QUALIDADE DE SERVIÇO DE SISTEMAS COMPUTACIONAIS: AVALIAÇÃO DE SEGURANÇA DE FUNCIONAMENTO QUANTO A FALHAS DE CONCEPÇÃO HARDWARE E SOFTWARE

TESE DE DOUTORADO SUBMETIDA À
FACULDADE DE ENGENHARIA ELÉTRICA
DA UNIVERSIDADE ESTADUAL DE CAMPINAS

1º de Julho de 1991

Este exemplar corresponde à redação final da tese defendida por Marta Rettelbusch de Bastos Martini e aprovada pela Comissão Julgadora em 01/07/1991.

Orientador

Marta Rettelbusch de Bastos Martini

Orientador: Jorge Moreira de Souza
Co-orientador: Hermano M.F. Tavares

Banca Examinadora

Membros Titulares:

- Dra. Karama Kanoun, Centre National de Recherche Scientifique, Laboratoire d' Automatique et d' Analyse des Systèmes;
- Dr. Jorge Moreira de Souza, Centro de Pesquisa e Desenvolvimento da TELEBRÁS;
- Prof. Dr. Hermano Medeiros Ferreira Tavares, Faculdade de Engenharia Elétrica da UNICAMP;
- Prof. Dr.Rege Scarabucci, Faculdade de Engenharia Elétrica da UNICAMP;
- Prof. Dr. Ivanil S. Bonatti, Faculdade de Engenharia Elétrica da UNICAMP.

Membros Suplentes:

- Prof. Dr. Julius Cesar Barreto Leite, Faculdade de Engenharia Elétrica da Pontifícia Universidade do Rio de Janeiro;
- Prof. Dr. Mario Jino, Faculdade de Engenharia Elétrica da UNICAMP, e
- Prof. Dr. Edson Moschin, Faculdade de Engenharia Elétrica da UNICAMP.

A Lívia
A Júlia

"O estudo da arte da manutenção de motocicletas é realmente um estudo da arte de auto-racionalização. Reparando uma motocicleta, trabalhando bem, com cuidado, tornamo-nos parte de um processo cujo fim é alcançar uma íntima paz de espírito. A motocicleta é principalmente um fenômeno mental."

Robert Pirsig
Zen e a Arte da Manutenção de Motocicletas

Agradecimentos

O trabalho que deu origem a esta tese foi feito dentro das atribuições da Área de Garantia de Qualidade de Sistemas do Departamento de Comutação do Centro de Pesquisa e Desenvolvimento da TELEBRÁS (CPqD-TELEBRÁS).

Eu agradeço à TELEBRÁS o apoio institucional que me foi dado durante estes anos de trabalho nas pessoas dos senhores:

- Romualdo Monteiro de Barros, Chefe do Departamento de Comutação;
- Aldemar Fernandes Parola, que o antecedeu nesta função;
- Veríssimo Pires Filho, Coordenador de Áreas de Engenharia de Sistemas;
- Dr. Jorge Moreira de Souza, Chefe da Área de Garantia de Qualidade de Sistemas.

Uma parte significativa deste trabalho foi desenvolvida dentro do Acordo de Cooperação Técnica entre o CPqD-TELEBRÁS e o Laboratoire d' Automatique et d' Analyse des Systèmes (L.A.A.S) do Centre National de la Recherche Scientifique da França (CNRS). Desta forma, tenho muito a agradecer a esta instituição, nas pessoas dos senhores:

- Dr. Alan Costes, Diretor do L.A.A.S;
- Dr. Jean Claude Laprie, Diretor de Pesquisa do CNRS, responsável pelo Grupo de Tolerância a Defeitos e Segurança de Funcionamento (TSF) do L.A.A.S.;
- Dra. Karama Kanoun, pesquisadora do CNRS no L.A.A.S.,

por me haverem acolhido por duas vezes em sua equipe, e partilhado comigo sua grande experiência nesta área de trabalho. Este apoio teve importância fundamental na evolução desta tese.

Agradeço à Universidade Estadual de Campinas (UNICAMP) a confiança que depositou em mim, nesta empreitada de risco que é elaborar uma tese de doutorado fora da Universidade, na pessoa do professor Dr. Hermano Tavares, membro do departamento de Automação da Faculdade de Engenharia Elétrica.

Agradeço ao Programa das Nações Unidas para Paises em Desenvolvimento (PNUD), que possibilitou o intercâmbio de visitas da Dra. Karama Kanoun ao CPqD e minha ao L.A.A.S. para acompanhamento do Plano de Trabalho em Confiabilidade de Software, dentro do Acordo de Cooperação Técnica entre os dois laboratórios.

Os meus sinceros agradecimentos aos senhores doutores:

- Karama Kanoun, pesquisadora do CNRS no L.A.A.S.;
- Jorge Moreira de Souza, pesquisador do CPqD-TELEBRÁS;

- Hermano Medeiros Ferreira Tavares, professor da Faculdade de Engenharia Elétrica da UNICAMP;
- Rege Scarabucci, professor da Faculdade de Engenharia Elétrica da UNICAMP, e
- Ivanil S. Bonatti, professor da Faculdade de Engenharia Elétrica da UNICAMP

a honra de terem aceitado participar da banca de exame desta tese, assim como aos membros suplentes, professores doutores:

- Julius Cesar Barreto Leite, professor da Faculdade de Engenharia Elétrica da Pontifícia Universidade do Rio de Janeiro;
- Mario Jino, professor da Faculdade de Engenharia Elétrica da UNICAMP, e
- Edson Moschin, professor da Faculdade de Engenharia Elétrica da UNICAMP.

Agradeço muito particularmente ao Dr. Jorge Moreira de Souza o privilégio de poder contar com sua orientação. Sem o seu incentivo e apoio este trabalho não teria sequer começado. Tenho um grande respeito por sua maneira de promover o aperfeiçoamento do trabalho sem jamais impor suas idéias.

Ao professor Dr. Hermano M.F. Tavares, que me deu a honra da co-orientação desta tese, eu agradeço pela manifestação de confiança e respeito que isto representou para mim.

Agradeço a todos os pesquisadores do DCT que ajudaram na árdua tarefa de coleta de dados de falha e de interpretação dos resultados:

- à equipe do projeto APCC, na pessoa do seu coordenador eng. Antonio Palma Neto, e do eng. Marcio Machado Pereira,
- à equipe do projeto SAMSAT, nas pessoas do eng. Ranieri Araújo Gonçalves que me liberou os dados deste projeto e do eng. Giovanni Moura de Holanda, que trabalhou comigo na avaliação e análise dos resultados deste sistema;
- a toda a equipe dos projetos TRÓPICO R e TRÓPICO RA. O seu compromisso com a qualidade do sistema levou à implantação de um acompanhamento sistematizado de sua concepção que foi o embrião da idéia de Programa de Segurança de Funcionamento apresentada nesta tese.

Agradeço aos colegas Thierry Sabourin, da CIT ALCATEL, Mohamed Kàaniche do L.A.A.S. e Sylvain Metge que me acompanharam em diferentes etapas deste trabalho, pelas discussões proveitosas e trocas de idéias que me fizeram ver que não estava só em minhas dúvidas e convicções.

Agradeço aos colegas da equipe de Engenharia de Sistemas do CPqD-TELEBRÁS que contribuiram para que eu pudesse desenvolver o meu trabalho dentro de um ambiente amigável e me deram força nos momentos de desânimo. Um obrigada especial a Giovanni Moura de Holanda e Flávia A.M.Vieira Silva, pelo calor de sua amizade.

Nada mais valioso do que a disponibilidade de um colega para nos ajudar na hora em que mais precisamos. Meu muito obrigada a Paulo Alves Lima Neto, Laura Fazano e Eduardo dos Santos Martarello.

O meu agradecimento a Karama Kanoun, que soube dar cor ao nosso relacionamento de trabalho, transformando-o num grande incentivo e numa relação de amizade.

Dirijo um agradecimento especial ao meu amigo Jorge M. de Souza pelo respeito, compreensão e carinho que me dispensou em todos estes anos que trabalhamos juntos.

A meus pais Emmanuel e Anna sou grata pelo seu respeito às minhas opções de vida e pela confiança irrestrita que sempre me dedicaram.

Agradeço também a Cícera, que foi o meu braço direito e a amiga dedicada com quem sempre contei.

O meu agradecimento maior é para Martini, meu marido, e minhas filhas Lívia e Júlia, por terem compreendido a importância deste trabalho para mim e terem aberto mão de tantos momentos de nossa vida em família para que ele pudesse se concretizar.

Agradeço, enfim, a todos os amigos que, de perto ou de longe, me incentivaram e ajudaram nesta tarefa.

Conteúdo

Resumo

À medida que a importância dos sistemas computacionais cresce na sociedade moderna, mais rigorosos se tornam os requisitos de qualidade a que têm que atender. Atualmente a complexidade do processo de desenvolvimento do hardware e do software força a que os defeitos de concepção sejam também considerados na avaliação da segurança de funcionamento de sistemas, além da deterioração física dos seus componentes.

A modelagem e a análise de dados de falhas de concepção que o sistema apresentou em operação são técnicas de controle de qualidade usadas para avaliar a qualidade de serviço de sistemas, a fim de se obter resultados quantitativos.Estes reseultados são um instrumento valioso para a gerência do desenvolvimento ao longo de todo o processo de desenvolvimento do sistema.

Os defeitos de concepção do hardware e do software são da mesma natureza. É mostrado que os modelos e teorias desenvolvidos nas últimas duas décadas para a garantia de funcionamento de software podem ser utilizados para analisar a concepção do hardware.

Este trabalho apresenta uma metodologia de garantia de funcionamento quanto a falhas de concepção baseada na análise prévia da tendência da série de dados de falha antes da modelagem do processo de falha. O processo de falha é dividido em períodos no tempo nos quais a tendência da série de dados é relacionada à hipótese de tendência dos modelos aplicados. A metodologia proposta é de uso geral e permite analisar aspectos importantes tais como processos de falha complexos, eficiência de teste, impacto de mudanças de especificação na segurança de funcionamento de sistemas e previsão de falhas.

É proposto um Programa de Segurança de Funcionamento quanto a falhas de componentes e de concepção que relaciona as atividades de garantia de segurança de funcionamento às fases do ciclo de vida do sistema. Este Programa tem dois objetivos principais:

1. assegurar que o sistema satisfaz os requisitos de segurança de funcionamento especificados, e

2. ajudar a gerência do desenvolvimento a obter uma concepção confiável dentro do cronograma estipulado.

Diversos processos de falha de sistemas reais de telecomunicações baseados em computador são usados para exemplificar a abordagem proposta e atestar os bons resultados obtidos.

Abstract

As the importance of computer systems increases in modern society, more stringent become the quality requirements they have to meet. Nowadays the complexity of software and hardware development processes forces design faults to be taken into account in system dependability analysis, besides physical deterioration of components.

Modeling and analysis of experimental design failure data are quality control techniques used to evaluate the dependability of systems, in order to provide quantitative results which are a major aid to management during the system development process.

Software and hardware design faults have the same nature. It shown that the models and theories developed during the last two decades for software dependability assessment can be used to analyze hardware design.

This work presents a general purpose design dependability assessment methodology based on the analysis of failure data trend before modeling failure process. Failure process is divided into time periods in which the trend is related to the trend hypothesis of the models. Important aspects as complex failure processes, test eficiency, impact of specification changes in system dependability and failure forecasting are analyzed using the proposed modeling methodology.

It is proposed a Dependability Assurance Program for design and component failures which relates dependability assessment activities to the system life-cycle phases with two main objectives:

1. ensure that the system satisfies the specified requirements, and
2. help the development management to obtain a reliable design within the stipulated schedule.

Several real computer based telecommunication systems failure processes are used to exemplify the proposed approach and testify the good results obtained.

Introdução Geral

"A motocicleta é um sistema de idéias moldado em aço. Nela não há peças nem formas que não sejam fruto do pensamento humano." [PIR 84]

A sociedade moderna é altamente computadorizada. Quanto mais importante se torna o papel dos sistemas computacionais, maior é a perda sofrida quando eles apresentam uma falha. Por este motivo os sistemas computacionais devem ser muito confiáveis.

Para garantir que se pode ter confiança no funcionamento de um sistema computacional é necessário, antes de mais nada, identificar o que pode levar o sistema a falhar.

Os sistemas computacionais são tradicionalmente divididos em partes materiais (hardware) e lógicas (software). Uma falha no sistema pode ser causada pela falha de um dos componentes hardware que o compõem devido à fadiga natural do material de que é feito. Cada componente tem associada a si uma probabilidade de falha que é função de suas características físicas e de utilização e é relativamente independente do seu tempo de funcionamento. Um sistema pode também falhar devido a um defeito na lógica com que ele foi concebido e desenvolvido. O esforço de concepção está relacionado à especificação e implementação do software e dos circuitos de componentes hardware.

A falha de um componente hardware pode levar a uma indisponibilidade parcial ou mesmo total do serviço até que o componente defeituoso seja substituído. Geralmente a restauração do serviço, no caso de uma falha deste tipo, se faz através da substituição da placa de circuitos com o componente defeituoso por outra semelhante em estoque. O sistema volta à condição de funcionamento anterior à falha. A correção do defeito consiste na substituição do componente na placa removida.

Os defeitos de concepção são essencialmente o resultado de falhas humanas no processo de desenvolvimento do sistema. Estes defeitos permanecem latentes no sistema até que uma circunstância de utilização permita que eles se manifestem sob a forma de uma falha no sistema. A condição de serviço é geralmente restaurada através da reiniciação parcial ou total do sistema, uma vez que o sistema pode ainda operar em condições de

utilização diferentes daquela em que o defeito se manifestou. A correção de um defeito de concepção consiste na alteração da implementação ou mesmo do projeto do sistema, processo este bastante mais complexo que a reparação de falhas de componentes. A remoção de um defeito de concepção é, pois, um aperfeiçoamento do sistema, uma vez que a falha gerada por este defeito jamais voltará a se manifestar.

O acompanhamento da evolução da segurança de funcionamento de um sistema, à medida que os defeitos de concepção vão sendo corrigidos, é muito importante, pois esta é a visão que o usuário tem do sistema ao longo do seu tempo de vida útil. A avaliação da segurança de funcionamento de concepção pode ser feita através de métodos matemáticos e tem os seguintes objetivos gerais:

- Durante a fase de desenvolvimento: obter índices quantitativos que orientem as atividades, de forma a se obter um sistema confiável dentro dos prazos estipulados.
- Em operação comercial: assegurar que a segurança de funcionamento do sistema atende os requisitos de especificação do sistema, permitindo ao usuário confiar no serviço recebido.
- Desenvolvimentos futuros: acompanhar diversos sistemas desenvolvidos em uma mesma organização, melhorando a capacidade de desenvolvimento de sistemas futuros.

A principal vantagem de um acompanhamento de segurança de funcionamento de concepção, no entanto, está em proporcionar um conhecimento melhor do sistema que está sendo desenvolvido.

A contribuição de cunho mais geral deste trabalho está na ampliação do conceito de segurança de funcionamento de sistemas, considerando-o de maneira a englobar aspectos de falhas de componentes e falhas lógicas ou de concepção, tanto software quanto hardware. Além disso é dado um enfoque prático à avaliação e acompanhamento da segurança de funcionamento de um sistema ao longo de todo o seu ciclo de vida, de modo que os resultados assim obtidos revertam em recomendações gerenciais que auxiliem e aprimorem não só o sistema mas também o seu próprio processo de desenvolvimento. A partir deste ponto de vista discute-se a idéia de um Programa de Segurança de Funcionamento, onde é proposta uma série de atividades paralelas ao ciclo de desenvolvimento e operação comercial do sistema, com o objetivo de garantir que os requisitos de qualidade serão atingidos.

Quanto ao aspecto mais técnico, é apresentada uma metodologia de aplicação dos modelos de crescimento de confiabilidade (originalmente desenvolvidos para a avaliação da confiabilidade de software), A metodologia é baseada na aplicação destes modelos dentro das hipóteses básicas com que eles foram desenvolvidos e aplica-se a qualquer modelo e, além disso, serve também na avaliação da segurança de funcionamento quanto à concepção do hardware.

Este trabalho está organizado em cinco capítulos:

No primeiro capítulo é caracterizado o que se entende por confiabilidade de concepção de sistemas computacionais, apresentando o contexto do trabalho.

O segundo capítulo trata do acompanhamento da segurança de funcionamento dentro do aspecto evolutivo do sistema, mostrando que resultados podem ser obtidos e como eles podem ser utilizados na reorientação do processo de desenvolvimento.

O terceiro capítulo é consagrado à apresentação dos métodos de avaliação e validação da segurança de funcionamento de sistemas. São apresentados inicialmente alguns dos numerosos modelos de crescimento de confiabilidade organizados segundo suas hipóteses básicas. A seguir são apresentados os métodos de teste de tendência de séries numéricas, que servem à investigação da natureza do processo de falha estudado. São mostrados também alguns critérios de validação dos resultados da aplicação dos modelos.

No capítulo quatro é feita a proposta de um método de utilização dos modelos de crescimento de confiabilidade baseado na análise prévia da tendência dos dados de falha e, finalmente, no capítulo cinco, é mostrado como a metodologia proposta pode ser empregada na qualificação de sistemas, através de estudos de sistemas reais.

Capítulo 1

Segurança de Funcionamento de Sistemas

1.1 Introdução

A sociedade moderna depende cada vez mais dos serviços fornecidos por sistemas computacionais. Torna-se então fundamental garantir ao usuário que ele pode ter confiança na qualidade do serviço fornecido pelo sistema.

Um modelo comum de prestação de serviços computacionais em vários setores (telecomunicações, consultoria e centros de processamento de dados em geral) é aquele em que administrações prestadoras adquirem sistemas desenvolvidos e fabricados por outras empresas a fim de fornecer serviço ao usuário final. Ao enfrentar o problema de compra de um sistema computacional o prestador de serviços avalia a qualidade de diferentes sistemas e fornecedores segundo uma série de aspectos:

- *Capacidade de auxílio ao planejamento técnico*: o fornecedor de sistemas computacionais deve ser capaz de auxiliar no planejamento dos serviços, mostrando a viabilidade técnico-econômica do produto por ele comercializado.

- *Qualidade técnica do equipamento*: o fornecedor deve provar que o produto é confiável e que atende às necessidades do usuário em uma dada aplicação. Avaliando fatores como qualidade do serviço, qualidade da tarifação ou cobrança por esse serviço ao usuário, e operabilidade do equipamento, pode-se avaliar a qualidade técnica do equipamento.

- *Qualidade de Fabricação*: o fornecedor deve ter uma planta industrial adequada, com processos que envolvam metodologias de controle de qualidade de fabricação.

- *Capacidade de Fornecimento de Serviços de Manutenção*: o fornecedor deve ser

capaz de assistir o produto tanto do ponto de vista de correção de defeitos remanescentes, quanto da incorporação de novas facilidades.

- *Outros*: dependendo do tipo de sistema devem ser analisados requisitos de qualidade relacionados a sincronismo, transmissão, etc...

A capacidade técnica do equipamento é, certamente, o aspecto mais relevante em um contexto de desenvolvimento, pois está relacionada à qualidade do equipamento em si.

Do ponto de vista do usuário final o fator que mais se destaca entre os que contribuem para afirmar a capacidade técnica do equipamento é justamente aquele que avalia a qualidade do serviço recebido por ele. Na seção a seguir são apresentados alguns dos conceitos relativos a qualidade de serviço.

1.1.1 Terminologia

Ainda não existe em português uma terminologia de consenso na área de qualidade de serviço. No entanto, alguns conceitos fundamentais à compreensão deste trabalho serão definidos, ainda que de um ponto de vista particular.

Entenda-se por **serviço** o comportamento do sistema visto por seus usuários. **Especificação de serviço** é uma descrição formal bem definida do comportamento esperado do sistema por seus usuários. Quando um sistema oferece um serviço de acordo com o especificado diz-se que o serviço é **apropriado**.

A qualidade de serviço está ligada a aspectos de desempenho e de segurança de funcionamento. O desempenho de um sistema diz respeito à eficiência com que o sistema fornece o serviço apropriado supondo que não ocorra nenhuma falha.

A **segurança de funcionamento** de um sistema é a qualidade que permite a seus usuários depositar uma confiança justificada no serviço fornecido por esse sistema [LAP84a, FRA87, ABN 81] considerando a ocorrência de falhas.

Uma **falha** do sistema ocorre quando o serviço fornecido difere do serviço especificado. É percebida pelo usuário e pode ser reproduzida e avaliada. Um **erro** é um estado interno ao sistema suscetível de provocar uma falha. A causa suposta ou atribuida de um erro é um **defeito**. Neste contexto, a definição de defeito não deve ser somente relacionada a problemas de fabricação, mas também a problemas com a especificação e

implementação física e de produtos intelectuais (projeto ou concepção).

Os conceitos de falha, erro e defeito podem ser aplicados a vários níveis do sistema: o que é uma falha num certo nível pode ser visto como um defeito num nível superior. A consequência de um defeito de programação no software, por exemplo, é um erro latente (instrução ou dado errado); a ativação desse trecho de código pode gerar dados errados, caracterizando uma falha. Em um nível mais alto, a execução errônea dessa função software (defeito) pode gerar um outro estado de erro latente no equipamento. Quando ativado, por exemplo, o erro pode levar à reiniciação do processador em que o software é executado (falha), e assim por diante.

A taxa de falha é a razão entre o incremento do número de falhas que foram percebidas pelo usuário e o incremento de tempo correspondente [ABN 81].

Como um sistema raramente falha de uma mesma maneira, isso sugere a noção de **consequência** de uma falha sobre o seu ambiente. O **ambiente** é composto pelos demais sistemas com os quais o sistema analisado interage. As falhas podem ser classificadas em ordem crescente de severidade de suas consequências, ou **criticalidade**. Um caso especial é o de sistemas cujas falhas podem ser agrupadas em duas classes, segundo sua criticalidade :

- **falhas benignas** : quando as consequências são de mesma ordem de magnitude (geralmente em termos de custo) das do serviço fornecido na ausência de falhas ;
- **falhas malignas ou catastróficas** : cujas consequências são ordens de magnitude maiores do que as consequências do serviço na ausência de falhas;

As principais medidas de segurança de funcionamento são:

confiabilidade : é a probabilidade de que um sistema venha a fornecer continuamente um serviço apropriado, durante um intervalo de tempo especificado, ou, o que é equivalente, a probabilidade de que uma falha não ocorra durante este período;

disponibilidade : é a probabilidade de um sistema executar um serviço apropriado quando requisitado, levando em conta a alternância serviço próprio/ serviço impróprio;

criticalidade : medida de continuidade de oferta de um serviço seguro, que é o tempo até uma falha catastrófica;

mantenabilidade : medida de interrupção do serviço (correção do defeito e restauração do serviço).

Os conceitos relacionados a segurança de funcionamento estão relacionados entre si na Figura 1.1. A hierarquização das medidas é feita partindo da medida mais geral (que necessita, portanto, de mais informação para o seu cálculo) para a mais particular.

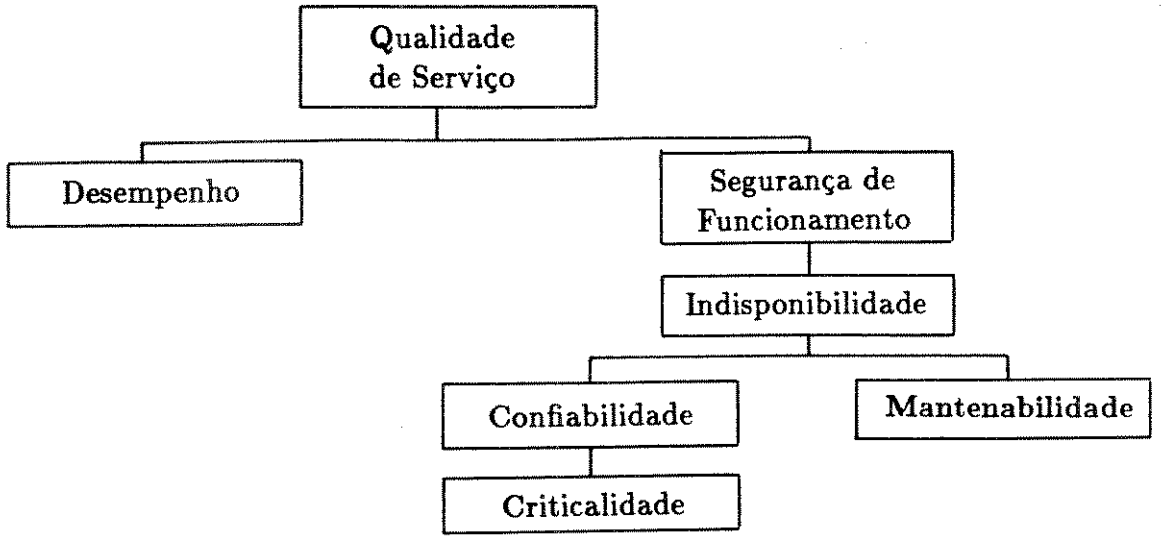

Figura 1.1 - Taxonomia da Qualidade de Serviço.

Este trabalho se concentra no aspecto de segurança de funcionamento, não abordando o problema de desempenho.

As medidas de segurança de funcionamento se tornam mais ou menos importantes de acordo com a aplicação do sistema. No caso de sistemas embarcados, para os quais não há possibilidade de reparação durante a missão (satélites, mísseis, etc...) a confiabilidade é extremamente importante. Nos sistemas de cujo serviço dependem vidas humanas (aviões, espaçonaves, etc...) a confiabilidade das funções críticas é extremamente valorizada. Outras aplicações, no entanto, admitem a ocorrência de falhas, desde que suas consequências no serviço se mantenham dentro de certos limites toleráveis durante um certo intervalo de tempo. Para estes sistemas o problema de segurança de funcionamento se baseia em detectar as falhas e remover os defeitos o mais rapidamente possível, sendo medido em termos de disponibilidade e mantenabilidade.

1.1.2 Técnicas de Segurança de Funcionamento

As técnicas de prevenção e tolerância a defeitos podem ser vistas como **técnicas de obtenção** dos requisitos de segurança de funcionamento:

- técnicas de **prevenção de defeitos**: como evitar, *por construção*, durante o projeto, a introdução de defeitos, ou seja, como construir um sistema zero-defeito e por consequência evitar a ocorrência de falhas . São as metodologias de concepção, as notações uniformizadas, "burn-in", técnicas de seleção de componentes, etc... Estas técnicas são limitadas pelo estado da arte e podem levar a custos elevados só justificáveis em algumas aplicações críticas.

- técnicas de **tolerância a defeitos**: como fornecer, *por meio de redundância*, um serviço conforme o especificado, apesar da existência de defeitos. Estas técnicas exigem hardware e software adicionais que são redundantes durante a operação normal e ativados quando há a ocorrência de falha. Desta forma a redundância garante a manutenção do serviço mesmo em presença de defeitos no sistema que possam vir a se manifestar. (técnicas de programação defensiva: programação N-versões e blocos de recuperação, redundância de itens, etc...). Estas técnicas implicam no custo das partes redundantes e sua eficácia está ligada à dos mecanismos de detecção e cobertura de falhas.

Quaisquer que sejam as técnicas de implementação empregadas durante o desenvolvimento de sistemas para se obter os índices de segurança de funcionamento desejados, é necessário utilizar **técnicas que assegurem**:

- que a implementação está correta: **técnicas de verificação**
- que o sistema presta o serviço conforme o especificado: **técnicas de validação**

As técnicas de verificação & validação compreendem revisões de projeto, revisão de código e testes, seguidos de diagnose e correção e técnicas de previsão de ocorrência e consequência de falhas.

Uma outra noção importante é a de **cobertura**, que está ligada à validação da validação, ou seja, como ter confiança nos métodos e ferramentas usados para conferir confiabilidade ao sistema. Cobertura se refere à medida de representatividade das situações às quais é submetido o sistema na fase de validação com respeito às situações reais com as quais o sistema se confrontará durante a sua vida operacional.

Dentro do contexto de segurança de funcionamento, três fatores tem contribuido para uma nova visão do conceito de qualidade:

1. A atitude cada vez mais exigente do usuário tem levado a uma abordagem do problema de garantia de qualidade voltada à satisfação de requisitos de segurança de funcionamento na forma percebida pelo usuário final.

2. A complexidade maior dos sistemas tem forçado uma revisão do conceito de qualidade, de modo a torná-lo mais geral, abrangendo aspectos que até então eram considerados irrelevantes.

3. A competitividade do mercado tem estimulado o compromisso dos produtores de sistemas computacionais com o aspecto qualidade. Isso os tem levado à adoção de programas cada vez mais severos de garantia de qualidade. Esses programas passam a atuar ao longo de todo o ciclo de desenvolvimento do produto, visando a satisfação do usuário final.

Estes fatores têm levado a uma abordagem mais ampla e mais severa da segurança de funcionamento durante o processo de desenvolvimento. Um dos aspectos que precisam ser incorporados é o de concepção, como será justificado na seção a seguir.

1.2 Abordagem da Segurança de Funcionamento de Sistemas Abrangendo Aspectos de Concepção

Nos primórdios da digitalização a qualidade de sistemas computacionais esteve estreitamente ligada à qualidade dos componentes materiais de que eles eram constituidos. Isto era bastante razoável, uma vez que o hardware representava mais de 90% do sistema. As teorias de segurança de funcionamento relativas a falhas de componentes físicos são fruto de mais de 40 anos de experiência [KLI 80].

Com o avanço da tecnologia, muitas das funções de um sistema passaram a ser desempenhadas pelo software, que vem se tornando mais e mais complexo. A situação em relação ao hardware se inverte. Hoje, em algumas aplicações, mais de 90% do esforço de desenvolvimento, teste e manutenção de sistemas computacionais estão relacionados ao software [BOE 81]. Para sistemas de grande complexidade como os sistemas de comutação digital foi comprovado que só através de uma abordagem sistêmica do problema de segurança de funcionamento relacionado a falhas software é que seria possível atingir os severos requisitos de confiabilidade impostos a estes sistemas [FAL 90]. O software passa a ser significativo. Surge, então, a necessidade do estudo de sua qualidade, a fim de, juntamente com os componentes hardware, qualificar o sistema. Há cerca de 20 anos vem sendo desenvolvida a teoria de confiabilidade do software. Sem a característica de degradação física, as falhas software tem como origem os defeitos de projeto, que são também chamados de defeitos de concepção.

Um sistema pode falhar, também, devido a falhas de concepção do hardware. Apesar do avanço tecnológico vir produzindo componentes cada vez mais confiáveis, ainda existe a questão da qualidade da concepção de módulos a partir de componentes hardware. Com o crescimento da complexidade dos projetos este é mais um aspecto cuja influência deve ser avaliada dentro do contexto de qualificação de sistemas. Defeitos de projeto, defeitos de implementação ou introduzidos durante a correção de outros defeitos devem ser classificados como defeitos de concepção.

A teoria clássica da confiabilidade de componentes é um conceito que pode ser compreendido através da teoria de probabilidade. Um dado componente tem uma certa distribuição de probabilidade de falhas ao longo de sua vida útil. Em relação a falhas de concepção a situação é diferente. Um sistema tem latentes centenas de defeitos de concepção, porém este é um número determinístico. O elemento estocástico é introduzido pela incerteza de que estes defeitos dêem origem a falhas de operação. Dependendo das condições de operação, alguns defeitos nunca se manifestarão. Um estudo bastante completo sobre a natureza do problema de segurança de funcionamento de sistemas quanto a falhas de concepção software pode ser encontrado em [BOU 82].

Os defeitos de concepção se diferenciam dos defeitos de componentes hardware pelas seguintes características:

1. Módulos software e o projeto de módulos hardware não degradam com o uso nem estão sujeitos a fadiga, como acontece com os componentes hardware.

2. Defeitos de concepção software ou de projeto hardware podem ser introduzidos em qualquer fase do desenvolvimento, inclusive durante o processo de correção de defeitos. No caso de sistemas computacionais o potencial de falhas devido a componentes hardware pode ser considerado constante durante a maior parte de sua vida operacional.

3. Defeitos de concepção software e de projeto hardware, uma vez corrigidos, não voltam a se manifestar. É de se esperar que a confiabilidade cresca ao longo do tempo (evolução). Após a substituição do componente hardware com defeito a confiabilidade volta ao mesmo índice anterior à falha (manutenção).

4. Os defeitos de concepção hardware/software permanecem latentes no sistema até que uma certa combinação dos dados de entrada ou tipo de solicitação faça com que eles provoquem uma falha.

5. As causas de falhas associadas a componentes hardware funcionalmente idênticos mas fisicamente individuais são frequentemente independentes. Para o software e projeto hardware, as cópias de módulos são idênticas não só com respeito à sua função, mas também em relação aos defeitos causadores de falha. Embora a manifestação de uma falha em uma das cópias seja independente das outras, nos casos em que a solicitação é aleatória, a correção do defeito que a originou deve ser feita em todas as cópias.

6. A fabricação de componentes físicos pode afetar sua qualidade, enquanto que o processo de reprodução do software e do projeto hardware pode ser feito dentro de altos padrões de qualidade.

Os defeitos de concepção do hardware apresentam as mesmas características dos defeitos do software e portanto, em tese, poderiam ser estudados pelos mesmos métodos [MUS 87]. Este aspecto tem sido reforçado pela analogia que se faz atualmente entre a engenharia de software e o projeto de "chips". A complexidade de concepção de circuitos VLSI e HVLSI e a facilidade de alteração do projeto auxiliado por ferramentas CAD mostram a semelhança deste processo com o desenvolvimento de software. Até hoje, no entanto, não havia notícia de um estudo de caso real de avaliação de confiabilidade de concepção de hardware utilizando as teorias desenvolvidas para o software chegando a resultados práticos que confirmassem essa tese.

1.2.1 Evolução da Taxa de Falha

A segurança de funcionamento está ligada à caracterização do processo de falha do sistema. Este processo pode ser caracterizado através da evolução de sua taxa de falha. A seguir é analisada a forma de evolução da taxa de falha de componentes hardware e da taxa de falha de concepção de hardware e de software.

Falha de Componentes Hardware

É a falha no desempenho de uma função hardware devido a falha de um de seus componentes, supondo que todos eles operam dentro das condições especificadas.

As falhas de componente são confinadas ao módulo em falha e sanadas através de uma operação que visa restabelecer a condição anterior à falha (manutenção). Esta atividade geralmente consiste em substituir o componente em falha, através de intervenção humana.

A evolução da taxa de falha para uma dada população de componentes hardware ao longo de seu ciclo de vida [RDH 76], conhecida como curva da banheira, é mostrada na Figura 1.2. Supõe-se que os componentes sejam independentes entre si.

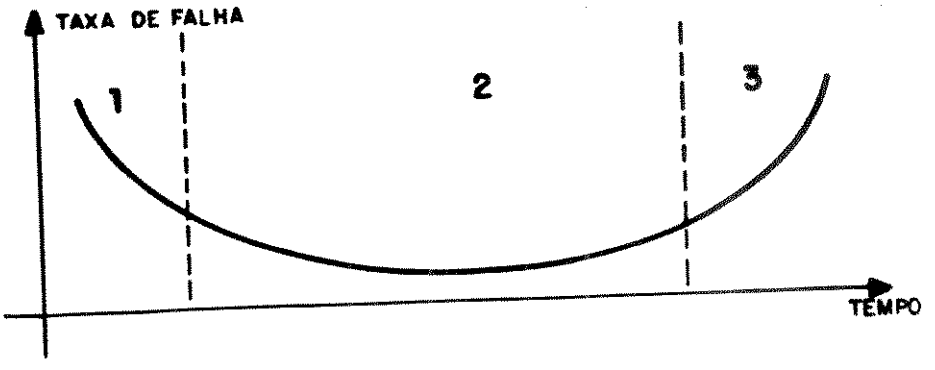

Figura 1.2 - Evolução da taxa de falha de componentes hardware ao longo do tempo para uma dada população

Pode-se distinguir três períodos:

1. Período de Mortalidade Infantil
   A taxa de falha da população de componentes é grande no início de sua vida operacional e decresce rapidamente até estabilizar em um valor aproximadamente constante. Este comportamento inicial pode ser principalmente causado por: defeitos de projeto , implementação, fabricação ou instalação do componente. A maioria das falhas nesse período pode ser prevenida através do controle de qualidade do processo de fabricação.

2. Período de Vida Útil
   Após o período de mortalidade infantil a taxa de falha atinge o seu nível mais baixo, permanecendo aproximadamente constante por um período de tempo relativamente longo, chamado de **vida útil**. A duração deste período é função do tipo de componente e do tipo de tecnologia hardware e é o período mais significativo no estudo de qualidade. As falhas que ocorrem nesse período são principalmente devidas ao esforço normal de utilização dos componentes.

3. Período de Degradação
   O terceiro período se inicia quando a taxa de falha global do equipamento começa a crescer. Esse ponto identifica o fim da vida útil do componente. A partir dali a taxa de falha cresce rapidamente, tornando-se inaceitavelmente alta.

As falhas nesse período são principalmente causadas pelo colapso dos componentes causado por fenômenos fisico-químicos como fadiga do material, oxidação e corrosão, que o inutilizam.

Seja um módulo hardware cujos componentes estejam em seu período de vida útil. Por hipótese os componentes são independentes entre si quanto a falha, com taxa de falha aproximadamente constante. Suponha-se que a cada falha de componente este seja substituido por um outro em boas condições de funcionamento. A curva da taxa de falha de uma população de módulos do mesmo tipo é dada pela Figura 1.3.

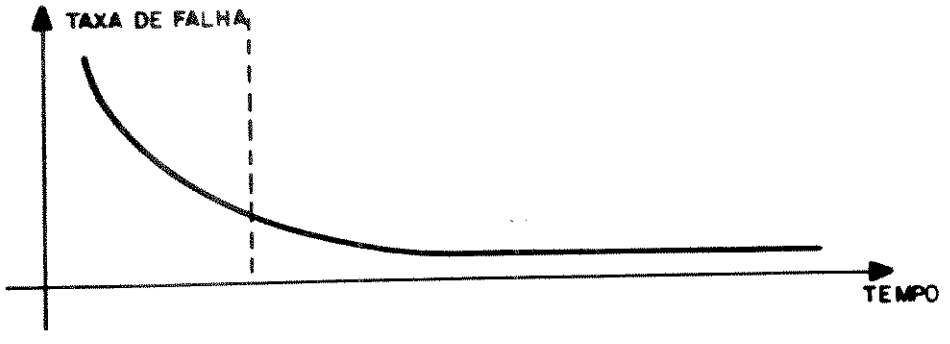

Figura 1.3 - Evolução da Taxa de Falha de um Módulo hardware ou software.

Falhas de Concepção

Falha na execução de uma função devido a defeito de especificação ou implementação de um módulo hardware ou software. Um defeito de concepção é comum a todas as cópias do mesmo módulo e é removido através da revisão do seu projeto, dando origem a uma nova edição de um ou mais módulos.

O período inicial de falha é decrescente devido a defeitos de concepção do módulo que levam a falhas de seus componentes. À medida que os defeitos de concepção dos circuitos do módulo são eliminados a taxa de falha decresce. Se houver degradação da manutenção ou se a evolução do sistema exigir alterações seguidas de suas especificações, a taxa de falha passa a crescer, definindo um período de degradação que marca o fim da vida útil do sistema. Considerando-se que a velocidade de evolução atual dos equipamentos digitais leva geralmente a uma substituição tecnológica antes do período de degradação, a curva da taxa de falha de concepção tem a forma apresentada na Figura 1.3.

A evolução da taxa de falha de concepção de um módulo software se dá de maneira análoga à da concepção do hardware, uma vez que os dois processos de falha têm as mesmas características. No entanto, para um sistema composto de hardware e software as taxas de falha não evoluem com a mesma velocidade, devido às diferentes técnicas de depuração aplicadas a estes dois sub-sistemas.

Pode-se compreender um sistema como um conjunto de módulos software ou hardware operando de forma independente. A evolução da taxa de falha de concepção de um sistema, se dá na forma de uma composição de tantas curvas do tipo mostrado na Figura 1.3 quantos forem os seus módulos, de modo que a curva resultante tem também o mesmo formato. Pode-se avaliar a confiabilidade de concepção de um módulo, de um conjunto de módulos ou do sistema como um todo, dependendo do objetivo a que serve a avaliação.

Um defeito de concepção pode ser impedido de se manifestar pela existência de um outro defeito. Só depois deste último ocasionar uma falha, ser detectado e removido, é que pode acontecer uma condição de solicitação do sistema que leve à falha devido ao primeiro. Esta característica é denominada de dependência entre defeitos de concepção.

Admitindo a dependência entre defeitos, a evolução da taxa de falha do sistema não representa mais uma combinação linear dos processos de falha de seus módulos. A evolução da taxa de falha de concepcão de um módulo e sua influência na taxa de falha do sistema, admitindo-se a existência de dependência entre defeitos nos diversos módulos é analisada em [MET 90].

Pode-se fazer duas hipóteses quanto ao comportamento da taxa de falha após um longo período de operação do sistema (tempo de operação tendendo a infinito):

- Taxa de falha tendendo a zero: com a depuração contínua do sistema a taxa de falha decresce até se tornar nula no infinito, caracterizando um sistema zero defeito.

- Taxa de falha tendendo a uma constante diferente de zero: a dificuldade de caracterização de alguns defeitos que se manifestam muito raramente através de falhas benignas podem levar à opção de deixá-los permanecer no sistema. Isso resulta em uma taxa de falha residual constante cujo valor é convenientemente baixo.

Não existem ainda evidências práticas sobre a superioridade de uma das hipóteses. Uma delas é estabelecida a priori na avaliação da confiabilidade de concepção de sistemas.

1.2.2 Obtenção da Taxa de Falha

O problema de segurança de funcionamento, quer do ponto de vista de falhas de componentes hardware, quer de falhas de concepção, está ligado à evolução da taxa de falha do sistema. A evolução da taxa de falha é caracterizada através de sua avaliação periódica ao longo do processo de desenvolvimento de sistema.

Avaliação da taxa de falha de componentes Hardware

O interesse principal em relação à avaliação da taxa de falha de componentes hardware está ligado ao seu período de vida útil, no qual a taxa de falha é considerada constante. A partir das características físicas e tecnológicas do componente e de sua utilização é calculada a taxa de falha. Os métodos de avaliação da taxa de falha nesse período estão padronizados em normas gerais como a MIL-HDBK [MIL 81] ou especializadas para um dado campo de aplicação, como a norma do CNET [CNE 83] e outras.

A taxa de falha de um módulo pode ser obtida a partir das taxas de falha de seus componentes, geralmente adotando-se uma política de pior caso. A taxa de falha do módulo é expressa como uma combinação série-paralelo das taxas de falha dos componentes, levando em conta as redundâncias existentes [SCH 68]. A taxa de falha de funções críticas ou do sistema como um todo pode ser obtida a partir das taxas de falha dos módulos envolvidos na função.

A taxa de falha estimada a partir das normas existentes é em geral referendada posteriormente, durante a operação do sistema. É feito o registro das falhas detectadas em campo e a taxa de falha, obtida através de métodos estatísticos, é comparada à estimada.

Avaliação da taxa de Falha de Concepção

Ainda não existem normas para a concepção de sistemas que associem uma taxa de falha às suas características intrínsecas. Desta forma, só é possível obter a taxa de falha de concepção a partir dos dados de falhas detectadas durante sua vida operacional. A estes dados são aplicados métodos matemáticos, aqui referenciados como modelos de confiabilidade de concepção, para se obter a taxa de falha. No caso de falha de concepção, a taxa de falha não é constante como a taxa de falha de componentes, mas decrescente no tempo.

1.2.3 Requisitos de Segurança de Funcionamento

A segurança de funcionamento é especificada utilizando-se as medidas de disponibilidade, confiabilidade, criticalidade e mantenabilidade. É necessário, portanto, analisar o impacto dos defeitos de concepção nas medidas de segurança de funcionamento a fim de caracterizar a importância da consideração de falhas de concepção na avaliação global do sistema.

Indisponibilidade

Os defeitos de concepção são responsáveis por uma parcela importante do total de falhas causadoras de indisponibilidade e não podem ser relevados na avaliação deste requisito do sistema. É o que pode ser visto na a Tabela 1.1 [TOY 85], onde são apresentados resultados relacionados a sistemas de comutação telefônica digital e sistemas de processamento de transações. As falhas software e possivelmente parte das falhas de recuperação e de procedimentos incorretos podem ser contabilizadas como falhas de concepção.

Fontes de Indisponibilidade	Comutação Eletrônica	Processamento de Transações
Falhas Hardware	20%	40%
Ambiente		5%
Falhas Software	15%	30%
Deficiências de Recuperação	35%	
Procedimentos Incorretos	30%	20%
Outros		5%

Tabela 1.1 - Fontes de falhas que causaram a indisponibilidade de alguns sistemas.

Outros resultados importantes são apresentados na Tabela 1.2, também referentes a um sistema de comutação digital [SAB 87]. As falhas software foram classificadas de acordo com o grau de severidade de suas consequências no serviço fornecido pelo sistema. A porcentagem de falhas que causa a indisponibilidade total do equipamento, exigindo a reinicialização do sistema, é considerável.

Um resultado também interessante foi obtido pelo acompanhamento da indisponibilidade de sistemas de comutação digital em campo no início da digitalização da rede telefônica sueca [FAL 90]. A indisponibilidade média por central digital instalada aumentou durante os quatro primeiros anos até um nível de 80 minutos por ano em 1983. Este desempenho estava longe do requisito fixado, que era de três minutos por ano. A

situação tornou-se crítica e levou a uma perda de confiança do usuário na qualidade do serviço fornecido pela empresa Televerket. Uma análise cuidadosa das causas desta situação apontou as falhas relacionadas ao software como uma parcela importante e levou a empresa a adotar um Programa de Qualidade de Software a partir de 1983.

Indisponibilidade Total	13%
Indisponibilidade de Circuitos	15%
Redução da Capacidade de Tratamento de Chamadas	6%
Perda de Chamada	3%
Não afetam o Serviço	63%

Tabela 1.2 - Consequências de falhas software de um equipamento de comutação digital.

Por estes exemplos percebe-se que o aspecto de concepção não pode ser desconsiderado na avaliação do requisito de disponibilidade de sistemas.

Confiabilidade e Criticalidade:

Acredita-se, até hoje de uma forma empírica, que a taxa de falha de concepção é desprezível em relação à taxa de falha de componentes hardware e que, portanto, não precisa ser considerada na avaliação desse requisito. A relação entre a frequência de falhas de concepção de sistemas reais, no entanto, tem mostrado que é necessário fazer a avaliação em relação aos três processos de falha. Um exemplo é mostrado na Tabela 1.3, que dá o número de falhas por instalação por ano para sistemas de comutação digital [ALI 86]. A duração média do tempo em que o sistema ficou fora de serviço devido a cada tipo de falha é medida em minutos por instalação por ano.

Fonte da Falha	Tempo Fora de Serviço	Taxa de Falha
Hardware	0.91	0.028
Software	0.94	0.032
Procedimento	1.65	0.37

Tabela 1.3 - Relação entre as causas de falha de sistemas de comutação digital.

Para a avaliação de requisitos de criticalidade são válidas as mesmas considerações feitas para a confiabilidade, sendo que universo de falhas de componentes e de concepção se restringe às falhas críticas ou catastróficas.

Mantenabilidade:

A restauração do serviço interrompido devido a falha de componentes hardware consiste na substituição da placa que contém o componente em falha por outra em boas condições de funcionamento. Geralmente cada instalação tem associado um estoque de placas sobressalentes, dimensionado de acordo com a taxa de falha da placa e a quantidade de cópias desta placa na instalação. A manutenção é feita substituindo-se o componente por outro em boas condições de funcionamento.

Após uma falha software ou de concepção hardware o serviço em geral pode ser restaurado através da reiniciação total ou parcial do software ou recolocação da placa hardware em funcionamento, uma vez que o sistema ainda é capaz de operar em condições diversas daquela em que ocorreu a falha. A manutenção após uma falha de software demanda um trabalho intelectual especializado e considerável intervalo de tempo para a identificação, reprojeto, implementação, e recarga da nova edição do software no sistema em operação. É um trabalho caro e demorado. O mesmo acontece com a manutenção de falhas de concepção do hardware, com o agravante de que a correção de algumas falhas de concepção pode exigir alterações no processo de fabricação da placa.

Ao contrário dos outros atributos de segurança de funcionamento, a mantenabilidade não é medida diretamente. Dois requisitos são geralmente usados para caracterizar a mantenabilidade:

- tempo médio de reparo de defeitos, e
- número máximo de intervenções no sistema.

O tempo médio de reparo de defeitos influencia a indisponibilidade total ou parcial de funções do sistema. Mesmo considerando que a taxa de falha de concepção é muito menor que a taxa de falha de componentes, o maior tempo de reparo necessário aos defeitos de concepção pode comprometer os requisitos de indisponibilidade.

Para avaliar o número máximo admitido de intervenções no sistema por unidade de tempo, considere-se que uma intervenção no sistema pode ser causada por:

- falha de componente: substituição da placa com o componente defeituoso,
- falha de projeto hardware: substituição de uma ou mais placas por novas edições nas quais o defeito que gerou a falha tenha sido removido, ou

- falha software: recarga de novas edições de um ou mais módulos nos quais o defeito tenha sido corrigido.

Qualquer que seja a política de manutenção adotada, fica evidente a importância da consideração das falhas de concepção na verificação do requisito de número máximo de intervenções.

1.2.4 Avaliação da Segurança de Funcionamento Quanto a Falhas de Componentes e de Concepção

Os requisitos de segurança de funcionamento são estabelecidos a partir do ponto de vista do usuário, que determina que medidas são mais apropriadas para qualificar o sistema. Estes requisitos são globais, não importando os processos de falha que para eles contribuem. Embora se reconheça que o serviço recebido pelo usuário pode ser afetado por uma série de fatores (falha humana, falha de componente hardware, falha de concepção, falha de manutenção, problemas com o ambiente de operação, etc...) os requisitos de segurança de funcionamento têm sido geralmente avaliados somente do ponto de vista de falhas de componentes. As exceções a esta abordagem são poucas [ROH 72, COS 78, AVE 80, PIG 88]. Os motivos são históricos e foram apresentados no início desta seção.

A partir de um comprometimento mais sério com a qualidade há a necessidade de se considerar falhas de componentes e de concepção do software na avaliação da disponibilidade e de outras medidas de segurança de funcionamento de sistemas. A contribuição da taxa de falha de concepção se torna mais importante quanto mais severos são os requisitos para uma dada aplicação do sistema. Foi mostrado que não há impedimentos teóricos a esta abordagem e o desafio a investimentos no que é chamado de segurança de funcionamento do X-ware está lançado [LAP 87, LAP 89].

A busca de uma abordagem sistêmica levou nos últimos anos à publicação de alguns trabalhos que associavam mais de um processo de falha [LAP 84a, b, FIC 85, BAS 89b] na obtenção de medidas de segurança de funcionamento. Em [LAP84b, LAP86a, b, LAP 87, LAP 90] é feito o estudo da confiabilidade e da disponibilidade abrangendo os processos de falha de componentes hardware, falhas de concepção software e falhas de controle software. É feita uma análise partindo de um modelo de estados em que estas três fontes de falha poderiam tirar o sistema de serviço.

Stark [STA 87] modelou o problema de dependabilidade (confiabilidade e disponibilidade) de um sistema composto de hardware e software operando em tempo real como

um processo semi-Markoviano.

Partindo da necessidade de uma abordagem mais sistêmica do problema de segurança de funcionamento, este trabalho também contribui para a qualificação integrada de sistemas, na medida que associa processos de falha de componentes e falhas de concepção hardware e software na avaliação de requisitos de segurança de funcionamento.

Como foi dito ao final da seção 1.1, a competitividade do mercado tem levado a programas de qualidade mais abrangentes e mais severos. Na seção a seguir a segurança de funcionamento será vista como um atributo dentro de um programa de qualidade e o aspecto de concepção será analisado dentro deste contexto.

1.3 Programa de Segurança de Funcionamento de Sistemas

A antiga forma de comprometimento com a qualidade de um produto era proceder a um rigoroso teste de aceitação ao final de seu desenvolvimento, atribuindo um grau de qualidade relativo ao desempenho do sistema nesses testes. Hoje em dia está claro que um sistema deve ser projetado e implementado com qualidade, se o objetivo é obter produtos altamente confiáveis [CAV 85]. Desta forma, as metas de qualidade e eficiência devem permear todas as atividades de desenvolvimento.

A preocupação com a qualidade levou na década de 80 à implantação dos primeiros Programas de Garantia de Qualidade (PGQ), como o Software Quality Assurance Engineering da AT&T [PEN 90] e o já citado Software Quality Program da Televerket sueca [FAL 90]. Um PGQ consiste na avaliação periódica de um conjunto de atributos do sistema a fim de orientar o processo de desenvolvimento para o padrão de qualidade especificado. É importante dizer que um programa de qualidade está relacionado não só à qualidade do produto em si, mas também à do seu processo de desenvolvimento.

Pode-se considerar um programa de garantia de qualidade como sendo uma composição de programas de garantia de cada um dos atributos prioritários da qualidade de um produto. Dessa forma, um programa de qualidade deve compreender aspectos de flexibilidade, reusabilidade, correção, qualidade de serviço (abrangendo aspectos de desempenho e segurança de funcionamento), etc...

Dentro deste contexto, um Programa de Segurança de Funcionamento (PSF) faz parte de um Programa de Garantia de Qualidade. Um PSF tem o objetivo de registrar e avaliar o processo de falha, permitindo então que ele seja controlado, de modo a não desviar o serviço fornecido pelo sistema do padrão especificado.

Este trabalho se concentra no aspecto de segurança de funcionamento, descrevendo as atividades de um programa de garantia deste atributo do sistema ao longo do seu ciclo de vida.

1.3.1 Ciclo de Vida do Sistema

Um sistema computacional passa por diversas fases desde a sua concepção até estar apto a operar comercialmente. À sequência destas fases chama-se **ciclo de vida**. Durante as fases intermediárias do ciclo de vida o hardware e o software podem ser desenvolvidos separada e, até certo ponto, independentemente [BOR 84, BOE 76]. Um esquema básico do ciclo de vida de um sistema pode ser visto na Figura 1.4.

Um módulo hardware ou software é definido como um conjunto de funções que podem ser implementadas e testadas separadamendente e atendendo às restrições de tamanho (de placa de circuitos, de memória de processador, etc...) e de complexidade características do projeto. As fases do ciclo de vida são detalhadas a seguir:

Especificações e Requisitos do Sistema: os objetivos e requisitos do sistema, visto por seus usuários, são formalmente definidos em documentos, de maneira concisa e clara. Esses documentos devem identificar as funções lógicas a serem realizadas pelo sistema, o conjunto de serviços que o sistema deve ser capaz de fornecer, as condições de operação, o desempenho esperado do sistema e os requisitos de qualidade que lhe são impostos.

Revisão de Especificações e Requisitos: o objetivo desta revisão é assegurar a clareza da documentação, garantindo não haver ambiguidade em relação aos requisitos estabelecidos.

Projeto de Sistema: nesta fase é detalhada a decomposição do produto a ser desenvolvido e o relacionamento entre as partes. O sistema é decomposto em hardware e software e cada um destes sub-sistemas em módulos segundo suas funções. A decomposição facilita a implementação e o teste.

Revisão do Projeto de Sistema: esta revisão pretende assegurar que a partição em módulos, assim como o interfaceamento entre estes, é consistente, factível e atende às especificações funcionais iniciais.

Projeto de Módulo Software: nesta fase cada módulo software é especificado e detalhado a nível de suas funções individuais e sinais trocados com outros módulos.

Revisão do Projeto dos Módulos Software: esta revisão tem por objetivo verificar o atendimento dos padrões e a fidelidade do projeto em relação às especificações e requisitos concernentes a cada módulo.

Codificação do Software: os módulos são implementados usando uma linguagem específica.

Revisão de Código: as formas programáticas dos módulos software são comparadas aos documentos de projeto. O objetivo é garantir que o código reflita o projeto, verificando se os padrões de codificação são atendidos, descobrindo e corrigindo defeitos de codificação e de concepção.

Teste de Módulo Software: o objetivo é testar cada módulo isoladamente de tal forma que uma certa cobertura seja garantida, como por exemplo, todas as instruções ou todas as trajetórias lógicas no módulo sejam executadas.

Projeto de Módulo Hardware: cada módulo hardware tem seu projeto detalhado sob o ponto de vista funcional, mecânico, elétrico, e físico (componentes).

Revisão de Projeto de Módulo Hardware: esta fase tem por objetivo verificar o atendimento dos padrões e a fidelidade do projeto em relação às especificações e requisitos mecânicos, funcionais, elétricos, etc..., concernentes a cada módulo.

Montagem de Módulo Hardware: implementação dos módulos em placas segundo o projeto detalhado.

Revisão de Implementação de Módulo Hardware: as placas são examinadas com o propósito de assegurar que a implementação seja fiel ao projeto.

Teste de Módulo Hardware: os módulos hardware são testados isoladamente. Fazem parte desta fase os testes de parto e de reprodução de placas.

