MỤC LỤC

Lời nói đầu

TCVN 11391:2016 hoàn toàn tương đương với tiêu chuẩn EN 50128:2011

TCVN 11391: 2016 do Cục Đăng kiểm Việt Nam biên soạn, Bộ Giao thông vận tải đề nghị, Tổng cục Tiêu chuẩn – Đo lường – Chất lượng thẩm định, Bộ Khoa học và Công nghệ công bố.

.

Ứng dụng đường sắt – Hệ thống xử lý và thông tin tín hiệu – Phần mềm cho các hệ thống phòng vệ và điều khiển đường sắt

Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems

1 Phạm vi áp dụng

1.1 Tiêu chuẩn này quy định các yêu cầu về quá trình và kỹ thuật đối với việc phát triển phần mềm cho các hệ thống điện tử có thể lập trình sử dụng trong các ứng dụng điều khiển và phòng vệ đường sắt. Mục đích là để sử dụng trong tất cả các lĩnh vực có khả năng liên quan tới an toàn. Những hệ thống này có thể hoạt động sử dụng các bộ vi xử lý chuyên dụng, bộ điều khiển logic có thể lập trình, các hệ thống đa xử lý phân tán, các hệ thống xử lý trung tâm quy mô lớn hoặc các cấu trúc khác.

1.2 Tiêu chuẩn này có thể áp dụng riêng cho phần mềm và sự tương tác giữa phần mềm và hệ thống chứa phần mềm đó.

1.3 Tiêu chuẩn này không liên quan tới phần mềm được xác định là không ảnh hưởng tới an toàn, ví dụ: phần mềm có các hư hỏng không thể ảnh hưởng tới bất kì các chức năng an toàn nào đã được xác định.

1.4 Tiêu chuẩn này áp dụng cho tất cả các phần mềm liên quan tới an toàn được sử dụng trong các hệ thống điều khiển và phòng vệ đường sắt, bao gồm:

- Lập trình ứng dụng;
- Các hệ điều hành;
- Các chương trình hỗ trợ;
- Chương trình cơ sở điều khiển thiết bị.

Việc lập trình ứng dụng bao gồm lập trình cấp cao, lập trình cấp thấp và lập trình cho mục đích đặc biệt (ví dụ: bộ lập trình điều khiển ngôn ngữ bậc thang).

1.5 Tiêu chuẩn này cũng đề cập tới việc sử dụng các chương trình và phần mềm hiện có. Những phần mềm này có thể được sử dụng nếu đáp ứng đầy đủ các yêu cầu cụ thể trong mục 7.3.4.7 và 6.5.4.16 đối với phần mềm và chương trình hiện có trong mục 6.7.

1.6 Phần mềm được xây dựng theo tiêu chuẩn này sẽ được xem là phù hợp và không phụ thuộc vào các yêu cầu của phần mềm hiện có.

1.7 Tiêu chuẩn này cho rằng thiết kế ứng dụng hiện đại thường sử dụng phần mềm chung phù hợp làm cơ sở cho các ứng dụng khác nhau. Phần mềm chung này sau đó được cấu hình theo dữ liệu, thuật toán, hoặc cả hai để tạo ra phần mềm có thể thực hiện ứng dụng. Nhìn chung từ Điều 1 đến Điều 6 và Điều 9 của tiêu chuẩn này áp dụng cho phần mềm chung cũng như cho các thuật toán và dữ liệu ứng dụng. Điều 7 chỉ áp dụng cụ thể cho phần mềm chung trong khi Điều 8 đưa ra các yêu cầu cụ thể cho các thuật toán và dữ liệu ứng dụng.

1.8 Tiêu chuẩn này không đề cập tới các vấn đề thương mại. Vấn đề này nên được đưa ra xem xét trong thỏa thuận hợp đồng. Nên xem xét cẩn thận tất cả các nội dung của tiêu chuẩn này trong mọi tình huống có tính chất thương mại.

1.9 Tiêu chuẩn này không áp dụng cho các hệ thống hiện đang sử dụng mà chủ yếu áp dụng cho các hệ thống xây dựng mới, và chỉ áp dụng toàn bộ cho các hệ thống hiện có nếu có những thay đổi quan trọng. Đối với những thay đổi nhỏ, chỉ áp dụng mục 9.2. Đơn vị đánh giá phải phân tích các bằng chứng được cung cấp trong hồ sơ phần mềm để xác nhận tính đầy đủ trong việc xác định bản chất và phạm vi của các thay đổi phần mềm. Tuy nhiên, khuyến nghị cao việc ứng dụng tiêu chuẩn này trong việc nâng cấp và bảo trì phần mềm hiện có.

2 Tài liệu viện dẫn

Các tài liệu tham khảo sau là cần thiết đối với việc ứng dụng tài liệu này. Đối với các tham khảo có năm, chỉ áp dụng các phiên bản được trích dẫn. Đối với các tham khảo không có năm, phiên bản mới nhất của văn bản được khuyến nghị áp dụng (bao gồm cả phần bổ sung).

TCVN 10935-1 (EN 50126-1:1999), *Ứng dụng đường sắt – Quy định và chứng minh độ tin cậy, tính sẵn sàng, khả năng bảo dưỡng và độ an toàn – Phần 1: Các yêu cầu cơ bản và quy trình chung.*

EN 50129:2003, Railway applications – Safety related electronic systems for signalling *(Ứng dụng đường sắt – Các hệ thống điện tử liên quan đến an toàn cho hệ thống tín hiệu).*

TCVN ISO 9000, *Hệ thống quản lý chất lượng – Các nguyên tắc cơ bản và từ vựng.*

TCVN ISO 9001, *Hệ thống quản lý chất lượng – Các yêu cầu.*

ISO/IEC 90003:2004, Software engineering – Guidelines for the application of ISO 9001:2000 to computer software *(Kỹ thuật phần mềm – Hướng dẫn áp dụng ISO 9001:2000 cho phần mềm máy tính).*

ISO/IEC 9126 series, Software engineering – Product quality *(Kỹ thuật phần mềm – Chất lượng sản phẩm)*

3 Thuật ngữ, định nghĩa và các từ viết tắt

3.1 Thuật ngữ và định nghĩa

Đối với tiêu chuẩn này, áp dụng những thuật ngữ dưới đây.

3.1.1 Đánh giá (Assessment)

Quá trình phân tích để xác định liệu phần mềm (bao gồm các thành phần phần cứng và/hoặc phần mềm của hệ thống con, hệ thống, tài liệu, quá trình) có đáp ứng các yêu cầu được quy định và để đưa ra kết luận phần mềm phù hợp với mục đích dự định sử dụng. Việc đánh giá an toàn sẽ tập trung nhưng không giới hạn ở các đặc tính an toàn của hệ thống.

3.1.2 Đơn vị đánh giá (Assessor)

Người hoặc tổ chức thực hiện việc đánh giá.

3.1.3 Phần mềm thương mại phổ biến (commercial off-the-shelf (COTS))

Phần mềm được xác định theo nhu cầu của thị trường, sẵn có trên thị trường và sự phù hợp với mục đích đã được chứng minh bởi nhiều người sử dụng.

3.1.4 Thành phần (component)

Là một phần để tạo thành phần mềm có các giao diện và sự hoạt động được định nghĩa rõ ràng về mặt thiết kế và cấu trúc phần mềm và đáp ứng các tiêu chí sau:

- Được thiết kế tuân theo “Các thành phần” (xem Bảng A.20);
- Bao gồm một nhóm cụ thể các yêu cầu phần mềm;
- Được xác định rõ ràng và có phiên bản độc lập bên trong hệ thống quản lý cấu hình hoặc thuộc nhóm các thành phần có phiên bản độc lập (ví dụ: các hệ thống con).

3.1.5 Đơn vị quản lý cấu hình (configuration manager)

Tổ chức hoặc cá nhân chịu trách nhiệm thực hiện và tiến hành các quá trình để quản lý cấu hình các tài liệu, phần mềm và các công cụ liên quan, bao gồm cả quản lý sự thay đổi.

3.1.6 Khách hàng (Customer)

Tổ chức hoặc cá nhân mua hệ thống phòng vệ và điều khiển đường sắt có bao gồm phần mềm.

3.1.7 Đơn vị thiết kế (Designer)

Tổ chức hoặc cá nhân thực hiện phân tích và chuyển đổi các yêu cầu được quy định thành các giải pháp thiết kế có thể chấp nhận được và có mức toàn vẹn về an toàn yêu cầu.

3.1.8 Tổ chức hoặc cá nhân (Entity)

Cá nhân, nhóm người hoặc tổ chức đáp ứng vai trò như được nêu ra trong tiêu chuẩn này.

3.1.9 Lỗi, sự cố (Error, fault)

Khuyết tật, lỗi hoặc sự không chính xác mà có thể gây ra hư hỏng hoặc sai khác so với tính năng hoạt động dự định.

3.1.10 Hư hỏng (Failure)

Sai khác không thể chấp nhận giữa tính năng hoạt động được yêu cầu và tính năng hoạt động thực hiện.

3.1.11 Khả năng chấp nhận sự cố (Fault tolerance)

Khả năng được xây dựng sẵn trong một hệ thống để đáp ứng sự cung cấp dịch vụ chính xác liên tục như quy định, khi xuất hiện của một số lượng nhất định các sự cố phần cứng hoặc phần mềm.

3.1.12 Chương trình cơ sở điều khiển thiết bị (Firmware)

Phần mềm được lưu trữ trong bộ nhớ (chỉ được đọc) hoặc trong bộ nhớ bán vĩnh cửu (bộ nhớ flash), độc lập về mặt chức năng đối với phần mềm ứng dụng.

3.1.13 Phần mềm chung (Generic software)

Phần mềm có thể được sử dụng cho các cài đặt khác nhau, đơn giản chỉ bằng việc đưa ra các thuật toán và/hoặc dữ liệu ứng dụng cụ thể.

3.1.14 Đơn vị thực hiện (implementer)

Tổ chức hoặc cá nhân thực hiện chuyển đổi các thiết kế đã được quy định thành sản phẩm thực tế.

3.1.15 Sự tích hợp (integration)

Quá trình lắp ráp các hạng mục phần mềm và/hoặc phần cứng, theo chỉ dẫn cấu trúc và thiết kế, và kiểm thử đối tượng được tích hợp.

3.1.16 Đơn vị tích hợp (integrator)

Tổ chức hoặc cá nhân thực hiện việc tích hợp phần mềm.

3.1.17 Phần mềm hiện có (pre-existing software)

Phần mềm được xây dựng trước đây, bao gồm COTS và phần mềm mã nguồn mở.

3.1.18 Phần mềm mã nguồn mở (open source software)

Mã nguồn có sẵn sử dụng rộng rãi mà không có bản quyền hoặc hết thời hạn về bản quyền trước đó.

3.1.19 Bộ điều khiển logic có thể lập trình (Programmable logic controller)

Hệ thống vi điều khiển có một bộ nhớ lập trình theo người sử dụng để lưu trữ các chỉ lệnh thực hiện các chức năng cụ thể.

3.1.20 Quản lý dự án (project management)

Là việc điều hành dự án về mặt hành chính và/hoặc kỹ thuật, bao gồm các vấn đề về an toàn.

3.1.21 Đơn vị quản lý dự án (project manager)

Cá nhân hoặc tổ chức thực hiện việc quản lý dự án.

3.1.22 Độ tin cậy (Reliability)

Khả năng của một hạng mục thực hiện một chức năng theo yêu cầu dưới các điều kiện đã định trong một khoảng thời gian cho trước.

3.1.23 Độ chắc chắn (robustness)

Khả năng một hạng mục phát hiện và xử lý các tình huống bất thường.

3.1.24 Đơn vị quản lý các yêu cầu (requirements manager)

Tổ chức hoặc cá nhân thực hiện việc quản lý các yêu cầu.

3.1.25 Quản lý các yêu cầu (requirements management)

Quá trình đưa ra, ghi chép, phân tích, xếp hạng, thỏa thuận về các yêu cầu và sau đó kiểm soát sự thay đổi và liên kết với các đơn vị liên quan. Đây là quá trình liên tục xuyên suốt trong một dự án.

3.1.26 Rủi ro (Risk)

Sự kết hợp của tỉ lệ xuất hiện các tai nạn và các sự cố gây hại (do một nguy hiểm tạo ra) và mức độ nghiêm trọng của thiệt hại đó.

3.1.27 An toàn (Safety)

Trạng thái không tồn tại các mức độ rủi ro gây thiệt hại cho con người không thể chấp nhận được.

3.1.28 Cơ quan quản lý về an toàn (Safety Authority)

Cơ quan chịu trách nhiệm đối với việc chứng nhận phần mềm hoặc dịch vụ liên quan tới an toàn phù hợp với các yêu cầu về an toàn theo quy định của pháp luật liên quan.

3.1.29 Chức năng an toàn (safety function)

Chức năng mà thực hiện một phần hoặc toàn bộ một yêu cầu an toàn.

3.1.30 Phần mềm liên quan tới an toàn (Safety-related software)

Phần mềm thực hiện các chức năng an toàn.

3.1.31 Phần mềm (Software)

Là sản phẩm trí tuệ bao gồm các chương trình, quy trình, quy tắc, dữ liệu và mọi tài liệu ghi chép kết hợp liên quan tới sự vận hành của hệ thống.

3.1.32 Cơ sở phần mềm (software baseline)

Tập hợp hoàn chỉnh và thống nhất mã nguồn, các tệp chạy chương trình, các tệp cấu hình, các đoạn mã cài đặt và tệp lưu trữ cần thiết để phát hành phần mềm. Các thông tin về trình biên dich, hệ điều hành, phần mềm hiện có và các chương trình phụ thuộc được lưu trữ như là một phần của cơ sở phần mềm. Cơ sở phần mềm sẽ giúp cho tổ chức có thể tái lập ra các phiên bản xác định và là đầu vào cho các phát hành phần mềm sau này khi tăng cường hoặc khi nâng cấp ở trong giai đoạn bảo trì.

3.1.33 Triển khai phần mềm (software deployment)

Việc chuyển đổi, cài đặt và kích hoạt cơ sở phần mềm chuyển giao mà đã được phát hành và đánh giá.

3.1.34 Vòng đời phần mềm (Software life cycle)

Các hoạt động xuất hiện trong một khoảng thời gian bắt đầu khi phần mềm được chuyển giao và kết thúc khi phần mềm không còn có thể sử dụng. Thông thường, vòng đời phần mềm bao gồm giai đoạn các yêu cầu, giai đoạn thiết kế, giai đoạn kiểm thử, giai đoạn tích hợp, giai đoạn triển khai và giai đoạn bảo trì.

3.1.35 Khả năng bảo trì phần mềm (Software maintainablity)

Khả năng hiệu chỉnh phần mềm để sửa chữa các sự cố, cải tiến tính năng hoạt động hoặc các thuộc tính khác, hoặc thay đổi cho phù hợp với môi trường khác.

3.1.36 Bảo trì phần mềm (Software maintenance)

Hoạt động hoặc một loạt các hoạt động được tiến hành trên phần mềm sau khi được người sử dụng cuối cùng chấp nhận. Mục đích của nó là cải tiến, tăng cường và/hoặc sửa chữa chức năng của nó.

3.1.37 Mức toàn vẹn về an toàn phần mềm (Software safety integrity level)

Một chỉ số phân loại xác định rõ các kỹ thuật và các biện pháp phải được áp dụng cho phần mềm.

3.1.38 Nhà cung cấp (supplier)

Tổ chức hoặc cá nhân thiết kế và xây dựng hệ thống điều khiển và phòng vệ đường sắt, bao gồm phần mềm hoặc các phần tương ứng.

3.1.39 Mức toàn vẹn về an toàn hệ thống (System safety integrity level)

Một chỉ số thể hiện mức độ tin cậy yêu cầu mà một hệ thống được tích hợp gồm cả phần cứng và phần mềm sẽ đáp ứng các yêu cầu về an toàn quy định.

3.1.40 Đơn vị kiểm thử (tester)

Tổ chức hoặc cá nhân thực hiện việc kiểm thử.

3.1.41 Kiểm thử (testing)

Quá trình chạy phần mềm dưới các điều kiện được kiểm soát để xác nhận sự hoạt động và hiệu năng của nó khi so sánh với chỉ dẫn các yêu cầu tương ứng.

3.1.42 Loại chương trình T1 (tool class T1)

Chương trình không tạo các đầu ra có thể tham gia trực tiếp hoặc gián tiếp vào mã chạy chương trình (bao gồm dữ liệu) của phần mềm.

Chú thích: Ví dụ về T1 bao gồm: một bộ hiệu chỉnh ký tự hoặc một yêu cầu hoặc một chương trình hỗ trợ thiết kế không có khả năng tự tạo mã tự động, các chương trình kiểm soát cấu hình.

3.1.43 Loại chương trình T2 (tool class T2)

Chương trình hỗ trợ cho việc kiểm thử hoặc thẩm tra thiết kế hoặc mã chạy chương trình, khi các lỗi trong chương trình có thể không phát hiện ra sai sót nhưng không thể trực tiếp tạo ra lỗi trong phần mềm chạy.

Chú thích: Ví dụ về T2 bao gồm: bộ hỗ trợ kiểm thử, chương trình đo đạc phạm vi kiểm thử; chương trình phân tích tĩnh.

3.1.44 Loại chương trình T3 (tool class T3)

Chương trình tạo các đầu ra có thể tham gia trực tiếp hoặc gián tiếp vào mã chạy chương trình (bao gồm dữ liệu) của hệ thống liên quan tới phần mềm.

Chú thích: Ví dụ về T3 bao gồm: một trình biên dịch mã nguồn, một trình biên dịch dữ liệu/thuật toán, một chương trình thay đổi các điểm set-point trong quá trình vận hành hệ thống, một trình biên dịch tối ưu khi có mối liên kết không rõ ràng giữa chương trình mã nguồn và mã đối tượng được tạo ra; một trình biên dịch kết hợp gói chạy chương trình vào mã chạy chương trình.

3.1.45 Khả năng theo dõi theo vết (Traceability)

Mức độ của mối quan hệ có thể được thiết lập giữa hai hoặc nhiều sản phẩm của quá trình phát triển, đặc biệt những sản phẩm có mối quan hệ cũ/mới hoặc chính/phụ với sản phẩm khác.

3.1.46 Thẩm định (Validation)

Quá trình phân tích để đưa ra kết luận dựa trên bằng chứng để xác định một hạng mục (ví dụ: quá trình, tài liệu, phần mềm hoặc ứng dụng) phù hợp với yêu cầu của người sử dụng, đặc biệt là về an toàn và chất lượng và nhấn mạnh vào sự phù hợp của hoạt động với mục đích của hạng mục đó trong môi trường dự định.

3.1.47 Đơn vị thẩm định (Validator)

Người hoặc tổ chức chịu trách nhiệm đối với việc thẩm định.

3.1.48 Thẩm tra (Verification)

Quá trình kiểm tra để đưa ra kết luận dựa trên bằng chứng về các hạng mục đầu ra (quá trình, tài liệu, phần mềm hoặc ứng dụng) của một giai đoạn phát triển cụ thể đáp ứng các yêu cầu của giai đoạn đó về mặt hoàn chỉnh, chính xác và thống nhất.

3.1.49 Đơn vị thẩm tra (Verifier)

Cá nhân hoặc tổ chức chịu trách nhiệm cho một hoặc nhiều hoạt động thẩm tra.

3.2 Từ viết tắt

Trong tiêu chuẩn này, sử dụng các từ viết tắt dưới đây.

ASR	Đơn vị đánh giá
COTS	Thương mại phổ biến
DES	Đơn vị thiết kế
HR	Khuyến nghị cao
IMP	Đơn vị thực hiện
INT	Đơn vị tích hợp
JSD	Phương pháp phát triển hệ thống Jackson
M	Bắt buộc
MASCOT	Biện pháp tiếp cận theo module đối với cấu trúc, vận hành và kiểm thử phần mềm
NR	Không khuyến nghị
PM	Đơn vị quản lý dự án
R	Khuyến nghị
RAMS	Độ tin cậy, tính sẵn sàng, khả năng bảo dưỡng và độ an toàn
RQM	Đơn vị quản lý các yêu cầu
SDL	Ngôn ngữ mô tả và chỉ dẫn kỹ thuật
SFC	Lược đồ hàm tuần tự
SIL	Mức toàn vẹn về an toàn
SOM	Lập mô hình định hướng dịch vụ
SSADM	Biện pháp thiết kế & phân tích các hệ thống cấu trúc
TST	Đơn vị kiểm thử
V&V	Thẩm tra và thẩm định
VAL	Đơn vị thẩm định
VER	Đơn vị thẩm tra

4 Mục đích, sự phù hợp và mức toàn vẹn về an toàn phần mềm

4.1 Việc phân bổ các chức năng của hệ thống liên quan tới an toàn cho phần mềm, cũng như các giao diện phần mềm phải được xác định trong tài liệu hệ thống. Hệ thống có chứa phần mềm phải được xác định đầy đủ các vấn đề sau đây:

- Các chức năng và các giao diện;
- Các điều kiện ứng dụng;

- Cấu hình hoặc cấu trúc của hệ thống;
- Các nguy hiểm được kiểm soát;
- Các yêu cầu về tính toàn vẹn an toàn;
- Việc phân bổ các yêu cầu và phân bổ SIL cho phần mềm và phần cứng;
- Các ràng buộc về thời gian.

Chú thích: Việc phân bổ các yêu cầu về tính toàn vẹn về an toàn có thể dẫn đến các SIL khác nhau cho các bộ phận phần mềm và phần cứng được phân tách tốt của một hệ thống con. Việc phân bổ này sẽ phụ thuộc vào sự tham gia của các bộ phận phần mềm và phần cứng của hệ thống con trong các tính năng liên quan tới an toàn và cơ chế giảm thiểu hư hỏng, bao gồm sự phân chia chức năng có các SIL khác nhau.

4.2 Tính toàn vẹn về an toàn phần mềm phải được quy định là 1 trong 5 mức, từ SIL 0 (mức thấp nhất) cho tới SIL 4 (mức cao nhất).

4.3 Mức toàn vẹn về an toàn phần mềm yêu cầu phải được quyết định và đánh giá ở mức độ hệ thống, trên cơ sở mức toàn vẹn về an toàn hệ thống và mức độ rủi ro liên quan khi sử dụng phần mềm trong hệ thống.

4.4 Tối thiểu, các yêu cầu về SIL 0 trong tiêu chuẩn này phải được đáp ứng cho bộ phận phần mềm của các chức năng có tác động tới an toàn dưới SIL 1. Do sẽ tồn tại sự không chắc chắn trong việc đánh giá rủi ro, và cả trong việc xác định các nguy hiểm. Để xử lý các vấn đề không chắc chắn, phải cẩn trọng khi xác định mức toàn vẹn về an toàn thấp (được thể hiện bằng SIL 0), ngoài các đối tượng không có mức toàn vẹn về an toàn.

4.5 Đề phù hợp tiêu chuẩn này, phải thế hiện cho thấy được các yêu cầu đã được đáp ứng theo mức toàn vẹn về an toàn phần mềm đã xác định trước và thỏa mãn điều khoản mục tiêu đề ra.

4.6 Khi một yêu cầu đủ điều kiện cho cụm từ “theo mức toàn vẹn về an toàn phần mềm được yêu cầu” có nghĩa là có thể sử dụng các kỹ thuật và các biện pháp để thỏa mãn yêu cầu này.

4.7 Khi áp dụng mục 4.6, phải sử dụng các bảng trong Phụ lục A để hỗ trợ cho việc lựa chọn các kỹ thuật và các biện pháp phù hợp theo mức toàn vẹn về an toàn phần mềm. Việc lựa chọn phải được ghi lại trong Kế hoạch đảm bảo chất lượng phần mềm hoặc trong tài liệu khác được tham chiếu trong Kế hoạch đảm bảo chất lượng phần mềm. Hướng dẫn về những kỹ thuật này được đưa ra trong Phụ lục tham khảo D.

4.8 Nếu không sử dụng một kỹ thuật hoặc một biện pháp được xếp hạng là *khuyến nghị cao (HR*) trong các bảng thì lý do cho việc sử dụng kỹ thuật thay thế phải được nêu chi tiết và được ghi lại trong Kế hoạch đảm bảo chất lượng phần mềm hoặc trong các tài liệu khác được tham chiếu trong Kế hoạch đảm bảo chất lượng phần mềm. Điều này sẽ không cần thiết nếu sử dụng kết hợp các kỹ thuật được phê duyệt đưa ra trong bảng tương ứng. Các kỹ thuật được lựa chọn phải được chứng minh là áp dụng đúng và chính xác.

4.9 Nếu một kỹ thuật hoặc biện pháp được đề nghị sử dụng mà không có trong các bảng thì tính hiệu quả và khả năng phù hợp của nó khi đáp ứng các yêu cầu cụ thể và mục đích tổng thể của nội dung mục đó phải được kết luận và ghi lại trong Kế hoạch đảm bảo chất lượng phần mềm hoặc trong tài liệu khác tham chiếu khác từ Kế hoạch đảm bảo chất lượng phần mềm.

