

O DESENVOLVIMENTO DE SISTEMAS COMPLEXOS: APLICAÇÃO DO MODEL BASED DESIGN E MODEL BASED TESTING NA PRODUÇÃO DE AERONAVES

Antonio Cesar Galhardi
prof.galhardi@fatecjd.edu.br
CEETPS

Carlos Eduardo Cassiano Pinto
du.cassiano@uol.com.br
CEETPS

Resumo: Os processos tradicionais de desenvolvimento de sistemas são focados em documentações formais e estáticas. Enquanto empresas inovadoras introduzem novas técnicas e metodologias para o desenvolvimento de sistemas, afim de melhor atenderem o problema a ser solucionado. Este artigo discute técnicas e práticas inovadoras no desenvolvimento de sistemas complexos. O artigo apresenta os conceitos de Model Based Design - MBD e Model Based Testing - MBT; e como eles podem auxiliar na diminuição do ciclo de desenvolvimento de produto, diminuindo assim o time-to-marketing. Por fim o artigo sugere uma transição da técnica em

Palavras Chave: Model based testing - Model based desig - Simulação - Inovação tecnológica - Modelagem

1. INTRODUÇÃO

A busca por técnicas de testes de sistemas que permitam melhorar a eficiência no desenvolvimento do produto é fundamental para a sobrevivência das empresas que desenvolvem sistemas complexos; uma vez que as mesmas requerem que métodos sistemáticos de aplicação de testes sejam aplicados durante o projeto. Porém pouco se trata das aplicações práticas na elaboração de testes automáticos; quanto eles realmente agregam valor ao desenvolvimento e como realizá-los de forma organizada e eficiente durante todo o processo de desenvolvimento do produto.

O campo de validação e verificação de sistemas complexos vem atualmente sendo repensada em função da grande evolução nos processos de desenvolvimento do produto e a incorporação de técnicas inovadoras no processo de V&V. Um grande exemplo deste fato é a utilização de modelagem e simulação no desenvolvimento de sistemas complexos; sendo até mesmo considerada como uma inovação na área de desenvolvimento de sistemas.

O desenvolvimento de sistemas complexos como modernas aeronaves, satélites ou reatores nucleares, requerem métodos sistemáticos para o seu desenvolvimento e projeto. Tipicamente, o desenvolvimento se inicia com a captura de requisitos de alto nível que são registrados em formato textual. Estes requisitos geralmente formam a base para a especificação dos sistemas, que gradualmente são refinados e implementados por meio de um projeto detalhado.

Durante a implementação do projeto detalhado, os engenheiros projetistas utilizam cenários de testes estruturados para validarem os sistemas desenvolvidos e todo o seu ambiente contra os requisitos originais previamente definidos. A construção de sistemas complexos requer alta exigência em segurança, e que por sua vez exigem rigorosos processos de especificação do comportamento do sistema; além de uma campanha de testes para garantir que a implementação satisfaça totalmente os requisitos do sistema.

Em vista deste cenário, técnicas de modelagem e simulação tornam-se um fator significativo dentro do processo de desenvolvimento do produto; tais técnicas são conhecidas como: *Model-Based-Design – MBD* e *Model-Based-Testing – MBT*; que podem ser aplicadas para reduzir o custo do projeto do sistema e até mesmo o tempo do *Time-to-Marketing*, principalmente quando aplicadas desde o início do processo de desenvolvimento.

Com a facilidade de exercitar o sistema em desenvolvimento por meio de modelos é comum a “explosão” no número de testes possíveis e desejáveis a serem realizados durante as várias etapas do desenvolvimento, desde a validação de requisitos até os testes finais; trazendo a tona a necessidade de “frameworks” e recursos computacionais de hardware e software para suportar esta necessidade.

O objetivo geral deste artigo é caracterizar os processos de desenvolvimento de sistemas complexos, particularmente o *Model-Based-Design – MBD* e o *Model-Based-Testing – MBT*.

Os objetivos específicos tendem a responder as seguintes perguntas básicas de pesquisa:

- Os conceitos de “*Model-Based-Design*” (MBD) e “*Model-Based-Testing*” (MBT) são efetivos em quais etapas do processo de desenvolvimento do sistema?
- Como planejar estes testes ao longo do desenvolvimento do produto?
- É possível incorporar ao tradicional processo de desenvolvimento em “V” as técnicas de *Model-Based-Testing*?
- Como a computação distribuída poderá auxiliar na solução de gargalos de hardware e software, dado ao crescente número de testes?

2. REFERENCIAL TEÓRICO

2.1 PROCESSO DE VALIDAÇÃO E VERIFICAÇÃO

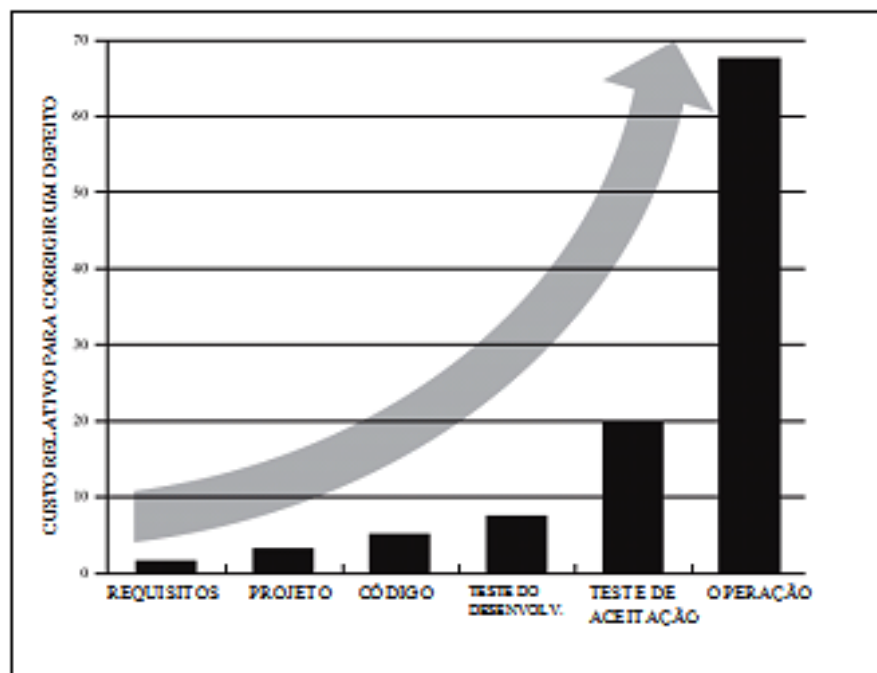
No final da década de 1970, 50% do tempo e 50% do custo eram empregados para testar programas ou sistemas. Hoje, apesar de linguagens de programação mais avançadas, e novos ambientes de desenvolvimento; os testes continuam tendo um papel fundamental para garantir a qualidade do produto. Para Myers (2011), um bom processo de validação e verificação é atingido por meio de testes.

