

PROCESSO DE VERIFICAÇÃO DE COMPONENTES DE SOFTWARE PARA SISTEMAS DE DRONES EM REDE

PROPÓSITO: O propósito do processo de verificação é detectar e reportar erros nos requisitos de software, na descrição arquitetural, no Código-Fonte do software, ou no código executável, que podem ser introduzidos através dos processos de desenvolvimento, codificação ou integração do software.

NOTA 1: Verificação não é simplesmente testar, porque é necessário para este caso demonstrar a ausência de erros, o que em geral não pode ser demonstrado apenas com os testes. Os objetivos do processo de verificação são discutidos tipicamente como a combinação de revisões, análises e testes.

NOTA 2: Os objetivos e resultados do processo de verificação variam de acordo com o nível de garantia de segurança de software.

NOTA 3: Para fins de categorização das condições de falha, o que se deseja para um sistema de drones em rede, utilizando o protocolo de comunicação RADNET, da UFRJ, um protocolo oportunista, é que cada um dos drones conectados à rede cumpra um trajeto pré-determinado e comunique-se entre si, informando a própria posição e vetor velocidade e seja capaz de processar e comandar para si próprio, em tempo real, a variação do vetor velocidade de modo a manter autonomamente a separação longitudinal e lateral em relação aos demais drones, ou obstáculos próximos e manter uma altitude mínima em relação ao solo.

NOTA 4: As condições de falha do sistema são categorizadas para este processo como Catastrófica, Perigosa, Maior e Sem efeito. As definições de cada condição de falha constam na **Tabela 6**.

NOTA 5: Partição é a técnica usada para isolar componentes do software com funcionalidades independentes para conter e/ou isolar falhas e reduzir os esforços no processo de verificação do software. Se a proteção por partição é usada, o nível de garantia de software para cada componente com partição pode ser determinado usando a categoria de condição de falha mais severa associada àquele componente.

NOTA 6: Independência (semelhante à avaliação por pares): separação de responsabilidades que assegura o cumprimento do objetivo da avaliação; (1) para as atividades do processo de verificação, independência é atingida quando a atividade de verificação é realizada por pessoas diferentes das que desenvolveram os itens, ferramentas podem ser utilizadas para atingir equivalência à atividade manual de verificação; (2) para o processo de garantia da qualidade de software, independência inclui a autoridade para assegurar ações corretivas.

Tabela 6 – Categorização das condições de falha para um sistema de drones em rede

Catastrófica	Condição que leve os nós (drones) à colisão com outros nós da rede de drones, com outros obstáculos, ou com o terreno;
Perigosa	Condição de falha que leve à redução de separação abaixo do limite mínimo calculado entre os nós da rede de drones, ou entre qualquer nó e obstáculos, ou solo ao longo do trajeto, ou que leve a um aumento da taxa de processamento, ou uso de recursos computacionais de um drone acima do limite máximo calculado, ou que leve ao aumento da taxa de ocupação da banda disponível para comunicação de dados na rede de drones acima do limite máximo calculado, ou que leve a um aumento acima do limite máximo calculado na probabilidade de atraso na transmissão e no processamento de mensagens operacionais em tempo real entre os drones em rede.
Maior	Condição de falha que leve à redução de separação abaixo do limite mínimo calculado entre os nós da rede de drones, ou entre qualquer nó e obstáculos, ou solo ao longo do trajeto, mas que não leve a um aumento da taxa de processamento, ou uso de recursos computacionais de um drone acima do limite máximo calculado, que não leve ao aumento da taxa de ocupação da banda disponível para comunicação de dados na rede de drones acima do limite máximo calculado e que não implique no aumento acima do limite máximo calculado da probabilidade de atraso na transmissão e no processamento de mensagens operacionais em tempo real entre os drones em rede.
Sem efeito	Condição de falha que não leve à redução de separação abaixo do limite mínimo calculado entre os nós da rede de drones, ou entre qualquer nó e obstáculos, ou solo ao longo do trajeto, que não leve a um aumento da taxa de processamento, ou uso de recursos computacionais de um drone acima do limite máximo calculado, que não leve ao aumento da taxa de ocupação da banda disponível para comunicação de dados na rede de drones acima do limite máximo calculado e que não implique no aumento acima do limite máximo calculado da probabilidade de atraso na transmissão e no processamento de mensagens operacionais em tempo real entre os drones em rede.