4.10 Phải thẩm tra sự phù hợp với các yêu cầu của một điều khoản cụ thể và các kỹ thuật và các biện pháp tương ứng của nó được nêu chi tiết trong các bảng, bằng cách kiểm tra các tài liệu được yêu cầu theo tiêu chuẩn này. Nếu phù hợp phải tính tới các bằng chứng khách quan khác, các đánh giá và chứng kiến kiểm thử.

5 Quản lý và tổ chức phần mềm

5.1 Tổ chức, vai trò và trách nhiệm

5.1.1 Mục đích

Để đảm bảo tất cả các cá nhân chịu trách nhiệm về phần mềm được tổ chức, giao quyền và đủ khả năng hoàn thành các trách nhiệm đó.

5.1.2 Các yêu cầu

5.1.2.1 Tối thiểu, nhà cung cấp phải thực hiện các nội dung của TCVN ISO 9001 đối với việc tổ chức và quản lý các cá nhân và trách nhiệm.

5.1.2.2 Các trách nhiệm phải phù hợp với các yêu cầu được xác định trong Phụ lục B.

5.1.2.3 Cá nhân được chỉ định các vai trò liên quan đến việc phát triển hoặc bảo trì phần mềm phải có tên và được ghi lại.

5.1.2.4 Đơn vị đánh giá phải được nhà cung cấp, khách hàng hoặc Cơ quản quản lý về an toàn chỉ định.

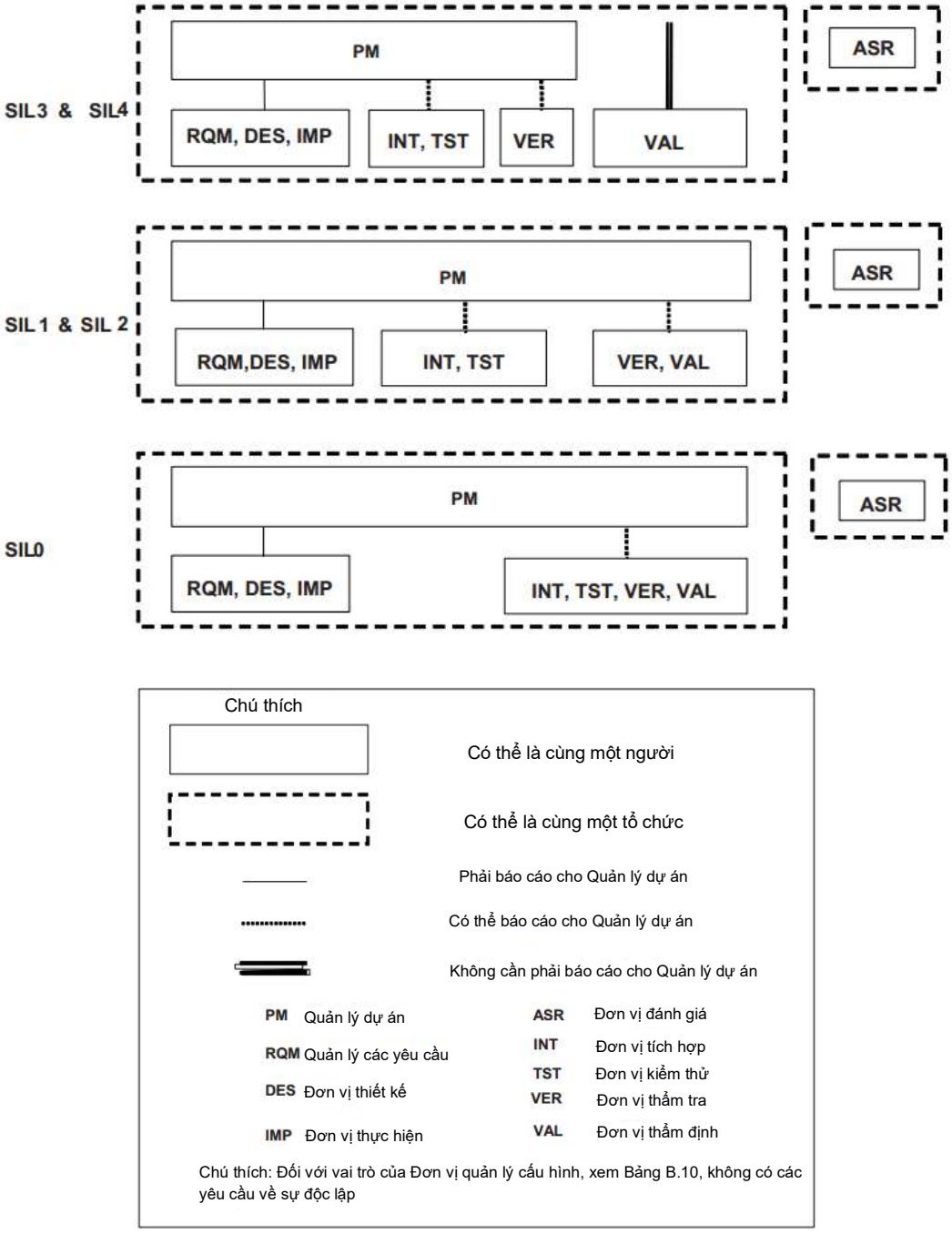
5.1.2.5 Đơn vị đánh giá phải độc lập với nhà cung cấp, hoặc theo chỉ đạo của Cơ quan quản lý về an toàn thuộc tổ chức của nhà cung cấp hoặc tổ chức của khách hàng.

5.1.2.6 Đơn vị đánh giá phải độc lập với dự án.

5.1.2.7 Đơn vị đánh giá phải được trao thẩm quyền thực hiện việc đánh giá phần mềm.

5.1.2.8 Đơn vị thẩm định phải đưa ra sự đồng ý/không đồng ý khi phát hành phần mềm.

5.1.2.9 Trong toàn bộ vòng đời phần mềm, việc chỉ định vai trò cho các cá nhân phải phù hợp với mục 5.1.2.10 đến 5.1.2.14 theo mức SIL phần mềm yêu cầu.



Hình 2 – Minh họa cấu trúc tổ chức ưu tiên

5.1.2.10 Sơ đồ tổ chức khuyến nghị cho mức SIL 3 và SIL 4 là:

a) Đơn vị quản lý các yêu cầu, Đơn vị thiết kế và Đơn vị thực hiện đối với một thành phần phần mềm có thể là cùng một đơn vị.

b) Đơn vị quản lý các yêu cầu, Đơn vị thiết kế và Đơn vị thực hiện thành phần phần mềm phải báo cáo cho Đơn vị quản lý dự án.

c) Đơn vị tích hợp và Đơn vị kiểm thử thành phần mềm có thể là cùng một đơn vị.

d) Đơn vị tích hợp và Đơn vị kiểm thử thành phần phần mềm có thể báo cáo cho Đơn vị quản lý dự án hoặc cho Đơn vị thẩm định.

e) Đơn vị thẩm tra có thể báo cáo cho Đơn vị quản lý dự án hoặc cho Đơn vị thẩm định.

f) Đơn vị thẩm định không phải báo cáo cho Đơn vị quản lý dự án, ví dụ: Đơn vị quản lý dự án phải không tác động đến các quyết định của Đơn vị thẩm định, nhưng Đơn vị thẩm định phải thông báo cho Đơn vị quản lý dự án về các quyết định của mình.

g) Người làm quản lý các yêu cầu, thiết kế hoặc thực hiện thành phần phần mềm phải không phải là Đơn vị kiểm thử, cũng không phải là Đơn vị tích hợp cho cùng một thành phần phần mềm.

h) Người làm tích hợp hoặc kiểm thử thành phần phần mềm phải không là Đơn vị quản lý các yêu cầu, thiết kế hoặc thực hiện đối với cùng một thành phần phần mềm.

i) Người làm Thẩm tra phải không phải là Đơn vị quản lý các yêu cầu, thiết kế, thực hiện, tích hợp, kiểm thử hoặc thẩm định.

j) Người làm Thẩm định phải không phải là đơn vị quản lý các yêu cầu, thiết kế, thực hiện, tích hợp, kiểm thử hoặc thẩm tra.

k) Người làm quản lý dự án có thể thực hiện bổ sung các vai trò của Đơn vị quản lý các yêu cầu, thiết kế, thực hiện, tích hợp, kiểm thử hoặc thẩm tra, miễn là các yêu cầu về sự độc lập giữa các vai trò bổ sung này phải được tôn trọng.

l) Đơn vị quản lý dự án, Đơn vị quản lý các yêu cầu, Đơn vị thiết kế, Đơn vị thực hiện, Đơn vị tích hợp, Đơn vị kiểm thử, Đơn vị thẩm tra và Đơn vị thẩm định có thể thuộc cùng một tổ chức.

m) Đơn vị đánh giá phải độc lập và độc lập về mặt tổ chức với vai trò của Đơn vị quản lý dự án, Đơn vị quản lý các yêu cầu, Đơn vị thiết kế, Đơn vị thực hiện, Đơn vị tích hợp, Đơn vị kiểm thử, Đơn vị thẩm tra và Đơn vị thẩm định.

Tuy nhiên, có thể áp dụng các lựa chọn sau:

n) Người làm thẩm định cũng có thể thực hiện vai trò của Đơn vị thẩm tra, những vẫn duy trì sự độc lập với Đơn vị quản lý dự án. Trong trường hợp này, các tài liệu đầu ra của Đơn vị thẩm tra phải được một cá nhân khác có đủ khả năng xem xét ở cùng mức độ độc lập như Đơn vị thẩm định. Lựa chọn về mặt tổ chức phải tùy thuộc vào sự chấp thuận của Đơn vị đánh giá.

o) Người làm thẩm tra cũng có thể thực hiện vai trò của Đơn vị tích hợp và kiểm thử, trong trường hợp này, vai trò của Đơn vị thẩm định phải kiểm tra sự phù hợp của bằng chứng được ghi lại từ quá trình tích hợp và kiểm thử với các mục tiêu thẩm tra được quy định, từ đó duy trì 2 mức độ kiểm tra trong tổ chức dự án.

5.1.2.11 Sơ đồ tổ chức khuyến nghị cho mức SIL 1 và SIL 2 là:

a) Đơn vị quản lý các yêu cầu, Đơn vị thiết kế và Đơn vị thực hiện cho một thành phần phần mềm có thể là cùng một người và báo cáo cho Đơn vị quản lý dự án.

b) Đơn vị tích hợp và Đơn vị kiểm thử thành phần mềm mềm có thể là cùng một người.

c) Đơn vị tích hợp và Đơn vị kiểm thử thành phần phần mềm có thể báo cáo cho Đơn vị quản lý dự án hoặc cho Đơn vị thẩm định.

d) Đơn vị thẩm tra và Đơn vị thẩm định có thể là cùng một người.

e) Đơn vị thẩm tra và đơn vị thẩm định có thể báo cáo cho Đơn vị quản lý dự án.

f) Người làm Quản lý các yêu cầu, thiết kế hoặc thực hiện thành phần phần mềm phải không phải là Đơn vị kiểm thử, cũng không phải là Đơn vị tích hợp cho cùng một thành phần phần mềm.

g) Người làm tích hợp hoặc kiểm thử thành phần phần mềm phải không là Đơn vị quản lý các yêu cầu, thiết kế hoặc thực hiện đối với cùng một thành phần phần mềm.

h) Người làm Thẩm tra hoặc thẩm định phải không phải là Đơn vị quản lý các yêu cầu, thiết kế, thực hiện, tích hợp, kiểm thử.

i) Người làm quản lý dự án có thể thực hiện bổ sung các vai trò của Đơn vị quản lý các yêu cầu, thiết kế, thực hiện, tích hợp, kiểm thử, thẩm tra hoặc thẩm định, miễn là các yêu cầu về sự độc lập giữa các vai trò bổ sung này được tôn trọng.

j) Đơn vị quản lý dự án, Đơn vị quản lý các yêu cầu, Đơn vị thiết kế, Đơn vị thực hiện, Đơn vị tích hợp, Đơn vị kiểm thử, Đơn vị thẩm tra và Đơn vị thẩm định có thể thuộc cùng một tổ chức.

k) Đơn vị đánh giá phải độc lập và độc lập về mặt tổ chức với vai trò của Đơn vị quản lý dự án, Đơn vị quản lý các yêu cầu, Đơn vị thiết kế, Đơn vị thực hiện, Đơn vị tích hợp, Đơn vị kiểm thử, Đơn vị thẩm tra và Đơn vị thẩm định.

Tuy nhiên, có thể áp dụng các lựa chọn sau:

l) Người làm Thẩm tra cũng có thể thực hiện vai trò của Đơn vị tích hợp và kiểm thử, trong trường hợp này, vai trò của Đơn vị thẩm định phải bao gồm việc xem xét các tài liệu kết quả của Đơn vị thẩm tra, từ đó duy trì 2 mức độ kiểm tra trong tổ chức dự án.

m) Người làm Thẩm định cũng có thể thực hiện vai trò của Đơn vị thẩm tra, tích hợp và kiểm thử. Trong trường hợp này, các tài liệu kết quả của Đơn vị thẩm tra phải được một cá nhân khác có đủ khả năng xem xét ở cùng mức độ độc lập như Đơn vị thẩm định. Lựa chọn về mặt tổ chức phải tùy thuộc vào sự chấp thuận của Đơn vị đánh giá.

5.1.2.12 Kết cấu tổ chức khuyến nghị cho mức SIL 0:

a) Đơn vị quản lý các yêu cầu, Đơn vị thiết kế và Đơn vị thực hiện cho một thành phần phần mềm có thể là cùng một người và phải được quản lý bởi cùng Đơn vị quản lý dự án.

b) Đơn vị tích hợp và Đơn vị kiểm thử, Đơn vị thẩm tra và Đơn vị thẩm định cho thành phần mềm mềm có thể là cùng một người.

c) Đơn vị tích hợp và Đơn vị kiểm thử, đơn vị thẩm tra và đơn vị thẩm định có thể được quản lý bởi cùng đơn vị quản lý dự án.

d) Người làm Quản lý các yêu cầu, thiết kế hoặc thực hiện thành phần phần mềm phải không phải là Đơn vị kiểm thử, cũng không phải là đơn vị tích hợp cho cùng một thành phần phần mềm.

e) Người làm Thẩm tra hoặc thẩm định phải không phải là đơn vị quản lý các yêu cầu, thiết kế, thực hiện.

f) Người làm quản lý dự án có thể thực hiện bổ sung các vai trò của Đơn vị quản lý các yêu cầu, thiết kế, thực hiện, tích hợp, kiểm thử, thẩm tra hoặc thẩm định, miễn là các yêu cầu về sự độc lập giữa các vai trò bổ sung này được tôn trọng.

g) Đơn vị quản lý dự án, Đơn vị quản lý các yêu cầu, Đơn vị thiết kế, Đơn vị thực hiện, Đơn vị tích hợp, Đơn vị kiểm thử, Đơn vị thẩm tra và Đơn vị thẩm định có thể thuộc cùng một tổ chức.

h) Đơn vị đánh giá phải độc lập và độc lập về mặt tổ chức với vai trò của Đơn vị quản lý dự án, Đơn vị quản lý các yêu cầu, Đơn vị thiết kế, Đơn vị thực hiện, Đơn vị tích hợp, Đơn vị kiểm thử, Đơn vị thẩm tra và Đơn vị thẩm định.

Tuy nhiên, có thể áp dụng các lựa chọn thay thế sau:

i) Đơn vị quản lý các yêu cầu, Đơn vị thiết kế, Đơn vị thực hiện, Đơn vị tích hợp và Đơn vị kiểm thử có thể là cùng một người.

j) Đơn vị thẩm định và Đơn vị thẩm tra cũng có thể là cùng một người.

k) Người làm thẩm tra hoặc thẩm định phải không được là Đơn vị quản lý các yêu cầu, thiết kế hay thực hiện.

5.1.2.13 Vai trò của Đơn vị quản lý các yêu cầu, Đơn vị thiết kế và thực hiện đối với một thành phần có thể thực hiện vai trò của Đơn vị kiểm thử và Đơn vị tích hợp cho một thành phần khác.

5.1.2.14 Vai trò của Đơn vị thẩm tra và thẩm định phải được xác định theo mức độ dự án và phải duy trì không thay đổi xuyên suốt quá trình phát triển dự án.

5.2 Năng lực cá nhân

5.2.1 Mục tiêu

5.2.1.1 Để đảm bảo tất cả các cá nhân có trách nhiệm đối với phần mềm có đủ năng lực để thực hiện các trách nhiệm, bằng cách chứng minh khả năng thực hiện các nhiệm vụ liên quan một cách chính xác, hiệu quả và thống nhất với chất lượng cao và theo các điều kiện thay đổi khác nhau.

5.2.2 Các yêu cầu

5.2.2.1 Phụ lục B xác định các năng lực chính cần thiết cho từng vai trò trong quá trình phát triển phần mềm. Nếu cần phải có kinh nghiệm, năng lực hoặc khả năng bổ sung cho một vai trò nào đó trong vòng đời phần mềm thì những năng lực này phải được xác định trong Kế hoạch đảm bảo chất lượng phần mềm.

5.2.2.2 Bằng chứng ghi lại về năng lực cá nhân, bao gồm kiến thức kỹ thuật, các chứng chỉ, kinh nghiệm liên quan và các quá trình đào tạo phù hợp phải được tổ chức của nhà cung cấp lưu giữ để chứng minh cơ cấu tổ chức an toàn là phù hợp.

5.2.2.3 Tổ chức phải duy trì các quy trình để quản lý năng lực cá nhân để phù hợp với các vị trí tương ứng theo các tiêu chuẩn chất lượng hiện hành.

Khi đã chứng minh được cho đơn vị đánh giá thấy sự thỏa mãn hoặc bằng việc chứng nhận năng lực của tất cả các cá nhân đã được chứng minh ở các vai trò khác nhau, từng cá nhân cần phải thể hiện quá trình duy trì và phát triển năng lực liên tục. Việc này có thể được chứng minh qua sổ ghi chép thể hiện hoạt động được thực hiện chuẩn xác thường xuyên, và các quá trình đào tạo bổ sung được thực hiện phù hợp với TCVN ISO 9001 và ISO/IEC 90003:2004, mục 6.2.2.

5.3 Vấn đề vòng đời và lưu trữ

5.3.1 Mục tiêu

5.3.1.1 Để tổ chức được việc phát triển phần mềm thành các giai đoạn và các hoạt động xác định.

5.3.1.2 Để ghi lại tất cả thông tin thích hợp với phần mềm trong suốt vòng đời phần mềm.

5.3.2 Các yêu cầu

5.3.2.1 Phải lựa chọn một mô hình vòng đời đối với việc phát triển phần mềm. Phải nêu chi tiết trong Kế hoạch đảm bảo chất lượng phần mềm, phù hợp với mục 6.5 của tiêu chuẩn này.

Ví dụ về hai mô hình vòng đời được đưa ra trong Hình 3 và Hình 4.

5.3.2.2 Mô hình vòng đời phải tính tới khả năng lặp lại trong và giữa các giai đoạn.

5.3.2.3 Các quy trình bảo đảm chất lượng phải tiến hành song song với các hoạt động vòng đời và sử dụng cùng các thuật ngữ.

5.3.2.4 Kế hoạch đảm bảo chất lượng phần mềm, Kế hoạch thẩm tra phần mềm, Kế hoạch thẩm định phần mềm và Kế hoạch quản lý cấu hình phần mềm phải được đưa ra ngay khi bắt đầu dự án và được duy trì xuyên suốt vòng đời phát triển phần mềm.

5.3.2.5 Tất cả các hoạt động được thực hiện trong suốt một giai đoạn phải được xác định và được lên kế hoạch trước khi bắt đầu giai đoạn đó.

5.3.2.6 Tất cả các tài liệu phải được cấu trúc sao cho có thể mở rộng liên tục song song với quá trình thiết kế.

5.3.2.7 Việc theo dõi theo vết các tài liệu phải có trong từng tài liệu và có chung thống nhất các tham chiếu và mối quan hệ xác định được ghi lại với các tài liệu khác.

5.3.2.8 Mỗi thuật ngữ, nhóm từ hoặc từ viết tắt phải có cùng nghĩa trong tất cả các tài liệu. Nếu việc này không thể thực hiện được do các lý do liên quan tới lịch sử quá trình, các ý nghĩa khác nhau phải được liệt kê và các tham chiếu phải được đưa ra.

5.3.2.9 Ngoại trừ các tài liệu liên quan tới phần mềm hiện có trước đó (xem mục 7.3.4.7), từng tài liệu phải được lập thành văn bản theo các quy tắc sau:

- Phải có hoặc thực hiện tất cả các điều kiện và các yêu cầu có thể áp dụng của tài liệu trước đây mà có mối liên quan về phân cấp;

- Phải không trái ngược lại với tài liệu trước đó.

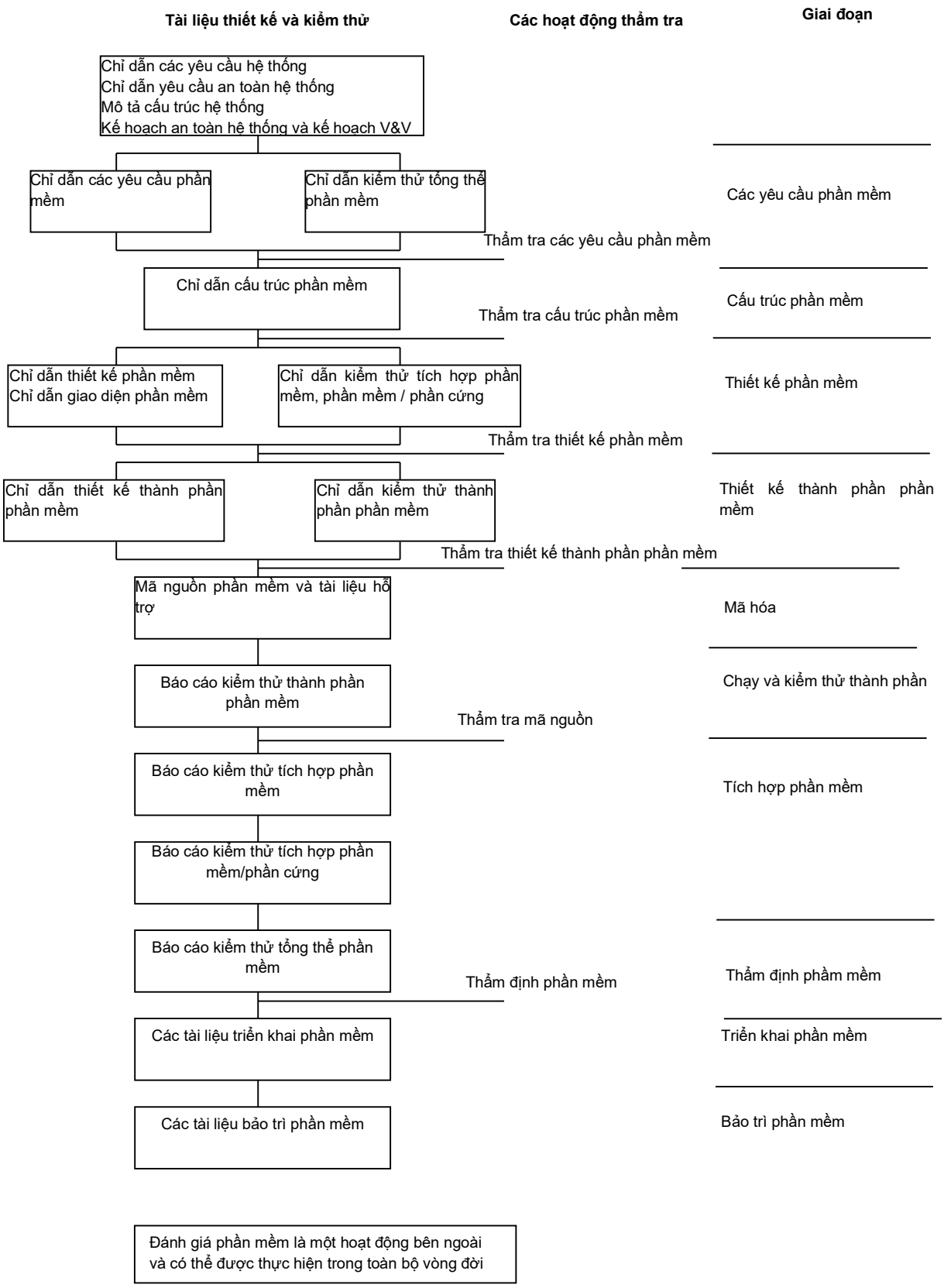
5.3.2.10 Từng khái niệm hoặc hạng mục phải được tham chiếu theo cùng tên hoặc mô tả trong mỗi tài liệu.