Entende-se verificar como confirmar com provas objetivas que requisitos especificados foram cumpridos; enquanto a validar é confirmar com provas objetivas que requisitos particulares para um determinado uso foram cumpridos. Para Pressman (2007): "*Verification*": "*Are we building the product right?*" e "*Validation*": "*Are we building the right product?*". Ainda para o autor, testar é o processo de executar um sistema com o objetivo de achar erros, "*Testing is the process of executing a program with the intent of finding errors*".

Ainda Pressman (2007) afirma que o processo de validação é muito utilizado durante a fase de levantamento de requisitos, para certificar que o sistema a ser construído irá ser capaz de atender aos requisitos de mercado. Já na verificação ela ocorre praticamente em todas as fases do desenvolvimento, para garantir que os produtos de uma dada fase implementem em sua totalidade as entradas e saídas especificadas, ou seja, o produto foi construído corretamente.

A citação: Teste e depois codifique (MYERS, 2010), mostra claramente a importância da realização de testes em todas as etapas do desenvolvimento do software; um conceito que fundamenta a presente pesquisa, no sentido de que a atividade de testes não seja custosa.

A Figura 1 é uma figura clássica na literatura que demonstra a evolução do custo da detecção de um erro ao longo do desenvolvimento do produto de software, utilizando como base um processo tradicional de desenvolvimento de sistemas.



Fonte: ADAPTADO DE: Burnstein - Practical Software Testing
Figura 1 - Custo de detecção de defeitos ao longo do desenvolvimento

2.2 MODEL BASED DESIGN AND TESTING

Processos baseados em modelagem e simulação tem sido largamente utilizados no desenvolvimento de sistemas complexos, a fim de minimizar os testes realizados em protótipos que usualmente possuem custo elevado e também para agilizar as fases de desenvolvimento do processo. Atualmente a EMBRAER, fabricante brasileira de aeronaves de pequeno e médio porte utiliza tal tecnologia em diversas etapas do desenvolvimento do produto, desde modelos para ensaio em túnel de vento, modelos para prototipação rápida de sistemas e em ferramentas de simulação *Computer Aided Engineering – CAE* (SILVA 2003).

Segundo o escritório de modelagem e simulação do departamento de defesa dos Estados Unidos, os modelos são representações físicas, matemáticas ou lógicas de um sistema, entidade, fenômeno ou processo. Modelar consiste basicamente em aplicar padrões e rigorosas metodologias para a criação e validação de um modelo. Esta afirmação refere-se aos modelos de forma geral; no entanto, no que se refere a modelagem e simulação em engenharia, é possível se destacar algumas aplicações como: desenvolvimento de controladores, especificação e validação de requisitos de sistemas, simulação computacional, cálculos estruturais, entre outras (DEFENSE, 2007).

Para Ferraz (2010): modelos podem ser entendidos como uma simulação computacional que modela o comportamento de algum sistema real ou imaginário, ao longo do tempo. Tais simulações são amplamente utilizadas para análise de sistemas e processos dinâmicos, como controle de tráfego aéreo, redes de comunicação, processos físicos e biológicos, e no desenvolvimento do projeto de sistemas em geral.

A adoção das técnicas de “*Model-Based-Design*” e “*Model-Based-Testing*” vem sendo altamente utilizada no desenvolvimento de sistemas computacionais embarcados, diversas companhias tem observado as vantagens obtidas nos processos de desenvolvimento, diminuição do tempo de desenvolvimento, aumento da qualidade final do produto e uma significativa redução de custos (ALTHOLFF, 2007).

Um dos grandes benefícios da utilização de modelos é a capacidade de desenvolver algo que pode ser entendido e utilizado por todas as pessoas envolvidas num projeto, facilitando reuso e testes. Segundo Altholff (2007, p78) “.....modelos podem ser tratados como especificações executáveis do sistema”.

Com o avanço no desenvolvimento de aplicações de sistemas embarcados as técnicas de MBD e MBT, normalmente incorporadas no desenvolvimento destes produtos, se tornaram cada vez mais difundidas. O desenvolvimento de aplicações embarcadas normalmente exige: automação dos testes devido a quantidade requerida; projeto e testes orientados a modelos; modelagem matemática para atender aos requisitos de confiabilidade e desempenho; e componentes desenvolvidos para reuso (GRECA, 1999).