NOTA 7: Os níveis de garantia de segurança de software são definidos como:

- **Nível de Garantia 1 (AL1):** softwares nos quais o comportamento anômalo, conforme descrição feita no processo de avaliação da segurança do sistema, causaria, ou contribuiria para falhas de funções do sistema, resultando em uma condição de falha catastrófica para o sistema de drones em rede;
- **Nível de Garantia 2 (AL2):** condição de falha perigosa;
- **Nível de Garantia 3 (AL3):** condição de falha maior;
- **Nível de Garantia 4 (AL4):** softwares nos quais falhas não produziram efeitos sobre a capacidade operacional do sistema de drones em rede.

RESULTADOS ESPERADOS

Os resultados esperados para o processo de verificação, uma vez realizado com sucesso são:

1. Que os requisitos de *software* de alto nível tenham sido verificados.
2. Que os requisitos de *software* de baixo nível e a arquitetura do *software* tenham sido verificados.
3. Que o Código-Fonte tenha sido verificado quanto à sua compatibilidade com os requisitos de baixo nível, arquitetura do *software* e especificação do *hardware*.
4. Que o código executável integrado ao *hardware* tenha sido testado.
5. Que as restrições quanto aos requisitos, arquitetura e projeto tenham sido identificadas.
6. Que o processo de verificação tenha sido verificado quanto à correção dos procedimentos, resultados dos testes e quanto à explicação dos desvios.
7. Que as informações necessárias para eventuais correções estejam disponíveis.
8. Que estejam disponíveis as evidências de que o *software* verificado é compatível com os requisitos, arquitetura e projeto.

Atividade 1: Verificação das Saídas do Processo de Requisitos de Software

Descrição: Verificar se os requisitos de alto nível são: compatíveis com os requisitos de sistema, se são acurados e consistentes, se são compatíveis com o hardware empregado para suportar a arquitetura de sistema proposta, se os requisitos de alto nível são verificáveis, se estão em conformidade com os padrões adotados pelo projeto e se são rastreáveis aos requisitos de sistema.

Artefatos de entrada: padrões de requisitos, requisitos de sistema, requisitos de alto nível de software, arquitetura do sistema, especificação do hardware proposto para suportar o software/sistema, árvore de controle de configuração e mudanças – matriz ou dados de rastreabilidade de requisitos.

Artefatos de saída: resultados da verificação de software em conformidade com a **NOTA 9** e **Tabela 1**, vide notas na coluna de descrição dos objetivos.

NOTA 10: O objetivo das revisões e análises dos requisitos de alto nível é detectar e reportar os erros nos requisitos introduzidos durante o processo de sua elaboração. As revisões e análises podem confirmar que os requisitos de alto nível satisfaçam aos objetivos listados e descritos a seguir:

a) conformidade com os requisitos de sistema: o objetivo é assegurar que as funções do sistema, realizadas pelo software, estejam definidas, que os requisitos funcionais, de desempenho e de segurança sejam satisfeitos pelo nível de garantia de software;

b) acurácia e consistência: o objetivo é assegurar que cada requisito de alto nível tenha acurácia, não seja ambíguo e seja suficientemente detalhado e que os requisitos em geral não sejam conflitantes;

c) compatibilidade com o hardware: o objetivo é assegurar que não existam conflitos entre os requisitos de alto nível e as características de hardware/software do dispositivo computacional, de modo particular, assegurar compatibilidade entre os tempos de resposta e as entradas e saídas de hardware requeridas pelo sistema e as características do hardware;

d) verificabilidade: o objetivo é assegurar que os requisitos de alto nível possam ser verificados;

e) conformidade com os padrões: o objetivo é assegurar que os padrões de requisitos de software sejam seguidos durante o processo de requisitos de software e que os desvios do padrão sejam justificados; e

f) rastreabilidade: o objetivo é assegurar que os requisitos sistema sejam desenvolvidos até que se obtenham os requisitos de software de alto nível, isto é, os requisitos de software de alto nível sejam rastreáveis a partir dos requisitos de sistema;

NOTA 9: Os resultados da verificação do software são produzidos pelas atividades do Processo de Verificação do Software. Os resultados da verificação do software devem:

a) para cada revisão, análise e teste, indicar cada procedimento que passou ou falhou, durante as atividades e o resultado final de sucesso/falha;

b) identificar o item de configuração ou versão do software revisada, analisada ou testada; e

c) incluir os resultados dos testes, revisões e/ou análises, incluindo as análises de cobertura e análises de rastreabilidade.