5.3.2.11 Nội dung của tất cả các tài liệu phải được ghi lại theo mẫu phù hợp để xử lý, giải quyết và lưu trữ.

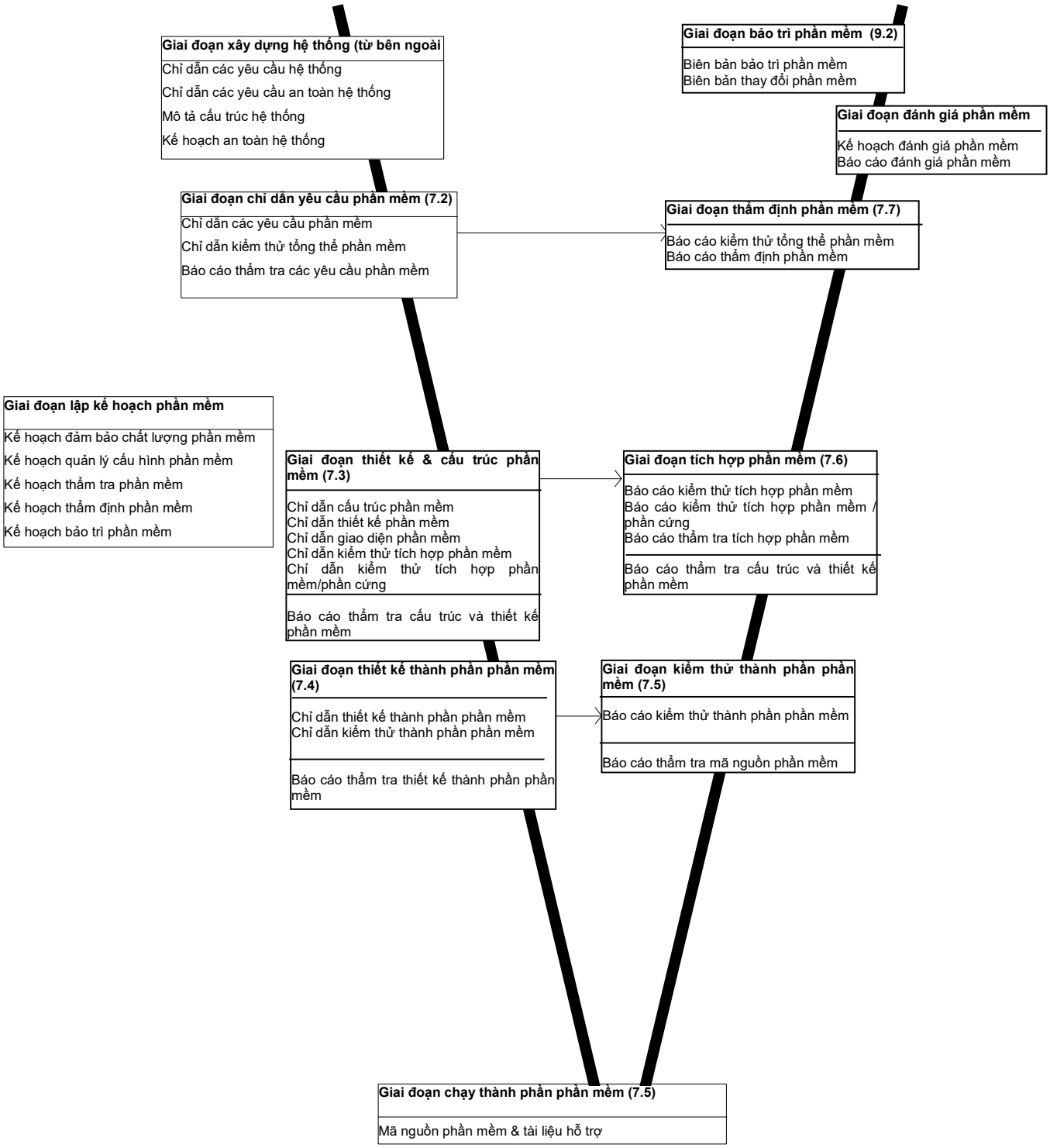
5.3.2.12 Khi các tài liệu do các cá nhân đảm nhiệm các vai trò khác nhau lập ra được kết hợp thành một tài liệu duy nhất, mối quan hệ giữa các phần được mọi cá nhân độc lập lập ra phải được theo dõi theo vết trong tài liệu.

5.3.2.13 Các tài liệu có thể được kết hợp hoặc phân chia theo mục 5.3.2.12. Một số bước xây dựng có thể kết hợp, phân chia hoặc khi được kết luận là loại bỏ theo hướng dẫn của Đơn vị quản lý dự án và sự đồng ý của Đơn vị thẩm định.

5.3.2.14 Khi chấp thuận mọi chu trình hoặc cấu trúc ghi chép thay thế, phải thể hiện được là đáp ứng tất cả các mục tiêu và yêu cầu của tiêu chuẩn này.



Hình 3 – Minh họa vòng đời phát triển 1



Hình 4 – Minh họa vòng đời phát triển 2

6 Đảm bảo phần mềm

6.1 Kiểm thử phần mềm

6.1.1 Mục tiêu

6.1.1.1 Mục tiêu kiểm thử phần mềm (được thực hiện bởi Đơn vị kiểm thử và/hoặc Đơn vị tích hợp) là để xác nhận tính năng hoạt động của phần mềm theo chỉ dẫn kỹ thuật kiểm thử tương ứng theo phạm vi kiểm thử lựa chọn có thể đạt được.

6.1.2 Tài liệu đầu vào

1) Tất cả các tài liệu về hệ thống, phần cứng và phần mềm cần thiết như được quy định trong Kế hoạch thẩm tra phần mềm.

6.1.3 Tài liệu đầu ra

1) Chỉ dẫn kiểm thử tổng thể phần mềm
2) Báo cáo kiểm thử tổng thể phần mềm
3) Chỉ dẫn kiểm thử tích hợp phần mềm
4) Báo cáo kiểm thử tích hợp phần mềm
5) Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng
6) Báo cáo kiểm thử tích hợp phần mềm / phần cứng
7) Chỉ dẫn kiểm thử thành phần phần mềm
8) Báo cáo kiểm thử thành phần phần mềm

6.1.4 Các yêu cầu

6.1.4.1 Đơn vị thẩm tra có thể chấp nhận các kiểm thử được các đơn vị khác tiến hành, như Đơn vị quản lý các yêu cầu, Đơn vị thiết kế hoặc Đơn vị thực hiện, nếu được ghi lại đầy đủ và phù hợp với các yêu cầu dưới đây.

6.1.4.2 Thiết bị đo đạc sử dụng để kiểm thử phải được hiệu chỉnh phù hợp. Mọi chương trình, phần cứng hoặc phần mềm được sử dụng để kiểm thử phải được thể hiện là phù hợp với mục đích.

6.1.4.3 Kiểm thử phần mềm phải được ghi lại trong Chỉ dẫn kiểm thử và Báo cáo kiểm thử, như được quy định trong các tài liệu dưới đây.

6.1.4.4 Từng Chỉ dẫn kiểm thử phải ghi lại những vấn đề sau:

a) Các mục tiêu kiểm thử.

b) Các trường hợp kiểm thử, dữ liệu kiểm thử, và các kết quả mong muốn.

c) Loại kiểm thử được tiến hành.

d) Môi trường kiểm thử, công cụ, cấu hình và chương trình.

e) Chỉ tiêu kiểm thử sẽ được kết luận khi hoàn thành kiểm thử.

f) Chỉ tiêu và mức độ kiểm thử đạt được.

g) Vai trò và trách nhiệm của các cá nhân liên quan trong quá trình kiểm thử.

h) Các yêu cầu được đề cập trong chỉ dẫn kiểm thử.

i) Lựa chọn và sử dụng thiết bị kiểm thử phần mềm.

6.1.4.5 Báo cáo kiểm thử phải được lập như sau:

a) Báo cáo kiểm thử phải đề cập tới tên của Đơn vị kiểm thử, nêu rõ các kết quả kiểm thử và xác định các mục tiêu kiểm thử và chỉ tiêu kiểm thử trong Chỉ dẫn kiểm thử có được đáp ứng. Các thất bại phải được lưu lại và tổng kết.

b) Các trường hợp kiểm thử và các kết quả phải được ghi lại, nên theo dạng để máy có thể đọc được cho các phân tích sau này.

c) Các kiểm thử phải có thể thực hiện lại và được thực hiện bằng phương pháp tự động nếu khả thi.

d) Các mã mô tả kiểm thử để chạy kiểm thử tự động phải được thẩm tra.

e) Nhận dạng và cấu hình của tất cả các hạng mục liên quan (phần cứng được sử dụng, phần mềm được sử dụng, thiết bị được sử dụng, thiết bị hiệu chuẩn, cũng như các dạng thông tin của chỉ dẫn kiểm thử) phải được lưu lại.

f) Việc đánh giá mức độ kiểm thử và việc hoàn thành kiểm thử phải được nêu ra và mọi sai lệch phải được ghi lại.

6.2 Thẩm tra phần mềm

6.2.1 Mục tiêu

6.2.1.1 Mục tiêu của việc thẩm tra phần mềm là để kiểm tra và đi đến kết luận dựa trên bằng chứng mà các đối tượng đầu ra (quá trình, tài liệu, phần mềm hoặc ứng dụng) của một giai đoạn phát triển cụ thể đáp ứng được các yêu cầu và các kế hoạch một cách hoàn thiện, chính xác và nhất quán. Các hoạt động này sẽ được Đơn vị thẩm tra quản lý

6.2.2 Các tài liệu đầu vào

1) Tất cả các tài liệu về hệ thống, phần mềm và phần cứng cần thiết.

6.2.3 Các **tài liệu đầu ra**

1) Kế hoạch thẩm tra phần mềm

2) Báo cáo thẩm tra phần mềm

3) Báo cáo thẩm tra đảm bảo chất lượng phần mềm

6.2.4 Các yêu cầu

6.2.4.1 Tối thiểu việc thẩm tra phải được ghi lại trong Kế hoạch thẩm tra phần mềm và trong một hoặc nhiều Báo cáo thẩm tra (liên quan đến quá trình).

6.2.4.2 Kế hoạch thẩm tra phần mềm phải được lập thành văn bản, do trách nhiệm của đơn vị thẩm tra, trên cơ sở các tài liệu cần thiết.

Các yêu cầu từ mục 6.2.4.3 đến 6.2.4.9 tham chiếu theo Kế hoạch thẩm tra phần mềm.

6.2.4.3 Kế hoạch thẩm tra phần mềm phải mô tả các hoạt động được tiến hành để đảm bảo cho việc thẩm tra được chính xác và các thiết kế cụ thể hoặc các yêu cầu thẩm tra khác được đưa ra một cách phù hợp.

6.2.4.4 Trong quá trình phát triển (và dựa vào quy mô của hệ thống), kế hoạch có thể được chia nhỏ thành một số tài liệu con và có thể được bổ sung thêm khi các yêu cầu chi tiết của việc thẩm tra trở nên rõ ràng hơn.

6.2.4.5 Kế hoạch thẩm tra phần mềm phải ghi lại tất cả các chỉ tiêu, kỹ thuật và chương trình được sử dụng trong quá trình thẩm tra. Kế hoạch thẩm tra phần mềm phải bao gồm các kỹ thuật và các biện pháp được lựa chọn trong các bảng A.5, A.6, A.7 và A.8. Việc kết hợp được lựa chọn phải được kết luận là thỏa mãn các yêu cầu 4.8, 4.9, 4.10.

6.2.4.6 Kế hoạch thẩm tra phần mềm phải mô tả các hoạt động được tiến hành để đảm bảo sự chính xác và thống nhất tương ứng với đầu vào cho giai đoạn đó. Những hoạt động này bao gồm cả việc rà soát, kiểm thử và tích hợp.

6.2.4.7 Trong từng giai đoạn phát triển, phải thể hiện được các yêu cầu về chức năng, hoạt động và an toàn đã được đáp ứng.

6.2.4.8 Các kết quả của từng việc thẩm tra phải được lưu lại theo dạng xác định hoặc được tham chiếu trong Kế hoạch thẩm tra phần mềm.

6.2.4.9 Kế hoạch thẩm tra phần mềm phải đề cập đến các vấn đề sau:

a) Việc lựa chọn chiến lược thẩm tra và các kỹ thuật (để trách sự phức tạp vô lý trong quá trình đánh giá việc thẩm tra và kiểm thử, phải đưa ra lựa chọn để chọn lọc các kỹ thuật có thể sẵn sàng phân tích).

b) Lựa chọn các kỹ thuật từ các Bảng A.5, A.6, A.7 và A.8.

c) Lựa chọn và ghi lại các hoạt động thẩm tra.

d) Đánh giá các kết quả thẩm tra thu được.

e) Đánh giá các yêu cầu về an toàn và độ chắc chắn.

f) Vai trò và các trách nhiệm của các cá nhân liên quan đến quá trình thẩm tra.

g) Mức độ phạm vi kiểm thử dựa trên chức năng được yêu cầu đạt được.

h) Kết cấu và nội dung của từng bước thẩm tra, đặc biệt đối với Thẩm tra yêu cầu phần mềm (7.2.4.22), Thẩm tra cấu trúc và thiết kế phần mềm (7.3.4.41, 7.3.4.42), Thẩm tra thành phần phần mềm (7.4.4.13), Thẩm tra mã nguồn phần mềm (7.5.4.10) và Thẩm tra tích hợp (7.6.4.13) sao cho tạo điều kiện thuận lợi cho việc xem xét theo Kế hoạch thẩm tra phần mềm.

6.2.4.10 Báo cáo thẩm tra đảm bảo chất lượng phầm mềm phải được lập thành văn bản, do trách nhiệm của đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào từ mục 6.2.2.

Yêu cầu trong mục 6.2.4.11 tham chiếu đến Báo cáo thẩm tra đảm bảo chất lượng phần mềm.

6.2.4.11 Khi đã lập được kế hoạch thẩm tra phần mềm, việc thẩm tra phải đề cập tới

a) Kế hoạch thẩm tra phần mềm đáp ứng các yêu cầu chung về khả năng sẵn sàng và khả năng theo dõi theo vết trong các mục 5.3.2.7 đến 5.3.2.10 và trong mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong 6.2.4.3 đến 6.2.4.9.

b) Sự thống nhất nội bộ của Kế hoạch thẩm tra phần mềm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra đảm bảo chất lượng phần mềm.

6.2.4.12 Mọi Báo cáo thẩm tra phần mềm phải được viết thành văn bản, do trách nhiệm của đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào. Các báo cáo này có thể được phân vùng để cho rõ ràng, thuận tiện và phải tuân theo Kế hoạch thẩm tra phần mềm. Yêu cầu trong mục 6.2.4.13 tham chiếu tới Báo cáo thẩm tra phần mềm.

6.2.4.13 Mỗi báo cáo thẩm tra phần mềm phải ghi lại các vấn đề sau:

a) Nhận dạng và cấu hình của các hạng mục được thẩm tra, cũng như tên đơn vị thẩm tra.

b) Các hạng mục không thỏa mãn các chỉ dẫn kỹ thuật.

c) Các thành phần, dữ liệu, cấu trúc và thuật toán được thay đổi kém linh hoạt theo vấn đề.

d) Các lỗi được phát hiện hoặc các sai lệch.

e) Việc đáp ứng hoặc sai lệch so với Kế hoạch thẩm tra phần mềm (trong tình huống sai lệch Báo cáo thẩm tra phải giải tích sự sai lệch là quan trọng hay không quan trọng).

f) Các giả thiết nếu có.

g) Tổng hợp các kết quả thẩm tra.

6.3 Thẩm định phần mềm

6.3.1 Mục tiêu

6.3.1.1 Mục tiêu của việc thẩm định phần mềm là để chứng minh các quá trình và các kết quả thuộc mức toàn vẹn về an toàn phần mềm được xác định, đáp ứng các yêu cầu phần mềm và phù hợp với ứng dụng dự định. Hoạt động này được Đơn vị thẩm định thực hiện.

6.3.1.2 Các hoạt động thẩm định chính là để chứng minh bằng phân tích và/hoặc kiểm thử xem tất cả các yêu cầu phần mềm được quy định, thực hiện, kiểm thử và đáp ứng theo yêu cầu của SIL có thể áp dụng, và để đánh giá mức độ quan trọng về an toàn của tất cả các vấn đề không bình thường và sự không phù hợp dựa trên các kết quả của việc rà soát, phân tích và kiểm thử.

6.3.2 Các tài liệu đầu vào

Tất cả các tài liệu về hệ thống, phần mềm và phần cứng như được quy định trong tiêu chuẩn này.

6.3.3 Các tài liệu đầu ra

1) Kế hoạch thẩm định phần mềm

2) Báo cáo thẩm định phần mềm

3) Báo cáo thẩm tra thẩm định phần mềm

6.3.4 Các yêu cầu

6.3.4.1 Các hoạt động thẩm định phần mềm phải được xây dựng và thực hiện, với các kết quả được đánh giá, bởi một đơn vị thẩm định ở mức độ độc lập phù hợp như được quy định trong mục 5.1.

6.3.4.2 Tối thiểu việc thẩm định phải được ghi lại trong Kế hoạch thẩm định phần mềm và Báo cáo thẩm định phần mềm, như được quy định dưới đây.

6.3.4.3 Kế hoạch thẩm định phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm định, trên cơ sở của các tài liệu đầu ra.

Các yêu cầu từ mục 6.3.4.4 đến 6.3.4.6 tham chiếu Kế hoạch thẩm định phần mềm.

6.3.4.4 Kế hoạch thẩm định phần mềm phải bao gồm cả việc tổng hợp kết luận về chiến lược thẩm định được lựa chọn. Theo mức toàn vẹn về an toàn phần mềm được yêu cầu, việc kết luận phải bao gồm việc xem xét:

a) Các kỹ thuật thủ công hoặc tự động, hoặc cả hai.

b) Các kỹ thuật tĩnh hoặc động, hoặc cả hai.

c) Các kỹ thuật phân tích hoặc thống kê, hoặc cả hai.

d) Kiểm thử trong môi trường thực hoặc mô phỏng, hoặc cả hai.

6.3.4.5 Kế hoạch thẩm định phần mềm phải xác định rõ các bước cần thiết để chứng minh sự đáp ứng đầy đủ các yêu cầu về an toàn được đưa ra trong Chỉ dẫn các yêu cầu an toàn hệ thống của Chỉ dẫn kỹ thuật phần mềm.

6.3.4.6 Kế hoạch thẩm định phần mềm phải xác định rõ các bước cần thiết để chứng minh sự phù hợp của Chỉ dẫn thử nghiệm phần mềm tổng thể như là một kiểm thử dựa trên Chỉ dẫn các yêu cầu phần mềm.

6.3.4.7 Báo cáo thẩm định phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm định, trên cơ sở các tài liệu đầu vào.

Các yêu cầu từ mục 6.3.4.8 đến 6.3.4.11 tham chiếu đến Báo cáo thẩm định phần mềm.

6.3.4.8 Các kết quả của việc thẩm định phải được ghi lại trong Báo cáo thẩm định phần mềm.

6.3.4.9 Đơn vị thẩm định phải kiểm tra quá trình thẩm tra có được hoàn thành không.

6.3.4.10 Báo cáo thẩm định phần mềm phải nêu rõ đầy đủ cơ sở phần mềm mà đã được thẩm định.

6.3.4.11 Báo cáo thẩm định phải xác định rõ mọi sai khác được biết trong phần mềm và các tác động có thể có đối với việc sử dụng phần mềm.

6.3.4.12 Báo cáo thẩm tra thẩm định phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào từ mục 6.3.2.

Các yêu cầu từ mục 6.3.4.13 đến 6.3.4.14 tham chiếu đến Báo cáo thẩm tra thẩm định phần mềm.

6.3.4.13 Khi đã thiết lập được Kế hoạch thẩm định phần mềm, việc thẩm tra phải đề cập tới:

a) Kế hoạch thẩm định phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể từ mục 6.3.4.4 đến 6.3.4.6.

b) Sự thống nhất nội bộ của Kế hoạch thẩm định phần mềm.

6.3.4.14 Khi đã thiết lập được Báo cáo thẩm định phần mềm, việc thẩm tra phải đề cập tới:

a) Báo cáo thẩm định phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng theo dõi theo vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể từ mục 7.7.4.7 đến 7.7.4.11.

b) Sự thống nhất nội bộ của Báo cáo thẩm định phần mềm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra thẩm định phần mềm

6.3.4.15 Đơn vị thẩm định phải được giao quyền để yêu cầu hoặc thực hiện các hoạt động rà soát, phân tích và kiểm thử bổ sung.

6.3.4.16 Phần mềm chỉ có thể được đưa ra chạy sau khi có sự chấp thuận của Đơn vị thẩm định.

6.3.4.17 Việc mô phỏng và lập mô hình có thể được sử dụng để hỗ trợ cho quá trình thẩm định.

6.4 Đánh giá phần mềm

6.4.1 Mục tiêu

6.4.1.1 Để đánh giá các quá trình vòng đời và các kết quả đầu ra sao cho phần mềm có mức toàn vẹn về an toàn xác định 1-4 và phù hợp với ứng dụng dự định.

6.4.1.2 Đối với phần mềm có mức SIL 0, các yêu cầu của tiêu chuẩn này phải được đáp ứng, nhưng nếu có chứng chỉ tuyên bố phù hợp với tiêu chuẩn TCVN ISO 9001, thì sẽ không cần phải đánh giá.

6.4.2 Tài liệu đầu vào

1) Chỉ dẫn các yêu cầu an toàn hệ thống

2) Chỉ dẫn các yêu cầu phần mềm

3) Tất cả các tài liệu khác cần thiết để tiến hành quá trình đánh giá

6.4.3 Tài liệu đầu ra

1) Kế hoạch đánh giá phần mềm

2) Báo cáo đánh giá phần mềm

3) Báo cáo thẩm tra đánh giá phần mềm

6.4.4 Các yêu cầu

6.4.4.1 Việc đánh giá phần mềm phải được Đơn vị đánh giá tiến hành, với mức độ độc lập như trong mục 5.1.2.6 và 5.1.2.7.

6.4.4.2 Phần mềm có Báo cáo đánh giá phần mềm của một đơn vị đánh giá khác sẽ không cần phải đánh giá lại. Đơn vị đánh giá phải kiểm tra xem phần mềm có phù hợp với việc sử dụng dự định trong môi trường dự định, và việc đánh giá trước đó tuyên bố phần mềm đã đạt được mức toàn vẹn về an toàn ít nhất bằng với mức được yêu cầu.

6.4.4.3 Đơn vị đánh giá phải xem xét tất cả các tài liệu liên quan tới dự án trong suốt quá trình phát triển.

6.4.4.4 Kế hoạch đánh giá phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị đánh giá, trên cơ sở các tài liệu đầu vào ở mục 6.4.2. Nếu phù hợp, có thể sử dụng Kế hoạch đánh giá phần mềm chung hoặc quy trình được ghi lại trước đó. Yêu cầu trong mục 6.4.4.5 tham chiếu tới Kế hoạch đánh giá phần mềm.

6.4.4.5 Kế hoạch đánh giá phần mềm phải bao gồm các nội dung sau:

a) Các nội dung cần đánh giá.

b) Các hoạt động trong quá trình đánh giá và các mối liên quan sau đó đối với các hoạt động kỹ thuật.

c) Các tài liệu cần được xem xét.

d) Các kết luận về chỉ tiêu Đạt/Không đạt và cách xử lý các trường hợp không phù hợp.

e) Các yêu về nội dung và hình thức của Báo cáo đánh giá phần mềm.

6.4.4.6 Báo cáo thẩm tra đánh giá phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào trong mục 6.4.2.

Các yêu cầu trong mục 6.4.4.7 tham chiếu tới Báo cáo thẩm tra đánh giá phần mềm.

6.4.4.7 Khi đã thiết lập được Kế hoạch đánh giá phần mềm, việc thẩm tra phải đề cập tới:

a) Kế hoạch đánh giá phần mềm đáp ứng các yêu cầu chung về khả năng sẵn sàng và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 6.4.4.5.

b) Sự thống nhất nội bộ của Kế hoạch đánh giá phần mềm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra đánh giá phần mềm.

6.4.4.8 Đơn vị đánh giá phải đánh giá xem phần mềm của hệ thống có phù hợp với mục đích sử dụng và phản hồi đúng theo các vấn đề về an toàn phát sinh từ Chỉ dẫn các yêu cầu an toàn hệ thống.

6.4.4.9 Đơn vị đánh giá phải đánh giá nếu lựa chọn và áp dụng nhóm các kỹ thuật trong Phụ lục A, phù hợp với quá trình phát triển dự định theo mức toàn vẹn về an toàn được yêu cầu.

Ngoài ra Đơn vị đánh giá phải xem xét mức độ áp dụng của từng kỹ thuật trong Phụ lục A, ví dụ: liệu nó có được áp dụng cho tất cả hoặc chỉ một phần của phần mềm, và phải tìm kiếm các bằng chứng về việc nó đã được áp dụng đúng.

6.4.4.10 Đơn vị đánh giá phải đánh giá về hệ thống quản lý cấu hình, sự thay đổi, bằng chứng về việc sử dụng và áp dụng nó.

6.4.4.11 Đơn vị đánh giá phải xem xét bằng chứng về năng lực của nhân viên dự án theo Phụ lục B và phải đánh giá tổ chức phát triển phần mềm theo mục 5.1.

6.4.4.12 Với mọi phần mềm có các điều kiện áp dụng liên quan tới an toàn, Đơn vị đánh giá phải kiểm tra các sai lệch được ghi lại, sự không phù hợp với các yêu cầu và các vấn đề không phù hợp được ghi lại nếu có tác động đến an toàn, và đưa ra kết luận liệu việc giải thích trong dự án có thể chấp nhận được. Kết quả phải được nêu rõ trong báo cáo đánh giá.