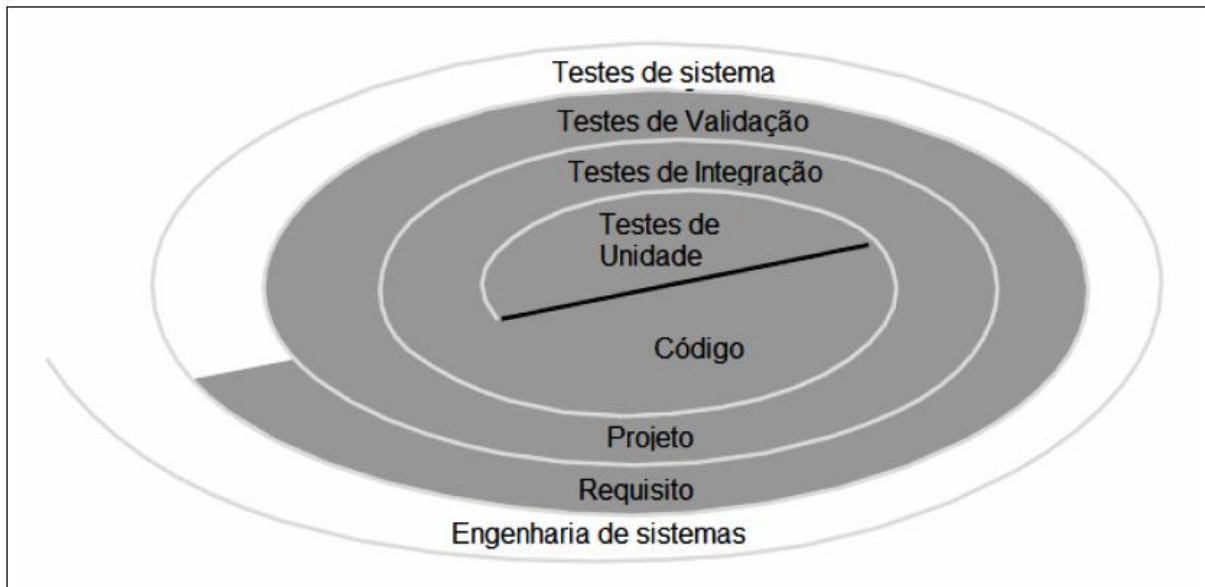
2.3 O PLANEJAMENTO DO PROCESSO DE VALIDAÇÃO E VERIFICAÇÃO

O Planejamento do processo de validação e verificação normalmente é realizado por meio de um Plano de Testes, que é um documento que apresenta: uma breve introdução específica os tipos de testes que serão executados ao longo do processo de desenvolvimento do produto, os objetivos destes testes, a metodologia utilizada, as técnicas e tipos de teste a serem realizados. (BURNSTEIN, 2009)

O Plano de Teste dá suporte para atingir aos objetivos dos testes, e geralmente é desenvolvido por um especialista de testes ou de qualidade. (VIJAYKUMAR, 2009).

A estratégia de testes pode ser vista em um contexto espiral, conforme se observa na Figura 2. Os testes de unidade se iniciam no centro da espiral e concentram-se em cada

unidade ou componente do software implementado no código fonte. O processo de teste progride de forma espiral passando aos testes de integração, onde o foco principal é o projeto e a construção da arquitetura do sistema. Efetuando mais uma volta na espiral encontra-se os testes de validação onde os requisitos são validados de acordo com o produto desenvolvido, e finalmente, após mais uma volta em torno da espiral chega-se aos testes de sistema, onde todas as partes do sistema são testadas como um todo. De forma geral a cada volta na espiral o escopo planejado dos testes é incrementado e concluído. (PRESSMAN, 2007).



Fonte: ADAPTADO DE Pressman 2007 - A Practitioner's Approach
Figura 2 - Estratégia de teste de software de forma espiral

2.4 COMPUTAÇÃO DISTRIBUÍDA

O contexto da computação distribuída tratado neste artigo utiliza o conceito de *Distributed High Throughput Computing* – DHTC, que trata de sua efetiva utilização para o gerenciamento e a exploração de todo recurso de computação disponível.

O Projeto Condor, criado em 1985 é um exemplo típico de uma de *Distributed High Throughput Computing*, é especializado no *High-Throughput Computing* clássico, que é um termo de ciência da computação para descrever o uso de muitos recursos computacionais durante longos períodos de tempo, para realizar uma tarefa computacional. (THAIN, TANNENBAUM e LIVNY, 2005). Este projeto consiste em um time de aproximadamente 35 professores, funcionários em tempo integral e estudantes que: enfrentam os desafios de engenharia de software em um ambiente UNIX / Linux / NT distribuído, estão envolvidos em colaborações em grids nacionais e internacionais, interagem ativamente com entidades e usuários acadêmicos e comerciais, mantem e suportam grandes ambientes distribuídos em produção e educam e treinam estudantes. É um sistema integrante de muitos grids computacionais ao redor do mundo. O sistema condor é um sistema repleto de “*features*” como: mecanismo de políticas de agendamento flexíveis via ClassAds, preempção, suspensão, requisitos, preferências, grupos, quotas, *fair-share* (compartilhamento), *system hold*, facilidades para gerenciar tanto CPUs dedicadas (clusters) quanto recursos não dedicados (desktops), não são necessário um “*shared file-system*”, Gerenciamento de Workflow (interdependências de tarefas), Suporte a muitos tipos de *jobs* – serial e paralelo Tolerante a falha: pode sobreviver a falhas, interrupções de rede, sem ponto único de falha,

APIs de desenvolvimento: via SOAP / *web services*, DRMAA (C), Perl *package*, GAHP, *flexible command-line tools*, MW (*Master Worker*) para plataformas Linux, Windows, MacOS, Solaris, HP-UX, AI. Ainda para os mesmos autores, dentre os principais usuários comerciais do projeto Condor de “*grid-computing*” estão as empresas: JP Morgan Chase (Financeiro), Bank of America (Financeiro), Lockheed Martin (Aeroespacial, Segurança), Airbus (Aeroespacial), Eli Lilly (Farmacêutica), Pfizer (Farmacêutica), Hartford (Financeiro); e na área acadêmica universidades como: Wisconsin-Madison, UNESP, Pardue, Clemson, Nebraska-Lincoln, Marquette, Northeastern, Notre Dame, Indiana, Louisiana, UC San Diego, Texas, Florida, Ecole Polytechnique Fédérale de Lausanne, INFN - Instituto Nazionale di Fisica Nucleare, Universitat Autònoma de Barcelona.

3. METODOLOGIA

Esta pesquisa caracteriza-se por uma metodologia Qualitativa-Exploratória-Descritiva, associado ao relato de aplicação do *grid-computing*, em um fabricante brasileira de aeronaves regionais e executivas de médio porte.

A pesquisa bibliográfica contempla os conceitos e definições relacionados ao tema, além de exemplos de aplicações do *Model-Based-Design*, em do ramo aeronáutico e de sistemas embarcados.

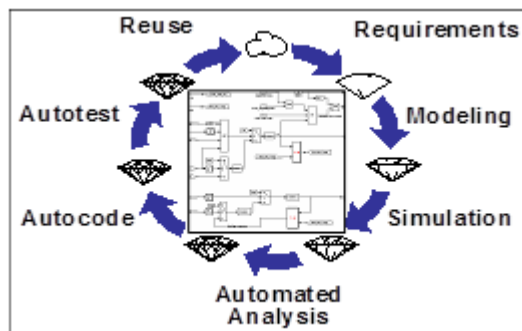
A análise das aplicações nas etapas do processo de desenvolvimento do produto e os resultados obtidos permitiram obter subsídios para encontrar as respostas as perguntas básicas de pesquisa.