Tabela 1 – Verificação das Saídas do Processo de Requisitos de Software

Objetivos		Aplicabilidade				Resultados/Evidências
	Descrição	AL1	AL2	AL3	AL4	Descrição
1	Os requisitos de alto nível são compatíveis com os requisitos de sistema. NOTA 10.a	i	i	s	s	Resultados da verificação de software -NOTA 9
2	Os requisitos de alto nível são consistentes e acurados NOTA 10.b	i	i	s	s	Resultados da verificação de software -NOTA 9
3	Os requisitos de alto nível são compatíveis com o dispositivo computacional NOTA 10.c	s	s	n	n	Resultados da verificação de software -NOTA 9
4	Os requisitos de alto nível são verificáveis NOTA 10.d	s	s	n	n	Resultados da verificação de software -NOTA 9
5	Os requisitos de alto nível estão em conformidade aos padrões NOTA 10.e	s	s	n	n	Resultados da verificação de software -NOTA 9
6	Os requisitos de alto nível são rastreáveis aos requisitos de sistema NOTA 10.f	s	s	n	n	Resultados da verificação de software -NOTA 9

Legenda:

- s - objetivo deve ser satisfeito
- i - objetivo deve ser satisfeito com independência
- n - não aplicável

Atividade 2: Verificação das Saídas do Processo de Projeto de Software

Descrição: Verificar se os requisitos de baixo nível estão em conformidade com os requisitos de alto nível, se são acurados e consistentes, se são compatíveis com o dispositivo computacional e se são verificáveis, se estão em conformidade com os padrões, se são rastreáveis para os requisitos de alto nível, verificar se os algoritmos são acurados, se a arquitetura de software é compatível com os requisitos de alto nível, se a arquitetura de *software* é consistente e se é compatível com o dispositivo computacional, se a arquitetura de *software* é verificável, se está em conformidade com os padrões de projeto e se a integridade da partição é confirmada.

Artefatos de entrada: padrões de projeto, requisitos de alto nível, requisitos de baixo nível, arquitetura do sistema, arquitetura do software, especificação do hardware integrado para suportar o software/sistema, árvore de controle de configuração e mudanças – matriz ou dados de rastreabilidade de requisitos.

Artefatos de saída: Resultados da verificação de *software* em conformidade com a **NOTA 9** e **Tabela 2**, vide notas na coluna de descrição dos objetivos.

NOTA 11: O objetivo das revisões e análises de requisitos de baixo nível é detectar e reportar os erros nos requisitos que podem ser introduzidos durante o Processo de Projeto de Software. As revisões e análises podem confirmar que os requisitos de baixo nível satisfaçam aos objetivos listados e descritos a seguir:

a) conformidade aos requisitos de alto nível: o objetivo é assegurar que os requisitos de baixo nível satisfaçam aos requisitos de *software* de alto nível e que a base de projeto, a arquitetura e os requisitos de baixo nível derivados estejam corretamente definidos;

b) acurácia e consistência: o objetivo é assegurar que cada requisito de baixo nível tenha acurácia e não seja ambíguo e que os requisitos em geral não sejam conflitantes;

c) compatibilidade com o hardware: o objetivo é assegurar que não existam conflitos entre os requisitos de baixo nível e as características de *hardware/software* do dispositivo computacional, de modo particular, os aspectos relativos ao uso de recursos (como sobrecarga de barramento), aos tempos de resposta do sistema e às entradas e saídas de *hardware*;

d) verificabilidade: o objetivo é assegurar que os requisitos de baixo nível possam ser verificados;

e) conformidade com os padrões: o objetivo é assegurar que os padrões de projeto de *software* sejam seguidos durante o processo de projeto de *software* e que os desvios do padrão sejam justificados;

f) rastreabilidade: o objetivo é assegurar que os requisitos de alto nível sejam desenvolvidos até que se obtenham os requisitos de *software* de baixo nível, isto é, os requisitos de software de baixo nível sejam rastreáveis a partir dos requisitos de software de alto nível; e

g) aspectos relacionados com os algoritmos: o objetivo é assegurar a acurácia e o comportamento dos algoritmos propostos em relação aos requisitos aprovados para o projeto. Observar, particularmente, os casos de descontinuidade.