6.4.4.13 Đơn vị đánh giá phải đánh giá các hoạt động thẩm tra và thẩm định và bằng chứng hỗ trợ.

6.4.4.14 Đơn vị đánh giá phải đồng ý về phạm vi và nội dung của kế hoạch kiểm thử phần mềm. Việc đồng ý này phải khẳng định việc có mặt Đơn vị đánh giá trong quá trình kiểm thử.

6.4.4.15 Đơn vị đánh giá có thể tiến hành các hoạt động kiểm toán và kiểm tra (ví dụ: chứng kiến các kiểm thử) trong suốt quá trình phát triển. Đơn vị đánh giá có thể yêu cầu các công việc thẩm tra và thẩm định bổ sung.

Chú thích: đó là cơ hội để Đơn vị đánh giá sớm tham gia vào dự án.

6.4.4.16 Báo cáo đánh giá phần mềm phải được lập thành văn bản do trách nhiệm của Đơn vị đánh giá. Các yêu cầu từ mục 6.4.4.17 đến 6.4.4.19 tham chiếu đến Báo cáo đánh giá phần mềm.

6.4.4.17 Báo cáo đánh giá phần mềm phải đáp ứng các yêu cầu của Kế hoạch đánh giá phần mềm và đưa ra kết luận và các khuyến nghị.

6.4.4.18 Đơn vị đánh giá phải ghi lại các hoạt động của mình trên cơ sở thống nhất với Báo cáo đánh giá phần mềm. Các hoạt động này phải được tổng hợp lại trong Báo cáo đánh giá phần mềm.

6.4.4.19 Đơn vị đánh giá phải xác định rõ và đánh giá mọi sự không phù hợp với các yêu cầu của tiêu chuẩn này và kết luận về tác động của nó đến kết quả cuối cùng. Mọi vấn đề không phù hợp và các lý do không phù hợp phải được liệt kê trong Báo cáo đánh giá phần mềm.

6.5 Đảm bảo chất lượng phần mềm

6.5.1 Mục tiêu

6.5.1.1 Để xác định, giám sát và kiểm soát tất cả các hoạt động trên, thì cần thiết phải có cả biện pháp kỹ thuật và quản lý để đảm bảo phần mềm đạt được chất lượng yêu cầu. Việc này là cần thiết để tạo ra biện pháp phòng vệ chất lượng để phòng các lỗi mang tính hệ thống và để đảm bảo có thể thiết lập được phương thức đánh giá cho phép các hoạt động thẩm tra và thẩm định được tiến hành có hiệu quả.

6.5.1.2 Để đưa ra các bằng chứng về việc tất cả các hoạt động trên đã được tiến hành.

6.5.2 Tài liệu đầu vào

Tất cả các tài liệu sẵn có trong từng giai đoạn của vòng đời.

6.5.3 Tài liệu đầu ra

1) Kế hoạch đảm bảo chất lượng phần mềm

2) Kế hoạch quản lý cấu hình phần mềm, nếu không sẵn có ở mức độ hệ thống

3) Báo cáo thẩm tra đảm bảo chất lượng phần mềm

6.5.4 Các yêu cầu

6.5.4.1 Tất cả các kế hoạch phải được đưa ra ngay khi bắt đầu dự án và được cập nhật trong suốt vòng đời.

6.5.4.2 Tất cả các đơn vị tham gia vào việc phát triển phần mềm phải thực hiện và sử dụng Hệ thống đảm bảo chất lượng phù hợp với tiêu chuẩn TCVN ISO 9000, để hỗ trợ cho các yêu cầu trong tiêu chuẩn này. Chứng nhận phù hợp TCVN ISO 9001 được khuyến nghị cao.

6.5.4.3 Kế hoạch đảm bảo chất lượng phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào của mục 6.5.2.

Các yêu cầu từ mục 6.5.4.4 đến 6.5.4.6 tham chiếu Kế hoạch đảm bảo chất lượng phần mềm.

6.5.4.4 Kế hoạch đảm bảo chất lượng phần mềm phải được lập thành văn bản và phải cụ thể theo dự án. Phải thực hiện các yêu cầu của mục 6.5.4.5.

6.5.4.5 Tối thiểu, phải cụ thể hoặc tham chiếu các hạng mục sau trong Kế hoạch đảm bảo chất lượng phần mềm.

a) Xác định mô hình vòng đời, bao gồm:

1) Các hoạt động và các nhiệm vụ cơ bản thống nhất với các kế hoạch đặt ra, ví dụ: Kế hoạch an toàn được thiết lập ở cấp Hệ thống;

2) Chỉ tiêu đầu vào và đầu ra của từng hoạt động;

3) Các đầu vào và đầu ra của từng hoạt động;

4) Các hoạt động đảm bảo chất lượng chính;

5) Các đơn vị chịu trách nhiệm cho từng hoạt động.

b) Cấu trúc tài liệu.

c) Kiểm soát tài liệu:

1) Các vai trò liên quan đến việc lập văn bản, kiểm tra và phê duyệt;

2) Phạm vi phân phối;

3) Đạt được.

d) Theo dõi và truy vết các sai lệch;

e) Phương pháp, biện pháp và các chương trình để đảm bảo chất lượng theo mức toàn vẹn về an toàn đã được phân bổ (xem Phụ lục A).

f) Các căn cứ của từng sự kết hợp các kỹ thuật hoặc biện pháp được lựa chọn theo Phụ lục A phù hợp với mức toàn vẹn về an toàn phần mềm được xác định, theo mục 4.7 đến 4.9.

Một số Kế hoạch đảm bảo chất lượng phần mềm cần thông tin có thể có trong các tài liệu khác, như Kế hoạch Quản lý cấu hình phần mềm, Kế hoạch bảo trì, Kế hoạch thẩm tra phần mềm, và Kế hoạch thẩm định phần mềm. Các mục trong Kế hoạch đảm bảo chất lượng phần mềm phải tham chiếu các tài liệu chứa thông tin liên quan. Trong mọi trường hợp nội dung của từng mục trong Kế hoạch đảm bảo chất lượng phần mềm phải được quy định trực tiếp hoặc được tham chiếu tới tài liệu khác.

Các tài liệu được tham chiếu phải được rà soát để đảm bảo chúng đưa ra được tất cả các thông tin cần thiết và đề cập đến đầy đủ các yêu cầu của tiêu chuẩn này.

6.5.4.6 Các hoạt động, tài liệu đảm bảo chất lượng theo yêu cầu của tất cả các mục quy định trong tiêu chuẩn này phải được quy định hoặc tham chiếu trong Kế hoạch đảm bảo chất lượng phần mềm và được cụ thể theo dự án cụ thể.

6.5.4.7 Báo cáo thẩm tra đảm bảo chất lượng phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào của mục 6.5.2.

Các yêu cầu trong mục 6.5.4.8 tham chiếu tới Báo cáo thẩm tra đảm bảo chất lượng phần mềm.

6.5.4.8 Khi đã thiết lập được Kế hoạch đảm bảo chất lượng phần mềm, việc thẩm tra phải đề cập đến:

a) Kế hoạch đảm bảo chất lượng phần mềm đáp ứng các yêu cầu chung về khả năng sẵn sàng và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể từ mục 6.5.4.4 đến 6.5.4.6.

b) Sự thống nhất nội bộ của Kế hoạch đảm bảo chất lượng phần mềm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra đảm bảo chất lượng phần mềm.

6.5.4.9 Từng tài liệu được lập kế hoạch phải có nội dung quy định chi tiết về việc cập nhật trong toàn bộ dự án: tần suất, trách nhiệm, phương pháp.

6.5.4.10 Từng tài liệu phần mềm và sản phẩm chuyển giao phải được đặt dưới sự kiểm soát cấu hình tại thời điểm phát hành lần đầu.

6.5.4.11 Các thay đổi đối với tất cả các hạng mục trong Kiểm soát quản lý cấu hình phải được phê duyệt và ghi lại.

6.5.4.12 Đề bổ sung cho việc phát triển phần mềm, Hệ thống quản lý cấu hình cũng phải đề cập đến môi trường phát triển phần mềm được sử dụng trong vòng đời đầy đủ.

Việc bổ sung này (cần thiết để tái lập việc phát triển và các hoạt động bảo trì) phải bao gồm tất cả các chương trình, các chương trình chuyển đổi, dữ liệu và các tệp kiểm thử, các tệp thông số hóa, và các nền tảng phần cứng hỗ trợ.

6.5.4.13 Nhà cung cấp phải thiết lập các quy trình lưu trữ và duy trì để kiểm soát các nhà cung cấp bên ngoài, bao gồm:

- Các biện pháp và các biên bản liên quan để đảm bảo phần mềm được các đơn vị bên ngoài cung cấp tuân thủ chặt chẽ các yêu cầu được thiết lập. Phần mềm được phát triển trước đó phải được đảm bảo phù hợp với mức toàn vẹn về an toàn và độ tin cậy được yêu cầu. Phần mềm mới phải được phát triển và bảo trì phù hợp với Kế hoạch đảm bảo chất lượng phần mềm của nhà cung cấp hoặc Kế hoạch đảm bảo chất lượng phần mềm được nhà cung cấp bên ngoài chuẩn bị phù hợp với Kế hoạch đảm bảo chất lượng phần mềm của nhà cung cấp;

- Các biện pháp và các biên bản liên quan để đảm bảo các yêu cầu được đưa ra cho Nhà cung cấp bên ngoài là phù hợp và hoàn chỉnh.

6.5.4.14 Phải xem xét cẩn thận khả năng theo dõi theo vết theo các yêu cầu trong việc thẩm định hệ thống liên quan tới an toàn và phải đưa ra cách thức để chứng minh trong tất cả các giai đoạn của vòng đời.

6.5.4.15 Trong nội dung của tiêu chuẩn này, và theo mức độ phù hợp với mức toàn vẹn về an toàn phần mềm được quy định, khả năng truy vết phải đề cập cụ thể tới:

a) Khả năng truy vết các yêu cầu theo thiết kế hoặc các đối tượng khác đáp ứng chúng.

b) Khả năng truy vết các đối tượng thiết kế theo các đối tượng thực hiện chúng.

c) Khả năng truy vết các yêu cầu và các đối tượng thiết kế theo các kiểm thử (thành phần, sự tích hợp, các kiểm thử toàn bộ) và các phân tích để xác nhận chúng.

Khả năng truy vết phải phụ thuộc vào việc quản lý cấu hình.

6.5.4.16 Trong các trường hợp đặc biệt, ví dụ: phần mềm đã có trước đó hoặc phần mềm mẫu, khả năng theo dõi theo vết có thể được thiết lập sau khi chạy và/hoặc ghi lại mã, nhưng trước khi thẩm tra/thẩm định. Trong trường hợp này, phải thể hiện được việc thẩm tra/thẩm định là hiệu quả, đúng với khả năng theo dõi theo vết trong tất cả các giai đoạn.

6.5.4.17 Các đối tượng của các yêu cầu, thiết kế hoặc sự hoạt động mà không thể truy vết đầy đủ thì phải được chứng minh là không phụ thuộc vào mức toàn vẹn về an toàn của hệ thống.

6.6 Cải tiến và kiểm soát sự thay đổi

6.6.1 Mục tiêu

6.6.1.1 Để đảm bảo phần mềm hoạt động như được yêu cầu, duy trì mức toàn vẹn về an toàn và độ tin cậy khi cải tiến phần mềm.

6.6.1.2 Các đối tượng này được quản lý bởi Đơn vị quản lý cấu hình.

6.6.2 Các tài liệu đầu vào

1) Kế hoạch đảm bảo chất lượng phần mềm

2) Kế hoạch quản lý cấu hình phần mềm

3) Tất cả các tài liệu thiết kế, phát triển và phân tích

4) Các yêu cầu thay đổi

5) Phân tích tác động của thay đổi và quá trình phê duyệt.

6.6.3 Các tài liệu đầu ra

1) Tất cả các tài liệu đầu vào bị thay đổi

2) Các biên bản thay đổi phần mềm (xem 9.2.4.11)

3) Các biên bản cấu hình mới

6.6.4 Các yêu cầu

6.6.4.1 Quá trình quản lý sự thay đổi phải xác định tối thiểu các vấn đề sau:

a) Các tài liệu cần thiết của các hoạt động báo cáo vấn đề và/hoặc sửa chữa, với mục đích đưa ra phản hồi cho đơn vị quản lý có trách nhiệm.

b) Phân tích thông tin thu thập được trong báo cáo vấn đề để xác định các nguyên nhân.

c) Các hoạt động sau đó để báo cáo, theo dõi và xử lý các vấn đề được xác định trong giai đoạn phát triển và trong quá trình bảo trì phần mềm.

d) Các trách nhiệm về tổ chức cụ thể liên quan đến việc phát triển và bảo trì phần mềm.

e) Cách thức áp dụng các kiểm soát để đảm bảo các hoạt động sửa chữa đã được thực hiện và có hiệu quả.

f) Phân tích tác động của thay đổi đối với thành phần phần mềm trong quá trình phát triển hoặc đã được chuyển giao.

g) Phân tích tác động phải nêu rõ các hoạt động thẩm tra lại, thẩm định lại, đánh giá lại cần thiết cho sự thay đổi.

h) Khi áp dụng nhiều thay đổi, phân tích tác động phải xem xét tác động tích lũy.

i) Chú thích: Một số thay đổi có thể yêu cầu tích lũy việc kiểm thử lại toàn bộ.

j) Quá trình phê duyệt trước khi thực hiện.

6.6.4.2 Tất cả các thay đổi phải bắt đầu từ việc quay lại một giai đoạn phù hợp ở trong vòng đời. Tất cả các giai đoạn sau đó phải được tiến hành phù hợp với các quy trình được quy định cho các giai đoạn cụ thể phù hợp với các yêu cầu trong tiêu chuẩn này.

6.7 Các ngôn ngữ và chương trình hỗ trợ

6.7.1 Mục tiêu

6.7.1.1 Mục tiêu là đưa ra bằng chứng cho thấy các hư hỏng tiềm ẩn của các chương trình không tác động xấu tới thông số đầu ra của bộ chương trình được tích hợp theo một phương thức liên quan đến an toàn mà không được phát hiện bằng các biện pháp tổ chức và / hoặc kỹ thuật bên ngoài chương trình đó. Để thực hiện được việc này, các chương trình phần mềm được phân thành ba loại là T1, T2 và T3 tương ứng (xem định nghĩa ở mục 3.1).

6.7.2 Các tài liệu đầu vào

Chỉ dẫn hoặc hướng dẫn sử dụng các chương trình.

6.7.3 Các tài liệu đầu ra

Báo cáo thẩm định các chương trình (khi cần thiết, xem 6.7.4.4 hoặc 6.7.4.6).

6.7.4 Các yêu cầu

6.7.4.1 Các chương trình phần mềm phải được lựa chọn như một phần không thể tách rời trong các hoạt động phát triển phần mềm.

Chú thích: Các chương trình phù hợp hỗ trợ cho việc phát triển phần mềm nên được sử dụng để làm tăng mức toàn vẹn của phần mềm bằng cách giảm khả năng phát sinh hoặc khả năng không phát hiện các sự cố trong quá trình phát triển. Ví dụ về các chương trình liên quan tới các giai đoạn của vòng đời phát triển phần mềm bao gồm:

a) Các chương trình chuyển đổi hoặc chuyển dịch phần mềm hoặc dạng thiết kế (ví dụ: nội dung hoặc sơ đồ) từ một cấp cơ bản sang một cấp khác: các chương trình cải tiến thiết kế, các chương trình biên dịch, các chương trình hợp ngữ, các chương trình kết nối, các chương trình chạy và các chương trình tạo mã.

b) Các chương trình thẩm tra và thẩm định như chương trình phân tích mã tĩnh, màn hình giám sát kiểm thử, các chương trình hỗ trợ chứng minh nguyên lý, các chương trình mô phỏng và các chương trình kiểm tra mô hình.

c) Các chương trình chuẩn đoán được sử dụng để duy trì và giám sát phần mềm dưới các điều kiện hoạt động.

d) Các chương trình về hạ tầng như các hệ thống hỗ trợ quá trinh xây dựng.

e) Các chương trình kiểm soát cấu hình như các chương trình kiểm soát định dạng.

f) Các chương trình dữ liệu ứng dụng tạo ra hoặc duy trì các dữ liệu cần thiết để xác định các thông số và thực hiện các chức năng hệ thống, ví dụ: các thông số về chức năng, các dải biên độ, các mức cảnh báo và đóng ngắt, các tính trạng đầu ra được chấp nhận khi có hư hỏng, các bố trí địa lý.

Các chương trình được lựa chọn nên có khả năng kết hợp lại được với nhau. Trong tiêu chuẩn này, các chương trình sẽ kết hợp với nhau nếu các kết quả từ một chương trình có nội dung và định dạng phù hợp để tự động nhập vào chương trình sau đó, từ đó giảm thiểu tối đa khả năng phát sinh lỗi của con người trong quá trình tái lập hoạt động ở các kết quả trung gian.

Phải lựa chọn các chương trình và chứng minh nó tương thích với các yêu cầu của việc ứng dụng.

Phải xem xét tính sẵn sàng của các chương trình phù hợp để đưa ra các dịch vụ cần thiết trong toàn bộ vòng đời của phần mềm.

6.7.4.2 Việc lựa chọn các chương trình loại T2 và T3 phải được nêu rõ lý do (xem 7.3.4.12). Việc giải thích phải bao gồm xác định các hư hỏng tiềm ẩn có thể có trong các đầu ra của chương trình và các biện pháp để tránh hoặc xử lý các hư hỏng này.

6.7.4.3 Tất cả các chương trình loại T2 và T3 phải có chỉ dẫn kỹ thuật hoặc hướng dẫn sử dụng xác định rõ ràng sự hoạt động của chương trình và mọi hướng dẫn hoặc ràng buộc liên quan đến việc sử dụng.

6.7.4.4 Đối tới từng chương trình loại T3, phải có sẵn các bằng chứng cho thấy đầu ra của chương trình phù hợp với chỉ dẫn kỹ thuật của đầu ra hoặc các hư hỏng trong đầu ra là được phát hiện. Bằng chứng có thể dựa trên cùng các bước cần thiết để xử lý thủ công giống như khi thay thế chương trình và căn cứ được đưa ra, nếu những bước này bị thay thế bằng các bước xử lý khác (ví dụ: thẩm định chương trình). Bằng chứng có thể dựa trên:

a) Sự kết hợp phù hợp về lịch sử quá trình sử dụng tốt trong cùng các môi trường và cho cùng các ứng dụng giống nhau (trong tổ chức hoặc các tổ chức khác).

b) Việc thẩm định chương trình như quy định trong mục 6.7.4.5.

c) Việc mã hóa ràng buộc khác nhau cho phép phát hiện và kiểm soát các hư hỏng gây ra các sự cố của chương trình,

d) Sự phù hợp về mức toàn vẹn về an toàn trong việc phân tích rủi ro các quá trình và các quy trình có sử dụng chương trình.

e) Các biện pháp phù hợp khác để tránh hoặc xử lý các hư hỏng của chương trình.

Chú thích 1: Lịch sử về định dạng có thể đảm bảo độ chắc chắn của chương trình và biên bản về các lỗi / các vấn đề không rõ ràng liên quan đến quá trình sử dụng của công cụ trong môi trường đó.

Chú thích 2: Bằng chứng được liệt kê cho loại T3 cũng có thể sử dụng cho các chương trình loại T2 khi kết luận về độ chính xác của kết quả.

6.7.4.5 Phải lưu lại các kết quả của việc thẩm định chương trình, bao gồm các kết quả sau:

a) Biên bản về các hoạt động thẩm định;

b) Phiên bản hướng dẫn sử dụng chương trình đang được sử dụng;

c) Các chức năng của chương trình đang được thẩm định;

d) Các chương trình và thiết bị được sử dụng;

e) Các kết quả của hoạt động thẩm định; các kết quả thẩm định được ghi lại phải nêu rõ hoặc phần mềm đã được thẩm định đạt hoặc các lý do không đạt;

f) Các trường hợp kiểm thử và các kết quả sử dụng cho các phân tích sau đó;

g) Sự không thống nhất giữa các kết quả mong muốn và các kết quả thực tế.

6.7.4.6 Khi không có bằng chứng về sự phù hợp với mục 6.7.4.4, phải có các biện pháp hiệu quả để kiểm soát các hư hỏng của phần mềm liên quan đến an toàn có thể được hoạt động do các lỗi thuộc về chương trình.

Chú thích 1: Ví dụ: sự phát sinh các mã ràng buộc khác nhau cho phép phát hiện và kiểm soát các hư hỏng gây ra các sự cố của chương trình chuyển đổi.

Chú thích 2: Ví dụ: Sự phù hợp về mục đích của các chương trình biên dịch không đáng tin có thể được giải thích như sau.

Mã đối tượng của chương trình biên dịch phụ thuộc vào sự kết hợp các kiểm thử, kiểm tra và phân tích, có thể đảm bảo độ chính xác của đoạn mã theo mức thống nhất với Mức toàn vẹn về an toàn mục tiêu. Cụ thể, áp dụng các nội dung sau cho tất cả các kiểm thử, kiểm tra và phân tích:

- Việc kiểm thử phải cho thấy chương trình thực hiện đã xử lý ở mức độ cao đầy đủ của đoạn mã chạy. Nếu có đoạn mã nào đó không đạt được bằng kiểm thử, phải thể hiện bằng kiểm tra hoặc phân tích cho thấy chức năng liên quan hoạt động chính xác khi đoạn mã được truy cập đến đối tượng.

- Các kiểm tra và phân tích được áp dụng cho đoạn mã đối tượng và phải thể hiện là có khả năng phát hiện ra các dạng lỗi có thể phát sinh từ một sai sót trong chương trình biên dịch.

- Sau khi kiểm thử, kiểm tra và phân tích, chương trình biên dịch sẽ không thực hiện thêm các chuyển đổi.

- Nếu có thêm quá trình biên dịch hoặc chuyển đổi được tiến hành thì tất cả các kiểm thử, kiểm tra và phân tích sẽ được lặp lại.

6.7.4.7 Việc thể hiện phần mềm hoặc thiết kế (gồm có ngôn ngữ lập trình) được lựa chọn phải:

a) Có một chương trình chuyển đổi được đánh giá về mức độ phù hợp với mục đích, bao gồm cả việc đánh giá theo các tiêu chuẩn quốc tế hoặc quốc gia, nếu phù hợp.

b) Phù hợp với các đặc tính của việc ứng dụng.

c) Chứa các tính năng tạo điều kiện thuận lợi để phát hiện các lỗi về thiết kế hoặc lập trình,

d) Hỗ trợ các tính năng phù hợp với biện pháp thiết kế.

Ngôn ngữ lập trình là một trong các loại thể hiện phần mềm hoặc thiết kế. Chương trình chuyển đổi sẽ chuyển dạng thể hiện phần mềm hoặc thiết kế (ví dụ: nội dung hoặc sơ đồ) từ mức độ nền tảng sang mức độ khác. Ví dụ về chương trình chuyển đổi bao gồm: các chương trình cải tiến thiết kế, các chương trình chạy, chương trình hợp ngữ, các chương trình liên kết, các chương trình tải và các chương trình tạo mã.

Việc đánh giá Chương trình chuyển đổi có thể được thực hiện cho một dự án ứng dụng cụ thể, hoặc cho một loại ứng dụng. Ở trường hợp loại ứng dụng, người sử dụng chương trình phải có sẵn tất cả các thông tin cần thiết về chương trình liên quan đến việc sử dụng dự định và phù hợp với việc sử dụng chương trình. Việc đánh giá chương trình đối với một dự án cụ thể có thể được giảm bớt sau đó để kiểm tra khả năng phù hợp tổng thể của chương trình cho dự án và sự phù hợp với “chỉ dẫn kỹ

thuật hoặc hướng dẫn sử dụng" (ví dụ: sử dụng đúng chương trình). Việc sử dụng đúng chương trình có thể bao gồm các hoạt động thẩm tra bổ sung có trong dự án cụ thể.

Có thể sử dụng các hỗ trợ trong quá trình thẩm định để đánh giá mức độ phù hợp với mục đích của chương trình chuyển đổi theo chỉ tiêu xác định, chỉ tiêu này phải có các yêu cầu về chức năng và phi chắc năng. Đối với các yêu cầu chức năng của chương trình chuyển đổi, việc kiểm thử động có thể là một kỹ thuật thẩm định chính. Nếu có thể thì phải sử dụng các hỗ trợ kiểm thử tự động.

6.7.4.8 Khi không thể đáp ứng đầy đủ 6.7.4.7, phải đánh giá và làm rõ mức độ phù hợp với mục đích của ngôn ngữ lập trình và các biện pháp bổ sung đề cập đến tất cả các thiếu sót đã được xác định của ngôn ngữ lập trình.

Chú thích: Xem chú thích 2 trong mục 6.7.4.6.

6.7.4.9 Khi thực hiện việc tạo mã tự động hoặc chuyển đổi tự động tương đương, khả năng phù hợp của chương trình chuyển đổi tự động trong quá trình phát triển phần mềm liên quan tới an toàn phải được đánh giá tại thời điểm trong vòng đời phát triển khi lựa chọn được các chương trình hỗ trợ phát triển.