4. RESULTADOS E DISCUSSÃO

4.1 O PROCESSO DE MBD E MBT NO DESENVOLVIMENTO DO PRODUTO

Os processos de desenvolvimento baseados em modelos e testes, MBD e MBT, possibilitam desenvolver diversas soluções alternativas para validar a melhor tomada de decisão de projeto, utilizando-se de laboratórios e simulações. (BAKER e LONG, 2006).

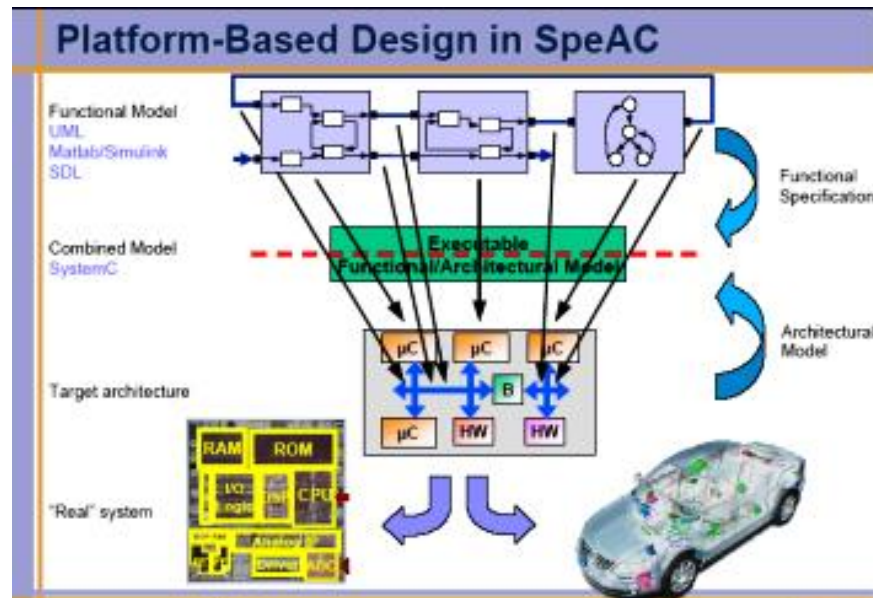
Uns dos principais objetivos da utilização de MBD e MBT são: redução do ciclo de desenvolvimento do produto; a antecipação da detecção de defeitos; facilitar a compreensão dos testes de sistemas; reuso dos artefatos de teste ao longo das etapas de desenvolvimento. Observa-se na Figura 3 abaixo um exemplo de desenvolvimento baseado em modelos utilizado pela Rockwell Collins.



Fonte: Universidade de Minnesota

Figura 3 - Modelo de desenvolvimento MBD utilizado pela Rockwell Collins

Os modelos geralmente seguem os mesmos níveis de abstração da etapa de desenvolvimento; por exemplo, durante a etapa de elaboração e validação de requisitos não se espera nesta fase, utilizar modelos detalhados do sistema e sim modelos que representem o mesmo nível de abstração dos requisitos. A Figura 4 ilustra um exemplo de evolução dos modelos conforme o a evolução do desenvolvimento do produto num processo de desenvolvimento do *Electronic Control Unit – ECU*.



Fonte: Universidade de Minesota (2010)

Figura 4 - Exemplo da evolução da maturidade dos modelos no desenvolvimento de ECU para veículos

A Eficiência e Eficácia das técnicas de *MBD* e *MBT* podem ser observadas no Quadro 1, por meio do registro dos ganhos obtidos por empresas de desenvolvimento de sistemas, de diferentes ramos de atuação.

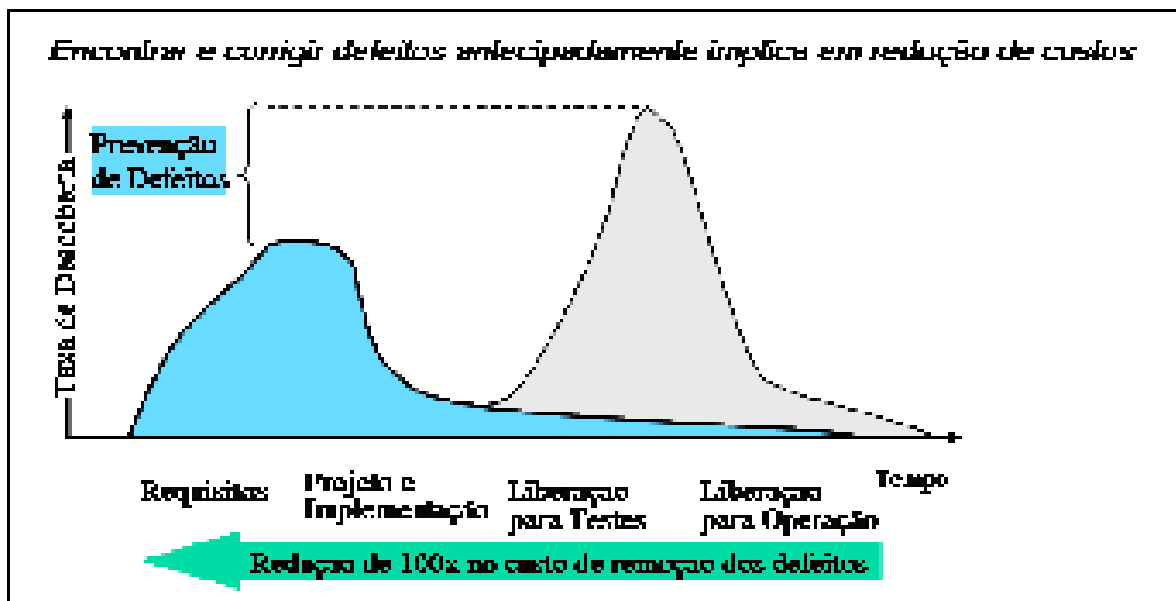
Quadro 1- Ganhos de qualidade e produtividade com uso de *MBD* e *MBT*