NOTA 12: O objetivo das revisões e análises da arquitetura de software é detectar e reportar os erros nos requisitos que podem ser introduzidos durante o processo de elaboração da arquitetura de software. As revisões e análises podem confirmar que os requisitos de baixo nível satisfaçam aos objetivos listados e descritos a seguir:

a) compatibilidade com os requisitos de alto nível: o objetivo é assegurar que a arquitetura de software não seja conflitante com os requisitos de software de alto nível. Especialmente as funções que garantam a integridade;

b) consistência: o objetivo é assegurar que a relação existente entre os componentes da arquitetura de software esteja em conformidade com o projeto do software. Essa relação existe via diagramas fluxo de dados e fluxo de controle;

c) compatibilidade com o *hardware*: o objetivo é assegurar que não existam conflitos, especialmente na inicialização, operação assíncrona, sincronização e interrupções, entre a arquitetura de *software* e as características de *hardware/software* do dispositivo computacional;

d) verificabilidade: o objetivo é assegurar que a arquitetura de software possa ser verificada para que, por exemplo, não haja algoritmos recursivos ilimitados;

e) conformidade com os padrões: o objetivo é assegurar que os padrões de projeto de software sejam seguidos durante o processo de projeto de software e que os desvios do padrão sejam justificados, especialmente quanto às restrições de complexidade e construções de projeto que não atendam aos objetivos de segurança; e

f) integridade da partição: o objetivo é assegurar que sejam evitadas brechas na partição.

NOTA 9: Os resultados da verificação do software são produzidos pelas atividades do Processo de Verificação do Software. Os resultados da verificação do software devem:

a) para cada revisão, análise e teste, indicar cada procedimento que passou ou falhou, durante as atividades e o resultado final de sucesso/falha;

b) identificar o item de configuração ou versão do software revisada, analisada ou testada; e

c) incluir os resultados dos testes, revisões e/ou análises, incluindo as análises de cobertura e análises de rastreabilidade.

Tabela 2 – Verificação das Saídas do Processo de Projeto de Software

Objetivos		Aplicabilidade				Resultados/Evidências
	Descrição	AL1	AL2	AL3	AL4	Descrição
1	Os requisitos de baixo nível estão em conformidade com os requisitos de alto nível - NOTA 11.a	i	i	s	s	Resultados da verificação de software - NOTA 9
2	Os requisitos de baixo nível são acurados e consistentes - NOTA 11.b	i	i	s	n	Resultados da verificação de software - NOTA 9
3	Os requisitos de baixo nível são compatíveis com o dispositivo computacional - NOTA 11.c	s	s	n	n	Resultados da verificação de software - NOTA 9
4	Os requisitos de baixo nível são verificáveis - NOTA 11.d	s	s	s	n	Resultados da verificação de software - NOTA 9
5	Os requisitos de baixo nível estão em conformidade com os padrões - NOTA 11.e	s	s	s	n	Resultados da verificação de software - NOTA 9
6	Os requisitos de baixo nível são rastreáveis aos requisitos de alto nível - NOTA 11.f	s	s	s	s	Resultados da verificação de software - NOTA 9
7	Os algoritmos são acurados - NOTA 11.g	i	i	s	n	Resultados da verificação de software - NOTA 9
8	A arquitetura é compatível com os requisitos de alto nível - NOTA 12.a	i	s	s	s	Resultados da verificação de software - NOTA 9
9	A arquitetura de <i>software</i> é consistente - NOTA 12.b	i	s	n	n	Resultados da verificação de software - NOTA 9
10	A arquitetura de <i>software</i> é compatível com o dispositivo computacional - NOTA 12.c	i	i	n	n	Resultados da verificação de software - NOTA 9
11	A arquitetura de software é verificável - NOTA 12.d	s	s	s	n	Resultados da verificação de software - NOTA 9
12	A arquitetura de software está em conformidade aos padrões - NOTA 12.e	s	s	s	n	Resultados da verificação de software - NOTA 9
13	A integridade da partição é confirmada - NOTA 12.f	s	s	s	s	Resultados da verificação de software - NOTA 9

Legenda:

- s - objetivo deve ser satisfeito
- i - objetivo deve ser satisfeito com independência
- n - não aplicável

Atividade 3: Verificação das Saídas do Processo de Codificação e Integração de Software

Descrição: Verificar se o código-fonte é compatível com os requisitos de baixo nível, se o código-fonte é compatível com a arquitetura de software, se é verificável, se está em conformidade com os padrões, se o código-fonte é rastreável em relação aos requisitos de baixo nível, se é acurado e consistente e verificar se os resultados do processo de integração são completos e corretos.