6.7.4.10 Quản lý cấu hình phải đảm bảo rằng đối với các chương trình loại T2 và T3, chỉ sử dụng các phiên bản đã được làm rõ.

6.7.4.11 Mỗi phiên bản mới của chương trình được sử dụng phải được làm rõ căn cứ (xem Bảng 1). Việc làm rõ này có thể dựa trên bằng chứng được đưa ra cho phiên bản trước đó nếu có bằng chứng đầy đủ về việc:

a) Các sai lệch về chức năng (nếu có) sẽ không ảnh hưởng đến khả năng tương thích của chương trình với phần còn lại của bộ chương trình.

b) Phiên bản mới không có khả năng có các sự cố mới đáng kể và chưa được nhận biết.

Chú thích: Bằng chứng về phiên bản mới không có khả năng có các lỗi mới chưa được nhận biết có thể dựa trên việc xác định tin cậy các thay đổi và dựa trên các phân tích về các hoạt động thẩm tra và thẩm định được tiến hành.

6.7.4.12 Mối quan hệ giữa các loại chương trình và các điều khoản áp dụng quy định trong Bảng 1.

Bảng 1. Mối quan hệ giữa các loại chương trình và các điều khoản áp dụng

Loại chương trình	Điều khoản có thể áp dụng
T1	6.7.4.1
T2	6.7.4.1, 6.7.4.2, 6.7.4.3, 6.7.4.10, 6.7.4.11
T3	6.7.4.1, 6.7.4.2, 6.7.4.3, 6.7.4.4, 6.7.4.5 hoặc 6.7.4.6, 6.7.4.7, 6.7.4.8, 6.7.4.9, 6.7.4.10, 6.7.4.11

7 Phát triển phần mềm chung

7.1 Vòng đời và tài liệu ghi lại đối với phần mềm chung

7.1.1 Mục tiêu

7.1.1.1 Để đưa ra sự mô tả về chính phần mềm, từ các mức độ trừu tượng cao hơn xuống các cải tiến chi tiết, để tạo ra cơ sở chứng minh độ an toàn đạt được cũng như cho các hoạt động bảo trì trong tương lai.

7.1.2 Các yêu cầu

7.1.2.1 Phải lập ra các tài liệu được liệt kê trong Bảng A.1 cho phần mềm chung theo mức toàn vẹn về an toàn phần mềm được yêu cầu.

7.1.2.2 Chuỗi các tài liệu chuyển giao như được mô tả trong Bảng A.1 thể hiện mô hình thác nước tuyến tính lý tưởng. Tuy nhiên mô hình này không dự định để tham chiếu cho các hoạt động chiến lược và liên kết, do thường khó khăn trong quá trình đạt được sự phù hợp cao trong hoạt động thực tế. Các giai đoạn có thể chồng lên nhau nhưng các hoạt động thẩm tra và thẩm định phải chứng minh sự thống nhất của các đầu vào và đầu ra (các tài liệu và phần mềm) ở giữa và trong các giai đoạn.

Tuy nhiên, mục đích chính của việc dự kiến tài liệu lưu trữ là để đưa ra sự mô tả về chính phần mềm, từ mức độ trừu tượng cao hơn xuống các cải tiến chi tiết, để tạo ra cơ sở cho việc chứng minh độ an toàn đạt được cũng như các hoạt động bảo trì trong tương lai.

7.2 Các yêu cầu phần mềm

7.2.1 Mục tiêu

7.2.1.1 Để mô tả tập hợp hoàn chỉnh các yêu cầu đối với phần mềm đáp ứng tất cả các yêu cầu hệ thống an toàn và đưa ra tập hợp hoàn chỉnh các loại tài liệu cho từng giai đoạn tiếp theo.

7.2.1.2 Để mô tả Chỉ dẫn kỹ thuật kiểm thử tổng thể phần mềm.

7.2.2 Các tài liệu đầu vào

1) Chỉ dẫn các yêu cầu hệ thống

2) Chỉ dẫn các yêu cầu về an toàn hệ thống

3) Mô tả cấu trúc hệ thống

4) Chỉ dẫn kỹ thuật giao diện với bên ngoài (ví dụ: Chỉ dẫn kỹ thuật giao diện phần mềm / phần mềm, chỉ dẫn kỹ thuật giao diện phần mềm / phần cứng)

5) Kế hoạch đảm bảo chất lượng phần mềm

6) Kế hoạch thẩm định phần mềm

7.2.3 Các tài liệu đầu ra

1) Chỉ dẫn các yêu cầu phần mềm

2) Chỉ dẫn kiểm thử tổng thể phần mềm

3) Báo cáo thẩm tra các yêu cầu phần mềm

7.2.4 Các yêu cầu

7.2.4.1 Chỉ dẫn các yêu cầu phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị quản lý các yêu cầu, trên cơ sở các tài liệu đầu vào từ mục 7.2.2.

Các yêu cầu từ mục 7.2.4.2 đến 7.2.4.15 tham chiếu đến Chỉ dẫn các yêu cầu phần mềm.

7.2.4.2 Chỉ dẫn các yêu cầu phần mềm phải thể hiện các đặc tính cần thiết của phần mềm đang được phát triển. Những đặc tính này (tất cả đều đã được xác định trong ISO/IEC 9126 (ngoại trừ các đặc tính an toàn)), phải bao gồm:

a) Chức năng (bao gồm khả năng và sự hiệu năng theo thời gian phản hồi).

b) Độ chắc chắn và khả năng bảo trì.

c) Độ an toàn (bao gồm các chức năng về an toàn và các mức độ toàn vẹn về an toàn phần mềm liên quan).

d) Tính hiệu quả.

e) Khả năng sử dụng.

f) Khả năng thay đổi.

7.2.4.3 Mức toàn vẹn về an toàn phần mềm phải được đưa ra như được quy định trong Điều 4 và được ghi lại trong Chỉ dẫn các yêu cầu phần mềm.

7.2.4.4 Theo mức toàn vẹn về an toàn phần mềm được yêu cầu, Chỉ dẫn các yêu cầu phần mềm phải được thể hiện và được cấu trúc sao cho:

a) Hoàn chỉnh, chính xác rõ ràng, không mập mờ, có thể thẩm tra, có thể kiểm thử, có thể duy tu bảo dưỡng và khả thi.

b) Có thể theo dõi theo vết theo tất cả các tài liệu đầu vào.

7.2.4.5 Chỉ dẫn các yêu cầu phần mềm phải bao gồm các dạng thể hiện và mô tả để các cá nhân chịu trách nhiệm liên quan trong toàn bộ vòng đời hệ thống có thể hiểu được.

7.2.4.6 Chỉ dẫn các yêu cầu phần mềm phải xác định và ghi lại tất cả các giao diện với các hệ thống khác, bên trong hoặc bên ngoài thiết bị được điều khiển, bao gồm người vận hành, bất kì khi nào có sự tồn tại liên kết trực tiếp hoặc được lập kế hoạch.

7.2.4.7 Tất cả các chế độ hoạt động liên quan phải được đưa ra chi tiết trong Chỉ dẫn các yêu cầu phần mềm.

7.2.4.8 Tất cả các chế độ hoạt động liên quan của các thiết bị điện tử lập trình, đặc biệt ở chế độ vận hành hư hỏng, phải được nêu chi tiết trong Chỉ dẫn các yêu cầu phần mềm.

7.2.4.9 Mọi ràng buộc giữa phần cứng và phần mềm phải được xác định và được ghi lại trong Chỉ dẫn các yêu cầu phần mềm.

7.2.4.10 Chỉ dẫn các yêu cầu phần mềm phải xem xét việc tự kiểm tra phần mềm và kiểm tra phần cứng bằng phần mềm theo mức độ mô tả được yêu cầu của việc ghi chép hệ thống. Việc tự kiểm tra phần mềm bao gồm cả việc phát hiện và báo lỗi hư hỏng và lỗi của chính phần mềm.

7.2.4.11 Chỉ dẫn các yêu cầu phần mềm phải bao gồm các yêu cầu đối với việc kiểm thử định kỳ các chức năng theo mức độ do Chỉ dẫn các yêu cầu an toàn hệ thống yêu cầu.

7.2.4.12 Chỉ dẫn các yêu cầu phần mềm phải bao gồm các yêu cầu cho phép kiểm thử được tất cả các chức năng về an toàn trong suốt quá trình vận hành tổng thể hệ thống, theo mức độ do Chỉ dẫn các yêu cầu an toàn hệ thống yêu cầu.

7.2.4.13 Tất cả các chức năng được phần mềm thực hiện, đặc biệt là các chức năng liên quan tới việc đạt được mức toàn vẹn về an toàn hệ thống yêu cầu, thì những chức năng này phải được xác định rõ ràng trong Chỉ dẫn các yêu cầu phần mềm.

7.2.4.14 Khi phần mềm được yêu cầu thực hiện các chức năng không liên quan tới an toàn thì những chức năng này phải được xác định rõ ràng trong Chỉ dẫn các yêu cầu phần mềm.

7.2.4.15 Chỉ dẫn các yêu cầu phần mềm phải được hỗ trợ bằng các kỹ thuật và các biện pháp trong Bảng A.2. Phải làm rõ các kết hợp được lựa chọn thỏa mãn mục 4.8 và 4.9.

7.2.4.16 Chỉ dẫn kiểm thử tổng thể phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị kiểm thử, trên cơ sở Chỉ dẫn các yêu cầu phần mềm.

Các yêu cầu từ mục 7.2.4.17 đến 7.2.4.19 tham chiếu đến Chỉ dẫn kỹ thuật kiểm thử tổng thể phần mềm.

7.2.4.17 Chỉ dẫn kỹ thuật kiểm thử tổng thể phần mềm phải mô tả các kiểm thử được tiến hành trên phần mềm hoàn chỉnh.

7.2.4.18 Chỉ dẫn kỹ thuật kiểm thử tổng thể phần mềm phải được hỗ trợ bằng các kỹ thuật và các biện pháp trong Bảng A.7. Phải làm rõ các kết hợp được lựa chọn thỏa mãn mục 4.8 và 4.9.

7.2.4.19 Chỉ dẫn kiểm thử tổng thể phần mềm phải xác định rõ các trường hợp kiểm thử cho từng chức năng được yêu cầu, bao gồm:

a) Các tín hiệu đầu vào được yêu cầu với các dải trị số và các giá trị của chúng.

b) Các tín hiệu đầu ra được dự báo với các dải trị số và các giá trị của chúng.

c) Các chỉ tiêu kiểm thử thành công, bao gồm hai mặt hiệu năng và chất lượng.

7.2.4.20 Báo cáo thẩm tra các yêu cầu phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở của Chỉ dẫn các yêu cầu an toàn hệ thống, Chỉ dẫn các yêu cầu phần mềm, Chỉ dẫn kiểm thử tổng thể phần mềm và Kế hoạch đảm bảo chất lượng phần mềm.

Các yêu cầu từ mục 7.2.4.21 đến 7.2.4.22 tham chiếu đến Báo cáo thẩm tra các yêu cầu phần mềm.

7.2.4.21 Báo cáo thẩm tra các yêu cầu phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập cho tất cả các Báo cáo thẩm tra (xem 6.2.4.13).

7.2.4.22 Khi đã thiết lập được Chỉ dẫn các yêu cầu phần mềm, việc thẩm tra phải đề cập tới:

a) Chỉ dẫn các yêu cầu phần mềm đáp ứng đầy đủ của các yêu cầu được đưa ra trong Chỉ dẫn các yêu cầu hệ thống, Chỉ dẫn các yêu cầu an toàn hệ thống và Kế hoạch đảm bảo chất lượng phần mềm.

b) Chỉ dẫn các yêu cầu phần mềm đáp ứng được các yêu cầu chung về khả năng sẵn sàng và khả năng theo dõi theo vết trong các mục từ 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể từ mục 7.2.4.2 đến 7.2.4.15.

c) Sự đầy đủ của Chỉ dẫn kỹ thuật kiểm thử tổng thể phần mềm như là Chỉ dẫn các yêu cầu phần mềm.

d) Xác định mọi hoạt động bổ sung để chứng minh phạm vi chính xác của các yêu cầu không thể kiểm thử được.

e) Sự thống nhất nội bộ trong Chỉ dẫn các yêu cầu phần mềm.

f) Chỉ dẫn các yêu cầu phần mềm đáp ứng đầy đủ hoặc tính tới các ràng buộc giữa phần cứng và phần mềm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra các yêu cầu phần mềm.

7.3 Cấu trúc và thiết kế

7.3.1 Mục tiêu

7.3.1.1 Để xây dựng cấu trúc phần mềm đạt được các yêu cầu của phần mềm.

7.3.1.2 Để xác định và đánh giá mức độ tương tác phần cứng / phần mềm đối với an toàn.

7.3.1.3 Để lựa chọn phương pháp thiết kế nếu trước đó chưa xác định được.

7.3.1.4 Để thiết kế phần mềm có mức toàn vẹn về an toàn phần mềm xác định từ các tài liệu đầu vào.

7.3.1.5 Để đảm bảo hệ thống tạo ra và phần mềm của nó có thể sẵn sàng kiểm thử từ ban đầu. Khi việc thẩm tra và kiểm thử là công việc chính trong quá trình thẩm định, phải xem xét cụ thể các yêu cầu thẩm tra và kiểm thử trong quá trình hoạt động.

7.3.2 Tài liệu đầu vào

1) Chỉ dẫn các yêu cầu phần mềm.

7.3.3 Tài liệu đầu ra

1) Chỉ dẫn cấu trúc phần mềm.
2) Chỉ dẫn thiết kế phần mềm
3) Chỉ dẫn giao diện phần mềm
4) Chỉ dẫn kiểm thử tích hợp phần mềm
5) Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng
6) Báo cáo thẩm tra cấu trúc và thiết kế phần mềm

7.3.4 Các yêu cầu

7.3.4.1 Chỉ dẫn cấu trúc phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thiết kế, trên cơ sở của Chỉ dẫn các yêu cầu phần mềm.

Các yêu cầu từ mục 7.3.4.2 đến 7.3.4.14 tham chiếu Chỉ dẫn cấu trúc phần mềm.

7.3.4.2 Phải thiết lập và nêu chi tiết Cấu trúc phần mềm được đề xuất trong Chỉ dẫn cấu trúc phần mềm.

7.3.4.3 Chỉ dẫn cấu trúc phần mềm phải xem xét tính khả thi của việc đạt được Chỉ dẫn các yêu cầu phần mềm theo mức toàn vẹn về an toàn phần mềm yêu cầu.

Chú thích: Cấu trúc phần mềm phải giảm thiểu tối đa mức độ và tính phức tạp của phần ứng dụng về an toàn.

7.3.4.4 Chỉ dẫn cấu trúc phần mềm phải xác định, phân tích và nêu chi tiết mức độ của tất cả các tương tác phần cứng/phần mềm.

7.3.4.5 Chỉ dẫn cấu trúc phần mềm phải xác định rõ tất cả các thành phần của phần mềm và đối với những thành phần này phải chỉ rõ:

a) Liệu những thành phần này là mới hay là cũ.

b) Liệu những thành phần này đã được thẩm định trước đó chưa và các điều kiện thẩm định.

c) Mức toàn vẹn về an toàn phần mềm của thành phần.

7.3.4.6 Các thành phần của phần mềm phải

a) Bao trùm một tập hợp xác định các yêu cầu phần mềm.

b) Được xác định rõ ràng và có các định dạng độc lập bên trong hệ thống quản lý cấu hình.

7.3.4.7 Việc sử dụng phần mềm thương mại phổ biến phải xem xét đến những hạn chế dưới đây:

a) Đối với tất cả các mức toàn vẹn về an toàn phần mềm, phải xác định và ghi lại rõ ràng các thông tin dưới đây:

- Các yêu cầu mà phần mêm hiện có trước đó dự định đáp ứng;

- Các giả thiết về môi trường hoạt động của phần mềm trước đó;

- Các giao diện với các bộ phận khác trong phần mềm.

b) Đối với tất cả các mức toàn vẹn về an toàn phần mềm, quá trình thẩm định toàn bộ phần mềm đã thực hiện thẩm định phần mềm đã có trước đó.

c) Đối với các mức toàn vẹn an toàn phần mềm SIL 3 hoặc SIL 4, phải chú ý những vấn đề sau:

- Phải tiến hành phân tích các sự cố hư hỏng có thể của phần mềm hiện có trước đó và các hậu quả của những hư hỏng này;

- Phải xác định chiến lược phát hiện các hư hỏng của phần mềm hiện có trước đó và bảo vệ hệ thống không bị những hư hỏng này;

- Quá trình thẩm tra và thẩm định phải đảm bảo:

1) Phần mềm hiện có trước đó đáp ứng được các yêu cầu đã được phân bổ.

2) Các hư hỏng của phần mềm hiện có trước đó phải được phát hiện và hệ thống tích hợp phần mềm hiện có trước đó được bảo vệ không bị những hư hỏng này.

3) Đáp ứng được các giả thiết về môi trường hoạt động của phần mềm hiện có trước đó.

d) Phần mềm hiện có trước đó phải đính kèm bản mô tả chính xác đầy đủ (ví dụ: bị giới hạn theo các chức năng được sử dụng) và hoàn chỉnh (ví dụ: các chức năng, các ràng buộc và bằng chứng). Bản mô tả phải có các ràng buộc đối với phần cứng và/hoặc phần mềm để Đơn vị tích hợp nhận thức được và xem xét trong quá trình ứng dụng. Cụ thể bản mô tả này sẽ thiết lập phương tiện để đưa thông tin cho Đơn vị tích hợp về phần mềm được xây dựng, các đặc tính, sự hoạt động kỹ thuật của nó.

Chú thích: Có thể sử dụng bằng chứng mang tính thống kê trong Kế hoạch thẩm định phần mềm hiện có trước đó.

7.3.4.8 Ưu tiên sử dụng các thành phần phần mềm đã được thẩm tra hiện có được xây dựng theo tiêu chuẩn này trong quá trình thiết kế bất cứ khi nào có thể.

7.3.4.9 Nếu phần mềm bao gồm các thành phần có các mức toàn vẹn về an toàn phần mềm khác nhau thì tất cả các thành phần phần mềm phải được xử lý như đối với các mức cao nhất, trừ khi có bằng chứng về sự độc lập giữa các thành phần có mức toàn vẹn về an toàn phần mềm cao hơn và các thành phần có mức toàn vẹn về an toàn phần mềm thấp hơn. Phải ghi lại bằng chứng trong Chỉ dẫn cấu trúc phần mềm.

7.3.4.10 Chỉ dẫn cấu trúc phần mềm phải xác định rõ chiến lược phát triển phần mềm theo mức độ toàn vẹn về an toàn được yêu cầu. Chỉ dẫn cấu trúc phần mềm phải được thể hiện và cấu trúc theo cách:

a) Hoàn chỉnh, chính xác, rõ ràng, không mập mờ, có thể thẩm tra, có thể kiểm thử, có thể duy trì và khả thi.

b) Có thể truy vết theo Chỉ dẫn các yêu cầu phần mềm.

7.3.4.11 Các biện pháp xử lý các sự cố phải có trong Chỉ dẫn cấu trúc phần mềm để đạt được sự cân bằng giữa việc các chiến lược tránh sự cố và các chiến lược xử lý sự cố.

7.3.4.12 Chỉ dẫn cấu trúc phần mềm phải kết luận rằng các kỹ thuật, biện pháp và các chương trình được lựa chọn tạo thành một nhóm thỏa mãn Chỉ dẫn các yêu cầu phần mềm theo mức toàn vẹn về an toàn phần mềm yêu cầu.

7.3.4.13 Chỉ dẫn cấu trúc phần mềm phải tính tới các yêu cầu từ mục 8.4.8 khi phần mềm được cấu hình bằng dữ liệu hoặc các thuật toán ứng dụng.

7.3.4.14 Chỉ dẫn cấu trúc phần mềm phải lựa chọn các kỹ thuật và các biện pháp ở Bảng A.3, Phải kết luận sự kết hợp được lựa chọn thỏa mãn mục 4.8 và 4.9.

7.3.4.15 Phải cân bằng giữa quy mô và mức độ phức tạp của cấu trúc phần mềm được phát triển.

7.3.4.16 Có thể sử dụng cách lập mẫu trong mọi giai đoạn để tìm ra các yêu cầu hoặc có được cái nhìn chi tiết hơn về các yêu cầu và các hệ quả của nó.

7.3.4.17 Chỉ có thể sử dụng các đoạn mã trong mẫu ở hệ thống mục tiêu nếu chứng minh được việc mã hóa và quá trình phát triển và ghi chép của nó đáp ứng tiêu chuẩn này.

7.3.4.18 Chỉ dẫn giao diện phần mềm đối với tất cả các giao diện giữa các thành phần của phần mềm và giới hạn của phần mềm tổng thể phải được lập thành văn bản, do trách nhiệm của Đơn vị thiết kế, trên cơ sở của Chỉ dẫn các yêu cầu phần mềm và Chỉ dẫn cấu trúc phần mềm.

Các yêu cầu trong mục 7.3.4.19 tham chiếu tới Chỉ dẫn giao diện phần mềm.

7.3.4.19 Việc mô tả các giao diện phải đề cập tới:

a) Các điều kiện trước / sau.

b) Xác định và mô tả tất cả các giá trị giới hạn biên cho tất cả các dữ liệu được quy định.

c) Sự hoạt động khi vượt quá giá trị giới hạn biên.

d) Sự hoạt động khi đạt giá trị giới hạn biên.

e) Đối với các dữ liệu đầu vào và đầu ra chủ chốt theo thời gian:

1) Các ràng buộc về thời gian và các yêu cầu để hoạt động chính xác.

2) Quản lý các tình huống bất thường.

f) Bộ nhớ được phân bổ cho các vùng đệm giao diện và các cơ chế để phát hiện bộ nhớ không thể được phân bổ thêm hoặc tất cả các vùng đệm đã bị đầy, nếu có thể áp dụng.

g) Sự tồn tại các cơ chế đồng bộ hóa giữa các chức năng (xem mục e).

Phải xác định tất cả các dữ liệu đến và đi ra khỏi các giao diện cho toàn bộ dải giá trị được xác định bằng loại dữ liệu, bao gồm các dải không được sử dụng khi các chức năng xử lý:

a) Các bản xác định và mô tả tất cả các cấp độ tương đương đối với tất cả các dữ liệu được quy định và từng chức năng sử dụng các bản này.

b) Việc xác định các cấp độ tương đương không được sử dụng hoặc bị cấm.

Chú thích: Loại dữ liệu bao gồm:

1) Các thông số đầu vào và các kết quả đầu ra của các chức năng và/hoặc các quy trình.

2) Dữ liệu được quy định ở dạng sóng vô tuyến hoặc các nhóm liên lạc.

3) Dữ liệu của phần cứng.

7.3.4.20 Chỉ dẫn thiết kế phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thiết kế, trên cơ sở của Chỉ dẫn các yêu cầu phần mềm, Chỉ dẫn cấu trúc phần mềm và Chỉ dẫn giao diện phần mềm.

Các yêu cầu từ mục 7.3.4.21 đến 7.3.4.24 tham chiếu Chỉ dẫn thiết kế phần mềm.

7.3.4.21 Phải có sẵn các tài liệu đầu vào trước khi bắt đầu quá trình thiết kế, mặc dù không cần thiết phải tổng kết lại.

7.3.4.22 Chỉ dẫn thiết kế phần mềm phải mô tả thiết kế phần mềm dựa trên việc phân chia thành các thành phần, mỗi thành phần có Chỉ dẫn thiết kế thành phần phần mềm và Chỉ dẫn kiểm thử thành phần phần mềm riêng.

7.3.4.23 Chỉ dẫn thiết kế phần mềm phải đề cập tới:

a) Các thành phần phần mềm được theo dõi theo vết ngược lại cấu trúc phần mềm và mức toàn vẹn về an toàn của chúng.

b) Các giao diện của các thành phần phần mềm với môi trường.

c) Các giao diện giữa các thành phần phần mềm.

d) Các cấu trúc dữ liệu.

e) Việc phân bổ và truy vết các yêu cầu của các thành phần,

f) Các thuật toán chính và quá trình lập chuỗi,

g) Các cơ chế báo cáo lỗi.

7.3.4.24 Chỉ dẫn thiết kế phần mềm phải lựa chọn các kỹ thuật và các biện pháp từ Bảng A.4. Phải chứng minh sự kết hợp được lựa chọn thỏa mãn mục 4.8 và 4.9.

7.3.4.25 Các tiêu chuẩn mã hóa phải được phát triển và quy định:

a) Việc ứng dụng lập trình tốt, như được quy định trong Bảng A.12,

b) Các biện pháp để tránh hoặc phát hiện các lỗi có thể phát sinh trong quá trình ứng dụng ngôn ngữ và không thể phát hiện được trong quá trình thẩm tra (xem 7.5 và 7.6). Các hư hỏng này sẽ được tìm ra bằng việc phân tích tất cả các đặc tính của ngôn ngữ.

c) Các quy trình để ghi lại mã nguồn.

7.3.4.26 Việc lựa chọn tiêu chuẩn mã hóa phải được làm rõ theo mức độ của mức toàn vẹn về an toàn phần mềm.

7.3.4.27 Phải sử dụng các tiêu chuẩn mã hóa để phát triển tất cả các phần mềm và tham chiếu các tiêu chuẩn này trong Kế hoạch đảm bảo chất lượng phần mềm.