Company	Product	Tools	Specified & Autocoded	Benefits Claimed
Airbus	A340	SCADE With Code Generator	<ul style="list-style-type: none"> 70% Fly-by-wire Controls 70% Automatic Flight Controls 50% Display Computer 40% Warning & Maint Computer 	<ul style="list-style-type: none"> 20X Reduction in Errors Reduced Time to Market
Eurocopter	EC-155/135 Autopilot	SCADE With Code Generator	<ul style="list-style-type: none"> 90% of Autopilot 	<ul style="list-style-type: none"> 50% Reduction in Cycle Time
GE & Lockheed Martin	FADEC Engine Controls	ADI Beacon	<ul style="list-style-type: none"> Not Stated 	<ul style="list-style-type: none"> Reduction in Errors 50% Reduction in Cycle Time Decreased Cost
Schneider Electric	Nuclear Power Plant Safety Control	SCADE With Code Generator	<ul style="list-style-type: none"> 200,000 SLOC Auto Generated from 1,200 Design Views 	<ul style="list-style-type: none"> 8X Reduction in Errors while Complexity Increased 4x
US Spaceware	DCX Rocket	MATRIXx	<ul style="list-style-type: none"> Not Stated 	<ul style="list-style-type: none"> 50-75% Reduction in Cost Reduced Schedule & Risk
PSA	Electrical Management System	SCADE With Code Generator	<ul style="list-style-type: none"> 50% SLOC Auto Generated 	<ul style="list-style-type: none"> 60% Reduction in Cycle Time 5X Reduction in Errors
CSEE Transport	Subway Signaling System	SCADE With Code Generator	<ul style="list-style-type: none"> 80,000 C SLOC Auto Generated 	<ul style="list-style-type: none"> Improved Productivity from 20 to 300 SLOC/day
Honeywell Commercial Aviation Systems	Primus Epic Flight Control System	MATLAB Simulink	<ul style="list-style-type: none"> 60% Automatic Flight Controls 	<ul style="list-style-type: none"> 5X Increase in Productivity No Coding Errors Received FAA Certification

Fonte: Universidade de Minnesota 2010

Com os resultados capturados no Quadro I é possível deduzir a aplicabilidade da Figura 5, onde a aplicação de modelos e testes no início do desenvolvimento antecipa a detecção de defeitos. Um fato bem interessante é que os defeitos continuarão existindo, porém em menor amplitude, se descobertos antecipadamente.

Observa-se ainda na Figura 5 dois movimentos linearmente dependentes: um de antecipação e outro de redução da área abaixo da curva. Para que isto seja possível, facilitadores destes movimentos devem ser considerados na empresa como: a competência dos funcionários; o uso de modelagem e simulação; o reuso e automação nos processos de validação e verificação.



Fonte: Safford Software Technology Conference 2000.

Figura 5: Curva cinza e curva Azul

4.2 PLANEJAMENTO DOS TESTES AO LONGO DO DESENVOLVIMENTO UTILIZANDO MBT.

Planejar os testes significa desenvolver métodos, procedimentos e ferramentas para criar, executar e analisar os sistemas a serem testados ao longo do ciclo do desenvolvimento do produto. O “*Model-Based-Testing*” deve otimizar as oportunidades de automação dos testes e as suas capacidades, possibilitar reuso dos artefatos de testes em todo o ciclo de desenvolvimento do produto (UTTING, 2007).

A Utilização de MBD e MBT possibilita a escrita de testes efetivos para todas as principais etapas de desenvolvimento do produto, já no início do desenvolvimento na especificação de requisitos através de modelos já é possível identificar a dificuldade de implementação de determinados requisitos.

Desenvolver testes para validação e verificação de sistemas complexos exige um grande conhecimento sobre o sistema desenvolvido. Desenvolvendo testes em paralelo com o projeto e desenvolvimento do sistema, possibilita o engajamento do engenheiro de testes ainda nas fases preliminares do desenvolvimento do sistema facilitando o processo de Validação e Verificação. Esta colaboração resulta numa verificação antecipada do produto reduzindo custos e aumentando o tempo para correção de erros sem afetar a entrega final do produto.

O reuso de testes é a palavra chave da metodologia "*Model-Based-Design*". A capacidade de executar os mesmos testes já realizados por meio de modelos nos hardwares reais. Possibilita saber exatamente como os componentes físicos devem se comportar no laboratório. Se os testes encontrarem valores inesperados, o reuso dos testes nos possibilita de forma fácil e rápida interagir com o projeto e solucionar o problema.

Tempo e custo geralmente são os limites da quantidade e variabilidade de características testadas no produto. A possibilidade de realização de testes em ambientes simulados possibilita a simulação de cenários de teste de forma rápida e até mesmo de forma paralela, explorando os domínios do problema em laboratórios,

Um dos principais princípios dos padrões de qualidade é a documentação. São boas práticas numa campanha de validação e verificação, documentar requisitos, processos e resultados. Enquanto a documentação é vista como uma atividade que demanda bastante tempo; várias ferramentas proporcionam documentação automática para modelos e testes (IEEE, 2008).

Outro fator importante é documentar a rastreabilidade entre os requisitos para os modelos e os testes, para se permitir identificar que todos os requisitos estão sendo modelados e validados apropriadamente. Este mapeamento auxilia: na validação antecipada dos modelos, na verificação dos modelos de simulação e na verificação dos testes.

4.3 TRANSIÇÃO DA METODOLOGIA EM "V" PARA MÚLTIPLOS "V" NO PROCESSO DE VALIDAÇÃO E VERIFICAÇÃO POR MEIO DE "*MODEL-BASED-TESTING*"

O ciclo de desenvolvimento tradicional em "V" observado na grande maioria das empresas que não utilizam MBD e MBT pode ser observado como na figura 10 abaixo. O processo de desenvolvimento feito sem estas técnicas concentram as atividades de validação e verificação na segunda etapa do "V" onde protótipos e equipamentos já foram desenvolvidos e os custos de correção de problemas já estão elevados. Neste exemplo verifica-se o processo "tradicional" de desenvolvimento de uma aeronave, utilizado anteriormente por empresas como AIRBUS e EMBRAER.