Artefatos de entrada: padrões de codificação de software, requisitos de baixo nível, arquitetura do *software*, código-fonte.

Artefatos de saída: Resultados da verificação de *software* em conformidade com a **NOTA 9** e **Tabela 3**, vide notas na coluna de descrição dos objetivos.

NOTA 13: O objetivo das revisões e análises do Código-Fonte é detectar e reportar os erros nos requisitos que podem ser introduzidos durante o processo de codificação. As revisões e análises podem confirmar que os produtos desse processo são acurados, completos e podem ser verificados. A atenção é voltada para a compatibilidade dos requisitos com a arquitetura e conformidade com os padrões de codificação adotados. As revisões e análises se restringem ao código e podem incluir:

a) compatibilidade com os requisitos de baixo nível: o objetivo é assegurar que o Código-Fonte é acurado e completo com respeito aos requisitos de baixo nível e que o Código-Fonte não implementa uma função não documentada;

b) compatibilidade com a arquitetura: o objetivo é assegurar que o Código-Fonte corresponde ao fluxo de dados e fluxo de controle definidos na arquitetura de software;

c) verificabilidade: o objetivo é assegurar que o Código-Fonte não contém instruções e estruturas que não possam ser verificadas e que o código não precise ser alterado para realização de testes;

d) conformidade com os padrões: o objetivo é assegurar que os padrões de codificação do software foram seguidos durante o desenvolvimento de código, especialmente as restrições de complexidade e as checagens de código que devem ser consistentes com os objetivos de segurança. Complexidade inclui o grau de acoplamento entre os componentes de software, os níveis de estruturas aninhadas de controle e a complexidade de expressões lógicas ou numéricas. Essa análise também assegura que os desvios do padrão foram justificados;

e) rastreabilidade: o objetivo é assegurar que os requisitos de baixo nível foram desenvolvidos no Código-Fonte; e

f) acurácia e consistência: o objetivo é determinar a correção e consistência do Código-Fonte, incluindo o uso de pilha, resolução e overflow de aritmética de ponto-flutuante, contenção de recursos, tempo de execução do pior caso, tratamento de exceções, uso de variáveis ou constantes não inicializadas, corrupção de dados devido a conflito de tarefas ou interrupções.

NOTA 14: O objetivo das revisões e análises dos produtos do processo de integração é assegurar que o processo de integração é completo e correto. Isso pode ser feito através do exame detalhado com a ligação e carregamento de dados e do mapa de memória. Verificando:

a) endereçamento incorreto de hardware;

b) sobreposição de memória; e

c) falta de componentes de software.

NOTA 9: Os Resultados da Verificação do Software são produzidos pelas atividades do Processo de Verificação do Software. Os Resultados da Verificação do Software devem:

a) para cada revisão, análise e teste, indicar cada procedimento que passou ou falhou, durante as atividades e o resultado final de sucesso/falha;

b) identificar o item de configuração ou versão do software revisada, analisada ou testada; e

c) incluir os resultados dos testes, revisões e/ou análises, incluindo as análises de cobertura e análises de rastreabilidade.

Tabela 3 – Verificação das Saídas do Processo de Codificação e Integração de Software

Objetivos		Aplicabilidade				Resultados/Evidências
	Descrição	AL1	AL2	AL3	AL4	Descrição
1	O código-fonte é compatível com os requisitos de baixo nível NOTA 13.a	i	i	s	s	Resultados da Verificação de software - NOTA 9
2	O código-fonte é compatível com a arquitetura de software NOTA 13.b	i	s	s	s	Resultados da Verificação de software - NOTA 9
3	O código-fonte é verificável NOTA 13.c	s	s	n	n	Resultados da Verificação de software - NOTA 9
4	O código-fonte está em conformidade com os padrões NOTA 13.d	s	s	n	n	Resultados da Verificação de software - NOTA 9
5	O código-fonte tem rastreabilidade aos requisitos de baixo nível NOTA 13.e	s	s	s	s	Resultados da Verificação de software - NOTA 9
6	O código-fonte é acurado e consistente NOTA 13.f	i	s	s	n	Resultados da Verificação de software - NOTA 9
7	Os resultados do processo de integração são completos e corretos NOTA 14	s	s	s	n	Resultados da Verificação de software - NOTA 9

Legenda:

s - objetivo deve ser satisfeito

i - objetivo deve ser satisfeito com independência

n - não aplicável

Atividade 4: Testes dos Resultados do Processo de Integração

Descrição: Testar se o código executável é compatível com os requisitos de alto nível, se o código executável é robusto com respeito aos requisitos de alto nível, se o código executável é compatível com os requisitos de baixo nível, se o código executável é robusto com respeito aos requisitos de baixo nível, se o código executável é compatível com o hardware.