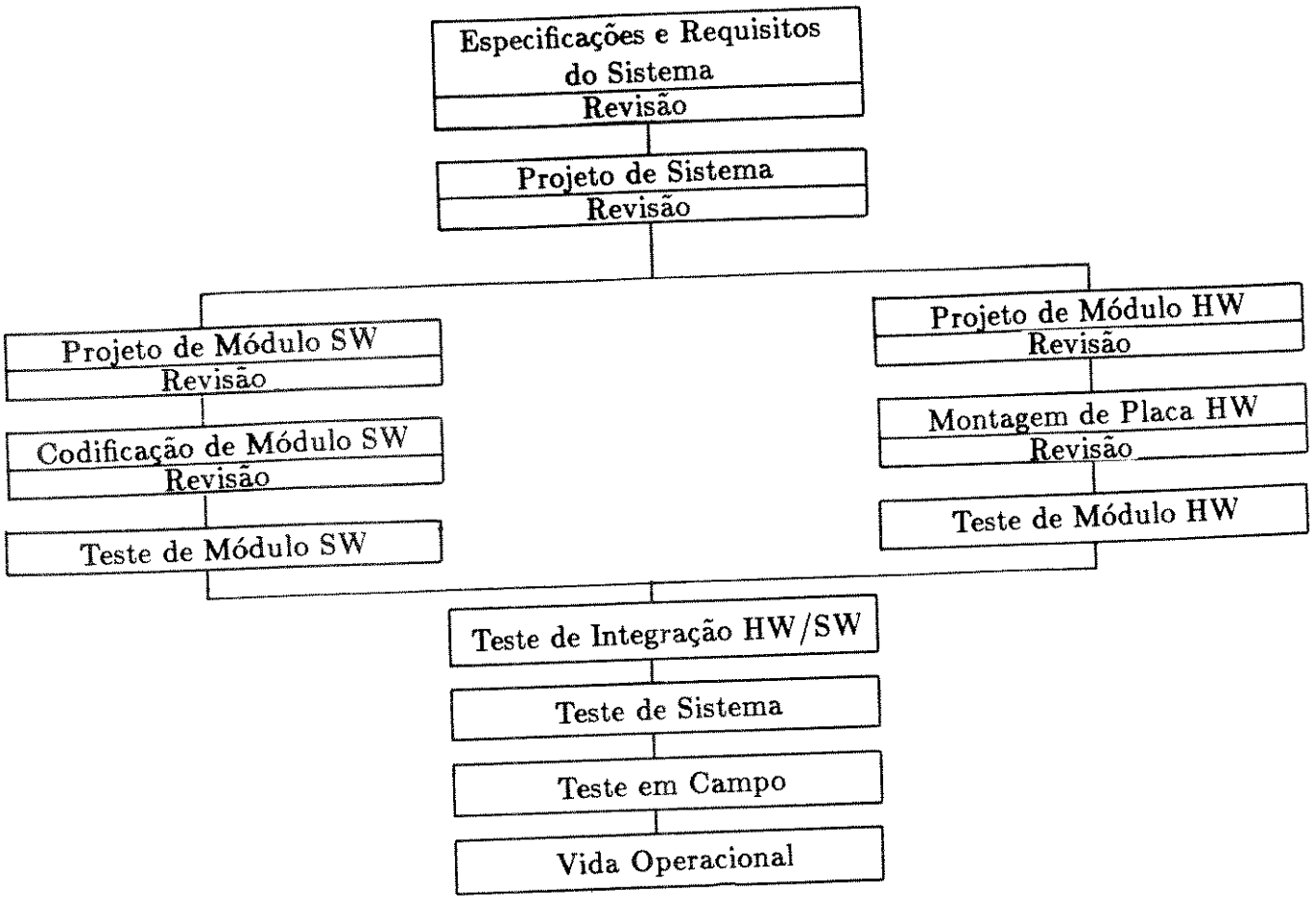

Figura 1.4 - Ciclo de vida de um sistema.

Teste de Integração Hardware-Software: o objetivo dos testes de integração é verificar se cada módulo do sistema interage corretamente com seu ambiente, ou seja, os outros módulos hardware e software com os quais ele mantém interface. Os testes são executados integrando um módulo a cada passo, partindo das camadas mais básicas, concentrando nele os testes.

Teste de Sistema: nesta fase o sistema é testado do ponto de vista funcional, visando descobrir qualquer implementação incorreta, ou mesmo falha de projeto, em relação às especificacões e requisitos.

Teste em Campo: nesta fase o sistema é testado em ambiente real de operação, sendo exigido para a execução de muitas atividades simultâneas. O objetivo nesta fase é

testar o sistema do ponto de vista do operador e do usuário, usando as mesmas operações por eles feitas.

Vida Operacional: fase em que o produto está em operação comercial, sujeito a manutenção, e inclusão de novas funcionalidades e melhorias.

A cada etapa do ciclo de vida deve corresponder uma atividade de um Programa de Segurança de Funcionamento, segundo o estágio de desenvolvimento do sistema e os objetivos a que serve este Programa.

Após cada uma das atividades de especificação, projeto e implementação é prevista uma atividade de revisão. As inspeções intermediárias do produto são parte importante dentro de um Programa de Segurança de Funcionamento. Resultados práticos relacionados a inspeção de software de sistemas de comutação digital mostram que a eficiência de uma revisão é duas a quatro vezes maior que a do teste de sistema [RUS 91]. Os defeitos de concepção identificados nas revisões diminuem a probabilidade do sistema falhar em operação.

A partir do resultado de uma revisão pode ser necessário retrabalhar uma ou mais etapas anteriores.

1.3.2 Objetivos da Aplicação de um Programa de Segurança de Funcionamento

Um Programa de Segurança de Funcionamento de um sistema aplicado ao longo das etapas do seu desenvolvimento tem como objetivos principais:

1. Assegurar que a concepção satisfaz os requisitos de confiabilidade especificados: a avaliação de confiabilidade, feita em certos pontos-chaves do desenvolvimento de um sistema, determina se o produto atingiu o índice desejável naquele estágio, habilitando-o ou não a passar para uma etapa posterior do ciclo de vida.

2. Auxiliar na gerência das atividades de desenvolvimento a fim de obter um projeto confiável dentro dos prazos estipulados: através da avaliação de confiabilidade em um certo momento é possível projetar o nível de confiabilidade que o produto atingirá no próximo ponto-chave do ciclo de vida. Comparando o resultado dessa previsão com os índices especificados podem ser tomadas medidas gerenciais a fim de corrigir a rota atual.

3. Fazer a avaliação quantitativa de tecnologias de engenharia de software e de projeto hardware: comparando os índices de confiabilidade antes e depois da aplicação de tecnologias de desenvolvimento (metodologias de projeto, de teste, de inspeções, etc...), ou de dois projetos semelhantes usando tecnologias diferentes, é possível avaliar se essas tecnologias estão sendo eficazes ,

4. Fazer sondagens de alternativas gerenciais: fazer ou não revisões e qual o custo de não fazê-las sempre foi uma grande dúvida na aplicação da engenharia de software. Antes de tomar uma decisão como esta, a sondagem de alternativas sob o ponto de vista de confiabilidade pode ser importante. Embora seja uma visão restrita do problema, é um resultado quantitativo mensurável da decisão.

Após a liberação do produto para comercialização, o Programa de Segurança de Funcionamento tem as seguintes finalidades:

1. Acompanhar a evolução do sistema em termos de confiabilidade: os índices crescentes de confiabilidade alcançados ao longo da vida operacional do produto atestam sua qualidade. A confiabilidade de concepção pode, dessa forma, ser vista como um atributo do sistema a ser vendido,

2. Auxiliar no dimensionamento do esforço de manutenção do projeto: as previsões do comportamento de falha do sistema, baseadas nos dados de falha detectadas até então, dão idéia do esforço que deve ser alocado para o suporte ao usuário em relação à depuração do produto,

3. Orientar a liberação de novas versões ou alterações no produto: sabe-se [MUS 87] que a introdução de novas funções atua como uma perturbação no sistema, chegando algumas vezes a diminuir a sua confiabilidade. É aconselhável, portanto, que as alterações se façam quando o sistema apresentar um alto índice de confiabilidade, sob pena de se perder o controle da qualidade. A avaliação de confiabilidade pode indicar o momento conveniente de alteração. O efeito dessas perturbações no sistema pode ser avaliado de modo a detectar o momento em que volta à condição anterior em termos de confiabilidade.

1.3.3 Atividades do Programa de Segurança de Funcionamento de Sistemas ao Longo do seu Ciclo de Vida

Um programa de segurança de funcionamento pode ser descrito como uma série de atividades executadas ao longo de todo o ciclo de vida de um sistema a fim de verificar e validar o sistema quanto a este atributo. Um programa desse tipo deve compreender

cinco atividades básicas:

1. Especificação da segurança de funcionamento,
2. Análise de viabilidade de soluções sistêmicas,
3. Projeção da segurança de funcionamento,
4. Verificação da segurança de funcionamento em laboratório,
5. Validação e previsão da segurança de funcionamento em campo.

Ciclo de Vida	Atividades do Programa de Segurança de Funcionamento	Resultados
Especificação do Sistema	Especificação da Segurança de Funcionamento	-Índices de segurança desejados
Projeto do Sistema	Análise de Viabilidade de Soluções	-Viabilidade das soluções sistêmicas
Projeto de Módulo HW/SW Implementação de Módulos HW/SW	Projeção da Segurança de Funcionamento	-Taxas de falha estimadas de módulo HW -Estimacão da taxa de falha HW das principais funções do sistema -Identificação de componentes críticos HW **-Identificação de Componentes críticos HW/SW** **-Taxa esperada de falha de concepção**
Teste Isolado de Módulos HW/SW Teste de Integração dos Módulos HW/SW Teste de Sistema Teste em Campo	Verificação da Segurança de Funcionamento em Laboratório	-Acompanhamento da taxa de falha de componentes críticos **-Avaliação da eficácia dos testes** **-Taxas previstas de falhas de concepção** **-Previsão do tempo até a obtenção dos índices de Segurança de Funcionamento especificados** **-Avaliação qualitativa do sistema** **-Previsão do número de falhas a serem detectadas num horizonte definido**

Ciclo de Vida	Atividades	Resultados
Vida Operacional	Validação da Segurança de Funcionamento em Campo	-Relação entre as taxas de falha de componentes projetada e observada **-Qualificação do sistema em relação aos requisitos de projeto quanto a falhas de componentes e de concepção** **-Oportunidades de introdução de melhorias**

Figura 1.5 - Programa de Segurança de Funcionamento.

Estas atividades estão relacionadas ao ciclo de vida do sistema e aos objetivos do programa como mostra a Figura 1.5, onde são listados os resultados obtidos classicamente para falhas de componentes hardware e ressaltados em negrito os propostos para falhas de concepção.

As atividades de um programa de segurança de funcionamento e seus resultados serão detalhados a seguir.

1.3.3.1 - Especificação da Segurança de Funcionamento

Tornar claros os objetivos do projeto em termos de qualidade de serviço é útil no sentido de fundamentar as expectativas de custo, cronograma e esforço necessários. Os requisitos de qualidade servirão de metas às quais devem ser comparados os resultados das avaliações de qualidade feitas ao longo do ciclo de vida.

Índices de Segurança de Funcionamento Desejados: Durante as fases de especificação e projeto de sistema devem ser definidos os índices de qualidade de serviço (indisponibilidade, confiabilidade e mantenabilidade) que o sistema deve atingir. A severidade destes requisitos depende do tipo de produto e sua aplicação.

Nas aplicações em telecomunicações geralmente são seguidos padrões de qualidade estabelecidos em órgãos normativos como o CCITT e CNET [CNE 83]. Os requisitos de qualidade de serviço são sistêmicos, constituidos de grandezas observáveis pelo usuário, sem a identificação dos processos de falha que para eles contribuem e portanto devem ser compreendidos em um contexto global.

1.3.3.2 - Análise de Viabilidade de Soluções Sistêmicas

Nessa fase do desenvolvimento são projetadas as grandes funções do sistema e algumas soluções de arquitetura são propostas.

Viabilidade das Soluções Sistêmicas: É necessário examinar a factibilidade das soluções de arquitetura frente aos requisitos de segurança de funcionamento do sistema. É possível identificar a necessidade de técnicas de tolerância a defeitos.

1.3.3.3 - Projeção da Segurança de Funcionamento

Uma vez fixadas metas de segurança de funcionamento, as diversas alternativas de projeto devem ser analisadas de modo a detectar, baseado em suas características intrínsecas, a exitência de algum fator que as impeça de alcancar os objetivos estabelecidos.

A hipótese básica é de que existe alguma relação entre a maneira pela qual o projeto é desenvolvido e sua segurança de funcionamento final. O estudo dessas relações pode levar a técnicas de desenvolvimento mais eficazes, à identificação de módulos potencialmente mais problemáticos e também avaliar se os requisitos de confiabilidade podem ser atingidos no tempo designado.

Taxas de Falha Estimadas de Módulos Hardware: A taxa de falha de componentes hardware em vida útil pode ser estimada através de normas internacionais como a MIL-HDBK-217E [MIL 81] em função das características físicas de cada componente, sua tecnologia de desenvolvimento, solicitação a que está submetido no projeto, etc...

Taxas de Falha Hardware das Principais Funções do Sistema: A partir da taxa de falha de cada componente é possível estimar a taxa de falha das principais funções do sistema devido a falha de componentes hardware e até mesmo a taxa de falha global do equipamento.

Identificação de Componentes Críticos Hardware: Com base nas taxas de falha estimadas é possível identificar os componentes críticos do sistema, ou seja, aqueles cuja taxa de falha em muito contribui para a taxa de falha global do equipamento.

Estimação da Taxa de Falha de Componentes Críticos Hardware e Software:

Em relação a falhas de concepção, não existem normas para a estimação da taxa de falha.

Uma forma comparativa de projeção da taxa de falha de concepção do hardware e do software pode ser feita através da observação das taxas de falha atingidas em campo por módulos com funções análogas de projetos já desenvolvidos no mesmo ambiente. As taxas de falha observadas em campo podem, neste caso, servir como referência para a projeção da taxa de falha para os novos módulos.

Identificação de Componentes Críticos Hardware e Software quanto ao Aspecto de Concepção: Um outro resultado que pode ser obtido nessa fase do projeto é a identificação de módulos críticos, ou seja, de módulos mais propensos a falha de concepção. Esta identificação é importante para a orientação dos testes aplicados a estes módulos, que devem ser mais severos que os aplicados aos demais. A projeção de segurança de funcionamento de um módulo ofensor pode levar a uma revisão do projeto, escolhendo uma outra alternativa mais confiável, ou à assunção conciente dos riscos que estão implícitos na alternativa escolhida, em termos de qualidade.

Uma das formas de se identificar módulos software mais propensos a falha a partir de suas características intrínsecas pode ser feita através de métricas de complexidade [ALB 82, FER 86] apropriadas à linguagem e à estrutura do produto. Através do índice de complexidade associado a cada módulo é possível identificar quais os mais complexos, e portanto mais propensos a falhas, e mesmo estabelecer um limite para a complexidade dos módulos, reorientando, assim, a partição do sistema.

Quanto a componentes hardware, a identificação de componentes críticos se faz com base em métricas de complexidade já relacionadas nas normas anteriormente citadas. Os fatores que definem se um módulo é mais complexo que outro são a tecnologia empregada, o número de pontos de solda, o número de componentes, etc...

1.3.3.4 - Verificação da Segurança de Funcionamento em Laboratório

Durante as fases de teste a avaliação periódica de qualidade é necessária ao propósito de acompanhar a evolução do sistema. A partir do comportamento observado é possível inferir o seu comportamento futuro, a fim de estimar se o sistema atingirá em tempo as metas fixadas ou se é necessário alterar a programação de testes.

Acompanhamento da Taxa de Falha de Componentes Críticos: Durante as fases de teste do sistema alguns protótipos já estão operacionais, em laboratório. Os componentes críticos identificados na fase anterior devem ser acompanhados a fim de determinar se a taxa de falha em operação confirma a que foi estimada. Isto pode levar a uma alteração do projeto, no caso do componente se mostrar realmente ofensor. Como a população de módulos é reduzida deve-se fazer o acompanhamento da taxa de falha dos componentes críticos dimensionando os protótipos de modo a obter uma população adequada ao estudo.

Projeção do Número de Defeitos de Concepção Detectados em um Intervalo de Tempo Futuro: Os dados de falha de concepção hardware e software devem ser registrados durante toda a fase de testes. O processo de falha de concepção a partir do teste de integração hardware/software deve ser acompanhado a fim de verificar se a taxa de falha aumenta ou diminui ao longo do tempo. A partir do comportamento do processo de falha pode-se projetar qual o número de defeitos corrigidos dentro do prazo estipulado para a atividade de teste.

Previsão do Tempo até a Obtenção dos Índices de Segurança de Funcionamento Especificados: a partir da evolução do processo de falha é possível estimar o tempo de teste necessário para atingir os requisitos especificados. Esta informação é útil na reorientação da atividade de teste e dos cronogramas do projeto.

Avaliação da Eficácia dos Testes: O acompanhamento da qualidade de concepção durante a fase de teste pode dar uma visão da eficácia das atividades de projeto, mostrando quais as funções testadas que revelaram mais defeitos, os tipos de defeitos mais frequentes, etc... Os resultados desse acompanhamento devem influir na atividade de teste, mostrando quais as funções do sistema para as quais mais testes devem ser especificados ou não.

Previsão de Taxas de Falha de Concepção: comparando o desempenho de falha de concepção dos módulos do sistema com outros módulos de funções similares de outros sistemas desenvolvidos na mesma estrutura de desenvolvimento pode-se fazer uma projeção do seu comportamento futuro.

Avaliação Qualitativa do Sistema: a partir das indicações dos processos de falha de componentes e de concepção em laboratório é possível fazer uma primeira avaliação do sistema em relação a requisitos sistêmicos de segurança de funcionamento. Essa é

uma primeira etapa da atividade de avaliação que será completada com os dados colhidos durante a vida operacional.

1.3.3.5 - Validação da Segurança de Funcionamento em Campo

Durante a vida operacional do produto é muito importante acompanhar a confiabilidade, a fim de qualificá-lo, determinando o índice de segurança de funcionamento atingido.

Relação entre as Taxas de Falha de Componentes Estimadas e Observadas: Devem ser coletados dados de falha de componentes ao longo de toda a vida operacional do sistema. As estimativas das taxas de falha feitas com base em normas devem ser validadas em relação às taxas de falha apresentadas em campo. Atenção especial deve ser dada à taxa de falha dos componentes críticos.

A estimação da segurança de funcionamento feita para os módulos hardware e software a partir de métricas de complexidade deve ser validada fazendo-se um estudo de correlação entre a complexidade dos módulos e as falhas realmente observadas.

Os defeitos de concepção detectados em campo são geralmente corrigidos sem utilizar a metodologia de desenvolvimento, aumentando a probabilidade da depuração ser imperfeita. Dessa forma, a correção de um defeito pode estar introduzindo novos defeitos. Isto tem sido fonte de problemas para o usuário e para o fornecedor, que sentem que a manutenção está degradando o produto, porém não têm como avaliar quantitativamente este fenômeno. Neste caso o acompanhamento da segurança de funcionamento pode avaliar a eficácia da manutenção.

Qualificação do Sistema em Relação aos Requisitos de Projeto, Quanto a Falhas de Componentes e de Concepção: Os índices de segurança de funcionamento devem ser avaliados a partir dos processos de falha de componentes hardware e de concepção hardware e software a fim de determinar se os requisitos de qualidade estabelecidos na fase de especificação do sistema foram atingidos.

Os resultados esperados são os índices de segurança de funcionamento atingidos levando em conta os processos de falha de componentes e de concepção, uma estimativa do tempo até atingir os índices especificados e do esforço de manutenção da concepção.

Identificação de Oportunidades de Introdução de Melhorias: A alteração do projeto do sistema a fim de implementar novas funções ou melhorias pode introduzir novos defeitos ou criar situações de utilização que levem defeitos, latentes até então, a se manifestar. O acompanhamento da evolução da taxa de falha de concepção indica as situações mais favoráveis à introdução de novas funções no sistema, ou seja, ocasiões em que a taxa de falha é bastante baixa para que a equipe de manutenção suporte sua eventual elevação.

1.4 Conclusão

Neste capítulo foi introduzido o conceito de falha de concepção englobando hardware e software. O problema de segurança de funcionamento de sistemas computacionais foi abordado de um ponto de vista mais amplo, abrangendo não só os aspectos de falha de componentes hardware, como também os de concepção hardware e software.

A segurança de funcionamento é interpretada como uma atividade a ser desenvolvida ao longo de todo o ciclo de vida de um produto, a fim de se obter sistemas mais confiáveis. Propõe-se que os requisitos de qualidade de serviço sejam avaliados com base nos aspectos de componentes e de concepção.

O cerne de um programa de segurança de funcionamento está na avaliação deste atributo quanto a cada um dos tipos de falha considerados. A avaliação de segurança de funcionamento de componentes hardware, através do cálculo de sua taxa de falha, é baseada em teoria já amadurecida e baseada em modelos matemáticos para os diversos componentes. A avaliação de segurança de funcionamento de concepção é também baseada em modelos matemáticos a partir dos quais a taxa de falha é determinada. No Capítulo 2 será tratado o problema da avaliação da segurança de funcionamento quanto a concepção.

Capítulo 2

Avaliação e Previsão da Segurança de Funcionamento Quanto a Concepção

2.1 Introdução

A determinação da taxa de falha e de sua tendência a crescimento ou decrescimento, é o ponto central dentro de um Programa de Segurança de Funcionamento de sistemas.

A avaliação da segurança de funcionamento quanto a concepção é baseada nos dados de falha coletados durante as fases de teste e de operação comercial do sistema. A análise desses dados indicará se o sistema está passando por um período de crescimento ou decrescimento da taxa de falha, ou se esta apresenta um padrão constante. A hipótese básica é que um sistema bem concebido tende, mais cedo ou mais tarde, a uma condição em que o número de falhas se torna monotonamente decrescente no tempo. Este comportamento é conhecido na literatura técnica da área como **crescimento de confiabilidade**, embora indique o crescimento de um atributo mais geral que é o da segurança de funcionamento. Atingido este padrão de crescimento de confiabilidade podem ser feitas previsões do comportamento futuro do processo de falha baseadas nos dados de falha até então registrados.

Durante a fase de testes do sistema é feito o acompanhamento da segurança de funcionamento. Observa-se, na prática, que durante a fase de testes o sistema não apresenta um crescimento sistemático de confiabilidade. Neste período o sistema oscila entre crescimento e decrescimento de confiabilidade em função da sequência de testes aplicada e das correções das falhas detectadas. Durante a vida operacional, quando o sistema já atin-

giu o padrão de crescimento monótono de confiabilidade, é feita a avaliação e previsão da segurança de funcionamento. Estas atividades são realizadas através da avaliação periódica do sistema.

Este capítulo trata do problema de avaliação da confiabilidade de concepção de sistemas, discutindo inicialmente o objeto da avaliação de acordo com o propósito gerencial a que ela deve servir. O insumo básico para a avaliação, que são os dados de falha de concepção, são definidos, e discutidos os aspectos relacionados ao seu tratamento. Apresenta-se a forma de avaliação da concepção, através dos modelos de crescimento de confiabilidade originalmente desenvolvidos para a avaliação da confiabilidade de software.

2.2 Modelo de Gerência da Concepção

O grande porte dos sistemas computacionais desenvolvidos hoje em dia exige um tratamento mais "industrializado", contrastando com as técnicas de controle mais simples dos pequenos projetos de duas décadas atrás.

Um sistema é composto por um conjunto de módulos hardware e software e pelos documentos associados a esses módulos (módulos de documentação), ao sistema e à sua utilização. Nos sistemas modernos há dezenas de módulos de cada tipo, a sua maioria com considerável complexidade de desenvolvimento.

Um modelo de gerência tem que levar em conta diversos aspectos ligados ao desenvolvimento, visando a qualidade e a produtividade. O controle da evolução do sistema dentro e fora do ambiente de desenvolvimento é um dos aspectos importantes de um modelo de gerência.

Um modelo clássico de gerência da evolução de sistemas software de grande porte [JOH 85] divide o tratamento do sistema em aspectos de projeto, de marketing e de produção. Este modelo pode ser generalizado para o caso de sistemas compostos de hardware e software e para ambientes nos quais a equipe de desenvolvimento é independente da de produção. Uma visão esquemática deste modelo é apresentada na Figura

2.1, onde os principais conceitos são: Sistema Fonte, Sistema de Produção, Sistema de Aplicação e Instalações.

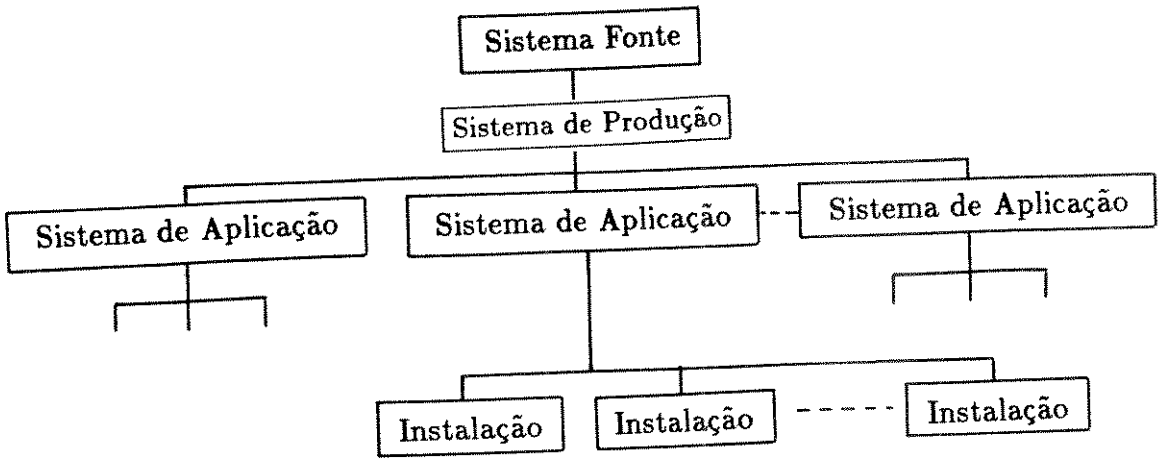

Figura 2.1 - Modelo de Gerência de Sistemas de Grande Porte.

As atividades de projeto são desenvolvidas dentro do contexto do Sistema Fonte. Cada módulo hardware, software ou de documentação depois de desenvolvido, testado e/ou revisto separadamente é confrontado com o Sistema Fonte no Teste de Integração, a fim de garantir sua interoperabilidade. O Sistema Fonte é integrado gradualmente incorporando mais módulos e portanto mais funções. Na verdade novos módulos podem ser incluidos em um ou mais Sistemas Fontes (no caso de reutilização de módulos de projetos anteriores). Os módulos do Sistema Fonte podem ser combinados em um certo número de aplicações diferentes em relação ao porte e funções. Estes módulos evoluem individualmente.

O Sistema de Produção é um instantâneo do Sistema Fonte liberado para servir de base para a produção, ou seja, para produzir os Sistemas de Aplicação. O Sistema de Produção forma um todo coerente e interoperante que serve como referência durante um certo tempo, enquanto que o Sistema Fonte não pára de evoluir.

O propósito de um Sistema de Aplicação é servir de base para a produção de um número de Instalações a partir de um subconjunto dos módulos do Sistema de Produção. Um sistema de Aplicação pode ser compreendido como uma versão do sistema criada através de restrições da flexibilidade do Sistema de Produção a fim de atender a uma solicitação particular do mercado. A partir de um Sistema de Aplicação é produzido um número de Instalações similares.

Uma Instalação do sistema é produzida a partir do Sistema de Aplicação de forma particularizada para as condições específicas de capacidade e ambiente de utilização. O hardware é selecionado a partir do Sistema de Aplicação e dimensionado individualmente para cada Instalação; os módulos software e de documentação são produzidos a partir do Sistema de Aplicação.

Resumindo: o trabalho de concepção gera o Sistema Fonte, que atualiza periodicamente o Sistema de Produção. Um Sistema de Aplicação nasce a partir do estudo mercadológico, ou seja, das necessidades que tem um certo mercado dentre as diversas funções do Sistema de Produção. As Instalações surgem a partir do trabalho de produção, de particularização do sistema ao seu ambiente de funcionamento e tipo de solicitação.

Os protótipos de laboratório são as Instalações, em fase de desenvolvimento, dimensionadas a partir de um ou mais Sistemas de Aplicação, de acordo com a conveniência de teste.

2.2.1 Gerência da Evolução do Sistema

A correção dos defeitos de componentes hardware causadores de uma falha mantém o sistema, restaurando a mesma qualidade que ele apresentava antes da falha. A evolução do sistema em termos de qualidade pode ser feita atavés da correção de defeitos de concepção, da introdução de melhorias ou pela incorporação de novas funções.

Dentro do contexto de Sistema Fonte a correção de um defeito em um módulo gera uma nova edição desse módulo, que substitui a versão anterior e lhe é totalmente compatível. A introdução de uma nova função ou melhoria em um módulo pode levar a uma

nova **versão** do mesmo, que convive com a anterior, sendo-lhe alternativa na composição dos Sistemas de Aplicação.

Uma falha é sempre detectada nas Instalações. O estudo da falha leva à identificação do(s) defeito(s) em um ou mais módulos do sistema que a provocaram. Os defeitos devem ser corrigidos nos módulos do Sistema Fonte.

A evolução dos módulos do Sistema Fonte em novas versões e edições pode levar a uma decisão gerencial de liberação de uma nova edição do Sistema de Produção. É possível que alguns módulos da nova edição do Sistema de Produção não tenham sido alterados e que outros tenham sofrido várias edições ou versões em relação à edição anterior do Sistema de Produção.

Um Sistema Fonte evolui a partir da evolução em edições ou versões de seus módulos, separadamente, de modo que o Sistema Fonte como um todo não sofre edições nem versões. A necessidade de criação de uma outra versão do Sistema de Produção caracteriza um novo produto. Este sistema só evolui em edições para um mesmo produto. Da mesma forma, uma nova versão de um Sistema de Aplicação é na verdade um outro Sistema de Aplicação. A evolução deste sistema é feita somente através de edições.

Uma nova edição do Sistema de Produção pode ou não levar a uma nova edição dos Sistemas de Aplicação e Instalações, dependendo dos módulos afetados pertencerem ou não a esses subconjuntos. No caso dos módulos afetados pertencerem a um Sistema de Aplicação ou Instalação, a correção dependerá da política de manutenção do produto e da maneira como os defeitos corrigidos na nova versão afetam o funcionamento das Instalações, como será visto no item 2.4.

A gerência de versões e configurações de um sistema coordena a evolução do sistema ao serem corrigidos os defeitos de concepção inerentes ao processo de desenvolvimento. A caracterização desta evolução se faz através da avaliação da segurança de funcionamento. O objeto de análise é o processo de falha de concepção do sistema.

2.3 Tratamento de Falhas de Concepção

A qualidade da concepção de um sistema está diretamente relacionada a seus defeitos. Os defeitos latentes tanto podem levar o sistema a oferecer um serviço que não atenda às especificações de segurança de funcionamento quanto demandar um trabalho de manutenção.

Um sistema pode ser avaliado quanto à sua concepção através da série histórica da detecção de falhas ou de sua correção. Para isso as falhas de concepção e de componentes devem ser bem caracterizadas. A ocorrência de falha deve ser convenientemente registrada ao longo das fases de desenvolvimento e operação comercial. A partir dos dados colecionados são feitas análises qualitativas e quantitativas do processo de falha do sistema.

2.3.1 Falhas de Concepção

Antes de mais nada é conveniente precisar o que se entende por falha de concepção. A primeira observação é de que, no contexto sistêmico, o conceito de falha (ver Capítulo 1) está associado ao serviço fornecido pelo sistema e portanto supõe que uma Instalação esteja operando total ou parcialmente. Por esta definição são consideradas as falhas de concepção as detectadas nas Instalações durante as fases de teste e operação. Este conceito não engloba os defeitos detectados durante inspeções e revisões estáticas do projeto.

Os defeitos de concepção são introduzidos no Sistema Fonte e propagados pelo Sistema de Produção, Sistemas de Aplicação e Instalações, já que os seus módulos são cópias dos módulos do Sistema Fonte.

É importante não confundir falhas com melhorias, inclusão de novas funções ou alterações no projeto devido a fatores externos, como, por exemplo, a suspensão da fabricação de componentes hardware empregados. As falhas do sistema não incluem as falhas de seu ambiente de desenvolvimento (banco de dados, ferramentas de teste, compiladores, sistemas de CAD/CAM, etc...). Estas falhas devem ser contabilizadas nas bases de dados relativas a estes outros sistemas. Em sistemas tolerantes a defeitos só

devem ser consideradas as falhas que levam a uma degradação do serviço não prevista na especificação.

2.3.2 Registro de Falhas de Concepção

O ponto nevrálgico de um programa de garantia de segurança de funcionamento está em obter dados confiáveis sobre os quais basear as avaliações. Estes dados, relativos a falhas, datas e condições de operação do sistema, devem ser paciente e minuciosamente registrados ao longo das etapas de teste e operação.

Alguns dos mais importantes dados de projeto ficam gravados apenas na memória de quem participou ativamente do desenvolvimento. Devido à volatilidade deste registro as informações acabam por não servir a ninguém. É necessário, antes de mais nada, o compromisso com a formalização destes dados, que só é possível através da compreensão de sua utilidade.

O registro de dados de falha deve ser feito de modo a :

- tornar o mecanismo de coleta o mais fácil possível,
- coletar e "limpar" os dados em tempo real,
- não coletar dados cuja finalidade não esteja bem clara,
- gerar relatórios periódicos para a equipe envolvida na atividade de detecção e registro das falhas e correção dos defeitos assim como para o gerente do projeto. Tanto a gerência como os projetistas devem estar motivados para a qualidade e devem saber interpretar os resultados dos relatórios.

Dentro deste painel é importante que a equipe responsável pela avaliação e previsão da segurança de funcionamento participe da definição do que se entende por falha no sistema e das atividades de teste, de modo que os conceitos e métodos possam ser revistos.

Uma falha de concepção é caracterizada pelos seguintes atributos:

1. Data de detecção da falha,

2. Instalação na qual a falha foi detectada,

3. Descrição da falha e de suas condições de ocorrência,

4. Gravidade: medida da consequência da falha no serviço recebido pelo usuário,

5. Abrangência: módulos hardware, software, firmware e de documentação afetados,

6. Tipo: relacionado com a atividade na qual o defeito foi introduzido (especificação, implementação, etc...),

Um exemplo de formulário de registro de dados de falha de concepção é dado no anexo A. Este formulário foi criado com o propósito de coletar dados de falha de concepção de todos os sistemas desenvolvidos no Departamento de Comutação do CPqD TELEBRÁS. É dirigido ao controle de modificações devido a falhas de concepção e à avaliação de confiabilidade, dentro da idéia de um Programa de Segurança de Funcionamento tal como discutido no capítulo 1.

Uma falha de concepção já registrada pode se repetir em Instalações diferentes do sistema e até na mesma Instalação onde ela foi primeiramente detectada. Isto pode ocorrer enquanto o defeito que gerou a primeira ocorrência não for corrigido. Como considerar a repetição de falhas?

Durante as atividades de teste a preocupação é com a qualidade do produto (evolução da taxa de falha, atendimento dos requisitos de qualidade de serviço) e com a eficácia dos testes. O objeto de acompanhamento é o Sistema de Produção em si, uma vez que este é a representação do produto. Para a avaliação destes atributos só a primeira ocorrência de uma falha é importante pois ela identifica a necessidade de manutenção e a sua correção implica em uma melhoria do produto.

Em operação comercial, além da preocupação com os aspectos ligados ao produto, passam a ser importantes os atributos vistos pelo usuário, que são a confiabilidade e a disponibilidade. O conceito de confiabilidade está relacionado ao tempo até a próxima falha, seja ela repetida ou não. A indisponibilidade do sistema dá a medida do tempo

em que o serviço fornecido ficou fora de suas especificações devido a uma falha de concepção, independentemente dela já ter sido detectada ou não. Neste caso é necessário o registro de todas as falhas detectadas em todas as Instalações do sistema e de alguns dados adicionais, tais como duração da atividade de correção do defeito que originou a falha, frequência de repetição da falha e tempo de indisponibilidade do serviço após a ocorrência da falha.

Seja um Programa de Segurança de Funcionamento cujo objetivo é acompanhar a manutenção do sistema ao longo de todo o ciclo de vida e a sua disponibilidade e confiabilidade em vida operacional. A política de registro de falhas que é a mais adequada aos objetivos propostos consiste em relatar a primeira ocorrência de uma falha quando o sistema está em teste em laboratório e todas as ocorrências de falha em cada Instalação operacional. Esta política permite:

- Durante as fases de teste, acompanhar a evolução do Sistema de Produção e da eficácia dos testes,
- Durante a operação comercial, avaliar e prever:
  - a mantenabilidade do Sistema Fonte (qualidade da concepção do sistema),
  - a mantenabilidade de um dado Sistema de Aplicação (qualidade da concepção do sistema sob o ponto de vista de uma certa aplicação),
  - a confiabilidade e a disponibilidade de uma Instalação em particular (qualidade da concepção do sistema do ponto de vista de um certo usuário).

2.3.3 Representação de Falhas de Concepção

O processo de detecção de falhas de concepção pode ser representado de várias maneiras no tempo. As mais utilizadas são:

(a) Número de falhas detectadas a cada unidade fixada de tempo: estipulada uma base de tempo, este método consiste na contagem do número de falhas detectadas dentro de cada unidade de tempo consecutiva.

(b) Intervalo entre Falhas: registro dos intervalos de tempo ordenados entre a detecção de falhas consecutivas.

O método do número de falhas em um intervalo de tempo reune os dados relativos a várias ocorrências em um só dado de falha e por isso é o mais indicado quando se trata de sistemas cujo número de falhas é elevado. Por outro lado, as informações sobre o processo de falha dentro do intervalo-padrão de tempo são perdidas nessa concentração.

Representando-se um processo de falha através do número acumulado de falhas a cada intervalo fixo de tempo perde-se a informação de como as falhas ocorreram dentro do intervalo. O método do intervalo entre falhas é mais exato porém pode levar a uma quantidade de dados muito grande.

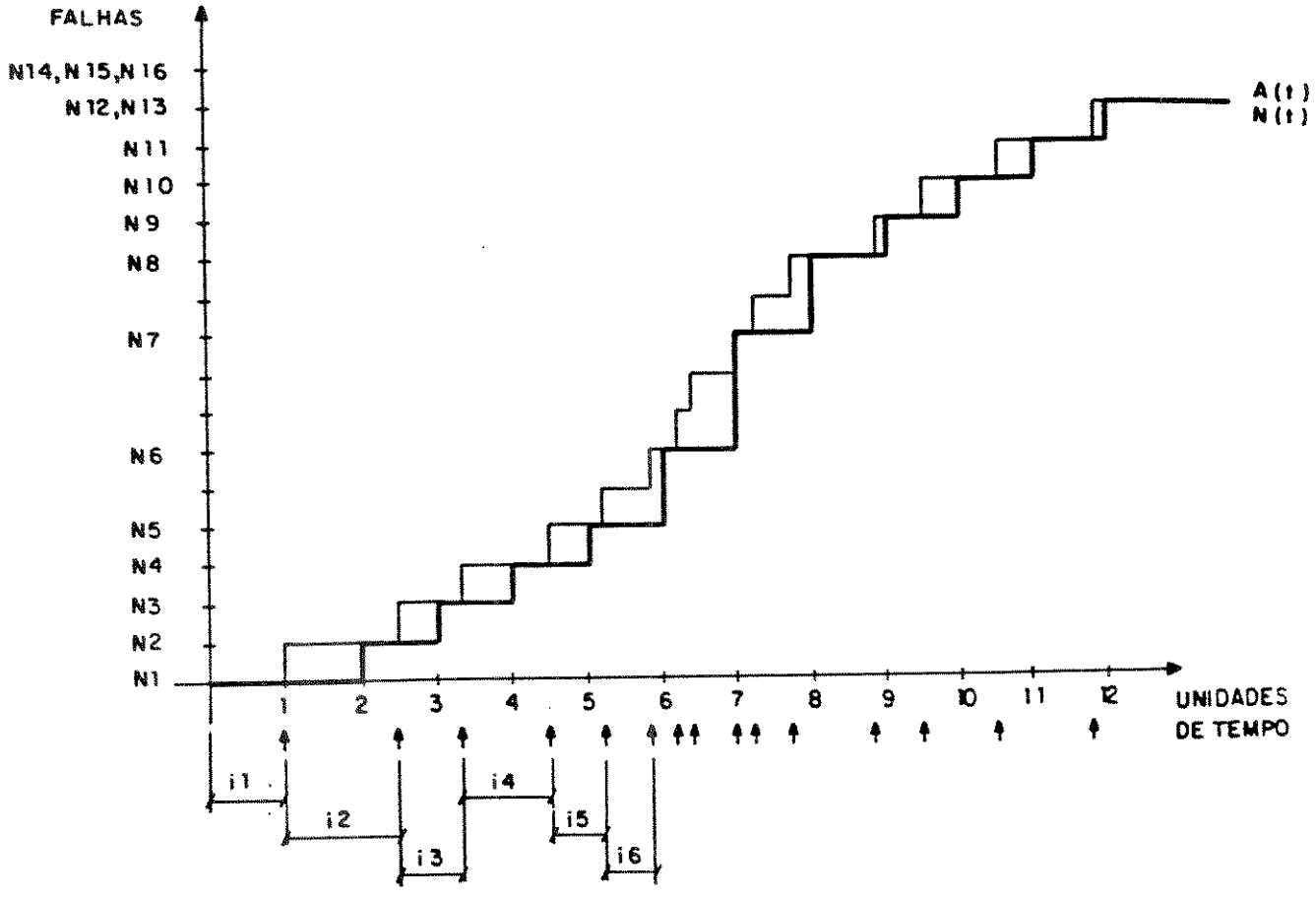

Figura 2.2 - Representação da curva de falha de um sistema através do intervalo entre falhas e do número de falhas em um intervalo fixado de tempo.

A representação do número acumulado de falhas no tempo segundo um desses dois métodos de representação forma o que se chama de curva de falha do sistema. Seja $N(t)$ o processo de contagem do número de falhas detectadas até o instante t e $A(k.tu)$ o processo de contagem do número de falhas acumulado até a k-ésima unidade de tempo tu. A re-

presentação da curva de falha de um sistema por estes dois métodos é vista na Figura 2.2.

Uma curva de falha N(t) côncava indica que o número de falhas por unidade de tempo aumenta no tempo (ou, o que é equivalente, o intervalo de tempo entre falhas diminui no tempo). Esse comportamento é conhecido como decrescimento de confiabilidade. Ao contrário, uma curva de falha convexa indica que o intervalo entre falhas aumenta no tempo e que o número de falhas por unidade de tempo diminui. Esse comportamento é denominado crescimento de confiabilidade. Um processo de falha estacionário, com taxa de falha aproximadamente constante, é correspondente a uma curva de falha que pode ser aproximada por uma reta.

2.3.4 Acompanhamento Qualitativo da Concepção

O objetivo do acompanhamento qualitativo da concepção a partir dos dados de falha é construir a curva de falha do sistema e avaliar de maneira qualitativa o desempenho do sistema em relação a falhas de concepção.

Supondo que sejam registradas as informações relacionadas no item 2.3.2 pode-se obter as seguintes informações:

Número de falhas por grau de severidade: número de falhas graves, sem maiores consequências, etc... Indica qual a parcela de falhas de concepção que leva a perda da função do sistema.

Número de falhas por módulos afetados: quantas falhas afetaram cada módulo do sistema. Esta informação, se confrontada com a previsão de falhas feita anteriormente dentro do Programa de Segurança de Funcionamento, pode indicar se o módulo está sendo convenientemente testado e validar as técnicas de previsão utilizadas (Métricas de Complexidade de Software, de Hardware, etc...). É possível identificar os módulos mais ofensores do sistema.

Número de falhas por tipo de módulo: quantas falhas afetaram o hardware, o soft-

ware, a documentação ou uma combinação desses elementos. Com esse resultado pode-se ponderar a importância da concepção do hardware, do software e da documentação no sistema como um todo.

Número de falhas por tipo de falha: o tipo de falha está normalmente relacionado à atividade de projeto durante a qual o defeito foi introduzido (número de falhas de especificação, codificação, etc...). Esta análise revela as atividades de desenvolvimento responsáveis pela introdução das parcelas mais significativas dos defeitos. Os resultados dão uma idéia da qualidade de execução das tarefas e podem realimentar a estrutura de desenvolvimento, sugerindo treinamento, melhoria dos padrões de especificação, etc...

Número de falhas que levaram a indisponibilidade dos serviços: assumindo uma política de manutenção, pode-se calcular a indisponibilidade total ou parcial do sistema devido a falhas de concepção.

Número de falhas por Sistema de Aplicação: indica o processo de falha típico de uma certa aplicação e o processo de falha esperado de uma nova Instalação produzida a partir daquele sistema.

Número de falhas por Instalação: indica o processo de falha percebido pelo usuário de cada Instalação.

2.3.5 Gerência das Atividades de Teste

Estudando estatisticamente o número de falhas de cada tipo em cada fase de teste e operação, é possível ter uma idéia de quais os tipos de defeitos mais comuns em cada fase. Como o custo de reparação de defeitos de concepção aumenta à medida que o sistema avança no seu ciclo de vida, esse estudo mostra também os tipos de defeitos mais onerosos.

A meta das atividades de teste é verificar e validar o sistema quanto a defeitos de concepção. Ao final da atividade de teste isolado dos módulos hardware e software, o sistema contém latentes todos os defeitos que não puderam ser identificados nas atividades de

revisão. A detecção de uma falha em testes de laboratório leva à depuração do Sistema Fonte com um certo custo de manutenção. Nas Instalações comerciais a ocorrência de uma falha gera um custo de manutenção mais alto além de denegrir a imagem de qualidade do produto. Deste modo, o desejável é ativar o maior número possível de falhas em laboratório e o menor número delas em vida operacional.

Normalmente os casos de teste são especificados funcionalmente, explorando as formas de ativação de uma dada função do sistema. Espera-se que logo nos primeiros testes seja detectada a maioria das falhas de concepção. A partir de um certo momento a eficácia dos testes na ativação de defeitos deve diminuir até se tornar praticamente nula devido à diminuição do número de defeitos latentes. Quando se troca a função a ser testada ou o tipo de teste aplicado, a curva pode subitamente se tornar convexa, indicando que a nova função contém defeitos que a nova bateria de testes está sendo capaz de ativar. A curva de falha em laboratório pode então apresentar uma sucessão de períodos de aumento e de diminuição da taxa de falha.

O acompanhamento do número de falhas ativadas pode ajudar na gerência da atividade de teste:

1. Seja uma certa bateria de testes funcionais para cuja aplicação foi alocado um certo intervalo de tempo e recursos. Se ela já se mostra pouco eficiente na ativação de falhas, isto sugere que a função já está razoavelmente depurada ou que a bateria de testes é ineficiente para mostrar. Seria estrategicamente mais indicado, neste caso, passar a uma nova bateria de testes, deixando os restantes, relativos à função anterior, para serem aplicados ao final do período de tempo previsto. Por outro lado se, ao final da aplicação de uma certa bateria de testes funcionais, a curva de falha ainda for côncava, isto indica que o processo de depuração daquela função deveria continuar por mais algum tempo.

2. É possível estabelecer limites máximo e mínimo para o número de falhas. Estes limites podem ser definidos a partir de índices máximos aceitáveis de falhas por linha de programa, por módulo de hardware, pelo esforço de manutenção exigido, etc... Um número de falhas detectadas acima do limitante superior indica que

a qualidade do sistema está abaixo do esperado/suportado. Pelo contrário, um número de falhas detectadas abaixo do limitante inferior (que pode ser definido como uma porcentagem do superior) revela uma atividade de teste com eficiência menor que a esperada, sugerindo uma revisão da sua programação.

2.3.6 Correção das Falhas de Concepção

Em laboratório a política de manutenção comumente usada é a de corrigir todos os defeitos que deram origem a falhas nos protótipos de desenvolvimento. As correções podem ser feitas uma a uma, à medida que as falhas se manifestarem, ou em grupos, periodicamente ou cada vez que o volume ou gravidade das falhas for considerável. Cada correção ou conjunto de correções dá origem a uma nova versão ou edição do módulo no Sistema Fonte. Os casos de teste mal sucedidos devido a falhas já detectadas porém não corrigidas são normalmente reapresentados após a correção.

Em operação comercial, geralmente, cada falha de componente hardware dá origem a uma intervenção. Um defeito de concepção dá origem a um número de intervenções que é função do número de sistemas em operação e da política de manutenção adotada.

Seja uma política de manutenção cuja premissa básica é a de corrigir todos os defeitos revelados em um sistema. Há dois tipos de manutenção:

1. manutenção corretiva: relacionada a falhas de componentes e de concepção, feita nas instalações do produto onde essas falhas foram detectadas, e

2. manutenção preventiva: relacionada a falhas de concepção, feita nas instalações do produto onde o defeito está latente, porém ainda não ocasionou uma falha.

Essas políticas de manutenção podem levar a diversas políticas de intervenção nos sistemas instalados:

- Intervenção imediata nas instalações que detectaram falhas de componentes ou de concepção (manutenção corretiva) e nas que possuem falhas de concepção latentes (manutenção preventiva).

- Intervenção imediata nas instalações que detectaram falhas de componentes ou de concepção e não intervenção nas instalações em que elas permanecem latentes até que uma falha de qualquer natureza demande uma intervenção naquela instalação.

No primeiro caso a evolução das Instalações é mais rápida em termos de segurança de funcionamento, porém o custo é maior.

2.4 Metodologia de Avaliação da Qualidade de Concepção

As atividades de estimação e previsão de segurança de funcionamento durante a fase de teste e de acompanhamento ao longo da vida operacional do sistema são feitas através de avaliações periódicas. O intervalo entre avaliações depende da fase de desenvolvimento. Em laboratório é interessante avaliar com maior frequência pois o processo de falha pode variar rapidamente. Depois de algum tempo de operação, quando o comportamento da curva de falha é menos sujeito a variações, a avaliação pode ser mais espaçada.

Os resultados da avaliação devem servir à equipe de teste e à gerência de desenvolvimento. A forma de apresentação deve ser a mais legível e concisa possível:

Em laboratório: devem ser afixados gráficos que descrevam o comportamento geral do sistema, porém enfatizem a situação do momento;

Para a gerência: deve-se apresentar os resultados sob a forma de relatório de acompanhamento contendo os aspectos qualitativos e quantitativos da avaliação e a interpretação dos resultados, indicando as ações gerenciais que estes sugerem. Algumas informações que devem estar contidas em um Relatório de Acompanhamento da Segurança de Funcionamento são:

a) Histórico da utilização do produto: registro de alterações na forma de solicitação do sistema (mudança da função a ser testada em laboratório, introdução de novas funções no sistema, liberação de nova carga software, aumento do número de sistemas em operação comercial, etc...). Estas informações são importantes para a correta interpretação da

forma da curva de falha do sistema. Uma mudança considerável na forma de utilização, por exemplo, pode explicar uma reversão no comportamento do processo de falha, enquanto que um período de férias coletivas deve ser desconsiderado no estudo de avaliação.

b) Avaliação Qualitativa da Segurança de Funcionamento: análise estatística dos fatores que descrevem as falhas e defeitos detectados no sistema. Os cruzamentos de dados são feitos de acordo com os interesses de cada projeto e da gerência correspondente.

c) Análise Quantitativa do Sistema: resultados da avaliação quantitativa da taxa de falha do sistema, interpretação dos resultados obtidos e previsão do comportamento do processo de falha a partir do observado até aquele momento.

d) Ações Sugeridas: a partir das análises qualitativa e quantitativa do sistema é possível gerar sugestões que sirvam de realimentação ao processo de desenvolvimento do sistema. Estes retornos podem ser do tipo intensificar os testes a fim de atingir os índices de segurança de funcionamento desejados no prazo previsto, passar ao teste de uma nova função ou buscar explicação para um determinado comportamento da curva de falha.

Um exemplo de Relatório de Acompanhamento de Segurança de Funcionamento, que está sendo usado no acompanhamento de diversos sistemas desenvolvidos no CPqD TELEBRAS, é mostrado no Anexo B.

2.5 Base de Tempo

Há basicamente duas unidades de tempo usadas na avaliação de confiabilidade : tempo de calendário e tempo de operação (que para o software se caracteriza pelo tempo de execução). Hoje em dia é consenso que o tempo de operação é preferível, uma vez que ele representa melhor o esforço de solicitação do sistema [MUS 84].

Nem sempre, no entanto, é possível medir o tempo de máquina utilizado. Seja um sistema distribuido, composto por m módulos diversos, cada um n-uplicado, sendo que o valor de n é específico para cada módulo e depende da configuração de cada Instalação. No caso do sistema não dispor de facilidade que contabilize o tempo de máquina consu-

mido por cada cópia, percebe-se a dificuldade de adoção dessa base de tempo.

Do ponto de vista de avaliação da segurança de funcionamento, o processo de falha está relacionado ao tempo de execução; por outro lado as atividades de teste e gerência do sistema estão relacionadas ao tempo de calendário e, portanto, as medidas de segurança de funcionamento terão maior significado quando expressas nessa base de tempo.

Uma das formas de representação da curva de falha de um sistema, quer seja utilizado tempo de máquina ou de calendário, supõe a fixação de uma unidade básica de tempo. A duração desta unidade deve ser orientada pela duração do período de observação e pelo número de falhas detectadas. Uma unidade de tempo muito grande comprime os dados de falha de modo que as informações locais dentro de cada unidade são perdidas. Por outro lado, uma unidade de tempo que seja tão breve a ponto de gerar muitas unidades sem ocorrência de falhas aumenta o número de dados, levando a um desperdício de esforço na modelagem.

A duração da unidade de tempo depende também:

- da fase do ciclo de vida: pode ser menor durante as fases de teste, quando a ocorrência de falhas é mais frequente,
- da necessidade dos resultados: durante um período crítico a unidade de tempo deve ser menor do que num período de normalidade,
- da forma de utilização do sistema: numa fase de intensa utilização a unidade de tempo deve ser pequena a fim de não se perder muita informação diodo.

2.5.1 Esforço de Teste

Na seção anterior foi visto que as medidas de segurança de funcionamento expressas em termos de tempo de calendário são mais significativas para os projetistas e gerentes e o processo de falha é mais facilmente registrado nesta base de tempo. No entanto a ocorrência de falhas está mais precisamente ligada ao tempo de operação ou execução do sistema. A conversão entre o tempo de calendário e o tempo de execução é feita através

de alguma grandeza que caracterize a maneira como os recursos humanos e computacionais são aplicados ao longo do desenvolvimento e vida operacional do sistema.

Uma maneira de relacionar o tempo de calendário ao tempo de execução é através do esforço de teste. É razoável que sejam detectadas mais falhas durante um esforço concentrado de teste do que durante épocas de pouca utilização do sistema. Da mesma forma a taxa de falha em operação comercial pode ser influenciada pela quantidade de Instalações do equipamento no mercado. A curva de falha expressa com base no tempo de calendário pode ser ponderada pela expressão do esforço de teste $E(t)$ nessa mesma base.

O esforço de teste $E(t)$, pode ser medido pelo número de casos de teste executados, quantidade de homens-hora envolvidos no teste, tempo gasto na execução dos testes, número de sistemas em funcionamento, etc... As formas mais simples de $E(t)$ são: esforço de teste constante, esforço de teste linearmente crescente ou decrescente, exponencial decrescente, ou dado por uma função de Rayleigh [YAM 86].

A curva do esforço de teste pode ser representada por uma dessas expressões, a partir do conhecimento do processo de teste. Os dados relativos ao esforço de teste aplicado devem, assim, ser registrados, na unidade de medida escolhida, a cada unidade de tempo de calendário. A unidade de tempo deve ser a mesma usada para o registro de falhas.Os parâmetros da curva de esforço podem ser estimados a partir dos dados de esforço de teste. A curva de falha, neste caso, se refere ao comportamento de falha do sistema na base de tempo de operação.

2.6 Modelos de Confiabilidade de Concepção

A avaliação de confiabilidade de concepção é feita através de modelos matemáticos inicialmente concebidos para avaliar processos de falha software. Como os defeitos de concepção do hardware têm as mesmas características do processo de falha software (vide capítulo 1), propõe-se generalizar a utilização destes modelos para avaliar a evolução de processos de falha de concepção.

Devido às características das falhas de concepção, os modelos de confiabilidade partem de uma premissa básica: a longo prazo um sistema cresce em confiabilidade. A detecção e remoção de uma falha de concepção diminui o número de falhas latentes e torna o sistema menos propenso a falhar. Alguns modelos admitem a possibilidade de decrescimento de confiabilidade no período inicial de funcionamento, seguido de posterior crescimento de confiabilidade.

Um modelo de confiabilidade de software especifica a forma geral da dependência do processo de falha em relação a fatores como: introdução de defeitos, remoção de defeitos e ambiente de operação. Em geral os modelos se distinguem entre si pela distribuição de probabilidade dos instantes de falha ou número de falhas detectadas e pela natureza da variação do processo aleatório de falha no tempo.

A partir de certas hipóteses sobre a natureza do processo de falha é derivada uma expressão analítica que representa a dependência do processo de falha no tempo, em relação aos fatores acima mencionados. Esta expressão tem parâmetros que são estimados a partir dos dados de falha registrados, de modo que a representação seja o mais próximo possível da série de dados reais.

A expressão analítica com os parâmetros estimados a partir da série de dados de falha permite o cálculo dos indicadores de qualidade do sistema. Supondo que as condições de solicitação permaneçam inalteradas, pode-se fazer previsões sobre o futuro do processo de falha. Estas previsões são mais precisas para um horizonte de tempo curto, uma vez que para horizontes longos crescem as chances das condições de utilização serem alteradas.

A aplicação de um modelo de confiabilidade com o objetivo de avaliar a segurança de funcionamento de sistemas é feita em seis passos:

1. Escolha do modelo ou conjunto de modelos a ser aplicado à base de dados de falha estudada;

2. Inferência dos parâmetros do modelo a partir dos dados de falha;

3. Calibragem do modelo com os parâmetros estimados, obtendo uma expressão ma-

temática que passa a representar o processo real;

4. Validação do modelo quanto a sua adequação à representação do processo de falha;

5. Derivação das medidas de segurança de funcionamento (confiabilidade, disponibilidade, mantenabilidade) a partir do modelo calibrado;

Alguns dos indicadores adicionais de segurança de funcionamento que podem ser obtidos a partir dos modelos são:

- número médio de falhas detectadas em qualquer instante de tempo,
- número médio de falhas que ocorrerão em um certo intervalo de tempo futuro,
- número de falhas remanescentes no sistema (número de falhas a serem detectadas em um tempo infinito de operação),
- intensidade de falha em qualquer instante de tempo,
- intervalo de tempo até ser detectado um certo número de falhas no sistema,
- probabilidade de operação livre de falhas durante um certo período de tempo (confiabilidade)

6. Projeção da segurança de funcionamento a partir do modelo calibrado.

2.7 Conclusão

Neste capítulo foram abordados aspectos gerais da avaliação de segurança de funcionamento dentro de uma perspectiva gerencial da evolução de um sistema. O processo de detecção de falhas/correção de defeitos aperfeiçoa o sistema dando origem a novas edições que substituem as anteriores ou novas versões que coexistem com as demais. A gerência da evolução de um sistema visa coordenar esse processo de modo que o conjunto de módulos que compõem o sistema formem um todo coerente a cada passo.