7.3.4.28 Để phù hợp với mức toàn vẹn về an toàn phần mềm yêu cầu, phương pháp thiết kế được lựa chọn phải có các đặc tính hỗ trợ:

a) Nền tảng, tính modul hóa và các đặc tính khác kiểm soát mức độ phức tạp.

b) Sự thể hiện rõ ràng và chính xác của:

1) Chức năng.

2) Luồng thông tin giữa các thành phần.

3) Quá trình tạo chuỗi và thông tin liên quan theo thời gian.

4) Các quá trình hoạt động đồng thời.

5) Cấu trúc dữ liệu và các tính năng.

c) Sự nhận thức của con người.

d) Quá trình thẩm tra và thẩm định.

e) Quá trình bảo trì phần mềm.

7.3.4.29 Chỉ dẫn kiểm thử tích hợp phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị tích hợp, trên cơ sở của Chỉ dẫn các yêu cầu phần mềm, Chỉ dẫn cấu trúc phần mềm, Chỉ dẫn thiết kế phần mềm và Chỉ dẫn giao diện phần mềm.

Các yêu cầu từ mục 7.3.4.30 đến 7.3.4.32 tham chiếu Chỉ dẫn kiểm thử tích hợp phần mềm.

7.3.4.30 Chỉ dẫn kiểm thử tích hợp phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Chỉ dẫn kỹ thuật kiểm thử (xem 6.1.4.4).

7.3.4.31 Chỉ dẫn kiểm thử tích hợp phần mềm phải đề cập tới các vấn đề sau:

a) Phải thể hiện cho thấy từng thành phần phần mềm đưa ra các giao diện được quy định cho các thành phần khác bằng cách chạy các thành phần cùng với nhau.

b) Phải thể hiện cho thấy phần mềm hoạt động theo một cách thức phù hợp khi các giao diện bị phụ thuộc vào các đầu vào không nằm trong chỉ dẫn kỹ thuật.

c) Dữ liệu đầu vào cần thiết của chuỗi và các giá trị của nó phải là cơ sở của các trường hợp kiểm thử.

d) Dữ liệu đầu ra được dự báo cùng với các kết quả và các giá trị của chung phải là căn cứ của các trường hợp kiểm thử.

e) Phải thể hiện cho thấy các kết quả của việc kiểm thử tổng thành (xem 7.5.4.5 và 7.5.4.7) sẽ được sử dụng lại để kiểm thử tích hợp phần mềm.

7.3.4.32 Chỉ dẫn kiểm thử tích hợp phần mềm phải lựa chọn các kỹ thuật và các biện pháp trong Bảng A.5. Phải chứng minh sự kết hợp được lựa chọn thỏa mãn mục 4.8 và 4.9.

7.3.4.33 Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng phải được lập thành văn bản, do trách nhiệm của Đơn vị tích hợp, trên cơ sở Bản mô tả thiết kế hệ thống, Chỉ dẫn các yêu cầu phần mềm, Chỉ dẫn cấu trúc phần mềm và Chỉ dẫn thiết kế phần mềm.

Các yêu cầu từ mục 7.3.4.34 đến 7.3.4.39 tham chiếu Chỉ dẫn kiểm thử tích hợp phần mềm/phần cứng.

7.3.4.34 Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng nên sớm được lập ra trong vòng đời phát triển, để việc kiểm thử tích hợp có thể được hướng dẫn đúng và để việc thiết kế chi tiết hoặc các yêu cầu tích hợp khác có thể được đưa ra phù hợp. Phụ thuộc vào quy mô hệ thống, chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng có thể được phân chia nhỏ trong quá trình phát triển ra một số loại tài liệu con và được bổ sung lẫn nhau, khi các thiết kế phần cứng và phần mềm được phát triển và các yêu cầu tích hợp chi tiết trở nên rõ ràng hơn.

7.3.4.35 Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng phải phân biệt rõ giữa các hoạt động có thể được nhà cung cấp thực hiện theo đề nghị của họ và các hoạt động đó yêu cầu có sự tiếp cận đến lĩnh vực của người sử dụng.

7.3.4.36 Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng phải đề cập tới các vấn đề sau:

a) Phải thể hiện cho thấy phần mềm hoạt động đúng trên phần cứng, sử dụng phần cứng theo các giao diện phần cứng được quy định.

b) Phải thể hiện cho thấy phần mềm có thể xử lý các sự cố phần cứng theo yêu cầu.

c) Phải chứng minh được thời gian và hiệu năng cần thiết.

d) Dữ liệu đầu vào cần thiết cùng với các kết quả và các giá trị phải là cơ sở của các trường hợp kiểm thử.

e) Dữ liệu đầu ra được dự báo với các kết quả và các giá trị phải là cơ sở của các trường hợp kiểm thử.

f) Phải thể hiện cho thấy các kết quả của việc kiểm thử thành phần (xem 7.5.4.5) và kiểm thử tích hợp phần mềm (xem 7.6.4.3) sẽ được sử dụng lại trong kiểm thử tích hợp phần mềm / phần cứng.

7.3.4.37 Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng phải ghi lại các vấn đề sau:

a) Các trường hợp kiểm thử và dữ liệu kiểm thử.

b) Các loại hình kiểm thử được tiến hành.

c) Môi trường kiểm thử, bao gồm các chương trình, phần mềm hỗ trợ và bản mô tả cấu hình.

d) Chỉ tiêu hoàn thành kiểm thử.

7.3.4.38 Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Chỉ dẫn kỹ thuật kiểm thử (xem 6.1.4.4).

7.3.4.39 Chỉ dẫn kỹ thuật kiểm thử tích hợp phần mềm / phần cứng phải lựa chọn các kỹ thuật và các biện pháp trong Bảng A.5. Việc lựa chọn kết hợp phải được chứng minh thỏa mãn mục 4.8 và 4.9.

7.3.4.40 Báo cáo thẩm tra cấu trúc và thiết kế phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở Chỉ dẫn các yêu cầu phần mềm, Chỉ dẫn cấu trúc phần mềm, Chỉ dẫn thiết kế phần mềm, Chỉ dẫn kiểm thử tích hợp phần mềm và Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng.

Các yêu cầu từ mục 7.3.4.41 đến 7.3.4.43 tham chiếu tới Báo cáo thẩm tra cấu trúc và thiết kế phần mềm.

7.3.4.41 Chỉ dẫn thẩm tra thiết kế và cấu trúc phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Báo cáo thẩm tra (xem 6.2.4.13).

7.3.4.42 Sau khi lập xong Chỉ dẫn thiết kế, giao diện và cấu trúc phần mềm, việc thẩm tra phải đề cập đến:

a) Sự thống nhất nội bộ của Chỉ dẫn thiết kế, giao diện và cấu trúc phần mềm.

b) Sự đáp ứng đầy đủ Chỉ dẫn kỹ thuật các yêu cầu phần mềm về mặt thống nhất và hoàn chỉnh của Chỉ dẫn thiết kế, giao diện và cấu trúc phần mềm.

c) Chỉ dẫn cấu trúc phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.3.4.1 đến 7.3.4.14.

d) Chỉ dẫn giao diện phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.3.4.18 đến 7.3.4.19.

e) Chỉ dẫn cấu trúc phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.3.4.20 đến 7.3.4.24.

f) Sự xem xét đầy đủ các ràng buộc của Chỉ dẫn cấu trúc phần mềm và Chỉ dẫn thiết kế phần mềm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra cấu trúc và thiết kế phần mềm.

7.3.4.43 Sau khi lập xong Chỉ dẫn thiết kế, giao diện và cấu trúc phần mềm, việc thẩm tra phải đề cập đến:

a) Chỉ dẫn kiểm thử tích hợp phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.3.4.29 đến 7.3.4.32.

b) Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.3.4.33 đến 7.3.4.39.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra cấu trúc và thiết kế phần mềm.

7.4 Thiết kế thành phần

7.4.1 Mục tiêu

7.4.1.1 Để xây dựng thiết kế thành phần phần mềm đạt được các yêu cầu của Chỉ dẫn thiết kế phần mềm theo mức toàn vẹn về an toàn phần mềm được yêu cầu.

7.4.1.2 Để xây dựng chỉ dẫn kiểm thử thành phần phần mềm đạt được các yêu cầu của Chỉ dẫn thiết kế thành phần phần mềm theo mức toàn vẹn về an toàn phần mềm được yêu cầu.

7.4.2 Các tài liệu đầu vào

1) Chỉ dẫn thiết kế phần mềm.

7.4.3 Các tài liệu đầu ra

1) Chỉ dẫn thiết kế thành phần phần mềm

2) Chỉ dẫn kiểm thử thành phần phần mềm

3) Báo cáo thẩm tra thiết kế thành phần phần mềm

7.4.4 Các yêu cầu

7.4.4.1 Đối với từng tổng thành, Chỉ dẫn thiết kế thành phần phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thiết kế, trên cơ sở của Chỉ dẫn thiết kế phần mềm.

Các yêu cầu từ mục 7.4.4.2 đến 7.4.4.6 tham chiếu Chỉ dẫn thiết kế thành phần phần mềm.

7.4.4.2 Đối với từng thành phần phần mềm, phải có các thông tin dưới đây:

- Tác giả;
- Lịch sử cấu hình;
- Bản mô tả ngắn gọn.

Lịch sử cấu hình phải bao gồm việc nhận dạng chính xác các định dạng phiên bản hiện tại và tất cả các định dạng trước đó của thành phần phần mềm, quy định về phiên bản, ngày, tác giả và bản mô tả về các thay đổi được thực hiện so với phiên bản trước đó.

7.4.4.3 Chỉ dẫn thiết kế thành phần phần mềm phải đề cập tới:

a) Việc xác định tất cả các đơn vị thành phần phần mềm cấp thấp nhất (ví dụ: các đường dẫn, biện pháp, quy trình) được truy vết ngược về cấp độ cao hơn.

b) Các giao diện chi tiết của chúng với môi trường và các thành phần khác với các đầu vào và đầu ra chi tiết.

c) Mức toàn vẹn về an toàn của thành phần mà không được phân bổ thêm nữa trong nội bộ thành phần.

d) Thuật toán chi tiết và cấu trúc dữ liệu.

Từng Chỉ dẫn cấu trúc thành phần phần mềm phải tự thống nhất và cho phép chuyển đổi thành đoạn mã của thành phần phần mềm tương ứng.

7.4.4.4 Mỗi chỉ dẫn kỹ thuật thiết kế thành phần phần mềm phải có thể đọc được, hiểu được và có thể kiểm thử.

7.4.4.5 Quy mô và mức độ phức tạp của từng thành phần phần mềm được phát triển phải cân bằng với nhau.

7.4.4.6 Chỉ dẫn thiết kế thành phần phần mềm phải lựa chọn các kỹ thuật và các biện pháp trong Bảng A.4. Phải chứng minh sự kết hợp được lựa chọn là thỏa mãn mục 4.8 và 4.9.

7.4.4.7 Đối với từng thành phần, Chỉ dẫn kiểm thử thành phần phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị kiểm thử, trên cơ sở Chỉ dẫn thiết kế thành phần phần mềm.

Các yêu cầu từ mục 7.4.4.8 đến 7.4.4.10 tham chiếu tới Chỉ dẫn kiểm thử thành phần phần mềm.

7.4.4.8 Chỉ dẫn kiểm thử thành phần phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Chỉ dẫn kiểm thử (xem 6.1.4.4).

7.4.4.9 Phải kiểm thử thành phần theo chỉ dẫn kiểm thử thành phần phần mềm được lập ra. Những kiểm thử này phải thể hiện được từng thành phần thực hiện chức năng của nó. Chỉ dẫn kiểm thử thành phần phần mềm phải xác định và chứng minh chỉ tiêu cần thiết và mức độ kiểm thử được thực hiện theo mức toàn vẹn về an toàn được yêu cầu. Các kiểm thử này phải được thiết kế sao cho đáp ứng ba mục tiêu:

a) Để xác nhận thành phần thực hiện các chức năng dự định của nó (kiểm thử hộp đen).

b) Để kiểm tra cách thức tương tác của các bộ phận bên trong của thành phần để thực hiện các chức năng dự định của nó (kiểm thử hộp đen/trắng).

c) Để xác nhận tất cả các yếu tố của thành phần được kiểm thử (kiểm thử hộp trắng).

7.4.4.10 Chỉ dẫn kiểm thử thành phần phần mềm phải lựa chọn các kỹ thuật và các biện pháp trong Bảng A.5. Phải chứng minh kết hợp được lựa chọn là thỏa mãn mục 4.8 và 4.9.

7.4.4.11 Báo cáo thẩm tra thiết kế thành phần phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở Chỉ dẫn thiết kế phần mềm, Chỉ dẫn thiết kế thành phần phần mềm, Chỉ dẫn kiểm thử thành phần phần mềm.

Các yêu cầu từ mục 7.4.4.12 đến 7.4.4.13 tham chiếu tới Báo cáo thẩm tra thiết kế thành phần phần mềm.

7.4.4.12 Báo cáo thẩm tra thiết kế phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Báo cáo thẩm tra (xem 6.2.4.13).

7.4.4.13 Sau khi lập xong Chỉ dẫn thiết kế thành phần phần mềm, việc thẩm tra phải xem xét:

a) Sự đáp ứng đầy đủ của Chỉ dẫn thiết kế thành phần phần mềm với Chỉ dẫn thiết kế phần mềm.

b) Chỉ dẫn thiết kế thành phần phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.4.4.1 đến 7.4.4.6.

c) Sự đáp ứng đầy đủ Chỉ dẫn thiết kế thành phần phần mềm của Chỉ dẫn kiểm thử thành phần phần mềm.

d) Chỉ dẫn kiểm thử thành phần phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng theo dõi theo vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể từ mục 7.4.4.7 đến 7.4.4.10.

e) Việc phân chia Chỉ dẫn thiết kế phần mềm thành các thành phần phần mềm và Chỉ dẫn thiết kế phần mềm có xem xét tới:

1) Tính khả thi của hiệu năng yêu cầu.

2) Khả năng kiểm thử của các thẩm tra khác.

3) Khả năng bảo trì để cho phép các phát triển sau thêm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra thiết kế thành phần phần mềm.

7.5 Chạy và kiểm thử thành phần

7.5.1 Mục tiêu

7.5.1.1 Để tạo ra được phần mềm có thể phân tích, có thể kiểm thử, có thể thẩm tra và có thể bảo trì. Việc kiểm thử tổng thành cũng sẽ nằm trong giai đoạn này.

7.5.2 Các tài liệu đầu vào

1) Chỉ dẫn thiết kế thành phần phần mềm

2) Chỉ dẫn kiểm thử thành phần phần mềm

7.5.3 Các tài liệu đầu ra

1) Mã nguồn phần mềm và tài liệu hỗ trợ

2) Báo cáo kiểm thử thành phần phần mềm

3) Báo cáo thẩm tra mã nguồn phần mềm

7.5.4 Các yêu cầu

7.5.4.1 Mã nguồn phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thực hiện trên cơ sở của Chỉ dẫn thiết kế thành phần phần mềm. Các yêu cầu của từ mục 7.5.4.2 đến mục 7.5.4.4 tham chiếu tới mã nguồn phần mềm.

7.5.4.2 Phải cân bằng quy mô và độ phức tạp của mã nguồn được xây dựng.

7.5.4.3 Mã nguồn phần mềm phải có thể đọc được, có thể hiểu được và có thể kiểm thử được.

7.5.4.4 Mã nguồn phần mềm phải được kiểm soát cấu hình trước khi bắt đầu ghi lại quá trình kiểm thử.

7.5.4.5 Báo cáo kiểm thử thành phần phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị kiểm thử, trên cơ sở của Chỉ dẫn kiểm thử thành phần phần mềm và Mã nguồn phần mềm.

Các yêu cầu từ 7.5.4.6 tham chiếu tới Báo cáo kiểm thử thành phần phần mềm.

7.5.4.6 Báo cáo kiểm thử thành phần phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Báo cáo kiểm thử (xem 6.1.4.5).

7.5.4.7 Báo cáo kiểm thử thành phần phần mềm phải có các đặc điểm sau:

a) Kết luận về các kết quả kiểm thử và liệu mỗi thành phần có đáp ứng được các yêu cầu của Chỉ dẫn thiết kế thành phần phần mềm.

b) Kết luận về mức độ kiểm thử đưa ra cho từng thành phần, thể hiện được mức độ kiểm thử cần thiết đã đạt được cho tất cả các chỉ tiêu yêu cầu.

7.5.4.8 Báo cáo thẩm tra mã nguồn phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở của Chỉ dẫn thiết kế thành phần phần mềm, Chỉ dẫn kiểm thử thành phần phần mềm và Mã nguồn phần mềm.

Các yêu cầu từ mục 7.5.4.9 đến 7.5.4.10 tham chiếu tới Báo cáo thẩm tra mã nguồn phần mềm.

7.5.4.9 Báo cáo thẩm tra mã nguồn phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Báo cáo thẩm tra (xem 6.2.4.13).

7.5.4.10 Sau khi lập xong Báo cáo kiểm thử mã nguồn phần mềm và thành phần mềm, việc thẩm tra phải xem xét:

a) Sự hoạt động đầy đủ của Mã nguồn phần mềm theo Chỉ dẫn thiết kế thành phần phần mềm.

b) Việc sử dụng đúng các kỹ thuật và biện pháp được lựa chọn trong Bảng A.4 thỏa mãn mục 4.8 và 4.9.

c) Việc xác định quá trình ứng dụng đúng các tiêu chuẩn mã hóa.

d) Mã nguồn phần mềm đáp ứng các yêu cầu chung về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.5.4.1 đến 7.5.4.4.

e) Sự đầy đủ của Báo cáo kiểm thử thành phần phần mềm, ghi lại các kiểm thử được tiến hành phù hợp với Chỉ dẫn kiểm thử thành phần phần mềm.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra mã nguồn phần mềm.

7.6 Tích hợp

7.6.1 Mục tiêu

7.6.1.1 Để thực hiện việc tích hợp phần mềm và tích hợp phần mềm/phần cứng.

7.6.1.2 Để chứng minh phần mềm và phần cứng tương tác đúng với nhau để thực hiện các chức năng dự định.

7.6.2 Các tài liệu đầu vào

1) Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng

2) Chỉ dẫn kiểm thử tích hợp phần mềm

7.6.3 Các tài liệu đầu ra

1) Báo cáo kiểm thử tích hợp phần mềm

2) Báo cáo kiểm thử tích hợp phần mềm / phần cứng

3) Báo cáo thẩm tra tích hợp phần mềm.

7.6.4 Các yêu cầu

7.6.4.1 Việc tích hợp các thành phần phần mềm phải là quá trình kết hợp liên tục các thành phần độc lập và đã được kiểm thử trước đó thành một tập hợp sao cho các giao diện giữa các thành phần và phần mềm được kết hợp lại có thể được chứng minh đầy đủ trước khi tích hợp hệ thống và kiểm thử hệ thống.

7.6.4.2 Trong quá trình tích hợp phần mềm / phần cứng, mọi sự cải tiến hoặc thay đổi hệ thống được tích hợp phải được nghiên cứu tác động xác định rõ tất cả các thành phần bị tác động và các hoạt động thẩm tra lại cần thiết.

7.6.4.3 Báo cáo kiểm thử tích hợp phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị tích hợp, trên cơ sở của Chỉ dẫn kiểm thử tích hợp phần mềm.

Các yêu cầu từ mục 7.6.4.4 đến 7.6.4.6 tham chiếu Báo cáo kiểm thử tích hợp phần mềm.

7.6.4.4 Báo cáo kiểm thử tích hợp phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được quy định trong Báo cáo kiểm thử (xem 6.1.4.5).

7.6.4.5 Báo cáo kiểm thử tích hợp phải được lập như sau:

a) Báo cáo kiểm thử tích hợp phần mềm được lập phải nêu rõ các kết quả kiểm thử và liệu các mục tiêu và chỉ tiêu của Chỉ dẫn kiểm thử tích hợp phần mềm có được đáp ứng. Nếu có hư hỏng, phải ghi lại các trường hợp hư hỏng đó.

b) Phải ghi lại các trường hợp kiểm thử và các kết quả, ưu tiên theo dạng máy có thể đọc được để phân tích sau này.

c) Các kiểm thử phải có thể được lặp lại và có thể thực hiện bằng các phương pháp tự động nếu có thể thực hiện được.

d) Báo cáo kiểm thử tích hợp phần mềm phải ghi lại định dạng và cấu hình của tất cả các hạng mục có liên quan.

7.6.4.6 Báo cáo kiểm thử tích hợp phần mềm phải chứng minh việc sử dụng đúng các kỹ thuật và biện pháp được lựa chọn trong Bảng A.6 là thỏa mãn mục 4.8 và 4.9.

7.6.4.7 Báo cáo kiểm thử tích hợp phần mềm / phần cứng phải được lập thành văn bản, dưới trách nhiệm của Đơn vị tích hợp, trên cơ sở của Chỉ dẫn kiểm thử tích hợp phần mềm / phần cứng.

Các yêu cầu từ mục 7.6.4.8 đến 7.6.4.10 tham chiếu Báo cáo kiểm thử tích hợp phần mềm / phần cứng.

7.6.4.8 Báo cáo kiểm thử tích hợp phần mềm/phần cứng phải được lập thành văn bản phù hợp với các yêu cầu chung được quy định trong Báo cáo kiểm thử (xem 6.1.4.5).

7.6.4.9 Báo cáo kiểm thử tích hợp phải được lập như sau:

a) Báo cáo kiểm thử tích hợp phần mềm/phần cứng phải nêu rõ các kết quả kiểm thử và liệu các mục tiêu và chỉ tiêu của Chỉ dẫn kiểm thử tích hợp phần mềm/phần cứng có được đáp ứng. Nếu có hư hỏng, phải ghi lại các trường hợp hư hỏng đó.

b) Phải ghi lại các trường hợp kiểm thử và các kết quả, ưu tiên theo dạng máy có thể đọc được để phân tích sau này.

c) Các kiểm thử có thể lặp lại được và có thể thực hiện bằng các phương pháp tự động nếu khả thi.

d) Báo cáo kiểm thử tích hợp phần mềm/phần cứng phải ghi lại định dạng và cấu hình của tất cả các hạng mục có liên quan.

7.6.4.10 Báo cáo kiểm thử tích hợp phần mềm/phần cứng phải chứng minh việc sử dụng đúng các kỹ thuật và biện pháp được lựa chọn trong Bảng A.6 là thỏa mãn mục 4.8 và 4.9.

7.6.4.11 Báo cáo thẩm tra tích hợp phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở của Chỉ dẫn kiểm thử tích hợp phần mềm và phần mềm / phần cứng và các báo cáo kiểm thử tương ứng.

Các yêu cầu từ mục 7.6.4.12 đến 7.6.4.13 tham chiếu Báo cáo thẩm tra tích hợp phần mềm.

7.6.4.12 Báo cáo thẩm tra tích hợp phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Báo cáo thẩm tra (xem 6.2.4.13).

7.6.4.13 Sau khi lập xong Báo cáo kiểm thử tích hợp phần mềm và Báo cáo kiểm thử tích hợp phần mềm/phần cứng, việc thẩm tra phải xem xét:

a) Sự đầy đủ của Báo cáo kiểm thử tích hợp phần mềm, ghi lại các kiểm thử được tiến hành phù hợp với Chỉ dẫn kiểm thử tích hợp phần mềm.

b) Liệu Báo cáo kiểm thử tích hợp phần mềm có đáp ứng các yêu cầu về khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.6.4.3 đến 7.6.4.6.

c) Sự đầy đủ của Báo cáo kiểm thử tích hợp phần mềm/phần cứng, ghi lại các kiểm thử được tiến hành phù hợp với Chỉ dẫn kiểm thử tích hợp phần mềm/phần cứng.

d) Liệu Báo cáo kiểm thử tích hợp phần mềm có đáp ứng các yêu cầu chung khả năng có thể đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.16 cũng như các yêu cầu cụ thể từ mục 7.6.4.7 đến 7.6.4.10.

7.7 Kiểm thử tổng thể phần mềm / Thẩm định lần cuối

7.7.1 Mục tiêu

7.7.1.1 Để phân tích và kiểm thử phần mềm và phần cứng được tích hợp đảm bảo thỏa mãn Chỉ dẫn các yêu cầu phần mềm, nhấn mạnh cụ thể vào các vấn đề chức năng và an toàn theo mức toàn vẹn về an toàn phần mềm và để kiểm tra xem liệu phần mềm có phù hợp với ứng dụng.

7.7.2 Tài liệu đầu vào

1) Chỉ dẫn các yêu cầu phần mềm
2) Chỉ dẫn kiểm thử tổng thể phần mềm
3) Kế hoạch thẩm tra phần mềm
4) Kế hoạch thẩm định phần mềm
5) Tất cả các tài liệu phần mềm và phần cứng, bao gồm các kết quả thẩm tra trung gian.
6) Chỉ dẫn các yêu cầu an toàn hệ thống

7.7.3 Tài liệu đầu ra

1) Báo cáo kiểm thử tổng thể phần mềm
2) Báo cáo thẩm định phần mềm
3) Thông tin lưu ý về phiên bản phần mềm sử dụng

7.7.4 Các yêu cầu

7.7.4.1 Báo cáo kiểm thử tổng thể phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị kiểm thử, trên cơ sở của Chỉ dẫn kiểm thử tổng thể phần mềm.

