

Relatório de Avaliação do Software do TSE realizada pela Fundação COPPETEC

Ana Regina Cavalcanti da Rocha
Guilherme Horta Travassos
Gleison Santos Souza
Sômulo Mafra

Programa de Engenharia de Sistemas e Computação
COPPE/UFRJ

9 de agosto de 2002

I. Introdução e Conceitos Fundamentais

Controle da Qualidade de Software reúne um conjunto de atividades que verificam e garantem uma série de características de qualidade que devem estar presentes no produto para que este atenda às necessidades de seus usuários. Entende-se por produto de software o código e sua documentação.

A norma ISO 9126 estabelece e define um conjunto de características e sub-características de qualidade que estão listadas a seguir:

- i. **Funcionalidade:** conjunto de atributos que evidenciam a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas ou implícitas¹.

Subcaracterísticas:

- **Adequação:** atributos do software que evidenciam a presença de um conjunto de funções de sua apropriação para as tarefas especificadas;
- **Acurácia:** atributos do software que evidenciam a geração de resultados ou efeitos corretos ou conforme acordados;
- **Interoperabilidade:** atributos do software que evidenciam sua capacidade de interagir com sistemas especificados;
- **Conformidade a padrões:** atributos do software que fazem com que ele esteja de acordo com as normas, convenções ou regulamentações previstas em leis e descrições similares, relacionadas à aplicação;
- **Segurança de acesso:** atributos do software que evidenciam sua capacidade de evitar o acesso não autorizado, acidental ou deliberado, a programas e dados.

¹ Este conjunto de atributos caracteriza o que o software faz para satisfazer as necessidades, enquanto que os outros conjuntos caracterizam principalmente quando e como ele faz.

- ii. **Confiabilidade:** conjunto de atributos que evidenciam a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo estabelecido².

Subcaracterísticas:

- **Maturidade:** atributos do software que evidenciam a frequência de falhas por defeitos no software;
- **Tolerância a falhas:** atributos do software que evidenciam sua capacidade em manter o nível de desempenho³ especificado nos casos de falhas no software ou de violação nas interfaces especificadas⁴.
- **Recuperabilidade:** atributos do software que evidenciam sua capacidade de estabelecer seu nível de desempenho e recuperar os dados diretamente afetados, em caso de falha, e no tempo e esforço necessários para tal.

- iii. **Usabilidade:** conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso por um conjunto explícito ou implícito de usuários.

Subcaracterísticas:

- **Inteligibilidade:** atributos do software que evidenciam o esforço do usuário para reconhecer o conceito lógico e sua aplicabilidade;
- **Apreensibilidade:** atributos do software evidenciam o esforço do usuário para aprender sua aplicação (por exemplo: controle de operação, entradas, saídas);
- **Operacionalidade:** atributos do software que evidenciam o esforço do usuário para operação e controle de sua operação.

- iv. **Eficiência:** conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas⁵.

Subcaracterísticas:

- **Comportamento em relação ao tempo:** atributos do software que evidenciam seu tempo de resposta, tempo de processamento e velocidade na execução de suas funções;

² Em software não ocorre desgaste ou envelhecimento. As limitações em confiabilidade são decorrentes de defeitos na especificação dos requisitos, projeto ou implementação. As faltas decorrentes desses defeitos dependem de como o produto de software é usado e das opções de programa selecionadas e não do tempo decorrido.

³ Na especificação do nível de desempenho, deverá ser incluída a capacidade de suportar falhas sem comprometer a segurança

⁴ Um exemplo de violação da interface é a entrada de valores fora das faixas especificadas.

⁵ Os recursos podem incluir outros produtos de software, hardware, materiais (por exemplo: papel para impressora, discos flexíveis) e serviços de operação, manutenção ou suporte.

- **Comportamento em relação aos recursos:** atributos do software que evidenciam a quantidade de recursos usados e a duração de seu uso na execução de suas funções.

v. **Manutenibilidade:** conjunto de atributos de que evidenciam o esforço necessário para fazer modificações específicas no software.