Um sistema evolui através da correção dos defeitos que se manifestaram causando falhas na operação. Para controlar esta evolução é necessário que as falhas e as correções

sejam convenientemente registradas. A partir destes registros pode-se representar o processo de falha a ser analisado como uma curva de falhas.

O acompanhamento do processo de falha ao longo do tempo permite reorientar a atividade de teste de modo a torná-la mais produtiva. Foi apresentado um método de acompanhamento do processo de falha do sistema ao longo do seu desenvolvimento e operação.

A representação do processo de falha exige uma base de tempo que sirva como referência. Neste capítulo foram discutidas as vantagens e inconvenientes da utilização do tempo de execução. O tempo medido pelo calendário foi apresentado como uma base alternativa, desde que a sua relação com o tempo de execução seja levada em conta. Como sugestão de representação desta relação foi discutido o conceito de esforço de teste.

A análise da curva de falhas se faz qualitativamente pelo estudo dos tipos de falha e quantitativamente por modelagem, ou seja, pela representação analítica da curva. Os modelos de confiabilidade desenvolvidos para a concepção de sistemas partem da hipótese de que a remoção dos defeitos melhora a condição do sistema. Por este motivo estes modelos são conhecidos como modelos de crescimento de confiabilidade.

No capítulo 3 são apresentados alguns dos modelos de crescimento de confiabilidade existentes e proposta uma forma de aplicação que leve em conta as características do processo de falha. São apresentados e discutidos alguns critérios de validação dos resultados da modelagem.

Capítulo 3

Modelagem da Confiabilidade de Concepção

3.1 Introdução

A avaliação da segurança de funcionamento de concepção é feita através da modelagem da curva representativa do processo de falha no tempo. Para isto são empregados os modelos conhecidos como modelos de crescimento de confiabilidade, desenvolvidos originalmente para a replicação de processos de falha software.

Como foi argumentado no Capítulo 1, a evolução dos sistemas computacionais levou ao desenvolvimento de métodos de confiabilidade adequados às características das falhas de concepção. Surgiram, a partir de então, dezenas de modelos de confiabilidade capazes de representar diferentes comportamentos de um processo de falha. Alguns trabalhos que fazem levantamento dos modelos de confiabilidade de software são [DAL 86, GOE 83b, LIT 80, YAM 83b].

Um modelo de confiabilidade de software, e, portanto, de concepção, é um modelo analítico que dá a forma geral do processo aleatório que descreve o processo de falha, através da caracterização da confiabilidade ou de uma grandeza a ela associada, em função do número de falhas detectadas ou em função do tempo [MUS 84]. Os parâmetros dessa função são dependentes das políticas de desenvolvimento ou de reparação das falhas detectadas ou das características intrínsecas do produto estudado. A partir da fórmula analítica do modelo e de seus parâmetros podem ser feitas previsões sobre o comportamento futuro do processo de falha e calculadas outras grandezas de interesse relacionadas

a este processo.

Este capítulo se dedica à análise de dois importantes aspectos da avaliação de sistemas: a escolha dos modelos a serem utilizados e a validação dos resultados encontrados.

O primeiro problema na avaliação de um sistema é escolher o modelo a ser utilizado. Este problema se reveste de maior importância quando se considera o grande número de modelos existentes.

Há dois aspectos importantes do problema de escolha de modelos:

- dada a grande quantidade de modelos existente quais implementar como ferramentas de avaliação dentro de um Programa de Segurança de Funcionamento.
- dado um conjunto de modelos implementados, qual ou quais aplicar a uma certa base de dados estudada.

A escolha do sub-conjunto de modelos a implementar deve ser feita de forma a cobrir os possíveis comportamentos de falha, evitando redundâncias e optando sempre pela simplicidade. Com o fim de orientar esta escolha, nas seções 3.2 e 3.3 são apresentados alguns dos modelos mais utilizados de confiabilidade de software e organizados segundo suas principais hipóteses e características.

Dado um certo conjunto de modelos disponíveis o mais adequado à modelagem de um dado processo de falha é aquele cujas hipóteses são compatíveis com as características reais do processo. É necessário conhecer as hipóteses em que se baseiam os modelos e analisar a natureza do processo de falha. Para análise do processo de falha é proposta uma técnica baseada nos testes de tendência, que é apresentada na seção 3.4.

Feita a modelagem de um processo de falha através de dois ou mais modelos, as técnicas de validação permitem comparar os resultados e determinar qual o modelo que mais fielmente representa o processo. Como será visto na seção 3.4 existem vários critérios, alguns conflitantes entre si, e a escolha deve ser feita a partir do objetivo da avaliação.

3.2 Classificação dos Modelos de Confiabilidade

Nos últimos dez anos foram propostas várias formas de diferenciação dos modelos de confiabilidade de concepção desenvolvidos. Cada autor adota um ponto de vista diferente para fazer sua classificação, porém algumas delas guardam estreita relação entre si. Em ordem cronológica, pode-se citar:

1. Schick e Wolverton [SCH 78] diferenciam duas abordagens na modelagem da confiabilidade de software, baseadas respectivamente no domínio dos dados de entrada do sistema e no domínio do tempo. Os modelos no domínio do tempo são divididos em modelos a distribuições discretas e modelos a distribuições contínuas dos instantes de detecção de falhas .

2. Ramamoorthy [RAM 82] apresenta um esquema de classificação baseado, em um primeiro nível, na fase do ciclo de vida do sistema na qual os modelos são aplicáveis.

3. Musa e Okumoto [MUS 84] propõem uma classificação a partir da forma funcional da intensidade de falha dos modelos. Duas categorias gerais são diferenciadas: a categoria de modelos a falhas finitas e a de modelos a falhas infinitas. Os modelos são organizados em tipos segundo a distribuição da quantidade de falhas detectadas. Os modelos a falhas finitas são organizados em classes segundo a forma funcional da intensidade de falha no tempo. Os modelos a falhas infinitas são classificados em famílias, de acordo com a forma funcional da intensidade de falha em termos do valor esperado do número de falhas detectadas no tempo.

4. Goel [GOE 85] organiza os modelos a partir de um critério misto, segundo a natureza do processo de falha estudado (modelos a intervalos entre falhas e modelos a contagem de falhas) e segundo a forma de aplicação dos modelos (modelos de implantação de falhas e modelos baseados no domínio dos dados de entrada).

5. Miller [MIL 86] divide os modelos em duas categorias: modelos a taxas de falha determinísticas e dados de entrada estocasticamente distribuidos e modelos duplamente estocásticos.

6. Mellor [MEL 87] classifica os modelos de confiabilidade de acordo com as hipóteses sobre o mecanismo de falha e sua estrutura matemática em dois grandes blocos: os modelos estruturais, que permitem a combinação da confiabilidade dos módulos na obtenção da confiabilidade global do sistema, e os modelos a caixa-preta, que levam em conta somente o comportamento global de falha do sistema. Os modelos a caixa preta se dividem em modelos a intervalos entre falhas e modelos que interpretam falhas como a manifestação de defeitos do sistema, onde cada defeito dá aleatoriamente origem a uma falha com uma certa taxa.

A classificação de um conjunto qualquer de elementos deve ser feita com base em características prórias desses elementos que os diferenciem dos demais. As classificações baseadas na forma ou no domínio dos dados de entrada ou no modo de aplicação dos modelos não parecem ser as mais indicadas pois não se referem às características dos modelos em si. Além disso, um modelo pode, por exemplo, atuar sobre os dados de entrada em ambas as formas, o que torna algumas classificações ambíguas. Supondo independência entre os processos de falha dos módulos que compõem um sistema, a maioria dos modelos do tipo caixa-preta pode ser aplicada a cada módulo e depois compor a confiabilidade global; pode-se notar, então, que este não é um critério que diferencie efetivamente os modelos.

Neste trabalho será adotada uma forma de classificação baseada na que foi proposta por Musa e Okumoto, estendendo-a de forma a incluir outros modelos e outra categoria que não se enquadram na proposta original. Esta forma de classificação permite detalhar as diferenças e semelhanças dos modelos através de suas características próprias.

A classificação apresentada organiza os modelos de confiabilidade de software segundo as duas tabelas a seguir, de acordo com os seguintes atributos :

categoria: número de falhas detectadas finito ou infinito,

tipo: distribuição do número de falhas detectadas até o instante t,

classe (somente para os modelos da categoria de falhas finitas): forma da função intensidade de falha no tempo,

família (somente para os modelos da categoria de falhas infinitas com intensidade de falha tendendo a zero): forma da função intensidade de falha em termos do número de falhas detectadas.

espécie (somente para os modelos da categoria de falhas infinitas, intensidade de falha não tendendo a zero e processo transitório com falhas finitas) : forma da função intensidade de falha do processo transitório.

grupo (somente para os modelos da categoria de falhas infinitas, intensidade de falha não tendendo a zero e processo transitório com falhas infinitas) : forma da função intensidade de falha do processo transitório em termos da função valor médio do processo transitório.

A partir destes critérios é possível classificar alguns dos modelos mais utilizados na literatura e ampliar as tabelas mostradas em [MUS 87]. O tipo de modelos a processo de poisson não homogêneo será denotado nas tabelas a seguir pela sigla PPNH.

Sejam:
$h(t)$ a função intensidade do processo de falha
$H(t) = \int_0^t h(x)dx$ a função valor médio do número esperado de falhas detectadas.

Suponha-se que
$h(t) = h_T(t) + h_P(t) = h_T(t) + \alpha$
$H(t) = H_T(t) + H_P(t) = H_T(t) + \alpha t$

Pode-se entender um processo de falha assim representado como a soma de um processo de falha transitório (caracterizado por $H_T(t)$ e $h_T(t)$) com um processo permanente (caracterizado por $H_P(t)$ e $h_P(t)$).

Classe	Tipo	Tipo	Tipo
$h(t)$	PPNH	Binomial	Outros
Exponencial	Musa [MUS 75] Moranda [MOR 75] Schneidewind [SCH 75] Goel-Okumoto [GOE 79b]	Jelinski-Moranda [JEL 72] Shooman [SHO 72]	Keiller-Littlewood [KEI 83]
Hiperexponencial	Ohba [OHB84a,b]		
Racional			Musa[MUS 79]
Weibull	Fukushima-Kishida [FUK 86]	Schick-Wolverton [SCH 73] Wagoner [WAG 73]	
Pareto		Littlewood [LIT 81a]	
Gama	Yamada-Ohba-Osaki [YAM 83a]		
S	Ohba [OHB 84a,b]		

Tabela 3.1 - Modelos da Categoria de Intensidade de Falha Nula e Falhas Finitas.

Família	Tipo	Tipo
$h(H(t))$	PPNH	Outros
Exponencial	Musa-Okumoto [MUS 84]	Moranda [MOR 75]
Potência	Crow [CRO 74]	
Linear Inversa		Littlewood-Verral [LIT 73]
Polinomial Inversa de Segunda Ordem		Littlewood-Verral [LIT 73]

Tabela 3.2 - Modelos da Categoria de Intensidade de Falha Nula e Falhas Infinitas .

Espécie	Tipo
$h_T(t)$	PPNH
KL	Kanoun-Laprie [KAN 85]

Tabela 3.3 - Modelos da Categoria de Intensidade de Falha Não Nula e Falhas Finitas no Processo Transitório.

Grupo	Tipo
$h_T(H_T(t))$	PPNH
Potência	Finkelstein [FIN 79]

Tabela 3.4 - Modelos da Categoria de Intensidade de Falha Não Nula e Falhas Infinitas no Processo Transitório.

A seguir serão analisadas algumas das características dos modelos a fim de compará-los.

3.2.1 Comportamento da Intensidade de Falha no Infinito

Um processo de falha pode ter dois comportamentos distintos após um longo tempo de operação do sistema. Uma primeira hipótese é de que todos os defeitos são removíveis, de modo que a intensidade de falha tende a se anular quando $t \rightarrow \infty$. A outra hipótese é que alguns defeitos permanecem no sistema devido à impossibilidade de caracterizá-los ou porque a taxa de falha causada por eles é tão baixa que não justifica o esforço de depuração. Os defeitos residuais levariam a uma intensidade de falha constante em $t \rightarrow \infty$ dada por a. Não existem evidências da superioridade de uma destas hipóteses sobre a outra.

As duas primeiras tabelas tratam de modelos que assumem a hipótese de que o processo de falha se interrompe em um tempo infinito. Estes modelos representam processos puramente transitórios. As outras duas tabelas foram acrescentadas à forma de classificação proposta por Musa a fim de considerar também os modelos que assumem que um processo de falha evolui até se tornar estacionário, com uma intensidade de falha tolerável. Os processos descritos por estes modelos são compostos por um processo transitório cuja intensidade de falha se anula quando o tempo tende a infinito e por um processo permanente com intensidade de falha constante atuando durante todo o tempo de operação. A categoria de modelos com intensidade de falha nula pode ser considerada como um caso particular da outra quando $\alpha = 0$.

3.2.2 Tendência do Número de Falhas Detectadas no Infinito

Dentro destas duas categorias distinguem-se os modelos a falhas finitas e os a falhas infinitas no processo transitório. Os modelos da categoria de falhas finitas têm um número de falhas detectadas em um tempo infinito que é limitado. Para os modelos da categoria de falhas infinitas o número de falhas em um tempo infinito é ilimitado.

As hipóteses básicas implícitas nos modelos de cada tabela são resumidas na Tabela 3.5. Observa-se que dentro de cada quadro das tabelas 3.1 a 3.4 as hipóteses dos modelos são idênticas. Isto significa que os modelos dentro de um certo quadro podem ser considerados equivalentes em termos das características do processo que descrevem.

Tabela	$h(t)$	$h_T(t)$	$h_P(t)$	$H(t)$	$H_T(t)$	$H_P(t) = \alpha t$
3.1	0	0	0	finito	finito	0
3.2	0	0	0	∞	∞	0
3.3	$\neq 0$	0	$\neq 0$	∞	finito	∞
3.4	$\neq 0$	0	$\neq 0$	∞	∞	∞

Tabela 3.5 - Hipóteses básicas dos modelos referentes ao limite das funções usadas para sua classificação nas Tabelas de 3.1 a 3.4.

3.2.3 Análise dos Modelos Segundo a Hipótese de Crescimento de Confiabilidade

Além de hipóteses sobre o comportamento do processo de falha no limite quando $t \rightarrow \infty$, os modelos trazem implícita uma hipótese sobre o processo transitório.

A análise da função intensidade de falha no período transitório $h(t)$ permite identificar se o modelo é capaz de representar um processo em que há crescimento de confiabilidade, decrescimento, estacionariedade, ou mesmo uma composição destes padrões.

Um processo de falha descrito por uma função intensidade de falha $h(t)$ decrescente tem sempre crescimento de confiabilidade. De modo inverso, se $h(t)$ é crescente o processo de falha representado tem decrescimento de confiabilidade. Uma função $h(t)$ constante no período transitório descreve um processo estacionário.

Alguns modelos têm uma função $h(t)$ que cresce (decresce) até um instante t e decresce (cresce) dali em diante. Estes modelos descrevem um processo que passa por uma fase inicial de decrescimento (crescimento) de confiabilidade, antes de assumir um padrão de crescimento (decrescimento) de confiabilidade. O instante t é chamado de ponto de inflexão da função $h(t)$. As tabelas 3.6 a 3.8 dão os tipos de comportamento do processo de falha que podem ser descritos pelos modelos. Nota-se que:

1. a grande maioria dos modelos de confiabilidade descreve processos onde há sempre crescimento de confiabilidade;

2. os modelos Potência e Weibull permitem descrever processos com crescimento ou estacionariedade de confiabilidade ou um processo composto de crescimento seguido de decrescimento de confiabilidade, dependendo dos valores dos parâmetros.

3. somente os modelos Gama e S permitem descrever um processo em que há decrescimento inicial de confiabilidade seguido de crescimento.

Tendência Classe	Crescimento	Estacionariedade	Decrescimento	Crescimento/ Decrescimento
Exponencial	Sim	$\beta_0 \to \infty$ e $\beta_1 \to 0$	Não	Não
Hiperexponencial	Sim	$\beta_{0i} \to \infty$ e $\beta_{1i} \to 0$	Não	Não
Weibull	$0 < \beta_2 < 1$	$\beta_2 = 1$	$\beta_2 > 1$ e $t^* \to 0$	$t^* = \frac{\beta_2 - 1}{\beta_1 - \beta - 2}$ para $\beta_2 > 1$
Pareto	Sim	Não	Não	Não
Gama	Não	Não	Não	$t^* = \frac{1}{\beta_1}$
S	$\beta_2 < 1$	$\beta_1 \to 0$	Não	$t^* = ln\frac{\beta_2}{\beta_1}$ para $\beta_2 < 1$

Tabela 3.6 - Comportamentos do processo de falha representados pelos modelos da Tabela 3.1.

Tendência Família	Crescimento	Estacionariedade	Decrescimento	Crescimento/ Decrescimento
Exponencial	Sim	$\beta_0 \to \infty$ e $\beta_1 \to 0$	Não	Não
Linear Inversa	Sim	Não	Não	Não
Polinomial	Sim	$\beta_1 = 0$	Não	Não
Inversa				
Potência	$\beta_2 < 1$	$\beta_2 = 1$	$\beta_2 > 1$	Não

Tabela 3.7 - Comportamentos do processo de falha representados pelos modelos da Tabela 3.2.

Tendência Espécie/Grupo	Crescimento	Estacionariedade	Decrescimento	Crescimento/ Decrescimento
KL	Sim	Não	Não	Não
Exponencial	Sim	$\beta_0 \to \infty$ e $\beta_1 \to 0$	Não	Não
Potência	$\beta_2 < 1$	$\beta_2 = 1$	$\beta_2 > 1$	Não

Tabela 3.8 - Comportamentos do processo de falha representados pelos modelos das Tabelas 3.3 e 3.4.

3.2.4 Análise dos Modelos Segundo o Número de Parâmetros

A complexidade de um modelo pode ser medida pelo número de seus parâmetros. Quanto mais parâmetros mais difíceis são a implementação do modelo e a convergência dos algoritmos de otimização usados na estimação [MUS 87].

Número de Parâmetros Tabelas	2	3	4
3.1	Exponencial Hiperexponencial Gama	Weibull Pareto S	
3.2	Exponencial Linear Inversa Polinomial Inversa Potência		
3.3		LK	
3.4			Potência

Tabela 3.9 - Número de Parâmetros dos Modelos.

A prática tem demonstrado que os modelos mais utilizados são também os mais simples. Por esta razão os modelos apresentados nesta seção têm entre dois e quatro parâmetros. Nos levantamentos mais completos de modelos anteriormente citados, no entanto, pode-se encontrar modelos com cinco parâmetros.

A tabela 3.9 dá o número de parâmetros de cada modelo das tabelas 3.1 a 3.4. Note-se que os modelos a Processo Permanente têm naturalmente mais parâmetros que os modelos que descrevem somente um Processo Transitório.

3.2.5 Tipos Processo Poissoniano Não-Homogêneo e Binomial

Um grande número de modelos pertence aos tipos Poissoniano Não-Homogêneo (PPNH) e Binomial. Esse grupo reune os modelos mais utilizados na prática e por isso merece uma atenção especial. Tanto os modelos Poissonianos Não-Homogêneos como os Binomiais podem ser descritos como um processo de Markov.

Seja um processo de falha de concepção tal como o descrito na Figura 1.4 e $N(t)$ a variável aleatória que representa o número de falhas detectadas até o instante t em um sistema. A propriedade de Markov para o processo $N(t)$ pode ser resumida na forma: para um intervalo Δt, o número de falhas detectadas até $t + \Delta t$ depende somente do estado atual do processo no instante t, ou seja, a probabilidade condicional:

$$P[N(t + \Delta t) = j / N(t) = i] \tag{3.1}$$

descreve o comportamento futuro do processo $N(t)$. A probabilidade condicional, que também é chamada de função de transição do processo, depende de t e Δt, além de i e

de j. Chamando a função de transição de $P_{ij}(t, \Delta t)$, tem-se:

$$P[N(t+\Delta t) = j] = \sum_i P_{ij}(t, \Delta t) P(N(t)) = i] \tag{3.2}$$

Segundo as hipóteses dos modelos, Musa e Okumoto [MUS 83] mostram que eles podem ser divididos nos seguintes tipos, de acordo com a distribuição do número de falhas detectadas até o instante t :

- Modelos Poissonianos Não-Homogêneos
- Modelos Binomiais

As tabelas 3.1 a 3.4 apresentam os modelos separados em Poissonianos Não-Homogêneos, Binomiais e outros em diferentes colunas verticais.

a) Modelos Poissonianos Não-Homogêneos

Sejam :

$N(t)$ o número de falhas detectadas até o instante t,

t_i o instante de ocorrência da i-ésima falha.

Considere-se as seguintes hipóteses :

1. $P[N(0) = 0] = 1$
2. $[N(t+\Delta t) - N(t)]$ é independente dos valores anteriores de $H(t)$ (incrementos independentes)
3. A probabilidade de que uma falha ocorra em $(t, t+\Delta t]$ é $h(t)\Delta t + o(\Delta t)$ onde $h(t)$ é a intensidade de falha do processo, sendo que

$$\lim_{\Delta t \to 0} = \frac{o(\Delta t)}{\Delta t} = 0$$

4. A probabilidade de que mais de uma falha ocorra em $(t, t+\Delta t]$ é $o(\Delta t)$.

Das hipóteses 2,3 e 4 encontra-se que a função de transição do processo $N(t)$ é:

$$P_{ij}(t, \Delta t) = \begin{cases} 1 - h(t)\Delta t + o(\Delta t) & , j = i \\ h(t)\Delta t & , j = i+1 \\ o(\Delta t) & , \textit{nos demais casos} \end{cases} \tag{3.3}$$

Seja $P_m(t)$ a probabilidade de que $N(t) = m$. Substituindo [3.3] em [3.2] e arrumando tem-se:

$$\frac{P_m(t+\Delta t) - P_m(t)}{\Delta t} = -h(t)P_m(t) + h(t)P_{m-1}(t) + \frac{o(\Delta t)}{\Delta t} \tag{3.4}$$

Resolvendo de forma recursiva, sabendo que $P_{m-1}(0) = 0$ chega-se a:

$$P_m(t) = \frac{[H(t)]^m}{m!} exp[-H(t)], onde \tag{3.5}$$

$$H(t) = \int_0^t h(x)dx \tag{3.6}$$

ou seja, processo de falha é poissoniano não homogêneo e com média e variância dadas por $H(t)$, que é chamada de função valor médio do processo. A função intensidade de falha $h(t)$ é a taxa de variação da função valor médio, dada pela derivada de $H(t)$, e é um valor instantâneo. O processo de falha $N(t)$ pode ser então particularizado especificando-se a função intensidade de falha $h(t)$.

O total de falhas nos modelos Poissonianos Não-Homogêneos é uma variável aleatória com média u_0. Esta é uma hipótese bastante razoável uma vez que o total de defeitos num sistema é uma função extremamente complexa de vários fatores independentes, como seu tamanho, complexidade, e desempenho de quem desenvolveu o sistema. O parâmetro uo nesses modelos não está restrito a um valor inteiro embora, é claro, suas realizações tenham somente valores inteiros. Na prática estima-se o valor deste parâmetro como sendo real e aproxima-se para o inteiro mais próximo.

b) Modelos Binomiais

Os modelos binomiais são desenvolvidos com base nas seguintes hipóteses :

1. Sempre que ocorre uma falha de concepção o defeito que a causou é removido instantaneamente.

2. Há u_0 defeitos inerentes ao sistema.

3. Cada falha, causada por um defeito, ocorre independente e aleatoriamente no tempo de acordo com a taxa de falha por defeito $z_a(t)$. Dessa forma as taxas de falha para todos os defeitos são iguais.

A hipótese 1 implica que os modelos binomiais não permitem considerações sobre defeitos não localizados, defeitos introduzidos durante a correção de outros defeitos ou falhas não caracterizadas; em outras palavras, esses modelos supõem caracterização e correção perfeitas. Para os modelos binomiais o número de defeitos removidos até t é

igual ao número de falhas detectadas até esse instante. O número de falhas detectadas em um tempo infinito é igual ao número de defeitos inerentes u_0. Na verdade u_0 varia com mudanças no ambiente de operação, com a tipo de operação do sistema e com a eficácia das atividades de manutenção, porém os modelos binomiais consideram o ambiente de operação fixo e um comportamento médio dos outros fatores de modo que possam ser considerados constantes. Os modelos binomiais não admitem também a existência de dependência entre falhas.

Seja T_a a variável aleatória que representa o tempo até a falha causada por um defeito a, onde $a = 1, 2, ...u_0$. Então as variáveis T_a são independentes e identicamente distribuidas como $F_a(t)$ para todos os defeitos remanescentes. A função distribuição cumulativa $F_a(t)$ para os modelos binomiais pode ser obtida a partir da taxa de falha $z_a(t)$ segundo :

$$F_a(t) = 1 - exp[-\int_0^t z_a(x)dx] \tag{3.7}$$

Os modelos binomiais têm função taxa de falha descontínua cujas descontinuidades ocorrem a cada detecção e tem valor $z_a(t)$.

A taxa de falha dentro do i-ésimo intervalo entre falhas T_i' condicionada ao instante de detecção da última falha $T_{i-1} = t_{i-1}$ é dada por

$$z(t_i'/t_{i-1}) = (u_0 - i + 1)z_a(t_{i-1} + t_i') \tag{3.8}$$

onde $t_i' = \sum_{j=0}^{i} t_j$.

Dessa forma para os modelos binomiais a taxa de falha $z_a(t)$ caracteriza a distribuição de T_a. Os modelos dessa categoria se diferenciam através da escolha da função $z_a(t)$. A média da distribuição binomial é dada por

$$H(t) = u_0 F_a(t) \tag{3.9}$$

e a intensidade de falha é então :

$$h(t) = u_0 f_a(t) \tag{3.10}$$

Os modelos binomiais tomam essa denominação pelo fato de que o número de falhas

detectadas até um instante t é binomialmente distribuido da forma

$$P[H(t) = m] = \begin{pmatrix} u_0 \\ m \end{pmatrix} [F_a(t)]^m [1 - F_a(t)]^{u_0 - m}, m = 0, 1..., u_0 \qquad (3.11)$$

Os modelos Poissonianos Não-Homogêneos a falhas finitas podem ser deduzidos a partir dos modelos binomiais modificando-se a hipótese 2 da forma :
2.a) O total de defeitos remanescentes no sistema em $t = 0$ é uma variável aleatória de Poisson com média $w0$.

Pode-se fazer as seguintes considerações sobre os modelos PPNH e Binomiais:

1. Tanto os modelos Binomiais quanto os PPNH assumem que as taxas de falhas são iguais para todos os defeitos latentes.

2. Os modelos PPNH incorporam os casos em que o número inicial de defeitos no sistema não é conhecido com certeza. O número esperado de falhas inicial w_0 é conhecido mas não há um número fixo de defeitos inicial como nos modelos Binomiais. Como a quantidade de defeitos em um sistema depende de muitos fatores imprevisíveis (além dos fatores conhecidos como complexidade e ambiente de desenvolvimento) é bastante razoável modelar o seu número inicial como uma variável aleatória.

3. Os modelos PPNH representam também, ainda que de maneira aproximada, uma depuração imperfeita e a introdução de novas falhas durante a reparação.

4. Os modelos Binomiais têm taxa de falha com descontinuidades a cada ocorrência. Embora as correções efetivamente modifiquem a taxa de falha, elas ocorrem em um instante de tempo variável depois das falhas e que pode muito bem ser modelado como aleatório.

5. Os procedimentos de estimação dos parâmetros são os mesmos para os modelos Binomiais e PPNH, o que sugere que as curvas descontínuas das taxas de falha dos modelos Binomiais podem ser aproximadas pelas curvas contínuas dos modelos PPNH.

A partir destas considerações é possível uma conclusão importante: um modelo PPNH de uma certa classe representa também um modelo Binomial dessa mesma classe e vice-versa.

3.2.6 Considerações Gerais sobre os Modelos

Ao adotar um Programa de Segurança de Funcionamento o fiabilista se depara com o problema de quais modelos implementar, dada a grande quantidade de modelos existente.

Das análises apresentadas nesta seção é possível resumir algumas conclusões úteis na orientação desta escolha, permitindo que ela seja o mais abrangente possível e evitando redundâncias:

1. Dentro de uma mesma cela das tabelas 3.1 a 3.4 os modelos são equivalentes quanto às hipóteses sobre o processo de falha;
2. Os modelos Binomiais e PPNH são equivalentes, com vantagens para os primeiros;
3. Modelos a processo permanente podem ser obtidos a partir de modelos a processo transitório. A implementação de um ou outro tipo depende da hipótese feita sobre o processo de falha quando o tempo tende a infinito;
4. Os modelos mais simples são mais fáceis de implementar e os que têm menor probabilidade de levar os métodos de estimação dos parâmetros a divergir. É interessante, pois, dar preferência aos modelos a dois ou três parâmetros.
5. É interessante implementar um subconjunto de modelos que seja capaz de representar as tendências que pode apresentar um processo de falha: crescimento, decrescimento e estacionariedade de confiabilidade.

Estas conclusões permitem guiar a escolha dos modelos a serem implementados. Existem muitas combinações de modelos que atendem a estas orientações, no entanto, o número de modelos a implementar em qualquer delas é pequeno e constitui um bom ferramental de trabalho.

3.3 Apresentação dos Modelos

3.3.1 Modelos de Processo Transitório e Falhas Finitas

a) Modelos Poissonianos Não-Homogêneos

a.1) Classe Exponencial

Para esses defeitos a taxa de falha por defeito é constante e igual a β_1 .

$$H(t) = \beta_0[1 - exp(-\beta_1 t)] \tag{3.12}$$

$$h(t) = \beta_0\beta_1 exp(-\beta_1 t) \tag{3.13}$$

Assume-se a hipótese de que a intensidade de falha é exponencialmente decrescente no tempo, com valor inicial $h_0 = \beta_0\beta_1$. A função intensidade de falha $h(t)$ é sempre decrescente. Estes modelos representam um comportamento de crescimento de confiabilidade. Quando aplicado a um processo estacionário $\beta_0 \rightarrow \infty$ e $\beta_1 \rightarrow 0$.

a.2) Classe Hiperexponencial

Esta classe de modelos foi incorporado à tabela original de Musa.

Um sistema é dividido em n conjuntos de módulos com características similares, ou clusters. Para cada cluster de módulos a intensidade de falha é representada por uma função exponencial, de forma que a curva resultante é hiperexponencial. Para cada cluster são válidas as mesmas hipóteses da classe exponencial. A hipótese adicional é de que os clusters têm processos de falha diferenciados.

$$H(t) = \sum_{i=1}^{n} \beta_{0i}[1 - exp(-\beta_{1i} t)] \tag{3.14}$$

$$h(t) = \sum_{i=1}^{n} \beta_{0i}\beta_{1i} exp(-\beta_{1i} t) \tag{3.15}$$

A função intensidade de falha é decrescente para cada cluster de módulos. Isso corresponde à hipótese de crescimento de confiabilidade em cada cluster e no sistema como um todo. É válida a mesma observação sobre a estacionariedade feita para a classe Exponencial.

a.3) Classe Weibull

Os modelos desta classe são caracterizados pelas seguintes equações:

$$H(t) = \beta_0[1 - exp(-\beta_1 t^{\beta_2}] \tag{3.16}$$

$$h(t) = \beta_0\beta_1\beta_2 t^{\beta_2 - 1} exp(-\beta_1 t^{\beta_2}) \tag{3.17}$$

Dependendo da relação entre os parâmetros o modelo pode representar processos de falha diferentes. Para $0 < \beta_2 < 1$ a função $h(t)$ é decrescente e descreve um crescimento de confiabilidade. Para $\beta_2 = 1$ a função $h(t)$ é constante, o que caracteriza um processo estacionário. Quando $\beta_2 > 1$ o modelo apresenta um ponto de inflexão em $t^* = [(\beta_2 - 1)/\beta_1\beta_2]^{\frac{1}{\beta_2}}$: a função $h(t)$ é decrescente até t^* e crescente a partir dali. Para $\beta 2 > 1$ estes modelos representam um processo com crescimento de confiabilidade até o ponto de inflexão, seguido de decrescimento de confiabilidade. Ajustando os parâmetros de forma que $t^* \rightarrow 0$ o modelo pode descrever um processo de decrescimento de confiabilidade.

a.4) Classe Gama

A hipótese, neste modelo, é de que o tempo até um defeito causar uma falha segue uma distribuição gama com parâmetro de forma igual a dois.

$$H(t) = \beta_0[1 - (1 + \beta_1 t)exp(-\beta_1 t)] \tag{3.18}$$

$$h(t) = \beta_0\beta_1^2 texp(-\beta_1 t) \tag{3.19}$$

A função intensidade de falha $h(t)$ cresce de $t = 0$ até $t = 1/\beta_1$ e então decresce gradualmente, tendendo a zero. Este modelo descreve um processo de falha em que há decrescimento de confiabilidade até um certo instante $1/\beta_1$ e crescimento de confiabilidade dali em diante. O ponto de inflexão é fixado pelo parâmetro β_1, que rege também a variação da intensidade de falha no tempo. Uma curva de falha desta forma, côncava até t e convexa dali em diante também é conhecida como *curva em S*.

a.5) Classe S

A classe do modelo de inflexão em S sugerido por Ohba [OHB 84] está sendo acrescentada à tabela original de Musa. Este modelo incorpora os efeitos de um esforço de teste linearmente crescente ou decrescente ao longo do período de observação ou a dependência entre falhas latentes no sistema.

A função valor médio do número de falhas observado dada por este modelo é dada por :

$$H(t) = \beta_0 \frac{1 - exp(-\beta_1 t)}{1 + \beta_2 exp(-\beta_1 t)} \tag{3.20}$$

onde $\beta_2 = (1 - r)/r$, sendo que $r > 0$ exprime a declividade da reta que representa o esforço de teste ou a dependência entre as falhas. Quando $r = 1$ o modelo é equivalente ao modelo exponencial, significando que o esforço de teste é constante ou que todos os defeitos podem se manifestar e ser removidos logo no início dos testes. O modelo representa uma curva logística para valores de r próximos de zero, significando que poucas falhas podem se manifestar no início dos testes e que as falhas dependentes se tornam rapidamente detectáveis à medida que o tempo de teste avança. A função intensidade de falha para esse modelo é:

$$h(t) = \frac{\beta_0 \beta_1 exp(-\beta_1 t)}{[1 + \beta_2 exp(-\beta_1 t)]^2} \tag{3.21}$$

O ponto de inflexão deste modelo é localizado em $t^* = \frac{ln\beta_2}{\beta_1}$ quando $\beta_2 > 1$ e pode ser controlado através deste parâmetro. Para $t^* = 1/\beta_1$ o modelo é equivalente ao modelo da classe Gama. O modelo em S descreve decrescimento seguido de crescimento de confiabilidade com ponto de inflexão em $t = ln\beta_2/\beta_1$.

b) Modelos Binomiais

Os modelos dessa classe se dividem nas seguintes famílias, de acordo com a taxa de falha por defeito $z_a(t)$:

b.1)Classe Exponencial

A hipótese é a mesma dos modelos Poissonianos Não-Homogêneos desta classe: a taxa de falha por defeito é constante .

$$H(t) = \beta_0 \beta_1 \int_0^t exp(-\beta_1 t) \tag{3.22}$$

$$h(t) = \beta_0 \beta_1 exp(-\beta_1 t) \tag{3.23}$$

No modelo de Schooman a taxa de falha $z(t')$ é caracterizada pela densidade de defeitos inerente ao sistema w_1 , a proporção de instruções simples processadas α_1 , uma constante α_2 e os defeitos corrigidos por instrução por unidade de tempo $\alpha_3(t)$ que re-

presenta a proporção de defeitos que geraram falhas :

$$z(t') = [w_1 - \int_0^t \alpha_3(x)dx]\alpha_1\alpha_2 \tag{3.24}$$

Para estes modelos a função $h(t)$ é sempre decrescente, e representa um processo com crescimento de confiabilidade.

b.2) Classe de Weibull

A taxa de falha é da forma $z_a(t) = \phi\gamma t, \gamma > 0$. A taxa de falha dentro dos intervalos entre falhas é

$$z(t_i'/t_{i-1}) = (u_0 - i + 1)\phi\gamma(t_{i-1} + t_i')^{\gamma-1} \tag{3.25}$$

que tem descontinuidades $z_a(t)$ a cada falha. A taxa de falha por defeito pode ser decrescente $(0 < \gamma < 1)$, constante $(\gamma = 1)$ e crescente $(\gamma > 1)$.

O modelo de Schick-Wolverton [SCH 73] pode ser interpretado como tendo uma taxa de falha por defeito $z_a(t) = 2\phi t$ que é um caso especial da função de Weibull com $\gamma = 2$. Para esse modelo

$$H(t) = \beta_0[1 - exp(-\beta_1 t^{\beta_2})] \tag{3.26}$$

$$h(t) = \beta_0\beta_1\beta_2 t^{\beta_2-1} exp(-\beta_1 t^{\beta_2}) \tag{3.27}$$

O comportamento de $h(t)$ é semelhante ao discutido para os modelos Poissonianos Não-Homogêneos desta mesma classe. Com $\beta_2 > 1$ o modelo representa crescimento de confiabilidade, estacionariedade com $\beta_2 = 1$. Para $t = \beta_2 > 1$ o modelo descreve um processo em que há crescimento seguido de decrescimento de confiabilidade. Para $t^* \to 0$ o modelo consegue também representar decrescimento de confiabilidade.

b.3) Classe de Pareto

O modelo de Littlewood [LIT 81a] leva em conta a possibilidade de alguns defeitos serem mais prováveis de ocorrer que outros e dessa maneira tendem a ser reparados anteriormente. Neste caso a redução na taxa de falha é maior para as falhas mais prováveis. O modelo considera que cada defeito causa uma falha em um instante que é exponencialmente distribuido e independente dos demais.

$$H(t) = \beta_0[1 - \beta_1^{\beta_2}(t + \beta 2)^{\beta_2}] \tag{3.28}$$

$$h(t) = \beta_0\beta_2(t + \beta_1)^{-\beta_2 - 1} \tag{3.29}$$

A função $h(t)$ não tem ponto de inflexão e é sempre decrescente. Os modelos desta classe descrevem processos com crescimento de confiabilidade.

c) Outros

Alguns modelos que não se enquadram nos tipos Poisson Não-Homogêneo nem Binomial estão descritos a seguir:

c.1) Classe Exponencial

Littlewood e Verrall [LIT 73] propuseram um modelo bastante versátil que assume que a taxa de falha de um sistema é uma variável aleatória com distribuição gama com parâmetro de escala dado por $\xi(i)$ e parâmetro de forma dado por α. Dependendo dos parâmetros pode-se imaginar modelos que se enquadram de maneiras diferentes na classificação de Musa.

Keiller e outros [KEI 83] investigaram um modelo em que a forma do crescimento de confiabilidade é dada pelo parâmetro de forma da distribuição gama. O autor sugeriu uma função linear para o parâmetro de forma do tipo :

$$\alpha(i) = \beta_2 + \beta_3 i \tag{3.30}$$

Para este tipo de modelo

$$H(t) = \frac{1}{\beta_3} exp[\frac{\beta_3}{\beta_1}t + ln\beta_2] - \frac{\beta_2}{\beta_3} \tag{3.31}$$

$$h(t) = \frac{1}{\beta_1} exp[\frac{\beta_3}{\beta_1}t + ln\beta_2] \tag{3.32}$$

que têm a mesma forma funcional dos modelos de falhas finitas do tipo exponencial. A função $h(t)$ é sempre decrescente, revelando que este tipo de modelo assume a hipótese de crescimento de confiabilidade.

c.2) Classe Racional

Musa [MUS 79a] sugeriu que a função de crescimento de confiabilidade $\xi(i)$ poderia ser inversamente relacionada ao número de falhas remanescentes, representando-a na forma de uma função racional

$$\xi(i) = \frac{\beta_0\beta_1}{\beta_2(\beta_0 - i)} \tag{3.33}$$

onde β_1 é o parâmetro da distribuição gama. Neste modelo a forma da função de crescimento de confiabilidade é dada pelo parâmetro de escala da função de distribuição gama.

Para este tipo de modelo

$$H(t) = \beta_0 - exp[ln\beta_0 - \frac{\beta_2 t}{\beta_0\beta_1}] \tag{3.34}$$

$$h(t) = \frac{\beta_0}{\beta_1\beta_2} exp[ln\beta_0 - \frac{\beta_2 t}{\beta_0\beta_1}] \tag{3.35}$$

Como $h(t)$ é sempre decrescente ela descreve um processo com crescimento de confiabilidade.

3.3.2 Modelos de Processo Transitório e a Falhas Infinitas

Para estes modelos $H(t)$ é ilimitada quando t tende a infinito. Os modelos desta categoria têm função intensidade de falha nula no infinito.

a) Modelos Poissonianos Não-Homogêneos

Para os modelos desta categoria o número de falhas detectadas num tempo infinito é ilimitado, porém a função intensidade de falha é nula.

a.1) Família Exponencial

O modelo a tempo de execução PPNH logarítmico de Musa e Okumoto [MUS 84] tem função intensidade de falha que decresce exponencialmente com o número de falhas detectadas esperado:

$$H(t) = \frac{1}{\beta_1} ln(\beta_0\beta_1 t + 1) \tag{3.36}$$

$$h(t) = \frac{\beta_0}{\beta_0\beta_1 t + 1} \tag{3.37}$$

$$h(H(t)) = \beta_0 exp[-\beta_1 H(t)] \tag{3.38}$$

O valor β_0 é a intensidade de falha inicial e β_1 o parâmetro de decrescimento da intensidade de falha a cada correção. A intensidade de falha é uma função linear inversa do tempo. A função $h(t)$ é sempre decrescente. Os modelos desta família estão baseados na hipótese de que o processo de falha apresenta sempre crescimento de confiabilidade.

a.2) Família de Potência

A função valor médio e a função intensidade de falha, que neste caso tem a mesma forma da taxa de falha de Weibull, são:

$$H(t) = \beta_1 t^{\beta_2} \tag{3.39}$$

$$h(t) = \beta_1\beta_2 t^{\beta_2 - 1} \tag{3.40}$$

$$h(H(t)) = \beta_0^{\frac{-1}{\beta_1}} \beta_1 H(t)^{1-\frac{1}{\beta_1}} \tag{3.41}$$

Para $\beta_2 = 0$ a função $h(t)$ é constante no tempo e representa um processo estacionário. Para $\beta_2 < 1$ a função $h(t)$ é decrescente e descreve um processo com crescimento de confiabilidade. A função $h(t)$ é crescente para $\beta_2 > 1$, e pode modelar um processo de decrescimento de confiabilidade.

b) Outros Modelos a Falhas Infinitas

Alguns modelos não se enquadram na classificação de PPNH ou de Binomiais:

b.1) Família Exponencial

No modelo proposto por Moranda [MOR 75] a taxa de falha decresce, a cada falha corrigida, de uma fração constante k da taxa de falha antes da correção. A taxa de falha segue uma progressão geométrica no tempo dada por

$$z(t_i') = z_0 k^{i-1} \tag{3.42}$$

onde z_0 é a taxa de falha inicial,$0 < k < 1$ e t_i' é o intervalo entre a ocorrência da $(i-1)$-ésima e a i-ésima falha. A vantagem desse modelo, segundo o autor, é de aliviar

a hipótese de efeitos iguais das falhas, embutida nos modelos em que a taxa de falha decresce de um degrau constante.

As equações da intensidade de falha, e de sua relação com a função valor médio são dadas respectivamente por

$$H(t) = \frac{1}{\beta_1} ln[(\beta_0 \beta_1 t exp\beta_1 t + 1] \tag{3.43}$$

$$h(t) = \frac{\beta_0 exp\beta_1}{[\beta_0 \beta_1 exp_{\beta_1}]t + 1} \tag{3.44}$$

$$h(H(t)) = \beta_0 \beta_1 exp\beta_1 exp(-\beta_1 H(t)) \tag{3.45}$$

Como a função $h(t)$ é sempre decrescente, pode-se notar que este tipo de modelo assume também que o processo de falha descrito apresente sempre crescimento de confiabilidade.

Modelo Geral de Littlewood-Verral

Seja $f(t_i')$ a função densidade de probabilidade de T_i'. A distribuição condicional exponencial da taxa de falha é

$$f(t_i'/z_i) = z_i exp(-z_i t_i') \tag{3.46}$$

A função distribuição de T_i' é

$$f(t_i') = \alpha[\frac{\xi(i)}{t_i' + \xi(i)}]\frac{1}{t_i' + \xi(i)} \tag{3.47}$$

que tem uma distribuição de Pareto. A taxa de falha do sistema é dada pela equação

$$z(t_i') = \frac{\alpha}{t_i' + \xi(i)} \tag{3.48}$$

Esta função decresce continuamente com t_i' e tem descontinuidades não necessariamente decrescentes nos instantes de ocorrência de falhas.

Dependendo da forma de $\xi(i)$ o modelo se enquadra em diferentes classificações. Algumas formas sem restrições sobre i e que podem, portanto, ser classificadas entre os

modelos a falhas infinitas foram sugeridas pelos autores do modelo. São elas as seguintes :

b.2) Linear Inversa

A função crescimento de confiabilidade tem a forma

$$\xi(i) = \beta_0 + \beta_1 i \tag{3.49}$$

Fazendo, sem perda de generalidade, $\alpha = 1$ a função valor médio, a função intensidade de falha e a relação entre elas é dada por

$$H(t) = \frac{1}{\beta_1}(-\beta_0 + \sqrt{\beta_0^2 + \beta_1 t}) \tag{3.50}$$

$$h(t) = \frac{1}{\sqrt{\beta_0^2 + \beta_1 t}} \tag{3.51}$$

$$h(H(t)) = \frac{1}{\beta_0 + \beta_1 H(t)} \tag{3.52}$$

Neste caso $h(t)$ é decrescente, representando um processo de falha em que há sempre crescimento de confiabilidade.

b.3) Polinomial Inversa de Segunda Ordem

A função crescimento de confiabilidade é da forma

$$\xi(i) = \beta_0 + \beta_1 i^2 \tag{3.53}$$

Para este modelo:

$$H(t) = 3\xi_0(\sqrt[3]{t + \sqrt{t^2 + \xi_1}} - \sqrt[3]{t - \sqrt{t^2 + \xi_1}}) \tag{3.54}$$

$$h(t) = \frac{1}{\beta_2 + \beta_1 \sqrt[3]{t + \sqrt{t^2 + \xi_1}} - \sqrt[3]{t - \sqrt{t^2 + \xi_1}}} \tag{3.55}$$

$$h(H(t)) = \frac{1}{\beta_2 - \beta_1 H(t)^2} \tag{3.56}$$

Com $\beta_1 = 0$ a função $h(t)$ é constante, indicando um processo estacionário. Com $\beta_1 \neq 0$ a função $h(t)$ é decrescente, representando um processo de crescimento de confiabilidade.

3.3.3 Modelos de Processo Permanente a Falhas Finitas

Esta nova categoria de modelos está sendo introduzida na classificação de Musa a fim de incorporar modelos compostos por um processo de falha transitório cuja intensidade de falha tende a zero e por um processo permanente com intensidade de falha constante.

Modelos com estas características podem ser obtidos através da adição de um processo de falha permanente aos processos de falha transitórios dos modelos descritos na seção 3.3.1. Este artifício se faz através da estimação de mais um parâmetro α.

a) Modelos Poissonianos Não-Homogêneos

a.1) Espécie KL

O modelo de Kanoun-Laprie [LAP 84b, LAP 91], chamado de Modelo Hiperexponencial pelos autores, é o único encontrado na literatura que foi originalmente desenvolvido para representar um processo de falha composto.

Este modelo tem uma função intensidade de falhas que é dada por um coeficiente do tipo :

$$h(t) = \frac{\omega z_1 e^{-z_1 t} + (1-\omega) z_2 e^{-z_2 t}}{\omega e^{-z_1 t} + (1-\omega) e^{z_2 t}} \tag{3.57}$$

Para $t = 0$ sabe-se que $h_0 = \omega z_1 + (1-\omega) z_2$ e quando o tempo tende a infinito o intervalo entre detecções de falha é exponencialmente distribuido com taxa igual a $\inf(z_1, z_2)$. Escrevendo a intensidade de falha como a soma de um processo transitório e de um permanente e supondo, sem perda de generalidade, que $z_2 > z_1$:

$$h(t) = z1 + \frac{(1-\omega)(z_2 - z_1) exp - (z_2 - z_1)t}{\omega + (1-\omega) exp(z_1 - z_2)t} \tag{3.58}$$

A função transitória não tem uma forma clássica e portanto será enquadrada como espécie KL. A função valor médio é dada por :

$$H(t) = -ln[\omega e^{-z_1 t}] + (1-\omega) e^{-z_2 t}] \tag{3.59}$$

de onde se ve que $H(\infty) = \infty$.

Este modelo é constituido de dois processos : um processo transitório no qual a função intensidade de falha tende a zero e o número total de falhas é finito

$$H_T(t) = -ln[\omega + (1-\omega)e^{-(z_2-z_1)t}] \tag{3.60}$$

$$lim_{t\to\infty} H_T(t) = -ln\omega \tag{3.61}$$

e um permanente, em que a função intensidade é constante:

$$H_P(t) = z_1 t \tag{3.62}$$

Este modelo também trabalha com a hipótese de crescimento de confiabilidade. Quando $\omega = 1$ ou $z_1 = z_2$ o modelo descreve somente o processo permanente, que é estacionário.

3.3.4 Modelos de Processo Permanente e Falhas Infinitas

Esta categoria de modelos também foi incluida na classificação de Musa. Não foram encontrados na literatura modelos que tenham sido desenvolvidos com as hipóteses desta classe.

Da mesma forma que na seção anterior, pode-se obter modelos desta categoria a partir de modelos de processos transitórios e falhas infinitas. Isto se faz adicionando uma intensidade de falha constante às expressões da intensidade de falha dos modelos apresentados na seção 3.3.2. Foram encontrados na literatura técnica dois autores que seguiram este procedimento.

a) Modelos Poissonianos Não-Homogêneos

a.1) Grupo Exponencial

Hartler [HAR 89] apresenta um modelo com somente um parâmetro, chamado de modelo a função intensidade truncada, cuja função valor médio do número de falhas esperado é :

$$H(t) = t/\beta_0 - ln(1 + t/\beta_0), \beta_0 > 0 \tag{3.63}$$

que também pode ser interpretada como a soma de um processo transitório com um processo permanente a taxa de falha constante. Este modelo foi imaginado para o hardware e representa um processo de falha com decrescimento de confiabilidade. Invertendo-se o sinal da parcela transitória ter-se-ia um modelo para crescimento de confiabilidade:

$$H(t) = t/\beta_0 + ln(1 + t/\beta_0), \beta_0 > 0 \tag{3.64}$$

A função intensidade de falha seria dada por:

$$h(t) = 1/\beta_0 + (1 + t/\beta_0)/\beta_0 \tag{3.65}$$

A função intensidade de falha do processo transitório em termos da função valor médio do processo transitório é dada pela relação:

$$h_T(H_T(t)) = -1/\beta_0 exp(H_T(t)) \tag{3.66}$$

que caracteriza uma relação exponencial.

Este grupo de modelos também assume que o processo descrito apresenta crescimento de confiabilidade, uma vez que $h(t)$ é sempre decrescente. Este modelo não foi incluido na Tabela 3.4 pois não foi testado na prática na forma referente a crescimento de confiabilidade.

a.2) Grupo de Potência

Finkelstein [FIN 79] adaptou o modelo de Duane [DUA 64] descrito por Crow [CRO 74]. O modelo foi modificado a fim de incorporar um processo de falha permanente de maneira que a intensidade de falha fosse finita para $t = 0$. As expressões para esse modelo são:

$$h(t) = \begin{cases} \beta_3 + \beta_0 & 0 < t < \tau \\ \beta_0 + \beta_1\beta_2 t^{\beta_2 - 1} & t > \tau \end{cases} \tag{3.67}$$

$$H(t) = \begin{cases} (\beta_3 + \beta_0)t + \beta_1 t^{\beta_2} & 0 < t < \tau \\ \beta_3\tau + \beta_0 t + \beta_1 t^{\beta_2} & t > \tau \end{cases} \tag{3.68}$$

A relação $h_T(H_T(t)), t > \tau$, é da mesma forma potencial apresentada para o modelo de Crow na seção 3.3.2.

Para $\beta_2 = 1$ o modelo representa um processo estacionário. Para $\beta_2 > 1$ a função $h(t)$ é crescente, e descreve um processo de falha com decrescimento de confiabilidade. Para $\beta_2 < 1 h(t)$ é decrescente e modela um processo de falha com crescimento de confiabilidade.

O trabalho de Finkelstein é um exemplo de como se pode adaptar modelos que tem intensidade de falha nula no infinito para representar o comportamento de sistemas cujos processos de falha tendem a um valor residual. Este mesmo raciocínio pode ser empregado a outros modelos das Tabelas 3.1 e 3.2.

O conjunto de modelos apresentado tem por trás hipóteses sobre o comportamento do processo de falha real. Para que a modelagem seja fiel as hipóteses do modelo aplicado devem ser compatíveis com as características do processo modelado. É necessário, então, analisar a natureza do processo de falha a ser modelado. Os testes apresentados a seguir permitem determinar qual a tendência de uma série de dados de falha.

3.4 Testes de Tendência

A finalidade desta seção é apresentar alguns testes empregados na investigação de tendência de uma série de dados.

Os modelos de confiabilidade se baseiam em hipóteses sobre o comportamento do processo de falha, como foi visto na seção 3.2, e é inadequado empregá-los em bases de dados que descrevam comportamentos diferentes. Seja $N(t)$ o processo de contagem de falhas e $A(t)$ o processo de contagem do número acumulado de falhas a cada unidade fixada de tempo, cuja relação é vista na Figura 2.2. É razoável dizer que um sistema está melhorando (piorando) se ele falha cada vez menos (mais) a cada unidade de tempo. Em outras palavras, a confiabilidade aumenta (diminui) na medida em que o número de falhas por unidade de tempo tende a diminuir (aumentar).

Há dois tipos de teste de tendência: os testes gráficos e os testes estatísticos.

3.4.1 Testes Gráficos

São testes baseados puramente na observação do formato da curva de evolução de uma das grandezas associadas à qualidade de serviço. Há duas classes de testes gráficos, mais adequados respectivamente a dados de falha na forma de número acumulado de falhas a cada unidade fixa de tempo e intervalo de tempo entre falhas.

a) Testes Gráficos de Dados Acumulados de Falhas

Os testes gráficos mais utilizados quando os dados são representados na forma de número de falhas acumulado a cada unidade de tempo tu são:

a.1) Evolução do Número Acumulado de Falhas

Seja $A(t)$ a curva de evolução do número acumulado de falhas a cada unidade fixa de tempo tu. Se a confiabilidade aumenta então o número de falhas diminui ao longo do tempo, e $A(t)$ será côncava. Ao contrário, se a confiabilidade diminui serão detectadas cada vez mais falhas e $A(t)$ será convexa. Se não houver qualquer tendência, o número médio por unidade de tempo pode ser considerado constante e $A(t)$ será aproximadamente uma reta.

a.2) Evolução do Número de Falhas por Unidade de Tempo

Seja tu a unidade de tempo fixada e $Y(i)$ o número de falhas detectadas na i-ésima unidade de tempo . Plotando-se $Y(i)$ contra i percebe-se visualmente se ele tem tendência a crescer, decrescer ou a se manter estável. A tendência do número de falhas por unidade de tempo é inversa à da confiabilidade.

a.3) Evolução da Taxa de Falha Empírica

Seja a k-ésima unidade de tempo. Por definição, a taxa de falha empírica ao fim dessa unidade é

$$\lambda(k) = \frac{A(k.tu)}{k} \tag{3.69}$$

que varia no sentido inverso ao da confiabilidade.

Como esse método faz a média sobre todo o período de observação, ele tem a vantagem de oferecer uma curva mais suave que a do método anterior, evidenciando a tendência global. O aspecto local, no entanto, se perde.