Các yêu cầu từ mục 7.7.4.2 đến 7.7.4.4 tham chiếu Báo cáo kiểm thử tổng thể phần mềm.

7.7.4.2 Báo cáo kiểm thử tổng thể phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Báo cáo kiểm thử (xem 6.1.4.5).

7.7.4.3 Đơn vị thẩm định phải quy định và tiến hành các kiểm thử bổ sung theo sự hướng dẫn của mình hoặc yêu cầu Đơn vị kiểm thử tiến hành. Các kiểm thử tổng thể phần mềm sẽ chủ yếu dựa trên cấu trúc của Chỉ dẫn các yêu cầu phần mềm, giá trị được bổ sung mà Đơn vị thẩm định đưa ra là các kiểm thử tạo áp lực lên hệ thống trong các trường hợp phức tạp thể hiện yêu cầu thực tế của người sử dụng.

7.7.4.4 Các kết quả của tất cả các kiểm thử và phân tích phải ghi lại trong Báo cáo kiểm thử tổng thể phần mềm.

7.7.4.5 Phần mềm phải được trải nghiệm bằng cách kết nối các đối tượng phần cứng thực tế hoặc hệ thống thực sẽ tương tác trong hoạt động, hoặc bằng mô phỏng các tín hiệu đầu vào và các đối tượng theo đầu ra. Phần mềm phải được kiểm tra dưới các điều kiện hiện có trong quá trình hoạt động bình thường, các tình huống xuất hiện đã được dự đoán trước và các điều kiện không mong muốn yêu cầu có các hoạt động phòng vệ của hệ thống. Khi sử dụng các đầu vào hoặc tải được mô phỏng, phải thể hiện cho thấy những đầu vào và tải này không khác đáng kể so với các đầu vào và tải gặp phải trong hoạt động thực tế.

7.7.4.6 Báo cáo thẩm định phần mềm phải được lập thành văn bản, do trách nhiệm của đơn vị thẩm định, trên cơ sở của Kế hoạch thẩm định phần mềm.

Các yêu cầu từ mục 7.7.4.7 đến 7.7.4.11 tham chiếu Báo cáo thẩm định phần mềm.

7.7.4.7 Báo cáo thẩm định tổng thể phần mềm phải được lập thành văn bản phù hợp với các yêu cầu chung được thiết lập trong Báo cáo thẩm định (xem 6.3.4.7 đến 6.3.4.11).

7.7.4.8 Khi hoàn thiện việc tích hợp và hoàn thiện việc kiểm thử và phân tích tổng thể phần mềm, phải lập Báo cáo thẩm định phần mềm như sau:

a) Phải tuyên bố liệu các mục tiêu và chỉ tiêu của Kế hoạch thẩm định phần mềm có được đáp ứng. Phải ghi lại và làm rõ các sai lệch.

b) Phải đưa ra được kết luận tổng kết về các kết quả kiểm thử và liệu toàn bộ phần mềm trên máy mục tiêu có đáp ứng được các yêu cầu đưa ra trong Chỉ dẫn các yêu cầu phần mềm.

c) Phải đưa ra đánh giá về mức độ kiểm thử theo các yêu cầu của Chỉ dẫn các yêu cầu phần mềm.

d) Phải thực hiện đánh giá các hoạt động thẩm tra khác phù hợp với Kế hoạch và Báo cáo thẩm tra phần mềm cùng với kiểm tra việc truy vết các yêu cầu có được thực hiện và đáp ứng đầy đủ.

e) Nếu Đơn vị thẩm định đưa ra các tình huống kiểm thử của mình, không thông báo cho Đơn vị kiểm thử thì trong Báo cáo thẩm định phần mềm phải ghi lại các tình huống này phù hợp với mục 6.3.4.7 đến 6.3.4.11.

7.7.4.9 Báo cáo thẩm định phần mềm phải có sự xác nhận về sự phù hợp theo mức toàn vẹn về an toàn phần mềm xác định của từng hoạt động kết hợp các kỹ thuật hoặc các biện pháp được lựa chọn theo Phụ lục A. Phải có đánh giá về hiệu quả tổng thể của việc kết hợp các kỹ thuật và biện pháp được thông qua, tính tới quy mô và mức độ phức tạp của phần mềm tạo ra và tính tới các kết quả hoạt động kiểm thử, thẩm tra và thẩm định thực tế.

7.7.4.10 Báo cáo thẩm định phần mềm phải đề cập đến các vấn đề sau:

a) Ghi lại nhận dạng và cấu hình của phần mềm.

b) Kết luận về việc xác định thiết bị và phần mềm hỗ trợ kỹ thuật phù hợp.

c) Kết luận về việc xác định các mô hình mô phỏng được sử dụng phù hợp.

d) Kết luận về sự phù hợp của Chỉ dẫn kiểm thử tổng thể phần mềm.

e) Thu thập và duy trì việc truy vết mọi sai lệch được phát hiện.

f) Rà soát và đánh giá mọi sai lệch về mặt rủi ro (tác động).

g) Kết luận dự án có được tiến hành phù hợp, xử lý các hoạt động sửa chữa phù hợp với quá trình và các quy trình quản lý các thay đổi và xác định rõ ràng mọi vấn đề khác biệt được phát hiện.

h) Kết luận về từng hạn chế được đưa ra do sự sai lệch theo cách có thể truy vết.

i) Kết luận liệu phần mềm có phù hợp với ứng dụng, tính tới các điều kiện và ràng buộc ứng dụng.

7.7.4.11 Phải xác định rõ ràng mọi sai lệch được phát hiện, bao gồm các lỗi tìm thấy và các vấn đề không phù hợp với tiêu chuẩn này hoặc với các yêu cầu phần mềm hoặc kế hoạch, cũng như các ràng buộc và giới hạn trong các mục con của Báo cáo thẩm định phần mềm, được đánh giá theo mưc toàn vẹn về an toàn và có trong Thông tin lưu ý về phiên bản phần mềm sử dụng đi kèm với phần mềm được chuyển giao.

7.7.4.12 Thông tin lưu ý về phiên bản phần mềm sử dụng đi kèm với phần mềm được chuyển giao phải có tất cả các hạn chế khi sử dụng phần mềm. Những hạn chế này được rút ra từ

a) Các lỗi tìm thấy.

b) Các vấn đề không phù hợp với tiêu chuẩn này.

c) Mức độ đáp ứng các yêu cầu.

d) Mức độ đáp ứng mọi kế hoạch.

8 Phát triển các thuật toán hoặc dữ liệu ứng dụng: các hệ thống được cấu hình bằng các thuật toán hoặc dữ liệu ứng dụng

8.1 Mục tiêu

8.1.1 Đặc điểm kỹ thuật của các hệ thống đường sắt là cần phải thiết kế từng hạng mục lắp đặt đáp ứng các yêu cầu riêng biệt cho một điều kiện khai thác cụ thể. Một hệ thống được cấu hình bằng các dữ liệu ứng dụng và/hoặc các thuật toán ứng dụng sẽ cho phép phần mềm chung đã được chứng nhận được thay đổi theo các yêu cầu độc lập cho từng ứng dụng cụ thể.

Mục tiêu của việc phát triển dữ liệu ứng dụng là tìm ra dữ liệu chuẩn từ các lắp đặt trước đó và kiểm tra sự hoạt động dự định bằng cách đánh giá quá trình phát triển được sử dụng cho dữ liệu ứng dụng.

Các yêu cầu phát triển thuật toán ứng dụng cũng sẽ giống như các yêu cầu khi phát triển phần mềm chung được mô tả trong các mục 1-7 và 9.

Một ví dụ điển hình là hệ thống có phần chung được cấu hình trước đó cho một ứng dụng đường sắt chung bằng một nhóm các thuật toán ứng dụng, sau đó được cấu hình bổ sung cho một lắp đặt cụ thể bằng cách cài đặt và kết nối các thuật toán ứng dụng và bằng một nhóm dữ liệu cấu hình. Ví dụ: các nguyên lý phát tín hiệu của hệ thống khóa lẫn (ví dụ: quản lý tín hiệu, quản lý điểm) có thể được thực hiện bằng một nhóm các thuật toán ứng dụng.

Dữ liệu ứng dụng chủ yếu sẽ lấy dưới dạng các giá trị thông số hoặc các mô tả (nhận dạng, kiểu loại, vị trí,…) của các đối tượng từ bên ngoài. Các thuật toán ứng dụng có thể lấy dưới dạng các sơ đồ khối chức năng, các sơ đồ trạng thái và các sơ đồ chuyển tiếp bậc thang, xác định sự phản hồi mong muốn của hệ thống theo các đầu vào, trạng thái hiện tại và các giá trị thông số cụ thể. Các thuật toán ứng dụng bao gồm các liên kết logic và các phép tính được thực hiện.

Các thuật toán/dự liệu ứng dụng thường được tạo ra bằng cách sử dụng các chương trình chuyên dụng. Dữ liệu này có thể được thể hiện theo dạng bảng hoặc dạng sơ đồ (có thể được diễn giải hoặc biên dịch thành các mã chạy chương trình sau khi đã được chuyển đổi thành mã nguồn được xử lý thông qua các ngôn ngữ lập trình chuyên dụng (với cú pháp và ngữ nghĩa)).

Việc thay đổi các hệ thống thông qua khả năng cấu hình sẽ giúp cho đơn vị thiết kế có được các mức độ kiểm soát khác nhau trên toàn bộ chức năng phần mềm chi tiết.

8.1.2 Các quy trình và các chương trình được sử dụng để phát triển phải phù hợp với mức toàn vẹn về an toàn hệ thống được xác định bằng chức năng được phát triển.

8.1.3 Các điều khoản dưới đây sẽ mô tả các yêu cầu để phát triển bước đầu các hệ thống có thể cấu hình và để phát triển từng nhóm các thuật toán / dữ liệu ứng dụng cụ thể sau này.

8.2 Tài liệu đầu vào

1) Chỉ dẫn các yêu cầu phần mềm của phần mềm chung

2) Chỉ dẫn cấu trúc phần mềm của phần mềm chung

3) Các điều kiện áp dụng của phần mềm chung và các chương trình ứng dụng

4) Hướng dẫn sử dụng của phần mềm chung và các chương trình ứng dụng

8.3 Tài liệu đầu ra

1) Kế hoạch chuẩn bị ứng dụng

2) Chỉ dẫn các yêu cầu ứng dụng

3) Cấu trúc và thiết kế ứng dụng

4) Chỉ dẫn kiểm thử ứng dụng

5) Báo cáo kiểm thử ứng dụng

6) Báo cáo thẩm tra chuẩn bị ứng dụng

7) Mã nguồn của các thuật toán / dữ liệu ứng dụng

8) Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng

8.4 Các yêu cầu

8.4.1 Quá trình phát triển ứng dụng

8.4.1.1 Kế hoạch chuẩn bị ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị quản lý các yêu cầu hoặc Đơn vị thiết kế, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.1.2 đến 8.4.1.11 tham chiếu tới Kế hoạch chuẩn bị ứng dụng.

8.4.1.2 Kế hoạch chuẩn bị ứng dụng phải được lập để xác định và nêu chi tiết quá trình phát triển ứng dụng, bao gồm tất cả các hoạt động, các sản phẩm chuyển giao và các vai trò thực hiện. Kế hoạch có thể được lập cho từng ứng dụng cụ thể hoặc cho một loại ứng dụng cụ thể, ví dụ: cho một ứng dụng chung.

8.4.1.3 Kế hoạch chuẩn bị ứng dụng phải xác định cấu trúc tài liệu của quá trình chuẩn bị ứng dụng.

8.4.1.4 Kế hoạch chuẩn bị ứng dụng phải lựa chọn các kỹ thuật và các biện pháp trong Bảng A.11. Phải chứng minh các kết hợp được lựa chọn là thỏa mãn mục 4.8 và 4.9.

8.4.1.5 Kế hoạch chuẩn bị ứng dụng phải quy định các quy trình và các chương trình ứng dụng (theo loại dựa trên mục 6.7) được sử dụng trong quá trình phát triển ứng dụng.

8.4.1.6 Kế hoạch chuẩn bị ứng dụng phải có các hoạt động thẩm tra và thẩm định để đảm bảo các thuật toán / dữ liệu ứng dụng là hoàn thiện, đúng và tương thích với các hoạt động khác và với ứng dụng chung, và để đưa ra bằng chứng về việc đáp ứng các điều kiện ứng dụng của ứng dụng chung. Các hoạt động thẩm tra, thẩm định và bằng chứng có thể được thay thế bằng việc thẩm tra và thẩm định được thực hiện trên các chương trình lập ra các thuật toán / dữ liệu ứng dụng. Các kết quả được thu thập lại với nhau trong Báo cáo thẩm tra chuẩn bị ứng dụng và Báo cáo kiểm thử ứng dụng.

8.4.1.7 Kế hoạch chuẩn bị ứng dụng phải có các hoạt động thẩm tra và thẩm định để đảm bảo các chương trình ứng dụng và phần mềm chung là tương thích với nhau và tương thích với ứng dụng cụ thể và để đưa ra bằng chứng về việc đáp ứng các điều kiện ứng dụng.

8.4.1.8 Phải tiến hành phân tích rủi ro trong quá trình phát triển ứng dụng, bao gồm các chương trình ứng dụng và các quy trình, để thẩm định kế hoạch chuẩn bị ứng dụng và để đáp ứng mức toàn vẹn về an toàn phần mềm yêu cầu. Kế hoạch chuẩn bị ứng dụng phải bao gồm cả việc phân tích rủi ro.

8.4.1.9 Kế hoạch chuẩn bị ứng dụng phải quy định các yêu cầu về sự độc lập giữa nhân viên tiến hành các nhiệm vụ thẩm tra, thẩm định và chuẩn bị theo 5.1.

Chú thích: Các hoạt động chuẩn bị dữ liệu được tiến hành bởi các đơn vị thiết kế ứng dụng.

8.4.1.10 Kế hoạch chuẩn bị ứng dụng phải xác định loại chương trình cho tất cả các chương trình phần cứng hoặc phần mềm được sử dụng trong vòng đời chuẩn bị ứng dụng.

8.4.1.11 Nếu có thể, Kế hoạch chuẩn bị ứng dụng phải đặt ra các ký hiệu để quy định các yêu cầu và thiết kế mà quen thuộc với các kỹ sư ứng dụng. Nếu đưa ra các ký hiệu mới, phải đưa ra các tài liệu cần thiết cho người sử dụng, cũng như các quá trình đào tạo phù hợp.

8.4.1.12 Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào ở mục 8.2.

Các yêu cầu trong mục 8.4.1.13 tham chiếu Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng.

8.4.1.13 Khi đã lập xong Kế hoạch chuẩn bị ứng dụng, việc thẩm tra phải xem xét các vấn đề sau:

a) Kế hoạch chuẩn bị ứng dụng có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến mục 5.3.2.10 và từ mục 6.5.4.14 đến mục 6.5.4.17 cũng như các yêu cầu cụ thể từ mục 8.4.1.2 đến mục 8.4.1.11.

b) Sự thống nhất nội bộ của Kế hoạch chuẩn bị ứng dụng.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng.

8.4.1.14 Phải thẩm tra và thẩm định việc thực hiện Kế hoạch chuẩn bị ứng dụng cho từng ứng dụng cụ thể.

8.4.2 Chỉ dẫn các yêu cầu ứng dụng

8.4.2.1 Chỉ dẫn các yêu cầu ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị quản lý các yêu cầu, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.2.2 đến 8.4.2.3 tham chiếu tới Chỉ dẫn các yêu cầu ứng dụng.

8.4.2.2 Trong các yêu cầu đối với ứng dụng cụ thể phải có các yêu cầu cụ thể cho từng lắp đặt (ví dụ: bố trí đường ray, các cột tín hiệu, các giới hạn tốc độ của hệ thống tín hiệu), cũng như tóm tắt lại hoặc xem xét các điều kiện ứng dụng của phần mềm chung và các chương trình ứng dụng, và các tiêu chuẩn phải tuân thủ của ứng dụng (ví dụ: các nguyên tắc phát tín hiệu của hệ thống tín hiệu).

8.4.2.3 Phải quy định các yêu cầu liên quan tới các thuật toán và dữ liệu ứng dụng được xử lý bởi phần mềm chung của hệ thống tại giai đoạn này.

8.4.2.4 Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị quản lý các yêu cầu, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.2.5 tham chiếu tới Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng.

8.4.2.5 Khi đã lập xong Chỉ dẫn các yêu cầu ứng dụng, việc thẩm tra phải xem xét các vấn đề sau:

a) Chỉ dẫn các yêu cầu ứng dụng có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể từ mục 8.4.2.2 đến 8.4.2.3.

b) Sự thống nhất nội bộ của Chỉ dẫn các yêu cầu ứng dụng

Các kết quả phải được ghi lại trong Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng.

8.4.3 Cấu trúc và thiết kế

8.4.3.1 Phải quy định số lượng và kiểu loại các thành phần phần cứng và phần mềm chung được sử dụng trong ứng dụng cụ thể. Vị trí của các thành phần, các thuật toán và dữ liệu ứng dụng trong cấu trúc ứng dụng cụ thể phải được xác định. Các thuật toán và dữ liệu ứng dụng được xử lý bởi phần mềm chung phải được thiết kế tại giai đoạn này.

8.4.4 Phát triển các thuật toán / dữ liệu ứng dụng

8.4.4.1 Quá trình phát triển ứng dụng phải bao gồm việc lập và biên dịch mã nguồn của các thuật toán / dữ liệu ứng dụng cụ thể và chung, cũng như các hoạt động thẩm tra và kiểm thử liên quan đến quá trình lập này. Khuyến nghị sử dụng các ngôn ngữ có dạng sơ đồ để tạo mã nguồn cho các thuật toán ứng dụng. Tham chiếu Bảng A.16.

8.4.4.2 Báo cáo kiểm thử ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị kiểm thử, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.4.3 tham chiếu tới Báo cáo kiểm thử ứng dụng.

8.4.4.3 Báo cáo kiểm thử ứng dụng phải ghi lại các lần chạy chính xác và hoàn chỉnh của các kiểm thử được quy định trong Chỉ dẫn kiểm thử ứng dụng.

8.4.4.4 Báo cáo thẩm tra chuẩn bị ứng dụng phải:

a) Ghi lại mọi hoạt động được thực hiện để đảm bảo sự chính xác và hoàn chỉnh của dữ liệu / thuật toán và sự gắn kết với các nguyên lý ứng dụng và cấu trúc ứng dụng cụ thể.

b) Đánh giá mức độ tương thích của dữ liệu / thuật toán với ứng dụng chung.

8.4.4.5 Chỉ dẫn kiểm thử ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị kiểm thử, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.4.6 tham chiếu tới Chỉ dẫn kiểm thử ứng dụng.

8.4.4.6 Chỉ dẫn kiểm thử ứng dụng phải quy định các kiểm thử được tiến hành ở giai đoạn trung gian hoặc giai đoạn cuối cùng của quá trình chuẩn bị dữ liệu / thuật toán, để đảm bảo:

a) Sự gắn kết và hoàn chỉnh của dữ liệu/ thuật toán với các nguyên lý ứng dụng.

b) Sự gắn kết và hoàn chỉnh của dữ liệu/ thuật toán với cấu trúc ứng dụng cụ thể.

8.4.4.7 Báo cáo thẩm tra ứng dụng / thuật toán phải được lập thành văn bản, dưới trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.4.8 tham chiếu tới Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng.

8.4.4.8 Khi đã lập xong Chỉ dẫn kiểm thử ứng dụng, việc thẩm tra phải xem xét các vấn đề sau:

a) Chỉ dẫn các yêu cầu ứng dụng có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 8.4.4.6.

b) Sự thống nhất nội bộ của Chỉ dẫn kiểm thử ứng dụng.

Các kết quả phải được ghi lại trong Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng.

8.4.5 Tích hợp ứng dụng và chấp nhận kiểm thử

8.4.5.1 Đối với một số hệ thống, các thuật toán / dữ liệu ứng dụng có thể được tích hợp cùng với phần cứng và phần mềm chung trong kiểm thử xuất xưởng trước khi lắp đặt vào hệ thống mục tiêu. Điều này có thể không cần thiết nếu có thể đạt được đủ mức độ tin cậy bằng các phương pháp khác. Ứng dụng sau đó phải được lắp đặt vào hệ thống mục tiêu, và phải thực hiện các kiểm thử tích hợp trong các lắp đặt hoàn chỉnh. Hệ thống mục tiêu cuối cùng phải được chạy thử như một hệ thống hoạt động đầy đủ và thực hiện quá trình chấp nhận cuối cùng hệ thống mục tiêu khi lắp đặt hoàn chỉnh. Báo cáo kiểm thử ứng dụng phải ghi lại quá trình chạy chương trình đúng và hoàn chỉnh của các kiểm thử được quy định trong Chỉ dẫn kiểm thử ứng dụng. Báo cáo thẩm tra chuẩn bị ứng dụng phải kiểm tra sự hoàn chỉnh và sự chính xác của các kiểm thử được tiến hành khi lắp đặt hoàn chỉnh.

8.4.5.2 Chỉ dẫn kiểm thử ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị kiểm thử, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.4.8 tham chiếu tới Chỉ dẫn kiểm thử ứng dụng.

8.4.5.3 Chỉ dẫn kiểm thử ứng dụng phải quy định các kiểm thử được tiến để đảm bảo:

a) Sự tích hợp chính xác dữ liệu / thuật toán trên phần mềm và phần cứng chung, nếu cần thiết.

b) Sự tích hợp chính xác dữ liệu / thuật toán khi lắp đặt hoàn chỉnh.

8.4.5.4 Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào trong mục 8.2.

Các yêu cầu từ mục 8.4.5.5 tham chiếu tới Báo cáo thẩm tra các thuật toán / dữ liệu ứng dụng.

8.4.5.5 Khi đã lập xong Chỉ dẫn kiểm thử ứng dụng, việc thẩm tra phải xem xét xem Chỉ dẫn kiểm thử ứng dụng có đáp ứng được các yêu cầu cụ thể trong mục 8.4.5.3.

8.4.6 Thẩm định và đánh giá ứng dụng

Các hoạt động thẩm định và đánh giá phải đánh giá hiệu năng của từng giai đoạn vòng đời.

8.4.7 Các chương trình và các quy trình chuẩn bị ứng dụng

8.4.7.1 Đối với từng loại hệ thống mới được cấu hình bằng các thuật toán / dữ liệu ứng dụng, các chương trình và quy trình cụ thể phải được xây dựng để cho phép áp dụng quá trình phát triển ứng dụng quy định trong mục 8.4.1 cho các lắp đặt hệ thống mới. Việc phát triển những chương trình này phải được thực hiện phù hợp với tiêu chuẩn này, cũng như phù hợp với phần mềm và phần cứng

chung trong hệ thống. Các hoạt động thẩm tra, thẩm định và đánh giá phải đảm bảo các chương trình chuẩn bị dữ liệu và phần mềm chung là tương thích.

8.4.7.2 Mọi quá trình phát triển phải được thẩm định và đánh giá. Phải chú ý là sẽ thường cần các chương trình biên dịch chuyên dụng để chuyển đổi dữ liệu và thuật toán.

8.4.7.3 Tất cả các thuật toán / dữ liệu ứng dụng và tài liệu liên quan đến từng ứng dụng cụ thể phải theo các yêu cầu triển khai phần mềm như được quy định trong mục 9.1.

8.4.7.4 Tất cả các thuật toán / dữ liệu ứng dụng và tài liệu liên quan đến từng ứng dụng cụ thể phải theo các yêu cầu bảo trì phần mềm như được quy định trong mục 9.2.

8.4.7.5 Tất cả các thuật toán / dữ liệu ứng dụng và tài liệu liên quan đến từng ứng dụng cụ thể phải được quản lý cấu hình theo các yêu cầu quy định trong mục 6.5 và 6.7. Việc quản lý cấu hình các thuật toán / dữ liệu ứng dụng có thể độc lập với việc quản lý phần mềm chung.

8.4.7.6 Báo cáo thẩm tra ứng dụng chứng minh mức độ và sự bắt buộc các điều kiện ứng dụng của phần mềm chung và các chương trình ứng dụng.

8.4.8 Phát triển phần mềm chung

8.4.8.1 Việc phát triển phần mềm chung mà hỗ trợ cho việc chạy các thuật toán / dữ liệu ứng dụng phải phù hợp với các yêu cầu từ mục 7.1 đến 7.7 của tiêu chuẩn này. Các yêu cầu bổ sung dưới đây cũng phải được giám sát.

8.4.8.2 Phải xác định rõ các loại hoặc các cấp độ chức năng có thể được cấu hình bằng các thuật toán / dữ liệu ứng dụng trong từng hệ thống và hệ thống con trong các tài liệu Chỉ dẫn các yêu cầu phần mềm của phần mềm chung. Mức toàn vẹn về an toàn được chỉ định cho các chức năng sẽ xác định các tiêu chuẩn được áp dụng cho quá trình phát triển sau này của các thuật toán / dữ liệu ứng dụng cho tất cả các lắp đặt trong hệ thống.