Subcaracterísticas:

- **Analisabilidade:** atributos do software que evidenciam o esforço necessário para diagnosticar eficiências ou causas de falhas, ou para identificar partes a serem modificadas;
- **Modificabilidade:** atributos do software que evidenciam o esforço necessário para modificá-lo, remover seus defeitos ou adaptá-lo a mudanças ambientais;
- **Estabilidade:** atributos do software que evidenciam o risco de efeitos inesperados, ocasionados por modificações;
- **Testabilidade:** atributos do software que evidenciam o esforço necessário para validar o software modificado.

vi. **Portabilidade:** conjunto de atributos que evidenciam a capacidade do software ser transferido de um ambiente para outro⁶.

Subcaracterísticas:

- **Adaptabilidade:** atributos do software que evidenciam sua capacidade de ser adaptado a ambientes diferentes especificados, sem necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado;
- **Capacidade para ser instalado:** atributos do software que evidenciam o esforço necessário para sua instalação em um ambiente especificado;
- **Conformidade:** atributos do software que o tornam consoante com padrões ou convenções relacionadas à portabilidade;
- **Capacidade para substituir:** atributos do software que evidenciam sua capacidade e esforço necessário para substituir um outro software, no ambiente estabelecido para este outro software.

II. Processo de Avaliação Adotado

O processo de avaliação adotado foi constituído de quatro estágios: *Identificação do Objeto da Avaliação, Definição de Requisitos de Qualidade do Produto, Preparação da Avaliação e Realização da Avaliação*. Estes estágios correspondem a:

- **Identificação do Objeto da Avaliação**

Foram objeto de avaliação:

- O código fonte dos sistemas;
- A documentação dos sistemas, conforme disponibilizada pelo TSE.

⁶ O ambiente pode incluir ambiente organizacional, de hardware ou de software.

- **Definição de Requisitos de Qualidade do Produto**

O conjunto de características identificado como necessário para o código foi o conjunto de características e sub-características da Norma ISO 9126.

Para avaliação da documentação foram consideradas as seguintes características: completude, legibilidade, consistência, organização da documentação, capacidade de apoiar a manutenção do produto, capacidade de apoiar o entendimento da funcionalidade do produto, uso de técnicas adequadas de garantia de qualidade. Com relação ao uso destas técnicas, teve-se o objetivo de avaliar se foram utilizadas técnicas de revisão e testes sistemáticos adequados para assegurar a qualidade do produto.

Para avaliação do código fonte foram considerados os aspectos gerais que indicam algumas das propriedades do software, tais como estilo de programação, documentação interna, modularização, coesão, tratamento de erros, estruturação, existência de código inativo, segurança, interface, presença de código inativo e complexidade estrutural. Com relação à avaliação de complexidade estrutural, teve-se o objetivo de avaliar se existiu preocupação com o projeto arquitetural da aplicação e a identificação dos módulos com maior possibilidade de defeito, que poderiam comprometer a execução da aplicação.

- **Preparação da Avaliação**

Neste estágio, foram preparadas as bases para a avaliação o que envolveu:

- *Definição dos níveis de pontuação:*

O código do sistema foi avaliado segundo as características e sub-características da ISO 9126. Para cada característica o resultado da avaliação resultou na atribuição um valor na seguinte escala:

0	1	2	NA
---	---	---	----

onde: 0 – Não atende
1 – Atende parcialmente
2 – Atende
NA – Não se Aplica

Caso o avaliador não se considerasse em condições de avaliar uma determinada característica, deveria marcar a opção Naval (não Avaliado).

A documentação de cada sistema aplicativo foi avaliada quanto aos requisitos de qualidade definidos. O resultado da avaliação resultou na atribuição de uma das seguintes respostas às questões estabelecidas:

() Sim () Parcialmente () Não

- *Definição da ferramenta de apoio à avaliação de código:*

Foi selecionada a ferramenta *Understand for C++* da *Scientific Toolworks Inc.*, inicialmente licenciada para uso e avaliação pela COPPE/UFRJ, e obtida posteriormente também pelo TSE, que a disponibilizou em todos os equipamentos para medição do produto. Esta ferramenta objetiva a medição de produtos de software desenvolvidos em C/C++, linguagens utilizadas para construir parte das aplicações do sistema. Dentre as métricas extraídas pela ferramenta destas aplicações foram principalmente consideradas as que representam Complexidade Ciclomática, Porcentagem de código inativo, Porcentagem de Comentário, Tamanho do Produto (linhas de código), Perda de Coesão em Classe (LCOM), Resposta da Classe (RFC), Número de Filhos de uma classe (NOC), Profundidade de herança (DIT), Métodos da Classe (WCOM) já que existe evidência entre a combinação dos valores destas medidas com a possibilidade de falha de um software. Não foi possível a obtenção de ferramenta correspondente para medição das aplicações desenvolvidas em PASCAL (Delphi), razão pela qual as medidas não foram coletadas para esta parte do sistema.

- *Definição de formulários para registro das avaliações:*

Foram definidos um formulário para registro da avaliação da documentação (Anexo I) e dois formulários para registro da avaliação do código (Anexo II). Foram, também, definidos formulários para resumo das avaliações (Anexo III).

- **Realização da Avaliação**

Este estágio foi composto de três passos:

- *Medição:* cada produto (módulo do sistema/documentação do sistema) foi avaliado e o resultado foi expresso em valores na escala estabelecida, nos formulários dos Anexos I e II;
- *Síntese:* O conjunto de níveis pontuados foi sintetizado em tabelas resumo conforme os formulários do Anexo III
- *Julgamento:* O resultado da avaliação foi analisado e emitido um julgamento sobre a qualidade do produto bem como recomendações para sua melhoria.

A técnica utilizada foi Inspeção de Software, uma técnica de leitura de código com critérios de avaliação pré-definidos.

III. Resultados da Avaliação

A avaliação foi realizada por um grupo de quatro avaliadores do Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, contratados pelo Partido dos Trabalhadores (PT) através da Fundação COPPETEC: Profa. Ana Regina Cavalcanti da Rocha, Prof. Guilherme Horta Travassos e pesquisadores Gleison Santos de Souza e Sômulo Nogueira Mafra.

Os estágios de *Identificação do Objeto da Avaliação*, *Definição de Requisitos de Qualidade* e *Preparação da Avaliação* foram realizados previamente à semana de avaliação. O estágio *Realização da Avaliação*, no que se refere à medição, foi realizado de 5 a 8 de agosto nas dependências do TSE em Brasília. Os dias 8 e 9 de agosto foram, também, dedicados a atividades de síntese e julgamento e elaboração deste Relatório.

1. Avaliação da Documentação do Software

No que se refere à documentação do software foram analisados todos os documentos colocados à disposição na sala do TSE, preparada para a avaliação pelos partidos políticos.

Estes documentos referem-se a:

1. Sistema de Gerenciamento de Zona Eleitoral
2. Sistema Montador de Dados
3. Sistema Gerador de Mídias
4. Sistema de Recebimento de Boletins de Urna
5. Sistema de Totalização
6. Aplicativo de Voto em Terminais Múltiplos
7. Visualizador Validador de Fotos – VVFoto
8. Verificação Pré e Pós Eleição – VERPRE e VERPOS
9. Ajuste de Data e Hora Seguro – ADHSEG
10. Auto Teste para VE2002
11. Visualizador de Log – VLOG
12. Recuperador de Dados da Eleição – RED
13. Sistema de Conf. Dos Espelhos de Voto – SCEV
14. Sistema de Testes Exaustivos VE2002 e STE2002
15. Driver Hot Swap para Unidade de Flash Card
16. Plano de Desenvolvimento de Software (de software desenvolvido pela UNISYS)
17. Diretórios de Dados dos Arquivos da UE
18. Compactação de Arquivos
19. Software Básico
20. Software de Carga da UE
21. Gerenciamento de Aplicativos – GAP
22. Aplicativo de Voto
23. Aplicativo de Justificativa

Os documentos analisados podem ser divididos em dois grupos. No primeiro grupo estão os documentos relativos aos seguintes sistemas: Sistema de Gerenciamento de Zona Eleitoral, Sistema Montador de Dados, Sistema Gerador de Mídias, Sistema de Recebimento de Boletins de Urna e Sistema de Totalização. No segundo grupo estão todos os demais documentos listados acima. Os documentos de um mesmo grupo possuem características comuns quanto ao conteúdo, organização e qualidade. A seguir fazemos comentários gerais sobre os documentos apresentados. A avaliação detalhada de cada documento pode ser encontrada no Anexo IV.