Uma variante do teste de taxa de falha empírica, é obtida aplicando esse teste em janelas de tamanho fixo $j.tu, j < k$ da forma:

$$\lambda(j,k) = \frac{A(k.tu) - A((k-j)tu)}{j} \tag{3.70}$$

na qual a taxa de falha empírica é calculada somente em relação às j últimas falhas, enfatizando o comportamento local do processo.

b) Testes Gráficos de Dados de Intervalos entre Falhas

A seguir são descritos os testes gráficos adequados a dados na forma de intervalo entre a ocorrência de falhas.

b.1) Evolução da Curva de Contagem de Falhas

Este método é análogo ao do gráfico de $A(t)$, apenas agora plotando-se $N(t)$. Do mesmo modo, uma curva côncava revela um crescimento de confiabilidade, uma convexa indica decrescimento e uma reta a estacionariedade.

b.2) Evolução do Intervalo entre Falhas

Plotando o tempo entre falhas t em função de i é fácil perceber se eles tem tendência a crescer, decrescer ou mesmo à estacionariedade. A tendência da confiabilidade é aqui diretamente relacionada à dos intervalos entre falhas.

b.3) Evolução do MTTF Empírico

Seja T o instante de ocorrência da i-ésima falha, ou seja,

$$T_i = \sum_{l=1}^{i} t_l \tag{3.71}$$

Por definição, o $MTTF$ empírico dessa falha é

$$MTTF_i = \frac{T_i}{i} \tag{3.72}$$

que varia no mesmo sentido que a confiabilidade. Como o teste da taxa de falha empírica, este teste também não mostra a tendência local. Uma variante que ressalta esse aspecto pode ser obtida através do uso de janelas de tamanho j, da forma:

$$MTTF_i^k = \frac{1}{j}(T_i - T_{i-k}) \tag{3.73}$$

Os resultados fornecidos por todos os testes são coerentes para uma mesma base de dados, de modo que é suficiente aplicar um deles. Como são testes muito rápidos e fáceis, são também úteis para se ter uma visão geral do comportamento da tendência de uma série estudada. A validação matemática dos resultados dos testes gráficos pode ser feita por um dos testes estatísticos a seguir:

3.4.2 Testes Estatísticos

Os testes estatísticos de tendência são formulados como testes de hipóteses. Por impossibilidade de manipulação a hipótese nula feita na maioria dos testes de tendência é o inverso do que na verdade se quer testar, ou seja:

Ho : o processo do número de falhas a cada intervalo fixo de tempo é um processo de Poisson homogêneo, ou seja, com taxa constante ao longo do tempo, e portanto sem tendência.

Dado $Y(i)$, o processo de falha a cada unidade de tempo i, as hipóteses alternativas, dependendo do teste, são:

H1 : existe uma tendência decrescente de $Y(i)$

H2 : existe uma tendência crescente de $Y(i)$

H3 : existe uma tendência qualquer de $Y(i)$

A seguir são apresentados alguns dos testes mais utilizados:

a) **Teste de Laplace**

O teste de Laplace é baseado no estudo do sinal da diferença entre a média dos instantes de ocorrência das falhas e o meio do intervalo de observação [ASC 84].

Seja : t_t o tempo total de observação. Supondo que o processo de ocorrência de falhas é um processo de Poisson homogêneo, os instantes $Ti, i = 1, ..., n$ de ocorrências dentro do intervalo $[0, t_t]$ correspondem à ordem estatística de uma distribuição uniforme sobre $[0, t_t]$. Dessa forma, o coeficiente de tendência u a seguir tende a uma variável normal padrão. A aproximação é significativa a 95% para $n > 3$:

$$u = \frac{c - m}{t_t}\sqrt{12n} \tag{3.74}$$

onde n é o número de falhas dentro do intervalo de observação, m é o meio do intervalo de observação, ou seja, $m = t_t/2$, e c é o centro estatístico, correspondente à média dos instantes de ocorrência de falhas :

$$c = \frac{1}{n}\sum_{i=1}^{n} T_i \tag{3.75}$$

Quando o processo de falha é representado pelo número de falhas acumulado o coeficiente de Laplace pode ser deduzido conforme indicado em Cox [COX 78]. A dedução desta expressão é apresentada no Anexo C e é dada por:

$$u(k) = \frac{c - m}{\sqrt{\frac{k^2-1}{12A(k.tu)}}}, k = 2, ...p \tag{3.76}$$

$$c = \frac{\sum_{i=1}^{k}(i-1)Y(i)}{A(k.tu)}$$

$$m = \frac{k-1}{2}$$

onde

p : número total de unidades de tempo dentro do intervalo de observação,

$A(k.tu)$: número acumulado de falhas até a k-ésima unidade de tempo e

$Y(i)$: número de falhas durante a unidade de tempo i.

A hipótese Ho deve ser rejeitada ao nível de significância α contra:

H1 : se$P(u(p) < l1) = 1 - \alpha$

H2 : se $P(u(p) > l2) = \alpha$

H3 : se $P(1u(p)1 > l3) = \alpha$

O sinal de u corresponde ao sinal da diferença entre o centro estatístico c e o meio do intervalo de observação m. A interpretação deste teste é de que, em caso de crescimento de confiabilidade os T_i tenderão em média a ocorrer antes do meio do intervalo de observação.

Foi mostrado em alguns estudos [GAU 88,89] que o Teste de Laplace possui excelentes características de otimalidade para os modelos PPNH. Para modelos de outros tipos em alguns casos as propriedades são muito boas; para outros, porém, é impossível dizer quais as suas propriedades.

O crescimento (decrescimento) de confiabilidade é caracterizado por u negativo (positivo). Um processo estacionário deve apresentar $u = 0$. Na prática, porém, admite-se valores entre -2.5 e 2.5 como indicadores de um processo aproximadamente estacionário [KAN 89, GAU 88, 89].

Para um mesmo processo de falha os valores de $u(k)$ obtidos quando se expressa os dados em termos de intervalo entre falhas e número acumulado de falhas a intervalo fixo de tempo são diferentes. Os resultados das duas formas de aplicação do teste, porém, são coerentes.

Qualquer que seja a forma de utilização do teste de Laplace este indica a tendência global da série de dados, uma vez que leva em conta todos os dados anteriores ao índice de falha ou unidade de tempo considerado. A fim de se ter uma visão da evolução da confiabilidade, recomenda-se que o teste seja aplicado progressivamente, a um conjunto cada vez maior de dados de falha, incluindo um dado a cada passo. A tendência local pode ser detectada através do sentido de variação do coeficiente de tendência. Se o teste de Laplace for aplicado progressivamente à série de dados pode-se observar diferentes tipos de comportamento local e global de confiabilidade. A seguir são apresentados

alguns casos possíveis:

- nos casos em que u é positivo e, portanto, há decrescimento global de confiabilidade:
  - quando u tende a crescer, o decrescimento de confiabilidade se acentua,
  - quando u tende a decrescer, a tendência global ao decrescimento de confiabilidade vem sendo suavisada por um crescimento local de confiabilidade,
- nos casos em que u é negativo e, dessa forma, há crescimento global de confiabilidade:
  - quando u tende a decrescer, o crescimento de confiabilidade é acentuado,
  - quando u tende a crescer, o crescimento de confiabilidade global é atenuado por um decrescimento local de confiabilidade.

Esses casos são ilustrados pela Figura 3.1. A forma da curva da taxa de falha associada a $u(k)$ é representada a fim de indicar a variação local de confiabilidade.
Os pontos de inversão de tendência de $u(k)$ são os pontos de inflexão da curva número acumulado de falhas observadas $N(t)$. É de se esperar que estes pontos coincidam com os pontos de inflexão da curva $H(t)$ estimada pelos modelos de confiabilidade que sejam utilizados para modelar esse processo de falha.

Na Figura 3.1 pode-se ver que há dois tipos de ponto de inflexão:

Tipo 1 : $u(k)$ é crescente e passa a decrescer devido a uma tendência local ao crescimento de confiabilidade,

Tipo 2 : $u(k)$ é decrescente e passa a crescer devido ao surgimento de uma tendência local ao decrescimento de confiabilidade.

O resultado do teste de Laplace depende da forma de sua aplicação. Se, por exemplo, forem descartados os dados relativos à zona **A** da Figura 3.1 e a seguir avaliado o coeficiente de tendência para os dados correspondentes às zonas **B**, **C** e **D** somente, este coeficiente se torna negativo sobre todo o período analisado.Isto ocorre uma vez que a tendência a crescimento local de **B** se torna agora global sobre os dados estudados. Os valores obtidos para $u(k)$ são obviamente diferentes dos anteriores.

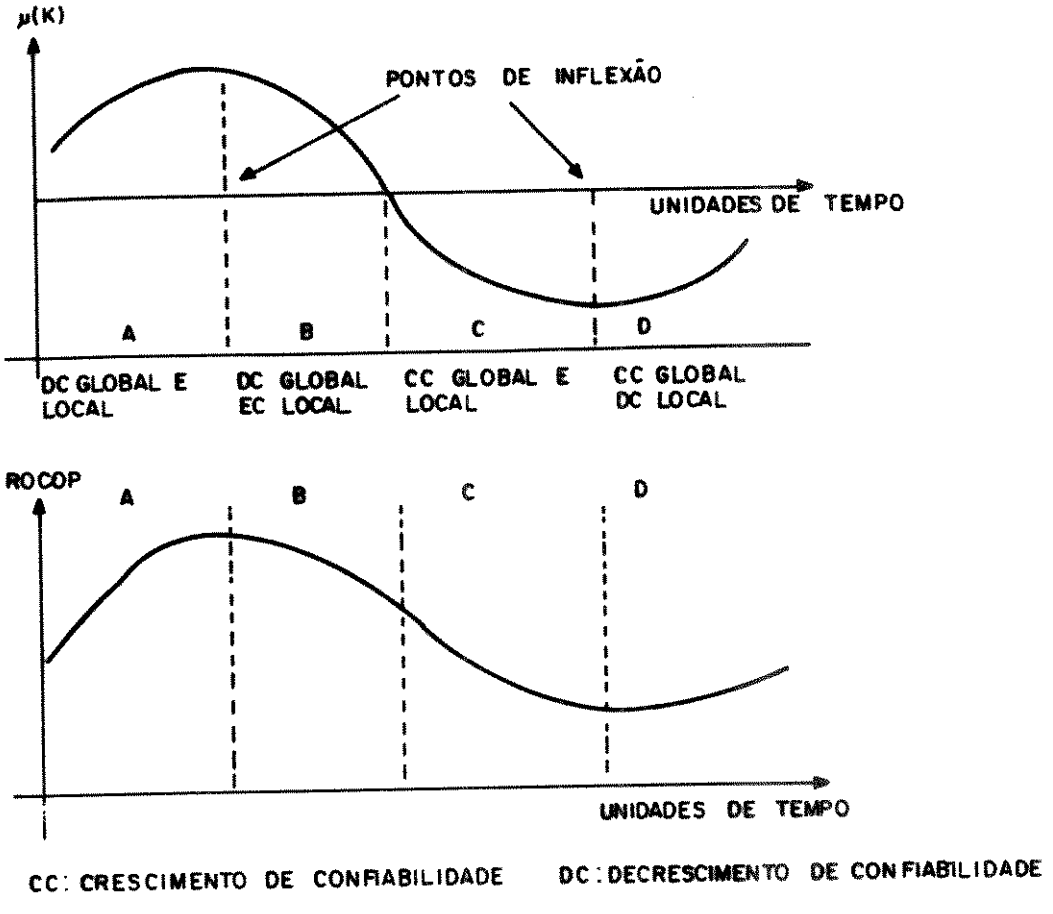

Figura 3.1 - Coeficiente de Laplace e taxa de ocorrência de falhas em função da evolução da confiabilidade.

Uma série de dados de falha pode apresentar diversos padrões de comportamento local e global ao longo do ciclo de vida do sistema.

Processo Transitório

As características das falhas de concepção, apresentadas no Capítulo 1, levam a crer que um processo de falhas desse tipo apresente sempre crescimento de confiabilidade. No entanto, um processo de falha pode alternar períodos de decrescimento e crescimento de confiabilidade e mesmo de estacionariedade durante a etapa de testes e início de vida operacional.

Um decrescimento de confiabilidade após um período de crescimento de confiabilidade denota aparentemente uma regressão na qualidade do sistema estudado, porém na verdade tem outras explicações. Um decrescimento local de confiabilidade pode ser o resultado de :

- dependência entre as falhas latentes : algumas falhas podem ser escondidas por outras, isto é, não podem ser ativadas enquanto outras não forem removidas [LIT 81b],
- a variação do intervalo de tempo entre a detecção de um defeito e a sua remoção. Este intervalo de tempo está intimamente ligado à natureza das falhas detectadas : algumas falhas são mais difíceis de serem identificadas que outras e necessitam de mais tempo para serem removidas,
- variação na utilização : a variação no esforço de teste durante a fase de depuração, a mudança do conjunto de testes, a ativação de novas instalações durante a vida operacional, etc...

Estes fatores podem atuar junta ou separadamente.

Processo de Crescimento de Confiabilidade

Após algum tempo de operação o processo de falha tende ao crescimento de confiabilidade, segundo a hipótese básica feita sobre a natureza das falhas de concepção. Durante este período $u(k) < 0$.

Os processos de falha que apresentarem somente crescimento de confiabilidade durante o período transitório têm uma curva de $u(k)$ sempre negativo e decrescente. Estes processos podem ser modelados através da grande maioria dos modelos.

Alguns processos apresentam decrescimento seguido de crescimento de confiabilidade. Para estes a evolução do coeficiente de tendência $u(k)$ segue o padrão **A-B-C** da Figura 3.1. Estes modelos podem ser representados pelos modelos Gama ou S. Os processos que não se enquadrarem nesses padrões não devem ser modelados diretamente, sob pena de se obter resultados sem sentido ou mesmo de levar à divergência dos métodos de estimação dos parâmetros. A proposta é que seja usado um método de partição e um método de

composição de modelos baseado no teste de tendência. Este método será apresentado no Capítulo 4.

Processo Permanente

Depois de um tempo relativamente longo de operação o processo tende a se estabilizar em uma das duas situações discutidas na apresentação dos modelos: a intensidade de falha assume um valor constante ou nulo.

1. Intensidade de falha nula a partir de k_e: O processo de falha se interrompe em k_e. A depuração do sistema elimina todos os seus defeitos e ele não falha mais a partir desse momento. O coeficiente de Laplace para esta série passa a ter um valor constante e igual a $u(k_e) < 0$.

2. Intensidade de falha constante a partir de k_e: A partir de k_e o coeficiente $u(k)$, que era negativo e decrescente no período anterior, passa a crescer até se estabilizar em $-2.5 < u(k) < 0$. No limite quando $k \rightarrow \infty$ $u(k) = 0$. O processo passa a ser estacionário com taxa igual à do processo permanente, dada por α.

Qualquer que seja a hipótese assumida o comportamento do processo de falha visto pelo usuário durante um longo tempo de vida operacional será de um processo estacionário com taxa de falha muito baixa, tendendo a se anular. Somente depois de muito tempo de operação é que o comportamento do processo de falha se distinguirá em função da tendência da intensidade de falha. Um comportamento típico de um processo de falha composto de um transitório com períodos de decrescimento e crescimento de confiabilidade é mostrado na Figura 3.2. A modelagem de processos de falha como este será discutida no próximo Capítulo.

A evolução do Coeficiente de Laplace pode também ser usada para identificar em que fase se encontra o sistema:

fase transitória : o processo de falha oscila entre crescimento e decrescimento de confiabilidade;

fase de crescimento de confiabilidade : nesta fase o processo apresenta uma intensidade de falha monotonamente decrescente;

fase estacionária : um processo estacionário após algum tempo de operação, depois de ter passado por uma fase de crescimento de confiabilidade e com uma intensidade de falha baixa indica que o sistema atingiu a maturidade. O valor aceitável da taxa de falha residual depende de cada projeto.

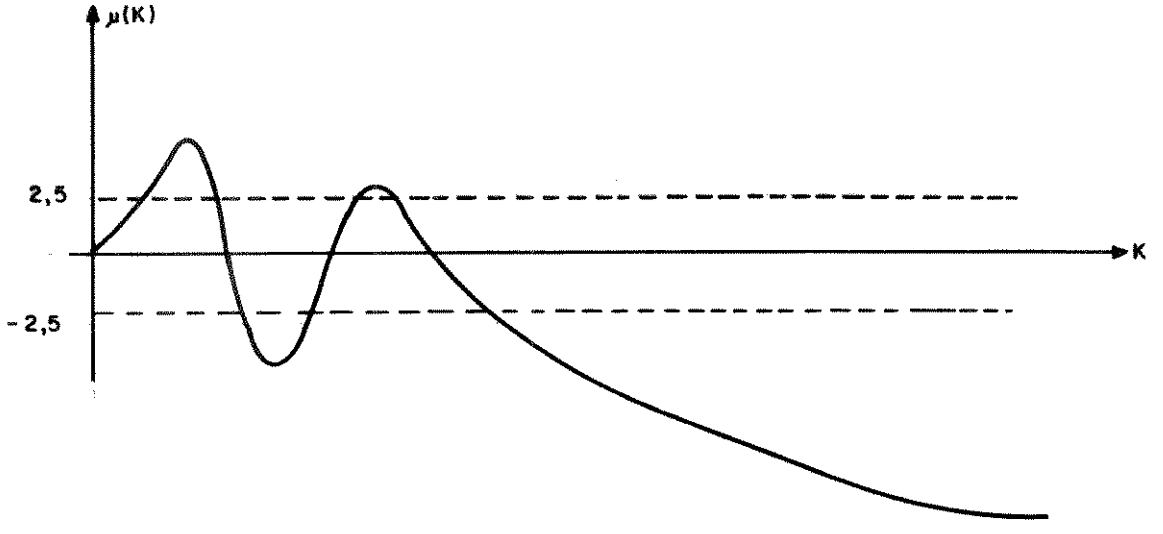

Figura 3.2 - Exemplo da evolução de $u(k)$ para um processo de falha com intensidade de falha tendendo a constante.

Suponha-se que pelos testes de tendência seja determinada a tendência do processo de falha estudado. Com base neste comportamento é possível selecionar entre os modelos de confiabilidade alguns que tenham características compatíveis. Suponha-se que estes modelos sejam aplicados à base de dados. Como determinar qual o modelo que melhor descreve o processo de falha é o tema da próxima seção.

3.5 Validação dos Modelos

A validação dos modelos de confiabilidade de concepção, ou seja, a avaliação da sua capacidade de modelagem de uma certa base de dados de falha pode ser feita de duas maneiras:

validação absoluta : consiste em determinar se um certo modelo é capaz ou não (dentro de um certo nível de confiança) de representar uma dada base de dados de

falha;

validação relativa : trata-se de estabelecer um critério segundo o qual a capacidade de dois ou mais modelos em representar uma certa base de dados é comparada.

Um modelo pode ser validado quanto à sua capacidade:

replicativa : reprodução do comportamento do processo de falha durante o período de observação;

previsiva : previsão do comportamento futuro do processo de falha com base no comportamento durante o período de observação.

Os critérios de validação são os mesmos para a validação replicativa e previsiva. Para a validação replicativa é avaliada a diferença entre a curva estimada e a detectada efetivamente durante o período de observação. Quanto à validação da previsão, o cálculo das diferenças é feito somente para o período previsto, excluindo o período de observação. Os critérios de validação mais utilizados são:

3.5.1 Critério de Kolmogorov-Smirnov:

Seja N o número de falhas detectadas até o momento e

$$F_N(t) = \frac{número\ de\ t_i \leq t}{N}$$

para qualquer t e $u_i = F_i(t)$ a probabilidade , segundo $F_i(t)$, que a função distribuição estimada do tempo até a próxima falha baseado nas observações anteriores da variável T_i ser menor que o valor realmente observado de t_i. Se $F_i(t)$ é realmente a função distribuição estimada de T_i então u_i é a realização de uma variável aleatória uniforme sobre $(0,1)$. Seja u_i uma sequência de u_i em ordem crescente. Chama-se *gráfico de u* ao desenho dos pontos u_i contra i/N, e por isso esse método é também chamado de *u-plot*.

Se os u_i são uniformemente distribuidos, os pontos deste gráfico estarão sobre uma linha de declividade unitária através da origem, a menos de um pequeno ruído aleatório. A medida da máxima distância vertical de qualquer dos u_i até a reta (distância de Kolmogorov-Smirnov) dá a medida de quanto a distribuição estimada está distante da "verdadeira".

Sejam as estatísticas:

$$D_N^+ = max_{1\leq i\leq N}(\frac{i}{N} - u_i) \tag{3.77}$$

$$D_N^- = max_{1\leq i\leq N}(u_i - \frac{i-1}{N}) \tag{3.78}$$

$$D_N = max(D_N^+, D_N^-) \tag{3.79}$$

Se a distância $d_{N,\alpha}$ é tal que $P(D_N > d_{N,\alpha})$ então o teste estatístico consiste em aceitar a hipótese de que a função estimada realmente descreve o processo, com um nível de confiança de $(1-\alpha)$ se o valor de D_N é superior a $d_{N,\alpha}$ (validação absoluta). Os valores limites de $d_{N,\alpha}$ são dados em tabelas [HOG 67].

Estes valores limites, no entanto, são calculados supondo que a distribuição a ser testada é inteiramente conhecida. Como os parâmetros são estimados, os valores limites não são adequados a uma validação absoluta, porém como D_N mede a diferença entre as funções estimada e real, ela permite servir de fator de comparação entre dois modelos diferentes aplicados a uma mesma base de dados (validação relativa).

3.5.2 Critério y-plot

Este critério foi desenvolvido com base no critério anterior, a fim de levar em conta a ordem de detecção das falhas, examinando assim a existência de tendência na previsão feita pelo modelo.

O método consiste em avaliar a distância de Kolmogorov-Smirnov sobre y_i , definida através da transformação

$$x = -ln(1-u_i) \tag{3.80}$$

$$y_i = \frac{\sum_{j=1}^{i} x_j}{\sum_{j=1}^{N} x_j} \tag{3.81}$$

onde os y_i são ordenados de forma crescente. A distância de Kolmogorov-Smirnov é calculada através da simples substituição de u_i por y_i nas equações D e a distância máxima entre os y_i e a reta de degrau unitário representa tendência nas previsões. Os testes de u-plot e y-plot podem ser usados de forma complementar.

A análise de tendência feita através do y-plot pode também ser feita usando o Teste de Laplace ou outros testes gráficos [ABD 86, MEL 87].

3.5.3 Verossimilhança Prequencial

Seja $F_i(t)$ a função distribuição do i-ésimo tempo até falhar T_i e considere $f_i(t)$ sua função distribuição de probabilidade. A verossimilhança prequencial para as N previsões de T_i é dada por

$$PL_N = \prod_{i=1}^{N} f_i(t) \tag{3.82}$$

Para dois modelos a e b aplicados ao mesmo conjunto de dados tem-se PL_N^a e PL_N^b . A razão:

$$PN_N^{a,b} = \frac{PL_N^a}{PL_N^b} \tag{3.83}$$

é chamada de razão de verossimilhança prequencial entre as estimações dos dois modelos. Se

$$lim PL_{N\to\infty}^{a,b} = \infty$$

o modelo b pode ser descartado em favor do modelo a, ou seja, o modelo a descreve aquela base de dados muito melhor que o modelo b (validação relativa). Ainda assim, porém, não se pode dizer que o modelo a descreve bem o processo (validação absoluta).

Se o limite acima não se verificar, não é possível, por esse critério, decidir entre dois modelos e eles são ditos equivalentes.

3.5.4 Critério dos Resíduos

Esse critério se baseia no cálculo das diferenças entre os valores observados e os valores estimados da variável aleatória que descreve o processo de falha, seja ela o intervalo de tempo entre a ocorrência de falhas ou o número acumulado de falhas em um certo intervalo fixo de tempo [KAN 89].

Sejam:

N(i) : o número de falhas observado durante a i-ésima unidade de tempo,

Y(i) : o número de falhas até o fim da i-ésima unidade de tempo,

H(i) : o valor da função valor médio estimado pelo modelo ao fim da i-ésima unidade de tempo,

k : unidade de tempo a partir da qual se fez a estimação dos parâmetros do modelo,

p : unidade de tempo até a qual se fez a estimação dos parâmetros do modelo.

O resíduo é definido por :

$$r_i = Y(i) - H(i), i = k, ..., p \tag{3.84}$$

Dependendo do interesse do usuário, a validação pode ser feita estudando-se as seguintes funções dos resíduos:

a)Soma dos Resíduos

$$Rs = \sum_{i=k}^{p} r_i \tag{3.85}$$

Indica se a estimação é otimista $(Rs > 0)$ ou pessimista $(Rs < 0)$.

b) Soma dos Quadrados dos Resíduos

$$Rq = \sum_{i=k}^{p} r_i^2 \tag{3.86}$$

Este critério indica qual o modelo com desempenho mais uniforme.

c) Média dos Valores Absolutos dos Resíduos

$$Ra = \frac{\sum_{i=k}^{p} |r_i|}{p - k - 1} \tag{3.87}$$

Este critério dá idéia do erro global cometido na estimação, tratando todos os resíduos da mesma forma.

d) Soma dos Valores Absolutos dos Resíduos Relativos

$$Rr = \sum_{i=k}^{p} |\frac{r_i}{Y(i)}| \tag{3.88}$$

Este critério prioriza os erros cometidos ao final do período de observação, uma vez que, supondo crescimento de confiabilidade, o número de falhas nos intervalos de tempo finais é bem menor que nos iniciais. A validação com base neste critério é apropriada quando o objetivo é fazer previsões com o modelo baseadas no comportamento recente do processo de falha.

Em qualquer das funções dos resíduos quanto menor o valor da função mais adequado é o modelo. É importante notar que ao comparar dois modelos um pode se mostrar melhor que o outro segundo uma função do resíduo e pior segundo outra. A evolução dos valores dos resíduos e de suas funções no tempo é muito mais rica em informações que o seu valor total no intervalo de observação. O estudo dos resíduos e suas funções para o caso em que o processo de falha é representado pelos intervalos entre falhas é análogo.

Todos os critérios de adequação de um modelo à representação de um processo de falha devem ser usados de forma comparativa. Não há um critério que permita dizer se um modelo segue bem uma curva de falha ou não. Neste sentido todos os critérios apresentados são equivalentes.

O objetivo da aplicação dos modelos de confiabilidade, neste trabalho, é mostrar que uma forma de utilização baseada nas características do processo de falha e do modelo empregado leva a melhores resultados. Dessa maneira, um critério comparativo é suficiente.

O critério de verossimilhanca prequencial é útil somente quando um modelo é muito melhor que o outro na representação de uma base de dados, o que não acontece frequentemente na prática.

O critério dos resíduos, além de poder ser usado de forma comparativa, explorando diversos aspectos da adequação dos modelos, tem a vantagem de dar uma ordem de

grandeza do erro entre a estimação e o realmente observado. Além disso, as funções dos resíduos tem um forte sentido físico que permite compreender melhor o desempenho dos modelos. Por estas razões será dada preferência aos critérios de resíduos.

A comparação dos resultados dos modelos aplicados nos Capítulos 4 e 5 deve dar uma idéia global do quanto a curva de falha estimada se afasta da real. Não é tão importante saber se a modelagem é otimista ou pessimista, se é uniforme ou se apresenta discrepâncias pontuais significativas ou se os erros são maiores em certo trecho da curva. O critério de validação mais adequado para os objetivos deste trabalho é a média dos valores absolutos dos resíduos, que será referenciada de ora em diante simplesmente como *erro*. Este critério tem como vantagens a simplicidade e o fato de poder ser diretamente comparado à taxa de falha do processo.

3.6 Conclusão

O objetivo principal deste capítulo foi o de apresentar alguns métodos matemáticos relacionados ao problema de avaliação da confiabilidade de concepção de sistemas. Os pontos abordados estão ligados ao conhecimento das hipóteses que sustentam os modelos existentes e das características do processo de falha estudado e da validação dos resultados da aplicação de modelos.

Foi apresentada uma forma de classificar os modelos de crescimento de confiabilidade inicialmente dedicados à avaliação de software. Esta classificação sem pretender ser exaustiva, permite organizar os principais modelos apresentados na literatura segundo suas características e hipóteses fundamentais. A forma inicial da classificação, proposta por Musa, foi estendida, a fim de incorporar os modelos a intensidade de falha não nula no infinito. Outros modelos foram também classificados de acordo com o esquema original e incluidos nas tabelas.

Alguns modelos foram apresentados de forma sucinta, destacando suas propriedades. Foram apresentados os tipos mais utilizados de testes de tendência, tanto gráficos como estatísticos. A forma de validar os resultados fornecidos pelos modelos na replicação e na previsão de maneira absoluta e comparativa do processo de falha foi apresentada. Os

principais critérios foram descritos.

Ao comparar as hipóteses dos modelos e os possíveis tipos de comportamento de um processo de falha pode-se perceber que só processos bem definidos podem ser seguidos diretamente por um modelo: processos de crescimento de confiabilidade e processos em que há um período inicial de decrescimento de confiabilidade seguido de crescimento. Qualquer processo de falha que siga um padrão diferente destes está sujeito a maus resultados de modelagem.

Como não é raro observar processos reais fora dos padrões dos modelos, o uso sem critério dos modelos de confiabilidade tem levado muitos pesquisadores a desanimar em vista dos resultados obtidos.

A maneira proposta de aplicar os modelos de crescimento de confiabilidade baseada no conhecimento dos modelos e do processo estudado é o assunto do capítulo 4 a seguir.

Capítulo 4

Metodologia de Aplicação dos Modelos de Crescimento de Confiabilidade

4.1 Introdução

A história da avaliação da confiabilidade de concepção esteve inicialmente ligada à confiabilidade de software, por motivos já expostos no item 1.1. Os primeiros esforços, realizados na primeira metade da década de 70, exploravam as diferenças de abordagem que se faziam necessárias em relação à confiabilidade de componentes hardware [DUA 64, WAG 73, CRO 74, END 75, SCH 75]. Foram propostos nessa época os primeiros modelos analíticos dedicados ao acompanhamento da confiabilidade de software [JEL 72, SCH 72, LIT 73, SCH 73, MOR 75, MUS 75], ou seja, modelos que supunham um crescimento de confiabilidade ao longo do tempo.

Na segunda metade da década de 70 e primeira da década de 80 os modelos se multiplicaram de modo a representar diversos comportamentos do processo de falha software [GOE 79b, LIT 81a, SCH 81, KRE 83, YAM 83a, CRO 84, MUS 84, OHB 84a, CAT 85, KAN 85]. No início dos anos 80, dada a proliferação de modelos, o trabalho foi direcionado à classificação e comparação dos modelos a fim de selecionar um (ou alguns) que fosse(m) o melhor (ou os melhores). Foram estudados critérios de comparação [IAN 84] e novos modelos surgiram a partir da visão geral assim adquirida [MUS 84, KAN 85, FUK 86]. Algumas questões específicas passaram a ser aprofundadas, como o aspecto de validação dos modelos [ANG 80], sua capacidade de previsão [KEI 83], tempo de liberação de novas versões [OKU 80] e técnicas de estimação dos parâmetros [LIT 81b, SPR 85].

Nos últimos cinco anos foram publicados trabalhos com resultados práticos de modelos existentes aplicados a bases de dados de falha reais de sistemas em operação, a fim de qualificá-los [SAB 86 e 87, KAN 87b, c e d, KAH 87]. A teoria desenvolvida nos anos anteriores estava dando frutos . Ainda assim, as análises de confiabilidade eram feitas post mortem, ou seja, depois do desenvolvimento do sistema estar concluído.

Estes resultados animaram a avaliação de sistemas a partir de dados de falha não só em operação comercial como também na fase de desenvolvimento [BAS 88a e b, BAS 89a e b, BAS 90a e b] e a utilizar as técnicas de avaliação de confiabilidade de concepção dentro de um programa atuante ao longo de todo o ciclo de vida de um sistema.

O novo enfoque buscado neste trabalho é o de partir dos modelos de avaliação de confiabilidade desenvolvidos e utilizados, em sua grande maioria, no âmbito acadêmico, e mostrar como usar os resultados da avaliação como auxílio à tomada de decisões gerenciais, ou seja, interferir no processo de desenvolvimento de sistemas (ver Capítulo 1).

Neste contexto podem ser feitas as seguintes considerações:

- O número de modelos existentes é grande, não encorajando novas propostas.
- Os modelos de crescimento de confiabilidade de software servem também à modelagem do processo de falha de concepção do hardware, uma vez que é de se esperar que este cresça em confiabilidade ao longo do tempo.
- Não existe um modelo que seja capaz de replicar satisfatoriamente qualquer processo de falhas.
- Existe um modelo mais conveniente à replicação de uma dada base de dados; isto é função da adequação do modelo a esses dados.
- As aplicações da análise de confiabilidade de concepção a sistemas em desenvolvimento e mesmo em operação ainda são incipientes, no sentido que raramente têm sido usadas para especificar, implementar, testar, qualificar o sistema, ou seja: suportar decisões gerenciais.

Com base nestas observações decidiu-se investir no estudo do modo de aplicação dos modelos. Buscou-se uma metodologia que permitisse, dado um projeto em andamento, acompanhar a confiabilidade e usar os resultados para reorientar o processo de desenvolvimento.

A metodologia proposta se baseia na aplicação do teste de tendência; é geral e válida para qualquer modelo empregado. Os pontos principais desta metodologia são:

1. Critérios de escolha do modelo a ser aplicado a uma base de dados e sua forma de aplicação: este problema será tratado na seção 4.2 e
2. Método de partição de uma série de dados de falha: alguns processos de falha são bastante complexos e não seguem nenhum dos comportamentos descritos pelos modelos de crescimento de confiabilidade existentes. Isto não requer que seja desenvolvido um modelo específico para o caso; o processo pode ser estudado através da aplicação conveniente de um ou mais modelos, bastando para isso particioná-lo de maneira adequada, como mostrado na seção 4.3.

Uma outra proposta deste trabalho é que sejam empregados os mesmos modelos desenvolvidos para o software, assim como a metodologia de aplicação destes modelos na avaliação da confiabilidade de concepção do hardware.

Para ilustrar a metodologia proposta foram escolhidos dois modelos que atendem aos critérios expostos na seção 3.2.6: o modelo Exponencial de Goel e Okumoto [GOE 79] e o modelo Gama de Yamada e Osaki [YAM 83a]. Ambos são modelos Poissonianos Não-Homogêneos, a falhas finitas e dois parâmetros. O primeiro é capaz de representar processos de crescimento de confiabilidade e estacionários. O segundo descreve processos de decrescimento seguido de crescimento de confiabilidade. Assume-se a hipótese de que os sistemas estudados serão depurados até que todos os defeitos latentes sejam eliminados. O método usado para a estimação dos parâmetros é o de máxima verossimilhança, com o método de otimização de Newton-Raphson, que são descritos no Anexo D.

O teste de tendência analítico utilizado é o teste de Laplace por suas qualidades em relação a processos de Poisson não-homogêneos, que é a hipótese feita acerca dos proces-

sos de falha pelos modelos que serão aplicados neste estudo.

As análises de sistemas reais mostradas nos exemplos deste capítulo são feitas a partir do ponto de vista de quem desenvolve o sistema, ou seja, estão relacionadas ao Sistema Fonte. A análise dos Sistemas de Aplicação e Instalações é análoga e não foi feita pela indisponibilidade de dados de falha por Aplicação e Instalação.

4.2 Comparação de Modelos

Vários autores dedicaram trabalhos à comparação do desempenho de modelos de crescimento de confiabilidade. A comparação tem sido feita em dois sentidos :

teórica [MEL 87, MUS 83 e 87,...]: analisando a plausibilidade das hipóteses que sustentam os modelos, sua facilidade de implementação e utilização, e

prática : a partir do surgimento dos primeiros dados de falha confiáveis de sistemas reais [MUS 79], os modelos eram aplicados e os resultados obtidos comparados. Um modelo que tivesse bom desempenho para diversas bases de dados era considerado eficiente [KAN 85, MAT 88].

Como foi dito no item 4.1, não existe um modelo melhor, de aplicação universal, com bom desempenho qualquer que seja a base de dados. Existe um modelo melhor para a replicação de um certo processo de falha por ser o mais adequado a ele, ou seja, as hipóteses do modelo são válidas para o processo.

Antes de aplicar um modelo de crescimento de confiabilidade é necessário analisar as características de tendência do processo de falha em questão e escolher um modelo que tenha hipóteses compatíveis.

Exemplo 4.1
Tome-se a base de dados de falha do software do sistema de comutação digital TRÓPICO R versão 4096 (ver Anexo E). O registro de falhas foi feito durante as fases de Teste de Sistema e Operação Comercial, na forma de falhas acumuladas a cada dez dias de teste/operação. O resultado do Teste de Laplace mostrando a evolução desse processo de falha é mostrado na Figura 4.1. O teste indica que há um decrescimento de confiabilidade até a unidade de tempo t=11 e crescimento de confiabilidade dali em diante. Esse comportamento sugere uma curva de falha em S.

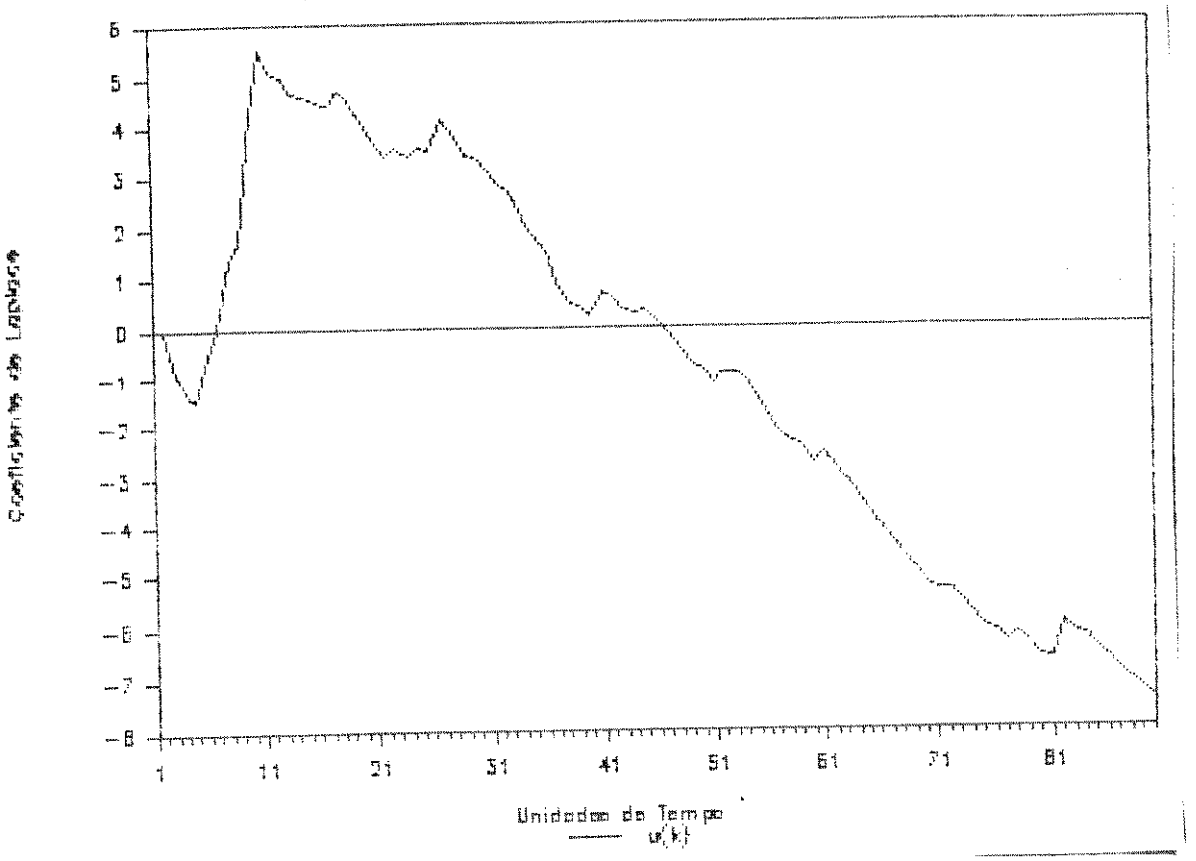

Figura 4.1 - Resultados do Teste de Laplace para o software do TRÓPICO R 4096.

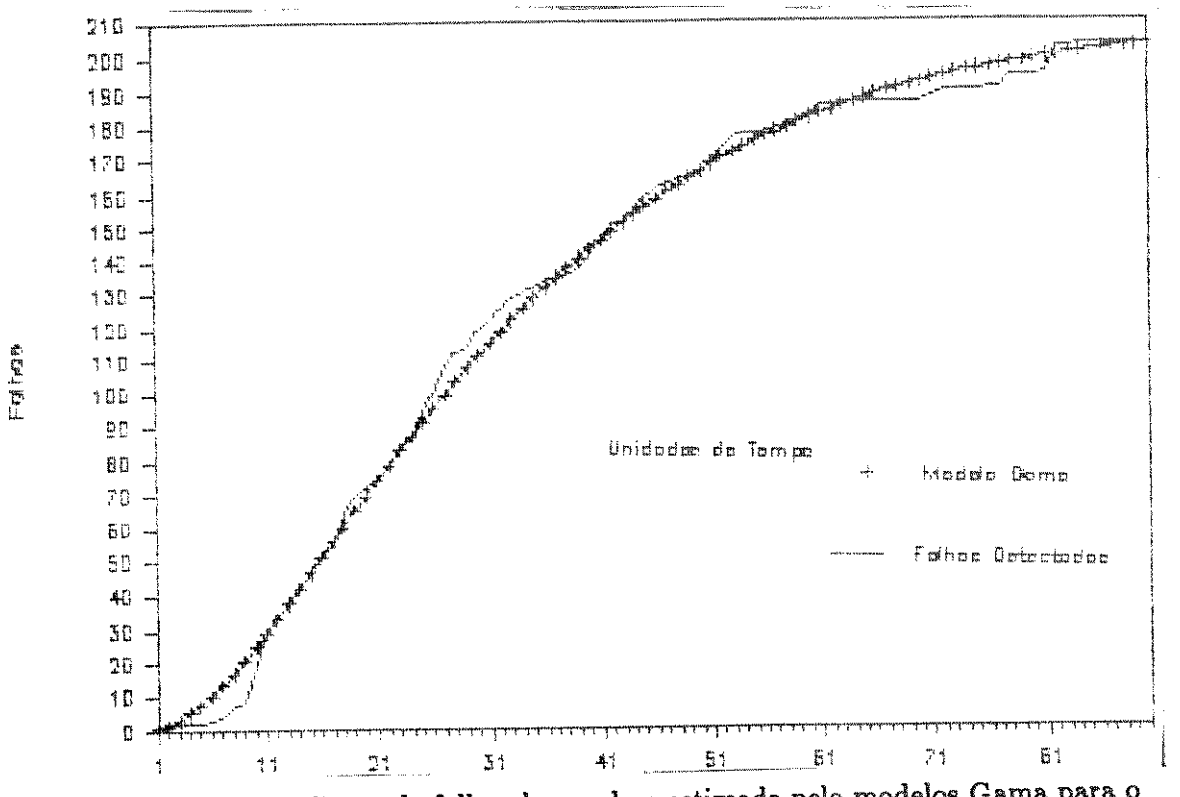

Figura 4.2 - Curva de falha observada e estimada pelo modelos Gama para o software do sistema TRÓPICO R 4096.

Certamente a aplicação de um modelo exponencial à base de dados completa não é indicada. O modelo Gama, por sua vez, é um modelo capaz de representar um processo como este, dada a hipótese que faz sobre a intensidade de falha. A curva de falha observada e a função valor médio H(t) estimada com base em todos os dados de falha pelo modelo Gama são mostrados na Figura 4.2. O erro médio cometido na modelagem é de 3.09 falhas por unidade de tempo. Aplicando o modelo Exponencial de Goel e Okumoto, o erro médio seria de 9.34 falhas por unidade de tempo.

A adequação de um modelo depende, além disso, do modo como ele é aplicado. Um processo com períodos de crescimento e decrescimento de confiabilidade pode ser modelado por um modelo de crescimento de confiabilidade, desde que a modelagem se restrinja aos períodos em que há somente crescimento de confiabilidade.

Exemplo 4.2
Seja a mesma base de dados do Exemplo 1, cuja modelagem através do modelo Exponencial é mostrada na Figura 4.3.

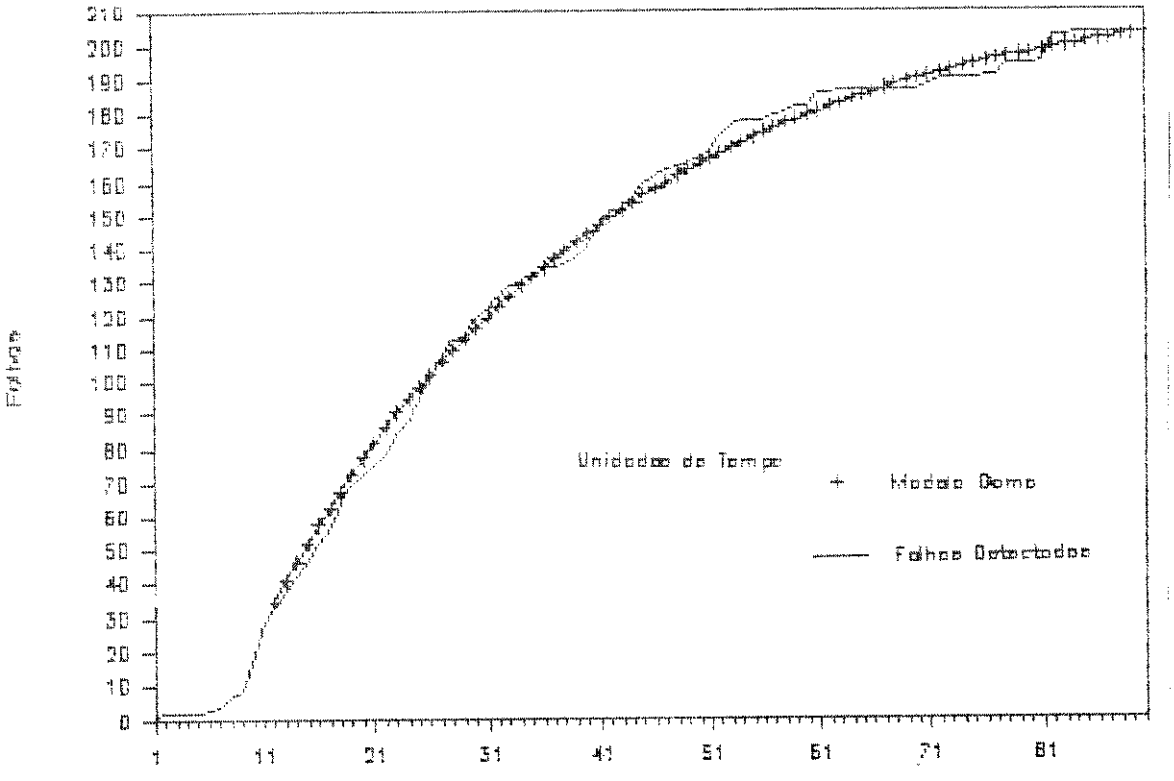

Figura 4.3 - Modelagem a curva de falha do software do sistema TRÓPICO R 4096 através do modelo Exponencial.

Um modelo Exponencial como o de Goel e Okumoto pode fornecer bons resultados desde que sejam desconsiderados, para efeito de estimação dos parâmetros e modelagem, os dados de falha anteriores à unidade de tempo t=11, relativos ao período

de decrescimento de confiabilidade. A hipótese do modelo passa a ser satisfeita no período restante. O erro médio cometido é de 3.16 falhas por unidade de tempo, que é um desempenho comparável ao do modelo Gama para a base de dados completa.

Há alguns tipos de comportamento de um processo de falha no instante de avaliação que trazem problemas para a modelagem:

Decrescimento Global de Confiabilidade: se este comportamento ocorrer durante um período de teste ou início de vida operacional, ele pode ser transitório e devido a uma solicitação diferente do sistema. Nesse caso deve-se investir na depuração do sistema. Este comportamento, se apresentado durante a vida operacional do produto, pode ser devido a manutenção predatória do sistema e deve levar a um exame sério dos procedimentos de manutenção adotados. Na maioria dos casos o processo de falha pode ser replicado por um modelo Exponencial, porém os parâmetros do modelo perdem seu significado físico, uma vez que podem ser negativos.

Confiabilidade Estacionária: este comportamento em vida operacional pode corresponder também a um processo de manutenção pouco eficiente. Em fase de teste o processo pode apresentar durante algum tempo um comportamento transitório enquanto a tendência a crescimento de confiabilidade ainda não estiver bem definida. Um período de estacionariedade sem crescimento de confiabilidade posterior pode ser replicado satisfatoriamente através de um modelo Exponencial. Uma condição de confiabilidade estacionária pode ocorrer também em uma situação de regime, em que a taxa de falha é baixa e tolerada. Um modelo a falhas infinitas e intensidade de falha não nula no infinito pode ser empregado.

Confiabilidade Oscilante: durante as etapas de teste é possível que o processo de falha se apresente instável durante um certo período, alternando curtos períodos de crescimento, decrescimento e estacionariedade de confiabilidade, devido a variação na intensidade e/ou eficiência dos testes. Neste caso não há modelo capaz de representar satisfatoriamente o processo e deve ser indicada uma reavaliação da política de testes.

Dependendo das características do processo de falha estudado a aplicação direta de modelos, sem a análise prévia da evolução de tendência pode não só levar a erros maiores de modelagem como a conclusões erradas sobre os modelos em comparações práticas.

Exemplo 4.3
Tome-se como exemplo a análise de cinco sistemas software feita em [MAT 88], cujos resultados de tendência são dados nas figuras a seguir.
Estes sistemas foram desenvolvidos por alunos da Universidade de Osaka, e estudo tem a finalidade de comparar alguns modelos de crescimento de confiabilidade.
Três desses sistemas (X132, X133 e X136) apresentam o coeficiente de tendência $-2.5 < u(k) < 2.5$, de modo que, pelo que foi apresentado no Capítulo 3, pode-se considerar que eles têm confiabilidade estacionária. O sistema X125 é um exemplo de processo com oscilação de confiabilidade e o sistema X134 apresenta um comportamento de decrescimento seguido de crescimento de confiabilidade. Se fosse feita uma comparação entre os resultados dos modelos Gama e Exponencial aplicados a estes sistemas, chegar-se-ia à conclusão equivocada de que o modelo Exponencial é "melhor" que o modelo Gama.

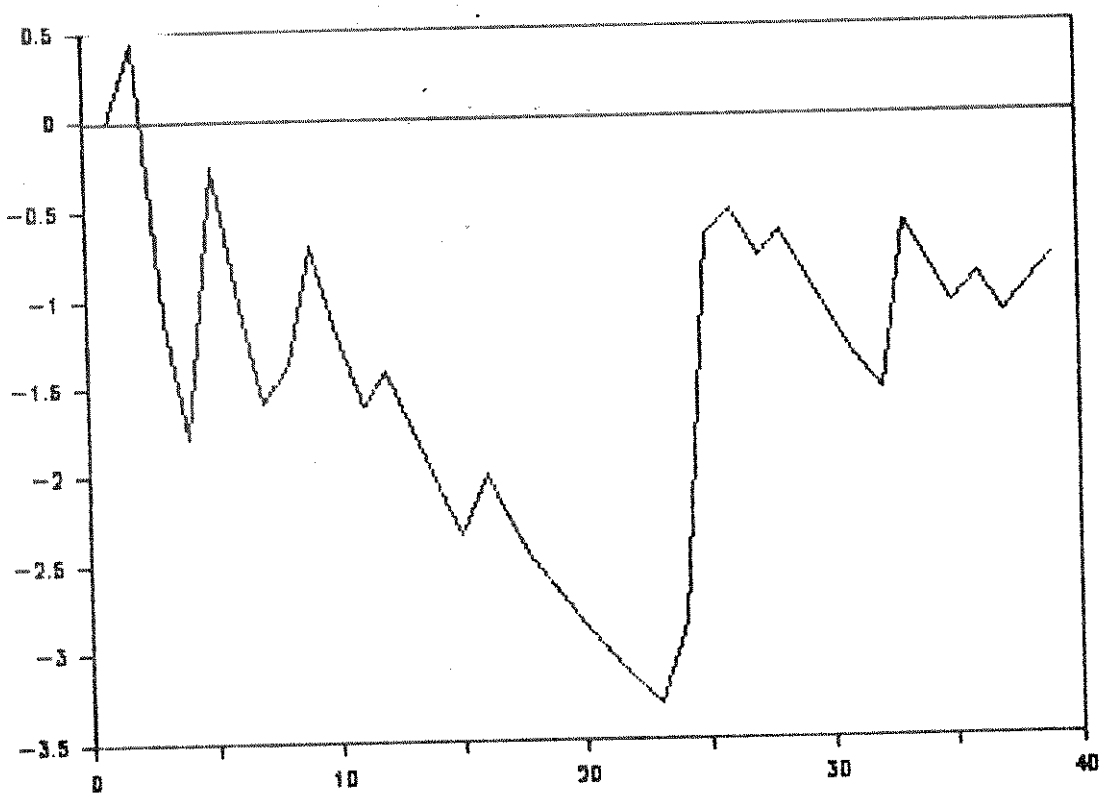

Figura 4.4 - Teste de tendência do sistema X125.

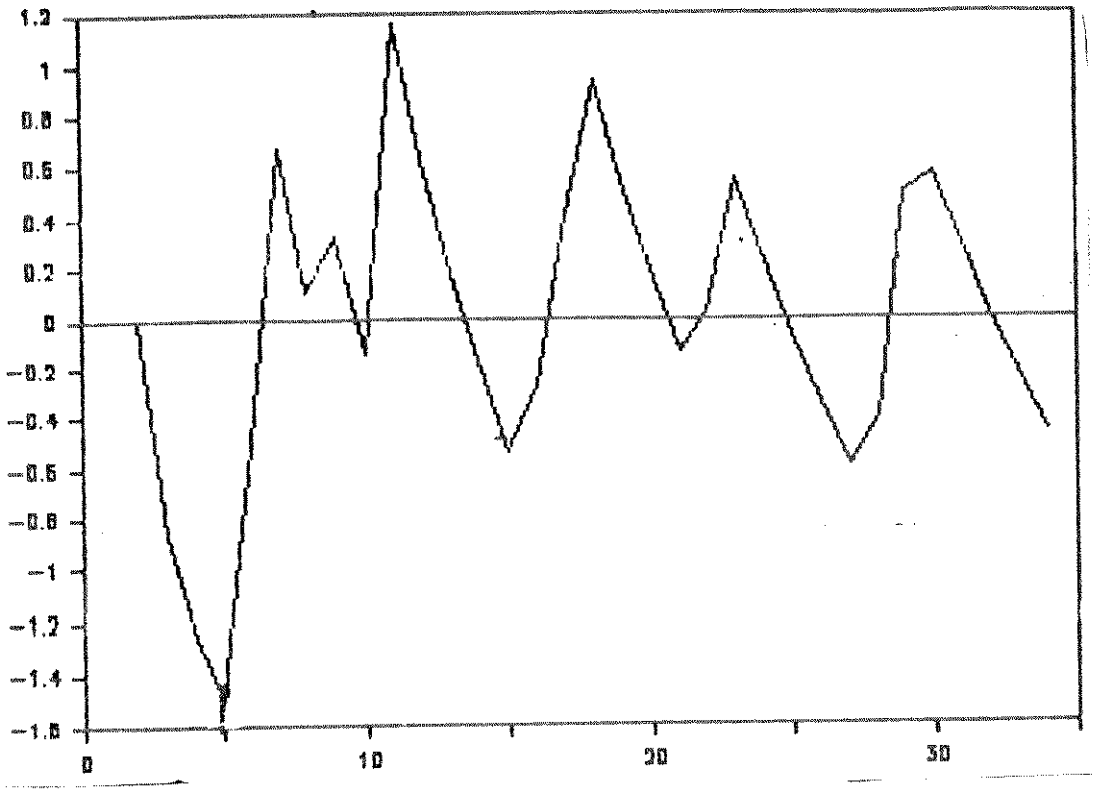

Figura 4.5 - Teste de tendência do sistema X132.

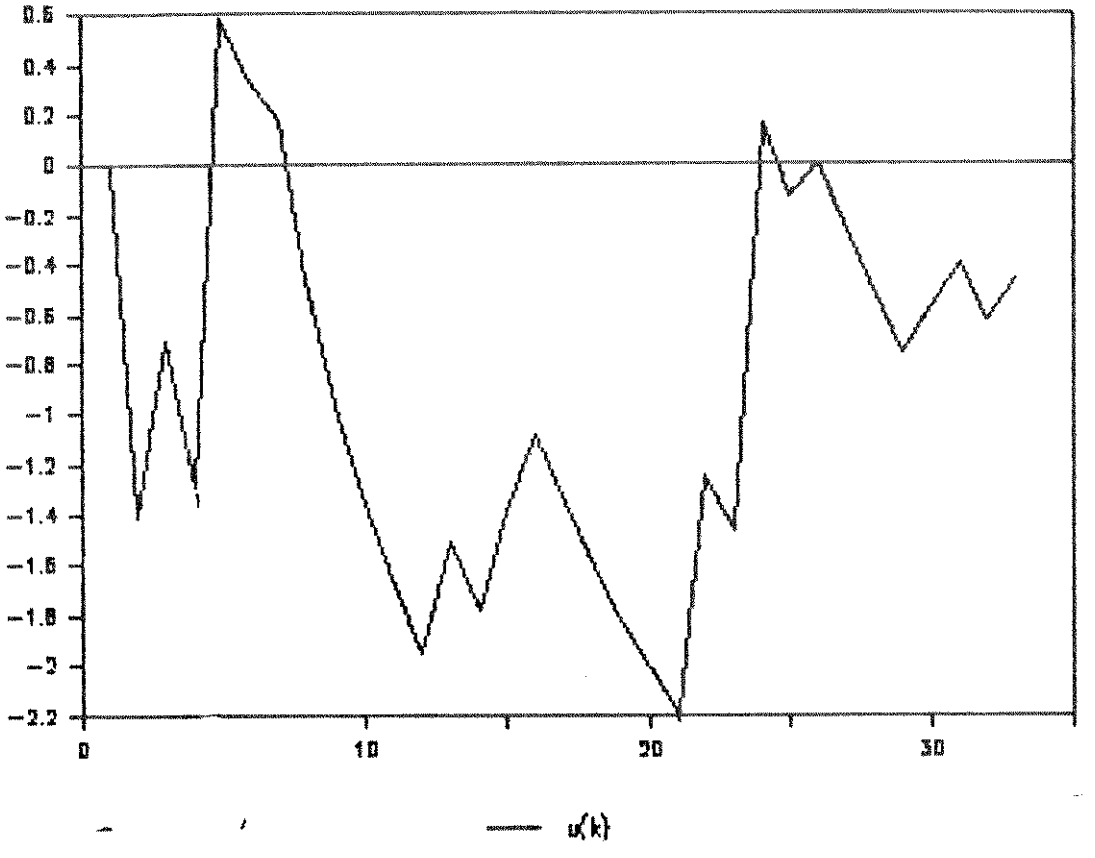

Figura 4.6 - Teste de tendência do sistema X133.