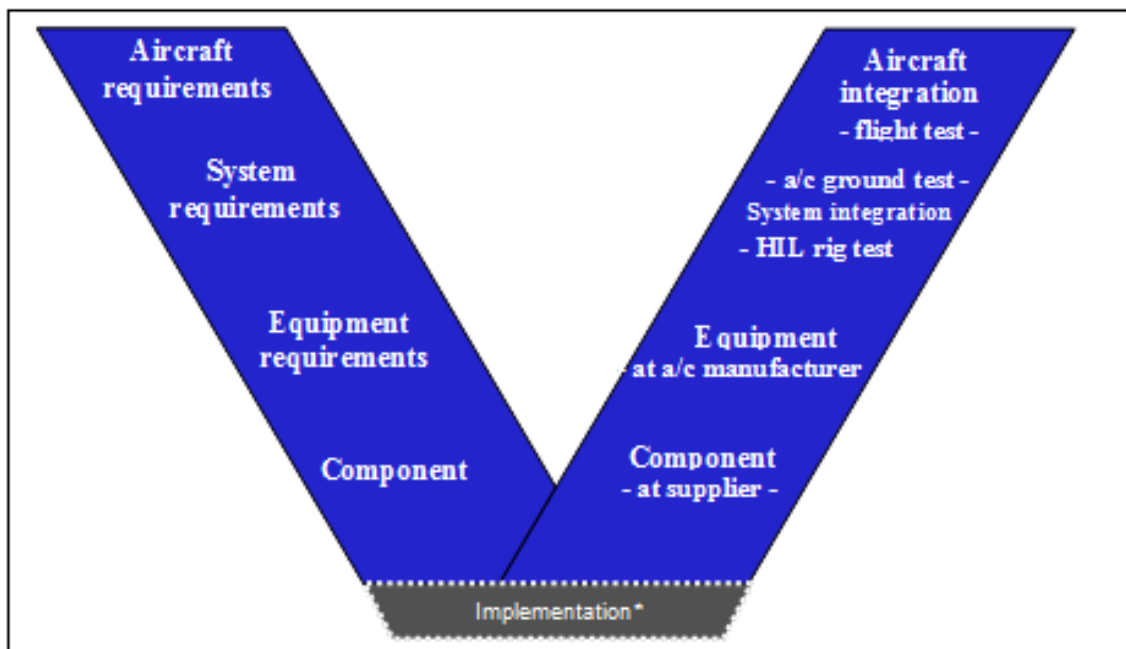


Figura 6 - Processo tradicional de Desenvolvimento em "V"

O ciclo de desenvolvimento utilizando as técnicas de MBD e MBT permite a validação e verificação contínua ao longo de todas as etapas do processo, conforme descrito anteriormente. Na Figura 7 apresenta-se o modelo de múltiplos “V”, que possibilitam o aumento da maturidade do produto em desenvolvimento e a antecipação de detecção de defeitos ao longo do desenvolvimento. No primeiro micro “V” pode-se validar e verificar em ambiente virtual os requisitos de mais alto nível do modelo. Com a evolução do desenvolvimento, os modelos e a arquitetura do produto vão adquirindo um nível de fidelidade maior, possibilitando outras etapas de validação, como testes de integração e testes de sistemas. Já nas etapas finais de desenvolvimento, técnicas como HIL “*Hardware-in-the-loop*” possibilitam a realização de validação e verificação em ambientes simulados com cenários bem próximos ao ambiente real em que o produto final irá ser submetido aos testes.

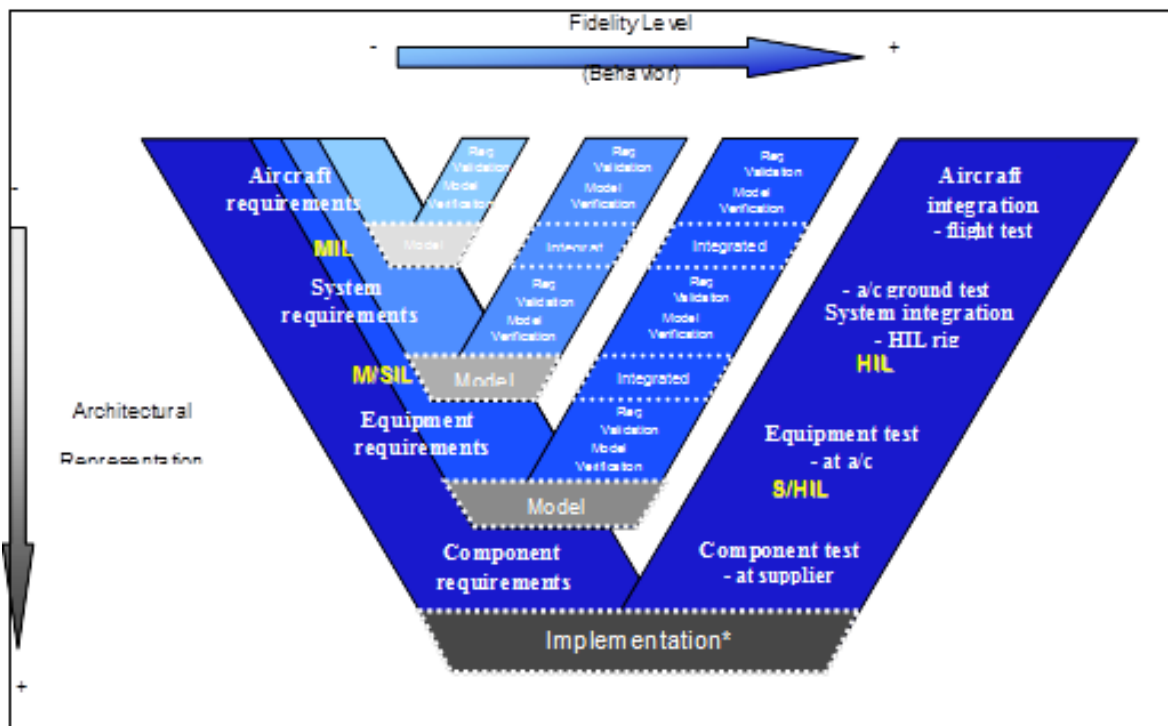


Figura 7 - Processo tradicional de Desenvolvimento em “V” modificado para múltiplos “V”

4.4 A COMPUTAÇÃO PARALELA NO AUXÍLIO DO PROCESSO DE MBD E MBT.

Com o crescimento do número de testes, os recursos computacionais de hardware e software passaram a ser gargalos. Assim a computação paralela pode auxiliar na solução deste problema, por meio do poder computacional oferecido pela tecnologia.