Artefatos de entrada: requisitos de alto nível, requisitos de baixo nível, código executável, especificação do hardware integrado ao *software*/sistema.

Artefatos de saída: procedimentos de casos de verificação de software, em conformidade com a **NOTA 9**, procedimentos e casos de verificação de *software*, em conformidade com a **NOTA 8** e **Tabela 4**, vide notas na coluna de descrição dos objetivos.

NOTA 15: O objetivo dos casos de testes com variação normal é demonstrar que o *software* responde as entradas em condições normais. Os testes com variação normal devem incluir as seguintes considerações:

- a) variáveis reais e inteiras devem ser exigidas usando uma classe de equivalência dos valores limites;
- b) para funções variantes no tempo, como filtros, integradores e temporizadores, múltiplas iterações do código devem ser realizadas para checar as características da função no seu contexto;
- c) para transições de estado, casos de testes devem ser desenvolvidos para exercitar as possíveis transições durante a operação normal; e
- d) para requisitos de *software* expressados por equações lógicas, a variação normal pode verificar o uso das variáveis e os operadores booleanos. Para o nível de segurança AL1 os operadores booleanos podem ser verificados por análise ou revisão, e para complementação dessa atividade, casos de testes podem ser escolhidos para prover cobertura de decisão/condição modificada (MC/DC).

NOTA 16: O objetivo dos casos de testes de robustez é demonstrar que o software responde as entradas em condições não normais. Os testes de robustez devem incluir os seguintes aspectos:

- a) variáveis reais e inteiras devem ser exigidas usando uma classe de equivalência dos valores inválidos;
- b) a inicialização do sistema deve ser exercitada fora das condições normais;
- c) os possíveis modos de falha dos dados de entrada devem ser determinados, especialmente dados digitais complexos organizados e enviados em sequência (*strings* de dados digitais) vindos de sistemas externos;
- d) para execuções sequenciais e repetitivas de comandos de software como parte de um algoritmo (loops) em que o contador do loop (variável que controla o número de vezes que a sequência de comandos será repetida) é um valor computado, os casos de testes devem ser desenvolvidos para tentar processar valores fora da variação normal e demonstrar a robustez do código do loop;
- e) a checagem deve assegurar os mecanismos de proteção para tempos de resposta maiores que o esperado;
- f) para funções temporais, como os filtros, integradores e temporizadores, os casos de testes devem desenvolver mecanismos de proteção de overflow aritmético (quando o resultado de um cálculo computacional aritmético extrapola a capacidade de armazenamento do número resultante na variável de memória); e

g) para transições de estados, os casos de testes devem provocar transições que não são permitidas pelos requisitos de software.

NOTA 17: Os métodos para testes baseados em requisitos consistem nos métodos de testes de integração de hardware e software, de integração de software e de requisitos de baixo nível. Com exceção de testes de integração de hardware e software, os métodos prescrevem ambiente de teste ou estratégia para teste. As instruções incluem:

a) testes de integração de hardware e software: esse método deve concentrar-se nas fontes de erros associadas com a operação dentro do ambiente em que o software opera, e com alto grau de funcionalidade. O objetivo dos testes de integração de hardware e software é assegurar que o software no computador satisfaz aos requisitos de alto nível. Típicos erros revelados por esse método incluem: incorreto tratamento de interrupções; falha para satisfazer aos requisitos de tempo de execução; resposta incorreta aos transientes ou falhas de hardware, como exemplo, sequência incorreta de partida, transientes nas entradas ou transientes de energia; problemas no barramento de dados ou com contenção de recursos, por exemplo, mapeamento de memória; problemas na detecção de falhas do BIT; erros nas interfaces de hardware/software; comportamento inadequado de loops; controle incorreto do gerenciamento de memória ou outro dispositivo com controle em software; estouro de pilha; operação incorreta dos mecanismos usados para carregamento de software em campo; e violações da partição;

b) testes de integração de software: esse método deve concentrar-se nos relacionamentos entre os requisitos de software e na implementação dos requisitos pela arquitetura de software. O objetivo dos testes de integração de software é assegurar que os componentes interagem corretamente entre si. Típicos erros revelados por esse método incluem: incorreta inicialização de variáveis e constantes; erros na passagem de parâmetros; corrupção de dados, especialmente os dados globais; inadequada resolução numérica; e incorreta sequência de eventos e operações;