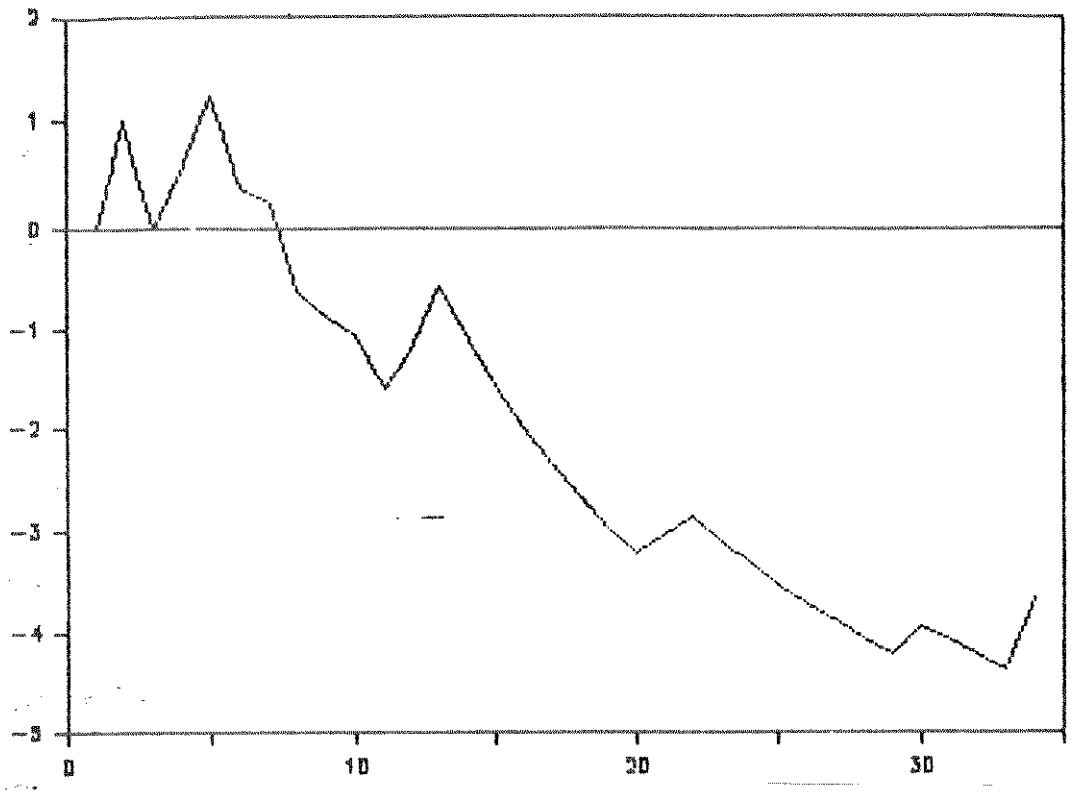

Figura 4.7 - Teste de tendência do sistema X134.

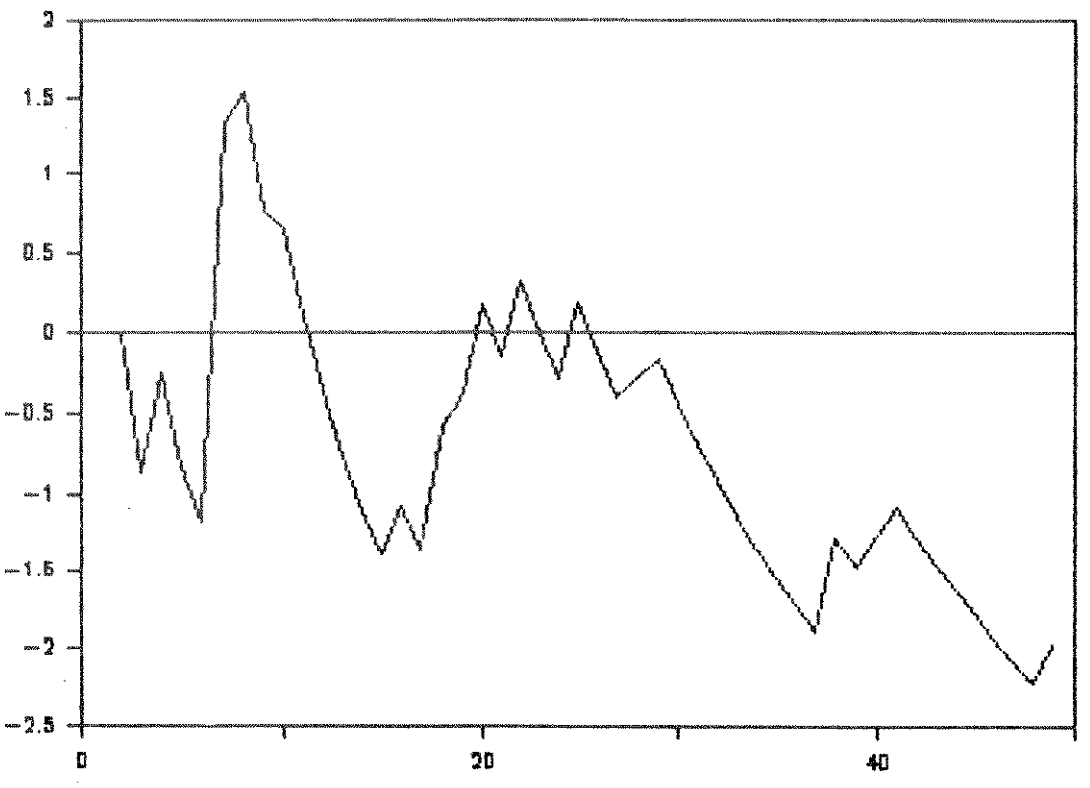

Figura 4.8 - Teste de tendência do sistema X136.

Um modelo dá bons resultados quando empregado em situações para as quais ele foi previsto e maus, caso contrário. Desse ponto de vista a comparação prática entre modelos tal como é feita em [MAT 88] não é adequada pois, a partir dos resultados, é possível dizer somente qual o modelo mais ou menos indicado à modelagem de uma certa base de dados e não o mais eficiente.

4.3 Partição dos Dados de Falha

Sistemas complexos, acompanhados desde o seu desenvolvimento até a vida operacional, podem apresentar processos de falha que dificilmente sejam modelados satisfatoriamente de maneira global pela aplicação direta de um modelo. Este fenômeno é ilustrado pelo estudo feito em [DER 90], no qual o autor expressa a sua decepção em relação ao desempenho de alguns modelos na replicação de curvas de falha de sistemas reais. A solução proposta para esses casos é de que a série de dados de falha seja particionada, e cada subconjunto de dados estudado separadamente. Em alguns casos é necessário mais de um modelo para representar o processo.

Fukushima e Kishida [FUK 86] foram os primeiros a propor a partição de uma base de dados. Neste trabalho os autores defendem que uma partição deve ser feita se e quando houver alguma adição ou modificação de especificação do sistema e/ou alteração no método de teste ou na função do sistema que está sendo testada. Este método supõe o conhecimento do processo de teste e baseia-se na hipótese de que toda modificação no sistema, no método de teste e na área do sistema testada leva a uma perturbação no comportamento do processo de falha.

Um outro método de partição dispensa o conhecimento das modificações e método de teste e baseia-se na hipótese de que o processo de falha não sofre variações bruscas e segue o comportamento local [SAB 87]. A estimação dos parâmetros de um modelo aplicado até um instante t é feita a partir de um número fixo j de dados anteriores, que define uma janela de visibilidade.

Em [BAS 88] foi proposto um outro método de partição: a série de dados deveria ser particionada se e quando houvesse uma mudança no método de teste (fase de teste)

e/ou o teste de tendência de Laplace indicasse a existência de um ponto de inflexão do tipo dois. Este método considera que a mudança no método de teste sempre afeta o processo de falha, e que dentro de um mesmo período de teste podem ocorrer inversões na tendência a crescimento de confiabilidade. Estas inversões devem ser detectadas através do teste de tendência. Se for identificado um ponto de inflexão tipo 2, então a série de dados deve ser particionada. Os modelos mais adequados devem ser aplicados a cada sub-conjunto de dados, ou seja, os modelos cuja tendência é aproximadamente a mesma observada no sub-período.

A evolução do estudo de confiabilidade levou à conclusão de que toda a série de dados de falha, coletados ao longo das diversas fases do ciclo de vida, deve ser analisada por meio do teste de tendência, e que as partições devem ser determinadas pelos pontos de inflexão tipo 2, se existirem.

Este método é baseado na observação feita em [BAS 90b] de que variações na solicitação do sistema (nova fase do ciclo de vida, nova bateria de testes, crescimento do número de sistemas em operação, aprendizado da atividade de teste, etc...) ou no sistema em si (liberação de novas edições, introdução de novas funções, remoção de defeitos, etc...) algumas vezes não causam impacto no processo de falha. Esta constatação também foi feita por Tohma e outros em [TOH 89], que propõem uma solução baseada em uma partição empírica da base de dados.

A hipótese que sustenta este método de partição é de que o impacto na curva de falhas causado por alterações na forma de utilização de um sistema, e a existência de fatores intrínsecos, como dependência entre falhas, são supostos observáveis através do exame de tendência, e não necessariamente definidos a priori.

O método de partição proposto consiste na identificação de trechos da curva de falha cujo comportamento de tendência corresponda à hipótese dos modelos a serem utilizados. No contexto deste trabalho, no qual são considerados os modelos Exponencial e Gama trata-se de identificar trechos de crescimento global de confiabilidade, decrescimento seguido de crescimento ou estacionariedade, pois estas são as tendências passíveis de serem modeladas. Do ponto de vista prático, a curva deve ser particionada onde a

confiabilidade começa a decrescer, depois de um período de crescimento, caracterizando um ponto de inflexão do tipo 2. Estes são os pontos nos quais modificações no produto ou na sua utilização são percebidas. A partir deste ponto as condições iniciais, e portanto os resultados da modelagem, não são mais válidos.

Exemplo 4.4
Seja a base de dados de falha do software do sistema de comutação digital TRÓPICO R, versão para 1500 terminais (Anexo E). O primeiro passo é estudar a tendência da série de dados. A evolução do valor do coeficiente de Laplace para cada unidade de tempo está mostrado na Figura 4.9.
Os dados de 1 a 30 correspondem à fase de Teste de Sistema de um protótipo em laboratório, as unidades de t= 31 a t= 42 à fase de teste em campo, em condições de operação próximas das reais, e de t= 43 em diante à operação comercial do produto. Há um decrescimento de confiabilidade de t=1 a t=5, antecedendo um crescimento de confiabilidade de t=6 a t=14. Em t=15 a tendência a crescimento de confiabilidade se inverte para comecar a crescer novamente em t=24. Em t=55 a tendência a crescimento é atenuada, porém volta a se acelerar depois de t=70.

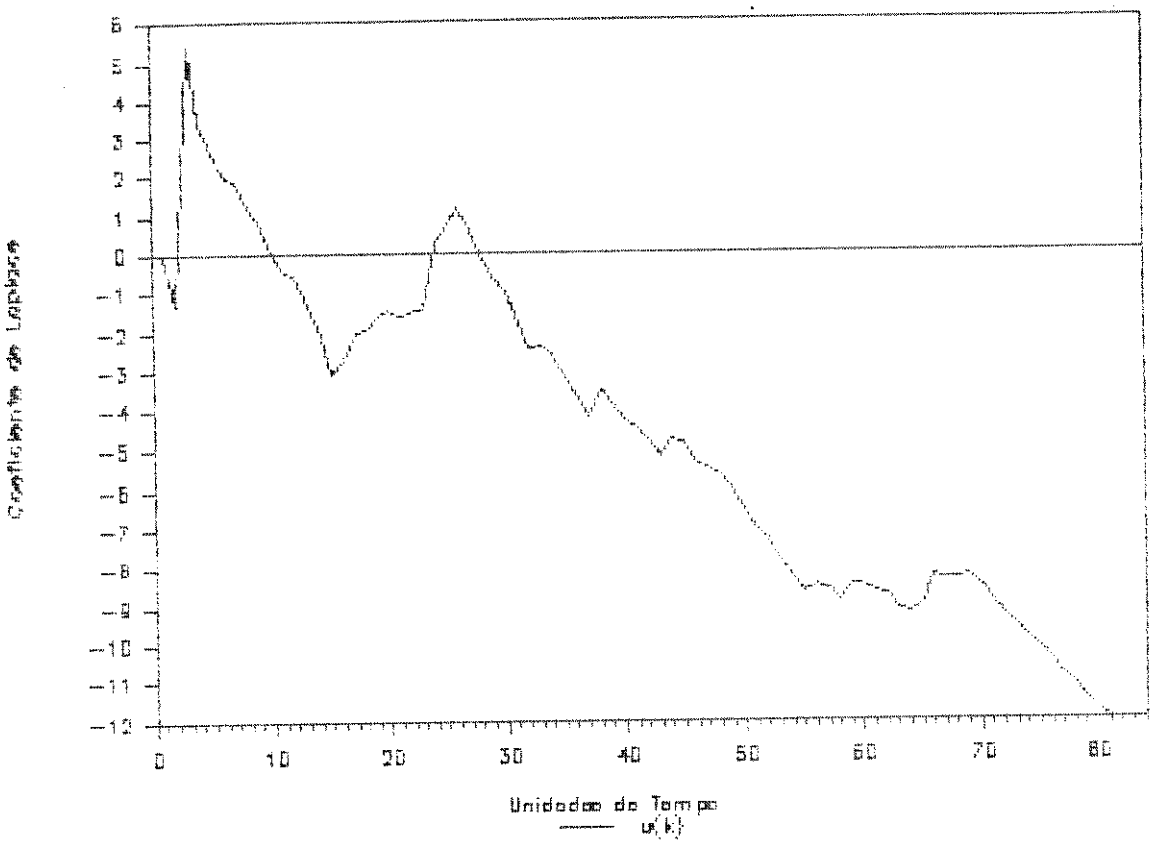

Figura 4.9 - Evolução do coeficiente de Laplace para o software do sistema TRÓPICO R 1500.

Examinando o período de Teste de Sistema nota-se que o processo de falha pode exibir um crescimento de confiabilidade seguido de decrescimento (ou vice-versa e mais de uma vez) durante a mesma fase do ciclo de vida. Este é um comportamento

difícil de ser previsto e mesmo replicado por um modelo. A explicação para este comportamento está na variação da forma de solicitação do sistema. No caso estudado, a reversão na tendência a crescimento de confiabilidade observada em t=15 foi causada pela instalação de um novo protótipo, a qual foram conectados alguns assinantes reais. Este protótipo foi submetido a testes com a finalidade de preparar o sistema para o Teste de Aceitação que precede a aceitação do sistema pelo usuário. A operação do sistema em ambiente que se assemelhava ao real permitiu a manifestação de alguns defeitos que permaneceram latentes em ambiente de teste controlado.
Não houve impacto significativo na tendência ao fim do Teste de Sistema ocorrido em t=30 (talvez por este ter sido atenuado pela instalação prévia do protótipo em condições reais de operação). Tampouco houve uma perturbação na curva de falhas digna de nota ao final do Teste em Campo, t=42. A confiabilidade continuou crescendo desde o final do Teste de Sistema até o início da Operação Comercial. Durante o primeiro semestre de 1986 (t=55 a t=70) a confiabilidade não cresceu, devido às falhas detectadas durante o Teste de Sistema da nova versão para 4096 terminais, que impactava a versão para 1500 terminais. Estas falhas estavam relacionadas aos módulos de software comuns a ambas as versões (ver Anexo E) e que foram contabilizados nas respectivas bases de dados de falha. No segundo semestre de 1986 (t=70 em diante) a confiabilidade da versão 4096 começou a crescer e consequentemente também a da versão 1500.

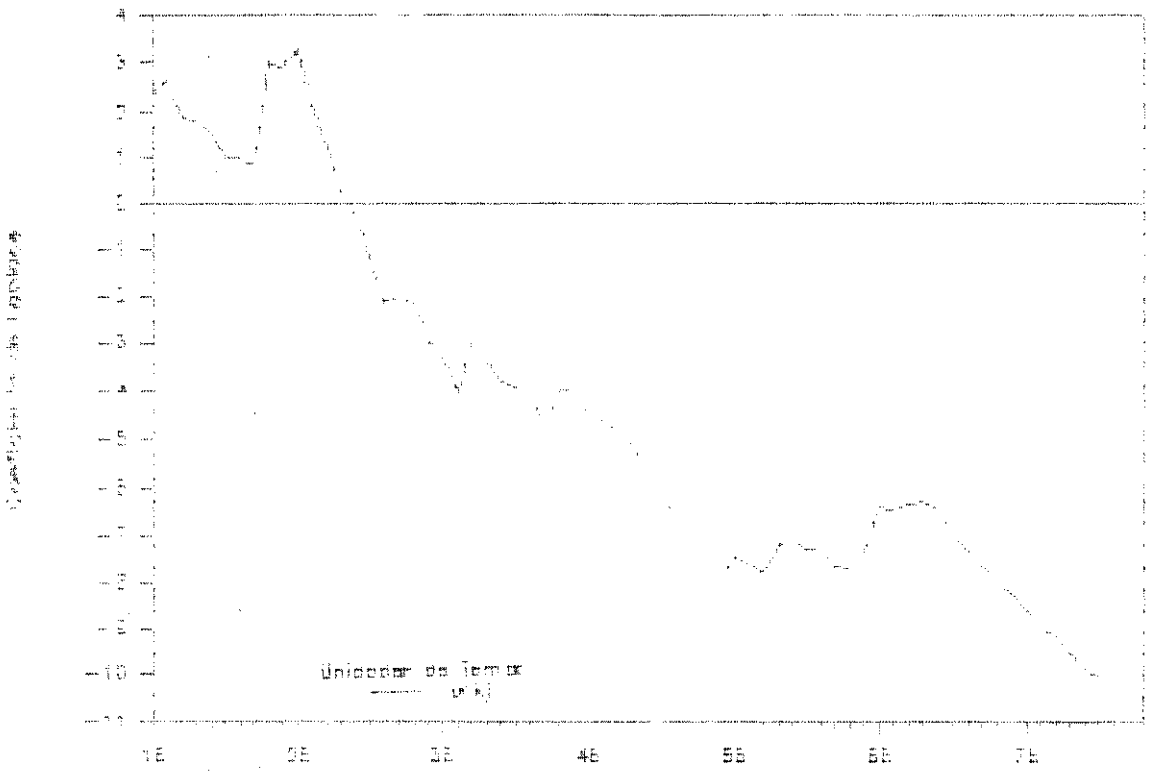

Figura 4.10 - Resultado do teste de tendência aplicado à base de dados excluindo G1.

É importante notar que as explicações para comportamento de falha foram buscadas em função dos resultados dos testes de tendência, e não consideradas anteriormente. De acordo com os resultados dos testes de tendência os dados de falha devem ser

particionados em: **G1**: t=1 a t=14 (primeiro ponto de inflexão do tipo 2). Aplicando novamente o teste de tendência aos dados restantes obtém-se a curva mostrada na Figura 4.10.
Um novo ponto de inflexão do tipo 2 é identificado em t=54, o que sugere mais uma partição: **G2**: t=15 a t=54.
Aplicando novamente o teste de tendência à base de dados restante obtém-se o resultado mostrado na Figura 4.11. Como não há outro ponto de inflexão do tipo 2, não é necessário fazer mais partições, e os demais dados restantes são agrupados em: **G3**: t=55 em diante.

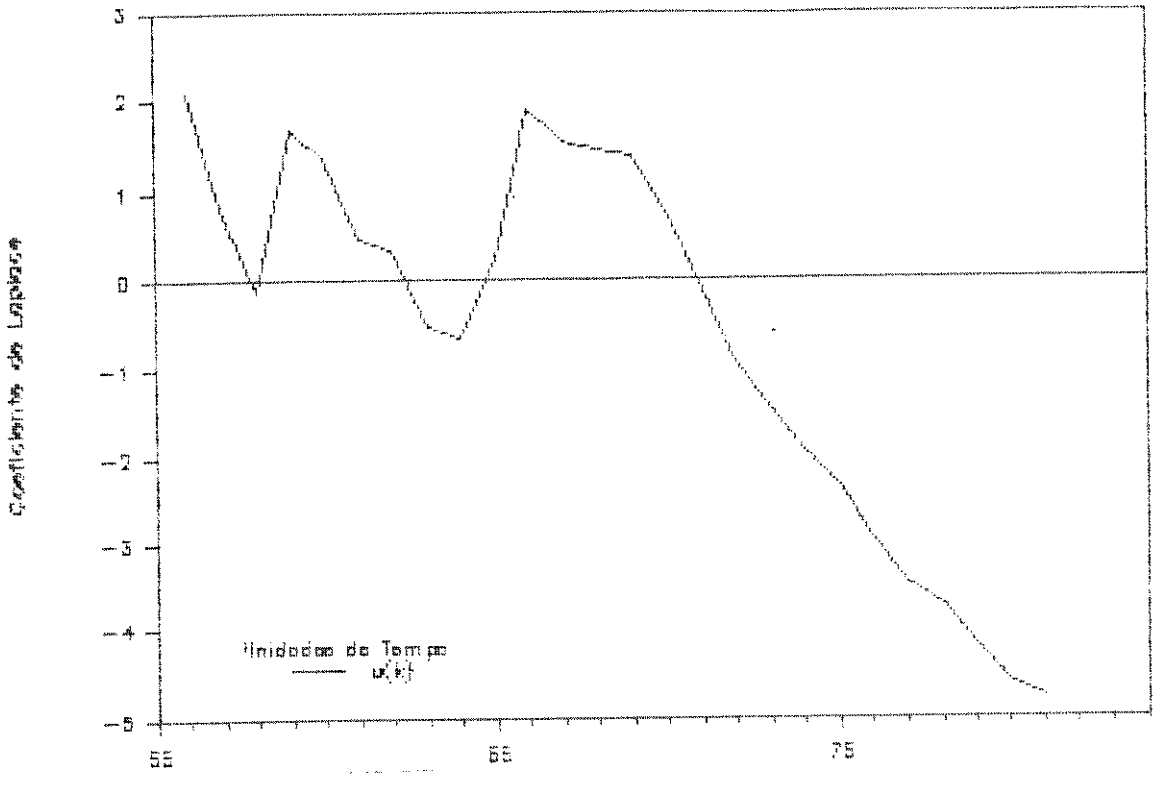

Figura 4.11 - Resultado do teste de tendência aplicado à base de dados excluindo G1 e G2.

As tendências apresentadas em G1 e G3 indicam uma curva em forma de S e sugerem a aplicação de um modelo desse tipo, como o modelo Gama. O sub-grupo G2 tem uma curva bastante irregular, de tendência oscilante que, como dito na seção 4.3, é difícil de ser modelada.
O modelo Gama aplicado a G1 e G3 dá erros de 2.23 e 2.84 falhas por unidade de tempo respectivamente. O período G2 é modelado com um erro de 6.93 pelo modelo Exponencial e de 7.77 falhas por unidade de tempo pelo modelo Gama. Como previsto pelo teste de tendência os resultados não são bons para este período.
A curva de falha estimada é composta pela curva estimada para cada partição, como é mostrado na Figura 4.12. Foi usado o modelo Exponencial para representar o período G2 por ter apresentado um erro menor.

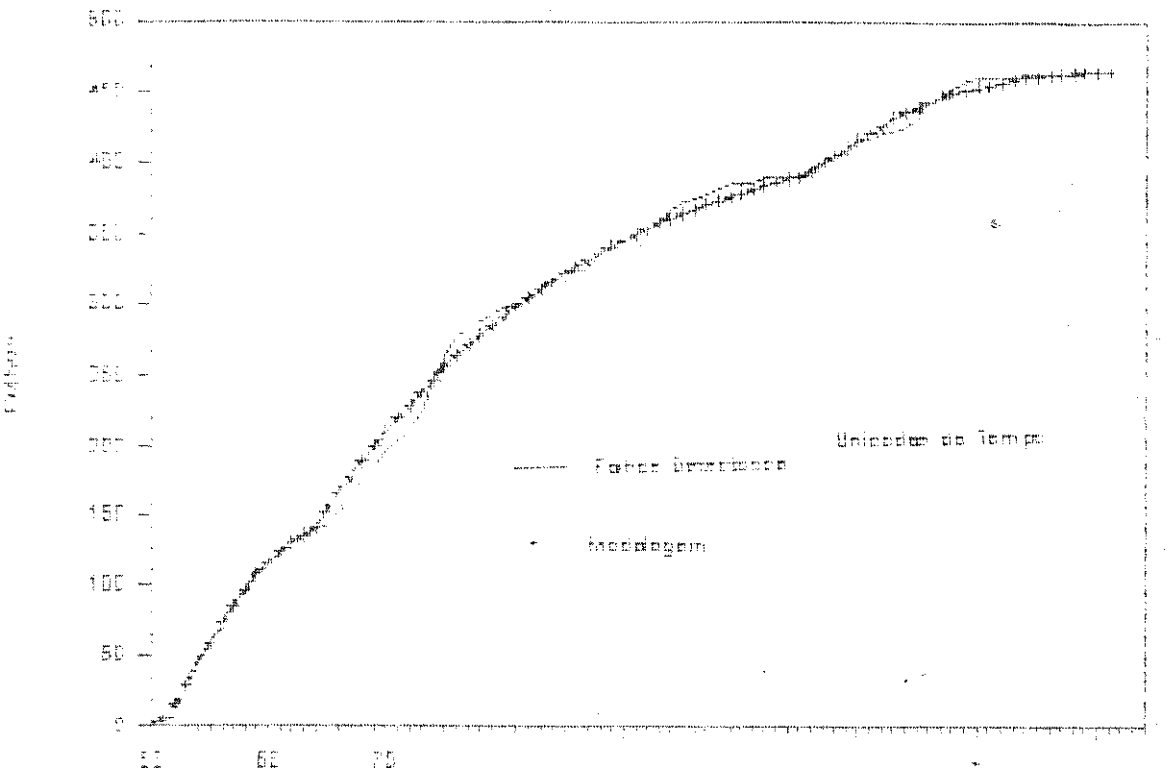

Figura 4.12 - Modelagem do processo de falha completo do software do sistema TRÓPICO R 1500.

O erro médio cometido pela replicação da curva de falha observada e pela curva estimada composta é de 4.75 falhas por unidade de tempo.
Se fossem aplicados o modelo Exponencial e o modelo Gama à representação de todo o processo de falha, teriam sido obtidos os erros de 8.78 e 16.63 falhas por unidade de tempo respectivamente, que são muito maiores que os atingidos através do método de partição.

Exemplo 4.5
Neste exemplo é aplicado o método de partição proposto à base de dados de falha do Sistema II apresentado em [TOH 89]. O resultado do teste de tendência é mostrado na Figura 4.13. O cronograma de testes aplicados a este sistema é o seguinte:
T1: de t=1 a t=31
T2: de t=32 a t=90
T3: de t=91 a t=119
T4: de t=120 a t=125
T5: de t=126 a t=185
A partição indicada pelo teste de tendência é dada por:
G1: t=1 a t=25, região de crescimento de confiabilidade, para a qual é indicado o modelo Exponencial;
G2: t=26 a t=40, região de decrescimento seguido de crescimento de confiabilidade: modelo Gama;
G3: t=41 a t=70, região de oscilação seguida de crescimento de confiabilidade: modelo Gama;

G4: t=71 a t=110, região de decrescimento seguido de crescimento de confiabilidade: modelo Gama;
G5: t=111 a t=125, região de decrescimento (embora oscilante) de confiabilidade: modelo Exponencial;
G6: t=126 a t=185, região de crescimento de confiabilidade: modelo Exponencial.

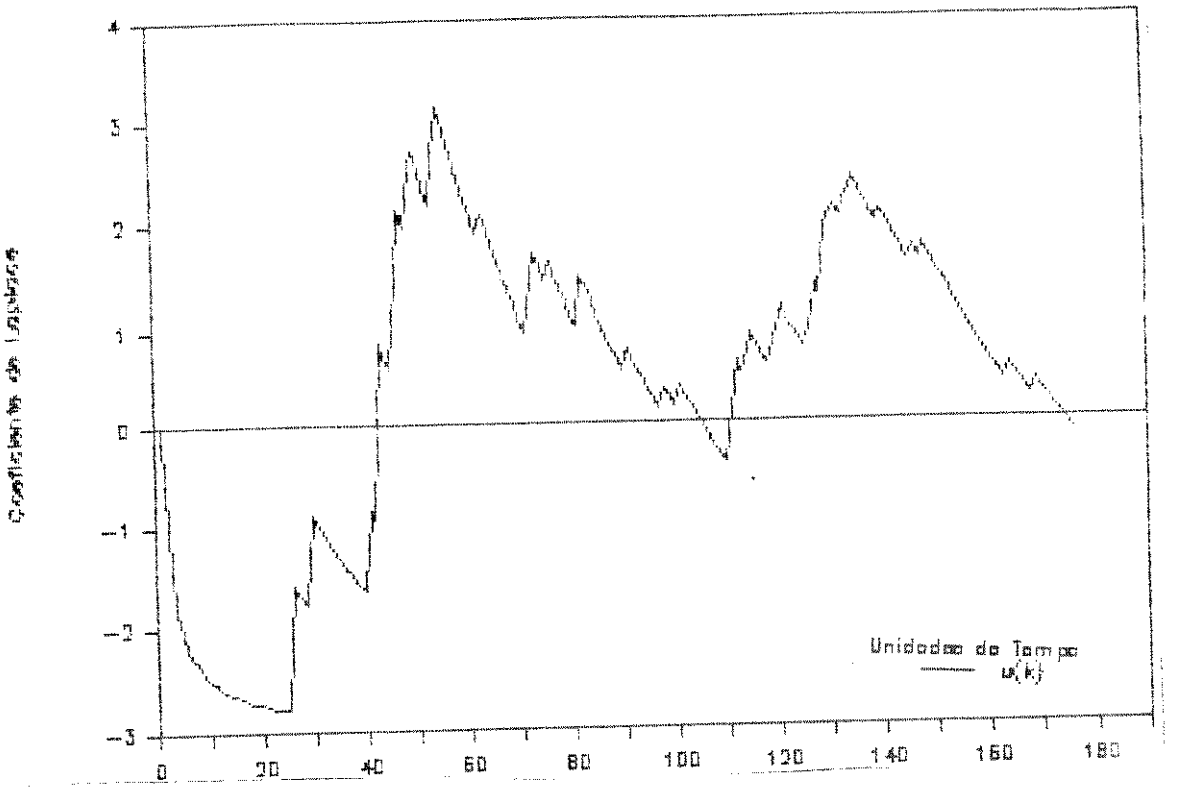

Figura 4.13 - Resultado do Teste de Laplace aplicado à base de dados do Sistema II de Tohma.

Mais uma vez pode-se observar que a partição não coincide obrigatoriamente com as alterações no tipo de testes aplicados . O resultado da modelagem deste processo de falha é mostrado na Figura 4.14. O erro médio cometido é de 0.48 falhas por unidade de tempo.
A partição empírica proposta por Tohma é:
G1: de t=1 a t=41
G2: de t=42 a t=109
G3: de t=110 a t=125
G4: de t=126 a t=185
que se aproxima da obtida pelo teste de tendência, apenas desconsiderando os pontos de inflexão em t=25 e t=70. No artigo foi usado um modelo baseado na distribuição hipergeométrica. Aplicando os modelos usados neste trabalho à partição de Tohma, o melhor resultado obtido corresponde a um erro médio cometido de 1.10 falhas por unidade de tempo, bem maior que para a partição indicada pelo teste de tendência.

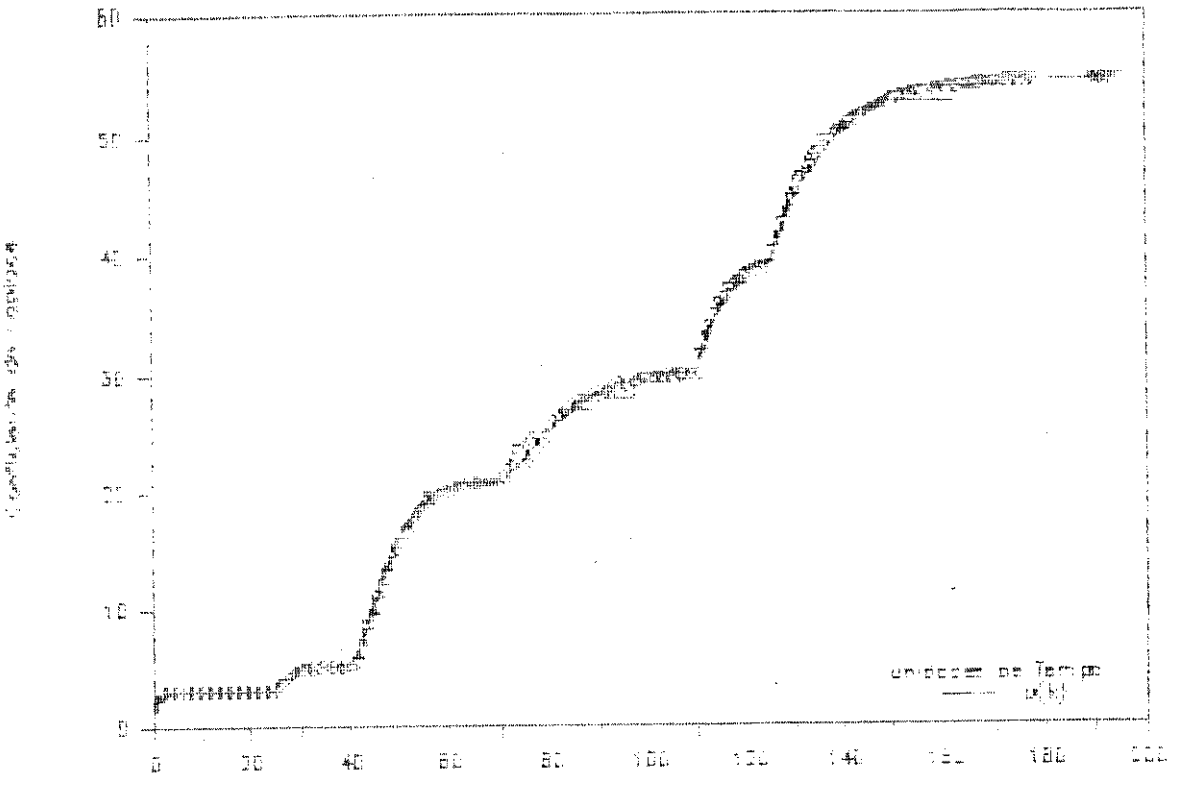

Figura 4.14 - Resultado da modelagem do Sistema II de Tohma segundo o método de partição proposto.

Neste exemplo pode-se perceber uma grande vantagem do método de partição proposto que é o de seguir um critério sistemático, baseado nos resultados do teste de tendência.
Por outro lado, uma desvantagem do método é que em algumas bases de dados o ponto de inflexão não fica muito bem definido, ou seja,não é fácil diferenciar entre um ponto de inflexão e uma oscilação aleatória local dos dados. Isto pode ser exemplificado pelos pontos t=70 e t=125 da base de dados de Tohma. No entanto, se o ponto de inflexão não é tão evidente isto significa que a inversão de tendência não é tão forte e o erro cometido pela não caracterização do ponto de inflexão também é pequeno. No caso do sistema deste exemplo o erro cometido na modelagem sem os pontos de inflexão em t=70 e t=125 é de 0.65 falhas por unidade de tempo.

4.4 Composição dos Processos de Falha Hardware e Software

A concepção de um sistema pode ser dividida em partes hardware e software. A concepção do software refere-se ao projeto e implementação do código e de dados armazenados no sistema. A concepção do hardware está relacionada ao projeto dos itens materiais do

equipamento (placas, bastidores, alimentação, painéis, etc...).

Seja um sistema composto por hardware e software. Como considerar e analisar os dados de falha relativos a cada parte? Como integrar esses resultados a fim de avaliar o sistema como um todo, segundo os requisitos de qualidade de serviço especificados?

Os processos de falha podem ser considerados independentes, uma vez que o hardware e o software são tradicionalmente desenvolvidos em separado, a partir das especificações de suas interfaces. Há no entanto algumas falhas que, embora causadas por um defeito localizado numa das partes, levam a correções em ambas. Para os sistemas usados como exemplo neste trabalho o número dessas falhas é da ordem de 10% do total. Suponha-se, por medida de simplificação, que essas falhas sejam debitadas somente na base de dados de falha da parte que contém o defeito.

A qualificação do projeto de sistema é feita usando os índices de confiabilidade obtidos para cada parte componente, uma vez que elas podem apresentar tendências diferentes.

Exemplo 4.6
Sejam os dados de falha hardware e software do sistema TRÓPICO RA, cujas características estão descritas no Anexo E. O teste de tendência aplicado à base de dados de falha de concepção do hardware fornece os resultados mostrados na Figura 4.15 a seguir. Os dados estão representados na forma do número de falhas acumulado ao final de cada semana de teste. Este sistema, que ainda está em desenvolvimento, tem sido analisado ao final de cada mês a partir da fase de Integração dos módulos hardware/software. A análise do sistema é feita para determinar a situação do sistema ao final do mês de julho de 1990, em meio aos Testes de Sistema.
Até o instante t=28 o sistema apresentou uma confiabilidade estacionária, com uma taxa de falha aproximadamente constante; a partir de t=29 o sistema sofre um decrescimento de confiabilidade, caracterizando um ponto de inflexão do tipo 2. Segundo o método de partição proposto na seção 4.3 o sistema é particionado neste ponto e o teste de tendência aplicado à base de dados restante, ao longo de todo o qual o sistema passa a apresentar crescimento de confiabilidade. A base de dados de falha é então dividida em dois grupos: G1 de t=1 a t=28 e G2 de t=29 até o instante de avaliação. É interessante notar que o segundo grupo de dados corresponde exatamente ao ano de 1990.

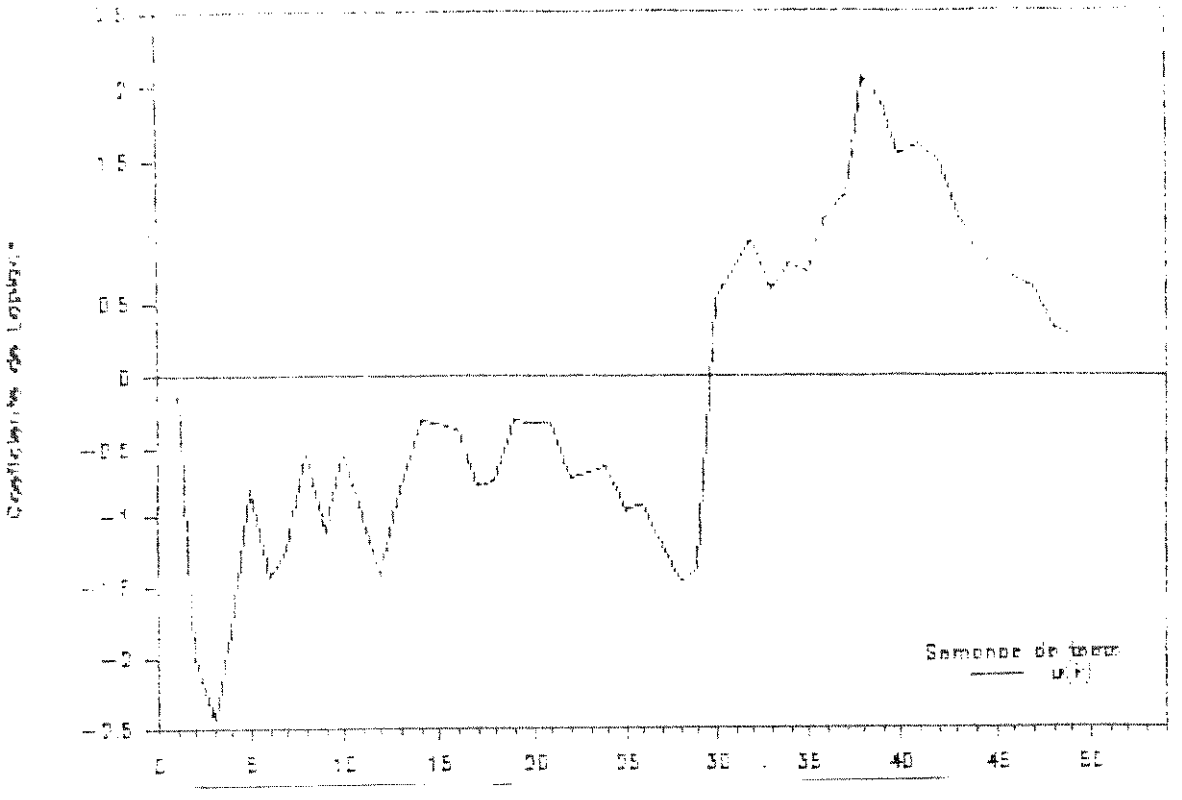

Figura 4.15 - Resultado do Teste de Tendência aplicado à base de dados de falha de concepção do hardware do sistema TRÓPICO RA.

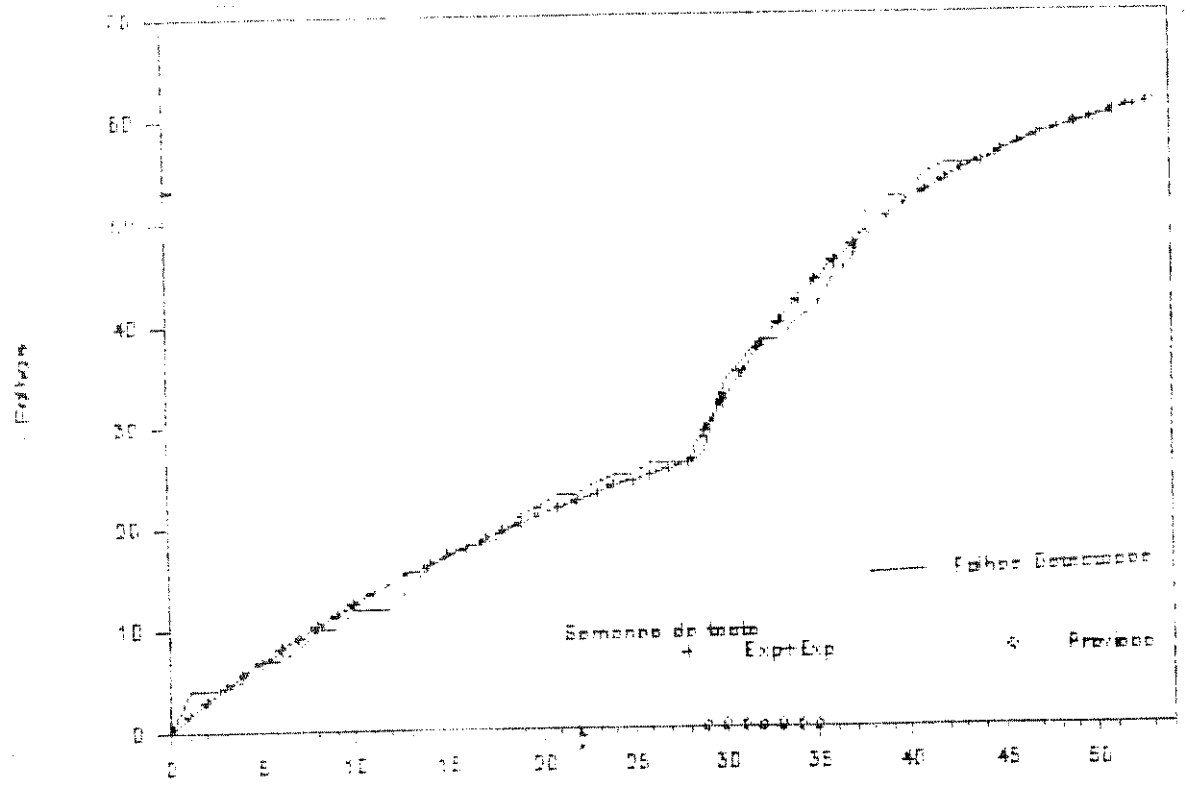

Figura 4.16 - Resultados da modelagem do Hardware do Sistema TRÓPICO RA através do modelo Exponencial.

Como os dois períodos apresentam um crescimento de confiabilidade é aplicado o modelo Exponencial a ambos. Os resultados da modelagem são apresentados na Figura 4.16. O erro médio cometido é de 0.99 falhas por semana. A partir da curva que modela o período G2 foram feitas previsões sobre o número de falhas esperado no próximo mês de teste; a curva de falhas projetada é também mostrada nesta figura. A previsão do número de falhas de concepção do hardware no próximo mês é de 1.94 falhas.

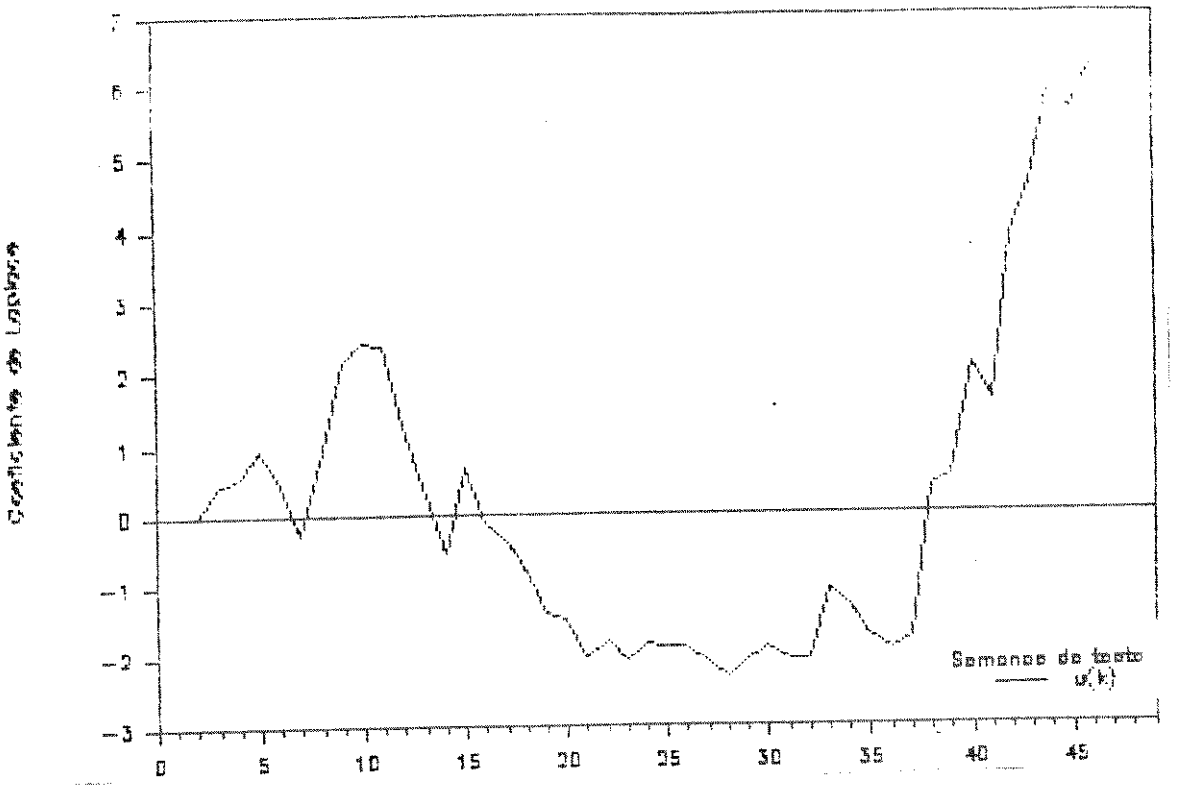

Figura 4.17 - Resultados do Teste de Tendência aplicado à base de dados de falha do software do Sistema TRÓPICO RA.

Quanto ao software do Sistema TRÓPICO RA, os resultados do teste de tendência são os mostrados na Figura 4.17. O sistema não apresentou falhas nas três primeiras semanas de observação. A confiabilidade decresce nas 13 semanas seguintes, passando então a crescer até a semana 36. Neste momento (30/05/90) a confiabilidade volta a crescer, e identifica-se um ponto de inflexão do tipo 2. A base de dados é particionada neste ponto, caracterizando um grupo de dados cujo comportamento de tendência é compatível com a hipótese do modelo Gama. O teste de tendência revela um processo estacionário cuja modelagem pode ser feita através do modelo Exponencial.
O resultado da modelagem do processo de falha software do TRÓPICO RA dividido nos períodos G1 de t=1 a t=36 e G2 de t=42 a t=50 está comparado à curva de falha real na Figura 4.18. A curva de falha estimada para o segundo período é projetada para as quatro semanas seguintes ao momento da avaliação. O erro médio cometido na modelagem é de 3.97 falhas por unidade de tempo.

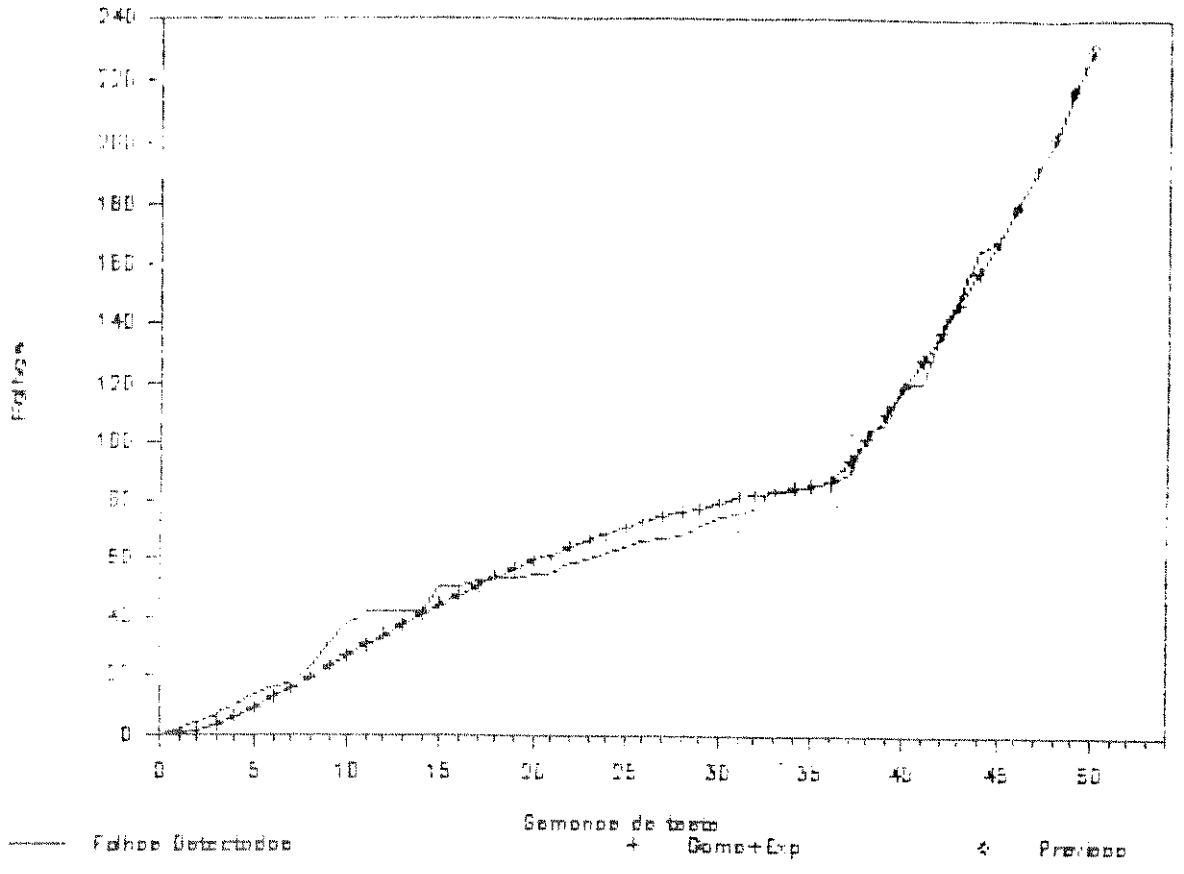

Figura 4.18 - Modelagem do processo de falha software do sistema TRÓPICO RA através do modelo Gama e do modelo Exponencial.

A alta taxa de falha experimentada nos últimos dois meses é fruto da mudança da fase de testes de Integração Hardware/Software para a fase de Testes de Sistema. Como foi comentado na seção anterior, a previsão do número de falhas feita pelo modelo Exponencial, quando o processo de falha é estacionário e o esforço de teste constante, é pessimista. Deste modo a previsão do número de falhas no próximo mês, que é de 50.37, funciona como um limitante superior do número de falhas esperado. Fazendo uma avaliação global do sistema no instante de observação chega-se às seguintes conclusões:

1. O hardware, que teve um processo de falha constante no primeiro meio ano de teste, passou por uma mudança de padrão de falha no início de 1990. A taxa de falha passou a ser bastante elevada em comparação com a do fim de 1989 e exponencialmente decrescente. A tendência do processo de falha não sofreu alteração com o início dos Testes de Sistema. No instante de observação o sistema já apresentava um ligeiro crescimento global de confiabilidade, sugerindo que o produto esteja alcançando um estágio em que (1) os Testes de Sistema não mais sejam eficientes para revelar falhas de concepção hardware ou (2) o sistema atinja a sua maturidade, com uma taxa mínima de falha.

2. O software, que do início do período de observação até o final do mês de maio de 1990 passou por um padrão de decrescimento seguido de crescimento de confiabilidade, tem uma mudança fundamental de comportamento com o início

da fase de Testes de Sistema. No instante de observação o processo de falha é estacionário com uma taxa de falha bastante elevada. Analisando estado corrente do sistema pode-se dizer que ele se encontra em uma fase de intensa depuração. Os testes de Sistema devem ser continuados até que a tendência observada seja de crescimento de confiabilidade.

Dos resultados mostrados neste exemplo vê-se que:

- o processo de falha hardware segue os mesmos padrões de crescimento/decrescimento de confiabilidade encontrado nos exemplos de sistemas software já mostrados neste trabalho e foi satisfatoriamente representado pelos modelos desenvolvidos para o software,
- o erro médio cometido na modelagem do processo de falha do hardware é comparável ao erro cometido na modelagem do processo de falha software,
- a avaliação dos processos de falha hardware e software deve ser feita separadamente, uma vez que esses dois sub-sistemas podem apresentar comportamentos completamente diferentes,
- a avaliação global do sistema é uma composição do estado dos dois sub-sistemas.

Os resultados aqui apresentados animam a insistir na utilização dos métodos de análise e modelagem desenvolvidos para o software ao estudo de processos de falha de concepção do hardware, embora de maneira independente.

4.5 Diretrizes Para Aplicação da Metodologia

A aplicação de modelos de crescimento de confiabilidade de forma coerente com suas hipóteses básicas fornece resultados melhores do que a modelagem feita sem levar em conta as características do processo de falha estudado, como foi mostrado nos exemplos deste capítulo. A metodologia aqui proposta baseia-se na análise do comportamento da base de dados a ser modelada a fim de respeitar as hipóteses básicas dos modelos.

A metodologia proposta é geral e se aplica também à modelagem de processos de falha de concepção hardware. Propõe-se que os processos de falha de concepção software e hardware sejam analisados separadamente e que o sistema seja qualificado quanto a

estes dois processos.

A metodologia proposta para a modelagem de processos de falha de concepção baseada no teste de tendência que foi apresentada nas seções anteriores pode se resumir nas seguintes diretrizes:

1. Aplicar o Teste de Tendência de Laplace a toda a base de dados de falhas detectadas;
2. Identificar os períodos de crescimento de confiabilidade, os períodos de confiabilidade estacionária e os períodos de decrescimento de confiabilidade;
3. Se o primeiro período apresentar uma tendência a crescimento de confiabilidade ou confiabilidade estacionária, então aplicar um modelo capaz de representar este comportamento a todo o período em que as hipótes dos modelos são verificadas;
4. Se for observado um decrescimento de confiabilidade ou estacionariedade, sem crescimento posterior de confiabilidade pode-se aplicar um modelo capaz de representar este comportamento. No entanto, deve-se considerar que os resultados das medidas de segurança de funcionamento obtidas a partir do modelo calibrado serão pessimistas, tendo em conta a hipótese fundamental de crescimento de confiabilidade dos processos de falha de concepção. Deve-se esperar por um período de crescimento de confiabilidade para uma avaliação mais realista.
5. Se for observado um decrescimento de confiabilidade seguido de crescimento, então aplicar um modelo capaz de representar este comportamento. Usar os dados de falha correspondentes aos períodos de decrescimento e de crescimento de confiabilidade. Este período se encerra com a ocorrência de um ponto de inflexão do tipo dois;
6. Para os períodos restantes proceder da mesma maneira exposta nos passos de 1 a 4.

4.6 Conclusão

A metodologia de aplicação dos modelos de confiabilidade que é proposta tem as seguintes vantagens:

1. Possibilita a aplicação dos modelos de forma coerente com as suas hipóteses diminuindo o erro de modelagem e evitando conclusões errôneas na comparação da eficiência dos modelos.

2. Permite a modelagem de sistemas complexos.

3. Identifica os impactos das alterações no sistema e em sua solicitação no processo de falha, permitindo melhor compreensão da sensibilidade do sistema a essas perturbações.

4. Prescinde de hipóteses feitas a priori sobre o tipo de solicitação, dependência entre falhas e impacto de alterações. Ao contrário, a partir da observação dos impactos sofridos pela curva de falha é que são identificadas as perturbações efetivas.

5. É extensiva ao estudo de falhas de concepção de hardware.

A modelagem de um processo de falha permite a obtenção de medidas que permitam qualificar o sistema quanto a seus requisitos de segurança de funcionamento.

O acompanhamento da evolução de um sistema ao longo de seu ciclo de desenvolvimento e vida operacional fornece indicadores úteis tanto ao usuário quanto ao projetista do sistema. O Capítulo 5 mostra como a metodologia de aplicação dos modelos proposta pode auxiliar na avaliação de sistemas através de experiências reais.