Pode-se observar após a análise do *primeiro grupo de documentos*:

- É dito no início destes documentos que foi utilizada a metodologia CTM/IS da empresa CTIS, contratada pelo TSE. Não havia entre os documentos disponibilizados um que descrevesse a metodologia. Foi solicitado ao TSE esta descrição, que foi então analisada. Pode-se então afirmar com relação à metodologia CTM/IS:
 - Trata-se de uma metodologia incompleta e em alguns aspectos ultrapassada e incoerente.
 - A metodologia não tem procedimentos claramente estabelecidos para garantia da qualidade do produto
 - A metodologia não foi seguida de forma completa
 - Informou-se que a empresa deveria customizar a metodologia para o TSE, mas que o TSE não possui esta customização documentada.
- É dito também que as especificações seguiram o padrão ISO 9001. Na realidade esta norma não contém padrão de especificação ao menos se refere a software.
- As especificações são inadequadas, incompletas e inconsistentes.
- A fase de projeto do software parece não ter sido realizada. O que poderia ser considerado projeto de aplicação, o Diagrama Hierárquico de Funções, de fato não é um diagrama hierárquico de funções e não pode ser considerado projeto do software. Deficiências no produto por falhas de projeto foram, posteriormente, comprovadas ao realizarmos a análise do código.
- Não existiram procedimentos de garantia da qualidade do produto ao longo do desenvolvimento ou estes não foram documentados.
- Não existiram procedimentos de gerência de configuração ou estes não foram documentados.
- Não existiram procedimentos de teste sistemático do produto ou estes não foram documentados.

O que se diz na documentação sobre testes é praticamente o mesmo em todos os documentos, no item Laudo de Testes: “*Nos testes realizados pelo analista (primeiro nível), pelos arquitetos de testes (segundo nível) e pelos usuários do sistema (terceiro nível) foram detectados erros de codificação, erros de especificação, ajustes de layout e ajustes da funcionalidade de telas e relatórios*”. Esta frase não significa absolutamente nada, especialmente vendo-se que é uma frase padrão nos documentos referentes aos vários sistemas. Não há qualquer outra informação sobre os testes realizados. A documentação remete a um anexo dizendo poder-se encontrar aí as ocorrências de testes, mas em todos os documentos o local referente a estas ocorrências está totalmente em branco.
- Não existem na documentação Planos de Teste, documento essencial, que deveria conter os tipos de testes realizados, casos de teste para cada tipo de teste, os resultados esperados e obtidos, bem como o grau de confiabilidade alcançado. Não há indicação de como foi feita a homologação interna do software, nem de quem aprovou o produto no TSE.
- Por nossa solicitação nos foi mostrada uma documentação de planejamento e resultados de teste em campo. Entretanto nossa análise não pode ser conclusiva sobre estes testes, pois a documentação dos testes é incompleta e não foi possível examinar sua adequação e cobertura. Também não nos foi possível examinar os resultados obtidos.

Pode-se observar após a análise do *segundo grupo de documentos*:

- É apresentado um Plano de Desenvolvimento de Software para o conjunto de produtos desenvolvido pela UNISYS.
- Para Compactação de Arquivos, o único documento apresentado é um Guia de Programação.
- Para todos os demais aplicativos deste grupo, sob a responsabilidade da UNISYS, a única documentação apresentada foi a especificação do produto.
- Foi solicitado o resto da documentação mas foi dito ser esta a documentação existente.
- Faz-se referência à existência de um Plano de Testes para cada aplicativo, mas estes não foram disponibilizados.
- Há alguns absurdos na documentação: “os códigos fonte atenderão à técnica de programação orientada a objetos utilizando a notação UML completa”. UML é uma linguagem de modelagem para especificação, análise e projeto que não foi utilizada na especificação do produto. Não pode ser utilizada para código.
- Faz-se referência ao uso de Palm (SREPalM) que não estava disponível para avaliação.
- Nos documentos Especificação do Produto são indicados sempre outros documentos de referência, que não se encontravam disponíveis.
- As descrições constantes no histórico de revisões são muito sucintas, deveria haver justificativa do porquê do problema e suas conseqüências. Ex.: “Correção de informações referentes ao código verificador da justificativa eleitoral.” Em que página que descreve onde problema ocorreu? Que testes foram realizados para detectá-lo e para verificar a correção?
- Durante o desenvolvimento foram utilizados vários compiladores, linguagens e bancos de dados diferentes. Seria bom haver um documento justificando tais opções.
- Quais as diferenças entre os sistemas de testes e o oficial? Não é em todos os documentos que isso fica claro. Também não fica claro como essas modificações são realizadas no código (diretivas de compilação? Parâmetros?) e como essas modificações são refletidas na documentação como um todo. Seria adequado padronizar.
- Sistemas com acesso a Banco de Dados: Não há modelo de dados para tais sistemas. Há uma única descrição das tabelas dos bancos de dados mas não há modelos de entidade relacionamento, apenas uma listagem de tabelas e respectivos campos. Não há como garantir que a solução adotada seja adequada.
- Em alguns lugares do sistema é dito que alguns arquivos são transferidos via FTP. Não fica claro na documentação se há alguma forma de proteção adicional (tipo firewall). No protocolo ftp as senhas trafegam pela rede sem criptografia. Poderia ser estudada alguma outra forma de fazer essa transferência (tipo SecureFTP).

A análise da documentação do produto mostra a forma como este foi desenvolvido. Processos de desenvolvimento de software de qualidade são indicadores de produtos de boa qualidade.

Como os representantes dos partidos não podiam testar o software, a análise do processo de desenvolvimento utilizado poderia se constituir um indicador de confiabilidade do software das Eleições 2002. No entanto, o que se pode observar na documentação não indicou o uso

de um processo adequado de desenvolvimento e garantia da qualidade. Não há registros sobre os testes realizados, nem sobre os índices de confiabilidade do produto.

Concluimos, portanto, que a forma como o software parece ter sido desenvolvido, isto é, o que se pode deduzir da documentação colocada para exame, não garante que este tenha a qualidade esperada e necessária. Foi utilizado um processo de software bastante *ad-hoc* e imaturo, o que em geral conduz a produtos de qualidade imprevisível, fortemente dependentes de características pessoais dos desenvolvedores.

Vários documentos fazem referência a datas de término da codificação. Estas datas mostram que a codificação ultrapassou a data de avaliação dos partidos. Evidenciam que o produto não estava pronto em 5/8/2002. É dito, por exemplo, que a codificação do sistema Montador de dados foi concluída em 6/8/2002 e que a codificação do Sistema Montador de Dados foi concluída em 5/8/2002. Estas datas indicam, claramente, que o sistema não poderia estar concluído, isto é integrado, testado e homologado internamente (aceito pelo TSE) em 5/8, data do início da avaliação pelos partidos políticos.

Com base no exame da documentação disponibilizada não se pode fazer afirmativas sobre a confiabilidade do produto. Não estamos nos referindo a fraudes, mas sim a falhas no produto. O produto, conforme dados extraídos da documentação disponibilizada, não estava pronto e testado na data prevista e portanto nada se pode garantir sobre o mesmo.

2. Avaliação do Código

A avaliação do código só foi possível de iniciar no dia 6/08 devido à necessidade de se reinstalar algumas partes que teriam sido corrigidas pelas equipes de desenvolvimento do TSE. Como não foi possível utilizar as aplicações, e nem mesmo tentar compilá-las já que os ambientes instalados nos equipamentos estavam incompletos, faltando algumas bibliotecas internas, decidimos por iniciar a inspeção das aplicações desenvolvidas em PASCAL (Delphi) concentrando os trabalhos principalmente nos módulos responsáveis pelo tratamento dos dados. Posteriormente, foram extraídas medidas das aplicações C/C++ que serviram como critério de escolha para as partes dos sistemas que deveriam ser revisadas. Após todo este trabalho, novas modificações foram realizadas em parte do código pelas equipes do TSE e instaladas nos equipamentos ao final do dia 6/08, o que evidenciou que parte dos produtos ainda se encontrava em desenvolvimento e não se apresentava pronta para uso. No dia 7/8 extraímos novamente as medidas das aplicações. Os dados coletados em 6 e 7/8 ficaram armazenados na pasta Métricas do computador 15 do TSE, disponibilizado na sala de avaliação.