c) testes de requisitos de baixo nível: esse método deve concentrar-se na demonstração de que cada componente é compatível com os requisitos de baixo nível. Típicos erros que devem ser levantados: falha no algoritmo para satisfazer ao requisito; operações incorretas em loops; decisões lógicas incorretas; falhas para processar combinações válidas de condições de entradas; respostas incorretas com perda ou corrupção nos dados de entrada; incorreto tratamento de exceções, como falhas aritméticas ou violação dos limites de matrizes; incorreta computação de sequências; e precisão, acurácia ou desempenho inadequados do algoritmo.

NOTA 8: Casos e Procedimentos de Verificação do Software detalham como as atividades do processo de verificação do software são implementadas e geram suas evidências. Estas evidências devem incluir as descrições:

a) dos procedimentos de análise e revisão: Detalhes, suplementares à descrição constante no Plano de Verificação do Software, que descrevem o escopo e a profundidade dos métodos de análise e revisão a serem utilizados;

b) casos de teste: o propósito de cada caso de teste, conjunto de entradas, condições, resultados esperados para atingir o critério de cobertura requerido e o critério de sucesso/falha; e

c) procedimentos de teste: as instruções passo a passo de como cada caso de teste deve ser preparado e executado, como os resultados do teste são avaliados e o ambiente de teste a ser utilizado.

NOTA 9: Os resultados da verificação do software são produzidos pelas atividades do Processo de Verificação do Software. Os resultados da verificação do software devem:

- a)** para cada revisão, análise e teste, indicar cada procedimento que passou ou falhou, durante as atividades e o resultado final de sucesso/falha;
- b)** identificar o item de configuração ou versão do software revisada, analisada ou testada; e
- c)** incluir os resultados dos testes, revisões e/ou análises, incluindo as análises de cobertura e análises de rastreabilidade.

Tabela 4 – Testes dos Resultados do Processo de Integração

Objetivos		Aplicabilidade				Resultados/Evidências
	Descrição	AL1	AL2	AL3	AL4	Descrição
1	O código executável é compatível com os requisitos de alto nível NOTAS: 15,17					Procedimentos e Casos de Verificação de Software – NOTA 8
		i	i	s	s	Resultados da Verificação de software - NOTA 9
2	O código executável é robusto com respeito aos requisitos de alto nível NOTAS: 16,17					Procedimentos e Casos de Verificação de Software – NOTA 8
		i	s	s	n	Resultados da Verificação de software - NOTA 9
3	O código executável é compatível com os requisitos de baixo nível NOTAS: 15,17					Procedimentos e Casos de Verificação de Software – NOTA 8
		i	i	s	s	Resultados da Verificação de software - NOTA 9
4	O código executável é robusto com respeito aos requisitos de baixo nível NOTAS: 16,17					Procedimentos e Casos de Verificação de Software – NOTA 8
		i	s	s	n	Resultados da Verificação de software - NOTA 9
5	O código executável é compatível com o dispositivo computacional NOTA 17.a					Procedimentos e Casos de Verificação de Software – NOTA 8
		s	s	s	n	Resultados da Verificação de software - NOTA 9

Legenda:

- s - objetivo deve ser satisfeito
- i - objetivo deve ser satisfeito com independência
- n - não aplicável

Atividade 5: Verificação dos Resultados do Processo de Verificação

Descrição: Verificar se os procedimentos de testes estão corretos, se os resultados dos testes são corretos e se as discrepâncias com o planejamento são explicadas, se a cobertura dos testes dos requisitos de alto nível é atingida, se a cobertura dos testes dos requisitos de baixo nível é atingida, se a cobertura das estruturas de software (cobertura de decisão/condição modificada) é atingida, se a cobertura das estruturas de software (cobertura de decisões) é atingida, se a cobertura da estrutura de *software* (cobertura de linhas executáveis) é atingida, se a cobertura da estrutura de *software* (acoplamento de dados e de controle) é atingida.

Artefatos de entrada: plano de verificação de software, procedimentos de análise e revisão, casos de teste, procedimentos de teste, conforme **NOTA 8**.

Artefatos de saída: resultados da verificação de *software*, em conformidade com a **NOTA 9** e **Tabela 5**, vide notas na coluna de descrição dos objetivos.