Capítulo 5

Qualificação da Concepção de Sistemas

5.1 Introdução

Em pontos-chaves do ciclo de vida de um sistema é necessário fazer a sua qualificação. Por qualificação entende-se a avaliação do sistema quanto a seus reqüisitos de qualidade. No contexto de segurança de funcionamento estes requisitos são expressos através das medidas descritas no Capítulo 1.

O acompanhamento da segurança de funcionamento ao longo do ciclo de vida permite, mais do que qualificar o produto, avaliar a eficiência do seu processo de desenvolvimento. Neste capítulo mostra-se como a metodologia proposta no Capítulo 4 pode auxiliar em alguns aspectos da qualificação do sistema e na orientação do seu desenvolvimento.

Este capítulo está dividido em duas partes: nas seções 5.2 a 5.5 são discutidas algumas propostas para a qualificação de sistemas, enquanto que nas seções 5.5 e 5.6 são apresentados estudos de casos de sistemas reais, aos quais se aplicam a metodologia descrita no Capítulo 4 a fim de qualificar o sistema ao longo de seu ciclo de vida, empregando as idéias contidas neste capítulo.

A qualificação de um sistema quanto a segurança de funcionamento pode ser feita em qualquer momento de suas fases de teste e vida operacional. Alguns problemas de modelagem podem decorrer das características que o processo de falha apresenta no momento em que se faz a qualificação, como é discutido na seção 5.2.

Durante as etapas de teste a preocupação fundamental está relacionada à depuração do sistema, atingindo índices de segurança de funcionamento intermediários, desde que sejam cumpridos os prazos estabelecidos para a atividade. O compromisso qualidade x prazos pode ser gerenciado através do desempenho mais eficiente da atividade de teste, como é visto na seção 5.3. Da mesma forma, nesta fase é necessário determinar qual o nível de qualidade que o sistema atingirá no próximo marco de qualificação do sistema, como é mostrado na seção 5.4. Se as previsões não forem satisfatórias deve-se modificar os prazos, a eficiência dos testes ou as metas a serem atingidas.

Um problema que afeta tanto o usuário quanto o projetista de sistemas em vida operacional é que algumas vezes são necessárias alterações nas especificações do sistema a fim de incorporar novas funções ou melhorias. Estas mudanças implicam na introdução de novos defeitos e algumas vezes na ativação de defeitos até então latentes no sistema. O impacto que estas modificações no sistema podem causar no processo de falha pode ser avaliado, como é mostrado na seção 5.5.

Nas seções 5.6 e 5.7 são apresentados dois estudos de casos reais, em que a metodologia de aplicação proposta é utilizada na qualificação de sistemas. O primeiro caso ressalta as dificuldades da análise em tempo real, e a evolução das informações ao longo do acompanhamento. O segundo caso mostra como a avaliação da confiabilidade de concepção do hardware e do software pode ser considerada, juntamente com os resultados de confiabilidade de componentes hardware, na validação de sistemas quanto a seus requisitos de qualidade de serviço.

5.2 Qualificação Quanto a Segurança de Funcionamento ao Longo do Ciclo de Vida

As análises de confiabilidade de sistemas reais encontradas na literatura tem basicamente o seguinte objetivo:

Análise Post Mortem: é apresentada uma base de dados completa de um sistema já totalmente desenvolvido e em operação comercial há bastante tempo e este processo de

falha é estudado com o fim de compreender sua natureza e/ou analisar o desempenho de modelos de confiabilidade. Neste caso o sistema se encontra geralmente em uma fase de crescimento de confiabilidade.

Neste trabalho é proposto um outro ponto de vista:
Análise ao Longo do Ciclo de Vida: os dados de falha de um sistema em teste ou operação comercial são estudados com a finalidade de se obter os índices de segurança de funcionamento atuais ou futuros (previsões). Esta análise é geralmente feita nos momentos em que o sistema muda de responsável ou em preparação para estes momentos.

Dentro deste contexto podem ser definidos vários marcos de qualificação. Ao final do desenvolvimento, quando o sistema está apto a operar comercialmente, deve ser feita a qualificação a fim de comprovar que o sistema cumpre os requisitos de segurança de funcionamento especificados. Durante a vida operacional podem haver outros marcos de qualificação, geralmente solicitada pelo cliente, a fim de avaliar o nível de qualidade do produto adquirido. Nas fases de teste, podem ser definidos índices intermediários de segurança de funcionamento a serem perseguidos nas atividades de teste.

Seguindo a metodologia de avaliação apresentada, a base de dados de falha de um sistema deve ser analisada pelo teste de tendência e particionada de acordo com estes resultados. Para a qualificação do sistema deve ser empregado somente o último grupo de dados. Este é o período que termina no instante em que é feita a qualificação. Este processo pode se encontrar nas seguintes condições de tendência no momento de qualificação:

Crescimento de confiabilidade: este é o estado mais favorável à avaliação. Quando o sistema tem um processo de falha com este padrão os indicadores de confiabilidade e os resultados das previsões são geralmente bons (próximos dos reais) desde que não haja inversão da tendência no intervalo para o qual são feitas as previsões.

Confiabilidade Estacionária: como foi comentado na seção 4.2, este comportamento em geral pode ser modelado através de um modelo exponencial com baixos resíduos de falha, se o objetivo for calcular grandezas locais, como intensidade de falhas. As pre-

visões de grandezas futuras, como número de falhas residuais, no entanto, será sempre pessimista, uma vez que a característica deste modelo é manter a tendência corrente de estacionariedade. O pessimismo decorre da hipótese básica sobre a confiabilidade de concepção, de que, mais cedo ou mais tarde ela crescerá.
O máximo da função de verossimilhança para modelos Exponenciais aplicados a bases de dados estacionárias pode ser, em alguns casos, atingido somente com os valores limites de seus parâmetros ($\beta_0 \rightarrow \infty$ e $\beta_1 \rightarrow 0$). Este fenômeno foi descrito por Littlewood [LIT 81] e é uma outra consequência do emprego deste modelo fora de sua hipótese básica de utilização.

Decrescimento de Confiabilidade: um processo neste estado pode ser modelado pelo modelo Exponencial com baixos resíduos. O significado físico de seus parâmetros, porém, se perde, podendo mesmo atingir valores negativos. Os índices de segurança de funcionamento presentes e futuros também perdem sua significação, e a modelagem passa a ser um mero ajuste de curvas. Nesta situação o mais prudente é monitorar a tendência do processo, aguardando uma reversão da tendência a decrescimento para avaliação do sistema.

Oscilação de Confiabilidade: como já foi comentado na seção 4.2, este é um comportamento difícil de ser modelado qualquer que seja o modelo aplicado.

Os resultados da avaliação, quaisquer sejam as características do processo de falha no momento da qualificação, devem ser interpretados à luz das condições de utilização do sistema.

5.3 Previsão da Segurança de Funcionamento

Durante as fases de teste é necessário administrar os prazos e a qualidade do sistema a fim de obter os melhores resultados possíveis dentro das restrições de tempo impostas. Ao final das atividades de teste devem ser fixados marcos em que deve ser feita a qualificação do sistema e estabelecidas metas intermediárias a serem alcançadas nestes marcos. As metas intermediárias podem ser estabelecidas em função das medidas tradicionais de se-

gurança de funcionamento ou, de maneira simplificada, em relação à intensidade de falha do processo. Dado um processo de falha, é possível fazer previsões sobre as condições em que o sistema atingirá o próximo marco de qualificação de modo a orientar a eficiência da atividade de teste.

Para a previsão deve ser utilizado somente o último período em que foi decomposto o processo de falha utilizando-se o método proposto. O modelo de confiabilidade adequado deve ser aplicado a este período. A expressão calibrada do modelo deve ser utilizada para extrapolar a função valor médio do número de falhas até o próximo ponto de qualificação. As previsões para as medidas de segurança de funcionamento são então feitas e comparadas às metas fixadas.

As dificuldades de modelagem de um processo estacionário ou decrescente em confiabilidade são as mesmas apresentadas na seção anterior, acrescidas de problemas com as hipóteses dos modelos, como é mostrado no exemplo 5.1.

Exemplo 5.1
Sejam os sistemas apresentados em [MAT 89], cujas curvas de tendência já foram estudadas no exemplo 4.3. O objetivo dos autores no artigo é comparar a capacidade preditiva de alguns modelos, entre eles os dois modelos usados neste trabalho. O critério de erro adotado é

$$E = |\frac{N_0 - N}{N_0}|$$

onde N é o número de falhas total no sistema (suposto conhecido) e N_0 é o número de falhas total estimado pelo modelo. Na comparação entre os resultados fornecidos pelos modelos Gama [YAM 83a], S-Inflection [OHB 84a] e Exponencial [GOE 79] para estes sistemas, este último apresenta sempre os piores resultados. Para alguns desses sistemas o critério de erro é favorável a este modelo, porém as previsões não são boas, uma vez que não está sendo usado de forma adequada, ou seja, para a estimação de grandezas locais. Os outros dois modelos apresentam resultados melhores em relação ao critério adotado pois forçam um crescimento de confiabilidade ao final do período de observação mesmo que isto não seja verificado na base de dados. Além disso, eles também não estão sendo empregados, para estes sistemas, dentro de suas hipóteses básicas de utilização. O modelo Exponencial fornece também resultados piores que os outros dois modelos na estimativa do número de falhas residuais do sistema X134, cuja análise de tendência mostra um decrescimento inicial de confiabilidade seguido de crescimento, que aconselham o uso de modelos em forma de S.
Como se pode ver neste exemplo o emprego de modelos de crescimento de confiabilidade a processos de falha com características de tendência incompatíveis com as hipóteses do processo estudado pode até mesmo levar a conclusões equivocadas,

como nesse trabalho, que indica que os modelos em forma de S são "superiores" ao modelo Exponencial.

5.4 Eficiência da Atividade de Teste

Os testes a serem aplicados a um sistema são geralmente organizados em grupos, também chamados de *baterias de teste*. As baterias de teste se diferenciam entre si pelas funções testadas e pelo contexto do sistema em que eles são aplicados (módulo, conjunto de módulos, sistema,etc..).

Ao se aplicar uma nova bateria de testes em geral se observa um aumento da taxa de falha. Este comportamento está ligado à evidenciação de um novo tipo de defeitos que as baterias de teste anteriores não foram capazes de ativar. À medida que estas falhas vão sendo detectadas e os defeitos vão sendo corrigidos a taxa de falha volta a decrescer. Desta forma, um decrescimento de confiabilidade em fase de testes indica normalmente que a bateria de testes está sendo eficiente na ativação de falhas. Um crescimento de confiabilidade nesta fase, por sua vez, indica que a forma de testar já não está sendo tão eficiente.
Dado o objetivo de chegar ao fim da atividade de teste com o sistema com o menor número possível de defeitos residuais, a orientação que o teste de tendência pode sugerir nesta fase é continuar aplicando os testes de uma dada bateria enquanto o teste de tendência resultar em decrescimento de confiabilidade e passar a outra bateria de testes a partir do momento que o teste de tendência indicar um crescimento de confiabilidade.

Exemplo 5.2
Considere-se o software do sistema SAMSAT, cujas características de interesse para este estudo estão descritas no Anexo E.
O teste de tendência dos dados de falha observadas durante todo o período de observação estão mostrados na Figura 5.1. As linhas verticais indicam os períodos relativos aos capítulos de teste aplicados, cujas funções testadas foram, respectivamente:
Capítulo 4: Comandos operacionais e de consulta,
Capítulo 5: Comunicação entre estações,
Capítulo 6: Mecanismos de supervisão,
Capítulo 7: Verificação de configuração do sistema após uma reconfiguração da estação remota de referência (ER0),
Capítulo 8: Tráfego e integridade de dados.

O sistema apresenta crescimento de confiabilidade até t=5, passando por um comportamento bastante instável até t=54, e crescimento dali em diante. De acordo com o método de partição segundo o teste de tendência a base de dados foi dividida nos seguintes períodos: G1 de t=1 a t=5, G2 de t=6 a t=15, G3 de t=16 a t=26, G4 de t=27 a t=47 e G5 de t=48 a t=80.
No instante t=80, um ponto intermediário do prazo de testes, foi feita uma análise de confiabilidade com a finalidade de obter informações sobre a eficiência da atividade de teste [HOL 90]. Esta análise tinha o objetivo de orientar os testes a serem aplicados no tempo que restava para esta atividade.

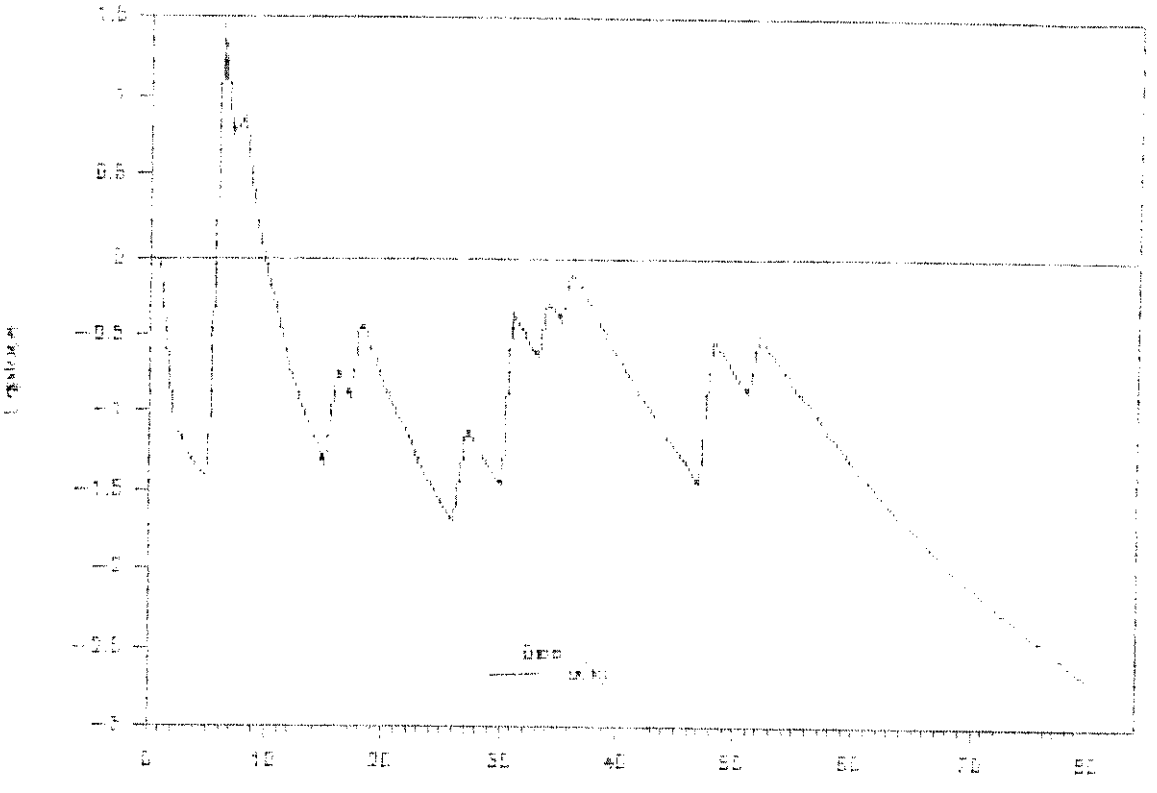

Figura 5.1 - Teste de tendência dos dados de falha do software do sistema SAMSAT.

O resultado da modelagem do sistema é visto na Figura 5.2, onde o modelo Gama foi aplicado ao período G4 e o modelo Exponencial aos demais períodos. O erro cometido na modelagem é de 0.07 falhas por unidade de tempo.
Ao estudar os resultados do teste de tendência em relação aos capítulos de teste pode-se fazer as seguintes observações:
a) os capítulos 4 e 6 revelaram as maiores concentrações de falha. As funções a que se referem estes capítulos são as que mais exigem do software,
b) durante a aplicação dos testes dos capítulos 2,5 e 8 não foi detectada nenhuma falha. A bateria de testes do capítulo 8 se caracterizou pelo funcionamento livre de interferência humana do sistema em laboratório.
c) o capítulo 6 foi o que mais propiciou a depuração do software. Além disso, ao final da aplicação dos testes deste capítulo a bateria ainda se mostrava eficiente.
A relação entre o tempo de teste e o número de falhas detectadas neste sistema nos capítulos 6 e 8, no entanto, assume significados diversos devido à diferença do esforço de teste nesses dois períodos. Na execução dos casos de teste do capítulo 6 era permitida a injeção de defeitos no sistema, o que corresponde a um teste de estresse, ou de

aceleração de vida útil. A execução do capítulo 8 de testes foi feita sem interferência no sistema, que operava em ambiente controlado de laboratório com baixa carga de tráfego, ou seja, em condições bem favoráveis de operação. Estas condições foram determinadas pelas dificuldades técnicas de simulação de tráfego intenso e da relação sinal/ruido degradada, etc..., as quais associaram aos testes um papel de observação normal de operação. Neste caso, o decrescimento de confiabilidade a partir de t=25 indica uma eficácia dos testes do capítulo 6 na ativação de defeitos latentes no software e o crescimento de confiabilidade a partir de t=53 indica que o tipo de testes contidos no capítulo 8 não estava sendo eficiente na ativação de falhas.

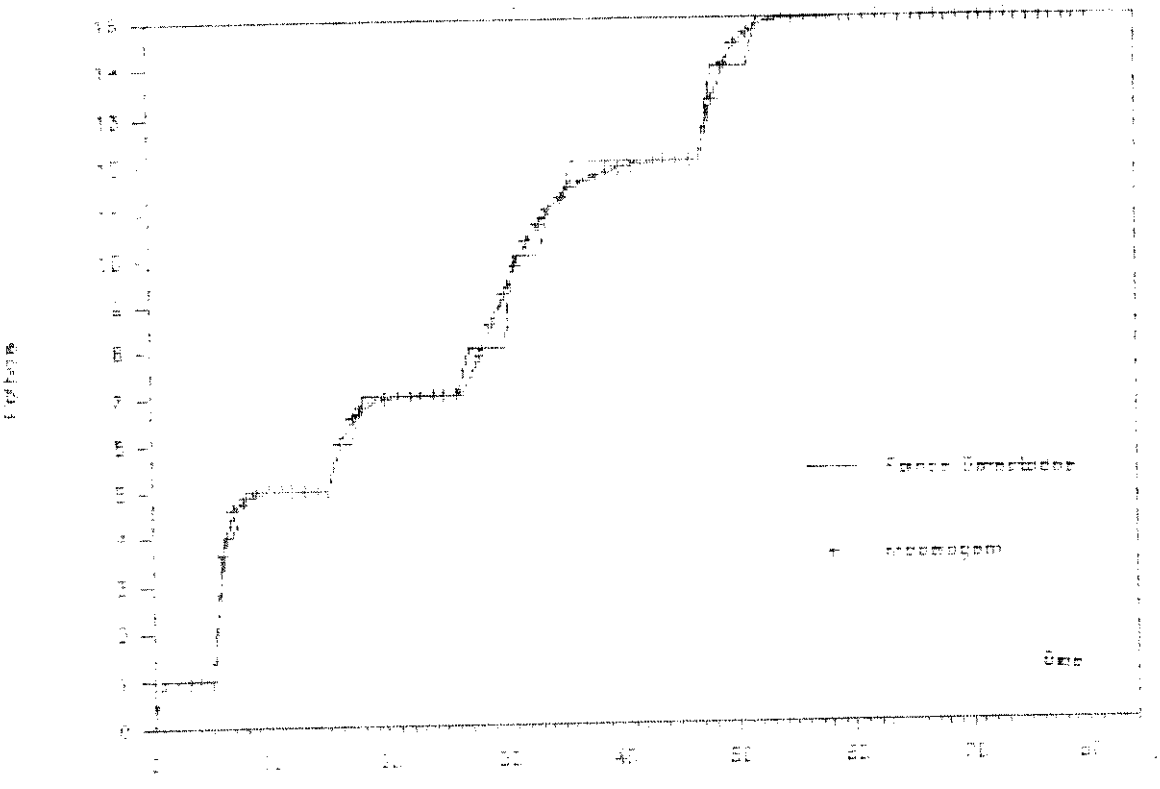

Figura 5.2 - Modelagem do processo de falha do software do sistema SAMSAT.

A conclusão global do estudo em t=80 indicou que o melhor investimento de tempo na depuração do sistema seria descontinuar os testes de Tráfego e Integridade de Dados e aplicar testes suplementares aos Mecanismos de Supervisão.

5.5 Impactos de Modificações no Processo de Falha

Algumas vezes é necessário fazer modificações nas especificações do sistema em plena vida operacional. Estas alterações podem ser caracterizadas pela inclusão de novas funções ou melhorias, incorporação de novas técnicas, como tolerância a defeitos e cobertura de falhas não previstas nas especificações originais, porém exigidas pelo usuário.

Estas modificações implicam em código e/ou hardware adicionais, que contém novos defeitos de concepção. Estes defeitos vêm se somar aos já latentes no sistema. O aumento do número de defeitos pode causar a elevação da taxa de falha do sistema. O impacto das modificações pode ser medido pela duração e pela amplitude desta elevação da taxa de falha.

Uma medida empírica da amplitude do efeito das alterações no projeto pode ser dada pela relação entre a intensidade de falha no instante posterior à perturbação tal que u(k) é máximo e o instante imediatamente anterior à perturbação.

A duração da perturbação pode ser entendida de várias maneiras. De acordo com o objetivo desta avaliação, algumas delas são:

1. o tempo que o processo de falha leva para atingir a intensidade de falha prevista para aquele instante nas condições anteriores à perturbação. Para isto é feita uma projeção da intensidade de falha através dos dados anteriores à perturbação e esta é comparada, a cada unidade de tempo, com a intensidade de falha estimada a partir dos dados de falha posteriores à perturbação. É possível, por este critério, que um processo de falha nunca se recupere de uma perturbação sofrida.

2. o tempo que o processo de falha leva para atingir a intensidade de falha no momento em que ocorreu a perturbação. Este critério é menos severo que o anterior.

3. o tempo que o sistema leva para atingir nova condição de crescimento de confiabilidade, independentemente do novo valor da intensidade de falha. Este critério é ainda menos severo que os outros dois.

A comparação da intensidade de falha pode também ser usada para avaliar o efeito de novas tecnologias, revisões, etc...

Exemplo 5.3
Seja a curva de falha do Ambiente de Programação CHILL do CPqD-TELEBRAS

(APCC-TELEBRÁS) (ver Anexo E). O resultado da aplicação do Teste de Laplace é mostrado na Figura 5.3.

Em t=80 a curva de falha, que vinha apresentando um crescimento de confiabilidade, sofre um impacto causado pela liberação de uma nova versão de um dos principais módulos do sistema (Depurador CHILL).Esta perturbação pode ser avaliada particionando os dados de falha em t=80 e analisando os dados restantes e exemplificar os dois métodos de análise de perturbações.

Seguindo o método de partição proposto neste trabalho a base de dados foi particionada em vários subgrupos, como é mostrado detalhadamente na seção 5.6. Para o estudo da perturbação causada pela nova versão do Depurador, no entanto, é suficiente considerar os dois grupos de dados definidos pelo ponto de inflexão localizado em t=79: G3 de t=35 a t=79 e G4 de t=80 a t=117. A intensidade de falha em t=79, estimada através da aplicação do modelo Exponencial a G3 é de 1.78 falhas.

A proposta é de que a intensidade da perturbação seja medida pela relação entre a intensidade de falha no momento em que ela é máxima (t=85) e a intensidade de falha em t=79. A intensidade de falha em t=85 estimada pelo modelo Exponencial aplicado de t=80 a t=85 (região de decrescimento de confiabilidade) é h(t=85)= 6.75. A introdução da nova versão do Depurador multiplicou a intensidade de falha em até 3.8.

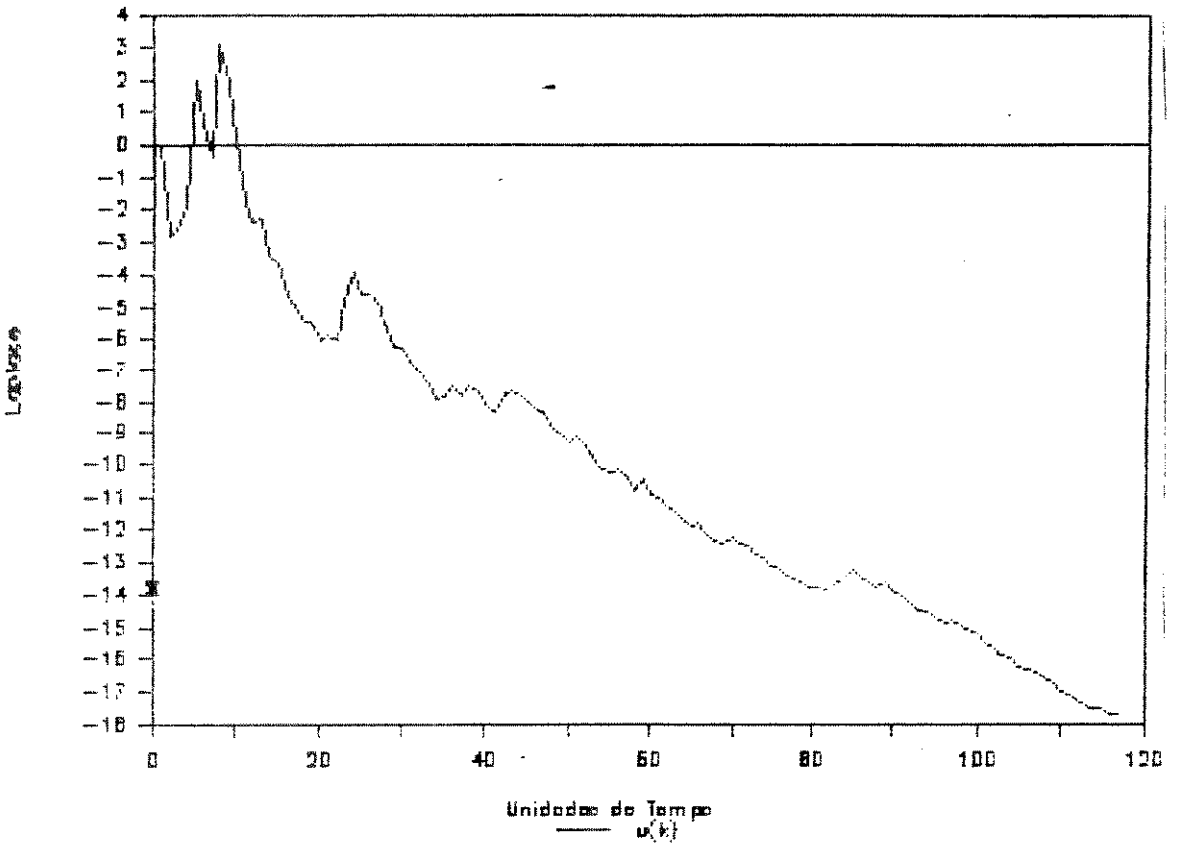

Figura 5.3 - Resultado do Teste de Laplace aplicado ao APCC.

A duração da perturbação será avaliada através dos três métodos sugeridos. Pelo primeiro método estima-se a curva de falhas a partir dos dados até o momento da perturbação, projeta-se o comportamento e obtém-se os valores de h(t) mostrados

na segunda coluna da Tabela 5.1 sob o título de projeção de h(t) a longo termo. Estes valores são comparados com os obtidos modelando a curva de falha a partir dos dados observados em G4, e que estão mostrados na terceira coluna da tabela sob o título de h(t) estimada passo a passo.

Pontos de Avaliação	h(t) Projetada	h(t) Estimada
83	1.57	
86	1.43	
89	1.31	2.73
92	1.19	1.32
95	1.09	1.08
98	0.99	1.34
101	0.91	0.91
104	0.83	0.84
107	0.75	0.78
114	0.63	0.56
117	0.58	0.72

Tabela 5.1 - Evolução da taxa de falha estimada a cada unidade de tempo.

A partir de t=101 o valor estimado pela aplicação do modelo passa a ser, se não menor, pelo menos comparável ao valor de $h(t)$ projetado a partir dos parâmetros estimados em t=79. Por este método, então pode-se dizer que a perturbação teve uma duração aproximada de sete meses.
Aplicando o segundo método de avaliação da duração da perturbação, nota-se que a partir de t=90 a intensidade de falha estimada passa a ser menor que a intensidade de falha anterior ao ao início da perturbação: h(t=79)=1.78. Vê-se que a duração da perturbação por este critério é de aproximadamente quatro meses.
A aplicação do terceiro método consiste simplesmente na observação da curva de tendência mostrada na Figura 5.3. A tendência volta a ser de crescimento de confiabilidade a partir de t=85. Por este critério a duração da perturbação é de aproximadamente dois meses.
A impressão da equipe responsável por este projeto sobre a duração da perturbação é de que ela foi de pouco mais de um mês, o que se aproxima do resultado obtido a partir do terceiro método.

Os três métodos de avaliação da duração de uma perturbação na curva de falha aqui propostos se diferenciam entre si pela interpretação do que caracteriza o retorno do sistema a uma condição anterior.

5.6 Avaliação do Ambiente de Programação CHILL do CPqD

O acompanhamento da confiabilidade deste sistema software foi feito mensalmente a partir da unidade de tempo t=71. Ao final de cada mês as novas falhas detectadas eram

agrupadas em três unidades de tempo e era gerado um Relatório de Acompanhamento de Confiabilidade (RAC) nos moldes do apresentado no Anexo B. Estes relatórios serviam à equipe de desenvolvimento, que controlava a qualidade do sistema e o esforço de manutenção demandado. O acompanhamento era feito sobre o Sistema Fonte.

O resultado da aplicação do teste de tendência à base de dados completa disponível no primeiro Relatório de Acompanhamento de Confiabilidade (RAC-01) é dado na Figura 5.4. Até t=3 a confiabilidade cresce, passando a decrescer até t=8, voltando a crescer até t=21. Os três primeiros dados foram desprezados, uma vez que um período tão pequeno quanto esse não é passível de ser modelado. O teste de tendência aplicado à base de dados a partir de t=3 é mostrado na Figura 5.5.

A primeira partição foi feita em t=21, onde ocorreu o segundo ponto de inflexão do tipo dois, dando origem ao primeiro grupo de dados G1 de t=3 a t=21, cujo perfil de tendência indica a aplicação do modelo Gama. Um novo teste de tendência foi aplicado à base de dados restante, cujo resultado é mostrado na Figura 5.6.

O sistema passa por um período de crescimento de confiabilidade até t=34, onde ocorre um novo ponto de inflexão do tipo dois. É definida uma nova partição: G2 de t=22 a t=34, cujo perfil de tendência indica a aplicação do modelo Exponencial. A seguir aplica-se novamente o teste de tendência à base de dados restante e o resultado é mostrado na Figura 5.7.

A tendência a partir de t=35 é de crescimento de confiabilidade até o instante de avaliação t=71. Foi aplicado o modelo Exponencial a este período e, a partir dos resultados, foi feita uma previsão para o próximo mês. O resultado da modelagem da curva de falha completa de t=1 a t=71 é mostrado na Figura 5.8. O erro médio cometido na modelagem é de 3.82 falhas por unidade de tempo e e o número de falhas previsto para o mês seguinte é de 6.5 falhas.

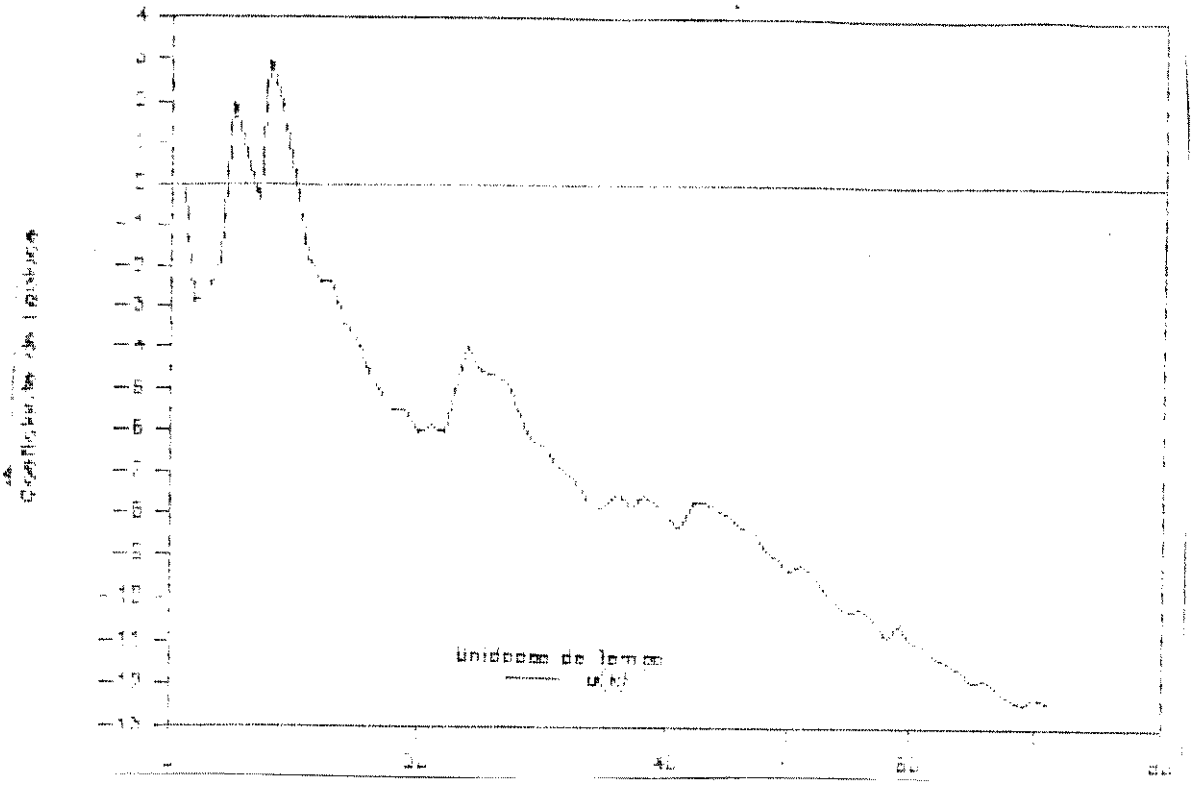

Figura 5.4 - Teste de tendência aplicado à base de dados de falha do projeto APCC disponível no primeiro ponto de avaliação do sistema.

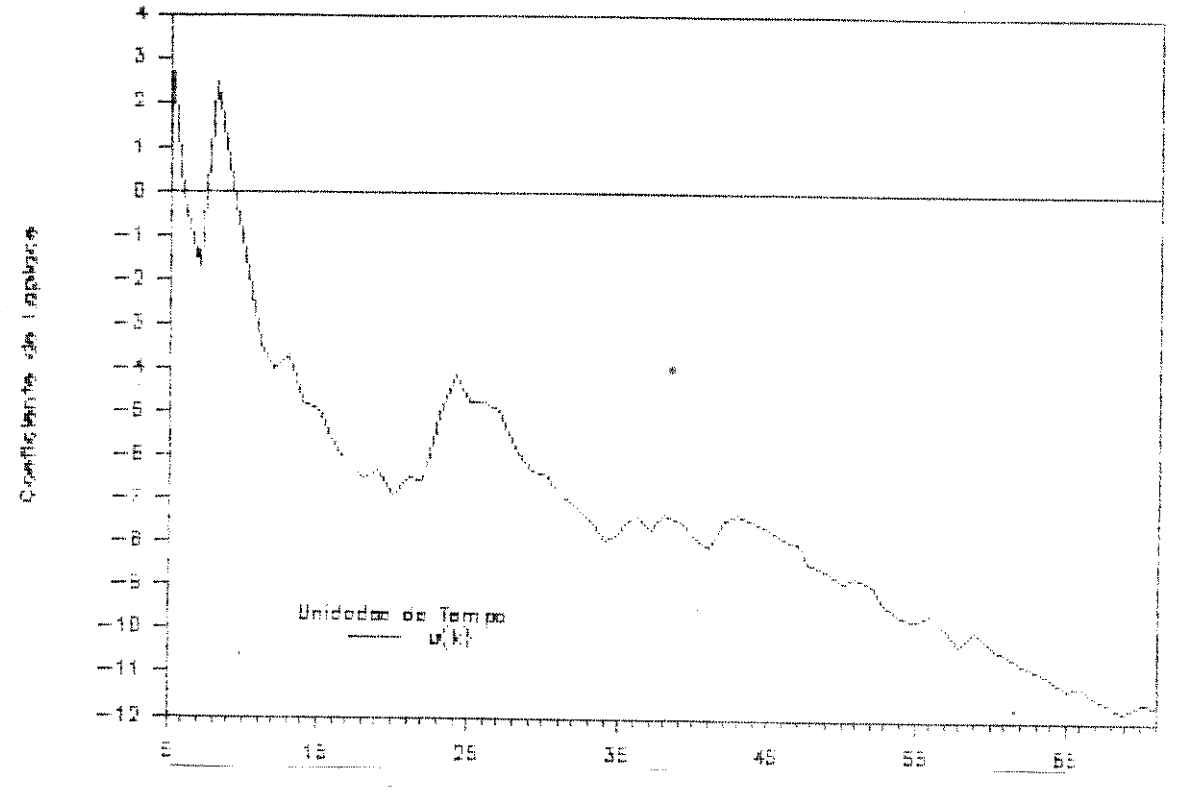

Figura 5.5 - Teste de tendência aplicado à base de dados de falha do projeto de t=3 a t=71.

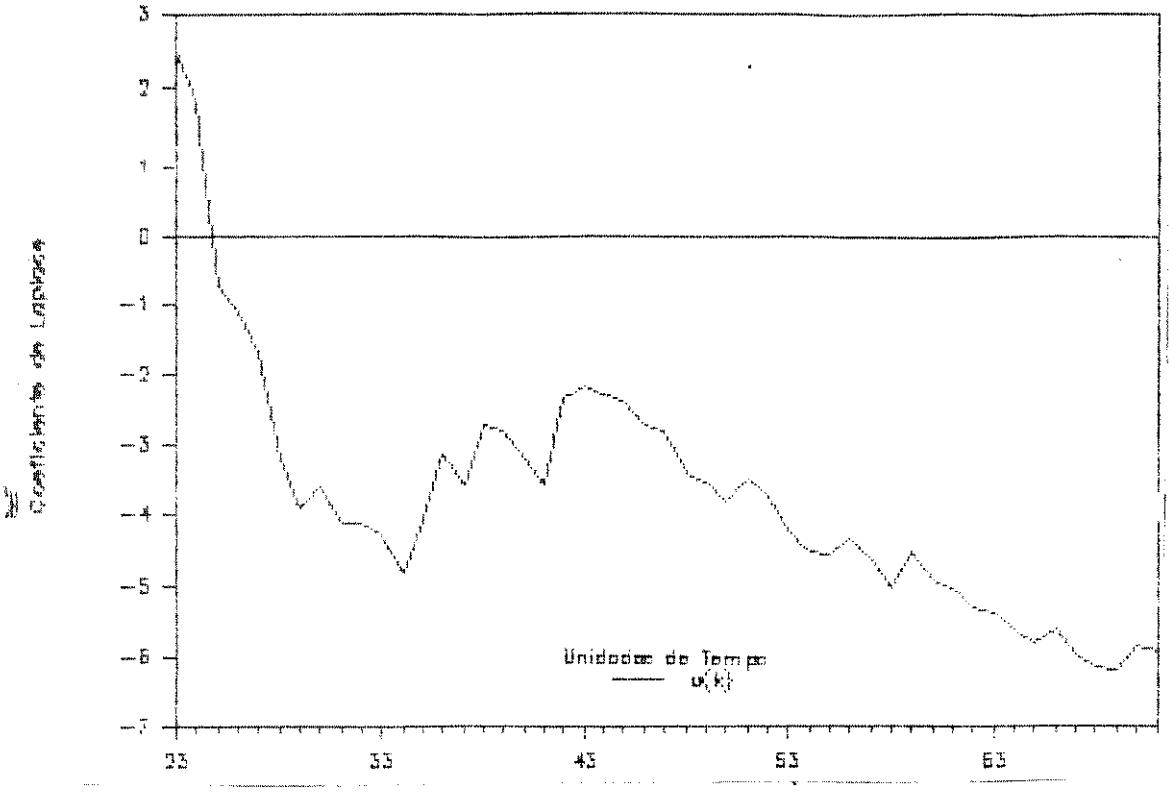

Figura 5.6 - Teste de tendência aplicado à base de dados de falha do projeto APCC de t=22 a t=71.

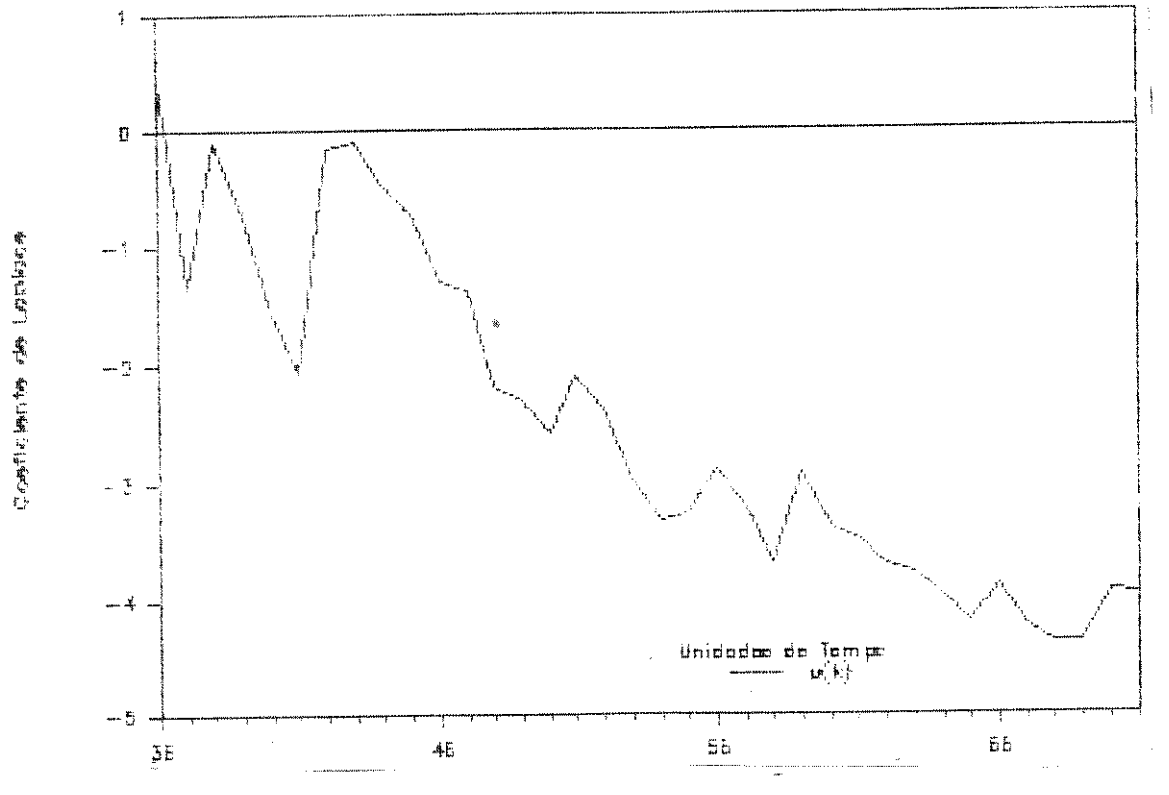

Figura 5.7 - Teste de tendência aplicado à base de dados de falha do APCC de t=35 a t=71.

Em t=74 e t=77 foram feitas novas avaliações do sistema, agora somente com a base de dados G3, uma vez que as anteriores não sofreram alterações. A tendência permaneceu sendo de crescimento de confiabilidade e o modelo Exponencial forneceu em t=74 um erro de 3.03 e previsão de 6.04 e em t=77 um erro de 2.68 e previsão de 5.33.

Em t=80 notou-se uma leve reversão da tendência a crescimento de confiabilidade e suspeitou-se da ocorrência de um novo ponto de inflexão do tipo 2. A suspeita ganhou um novo sentido quando a equipe do projeto informou que uma nova versão do módulo Depurador havia sido liberada em t= 80. Uma observação importante é que, embora não se soubesse a priori que o sistema havia sofrido uma alteração, esta intervenção foi notada na curva de tendência do processo de falha do sistema.

Em t=83, quarto ponto de avaliação periódica, o ponto de inflexão em t=79 se confirmou e a base de dados sofreu nova partição: G3 de t=35 a t=79. Os dados de falha do período seguinte ainda eram muito poucos (t=80 a t=83), de modo que ainda não se justificava a modelagem de G4. Nos dois pontos de avaliação seguintes (t=86 e t=89), foi seguida a evolução da tendência do processo de falha, cuja confiabilidade inicialmente decresce, passando a crescer após t=85.

No oitavo ponto de avaliação, em t=92, já se arriscou proceder à modelagem do processo G4, nesse momento com 12 unidades de tempo. Como foi discutido na seção 4.2.1, têm-se duas alternativas para a replicação de um processo do tipo de G4: aplicação do modelo Gama a todo o conjunto de dados G4, ou aplicação do modelo Exponencial à região de crescimento de confiabilidade: t=85 em diante, que será designada por G4'. Como o número de dados para a aplicação do modelo Exponencial ainda era muito pequeno, optou-se pelo modelo Gama. O modelo forneceu um erro de 0.93 e uma previsão do número de falhas de 2.34.

Em t=95 (nono ponto de avaliação) já havia dados suficientes para a aplicação do modelo Exponencial. A partir deste ponto de avaliação em diante (t=95, 98, 101, 104, 107, 11, 114 e 117) a avaliação era feita usando os dois modelos: Modelo Gama de t=80 em diante e modelo Exponencial de t=85 em diante. Os resultados eram comparados e

escolhido o modelo que fornecia menor erro, que, no caso, foi sempre o modelo Exponencial.

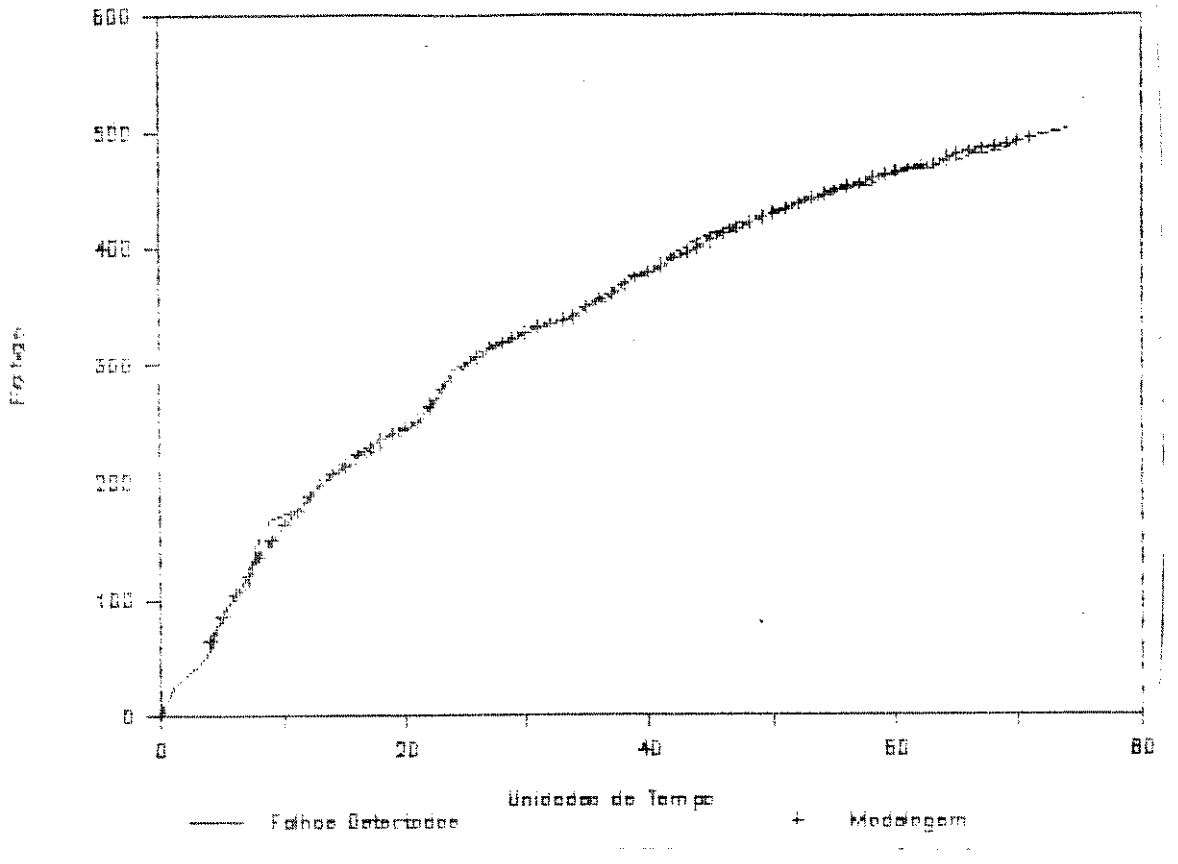

Figura 5.8 - Modelagem da curva de falha do APCC a partir do início do período de observação até o primeiro ponto de avaliação, em t=71.

O resultado das avaliações em todos os pontos em que elas foram feitas está mostrado na Tabela 5.1. A comparação entre o número de falhas previstas em cada ponto de avaliação e o número de falhas efetivamente detectadas nos meses onde não houve um ponto de inflexão é mostrada na Figura 5.9. O número previsto foi representado pelo inteiro mais próximo. O erro médio cometido nas previsões é de 1.09 falhas por unidade de tempo, o que é um bom resultado.

No final de 1989, t=98, ocorreu a renovação do projeto e nesta oportunidade se fazia necessária uma avaliação do esforço de manutenção exigido pelo sistema. O acompanhamento de confiabilidade estimava uma intensidade de falha decrescente, decorrente do crescimento de confiabilidade que o sistema vinha apresentando desde t=85. A intensidade de falha neste momento era de h(98)= 1.34 falhas nos próximos dez dias. A

estimativa era de que o número de correções fosse menor que 4 ao mês dali em diante, o que serviu para dimensionar a equipe de manutenção durante o novo período de vigência do projeto.

O Compilador CHILL é o maior módulo do Ambiente e o responsável pela grande maioria das falhas detectadas no sistema. Por este motivo, a partir de t=87 (RAC-07) passou-se a fazer uma avaliação paralela deste módulo em particular. Foram consideradas somente as falhas detectadas durante o período de acompanhamento da confiabilidade do sistema, de t=69 em diante. A evolução da tendência deste processo de falha observada em t=117 (RAC-16) era como mostrado na Figura 5.10.

RAC	t	Grupo	Modelo	a	b	Erro	Previsão
01	71	G3	Exponencial	570.57	0.0304	3.12	6.50
02	74	G3	Exponencial	573.54	0.030	3.03	6.04
03	77	G3	Exponencial	568.34	0.031	2.68	5.33
04	80	G3	Exponencial	576.42	0.029	2.82	5.15
05	83	G4	—	—	—	—	—
06	86	G4	—	—	—	—	—
07	89	G4	—	—	—	—	—
08	92	G4	Gama	556.56	0.300	0.93	2.34
09	95	G4'	Exponencial	565.36	0.116	0.94	2.74
10	98	G4'	Exponencial	583.54	0.062	1.22	3.67
11	101	G4'	Exponencial	578.59	0.073	1.16	2.66
12	104	G4'	Exponencial	580.43	0.068	1.03	2.29
13	107	G4'	Exponencial	583.81	0.061	1.04	2.76
14	111	G4'	Exponencial	584.93	0.058	0.99	1.76
15	114	G4'	Exponencial	585.88	0.057	0.97	1.54
16	117	G4'	Exponencial	599.41	0.039	1.63	2.05

Tabela 5.1 - Resultado da modelagem do APCC nos diversos pontos de avaliação.

O primeiro ponto de inflexão do tipo dois ocorre em t=13, definindo o primeiro grupo de dados: G1 de t=89 a t=91, com características compatíveis com as do modelo Exponencial. A curva de tendência a partir desse ponto é mostrada na Figura 5.11.

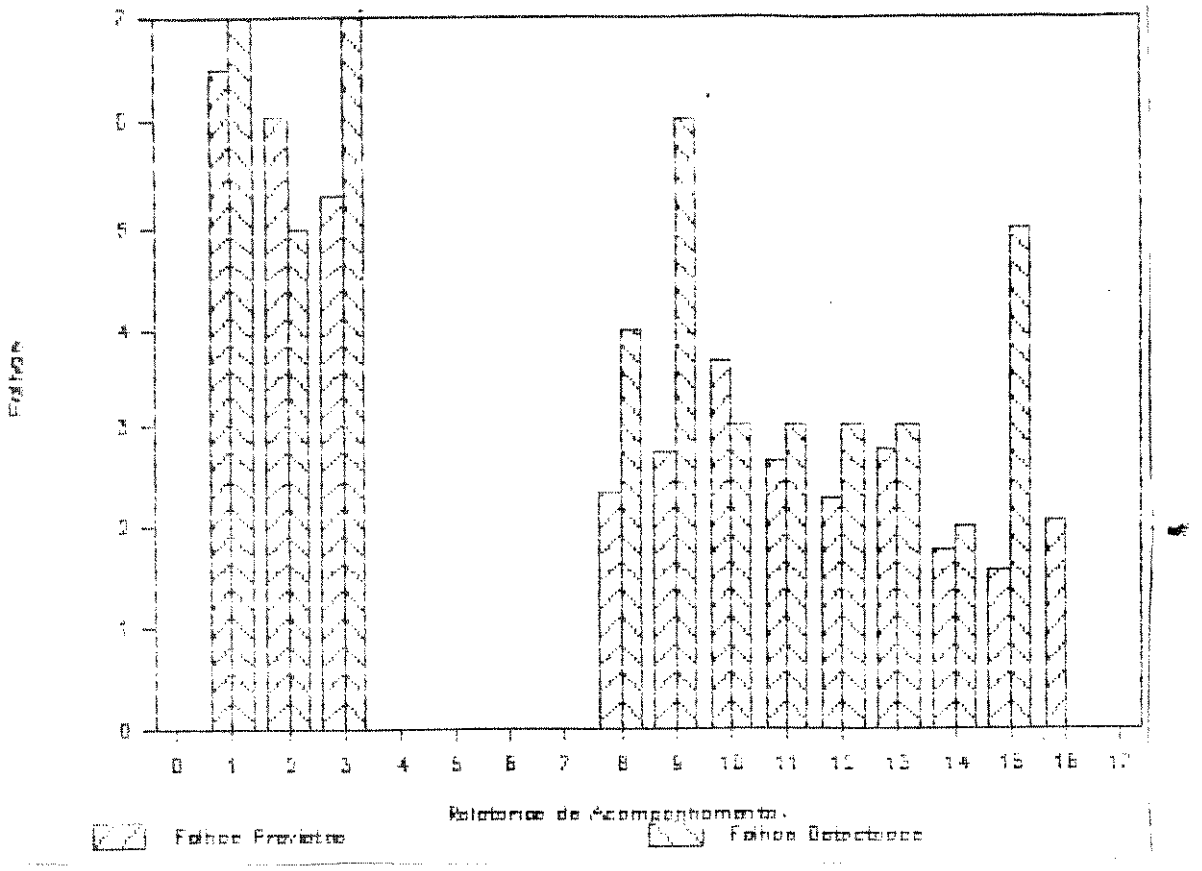

Figura 5.9 - Relação entre o número de falhas previsto para o mês seguinte e o número de falhas detectadas em cada ponto de avaliação.

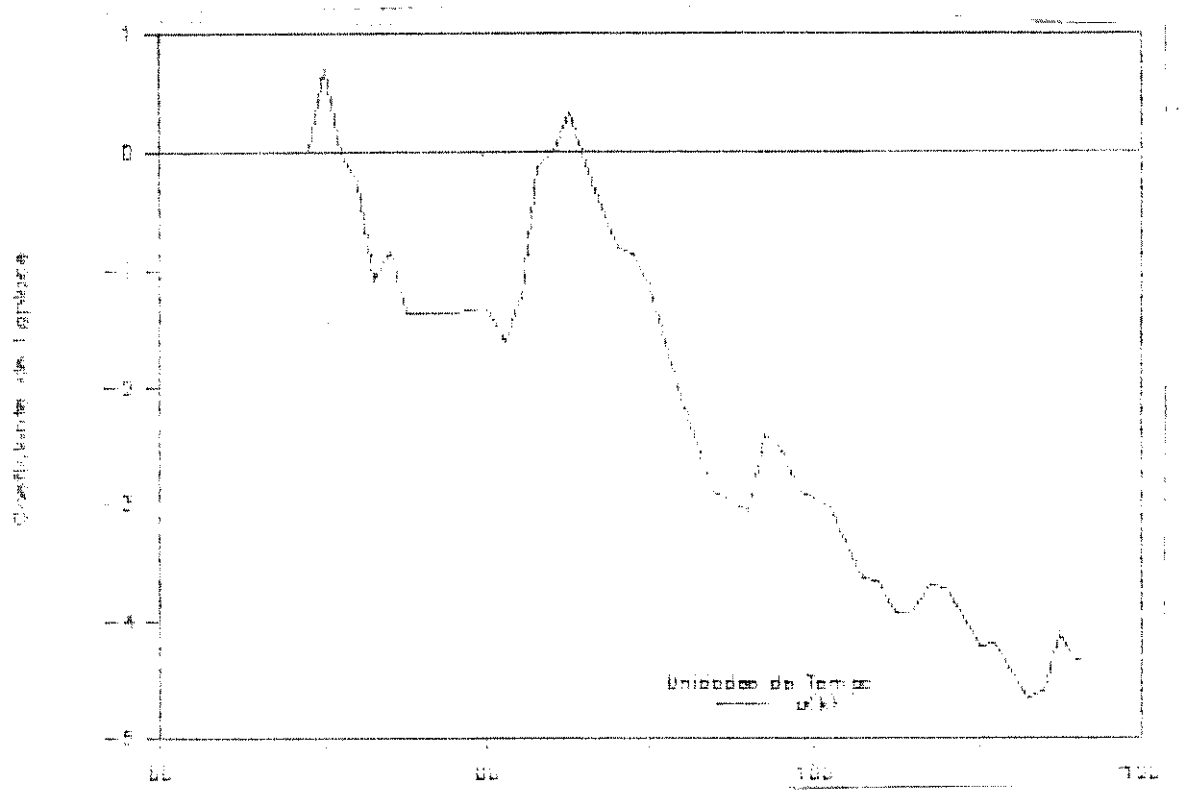

Figura 5.10 - Evolução da tendência do processo de falha do Compilador CHILL ao longo do período de acompanhamento da confiabilidade do sistema.