Atualmente as grandes empresas possuem um parque de desktops de engenharia que, em geral, ficam inoperantes, durante grande parte do dia. Esses recursos de hardware podem ser utilizados para realizar um grande número de testes.

Este potencial de processamento na forma de “*grid-computing*” pode ser utilizado em baixo acoplamento, ou seja, onde as tarefas de execução dos casos de teste são divididas entre os membros do *grid*. Observam-se na Figura 8, os casos de testes e o modelo de simulação distribuído entre os nós da rede; ou seja: nos desktops de engenharia. Observa-se ainda que o modelo possa ser sempre o mesmo em todos os casos.

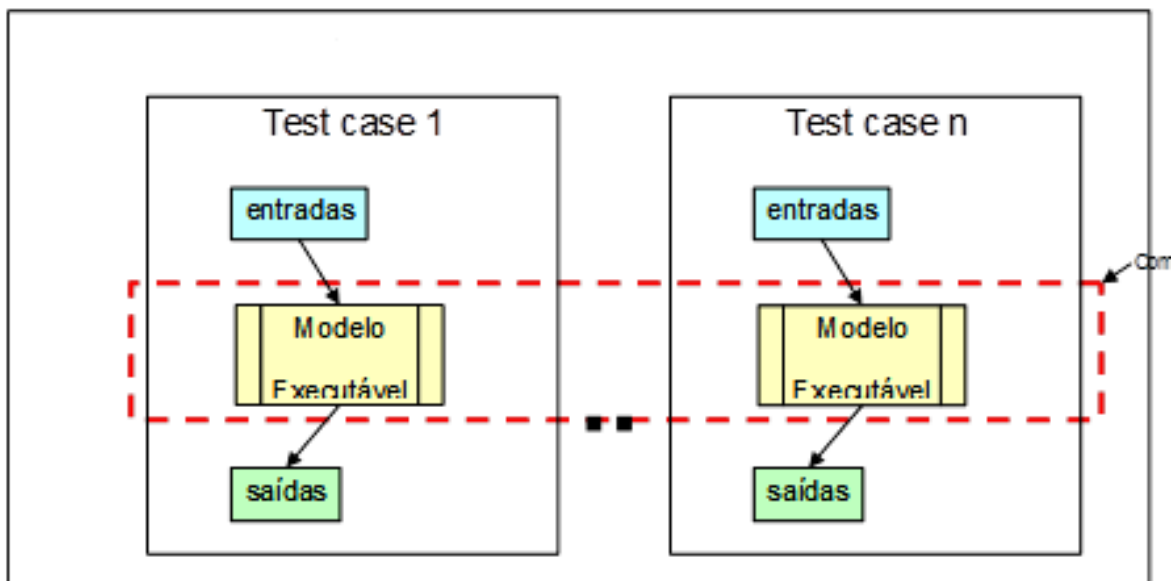


Figura 9 – Distribuição da tarefa de teste entre os nós do “grid-computing”

Uma aplicação prática para se determinar o melhor tempo de acomodação do controlador principal do sistema de ar condicionado de uma aeronave executiva de pequeno porte, tinha-se a necessidade de exercitar a varredura de cinco parâmetros do controlador com no mínimo vinte pontos por parâmetro o que resultava em cem casos de simulação; como cada caso demandava 10,2 minutos para sua execução, num total de 17h de simulação, empregou-se a técnica de *grid-computing*. Com a utilização do *grid computing* para realizar as simulações, obteve-se uma redução de 95% do tempo gasto, em um parque de 18 desktops de engenharia. Na Figura 10 observa-se a fronteira de todos os resultados encontrados nas simulações de teste; enquanto a Figura 11 ilustra a arquitetura utilizada para determinar o melhor ajuste dos ganhos do controlador. A ideia deste estudo foi demonstrar o ganho de desempenho com a utilização do *grid-computing*.