NOTA 18: O objetivo dessas revisões e análises de procedimentos, casos e resultados de testes é assegurar que o teste do código é feito de modo acurado e completo. Isso pode incluir:

- a) casos de testes:** a verificação de casos de testes é apresentada na **NOTA 8.b**;
- b) procedimentos de testes:** o objetivo é verificar que os casos de testes são precisamente desenvolvidos em procedimentos de testes e resultados esperados; e
- c) resultados de testes:** o objetivo é assegurar que os resultados dos testes são corretos e que as discrepâncias entre resultados obtidos e esperados sejam explicadas.

NOTA 19: A análise de cobertura é um processo de dois passos, envolvendo análise da cobertura de requisitos e a análise da cobertura estrutural. O primeiro passo analisa os casos de testes com relação aos requisitos de software para confirmar que os casos selecionados satisfazem critérios específicos. O segundo passo confirma que os testes exercitam a estrutura de código. A análise da cobertura estrutural pode não satisfazer a um critério específico. Instruções adicionais são dadas para os casos de código desativado.

1) Análise da Cobertura de Requisitos: o objetivo dessa análise é determinar quão bem os testes de requisitos verificaram a implementação dos requisitos de software. Essa análise pode revelar a necessidade de casos de testes adicionais. A análise mostra:

- a)** a existência de casos de testes para cada requisito de software; e
- b)** a satisfação dos testes com critérios de variação normal e robustez.

2) Análise da Cobertura Estrutural: o objetivo dessa análise é determinar quais estruturas de código não foram exercitadas pelos procedimentos de testes. Os casos de testes podem não exercitar completamente a estrutura de código, assim verificações adicionais são produzidas para se alcançar à cobertura estrutural. A análise deve:

- a)** confirmar o grau de cobertura apropriado para o nível de segurança de software;
- b)** pode ser feita no Código-Fonte, a menos que o nível de segurança de software seja AL1 e que o compilador gere Código-Objeto que não é diretamente rastreado às linhas de código. Então, verificações adicionais devem ser feitas para avaliar o Código-Objeto de modo a estabelecer que este foi gerado correto; e

c) deve confirmar o acoplamento entre dados e controle existentes no código.

NOTA 9: Os resultados da verificação do software são produzidos pelas atividades do Processo de Verificação do Software. Os resultados da verificação do software devem:

- a) para cada revisão, análise e teste, indicar cada procedimento que passou ou falhou, durante as atividades e o resultado final de sucesso/falha;
- b) identificar o item de configuração ou versão do software revisada, analisada ou testada; e
- c) incluir os resultados dos testes, revisões e/ou análises, incluindo as análises de cobertura e análises de rastreabilidade.

Tabela 5 – Verificação dos Resultados do Processo de Verificação

Objetivos		Aplicabilidade				Resultados/Evidências
	Descrição	AL1	AL2	AL3	AL4	Descrição
1	Os procedimentos de testes são corretos – NOTA 18.b	i	s	s	s	Resultados da Verificação de software - NOTA 9
2	Os resultados dos testes são corretos e as discrepâncias com o planejamento são explicadas – NOTA 18.c	i	s	s	s	Resultados da Verificação de software - NOTA 9
3	A cobertura dos testes dos requisitos de alto nível é atingida - NOTA 19.1	i	s	s	s	Resultados da Verificação de software - NOTA 9
4	A cobertura dos testes dos requisitos de baixo nível é atingida - NOTA 19.1	i	s	s	n	Resultados da Verificação de software - NOTA 9
5	A cobertura das estruturas de <i>software</i> (cobertura de decisão/condição modificada) é atingida - NOTA 19.2.a e NOTA 19.2.b	i	s	n	n	Resultados da Verificação de software - NOTA 9
6	A cobertura das estruturas de software (cobertura de decisões) é atingida - NOTA 19.2.a e NOTA 19.2.b	i	i	n	n	Resultados da Verificação de software - NOTA 9
7	A cobertura das estruturas de software (cobertura de linhas executáveis) é atingida - NOTA 19.2.a e NOTA 19.2.b	i	i	n	n	Resultados da Verificação de software - NOTA 9
8	A cobertura das estruturas de software (acoplamento de dados e de controle) é atingida - NOTA 19.2.c	i	i	n	n	Resultados da Verificação de software - NOTA 9

Legenda:

- s - objetivo deve ser satisfeito
- i - objetivo deve ser satisfeito com independência
- n - não aplicável