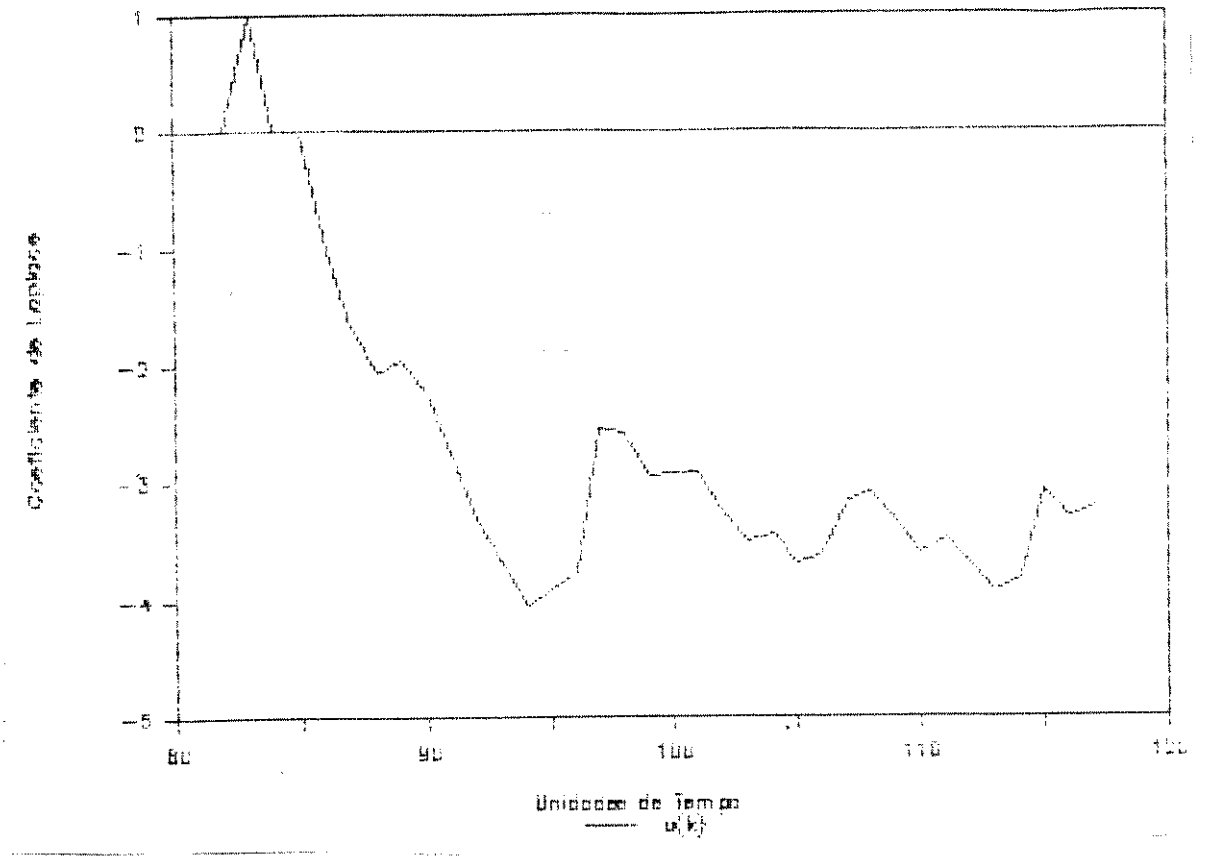

Figura 5.11 - Teste de tendência do processo de falha do Compilador excluindo os dados de G1.

Um novo ponto de inflexão ocorre em t=104 e a base de dados é mais uma vez subdividida em G2 de t=92 a t=104. O ligeiro decrescimento de confiabilidade no início desse período recomenda o modelo Gama. O teste de tendência da base de dados restante é mostrado na Figura 5.12.

Não foi registrado mais nenhum ponto de inflexão tipo dois neste período, de forma que os dados restantes, de t=105 a t=117 compõem o grupo de dados G3, ao qual aplica-se o modelo Exponencial. O resultado da modelagem do processo de falha completo do Compilador é mostrado na Figura 5.13. O erro médio cometido é de 0.77 falhas a cada unidade de tempo.

Infelizmente não estão disponíveis os dados relativos ao número de usuários do sistema ao longo do tempo, dessa maneira não é possível avaliar a confiabilidade média percebida por um usuário genérico do sistema. A intensidade de falha deste módulo em t=117 é de

0.63 falhas.

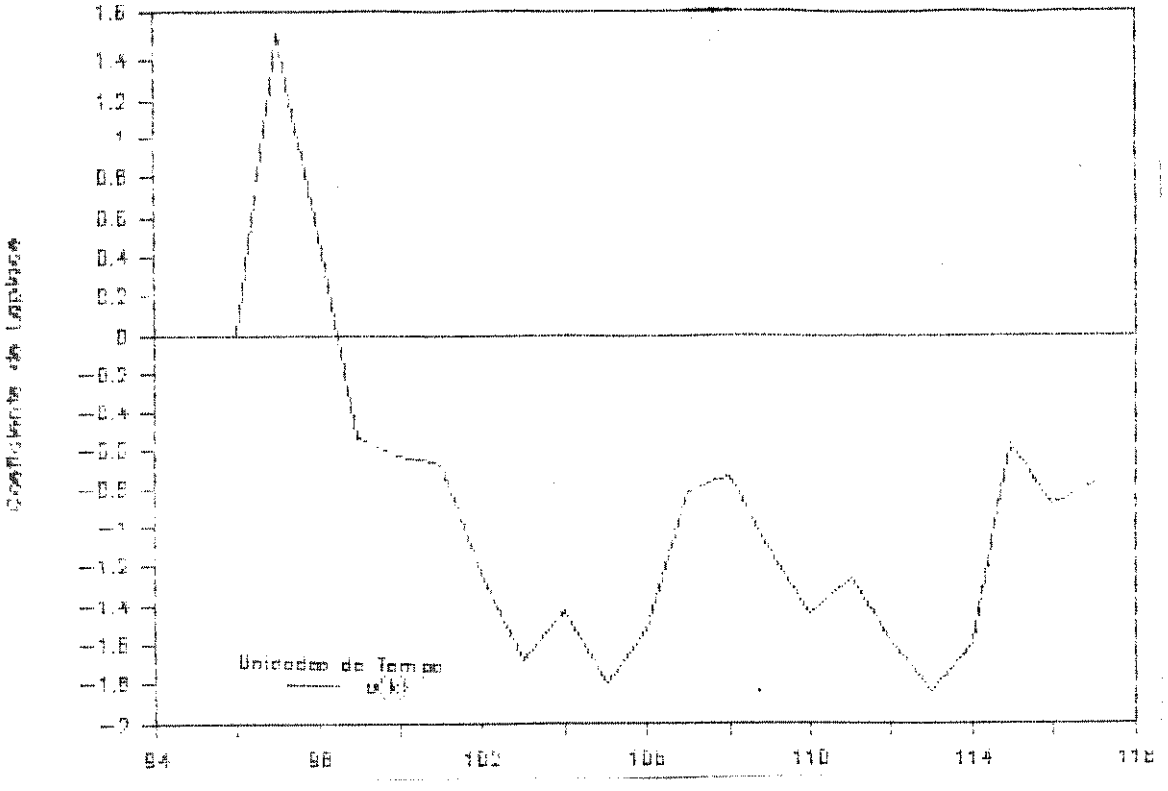

Figura 5.12 - Teste de tendência do processo de falha do Compilador excluindo os dados de falha de G1 e G2.

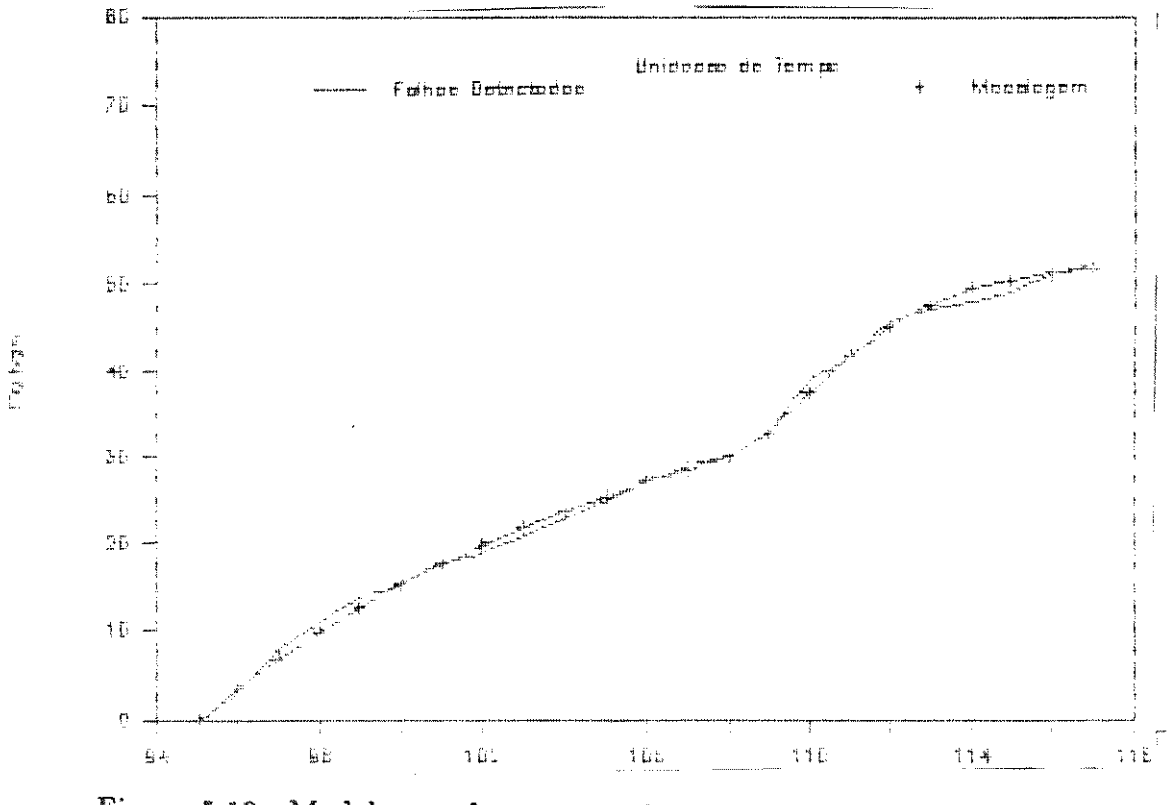

Figura 5.13 - Modelagem do processo de falha do Compilador CHILL.

5.6.1 Aspectos Qualitativos

Cada RAC compreende a análise de alguns aspectos qualitativos antecedendo resultados da aplicação dos modelos. No primeiro ponto de avaliação, por exemplo, foi feita uma análise da distribuição das falhas pelos projetos usuários do sistema APCC em que foram detectadas. Periodicamente este mapeamento foi refeito, de modo a acompanhar a evolução da depuração do produto por seus projetos usuários.

Um outro tipo de acompanhamento é o de falhas por módulo do sistema. Isto permitiu a identificação dos módulos mais sujeitos a falha, ou seja, dos ofensores de confiabilidade. No RAC-16, por exemplo, a distribuição das falhas detectadas durante o período de acompanhamento (t=71 a t=117) era a seguinte:

	Compilador	Depurador	LMP286	LNK286	P2C	Bibl.
RAC-01	12	0	1	0	1	0
RAC-02	19	0	1	0	1	0
RAC-03	24	0	2	0	1	0
RAC-04	30	1	2	0	1	0
RAC-05	40	3	2	0	1	0
RAC-06	47	7	2	1	1	0
RAC-07	51	11	3	1	1	0
RAC-08	52	11	5	1	1	1
RAC-09	53	13	6	1	1	1
RAC-10	57	15	6	1	1	1
RAC-11	59	18	6	1	1	1
RAC-12	60	19	6	1	1	1
RAC-13	64	19	6	1	1	1
RAC-14	65	19	7	1	1	1
RAC-15	68	19	8	1	1	1
RAC-16	72	19	9	1	1	1

Tabela 5.2 - Distribuição do número de falhas detectadas ao longo do período de acompanhamento de confiabilidade pelos módulos do APCC.

A partir deste mapeamento pode-se perceber que os módulos mais importantes do ponto de vista de falha são o Compilador, o Depurador e o ligador LMP286. A distribuição das falhas é coerente com o tamanho e importância destes módulos para o Ambiente.

No ponto de avaliação t=80 (RAC-04) foi feita uma análise de correlação entre o número de falhas detectadas nos módulos implementados em linguagem de alto nível (CHILL e Pascal) e o tamanho do seu código medido em número de linhas do código

fonte. Esta métrica foi utilizada devido à sua facilidade de aplicação e à indisponibilidade de ferramentas que implementassem outras métricas mais sofisticadas. Esta análise não forneceu um índice significativo de correlação, basicamente pela grande diferença de tamanho entre o Compilador e os demais módulos e pela constatação de que módulos aproximadamente equivalentes em número de linhas de código como o Depurador e o Tradutor Pascal-CHILL P2C terem apresentado uma evolução do número de falhas bastante diferente, como pode ser visto na Tabela 5.2. A conclusão naquele momento foi apenas de que o número de linhas do código fonte não é uma boa medida de complexidade para este sistema.

5.6.2 Avaliação dos Índices de Segurança de Funcionamento

Não foram fixados a priori requisitos de segurança de funcionamento para este sistema. Como não é conhecido o número de instalações e nem a consequência das falhas no serviço fornecido ao usuário, não é possível avaliar a indisponibilidade do sistema.
A expressão da probabilidade de que não ocorra uma falha durante um tempo especificado t_i^m,para modelos Poissonianos [MUS 87] é dada por:

$$R(t_i^m) = exp[-(H(t_{i-1} + t_i^m) - H(t_{i-1})] \tag{5.1}$$

A probabilidade de que o sistema não falhe na próxima unidade de tempo, calculada em t=117 é de 0.492.

Como o sistema se encontra em fase de crescimento de confiabilidade, a intensidade de falha no momento da qualificação funciona como um limitante superior da demanda de serviço de manutenção do sistema. Em t=117 avalia-se que a demanda futura seja sempre menor que 0.72 correções por unidade de tempo. Esta medida é útil no sentido de dimensionar a equipe de manutenção do sistema.

5.7 Sistema TRÓPICO R Versão 4096

O sistema de comutação digital TRÓPICO R foi desenvolvido no CPqD TELEBRÁS e posteriormente transferido às indústrias que obtiveram do Sistema Nacional de Telecomunicações o direito de fabricá-lo e comercializá-lo. A transferência da tecnologia e,

consequentemente, da responsabilidade de manter tanto o hardware quanto o software do sistema ocorreu em novembro de 1987. Nesta oportunidade foi feita uma avaliação global do sistema com o objetivo de retratar as condições em que as indústrias o estavam recebendo. É reproduzida aqui a análise dos processos de falha de concepção do hardware e do software.

5.7.1 Análise do Processo de Falha Software

Os resultados da análise de tendência e modelagem do software já foram apresentados na seção 4.2. Os resultados são refinados aplicando o método de partição da base de dados segundo o resultado do teste de tendência, que para este sistema é dado na Figura 4.1. É identificado somente um ponto de inflexão do tipo dois, em t=6, o que define o primeiro grupo de dados G1 de t=1 a t=6; a partir desse ponto há um decrescimento seguido de crescimento de confiabilidade, como mostra a Figura 5.14.

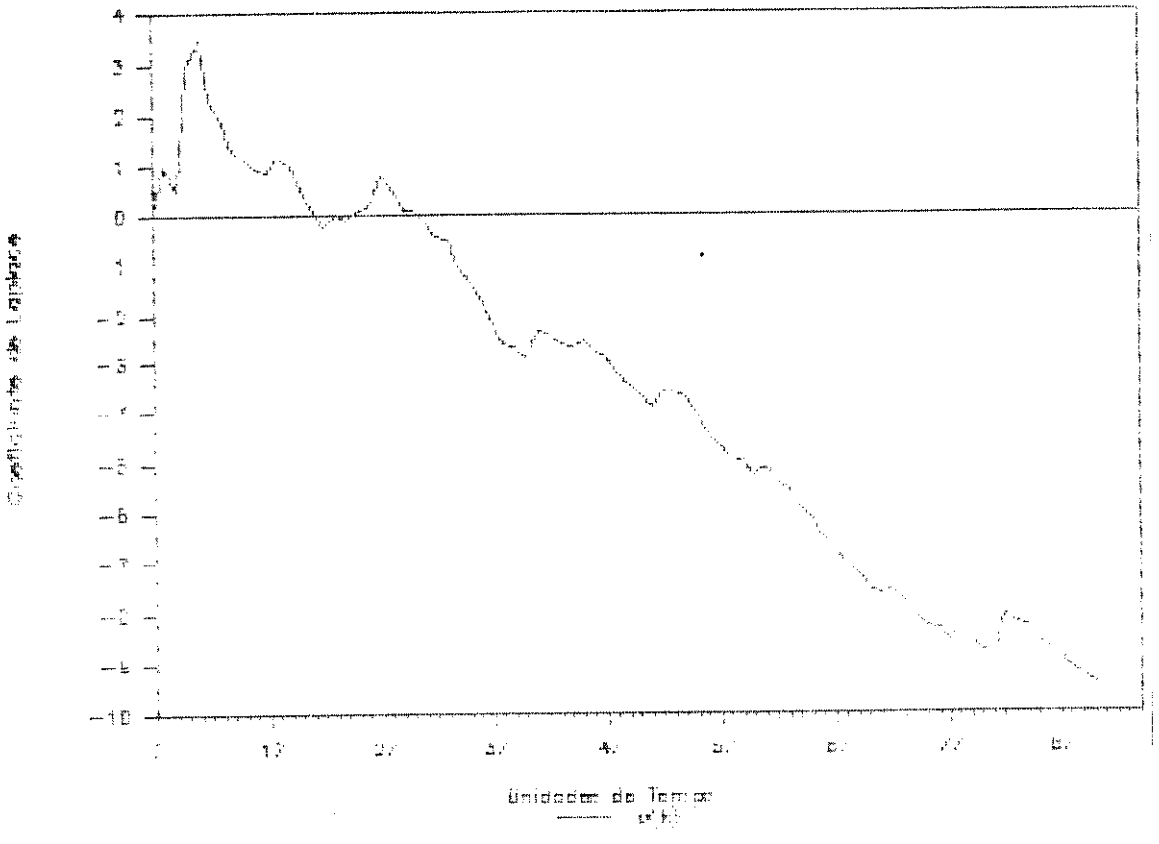

Figura 5.14 - Teste de Laplace dos dados de falha do software do sistema TRÓPICO R 4096 a partir de t=7.

Fica definido assim o segundo grupo de dados G2 de t=7 a t=90. Para o primeiro período é indicado o modelo Exponencial (somente decrescimento de confiabilidade). Para o segundo período pode ser aplicado o modelo Gama a todos os dados ou o modelo Exponencial ao período de crescimento de confiabilidade, que vai de t=10 em diante. Como as hipóteses destes dois modelos são satisfeitas, deve-se escolher o que fornecer melhores resultados. Como o período de crescimento de confiabilidade é bastante longo, o modelo Exponencial teve melhor desempenho, com erro de 2.59 contra um erro de 6.23 do modelo Gama.

A modelagem cometeu um erro médio de 2.43 falhas por unidade de tempo, como é mostrado na Figura 5.15. A intensidade de falha naquele momento era de 0.43.

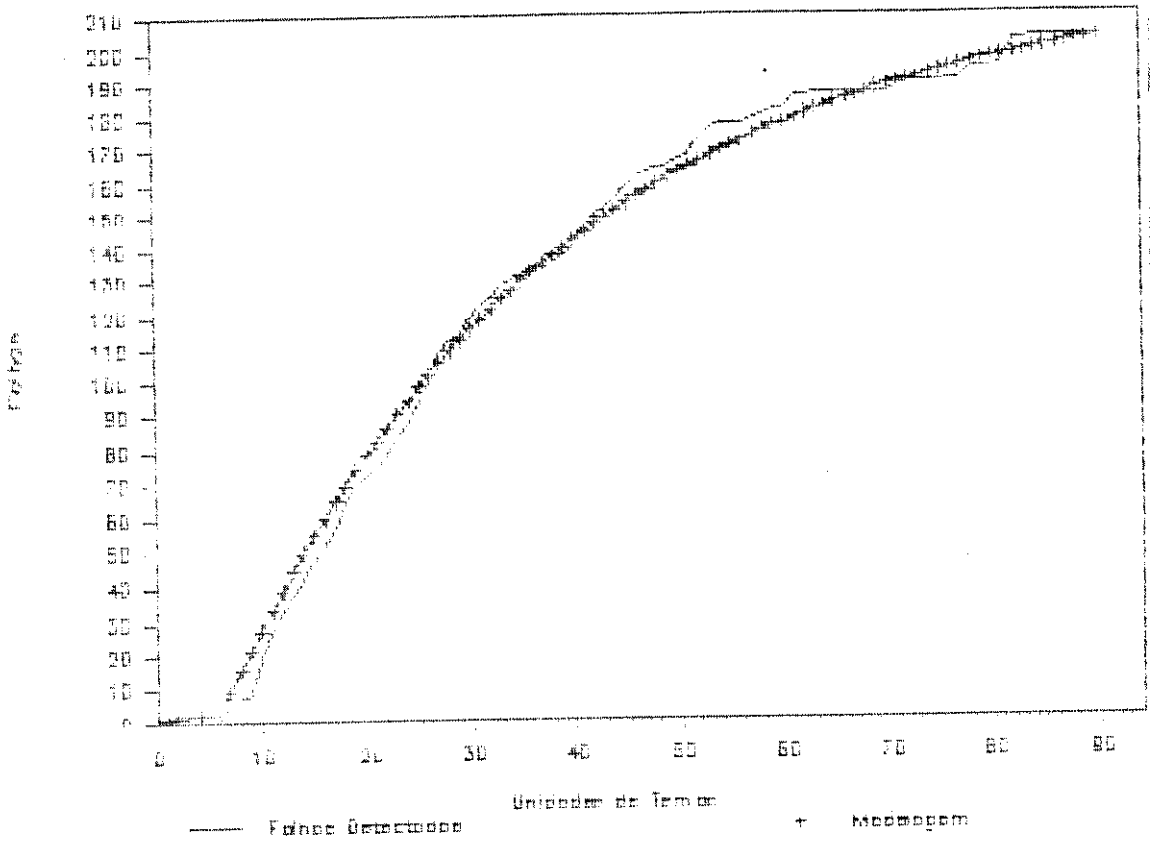

Figura 5.15 - Modelagem do processo de falha do software do sistema TRÓPICO R 4096.

5.7.2 Análise do Processo de Falha de Concepção do Hardware

A curva de tendência do processo de falhas de concepção do hardware é mostrada na Figura 5.16.

Os valores de u(k) para esta série se encontram entre -2.5 e 2.0. Esses valores baixos do módulo de u(k) indicam que não há crescimento de confiabilidade nem decrescimento, ou seja, o processo pode ser considerado estacionário. A curva de falha é aproximadamente linear e pode ser descrita por um modelo Exponencial. Aplicando o modelo de Goel e Okumoto obtem-se a curva estimada mostrada na Figura 5.17, onde o erro cometido foi de 2.27 falhas por unidade de tempo.

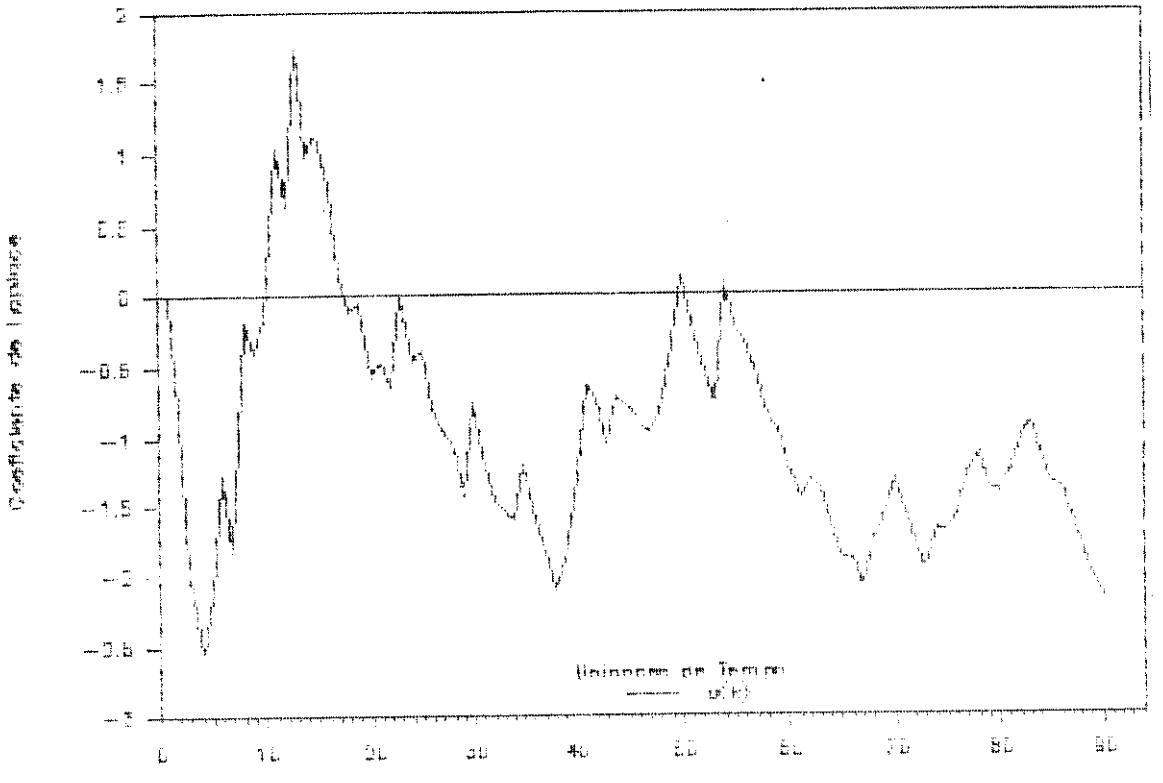

Figura 5.16 - Resultado da aplicação do teste de Laplace à base de dados de falha do hardware do sistema TRÓPICO R 4096.

Observa-se que neste exemplo o erro médio de replicação foi da mesma ordem do erro obtido na modelagem do processo de falha software desse mesmo sistema.

O hardware do TROPICO R 4096 é o mesmo hardware do TROPICO R 1500, de modo que pode-se considerá-lo um produto maduro, em plena vida útil. Isso pode explicar o comportamento estacionário de sua confiabilidade, enquanto que o software apresenta uma oscilação inicial na tendência de confiabilidade típica de sistemas jovens. A taxa de falha estimada no momento de avaliação t=90 era de 2.82 falhas por mês para o hardware.

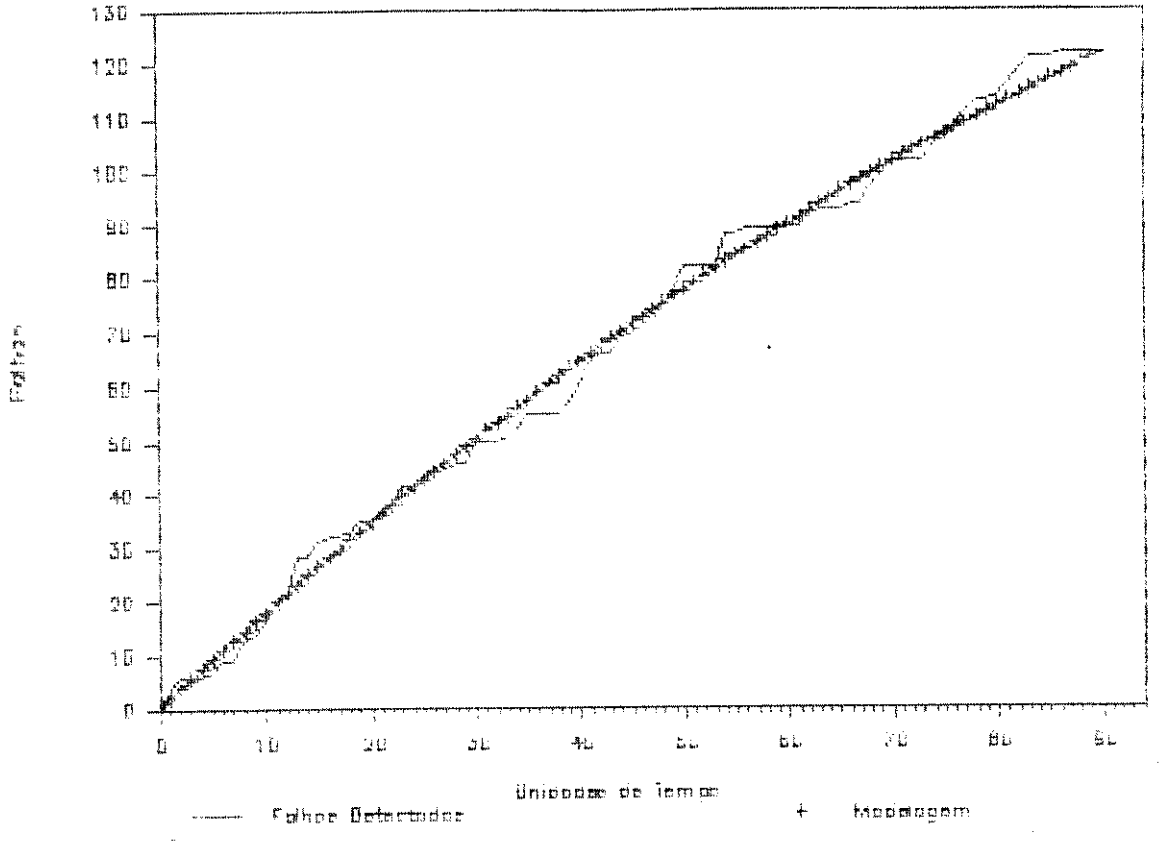

Figura 5.17 - Modelagem do processo de falha do hardware do sistema TRÓPICO R 4096 através do modelo Exponencial de Goel e Okumoto.

Dada a hipótese de independência, a taxa de falha do sistema pode ser obtida representando-o como uma composição em série das duas partes, e somando-se as taxas de falha de cada uma. Essa mesma taxa de falha poderia ser obtida pela aplicação dos modelos à base de dados do sistema, que seria a composição das duas bases. Para efeito de qualificação, no entanto, é importante estudar cada parte em separado. Como é visto no exemplo, o sistema pode ter o hardware estacionário e o software em franco crescimento de confiabilidade, com taxa de falha da ordem de 1/3 da do hardware. Somar as duas bases de dados não dá uma informação precisa de nenhum dos processo: a análise

do sistema composto diria que há uma lenta tendência a crescimento de confiabilidade.

5.7.3 Avaliação dos Requisitos de Segurança de Funcionamento

Veja-se como a avaliação de confiabilidade de concepção hardware e software pode ajudar na qualificação do sistema quanto a seus requisitos de segurança de serviço. Como já foi discutido no Capítulo 1, os requisitos especificados para sistemas normalmente não discriminam o tipo de falha a que se referem e devem ser avaliados de forma integrada, levando-se em conta todos os processos que para eles contribuem.

A avaliação da indisponibilidade quanto a falhas de concepção do sistema não pode ser feita pois não foram coletadas informações sobre a consequências das falhas registradas no serviço fornecido pelo sistema.
Em relação à criticalidade, os sistemas de comutação digital não estão sujeitos a nenhum requisito.

Em t=90 a probabilidade de que não ocorra uma falha de concepção software em uma unidade de tempo é de 0.657. A probabilidade de que não ocorra uma falha de concepção hardware é de 0.391.

Em relação a uma instalação média, a confiabilidade é de 0.910 para o hardware e de 0.959 para o software.

Quanto à manutenabilidade, o número máximo de intervenções por mês por 1000 terminais é especificado em 1,5 para sistemas de comutação digital. Uma intervenção numa instalação pode ser devida a uma falha de componente (geralmente caracterizada pela substituição de uma placa), ao projeto software (recarga de nova edição de um ou mais módulos software, nos quais um defeito tenha sido corrigido) ou a uma falha de projeto hardware (substituição de uma ou mais placas por novas edições).

A hipótese de manutenção é de que uma falha de componente ou de projeto leva a uma intervenção corretiva nas Instalações onde ela foi detectada (ressaltando que uma falha de concepção pode ser detectada em mais de uma Instalação). Nas Instalações onde

a falha de projeto ainda não foi detectada a manutenção preventiva é adiada até a primeira manutenção corretiva. De fato esta é uma política bastante realista, pois ela reduz o número de intervenções e consequentemente os custos de manutenção. É importante notar que neste contexto cada falha de componente leva a um pedido de manutenção porém somente a primeira detecção de uma falha de concepção é que gera uma correção do Sistema Fonte.

Infelizmente os dados relativos a ocorrências repetidas de uma falha não foram registrados para o sistema TRÓPICO R, de modo que os resultados são analisados considerando o número de Instalações onde a falha foi detectada como uma porcentagem variável do número total de Instalações.

Considerando-se que, para o sistema 4096:

- as falhas são independentes das Instalações,
- 100 Instalações são observadas,
- 150000 terminais estão em operação,
- o número de intervenções devido a falhas de componentes é de 1.3 por mês por 1000 terminais (este resultado foi obtido a partir dos dados de falha coletados por uma das indústrias fabricantes do equipamento, a partir dos sistemas em operação, entre as unidades t=20 e t=84).

O número de correções nas Instalações por mês por 1000 terminais é de 195 devido a falhas de componentes, 2.820 devido a concepção hardware e 0.978 devido ao software.

Do ponto de vista da equipe responsável pela manutenção, a reparação das falhas de componentes demanda um esforço muito maior do que a manutenção de concepção. No entanto a reparação de defeitos de concepção exige um trabalho mais qualificado e um tempo maior.

A tabela 5.3 dá o número de intervenções ao final do período de observação do hardware e do software do TRÓPICO R 4096. O número de intervenções por mês por 1000

terminais devido a falha de concepção é dado considerando o número médio de Instalações onde as falhas de concepção foram detectadas como 100%, 10% e 1% do total de Instalações. A política de 100% de detecção corresponde a fazer manutenção corretiva e preventiva imediatamente em todas as Instalações quando uma falha de concepção é detectada.

Instalações	Software	Concepção Hardware	Componentes Hardware	Soma
100%	0.652	1.880	1.3	3862
10%	0.065	0.188	1.3	1.553
1%	0.007	0.019	1.3	1.326

Tabela 5.3 - Taxas de Intervenção no TRÓPICO R 4096.

Considerando que intervenções devido a manutenção preventiva são feitas imediatamente a taxa total de intervenções é de 3.862 por mês por 1000 terminais. Com este desempenho o sistema não cumpre com os seus requisitos. Se menos de 10% das Instalações detectarem a mesma falha durante o mês o requisito é atingido desde que se considere apenas manutenção corretiva.

Do ponto de vista do Sistema Fonte, o número de solicitações de correções de falhas por mês devido a componentes hardware defeituosos é de 195, 1.88 com origem em defeitos de concepção do hardware e 0.51 devido a falhas de software, no momento em que foi feita a qualificação.

Duas observações importantes podem ser feitas a partir destes resultados:

1. A confiabilidade do software e da concepção do hardware devem ser levadas em conta na avaliação dos requisitos de confiabilidade do sistema, uma vez que elas podem representar um fator importante da qualidade de serviço percebida pelo usuário.

2. A taxa de falha de concepção, do ponto de vista de manutenção em fábrica pelo fornecedor, não sofre multiplicação com o número de Instalações do produto, como acontece com a taxa de falha de componentes. Por este motivo ela representa um fator importante no número de intervenções no início da vida operacional do sistema (número pequeno de Instalações) e passa a ser irrelevante à medida que o

número de Instalações cresce.

5.8 Conclusão

Este capítulo mostra como a metodologia de aplicação dos modelos de confiabilidade proposta no Capítulo 4 pode auxiliar na qualificação de sistemas quanto ao aspecto de concepção.

A qualificação é compreendida dentro do contexto de um Programa de Segurança de Funcionamento, como uma atividade periódica. Em cada ponto de qualificação são calculados os índices de segurança de funcionamento atingidos pelo sistema e feitas projeções do comportamento futuro.Estes índices e projeções reorientam o processo de desenvolvimento, de forma que os objetivos estabelecidos para o sistema sejam atingidos no prazo especificado.

A metodologia de aplicação dos modelos de confiabilidade proposta no Capítulo 4 permite avaliar as medidas de segurança de funcionamento de um sistema e analisar outros aspectos relevantes do desenvolvimento de sistemas.

O problema de avaliação de um sistema em qualquer ponto de seu desenvolvimento foi discutido à luz das hipóteses dos modelos existentes. Foi proposta uma forma de avaliação da eficiência de testes e impacto de modificações no sistema em vida operacional.

Foram apresentados dois exercícios de aplicação da metodologia proposta na qualificação de dois sistemas reais. No primeiro estudo de caso busca-se ressaltar o aspecto dinâmico do acompanhamento de confiabilidade feito periodicamente ao longo do ciclo de vida do sistema e de como os seus resultados podem auxiliar na tomada de decisões gerenciais. No segundo exemplo enfatiza-se a composição dos aspectos de confiabilidade do hardware e do software na avaliação global de um sistema e qual a influência desses aspectos em requisitos de qualidade de serviço do sistema.

Conclusão Geral

A principal ênfase neste trabalho é dada à qualidade na Engenharia de Concepção.

Os principais resultados expostos são relativos ao estabelecimento de programas e métodos de avaliação da qualidade:

- Programa de Segurança de Funcionamento levando em conta as falhas de concepção,
- Metodologia de Aplicação dos modelos de segurança de funcionamento quanto a concepção, e
- Qualificação de sistemas durante a fase de desenvolvimento e vida operacional.

A aplicação destes métodos reverte em informações importantes à gerência do desenvolvimento de sistemas, permitindo:

- aumentar a satisfação do cliente e do usuário,
- aumentar a produtividade e reduzir custos,
- diminuir o tempo de teste,
- reduzir a intensidade de falhas em campo a um nível aceitável,
- dimensionar o esforço de manutenção e
- guiar o aperfeiçoamento do processo de desenvolvimento de sistemas.

Procurou-se mostrar neste trabalho que o aspecto de concepção não pode ser desconsiderado na avaliação da segurança de funcionamento de sistemas . A complexidade crescente das funções hardware e software dos sistemas modernos, em particular dos sistemas de comutação digital, e a severidade dos seus requisitos de qualidade fazem com que a atribuição e a garantia da segurança de funcionamento passe a ser uma preocupação constante durante todo o processo de desenvolvimento destes sistemas. Dentro deste contexto se justifica falar de um Programa de Segurança de Funcionamento Quanto ao

Aspecto de Concepção.

Um Programa de Segurança de Funcionamento deve compreender técnicas de implementação e técnicas de garantia. Este trabalho trata somente de técnicas de verificação e validação, que permitem garantir a segurança de funcionamento desejada. As técnicas de implementação estão ligadas à engenharia de software e de hardware, e merecem um estudo complementar.

Os principais insumos de um Programa de Segurança de Funcionamento são os dados de falha do sistema. O registro destes dados não é uma tarefa simples, dados os prazos sempre exíguos para a execução das tarefas de desenvolvimento. Uma outra dificuldade é que a documentação completa de uma falha exige que ela seja relacionada ao defeito que a causou. Isto requer um conhecimento profundo e extenso do sistema, e que é difícil de ser alcançado em sistemas complexos e modularizados. Não obstante, os resultados obtidos neste trabalho são fruto da árdua tarefa de registro de falhas e espera-se que sejam um incentivo a ela.

Buscou-se sempre medidas quantitativas de segurança de funcionamento, por dois motivos:
1) só através de medidas é que se pode controlar o processo de desenvolvimento e
2) isto mantém uma coerência com a engenharia de segurança de funcionamento quanto a falhas de componentes hardware.

Foi proposta uma metodologia de aplicação dos modelos de segurança de funcionamento quanto a concepção originalmente desenvolvidos para a avaliação de confiabilidade do software. Esta metodologia se baseia na análise da natureza do processo estudado, através do teste de tendência, e na aplicação conveniente dos modelos, respeitando as hipóteses básicas com que eles foram concebidos. A metodologia pode se resumir nos seguintes pontos:

- conhecimento dos modelos e de suas hipóteses,
- conhecimento do comportamento de tendência do processo de falha estudado,

- partição conveniente da base de dados de falha,
- modelagem conveniente de cada uma das partições,
- composição das partições convenientemente modeladas,
- obtenção das medidas de segurança de funcionamento e
- previsão do comportamento futuro do processo de falha.

Os resultados obtidos mostram que esta metodologia, que é de aplicação razoavelmente simples, podendo diminuir bastante os erros de modelagem. Além disso ela permite estudar qualquer processo de falha, até mesmo os mais complexos. A metodologia é geral e independente dos modelos empregados.

É proposto que esta mesma metodologia seja também empregada no estudo de processos de falha de concepção do hardware. Os resultados obtidos são animadores neste sentido. Devido à independência do processo de desenvolvimento hardware e software a partir de certo ponto do ciclo de vida, propõe-se que os dois processos sejam estudados de forma independente, pois podem ter características diversas. A qualificação do sistema, porém, deve englobar estes dois aspectos, além da segurança de funcionamento quanto a falhas de concepção do hardware.

Quanto ao aspecto de qualificação de sistemas, propõe-se a análise de questões importantes como a eficiência da atividade de testes e do impacto de alterações das especificações do sistema no seu processo de falha. Foram tratadas também outras questões, como a avaliação do esforço de teste adicional e previsão dos índices de segurança de funcionamento em um certo instante futuro. É importante ressaltar que estes problemas surgiram da prática de avaliação da segurança de funcionamento de sistemas reais e solucionados a partir da metodologia proposta.

A qualificação de sistemas foi apresentada como sendo a avaliação das medidas de segurança de funcionamento em qualquer momento do ciclo de vida, a partir da integração dos módulos hardware/software. Esta abordagem traz algumas dificuldades que não eram sentidas na qualificação de sistemas já com longa vida operacional ou em qualificações

post mortem. Foram indicados os cuidados que devem ser tomados na interpretação dos resultados de modelagem destes processos de falha.

Os processos de falha dos sistemas estudados não permitiram a comprovação de nenhuma das duas hipóteses feitas sobre o comportamento da intensidade de falha quando $t \rightarrow \infty$. A maioria dos processos apresentava ainda crescimento de confiabilidade ao final do período de observação. Alguns processos apresentavam confiabilidade estacionária, porém com intensidade de falha ainda alta, não aceitável como intensidade de falha residual. Seria interessante acompanhar a evolução dos processos de falha dos sistemas APCC e TRÓPICO RA (que são os que continuam a ser acompanhados via Relatórios de Falha) com a finalidade de observar este aspecto. Uma outra questão a ser examinada diz respeito ao valor aceitável da taxa de falha residual de sistemas.

As falhas de concepção foram exploradas somente do ponto de vista quantitativo. O estudo qualitativo das falhas de concepção, classificando-as segundo sua causa, severidade, origem, tempo de correção e outras características, pode trazer informações valiosas sobre o sistema e seu processo de desenvolvimento. Este aspecto não foi estudado neste trabalho pela indisponibilidade de dados sobre as falhas dos sistemas tratados. Foi alterado o formulário de Relatório de Falha utilizado, de modo a incorporar estas informações. O novo modelo, que é apresentado no Anexo A, está sendo utilizado para registro das falhas do sistema TRÓPICO RA. Atualmente se está procedendo à análise qualitativa das falhas do software deste sistema e já se prevê resultados muito interessantes.

Parece possível estabelecer, em relação à concepção, programas de crescimento de segurança de funcionamento tradicionalmente aplicados a falhas de componentes hardware. Estes programas consistem na aplicação intensiva de técnicas de depuração visando reduzir a taxa de falha dos módulos hardware/software a um certo nível estabelecido.

Um outro ponto no qual parece ser interessante investir é na busca de uma relação entre o número de defeitos e a taxa de falha. Já existem algumas métricas que associam a complexidade dos módulos software ao número de defeitos neles latentes. Talvez a

partir desta estimação seja possível, considerando a forma de utilização destes módulos, estabelecer a evolução esperada do processo de falha de concepção. Da mesma forma isto poderia também ser feito para a concepção de módulos hardware. Estes fatores são também os utilizados em normas para a estimação da taxa de falha de componentes hardware.

Este trabalho contribuiu para o alargamento das fronteiras da avaliação e da validação de sistemas, englobando aspectos novos e mostrando sua utilidade prática na reorientação do processo de desenvolvimento. Espera-se que a partir desta nova perspectiva muitas outras questões venham a ser estudadas.

Bibliografia

ABD 86 Abdalla, A.A., P.Y. Chan, B. Littlewood, "Evaluation of Competing Software Reliability Predictions", *IEEE Transactions on Software Engineering*, vol.SE-12, número 9, Setembro 1986, pp. 950-967.

ABN 81 Associação Brasileira de Normas Técnicas, "Confiabilidade - Terminologia", NBR 5462, Setembro 1981.

ALB 82 Albin, J.L., R.Ferrèol, "Collecte et Analyse de Mesures de Logiciel", *TSI - Technique et Science Informatiques*, vol.1, número 4, 1982.

ALI 86 Ali, S.R., "Study of Total Outage Data for SPC Switching Systems", Proceedings International Conference os Communications ICC'86, Toronto, Canadá, Junho 1986.

ANG 80 Angus, J.E., R.E.Schafer, A.Sukert, "Software Reliability Model Validation", Proceedings 1980 Annual Reliability and Maintainability Symposium, 1980, pp.191-199.

ARA 88 Araújo, C.C.F., e outros, " System Design and Software Development Methodology in the TROPICO System", Proceedings International Conference on Communications (ICC88), Philadelphia, Junho 1988.

ASC 84 Ascher, H., H.Feingold, "Application of Laplace's Test to Repairable System Reliability", Proc. 1rst International Conference on Reliability and Maintainability, Paris, Junho 1978, pp.219-225.

AVE 80 Aveyard,R.L., F.T.Man, "A Study on the Reliability of the Circuit Maintenance System 1-B", "Bell System Technical Journal, vol.59, Outubro 1980, pp.1317-1332.

BAK 88 Baker, C.T., "Effects of Field Service on Software Reliability",*IEEE Transactions on Software Engineering*, vol. 14, número 2, Fevereiro 1988, pp. 254-258.

BAS 85 Bastani, F.B., "On the Uncertainty in the Correctness of Computer Programs", *IEEE Transactions on Software Engineering*, vol.SE-11, número 9, Setembro 1985, pp. 857-864.

BAS 88 Bastos Martini, M.R., J.Moreira de Souza, H.M.F.Tavares, "Software Reliability Growth Analysis for the TROPICO R Switching System ", Proc. 6th International Conference on Reliability and Maintainability, Strasbourg , Outubro 1988, pp. 213-218.

BAS 89a Bastos Martini, M.R., K.Kanoun e J. Moreira de Souza, "Um Método para Avaliação e Previsão da Confiabilidade de Software: Aplicação ao Sistema TROPICO R", Anais do 7o Simpósio Brasileiro de Telecomunicações, Setembro 1989, pp. 430-437.

BAS 89b Bastos Martini, M.R, e J.Moreira de Souza, "Avaliação da Confiabilidade de Concepção de Sistemas Computacionais", Anais do III Simpósio em Sistemas de Computadores Tolerantes a Falhas, Setembro 1989, pp. 19-37.

BAS 90a Bastos Martini, M.R. e J.Moreira de Souza, "Reliability Growth Analysis for Hardware and Software Design Failure Data: an Approach Based on Trend Test", Proceedings 7th International Conference on Reliability and Maintainability, Junho 1990, pp. 340-346.

BAS 90b Bastos Martini, M.R., K.Kanoun e J.Moreira de Souza, "Software Reliability Evaluation of the TROPICO R Switching System", *IEEE Transactions on Reliability*, vol.R-39, no 3, Agosto 1990, pp. 369-379.

BAS 90c Bastos Martini, M.R. e J.Moreira de Souza, "Modelos de Confiabilidade de Software", Relatório Técnico TELEBRAS, Setembro 1990.

BAS 90d Bastos Martini, M.R. e J.Moreira de Souza, "Reliability Assessment of Computer System Design",*Annales des Télécommunications*, tomo 45, número 11-12, Novembro-Dezembro 1990, pp.642-647.

BAS 91 Bastos Martini, M.R. e J.Moreira de Souza, "Reliability Assessment of Computer Systems Design ", *Microelectronics and Reliability*, vol.31, número 2/3, Fevereiro 1991, pp.237-244.

BOE 76 Bohem, B.W., "Software Reliability", *IEEE Transactions on Computers*, vol.C-25, número 12, Dezembro 1976.

BOE 81 Bohem, B.W., *Software Engineering Economics*, Prentice Hall Inc., Englewood Cliffs, N.J., 1981.

BOR 84 Borges, A.A., *Central Local/ Tandem Digital do Sistema TRÓPICO*, Edição CPqD TELEBRÁS, 1984.

BOU 82 Bourgonjon, R.H. e W.G.Ekas, "Reliability Aspects of Large Software Systems", Philips Telecommunications Review, vol. 40, número 1, Abril 1982, pp.35-44.

CAT 85 Catuneanu, V.M.,A.N. Mihalache, "Improving the Accuracy of the Littlewood-Verral Model",*IEEE Transactions on Reliability*, vol.R-34, número 5, Dezembro 1985, pp.418-424.

CAV 85 Cavano, J.P., "Toward High Confidence Software", *IEEE Transactions on Software Engineering*, vol. SE-11, número 12, Dezembro 1985, pp. 1149-1455.

CNE 83 CNET, "Recueil de Données de Fiabilité du CNET", Centre National d' Etudes des Télécommunications, 1983.

COS 78 Costes, A., C.Landrault, J.C.Laprie, "Reliability and Availability Models for Maintained Systems Featuring Hardware Failures and Design Faults", *IEEE Transactions for Computers*, vol.C-27, Junho 1978, pp.548-560.

COX 78 D.R.Cox, P.A.Lewis, *The Statistical Analysis of Series of Events*, Chapman and Hall, London, 1978.

CRO 74 Crow, L.H., "Reliability Analysis for Complex Repairable Systems",*Reliability and Biometry*, SIAM, Philadelphia, 1974, pp. 379-410.

CRO 84 Crow, L.H., N.D. Singpurwalla, "An Empirically Developed Fourier Series Model for Describing Software Failures", *IEEE Transactions on Reliability*, vol. R-33, número 2, Junho 1984, pp. 176-183.

DAL 86 Dale, C., "Software Reliability Models", Pergamon Infotech State of the Art Report, Pergamon Infotech, UK, Abril 1986, pp.31-44.

DER 90 Derriennic, H., "Une Expérience de L' Utilization des Modéles Classiques à Croissance de Fiabilité du Logiciel dans le Domaine des Télécommunications", Proceedings 7th International Conference on Reliability and Maintainability, Junho 1990, pp. 325 a 331.

DIA 88 Diaz, V.A.V., "Hardware Architecture for the TRÓPICO RA Switching System", Proceedings International Conference on Communications ICC88, Philadelphia, Junho 1988.

DUA 64 Duane, J.T., "Learning Curve Approach to Reliability Monitoring", *IEEE Transactions on Aerospace*, AS-2(2), 1964, pp. 563-566.

END 75 Endres, A., "An Analysis of Error and their Causes in System Programs", *IEEE Transactions on Software Engineering*, Junho 1975, pp 140-149.

FAL 90 Falkena, O., A.Sandberg e P.O.Nilsson, "Digital Switching - 10 Years of Operational Experience", Proceedings XIII International Switching Symposium, vol.V, pp.82-89.

FAR 87 Fard, N.S., D.L.Dietrich, "A Bayes Reliability Growth Model for a Development Testing Program", *IEEE Transactions on Reliability*, vol. R-36, número 5, Dezembro 1987, pp. 568-572.

FER 86 Ferréol, R., "Predetermination de la Fiabilité a Partir de la Complexité des Programmes", Analles Seminaire AFCET, Junho 1986.

FIC 85 Fiche, G., F. le Corre, "Methods for Evaluating the Dependability of a Telephone Switching System", *Commutation & Transmission*, número 2, 1985, pp. 91-104.

FIN 79 Finkelstein, J.M., *IEEE Transactions on Reliability*, vol.R-28, número 2, Junho 1979, pp.111-113.

FRA 87 Fraga, J., R. Lemos, "Conceituação e Terminologia em Sistemas Informáticos Tolerantes a Faltas e Intrusões", Anais do Segundo Simpósio em Sistemas de Computadores Tolerantes a Falhas, Campinas, Agosto 1987, pp. 117-188.

FUK 86 Fukushima K., e Y.Kishida, "Estimation of Bug Curve Using Experimental Regression Analysis of Office Automation Equipment Software", 5th International Conference on Reliability and Maintainability, 1986, pp 87-91.

GAU 88 Gaudoin, O., "Les Tests de Tendence de Fiabilité des Systèmes Reparables. Application a la Fiabilité des Logiciels", RT 41-1988, Laboratoire IMAG/TIM3, 1988.

GAU 89 Gaudoin, O., "Deux ou Trois Choses que je Sais du Test de Laplace", Laboratoire IMAG/TIM3, RT 48, Abril 1989.

GOE 79a Goel, A.L. e K.Okumoto, "An Analysis of Recurrent Software Errors in a Real-Time Control System", Proceedings ACM Conference,1979, pp. 496-501.

GOE 79b Goel, A.L. e K.Okumoto, "Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures", *IEEE Transactions on Reliability*, R-28(3), 1979, pp. 206-211.

GOE 79c Goel, A.L., K. Okumoto, "A Markovian Model for Reliability and Other Performance Measures of Software Systems", Proceedings National Computer Conference, 1979, pp. 769-774.

GOE 83a Goel,A.L., "A Guidebook for Software Reliability Assessment", RADC Report, RADC-TR-83-176, 1983.

GOE 83b Goel, A.L., "Software Reliability Models: Assumptions, Limitations and Applicability", *IEEE Transactions on Software Engineering*, vol. SE-11, número 12, Dezembro 1985, pp. 1141-1423.

GOE 85 Goel, A.L., "Software Reliability Models: Assumptions, Limitations and Applicability", *IEEE Transactions on Software Engineering*, Special Issue on Software Reliability, vol. SE-11, número 12, Dezembro 1985, pp.1141-1413.

GON 86 Gonçalves, R.A., "Sistemas de Comunicação de Dados Via Satélite Usando a Técnica AMDT com Multifrequência - SAMSAT", Anais 4o Simpósio Brasileiro de Telecomunicações", Rio de Janeiro, Setembro 1986, pp. 173-177.

HAR 89 Hartler, G., "The Nonhomogeneous Poisson Process - A Model for the Reliability of Complex Repairable Systems", *Microelectronics and Reliability*, vol. 29, número 3, 1989, pp. 381-386.

HOG 67 Hog, R.V. e A.T. Craig, *Introduction to Mathematical Statistics*, MacMillan Company, New York, 1967.

HOL 90 Holanda, Giovanni Moura de, M.R. de Bastos Martini, "Relatório Técnico de Acompanhamento de Confiabilidade de Concepção do SAMSAT-RAC02", Relatório Técnico CPqD TELEBRÁS 7413/90/06, 1990.

HUG 90 Hugueney Jr., C., "Introdução à Família TRÓPICO", *Revista TELEBRAS*, vol.14, número 48, Junho 1990, pp.3-7.

IAN 84 Iannino, A., J.D.Musa, K.Okumoto, B.Littlewood, "Criteria for Software Reliability Model Comparisons", *IEEE Transactions on Software Engineering*, vol. SE-10, número 6, Novembro 1984, pp. 687-691.

JEL 72 Jelinski, Z., P.B. Moranda, "Software Reliability Research", *Statistical Computer Performance Evaluation*, Academic, New York, 1972, pp. 465-4884.

JOH 85 Johanson, C. e I. Svenle, "Handling of AXE 10 Software", *Ericson Review*, vol. 62, 1985, número 1.

KAH 87 Kahn, P., "Construction et Verification de Logiciels Fiables en Milieu Industriel", Analles 9ème Séminaire Tuniso-Français d' Informatique, Tunis, Abril 1987A.

KAN 85 Kanoun, K., J.C.Laprie, "Modeling Software Reliability and Availability from Development-Validation up to Operation", Rapport de Recherche L.A.A.S., 1985, número 85-042.

KAN 87a Kanoun, T. Sabourin, "Software Dependability of a Téléphone Switching System", Proceedings 17th Symposium on Fault Tolerant Computer System, Pittsbourg, Pennsylvania, Julho 1987, pp 236-241.

KAN 87b Kanoun, K., T. Sabourin, "Analyse des Deffaillances et Évaluation de la Sureté de Fonctionement du Logiciel d' un Autocommutateur Téléphonique", *Technique et Science Informatique*, vol. 6, número 4, 1987, pp. 287-303.

KAN 88a Kanoun, K., J.C. Laprie, T. Sabourin, "A Method for Software Reliability Growth Analysis and Evaluation", *Le Génie Logiciel & ses Applications*, Toulouse, Dezembro 1988, pp. 859-878.

KAN 88b Kanoun, K., "Analyse de la Croissance de Fiabilité du Logiciel", Proc. 6th International Conference on Reliability and Maintainability, Outubro 1988, pp. 651-656.