Os seguintes sistemas, representados por algumas de suas aplicações, foram avaliados durante este curto espaço de tempo:

1. Sistema de Gerenciamento das Eleições 2002
2. Sistema de Ajuste de Data e Hora da UE
3. Sistema Gerador de Mídias
4. Sistema Justificativa de votos

5. Sistema de Preparação da Urna Eletrônica
6. Sistema RecBU
7. Sistema da BIOS das Urnas Eletrônicas
8. Sistema vLOG
9. Sistema Assina
10. Sistema Montador de Dados

Após revisão de grande parte deste código (cujos resultados podem ser encontrados nos laudos individuais do anexo IV), observamos em linhas gerais os seguintes problemas:

- Os softwares não têm padrão de interface claro. Não há documentos descrevendo-o.
- As soluções algorítmicas são repetitivas e em alguns casos inadequadas, demonstrando a utilização de procedimentos de reutilização sem critério por parte dos programadores. Isto demonstra a não aplicação de revisão do código pelos programadores e imaturidade para o desenvolvimento de aplicações críticas, podendo comprometer o desempenho da aplicação e dificultar a manutenção futura do produto.
- Não existe preocupação com possíveis problemas de execução e tratamento de erros, a grande maioria do código (principalmente aplicações em PASCAL) foi construído para situações ideais de funcionamento, não considerando a possibilidade de falhas de algum dispositivo ou situação inusitada de funcionamento. Desta forma, a aplicação fica dependente da estabilidade de todos os componentes e usuários, e portanto, suscetível a erros de execução.
- A organização interna das aplicações demonstra que não ocorreu preocupação com o projeto do software (o que já tinha sido observado no exame da documentação). Em alguns casos, as decisões são tomadas diretamente pelos programadores e diferenças de estilo e solução para problemas semelhantes são observadas diretamente pela leitura do código. Características de baixa coesão e alto acoplamento de módulos foram identificadas nas diferentes aplicações.
- Foi observada a presença de código inativo em várias aplicações críticas. Não existe explicação para a necessidade de existência deste código. Supõem-se que a falta de revisão e acompanhamento do desenvolvimento tenha provocado este problema.
- Para nenhuma das aplicações revisadas pudemos encontrar descrições sobre arquitetura, modularidade e decisões de projeto que levaram ao código fonte produzido.
- É possível observar, para as diferentes aplicações em C++, a não compreensão por parte da equipe de desenvolvimento de aspectos fundamentais de programação com Orientação a Objetos, o que representa um risco a mais na execução e manutenção destes sistemas.
- Em alguns aplicativos há diversas versões de treinamento com comportamentos diferentes. Seria bom diminuir ao máximo essas diferenças para facilitar o trabalho de testes.
- uso de diretivas de compilação devem ser muito bem documentadas pois, em muitos casos, é através delas que diferentes versões (simulação, oficial, etc.) dos softwares são gerados e/ou customizados.
- Impressão de BU: O papel não contém marca d'água ou outro dispositivo equivalente para garantir a autenticidade do material. A marca impressa no papel pode ser

facilmente retirada. Se isso ocorrer, não será possível determinar se o voto é válido ou não.

- Os comentários deveriam descrever claramente que elementos foram acrescentados ou suprimidos por causa de uma nova eleição;
- O uso de constantes deve ser incentivado para facilitar a manutenção do software e, também, o entendimento do código. Tanto no código quanto na especificação é comum os valores serem utilizados diretamente.

É preciso realizar uma (meta) análise detalhada a partir de tratamento estatístico das medidas extraídas das aplicações para sugerir os pontos críticos (maior possibilidade de falha) a serem avaliados, revisados ou modificados. Para isto necessitamos ter em mãos os dados obtidos a partir da utilização da ferramenta de extração de métricas que, pelas regras do TSE, não nos foi permitido trazer naquele momento. Esperamos poder ter acesso a estas informações no sentido de tentar complementar esta avaliação e completar nossas recomendações.

