprojeto de Engenharia de Software do VLS a tratar o problema de maneira especial estabelecendo formalmente uma Sub-Gerência de Garantia da Qualidade de Software (GGQS), responsável pelo estabelecimento dos procedimentos de visando a garantir níveis aceitáveis de qualidade e confiabilidade.

uma separação entre processo adotar diversos procedimentos, a fim de minimizar os problemas que possam comprometer o produto. O objetivo deste trabalho é descrever estes procedimentos.

Referências Bibliográficas

- [1] Brooks, F.P. "No Silver Bullet - Essence and Accident of Software Engineering". *IEEE Computer*, April 1987, pp. 10-19.
- [2] Cox, B. "There is a Silver Bullet". *Byte*, October 1990, pp. 209-218.
- [3] Boehm, B. "Software Reliability - Measurement and Management", *Proceedings of AIAA Software Management Conference, Los Angeles*, June 1990.
- [4] Forester, T.; P. Morrison. "A Insegurança do Computador e a Vulnerabilidade Social". *Revista de Administração de Empresas*, Nov/Dez 1991, pp. 73-83.
- [5] Hatley, D.J.; I.A. Pirbhai. *Estratégia para Especificação de Sistemas de Tempo Real*. Makron Books, 1990.
- [6] Ward, P.T.; S.J. Mellor. *Structured Development for Real-Time Systems*. Yourdon Press/Prentice Hall, 1985.
- [7] Moura, C.A.T. "O Desenvolvimento de um Simulador de Dinâmica de Vôo para o Veículo Lançador de Satélites - VLS". Tese de Mestrado em andamento. ITA, Divisão de Ciência da Computação, 1993.
- [8] Charette, R.N. *Software Engineering Environments*. Intertext Publications Inc, 1986.
- [9] DoD-Std-2167A - Defense System Software Development, 1988.
- [10] McCabe, T.J. "A Complexity Measurement". *Proceedings of the 2nd International Conference on Software Engineering*, June 1977.
- [11] Halstead, M. *Elements of Software Science*. Elsevier, 1977.
- [12] Boehm, B.M. Quantitative Evaluation of Software Quality. *IEEE - Proceedings of Second International Conference of Software Engineering*, 1976.
- [13] McCall, J.A. *An Introduction to Software Quality Metrics*". Do livro "Software Quality Management". Petrocelli Books, 1978.
- [14] ISO-9126. Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for their Use, 1991.
- [15] Braga, M.J.F.; M.A.S. Machado. "Confiabilidade de Software: Uma Abordagem Utilizando a Técnica de Análise por Árvores de Falhas". Pesquisa Naval: Suplemento Especial da *Revista Marítima Brasileira*. nº 5, Outubro 1992.

Figuras



- ATF - Advanced Tactical Fighter (europcu)
- B-1B - Bombardeiro Estratégico (americano)
- C-5A - Cargueiro (americano)
- F-111 - Fighter (americano)

Figura 1 - Software Embarcado em Sistemas Aeronáuticos [3]

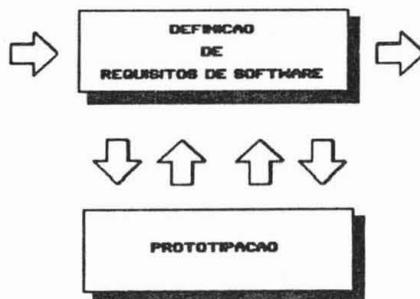


Figura 2 - Definição de Requisitos através de Prototipação

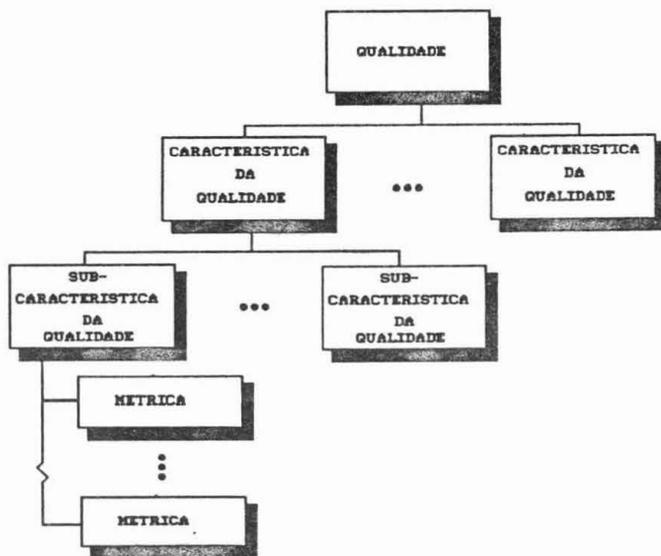


Figura 3 - Modelo de Avaliação de Produto de Software