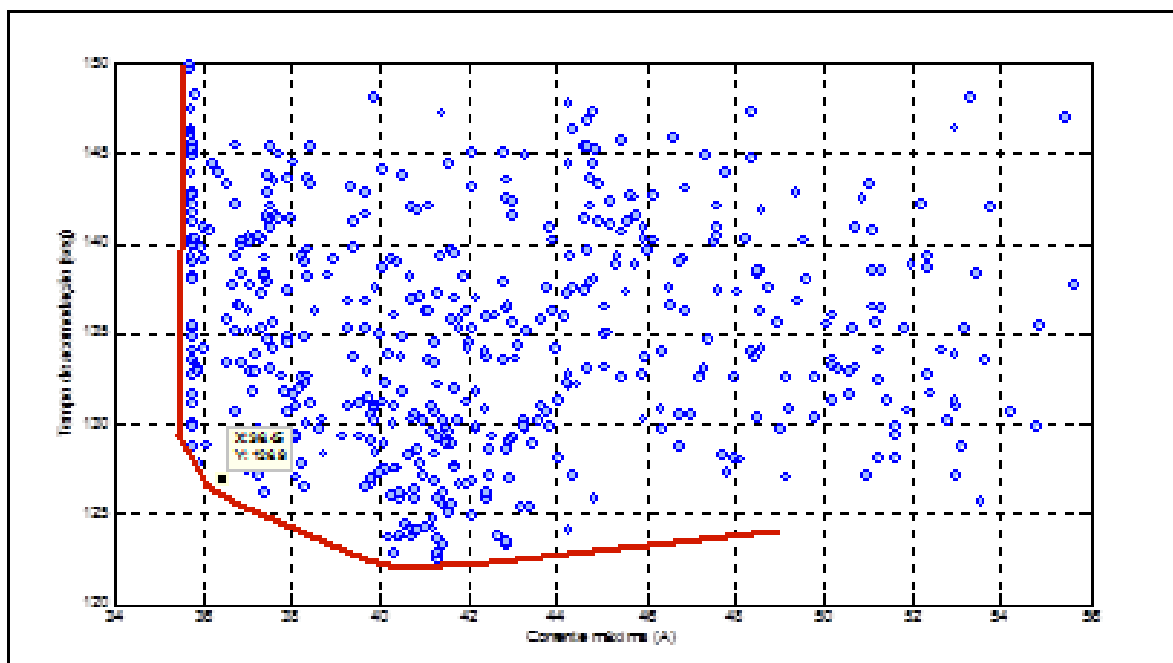


Figura 10 – Fronteira dos resultados dos Testes

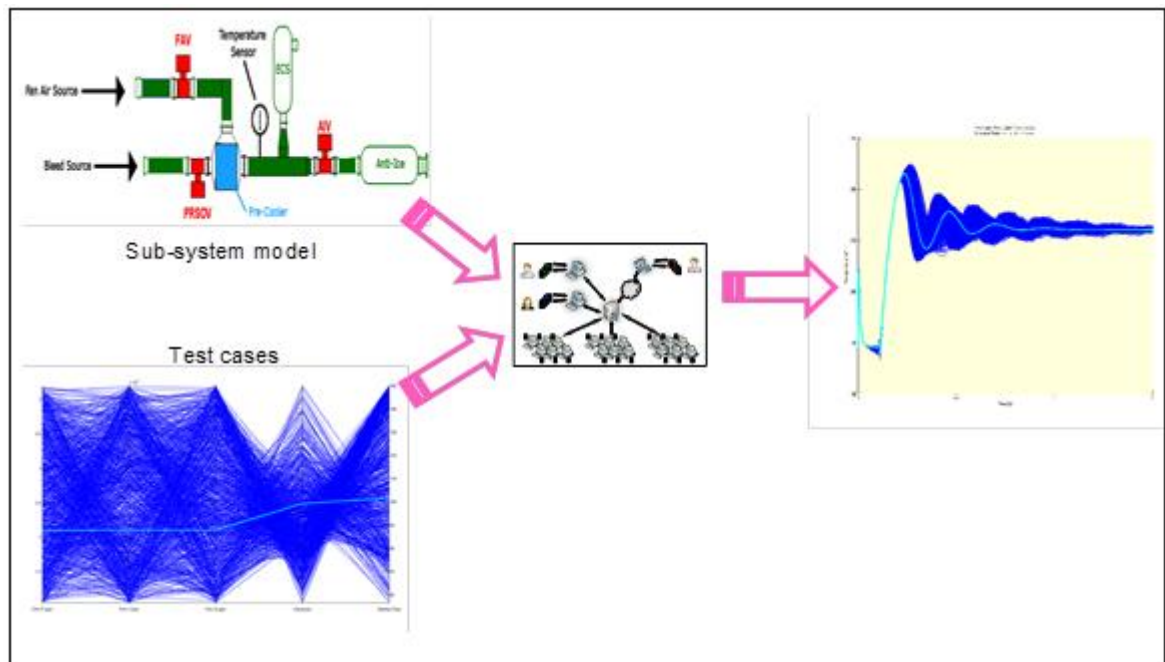


Figura 11 – Filosofia de execução de teste pelo *grid-computing*.

5. CONCLUSÕES

Nesta pesquisa pode-se verificar como as empresas que implementaram as técnicas inovadoras de “*Model-Based-Design*” e “*Model-Based-Testing*” têm obtido ganhos significativos no processo de desenvolvimento do produto em todas as suas etapas de desenvolvimento.

Foi possível verificar a importância do planejamento de testes ao longo do desenvolvimento a fim de garantir a maturidade do produto desenvolvido.

A transição do modelo tradicional em “V” para múltiplos “V”, verificada para o relato de estudo da Embraer, mostrou-se viável, pelas técnicas de “*Model-Based-Testing*”, que possibilitam o envolvimento do engenheiro de testes desde o início do desenvolvimento, e permitem antecipar a detecção de defeitos reduzindo custos. Por fim, ficaram evidenciados os ganhos da utilização da técnica de “*grid-computing*” na execução dos testes em ambiente simulado.

O Objetivo Geral e os Específicos foram totalmente atingidos

6. REFERÊNCIAS

- ALTHOLFF, Guilherme Figueira**, Using executable assertions for runtime fault detection in a Model-Based Software Development Approach. Dissertação de Mestrado do ITA , 2007.
- BAKER Loyd Jr. ,LONG James E.**, Role of System Engineering Across The System Life Cycle, INCOSE Interest Group White Paper for 2006 International Symposium.
- BURNSTEIN, Ilene**. Practical Software Testing, Springer, EUA 2003.
- DEPARTMENT OF DEFENSE**, MIL-STD-882-SD , standard practice for system safety. Fevereiro 2000.
- MYERS, Glenford J.**, The Art of Software Testing: John Wiley & Sons, 2010.
- FERRAZ, Rodrigo Barroca Dias**, Proposta de Ferramenta para desenvolvimento de simuladores, Dissertação de Mestrado USP - São Paulo, 2010.



Gestão e Tecnologia para a Competitividade

23.24.25 de Outubro de 2013

GREGA, W., “*Hardware-in-the-loop simulation and its application in control education*”. 29th ASEE/IEEE Frontiers in Education Conference, 1999, Session 12b6, pp. 7-12.

IEEE, “*Standard for Software and System Test Documentation*”, 829-2008, EUA 2008.

PRESSMAN, Roger, *Software Engineering: A Practitioner’s Approach*, 5º edition, EUA 2007.

SILVA, Edimar Thomaz, “*Quantitative Methods for management decisions in flight test campaigns for aeronautical certification*”. Dissertação de Mestrado do ITA, 2003.

THAIN Douglas, TANNENBAUM Todd e LIVNY Miron, “*Distributed Computing in Practice: The Condor Experience*” *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, pages 323-356, EUA , Abril 2005.

UNIVERSIDADE DE MINESOTA, UMN Critical System Group. EUA, 2010. Disponível em <<http://www.cs.umn.edu/crisys>> Acessado em 15 maio de 2013.

UTTING, M; LEGEARD B., *Practical Model-Based Testing: a Tools Approach*. Morgan Kaufmann, 2007.

VIJAYKUMAR, N.L., Testes de Software Baseados em Modelos- 10 a 13 de fevereiro de 2009. Notas de Aula LAC/INPE.