IV. Recomendações

As recomendações a seguir estão organizadas em recomendações a curto prazo (para as eleições 2002) e as recomendações a médio prazo (para as eleições 2004 e seguintes).

1. Recomendações a curto prazo (para as eleições 2002)

A curto prazo, considerando que estamos a menos de 60 dias das eleições não se pode pretender a realizações de modificações que não sejam emergenciais no código, com o objetivo de conferir ao mesmo o grau de confiabilidade necessário para não comprometer as eleições 2002. Alterações não emergenciais devem ser realizadas posteriormente às eleições 2002 e constam do item a seguir (Recomendações a médio prazo).

Consideramos que a curto prazo para o software ser homologado e aceito pelo TSE e pelos partidos políticos deve-se:

1. Definir um processo sistemático para testes e depuração que inclua os seguintes tipos de teste:
 - Testes de unidades
 - Neste caso avaliando-se as diferentes unidades existentes para os softwares desenvolvidos em C, C++ e PASCAL, além das unidades transacionais responsáveis pelo interfaceamento com o sistema de banco de dados com comandos SQL
 - Teste de integração
 - Visando avaliar não só a integração interna das aplicações mas, principalmente, a integração existente entre os diferentes componentes do sistema, permitindo assim uma melhor compreensão da arquitetura definida para o sistema e a viabilidade de seu funcionamento.

- Teste do sistema
 - Com a utilização de massa de dados coerente e, se possível, com realização de testes de *stress* visando a observação de utilização do sistema em carga máxima, principalmente para avaliar seu desempenho.
 - Teste de homologação interna (TSE)
 - Teste de homologação externa (pelos partidos políticos)
2. Para cada tipo de teste devem ser definidos:
- Técnica de teste utilizada
 - Casos de teste
 - Descrição da situação que se está tentando verificar
 - Massa de Dados utilizada para o teste
 - Resultados esperados
 - Responsável pela utilização do teste
 - Responsável pela aprovação

Após a realização dos testes deve ser registrado para cada caso de teste o resultado obtido.

3. Ao serem realizadas modificações no código devem ser realizados testes de regressão, isto é, deve-se testar novamente o produto com todos os casos de teste com os quais a versão anterior foi testada além dos novos casos de teste produzidos para testar as modificações. Só assim se pode garantir que ao se corrigir um erro as características anteriores não foram modificadas.
4. Recomendamos, finalmente, que a aceitação pelos partidos seja realizada com uma demonstração do software com os procedimentos normais de uma homologação:
- Realização de testes do sistema completo pelo TSE na presença dos representantes dos partidos
 - Disponibilização dos Casos de Teste utilizados neste teste para exame pelos representantes dos partidos políticos
 - Realização de testes para novos casos solicitados pelos partidos caso estes o solicitem por escrito

Consideramos este procedimento o mais eficaz para dar à sociedade a confiança no software das Eleições 2002. Como podem e devem ser utilizados os mesmos casos de teste previamente utilizados pelo TSE para teste do sistema e homologação interna, este procedimento não introduz nenhum esforço extra à equipe.

2. Recomendações a médio prazo (para as eleições 2004 e seguintes)

Nossa avaliação observou como principal problema no desenvolvimento do software para as Eleições 2002, o fato da equipe não ter utilizado um processo de desenvolvimento e garantia de qualidade adequado, disciplinado e capaz de garantir a qualidade do produto.

Nossa recomendação a médio prazo é, portanto que seja definido um processo adequado ao TSE, compatível com a norma internacional ISO 12207, que a equipe seja treinada para seguir o processo e que a gerência do projeto audite rigorosamente a obediência ao processo.

Como o software para as Eleições 2004 e seguintes será, provavelmente, uma evolução deste das Eleições 2002, será necessária uma reengenharia do mesmo, visando corrigir os problemas relatados neste relatório e que por razão de tempo não serão possíveis de serem corrigidos de imediato.