KAN 89 Kanoun, K., "Croissance de la Sureté de Fonctionement des Logiciels: Caracterization - Modélisation - Évaluation", tese de Doutorado de Estado apresentada ao Laboratoire dÁutomatique et dÁnalyse de Systémes, Setembro 1989.

KAN 91 Kanoun,K., M.R.Bastos Martini e J.Moreira de Souza, "A Method for Software Reliability Analysis and Prediction - Application to the TROPICO-R Switching System", *IEEE Transactions on Software Engineering*, vol.17, número 4, Abril 1991, pp.334-344.

KEI 83 Keiller, P.A., B. Littlewood, D.R.Miller e A. Sofer, "On The Quality of Software Reliability Prediction", *Electronic Systems Effectiveness and Life Cycle Costing*, NATO ASI Series, F3, Springer Verlag, Heidelberg, pp. 441-460.

KEN 61 Kendall, M.G. e A.Stuart, *The Advanced Theory of Statistics* ,vol. 1, Hafner, New York, 1961.

KEN 86 Kenet, R., M. Pollak, "A Semi-Parametric Approach to Testing for Reliability Growth, with Application to Software Systems", *IEEE Transactions on Reliability*, vol. R-35, número 3, Agosto 1986, pp. 304-311.

KLI 80 Kline, M.B., "Software & Hardware R&M: What are the Differences?", Proceedings Annual Reliability and Maintainability Symposium, 1980, pp. 179-185.

KRE 83 Kremer, W., "Birth-Death and Bug Counting", *IEEE Transactions on Reliability*, R-32(1), 1983, pp. 37-47.

LAN 85 Langberg, N. e N. Sigpurwalla, "A Unification of Some Software Reliability Models", *SIAM Journal on Scientific and Statistical Computing*, 6(3), 1985, pp. 781-790.

LAP 84a Laprie, J.C., "Dependability Evaluation of Software Systems in Operation", *IEEE Transactions on Software Engineering*, vol. SE-10, número 6, Novembro 1984, pp. 701-714.

LAP 84b Laprie, J.C., "Dependability Modeling and Evaluation of Software-and-Hardware Systems", trabalho convidado pela 2nd GI/NTG/GMR Conference on Fault-Tolerant Computing, Bonn, Setembro 1984.

LAP 86a Laprie, J.C., "The Dependability Approach to Critical Systems", contribuição a convite do 5th International Workshop on Trends in Safe Real Time Computer Systems SAFECOMP'86, França, Outubro 1986.

LAP 86b J.C.Laprie,"Dependability: A Unifying Concept for Reliable Computing and Fault Tolerance", *Resilient Computing Systems*, T.Anderson(ed), vol.2, Collins &Wiley 1978; Rapport de Recherche L.A.A.S., no. 86357, Dezembro 1986.

LAP 87 Laprie, J.C., "Vers une Théorie de la Fiabilité du X-iel", Rapport de Recherche L.A.A.S., número 87-053, Fevereiro 1987.

LAP 89 Laprie,J.C., "Hardware-and-software Dependability Evaluation", Proceedings IFIP 11th World Computer Congress, San Francisco, U.S.A, Setembro 1989.

LAP 90 Laprie, J.C. et alli, "The Transformational Approach to the Modeling and Evaluation of the Reliability and Availability Growth of Systems in Operation", Proceedings IEEE FTCS-20, Newcastle, Junho 1990.

LIT 73 Littlewood, B. e J.L.Verral, "A Bayesian Reliability Growth Model for Computer Software", *Journal Royal Statistical Society*, Series C, 22(3), 1973, pp. 332-346.

LIT 79 Littlewood, B., "How to Measure Software Reliability and How Not To", *IEEE Transactions on Reliability*, vol. R-28, número 2, Junho 1979, pp. 103-110.

LIT 80 Littlewood, B., "Theories of Software Reliability: How Good are They and How Can They be Improved", *IEEE Transactions on Software Engineering*, vol. SE-6, número 5, Setembro 1980, pp. 489-500.

LIT 81a Littlewood, B., "Stochastic Reliability-Growth: A Model for Fault Removal in Computer-Programs and Hardware Design", *IEEE Transactions on Reliability*, R-30(4), 1981, pp. 313-320.

LIT 81b Littlewood, B., J.L.Verral, "Likelyhood Function of a Debugging Model for Computer Software Reliability", *IEEE Transactions on Reliability* ,vol.R-30, número 2, Junho 1981, pp. 145-148.

MAT 88 Matsumoto, K. e outros, "Experimental Evaluation of Software Reliability Growth Models", Proceedings 18th International Symposium on Fault Tolerant Computing, Junho 1988, pp. 148-153.

MEL 87 Mellor, P., "Software Reliability Modelling: the State of the Art", *Information and Software Technology*, vol.29, número 2, Março 1987, pp. 82-98.

MET 90 Metge, S., "Analyse et Evaluation de la Fiabilité de Deux Logiciels de Télécommunication", Tese elaborada no L.A.A.S.-C.N.R.S. e apresentada ao Conservatoire National des Arts et Métiers, Centre Régional Associé de Toulouse, França, afim de obter o Diploma de Engenheiro C.N.A.M. em Informática, em 18/05/1990.

MIL 86a MIL-HDBK-217E, *Military Handbook Reliability Prediction of Electronic Equipment*, Department of Defense of the United States of America, 1986.

MIL 86b Miller, D.R., "Exponential Order Statistic Models of Software Reliability Growth", *IEEE Transactions on Software Engineering*, vol. SE-12, número 1, Janeiro 1986, pp. 12-24.

MIS 83 Misra, P.N., "Software Reliability Analysis", *IBM Systems Journal*, vol. 22, número 3, 1983, pp. 262-270.

MOR 75 Moranda, P.B., "Predictions of Software Reliability During Debugging", Proceedings Annual Reliability and Maintainability Symposium, Washington DC, 1975, pp.327-332.

MOR 84 Moreira de Souza, J., A.C.Lavelha e M.R.Bastos Martini, "Évaluation des Consequences Economiques des Defaillances: Application au Système TROPICO R", 4th International Conferencé on Reliability and Maintainability, Perros-Guirec, França, Maio de 1984.

MOR 86 Moreira de Souza, J., A.C.Lavelha e M.R.Bastos Martini, "Évaluation de la Qualité de Service dún Système a Degradation Progressive", 5th International Conference on Reliability and Maintainability, Biarritz, França, Outubro 1986.

MUS 75 Musa, J.D., "A Theory of Software Reliability and its Application", *IEEE Transactions on Software Engineering*, SE-1(3), pp 312-327.

MUS 79a Musa, J.D., comunicação particular a B.Littlewood, 1979.

MUS 79b Musa,J.D., "Software Reliability Data", Data and Analysis Center for Software, Rome Air Development Center (R.A.D.C), 1979.

MUS 80 Musa, J.D., "Software Reliability Measurement", *The Journal of Systems and Software*, vol. 1, 1980, pp. 223-241.

MUS 83 Musa, J.D e Okumoto, K. "Software Reliability Models: Concepts, Classification Comparisons and Practices", *Electronic Systems Effectiveness and Life Cycle Costing*, Séries NATO ASI, F3, Springer-Verlag, Heidelberg, pp 395-424.

MUS 84 Musa, J.D. e K. Okumoto, "A Logarithmic Poisson Excecution Time Model for Software Reliability Measurement", Proceedings Seventh International Conference on Software Engineering, Orlando, 1984, pp. 230-238.

MUS 87 Musa,J.D., A.Iannino e K. Okumoto, *Software Reliability Measurement, Prediction, Application*, McGraw-Hill Book Company, 1987.

MUS 89 Musa, J.D., "Tools for Measuring Software Reliability", *IEEE Spectrum*, Fevereiro 1989, pp. 39-42.

NAY 86 Nayak, T.K., "Software Reliability: Statistical Modeling & Estimation", *IEEE Transactions on Reliability*, vol. R-35, número 5, Dezembro 1986, pp. 566-570.

OHB 84a Ohba, M. e Y.Yamada, "S-shaped Software Reliability Growth Models", Proc. 4th International Conference on Reliability and Maintainability, Perros Guirec, France, 1984, pp.430-436.

OHB 84b Ohba, M., "Software Reliability Analysis Models", *IBM Journal of Research and Development*, número 4, Julho 1984.

OKU 80 Okumoto, K., A.L.Goel, "Optimum Release Time for Software Systems Based on Reliability Cost Criteria", *The Journal of Systems and Software I*, 1980, pp. 315-318.

OKU 85 Okumoto, K., "A Statistical Method for Software Quality Control",*IEEE Transactions on Software Engineering*, vol. SE-11,número 12, Dezembro de 1985, pp. 1424-1430.

PAL 89 Palma Neto, A.V.A e C.C. de Figueiredo, "The CPqD TELEBRAS CHILL Programming Environment", Proceedings TENCON'89, Bombain, Novembro 1989, pp. 316-319.

PAU 87 Paula Jr., A.R., et alli, *Minicurso de Introdução a Tolerância a Falhas*, livro do Minicurso do 2o Simpósio em Sistemas de Computadores Tolerantes a Falhas.

PEN 90 Pence, J.L., S.E.Hon, "Software Surveillance: A Buyer Quality Assurance Program", Proceedigs XIII International Switching Sysposium, vol IV, Estocolmo, Suécia, Junho 1990, pp.77-83.

PIG 88 Pignal, "An Analysis of Hardware and Software Availability Exemplified on the IBM 3725 Communication Controller", *IBM Journal of Research and Development*, vol.32, número 2, Março 1988, pp.268-278.

PIR 84 Pirsig, A.R., *Zen e a Arte da Manutenção de Motocicletas - Uma Investigação sobre Valores*, Paz e Terra, 1984.

PIT 89 Pitsch, J.M., "SAMSAT - A Medium Data-Rate TDMA System", Proceedings VIII International Conference on Data Satelite Cmmunication, seção A9, 1989.

RAM 82 Ramamoorthy, C.V., F.B.Bastani, "Software Reliability - Status and Perspectives", *IEEE Transactions on Software Engineering*, vol. SE-8, número 4, Julho 1982.

RDH 76 Reliability Analysis Center, "Reliability Design Handbook", Relatório RDH número 376, 1976.

ROH 72 Rohn, W.B., T.F.Arnold, "Design for Low Expected Downtime Control Systems", Proceedings 4th International Conference on Computer Communications, Philadelphia, Junho 1972, pp.16-25.

RUS 91 Russell,G.W., "Experience with Inspection in Ultralarge-Scale Developments", *IEEE Software*, Janeiro 1991, pp.25-31.

SAB 86 Sabourin, T., K.Kanoun, "Analyse des Defaillances et Modelisation du Comportement du Logiciel d' un Autocommutateur Telephonique", Proc. 5th International Conference on Reliability and Maintainability, 1986.

SAB 87 Sabourin, T., "Evaluation de la Fiabilité du Logiciel d' un Autocommutateur Téléphonique", Tese de Doutoramento no Institut Politechnique de Toulouse, Outubro 1987.

SCH 81 Schantikumar,J.G., "A State- and Time-Dependent Error Occurrence-Rate Software Reliability Model with Imperfect Debugging", Proceedings National Computer Conference, 1981, pp. 311-315.

SCH 73 Schick, G.J. e W.Wolverton, "Assesment of Software Reliability", Proceedings Operations Research, Physica-Verlag, Wurzburg-Wien, 1973, pp. 395-422.

SCH 78 Schick, G.J. e W.Wolverton, "An Analysis of Competing Software Reliability Models", *IEEE Transactions on Software Engineering*, SE-4(2), 1978, pp.104-120.

SCH 75 Schneidewind, N.F., "Analysis of Error Processes in Computer Software", Proceedings 1975 International Conference on Reliable Software, Los Angeles, 1975, pp. 337-346.

SCH 68 Schooman, M.L., *Probabilistic Reliability, An Engineering Approach*, MCGraw-Hill, 1968.

SCH 72 Schooman, M.L., "Probabilistic Models for Software Reliability Prediction", *Statistical Computer Performance Evaluation, Academic*, New York, 1972, pp. 485-502.

SPR 85 Spreij, P., "Parameter Estimation for a Specific Software Reliability Model", *IEEE Transactions on Reliability*, vol. R-34, núBmero 4, Outubro 1985, pp. 323-328.

STA 87 Stark, G.E., "Dependability Evaluation of Integrated Hardware/Software Systems", *IEEE Transactions on Reliability*, vol. R-36, Número 4, Outubro 1987, pp. 440-444.

TAK 90 Takahashi, R. e E.T. Pereira, "Características Técnicas e Aplicações da Central TRÓPICO RA", *Revista TELEBRAS*, vol.14, número 48, Junho 1990, pp.6-9.

TOH 89 Tohma, Y. et alli, "Structural Approach to the Evaluation of the Number of Residual Software Faults Based on the Hyper-Geometric Distribution", *IEEE Transactions on Software Engineering*, vol. 15, número 3, Março 1989, pp. 345-355.

TOY 85 Toy, W.N., "Modular Redundancy Concepts, Problems and Solutions", EPRI Seminar: Digital Control and Fault-Tolerant Computer Technology, Scottsdale, Arizona, Abril 1985.

VIA 88a B.Vianna, "R&D at TELEBRAS-CPqD: The TROPICO System", Proc. Int.Conf. on Communications (ICC88), Philadelphia, Junho 1988, pp 622-626.

VIA 88b B.Vianna, E.B.Cunha,F.F.Boin, "Hardware Quality Control of TROPICO R System", Proc.Int.Conf. on Communications (ICC88), Philadelphia, Junho 1988, pp 632-636.

WAG 73 Wagoner, W.L., "The Final Report on a Software Reliability Measurement Study", Aerospace Corporation, Report TOR-0074(4112)-1, 1973.

WON 88 Wong, K.L., "Reliability Growth - Myth or Mess", *IEEE Transactions on Reliability*, vol. R-37, número 2, Junho 1988, pp. 209.

YAM 83a Yamada, S., M. Ohba e S. Osaki, "S-Shaped Reliability Growth Modeling for Software Error Detection", *IEEE Transactions on Reliability*, R-32(5), 1983, pp. 475-478.

YAM 83b Yamada, S. e S. Osaki, "Reliability Growth Models for Hardware and Software Systems Based on Non-Homogeneous Poisson Processes: a Survey",*Microelectronics and Reliability*, vol.23, número 1,1983, pp.91-112.

YAM 86 Yamada,S., H.Ohtera, H.Narihisa, "Software Reliability Growth Models with Testing Effort, *IEEE Transactions on Reliability*, vol. R-35, número 1, Abril 1986, pp. 19-23.

Anexo A

Relatório de Falha

Este formulário faz parte do procedimento de registro de falhas e de controle de correções de defeitos detectados em laboratório e em operação dos sistemas desenvolvidos no Departamento de Comutação do CPqD TELEBRAS.

Na verdade este documento diz respeito a falhas, defeitos e correções. O usuário do sistema reporta a falha detectada e envia o documento à equipe responsável pela identificação dos defeitos causadores da falha; em seguida são feitas as correções destes defeitos e o sistema é retestado. Todas estas atividades são registradas no Relatório de Falha. A cada falha detectada é gerado um único Relatório de Falha e um relatório reporta também uma única falha.

Os Relatórios de falha fazem parte de uma base de dados para cada projeto, catalogados pelo Monitor de Relatórios de falha (MONRF), que fornece relatórios variados que servem não só à verificação e validação da segurança de funcionamento como também à gerência de edições e versões do Sistema Fonte e Sistema de Produção.

TELEBRÁS
CENTRO DE P&D

RELATÓRIO DE FALHA

Nº _____

1

2 DATA DE REGISTRO: ___/___/___

REGISTRO RF

3 ELABORADOR DO RF :

4 ÁREA / GRUP. / EMPRESA / TEL. :

5 RESP. CONTROLE DO RF :

6 ÁREA / GRUP. / TEL :

7 QUANTIDADE DE ANEXOS:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

8 CARACTERIZAÇÃO:

☐ RF PROCEDENTE ☐ RF IMPROCEDENTE

DETALHAMENTO DA FALHA

9 PROTÓTIPO E CONFIGURAÇÃO ONDE A FALHA FOI DETECTADA :

10 DATA DA DETECÇÃO : ___/___/___

11 ATIVIDADE DURANTE A QUAL FOI DETECTADA A FALHA :

- ☐ PROJETO DE SISTEMA
- ☐ IMPLEMENTAÇÃO
- ☐ INTEGRAÇÃO
- ☐ TESTE DE SISTEMA
- ☐ TESTE EM CAMPO
- ☐ FABRICAÇÃO
- ☐ OPERAÇÃO

12 DESCRIÇÃO DA FALHA / CONDIÇÕES DE OCORRÊNCIA :

ANEXOS :

13 TIPO DE FALHA :

○ SOFTWARE :
- ☐ ESPECIFICAÇÃO
- ☐ CONCEPÇÃO
- ☐ CODIFICAÇÃO

○ HARDWARE :
- ☐ ESPECIFICAÇÃO
- ☐ PROJETO DE CIRCUITOS
- ☐ IMPLEMENTAÇÃO DE PROJETO

○ SISTEMA :
- ☐ ESPECIFICAÇÃO
- ☐ PROJETO

○ EMPACOTAMENTO ELETROMECÂNICO

○ DOCUMENTO

14 EFEITO DA FALHA NO SERVIÇO VISTO PELOS USUÁRIOS DO SISTEMA :

- ☐ INDISPONIBILIDADE TOTAL DA FUNÇÃO _____
- ☐ PERDA PARCIAL DA FUNÇÃO _____
- ☐ NULO

DETALHAMENTO :

ANEXOS :

15 RESPONSÁVEL

16 ÁREA / GRUP. / TEL.:

17 DATA : ___/___/___

ELABORAÇÃO DE PROP. DE SOLUÇÃO

18 DESCRIÇÃO:

PRODUTOS E DOCUMENTOS AFETADOS:

ANEXOS:

19 RESPONSÁVEL PELA FASE | 20 ÁREA/GRUP/TEL. | 21 DATA: ___/___/___

ANÁLISE

22 ANÁLISE DAS SOLUÇÕES/COMENTÁRIOS:

ANEXOS:

23 RESPONSÁVEL PELA FASE | 24 ÁREA/GRUP/TEL. | 25 DATA:

APROVAÇÃO

26 DECISÃO:

☐ APROVADO ___ SOLUÇÃO ACEITA: _____

☐ REJEITADO ___ MOTIVO: _____

27 RESPONSÁVEL PELA FASE:/RUBRICA | 28 ÁREA/GRUP./TEL. | 29 DATA: ___/___/___

TESTE

30 TESTE/COMENTÁRIOS:

ANEXOS:

31 RESPONSÁVEL PELA FASE: | 32 AREA/GRUP./TEL. | 33 DATA: ___/___/___

ENCERRAMENTO DO RF

34 ESFORÇO DISPENDIDO EM SPU:

CARACTERIZAÇÃO DA FALHA _____
DETALHAMENTO DA FALHA _____
ELABORAÇÃO DE PROPOSTAS DE SOLUÇÃO _____
ANÁLISE DAS PROPOSTAS DE SOLUÇÃO _____
APROVAÇÃO DO RF _____
IMPLEMENTAÇÃO _____
REVISÃO DA IMPLEMENTAÇÃO _____
TESTE _____
ACEITAÇÃO DE DOC E PRODUTOS ALTERADOS _____
TOTAL _____

35 RESPONSÁVEL PELA FASE: | 36 AREA/GRUP./TEL. | 37 DATA:

Anexo B

Relatório de Acompanhamento de Confiabilidade

Esta forma de Relatório de Acompanhamento tem sido utilizada no acompanhamento da segurança de funcionamento dos projetos APCC e TRÓPICO RA. Periodicamente é feita uma avaliação da confiabilidade do projeto e gerado um Relatório de Acompanhamento, que é enviado à gerência do mesmo. Em paralelo são construidos painéis com os principais resultados e gráficos do sistema, que são afixados em laboratório. O índice de um Relatório de Acompanhamento padrão é mostrado a seguir.

RELATÓRIO DE ACOMPANHAMENTO DA CONFIABILIDADE DO PROJETO DE SISTEMAS

PROJETO:
RELATÓRIO NÚMERO:
DATA:

1 - Histórico do Projeto desde o Último Relatório de Acompanhamento

2 - Análise Qualitativa do Processo de Falha

3 - Resultados da Aplicação dos Modelos de Confiabilidade

4 - Análise dos Resultados Apresentados

5 - Estudo Previsivo

6 - Ações Sugeridas

Anexo C

Coeficiente de Laplace Para Dados Acumulados a Intervalo de Tempo Fixo

A expressão do coeficiente de Tendência de Laplace u(k) quando a variável aleatória considerada é o número acumulado de eventos (falhas, no caso)é derivado como indicado em [COX 78, p. 54].

Sejam k intervalos de tempo de igual duração h e:

N : variável aleatória que representa o número de eventos ocorridos no intervalo $((i-1)h, ih], i = 1, ..., k,$

n : a realização de N , isto é, o número de eventos ocorridos no intervalo $((i-1)h, ih], i = 1, ..., k.$

A ocorrência dos eventos segue um processo de Poisson não-homogêneo com taxa dada pela expressão:

$$\lambda(t) = e^{a+bt} \tag{.1}$$

Se $b = 0$ o processo de Poisson se torna homogêneo e a taxa de ocorrência é independente do tempo.

Se observados κ intervalos de tempo a verossimilhança é:

$$L(\kappa) = \prod_{i=1}^{\kappa} P(N_i = n_i) = \prod_{i=1}^{\kappa} \frac{[\sum_{(i-1)h}^{ih} \lambda(u)du]^{n_i}}{n_i!} exp(-\int_{(i-1)h}^{ih} \lambda(u)du) \tag{.2}$$

Para a taxa de ocorrência a função de verossimilhança se torna:

$$L(\kappa) = (e^{bh} - 1)^N e^{aN+bh} \frac{\sum_{i=1}^{\kappa}(i-1)n_i}{b^N \prod_{i=1}^{\kappa} n_i!} e^{\frac{e^a(1-e^{bh\kappa})}{b}} \tag{.3}$$

onde

$$N = \sum_{i=1}^{\kappa} n_i$$

Dessa forma a função densidade de probabilidade condicional das obsevações, dado que N eventos ocorreram em κ intervalos de tempo é obtida dividindo-se a expressão anterior pela probabilidade:

$$P(\kappa) = P[\sum_{i=1}^{k} n_i = N] = \frac{[\int_0^{\kappa h} \lambda(u)du]^N}{N!} exp[-\int_0^{\kappa h} \lambda(u)du] \tag{.4}$$

O resultado da divisão só depende da variável b:

$$\frac{L(\kappa)}{P(\kappa)} = \frac{N!(e^{bh}-1)^N e^{bh}}{(e^{bh\kappa}-1)^N} \frac{\sum_{i=1}^{\kappa}(i-1)n_i}{\prod_{i=1}^{\kappa} n_i!} \tag{.5}$$

A função de verossimilhança logarítmica condicional $L(b)$, partindo da expressão acima é então:

$$L(b) = logN! - log\prod_{i=1}^{\kappa} n_i! + bh\sum_{i=1}^{\kappa}(i-1)n_i + N[log(e^{bh}-1) - log(e^{bh\kappa}-1)]$$

de modo que:

$$L'(b) = \frac{dL(b)}{db} = \begin{cases} h\sum_{i=1}^{\kappa}(i-1)n_i + N[\frac{he^{bh}}{e^{bh}-1} - \frac{\kappa he^{bh\kappa}}{e^{bh\kappa}-1}] & b \neq 0 \\ h\sum_{i=1}^{\kappa}(i-1)n_i - \frac{Nh(\kappa-1)}{2} & b = 0 \end{cases}$$

e a função de informação $I(b)$ é:

$$I(b) = E(-L''(b)) = \begin{cases} Nh^2[\frac{e^{bh}}{(e^{bh}-1)^2} - \frac{k^2e^{bh\kappa}}{(e^{bh\kappa}-1)^2}] & b \neq 0 \\ \frac{Nh^2(k^2-1)}{12} & b = 0 \end{cases}$$

Sob a hipótese nula $b = 0$ (taxa de ocorrência constante), a estatística:

$$u(\kappa) = \frac{L'(0)}{\sqrt{I(0)}} = \frac{\sum_{i=1}^{\kappa}(i-1)n_i}{N} - \frac{\kappa-1}{2}\Big/\sqrt{\frac{k^2-1}{12N}} \tag{.6}$$

segue aproximadamente uma distribuição normal com média e variância unitária.

Valores positivos de $u(\kappa)$ indicam que a estatística considerada está acima da média e sugere então $b > 0$, isto é, uma taxa de ocorrência aumentando com o tempo. Por outro lado, valores negativos de $u(\kappa)$ sugerem $b < 0$ (uma taxa de falha decrescente).

Anexo D

Estimação dos Parâmetros

A estimação dos parâmetros dos modelos de crescimento de confiabilidade é feita através de um processo de otimização que visa maximizar a sua aderência aos dados de falha coletados.

Alguns dos métodos de estimação mais utilizados são :

Método da Máxima Verossimilhança (MMV): neste método busca-se um valor para o vetor de parâmetros do modelo para o qual os dados observados são mais prováveis. A função objetivo do problema de otimização dá a verossimilhança dos dados em relação a um certo valor do vetor de parâmetros.

Método dos Mínimos Quadrados (MMQ): a função valor médio do modelo é ajustada de modo a minimizar a soma dos quadrados das distâncias verticais entre cada ponto dela e da curva de falhas observada.

Estimação Bayesiana: uma estimação anterior da função distribuição de probabilidade do vetor de parâmetros do modelo é transformada numa função distribuição de probabilidade posterior condicional aos dados observados [MEL 87, MUS 87]. Estes métodos de inferência envolvem o cálculo de integrais multidimendionais de expressões complicadas, de modo que esta técnica tem sido, na prática, limitada a alguns modelos mais simples.

Outros: foram propostos na literatura alguns outros métodos de inferência particulares a um subconjunto de modelos, como o desenvolvido por Musa e Okumoto [OKU 85].

Os dois métodos gerais têm sido estudados na teoria e na prática, e os resultados têm se mostrado equivalentes [SCH 79], com vantagem para o MMQ no caso de pequenas amostras. O MMV, porém, tem sido o mais amplamente utilizado [MEL 87, MUS 87, GOE 79,...].

Ambos os métodos gerais de inferência se utilizam de um método de otimização de sua função objetivo. No caso do MMQ o problema consiste em minimizar a distância entre a curva de falha estimada pelo modelo e a curva de falha observada enquanto que no MMV o problema é maximizar a verossimilhança dos dados observados tendo como base a curva de falha estimada. Os métodos de otimização mais utilizados são:

Procedimento iterativo de Newton-Raphson [LAN 82 :] este método necessita das expressões das primeiras e segundas derivadas do modelo em relação a todos os seus parâmetros. Consiste em determinar as raizes de equações simultâneas dadas pelas primeiras derivadas do modelo em relação a seus parâmetros. É um processo

iterativo, que usa as tangentes como aproximações da curva $y = F(x)$ a fim de determinar as raizes de $F(x) = 0$. A derivada da função, $F'(x)$, é usada para escolher a próxima aproximação da raiz, através da relação recursiva:

$$x_{i+1} = x_i + \frac{F(x_i)}{F'(x_i)} \qquad (.7)$$

O processo iterativo termina quando $x_{(i+1)} - x_i < \epsilon$, onde ϵ é o erro tolerado. A raiz é então aproximada por x_{i+1}.

Procedimento iterativo Quasi-Newton [LAN 82 :] este método exige somente a explicitação da primeira derivada do modelo em relação a seus parâmetros.

Estes dois métodos são bastante rápidos porém apresentam problemas de convergência para modelos com muitos parâmetros [MUS 87]. As expressões a otimizar são geralmente bastante complexas e quanto mais parâmetros tiver o modelo maior será o trabalho analítico para se obter as derivadas.

A velocidade de convergência dos métodos depende do valor inicial dos parâmetros, que é fornecido pelo usuário. Devido à complexidade do problema deve ser feita uma busca numérica multidimensional. O desempenho dos métodos de otimização depende também do número de parâmetros do modelo; quanto maior mais restrito é o espaço de valores iniciais dos parâmetros que levam o método a divergir [MUS 87]. Problemas com o método de busca existem (não existência do valor ótimo, empacamento em um ótimo local, unicidade dos valores dos parâmetros [ANG 80]) porém não serão estudados neste trabalho.

Para modelos a três ou mais parâmetros alguns autores preferem aplicar métodos mais lentos porém menos dependentes dos valores iniciais, como o método de Nelder-Mead nas primeiras iterações e usar o resultado como valor inical para os métodos Quasi-Newton e Newton-Raphson [MUS 87].

Neste trabalho será utilizado o Método de Máxima Verossimilhança para a estimação dos parâmetros. Como os modelos utilizados têm somente dois parâmetros a complexidade do problema analítico de cálculo das derivadas é bastante simplificado. Ainda devido à simplicidade dos modelos escolhidos pôde ser empregado o método Quasi-Newton para maximizar a verossimilhança entre a curva de falhas estimada e a real. Os raros casos em que os modelos não convergiram para uma dada base de dados foram resolvidos com uma simples troca do valor inicial dos parâmetros.

D.1 - Estimação de Máxima Verossimilhança

Seja $\beta = (\beta 0, \beta 1, ..., \beta\omega)$ o vetor de parâmetros de um modelo. O Método de Estimação de Máxima Verossimilhança se baseia na construção da função de verossimilhança, definida como a densidade conjunta de probabilidade ds dados observados Y ,

que se considera ser função dos $\omega + 1$ parâmetros dados pelo vetor β :

$$L(\beta; Y)$$

A forma da função de verossimilhança quando há somente um parâmetro β_k é mostrada na Figura E.1. Nesse caso, um valor pequeno de $L(\beta_k; Y)$ para um certo valor de β_k pode ser interpretado como sendo a observação de Y um evento raro. É então razoável escolher um valor de β_k que maximize $L(\beta_k; Y)$; isso significa escolher β_k tal que os dados observados sejam mais prováveis que qualquer outra escolha.

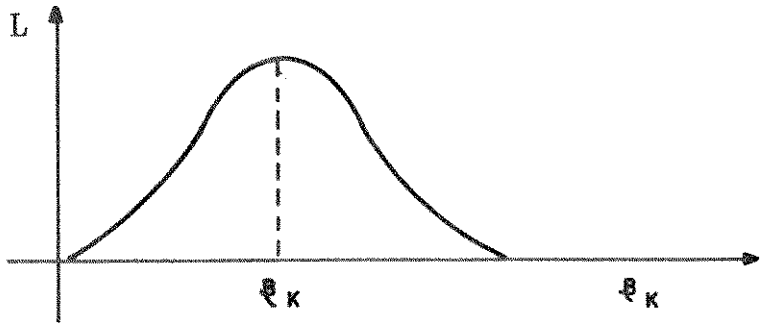

Figura D.1 - Forma da função de verossimilhança quando há somente o parâmetro β_k.

Os valores de β que fazem $L(\beta_k; Y)$ o maior possível são funções dos dados, chamadas de funções de máxima verossimilhança. Os valores que tomam essas funções são os estimadores de máxima verossimilhança.

Os estimadores de máxima verossimilhança podem ser obtidos resolvendo as seguintes equações simultâneas (uma para cada β_k):

$$\frac{L(\beta_k; Y)}{\delta \beta_k} = 0, k = 0, ..., \omega$$

As propriedades dos estimadores de máxima verossimilhança são: *consistência :* Se dois estimadores diferentes tiverem a mesma esperança , então o de menor variância é dito ser mais eficiente.

normalidade assintótica : um estimador é dito ser assintoticamente normal se a sua distribuição é quase normal para tamanhos de amostra suficientemente grandes. Por amostras suficientemente grandes entende-se aquelas em que o conteúdo de informação, que será definido adiante, se aproxima do infinito. Essa propriedade é importante pois frequentemente ela é a única alternativa disponível para se estabelecer intervalos de confiança para os parâmetros desconhecidos, como será visto mais adiante.

invariância : O estimador de máxima verossimilhança para qualquer função biunívoca de β é dada pela função de β^* .

A estimação de máxima verossimilhança fornece idênticos estimadores pontuais e intervalos para os modelos Binomiais e Poissonianos. Os estimadores para ambos os tipos de modelos são os mesmos, o que os torna indistintos do ponto de vista das duas formas de estimação.

Sejam:
p o número de dados de falha usados para a estimação dos parâmetros do modelo,

t_i a ordem das unidades de tempo t_u, sendo que $t_0 = y_0 = 0$

$y_i, i = 1, ..., p$ o número de falhas em $(0, t_i]$
a equaçao de verossimilhança a maximizar para o parâmetro β_1 do modelo Exponencial de Goel e Okumoto é:

$$L(\beta_1, Y) = \sum_{i=1}^{p} \frac{(y_i - y_{i-1})(t_i e^{-\beta_1 t_i}) - t_{i-1} e^{-\beta_1 t_{i-1}}}{e^{-\beta_1 t_{i-1}} - e^{-\beta_1 t_i}} - \frac{y_p t_p e^{-\beta_1 t_p}}{1 - e^{-\beta_1 t_p}} \qquad (.8)$$

A derivada desta função em relação ao parâmetro β_1 é dada por:

$$L'(\beta_1, Y) = \sum_{i=1}^{p} \frac{(y_i - y_{i-1})[(e^{-\beta_1 t_{i-1}} - e^{-\beta_1 t_i})(t_{i-1}^2 e^{-\beta_1 t_{i-1}} - t_i^2 e^{-\beta_1 t_i}) - (t_i e^{-\beta_1 t_i} - t_{i-1} e^{-\beta_1 t_{i-1}})^2]}{(e^{\beta_1 t_{i-1}} - e^{-\beta_1 t_i})^2}$$

$$+ \frac{y_p t_p^2 e^{\beta_1 t_p}}{(1 - e^{-\beta_1 t_p})^2} \qquad (.9)$$

A partir da maximização da função de verossimilhança obtém-se o valor ótimo do parâmetro β_1, designado por β_1^*. Pelas propriedades dos estimadores de máxima verossimilhança pode-se obter o valor ótimo do parâmetro $|beta_0$ deste modelo de:

$$\beta_0^* = \frac{y_p}{1 - e^{-\beta_1^* t_p}} \qquad (.10)$$

Para o modelo Gama a equação de verossimilhança para o parâmetro β_1 deste modelo é:

$$L(\beta_1, Y) = \sum_{i=1}^{p} \frac{(y_i - y_{i-1})(t_i^2 e^{-\beta_1 t_i} - t_{i-1}^2 e^{-\beta_1 t_{i-1}}}{[(1 + \beta_1 t_{i-1}) e^{-\beta_1 t_{i-1}} - (1 + \beta_1 t_i) e^{-\beta_1 t_i}]} - \frac{y_p t_p^2 e^{-\beta_1 t_p}}{[1 - (1 + \beta_1 t_p) e^{\beta_1 t_p}]} \qquad (.11)$$

cuja derivada em relação ao parâmetro β_1 é:

$$L'(\beta_1, Y) = \sum_{i=1}^{p} \frac{(y_i - y_{i-1})[(1 + \beta_1 t_{i-1}) e^{-\beta_1 y_{i-1}} - (1 + \beta_1 t_i) e^{-\beta_1 t_i}][t_{i-1}^3 e^{-\beta_1 t_{i-1}} - t_i^3 e^{-\beta_1 t_i}]}{[(1 + \beta_1 t_{i-1}) e^{-\beta_1 t_{i-1}} - (1 + \beta_1 t_i) e^{-\beta_1 t_i}]^2}$$

$$- \frac{\beta_1 (t_i^2 e^{-\beta_1 t_i} - t_{i-1}^2 e^{-\beta_1 t_{i-1}})^2]}{[(1 + \beta_1 t_{i-1}) e^{-\beta_1 t_{i-1}} - (1 + \beta_1 t_i) e^{-\beta_1 t_i}]^2} + \frac{y_p t_p^3 e^{-\beta_1 t_p} (1 - e^{-\beta_1 t_p})}{[1 - (1 + \beta_1 t_p) e^{-\beta_1 t_p})^2} \qquad (.12)$$

Maximizando-se a expressão de $L(\beta_1, Y)$ obtém-se o estimador ótimo β_1^*. A partir das propriedades dos estimadores de máxima verossimilhança o estimador ótimo de β_0 é dado por:

$$\beta_0^* = \frac{y_p}{1 - (1 + \beta_1^* t_p) e^{-\beta_1^* t_p}} \qquad (.13)$$

D.2 - Estimação dos Intervalos de Confiança dos Estimadores dos Parâmetros

Os estimadores pontuais dos parâmetros dos modelos são muito importantes, porém é desejável que sejam acompanhados de uma medida do erro possível na estimação, ou seja, de um intervalo no qual o valor real do parâmetro se localize com alguma medida de confiança.

Seja um modelo com um único parâmetro β_k. Pode ser mostrado [KEN 61] que o estimador de máxima verossimilhança de β_k tem distribuição assintoticamente normal com média β_k e variância $1/I(\beta_k)$, onde $I(\beta_k)$ é a informação esperada, ou de Fisher, dada por:

$$I(\beta_k) = E[-\frac{\delta^2 ln L(\beta_k; Y)}{\delta \beta_k^2}]$$

Pode-se escrever que :

$$\frac{\beta_k^* - \beta_k}{\sqrt{\frac{1}{I(\beta_k^2)}}} \sim N(0,1)$$

onde $N(0,1)$ é uma distribuição normal com média zero e variância unitária. Os limites superior e inferior de um intervalo de confiança de $100(1-\alpha)\%$ aproximado para βk são dados por :

$$\beta_k^* + ou - \frac{k_{1-\alpha/2}}{\sqrt{I(\beta_k^*)}}$$

onde $k_{1-\alpha/2}$ é o desvio normal apropriado.

O significado de I é discutível para tamanhos de amostra não suficientemente grandes [MUS 87].

Estendendo este resultado para modelos com mais de um parâmetro resulta em que β^* é aproximadamente distribuido como a distribuição normal multivariante com média β e matriz de covariância dada pelo inverso da matriz de informação de Fisher, cujos elementos são dados por:

$$I_{ij}(\beta) = E[-\frac{\delta^2 ln L(\beta; Y)}{\delta \beta_i \delta \beta_j}], i, j = 0, ..., \omega$$

Os elementos da diagonal da matriz de covariância dão a variância dos respectivos parâmetros. Os elementos fora da diagonal dão a covariância entre os elementos correspondentes.

Um intervalo de confiança para cada $\beta_k (k = 0, ..., \omega)$ pode ser estabelecido da mesma forma anterior, usando a equação da informação com $I(\beta_k^*)$ substituido por $I_{kk}(\beta^*)$. As ressalvas sobre a adequação da distribuição normal também se aplicam nesse caso.

Para os modelos Exponencial e Gama a função de informação é dada por:

$$I(\beta_1) = E[-L'(\beta_1, Y)] \tag{.14}$$

Anexo E

Bases de Dados de Falha

O sistema TRÓPICO R é uma central local/tandem com capacidade de tráfego de 320 Erl. O sistema foi desenvolvido inicialmente para atender até 1500 terminais, e ampliado para 4096 terminais em uma segunda versão. A arquitetura do sistema se caracteriza por um alto nível de distribuição, descentralização das funções de controle e grande modularidade [ARA 88, VIA 88a, b].

Por ser um sistema de pequena capacidade o TRÓPICO R é indicado preferencialmente para áreas rurais, embora possa também operar como central satélite, utilizando numeração dependente de centrais digitais de maior porte em zonas urbanas. Nesta última aplicação o sistema continua com comutação interna, facilidades de O&M locais e flexibilidade de entroncamento com diversas centrais.

O sistema tem uma estrutura hardware redundante, de modo a garantir uma degradação suave do serviço em presença de falha. O pequeno número de placas (25 tipos diferentes) facilita a produção e a manutenção.

E.1 - TRÓPICO R 1500

A coleta de dados de falha feita durante as fases de teste de sistema, teste em campo e operação comercial do TRÓPICO R não teve a finalidade de gerar dados para avaliação de segurança de funcionamento do sistema. O objetivo era o de registrar as falhas detectadas e controlar as modificações feitas no sistema. Por este motivo não foram coletadas algumas informações qualitativas importantes para o estudo da segurança de funcionamento, tais como a gravidade das falhas. Desse modo a hipótese feita é de que as falhas têm todas a mesma gravidade. A avaliação de confiabilidade de concepção foi feita post mortem, a partir dos dados na forma em que eles estavam disponíveis.

E.1.1 - Processo de Falha Software

Os dados de falha software correspondem ao período de 01/08/1984 a 20/10/1986, totalizando 81 unidades de tempo de 10 dias de calendário. A fase de teste de sistema compreende as unidades de tempo de 1 a 30, onde foram detectadas 298 falhas 9.9 em média por unidade de tempo), e a etapa de teste em campo as unidades de 31 a 42, quando foram detectadas 56 falhas (média de 4.66 por unidade de tempo). O período restante, de 43 a 81, descreve a distribuição no tempo das 112 falhas detectadas em vida operacional (2.87 em média por unidade de tempo).

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	28	288	56	397
1	4	29	292	57	400
2	5	30	298	58	402
3	35	31	301	59	411
4	47	32	302	60	416
5	60	33	311	61	418
6	72	34	317	62	422
7	86	35	319	63	423
8	96	36	323	64	426
9	107	37	324	65	433
10	114	38	338	66	445
11	122	39	342	67	449
12	132	40	345	68	454
13	137	41	350	69	459
14	141	42	354	70	461
15	143	43	356	71	461
16	153	44	367	72	461
17	166	45	373	73	462
18	175	46	374	74	463
19	186	47	379	75	464
20	196	48	383	76	464
21	204	49	385	77	464
22	214	50	386	78	465
23	223	51	386	79	465
24	248	52	389	80	465
25	261	53	389	81	466
26	278	54	386		
27	283	55	390		

Tabela E.1 - Falhas do software do TRÓPICO R versão 1500.

E.1.2 - Processo de Falha Hardware

As falhas hardware continuaram sendo registradas até 10/04/1987 e assim a base de dados é mais extensa, contendo 98 unidades de tempo, durante as quais foram detectadas 607 falhas.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	33	241	66	534
1	0	34	253	67	541
2	1	35	286	68	545
3	20	36	295	69	549
4	24	37	318	70	556
5	32	38	330	71	559
6	36	39	334	72	561
7	48	40	345	73	566
8	55	41	352	74	569
9	75	42	354	75	572
10	78	43	362	76	575
11	80	44	379	77	578
12	85	45	388	78	579
13	88	46	394	79	584
14	93	47	408	80	587
15	95	48	413	81	590
16	101	49	415	82	593
17	102	50	420	83	593
18	106	51	420	84	597
19	115	52	420	85	597
20	118	53	422	86	598
21	112	54	423	87	598
22	130	55	427	88	599
23	130	56	431	89	599
24	151	57	467	90	599
25	156	58	476	91	599
26	181	59	486	92	600
27	204	60	495	93	601
28	213	61	498	94	602
29	215	62	507	95	603
30	220	63	509	96	603
31	222	64	514	97	604
32	235	65	522	98	607

Tabela E.2 - Falhas do Hardware do TROPICO R 1500

E.2 - TRÓPICO R 4096

A versão TROPICO R para 4096 teminais é uma extensão da versão 1500. O hardware é o mesmo, porém os módulos software sofreram modificações substanciais: dos 29 módulos 16 foram alterados, um dos módulos foi particionado em dois, e dois novos módulos foram criados, enquanto que 12 (na sua maioria módulos do sistema operacio-

nal) não sofreram alterações.

Como as duas versões conviveram durante algum tempo no mercado e na produção consideramos dois Sistemas de Produção distintos. As falhas foram registradas em bases de dados diferentes, sendo que o registro de uma falha em módulos comuns aos dois sistemas eram duplicados nas duas bases de dados.

Após algum tempo a TELEBRAS, as indústrias responsáveis pela produção dos sistemas e as Operadoras do Sistema TELEBRAS, representando os usuários, decidiram interromper a fabricação da versão 1500.

O software da maioria das Instalações do TRÓPICO R 1500 foi substituido pala versão 4096, porém o sistema 1500 ainda se encontra operacional em todo o país.

E.2.1 - PROCESSO DE FALHA SOFTWARE

A tabela E.3 enumera as falhas detectadas no software do TRÓPICO R 4096 no período de 01/02/1986 a 22/09/1988. Nesse caso os dados se referem à fase de teste de sistema (unidades de 1 a 19, no qual 69 falhas foram detectadas) e de vida operacional (unidade de tempo 20 em diante, com 134 falhas detectadas). O sistema não passou pela fase de teste formal em campo.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	31	122	62	186
1	1	32	125	63	187
2	2	33	129	64	187
3	2	34	130	65	187
4	2	35	132	66	187
5	2	36	134	67	187
6	3	37	134	68	187
7	4	38	135	69	187
8	7	39	138	70	187
9	9	40	140	71	188
10	19	41	147	72	190
11	29	42	150	73	190
12	33	43	152	74	190
13	38	44	155	75	190
14	42	45	159	76	191
15	47	46	161	77	191
16	52	47	163	78	194
17	57	48	164	79	194
18	64	49	165	80	194
19	69	50	167	81	195
20	72	51	168	82	202
21	75	52	173	83	202
22	78	53	176	84	203
23	84	54	178	85	203
24	88	55	178	86	203
25	94	56	178	87	203
26	99	57	179	88	203
27	108	58	180	89	203
28	112	59	182	90	203
29	114	60	182		
30	119	61	186		

Tabela E.3 : Falhas do Software do TROPICO R versão 4096

E.2.2 - PROCESSO DE FALHA HARDWARE

As falhas do hardware do 4096 foram observadas no mesmo período do software, perfazendo um total de 122.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	31	50	62	92
1	4	32	50	63	93
2	6	33	51	64	93
3	6	34	52	65	93
4	6	35	55	66	94
5	7	36	55	67	94
6	9	37	55	68	97
7	9	38	55	69	99
8	13	39	57	70	102
9	14	40	60	71	102
10	16	41	65	72	102
11	21	42	66	73	102
12	22	43	66	74	105
13	28	44	69	75	106
14	28	45	70	76	108
15	31	46	71	77	111
16	32	47	72	78	113
17	32	48	74	79	113
18	33	49	78	80	114
19	35	50	82	81	116
20	35	51	82	82	119
21	37	52	82	83	121
22	38	53	82	84	121
23	42	54	88	85	121
24	42	55	88	86	122
25	44	56	89	87	122
26	44	57	89	88	122
27	45	58	89	89	122
28	46	59	90	90	122
29	46	60	90		
30	50	61	90		

Tabela E.4 : Falhas do Hardware do TRÓPICO R versão 4096.

E.3 - AMBIENTE DE PROGRAMAÇÃO CHILL DO CPqD

O projeto CHILL foi criado com o objetivo de implantar e manter no CPqD um ambiente de programação orientado ao desenvolvimento de software baseado na linguagem de programação de alto nível CHILL (Ambiente de Programação CHILL do CPqD - APCC). A linguagem CHILL é o padrão internacional definido e recomendado pelo CCITT para a produção de software destinado a sistemas de telecomunicações. O APCC é constituido por um conjunto integrado de ferramentas software e seu objetivo principal é prover o suporte necessário à atividade de desenvolvimento do software destinado aos equipamen-

tos e sistemas de telecomunicações desenvolvidos no CPqD [PAL 90].

O APCC é um ambiente que opera numa arquitetura distribuída baseada em máquinas hospedeiras e máquinas alvo. As máquinas hospedeiras do APCC - VAXes com VMS e estações de trabalho com sistema operacional UNIX- executam as atividades de gerência e manutencção de banco de dados de projeto, controle de versões de módulos, controle de configurações de programas, subsistema e sistemas, compilação, ligação, etc... As máquinas alvo do APCC são construidas com estrutura-padrão baseada em processadores de 16 bits e usadas para teste, depuração e integração dos programas desenvolvidos. O projeto foi iniciado em janeiro de 1984.

O compilador, o formatador, o montador, os ligadores o administrador de bibliotecas de módulos objeto e o tradutor Pascal-CHILL são as ferramentas do APCC que encontram-se operacionais nas máquinas hospedeiras. O monitor CHILL, o núcleo do sistema operacional SO88, as bibliotecas de suporte à execução de programas e o depurador simbólico são as ferramentas do APCC operacionais nas máquinas-alvo.

A versão corrente do APCC foi e/ou vem sendo usada no desenvolvimento do software básico e aplicativo dos projetos TRÓPICO RA, COM (Centralizado de Operação e Manutenção, já concluído) e CETEX (Central Comutadora de Pacotes). Não se tem idéia de quantas instalações estão operacionais, mas estima-se que cerca de 250 projetistas já tenham usado o APCC.

E.3.1 - PROCESSO DE FALHA SOFTWARE

Os dados de falha se referem ao período de 1/04/1987 a 30/03/1990, correspondente a:

- utilização do Ambiente em projetos de desenvolvimento no CPqD: equivalente a operação comercial
- teste do sistema em laboratório, pela equipe que o desenvolveu.

Os dados de falha na forma de total acumulado a cada 10 dias de calendário são mostrados na tabela D.8 a seguir. A evolução do número de usuários no tempo é desconhecida, de modo que assume-se uma intensidade constante de teste.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	40	380	80	515
1	24	41	383	81	518
2	32	42	396	82	521
3	42	43	403	83	527
4	55	44	408	84	532
5	90	45	413	85	538
6	102	46	416	86	540
7	112	47	421	87	541
8	156	48	422	88	543
9	169	49	426	89	548
10	175	50	429	90	549
11	177	51	436	91	550
12	186	52	439	92	552
13	200	53	440	93	552
14	204	54	442	94	554
15	213	55	446	95	556
16	217	56	452	96	557
17	223	57	454	97	561
18	229	58	455	98	562
19	238	59	463	99	564
20	242	60	464	100	565
21	252	61	467	101	565
22	259	62	469	102	566
23	279	63	472	103	566
24	296	64	474	104	568
25	300	65	476	105	568
26	309	66	481	106	569
27	316	67	482	107	571
28	317	68	484	108	572
29	320	69	487	109	572
30	327	70	493	110	572
31	330	71	496	111	574
32	335	72	499	112	574
33	339	73	500	113	575
34	341	74	503	114	576
35	349	75	504	115	579
36	359	76	506	116	579
37	362	77	508	117	581
38	372	78	510		
39	377	79	512		

Tabela E.5 - Falhas do Ambiente de Programação CHILL do CPqD.

E.4 - Sistema de Comutação Digital TRÓPICO RA

O TRÓPICO RA é mais um elemento da família de equipamentos digitais de comutação telefônica controlados por programas armazenados desenvolvida no CPqD TELEBRAS [HUG 90].

Além da capacidade maior (de 2 mil a 100 mil terminais) este equipamento foi projetado para oferecer uma série de novos serviços que os elementos anteriores não forneciam [TAK 90], tais como: sinalização por canal comum, serviços suplementares, facilidades operacionais adicionais e possibilidade de ser um nó da Rede Digital de Serviços Integrados (RDSI).

A estrutura hardware com processamento distribuído e redundância ativa das partes críticas operando em partição de carga, tem estrutura modular que proporciona grande flexibilidade para atender aos avanços futuros de tecnologia [DIA 88].

O sistema armazena um código executável de 6.5 megabytes, que também tem estrutura modular e pode ser expandido em novas funções, bastando para isso desenvolver os módulos a elas correspondentes. O software é escrito na linguagem CHILL, que, por ser isolada, é independente do processador, o que também é um atributo importante na provável evolução tecnológica do sistema. Para o desenvolvimento do software foram utilizadas as facilidades oferecidas pelo Ambiente de Programacção em CHILL do CPqD (APCC).

O desenvolvimento do sistema TRÓPICO RA está sendo feito em etapas, nas quais as funções irão sendo incorporadas a uma versão inicial, à qual correspondem os dados de falha apresentados neste trabalho. O período de observação vai de 28/07/1989 a 30/07/1990. Os dados de t=1 a t=44 dizem respeito à fase de Testes de Integração Hardware/Software e de t=45 em diante à fase de Testes de Sistema.

E.4.1 - Processo de Falha Software

Os dados de falha software estão mostrados na Tabela D.6, na forma de número acumulado de falhas ao final de cada semana de teste.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	17	50	34	84
1	0	18	52	35	85
2	2	19	53	36	85
3	4	20	53	37	86
4	7	21	55	38	89
5	10	22	55	39	103
6	14	23	58	40	107
7	16	24	59	41	120
8	17	25	62	42	120
9	23	26	64	43	140
10	31	27	66	44	150
11	37	28	67	45	166
12	42	29	68	46	169
13	42	30	71	47	209
14	42	31	74	48	265
15	42	32	75	49	330
16	50	33	77		

Tabela E.6- Dados de falha do software do sistema de comutação digital TRÓPICO RA.

E.4.2 - Processo de Falha Hardware

Os dados de falha de concepção hardware estão mostrados na Tabela E.7.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	17	18	34	40
1	4	18	19	35	41
2	4	19	21	36	44
3	4	20	22	37	46
4	5	21	23	38	51
5	7	22	23	39	52
6	7	23	24	40	52
7	8	24	25	41	54
8	10	25	25	42	55
9	10	26	26	43	55
10	12	27	26	44	55
11	12	28	26	45	56
12	12	29	27	46	57
13	14	30	34	47	58
14	16	31	36	48	58
15	17	32	38	49	59
16	18	33	38		

Tabela E.7- Dados de falha de concepção do Hardware do sistema de comutação digit
TRÓPICO RA.

Os dados de falha deste sistema têm sido registrados através do Relatório de Falhas apresentado no Anexo A. O acompanhamento de confiabilidade tem sido feito através de avaliações mensais, cujos resultados são apresentados em Relatórios de Acompanhamento como o mostrado no Anexo B.

E.5 - Sistema SAMSAT

O Sistema de Acesso Múltiplo por Divisão no Tempo Via Satélite (SAMSAT) foi desenvolvido pelo CPqD TELEBRAS, com a participação de indústrias nacionais, a partir de especificações elaboradas pela EMBRATEL [GON 86].

O SAMSAT é constituido por até 126 Estações Terminais Satélites (ETS) e por duas Estações de Referência: uma primária (ER0) e outra secundária (ER1) [PIT 89]. Cada ETS comporta até 16 unidades de linha, através das quais os usuários têm acesso aos seguintes serviços:
-Formação de redes privativas
-Entroncamento entre redes públicas
-Formação de redes públicas de comunicação de dados
-Acesso de usuários às redes públicas
-Entroncamento telefônico entre centrais e fornecimento de canal de serviço

As Estações de Referência são responsáveis pela execução centralizada das funções de controle e supervisão. A operação destas funções é redundadnte nas duas estações, segundo os princípios de redundância ativa, onde ER0 é preferencialmente a referência. A ER0 centraliza também as funções de operação do sistema, através de um conjunto de equipamentos de apoio, responsáveis pela interface homem/máquina, pela geração dos dados de configuração e pela impressão de relatórios operacionais e de supervisão.

A avaliação de confiabilidade do sistema foi feita em dois momentos: o primeiro em julho de 1989, quando o produto se encontrava em testes de sistema e a segunda em março de 1990, quando se aguardava o início dos testes em campo.

E.5.1 - Processo de Falha Software

A seguir são apresentados os dados de falha software relativos ao período de 7/6/1989 a 28/12/1989. Os dados de falha foram acumulados a cada dia útil, conforme mostrado na Tabela E.8 a seguir.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	28	8	56	15
1	1	29	8	57	15
2	1	30	10	58	15
3	1	31	10	59	15
4	1	32	10	60	15
5	1	33	11	61	15
6	4	34	11	62	15
7	4	35	12	63	15
8	5	36	12	64	15
9	5	37	12	65	15
10	5	38	12	66	15
11	5	39	12	67	15
12	5	40	12	68	15
13	5	41	12	69	15
14	5	42	12	70	15
15	5	43	12	79	15
16	6	44	12	72	15
17	6	45	12	73	15
18	7	46	12	74	15
19	7	47	12	75	15
20	7	48	14	76	15
21	7	49	14	77	15
22	7	50	14	78	15
23	7	51	14	79	15
24	7	52	14	80	15
25	7	53	14		
26	7	54	15		
27	8	55	15		

Tabela E.8 - Dados de falha do software do sistema SAMSAT.

E.5.2 - Processo de Falha Hardware

Os dados de falha coletados entre 7/6 e 28/12/1989 são mostrados na Tabela E.9 a seguir.

Os testes de sistemas foram organizados funcionalmente seguindo o seguinte cronograma de aplicação ao protótipo de laboratório:
-Geração de Arquivos de Dados: t=47 a t=51
-Sincronismo das Estações: t=0 a t=3
-Comandos Operacionais e de Consulta: t=4 a t=18
-Comunicação entre Estações: t=19 a t=24
-Mecanismos de Supervisão: t=25 a t=47
-Verificação de Configuração do Sistema após uma Reconfiguração da ER0: t=52 a t=54

-Tráfego e Integridade de Dados: t=55 a t=80

Os testes até t=54 impunham um esforço adicional de operação ao sistema. Na última bateria de testes o esforço foi reduzido ao normal.

Unidade de Tempo	Falhas	Unidade de Tempo	Falhas	Unidade de Tempo	Falhas
0	0	28	4	56	11
1	0	29	4	57	11
2	0	30	6	58	11
3	0	31	9	59	11
4	0	32	9	60	11
5	0	33	9	61	11
6	0	34	9	62	11
7	1	35	9	63	11
8	1	36	9	64	11
9	1	37	9	65	11
10	1	38	9	66	11
11	1	39	9	67	11
12	1	40	9	68	11
13	1	41	9	69	11
14	1	42	9	70	11
15	1	43	9	79	11
16	1	44	10	72	11
17	1	45	11	73	11
18	1	46	11	74	11
19	1	47	11	75	11
20	1	48	11	76	11
21	1	49	11	77	11
22	1	50	11	78	11
23	1	51	11	79	11
24	1	52	11	80	11
25	1	53	11		
26	1	54	11		
27	4	55	11		

Tabela E.9 - Dados de falha de concepção do hardware do sistema SAMSAT.