8.4.8.3 Trong quá trình thiết kế phần mềm chung, phải quy định các giao diện chi tiết giữa phần mềm chung và các thuật toán / dữ liệu ứng dụng, trừ khi các giao diện này đã được quy định ở giai đoạn vòng đời trước đó, ví dụ: theo yêu cầu về sử dụng một ngôn ngữ ứng dụng cụ thể hiện có.

8.4.8.4 Phải tăng cường sự phân tách rõ ràng giữa phần mềm chung và các thuật toán / dữ liệu ứng dụng, ví dụ: phải có khả năng biên dịch lại và cập nhật phần mềm chung hoặc các thuật toán / dữ liệu ứng dụng mà không cần phải cập nhật những hạng mục khác, trừ khi có sự thay đổi về giao diện xác định giữa phần mềm chung và các thuật toán / dữ liệu ứng dụng. Mặt khác, phải tách biệt các thuật toán / dữ liệu cụ thể theo các ứng dụng với các thuật toán / dữ liệu ứng dụng chung.

8.4.8.5 Các quy trình kiểm soát sự thay đổi phải đảm bảo mọi thay đổi của phần mềm chung chỉ có thể được lắp đặt sau khi đã xây dựng được phần mềm đã được sửa đổi là tương thích với các thuật toán / dữ liệu ứng dụng gốc hoặc các thuật toán / dữ liệu ứng dụng đã được sửa đổi.

8.4.8.6 Phải chú ý trong quá trình thẩm tra và giai đoạn kiểm thử thẩm định của phần mềm chung để đảm bảo tất cả các kết hợp dữ liệu và thuật toán liên quan đã được xem xét.

Nếu chưa xem xét tất cả các kết hợp dữ liệu và thuật toán liên quan trong quá trình thẩm tra, kiểm thử và thẩm định của phần mềm chung, phải xác định rõ các kết hợp này như là hạn chế sử dụng của phần mềm chung. Phải thực hiện hoàn chỉnh quá trình thẩm tra, kiểm thử và thẩm định phần mềm chung khi một số dữ liệu hoặc thuật toán được xác định là vượt quá hạn chế này.

8.4.8.7 Phần mềm chung phải được thiết kế để phát hiện các thuật toán / dữ liệu ứng dụng bị lỗi nếu việc thực hiện này khả thi.

8.4.8.8 Các Đơn vị thiết kế phải đưa ra các Thông tin lưu ý về phiên bản phần mềm chung và các chương trình ứng dụng trong giai đoạn Kiểm thử tổng thể phần mềm/thẩm định lần cuối của phần mềm chung và các chương trình ứng dụng. Nội dung của những tài liệu này phải theo các hoạt động thẩm tra và thẩm định.

Các nội dung dưới đây phải được đề cập đến trong tài liệu “Các điều kiện áp dụng của phần mềm chung và các chương trình ứng dụng”:

1) Các tham chiếu tới hướng dẫn sử dụng phần mềm chung và các chương trình ứng dụng.

2) Mọi ràng buộc của các thuật toán / dữ liệu ứng dụng, ví dụ: Cấu trúc hoặc các quy tắc mã hóa bắt buộc để đáp ứng các mức toàn vẹn về an toàn.

9 Triển khai và bảo trì phần mềm

9.1 Triển khai phần mềm

9.1.1 Mục tiêu

9.1.1.1 Để đảm bảo phần mềm hoạt động như được yêu cầu, duy trì mức toàn vẹn về an toàn yêu cầu và độ tin cậy khi triển khai phần mềm trong môi trường ứng dụng cuối cùng.

9.1.2 Tài liệu đầu vào

Tất cả các tài liệu thiết kế, phát triển và phân tích liên quan đến quá trình triển khai.

9.1.3 Tài liệu đầu ra

1) Kế hoạch phát hành và triển khai phần mềm

2) Hướng dẫn triển khai phần mềm

3) Thông tin lưu ý về phiên bản phần mềm sử dụng

4) Các biên bản triển khai

5) Báo cáo thẩm tra quá trình triển khai

9.1.4 Các yêu cầu

9.1.4.1 Phải thực hiện việc triển khai do trách nhiệm của đơn vị quản lý dự án.

9.1.4.2 Trước khi chuyển giao phát hành phần mềm, cơ sở của phần mềm phải được ghi lại và duy trì việc truy vết theo quá trình kiểm soát quản lý cấu hình. Phải bao gồm cả phần mềm đã có trước đó và phần mềm được phát triển theo phiên bản trước đây của tiêu chuẩn này.

9.1.4.3 Việc phát hành phần mềm phải được tái lập lại trong suốt vòng đời cơ sở.

9.1.4.4 Thông tin lưu ý về phiên bản phần mềm phải được lập thành văn bản, do trách nhiệm của Đơn vị thiết kế, trên cơ sở các tài liệu của mục 9.1.2.

Các yêu cầu trong mục 9.1.4.5 tham chiếu tới Thông tin lưu ý về phiên bản phần mềm sử dụng.

9.1.4.5 Thông tin lưu ý về phiên bản phần mềm sử dụng phải đưa ra:

a) Các điều kiện áp dụng phải được tuân thủ.

b) Thông tin về khả năng tương thích giữa các thành phần phần mềm và giữa phần mềm với phần cứng.

c) Tất cả các hạn chế trong quá trình sử dụng phần mềm (xem 7.7.4.12).

9.1.4.6 Hướng dẫn triển khai phần mềm phải được lập thành văn bản trên cơ sở các tài liệu đầu vào trong mục 9.1.2.

Các yêu cầu trong mục 9.1.4.7 tham chiếu Hướng dẫn triển khai phần mềm.

9.1.4.7 Hướng dẫn triển khai phần mềm phải xác định các quy trình để xác định rõ và thực hiện chính xác quá trình phát hành phần mềm.

9.1.4.8 Trong trường hợp triển khai mang tính bổ sung tăng cường (ví dụ: triển khai các thành phần đơn lẻ), khuyến nghị cao đối với SIL 3 và SIL 4, khuyến nghị với SIL 1 và SIL 2 việc phần mềm được

thiết kế có các bộ phận đảm bảo kích hoạt loại trừ các phiên bản không tương thích của các thành phần trong phần mềm.

9.1.4.9 Quá trình quản lý cấu hình phải đảm bảo việc có mặt đồng thời các phiên bản khác nhau của các thành phần phần mềm giống nhau không phát sinh ra sự gây hại nếu việc này không thể tránh được.

9.1.4.10 Phải có quy trình quay vòng (ví dụ: khả năng quay về các phiên bản trước đây) khi triển khai lắp đặt một phần mềm mới.

9.1.4.11 Phần mềm phải có các cơ chế tự xác định phiên bản được tích hợp, cho phép tự xác định trong quá trình hoạt động và sau khi nạp vào đối tượng. Cơ chế tự xác định nên chỉ rõ thông tin về phiên bản của phần mềm đối với phần mềm và mọi dữ liệu cấu hình cũng như định dạng sản phẩm.

Chú thích: Khuyến nghị bảo vệ các dữ liệu có trong các đoạn mã có chưa các thông tin về phát hành phần mềm trong suốt quá trình mã hóa (xem Bảng A.3 “Các mã phát hiện lỗi”).

9.1.4.12 Biên bản triển khai phải được lập thành văn bản trên cơ sở các tài liệu đầu vào trong mục 9.1.2.

Yêu cầu trong mục 9.1.4.13 tham chiếu tới Biên bản triển khai phần mềm.

9.1.4.13 Biên bản triển khai phải đưa ra bằng chứng về việc phần mềm đã được tải lên, bằng cách kiểm tra các cơ chế tự xác định đã được tích hợp sẵn (xem 9.1.4.11). Biên bản này phải được lưu trữ trong các tài liệu liên quan đến hệ thống được chuyển giao giống như các lần thẩm tra khác và là một phần trong quá trình thử hoạt động và chấp nhận.

9.1.4.14 Phần mềm được triển khai phải có khả năng truy vết theo các lắp đặt được chuyển giao.

Chú thích: Việc này đặc biệt quan trọng khi phát hiện ra các sự cố nghiêm trọng và cần phải được hiệu chỉnh trong nhiều hơn một lần cài đặt.

9.1.4.15 Phần mềm phải đưa ra được các thông tin chuẩn đoán, và là một phần của hoạt động giám sát sự cố.

9.1.4.16 Biên bản thẩm tra triển khai phải được lập thành văn bản, do trách nhiệm của Đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào của mục 9.1.2.

Các yêu cầu của mục 9.1.4.17 đến 9.1.4.19 tham chiếu Báo cáo thẩm tra triển khai.

9.1.4.17 Khi đã lập xong Hướng dẫn triển khai phần mềm, việc thẩm tra phải xem xét các vấn đề sau:

a) Hướng dẫn triển khai phần mềm có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 9.1.4.7.

b) Sự thống nhất nội bộ của Hướng dẫn triển khai phần mềm.

9.1.4.18 Khi đã lập xong Biên bản triển khai, việc thẩm tra phải xem xét các vấn đề sau:

a) Biên bản triển khai có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 9.1.4.13.

b) Sự thống nhất nội bộ của Biên bản triển khai.

9.1.4.19 Khi đã lập xong Hướng dẫn đi kèm, việc thẩm tra phải xem xét các vấn đề sau:

a) Hướng dẫn đi kèm có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 9.1.4.5.

b) Sự thống nhất nội bộ của Hướng dẫn đi kèm.

9.1.4.20 Phải có các biện pháp trong gói phần mềm để ngăn chặn hoặc phát hiện ra các lỗi phát sinh trong quá trình lưu trữ, chuyển đổi, chạy hoặc lặp lại các đoạn mã hoặc dữ liệu thực hiện. Khuyến nghị mã hóa các đoạn mã chạy trong quá trình nạp.

9.2 Bảo trì phần mềm

9.2.1 Mục tiêu

9.2.1.1 Để đảm bảo phần mềm hoạt động như yêu cầu, duy trì mức toàn vẹn về an toàn phần mềm yêu cầu và độ tin cậy khi tiến hành sửa chữa, cải tiến hoặc thay đổi phù hợp với chính phần mềm. Xem thêm mục 6.6 “Kiểm soát sự thay đổi và cải tiến” trong tiêu chuẩn này và giai đoạn 13 “Cải tiến và thay đổi” trong TCVN 10935-1.

9.2.2 Tài liệu đầu vào

Tất cả các tài liệu thiết kế, phát triển và phân tích liên quan.

9.2.3 Tài liệu đầu ra

1) Kế hoạch bảo trì phần mềm.

2) Các biên bản thay đổi phần mềm.

3) Các biên bản bảo trì phần mềm.

4) Báo cáo thẩm tra bảo trì phần mềm

9.2.4 Các yêu cầu

9.2.4.1 Mặc dù tiêu chuẩn này không nhằm áp dụng cho các phần mềm cũ mà chỉ áp dụng cho các phần mềm xây dựng mới và chỉ áp dụng cho toàn bộ phần mềm hiện có nếu nó có các thay đổi lớn, quá trình bảo trì phần mềm liên quan trong mục 9.2 này sẽ áp dụng cho tất cả các thay đổi, kể cả các thay đổi nhỏ. Tuy nhiên, khuyến nghị cao áp dụng toàn bộ tiêu chuẩn này trong quá trình nâng cấp và bảo trì các phần mềm hiện có.

9.2.4.2 Đối với mọi mức toàn vẹn về an toàn phần mềm, nhà cung cấp trước khi bắt đầu công việc thay đổi phải quyết định liệu các hoạt động bảo trì được xem là lớn hay nhỏ hoặc các phương pháp bảo trì hệ thống có phù hợp. Nhà cung cấp phải chứng minh và ghi lại quyết định này và đệ trình lên Đơn vị đánh giá để đánh giá.

9.2.4.3 Phải thực hiện việc bảo trì phù hợp với các hướng dẫn có trong tiêu chuẩn ISO/IEC 9000-3.

9.2.4.4 Phải thiết kế khả năng bảo trì như một đặc tính vốn có của phần mềm, đặc biệt theo các yêu cầu của mục 7.3, 7.4 và 7.5. Phải sử dụng bộ tiêu chuẩn ISO/IEC 9126 để thực hiện và thẩm tra mức độ khả năng bảo trì nhỏ nhất.

9.2.4.5 Kế hoạch bảo trì phần mềm phải được lập thành văn bản trên cơ sở các tài liệu đầu vào của mục 9.2.2.

Các yêu cầu trong mục 9.2.4.6 tham chiếu tới Kế hoạch bảo trì phần mềm.

9.2.4.6 Phải thiết lập và ghi lại các quy trình bảo trì phần mềm trong Kế hoạch bảo trì phần mềm. Các quy trình này cũng phải đề cập tới các vấn đề sau

a) Kiểm soát việc báo cáo lỗi, ghi lại lỗi, các biên bản bảo trì, các xác nhận thay đổi và cấu hình phần mềm/hệ thống và các kỹ thuật, biện pháp có trong Bảng A.10.

b) Việc thẩm tra, thẩm định và đánh giá mọi sự thay đổi.

c) Các xác nhận đối với việc thay đổi.

9.2.4.7 Biên bản bảo trì phần mềm phải được lập thành văn bản trên cơ sở các tài liệu đầu vào trong mục 9.2.2.

Yêu cầu trong mục 9.2.4.8 tham chiếu Biên bản bảo trì phần mềm.

9.2.4.8 Phải lập Biên bản bảo trì phần mềm cho từng hạng mục của phần mềm trước khi phát hành lần đầu tiên và phải duy trì việc này. Để bổ sung cho các yêu cầu trong ISO/IEC 90003:2004 về “Các biên bản và báo cáo bảo trì” (xen ISO/IEC 90003:2004, mục “Bảo trì”), biên bản này phải có:

a) Các tham chiếu đến tất cả các Biên bản thay đổi phần mềm cho hạng mục phần mềm đó.

b) Việc đánh giá tác động của thay đổi.

c) Các trường hợp kiểm thử các thành phần, bao gồm dữ liệu về việc thẩm định lại và kiểm thử hồi quy.

d) Lịch sử cấu hình phần mềm.

9.2.4.9 Biên bản thay đổi phần mềm phải được lập thành văn bản trên cơ sở các tài liệu đầu vào trong mục 9.2.2.

Yêu cầu trong mục 9.2.4.10 tham chiếu Biên bản thay đổi phần mềm.

9.2.4.10 Phải lập Biên bản thay đổi phần mềm cho từng hoạt động bảo trì. Biên bản này phải có:

a) Yêu cầu thay đổi hoặc cải tiến, phiên bản, bản chất của sự cố, thay đổi cần thiết và nguồn lực để thay đổi.

b) Phân tích tác động của hoạt động bảo trì đối với tổng thể hệ thống, bao gồm tương tác phần cứng, phần mềm, con người và môi trường và các tương tác có thể.

c) Chỉ dẫn kỹ thuật chi tiết việc thay đổi hoặc cải tiến được tiến hành.

d) Quá trình thẩm định lại, kiểm thử hồi quy và đánh giá lại việc thay đổi hoặc cải tiến theo mức toàn vẹn về an toàn phần mềm yêu cầu. Trách nhiệm đối với việc thẩm định lại có thể thay đổi theo dự án, tùy thuộc vào mức toàn vẹn về an toàn phần mềm. Đồng thời, tác động của việc thay đổi hoặc cải tiến đối với quá trình thẩm định lại có thể bị hạn chế theo các mức độ hệ thống khác nhau (chỉ có các tổng thành bị thay đổi, tất cả các tổng thành được xác định là bị tác động, toàn bộ hệ thống). Do đó Kế hoạch thẩm định phần mềm phải đề cập tới các vấn đề này theo mức toàn vẹn về an toàn phần mềm. Mức độ độc lập của quá trình thẩm định lại phải giống với mức độ như khi thẩm định mới.

9.2.4.11 Phải lập Báo cáo thẩm tra bảo trì phần mềm, do trách nhiệm của đơn vị thẩm tra, trên cơ sở các tài liệu đầu vào trong mục 9.2.2.

Các yêu cầu từ mục 9.2.4.12 đến 9.2.4.14 tham chiếu tới Báo cáo thẩm tra bảo trì phần mềm.

9.2.4.12 Khi đã lập xong Kế hoạch bảo trì phần mềm, việc thẩm tra phải xem xét các vấn đề sau:

a) Kế hoạch bảo trì phần mềm có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 9.2.4.6.

b) Sự thống nhất nội bộ của Kế hoạch bảo trì phần mềm.

9.2.4.13 Khi đã lập xong Biên bản bảo trì phần mềm, việc thẩm tra phải xem xét các vấn đề sau:

a) Biên bản bảo trì phần mềm mềm có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 9.2.4.8.

b) Sự thống nhất nội bộ của Biên bản bảo trì phần mềm.

9.2.4.14 Khi đã lập xong Biên bản thay đổi phần mềm, việc thẩm tra phải xem xét các vấn đề sau:

a) Biên bản thay đổi phần mềm có đáp ứng các yêu cầu chung về khả năng đọc được và khả năng truy vết từ mục 5.3.2.7 đến 5.3.2.10 và từ mục 6.5.4.14 đến 6.5.4.17 cũng như các yêu cầu cụ thể trong mục 9.2.4.10.

b) Sự thống nhất nội bộ của Biên bản thay đổi phần mềm.

9.2.4.15 Phải tiến hành các hoạt động bảo trì theo Kế hoạch bảo trì phần mềm.

9.2.4.16 Phải lựa chọn các kỹ thuật và các biện pháp trong Bảng A.10. Phải chứng minh việc kết hợp lựa chọn là thỏa mãn mục 4.8 và 4.9.

9.2.4.17 Phải thực hiện việc bảo trì tối thiểu với mức độ về kinh nghiệm, chương trình, ghi chép lưu trữ, lập kế hoạch và quản lý giống với mức độ khi phát triển phần mềm lúc đầu. Việc này cũng phải áp dụng cho việc quản lý cấu hình, kiểm soát thay đổi, kiểm soát tài liệu và mức độ độc lập của các bên liên quan.

9.2.4.18 Phải quản lý các hoạt động kiểm soát nhà cung cấp từ bên ngoài, các hoạt động báo cáo sự cố và hoạt động khắc phục với chỉ tiêu được quy định trong các nội dung liên quan trong mục Đảm bảo chất lượng phần mềm (6.5) giống như khi phát triển phần mềm mới.

9.2.4.19 Đối với từng sự cố hoặc thay đổi được báo cáo, phải thực hiện phân tích tác động đến an toàn.

9.2.4.20 Đối với phần mềm được bảo trì, phải tiến hành các hoạt động giảm thiểu được cân đối theo rủi ro đã xác định để đảm bảo tính toàn vẹn tổng thể của hệ thống, khi mà các vấn đề báo cáo được điều tra và khắc phục.

Phụ lục A

(Quy định)

Tiêu chí lựa chọn các kỹ thuật và biện pháp

Các điều khoản trong tiêu chuẩn được kết hợp trong phụ lục này thông qua các bảng (xem mục A.1, từ Bảng A.1 đến Bảng A.11) để minh họa các biện pháp đạt được sự phù hợp. Có những bảng ở mức độ thấp hơn, có bảng được nêu chi tiết (xem A.2, từ Bảng A.12 đến Bảng A.23), mở rộng dựa trên các chủ đề nhất định trong các bảng. Ví dụ, “Lập mô hình” trong Bảng A.2 và được nêu chi tiết trong Bảng A.17. Phụ lục tham khảo D tham chiếu từ các bảng này.

Với từng kỹ thuật hoặc biện pháp trong các bảng, sẽ có yêu cầu cho từng mức toàn vẹn về an toàn phần mềm (SIL). Trong tiêu chuẩn này, các yêu cầu cho các mức toàn vẹn về an toàn phần mềm 1 và 2 là giống nhau đối với từng kỹ thuật. Tương tự, mỗi kỹ thuật sẽ có cùng các yêu cầu cho mức toàn vẹn về an toàn 3 và 4. Những yêu cầu này có thể là:

‘M’ Ký hiệu này nghĩa là việc sử dụng kỹ thuật là bắt buộc (Mandatory)

‘HR’ Ký hiệu này nghĩa là kỹ thuật hoặc biện pháp là Khuyến nghị cao (High Recommended) đối với mức toàn vẹn về an toàn này. Nếu kỹ thuật hoặc biện pháp này không được sử dụng thì căn cứ cơ sở cho việc không sử dụng nó nên phải được nêu chi tiết trong Kế hoạch đảm bảo chất lượng phần mềm hoặc trong tài liệu khác tham chiếu trong Kế hoạch đảm bảo chất lượng phần mềm

‘R’ Ký hiệu này nghĩa là kỹ thuật hoặc biện pháp là (Khuyến nghị) Recommended đối với mức toàn vẹn về an toàn. Đây là mức khuyến nghị thấp hơn mức ‘HR’ và những kỹ thuật như vậy có thể được kết hợp để tạo nên một phần trong đó

‘-‘ Ký hiệu này nghĩa là Kỹ thuật hoặc biện pháp không được khuyến nghị hoặc đang được sử dụng phụ thuộc

‘NR’ Ký hiệu này nghĩa là kỹ thuật hoặc biện pháp là Không được khuyến nghị (Not Recommended) theo hướng tích cực đối với mức toàn vẹn về an toàn này. Nếu kỹ thuật hoặc biện pháp được sử dụng thì căn cứ cơ sở cho việc sử dụng nó nên được nêu chi tiết trong Kế hoạch đảm bảo chất lượng phần mềm hoặc trong tài liệu khác tham chiếu trong Kế hoạch đảm bảo chất lượng phần mềm

Việc kết hợp các kỹ thuật và biện pháp sẽ được tuyên bố trong Kế hoạch đảm bảo chất lượng phần mềm hoặc trong các tài liệu khác tham chiếu qua Kế hoạch đảm bảo chất lượng phần mềm cùng với một hoặc nhiều kỹ thuật, biện pháp được lựa chọn trừ khi có chú ý đi kèm với bảng đưa ra các yêu cầu khác. Những chú ý này có thể bao gồm tham chiếu đến các kỹ thuật và kết hợp các kỹ thuật đã được phê duyệt. Nếu những kỹ thuật hoặc việc kết hợp các kỹ thuật này, bao gồm tất cả các kỹ thuật bắt buộc tương ứng được sử dụng, thì Đơn vị đánh giá phải chấp nhận chúng là đúng và chỉ phải quan tâm liệu chúng có được áp dụng một cách chính xác. Nếu có sử dụng và có thể kết luận về một tập hợp các kỹ thuật khác thì khi đó Đơn vị đánh giá có thể đánh giá nó là chấp nhận được.

A.1 Các bảng

Bảng A.1 – Các vấn đề vòng đời và lưu trữ (5.3)

TÀI LIỆU	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
Lập kế hoạch					
1. Kế hoạch đảm bảo chất lượng phần mềm	HR	HR	HR	HR	HR
2. Báo cáo thẩm tra đảm bảo chất lượng phần mềm	HR	HR	HR	HR	HR
3. Kế hoạch quản lý cấu hình phần mềm	HR	HR	HR	HR	HR
4. Kế hoạch thẩm tra phần mềm	HR	HR	HR	HR	HR
5. Kế hoạch thẩm định phần mềm	HR	HR	HR	HR	HR
Các yêu cầu phần mềm					
6. Chỉ dẫn các yêu cầu phần mềm	HR	HR	HR	HR	HR
7. Chỉ dẫn kiểm thử tổng thể phần mềm	HR	HR	HR	HR	HR
8. Báo cáo thẩm tra các yêu cầu phần mềm	HR	HR	HR	HR	HR
Cấu trúc và thiết kế					
9. Chỉ dẫn cấu trúc phần mềm	HR	HR	HR	HR	HR
10. Chỉ dẫn thiết kế phần mềm	HR	HR	HR	HR	HR
11. Chỉ dẫn giao diện phần mềm	HR	HR	HR	HR	HR
12. Chỉ dẫn kiểm thử tích hợp phần mềm	HR	HR	HR	HR	HR
13. Chỉ dẫn kiểm thử tích hợp phần cứng/phần mềm	HR	HR	HR	HR	HR
14. Báo cáo thẩm tra thiết kế và cấu trúc phần mềm	HR	HR	HR	HR	HR
Thiết kế thành phần					
15. Chỉ dẫn thiết kế thành phần phần mềm	R	HR	HR	HR	HR
16. Chỉ dẫn kiểm thử thành phần phần mềm	R	HR	HR	HR	HR
17. Báo cáo thẩm tra thiết kế thành phần phần mềm	R	HR	HR	HR	HR
Xây dựng và kiểm thử thành phần phần mềm					
18. Tài liệu hỗ trợ và mã nguồn phần mềm	HR	HR	HR	HR	HR
19. Báo cáo kiểm thử thành phần phần mềm	R	HR	HR	HR	HR
20. Báo cáo thẩm tra mã nguồn phần mềm	HR	HR	HR	HR	HR
Tích hợp					
21. Báo cáo kiểm thử tích hợp phần mềm	HR	HR	HR	HR	HR
22. Báo cáo kiểm thử tích hợp phần cứng/phần mềm	HR	HR	HR	HR	HR
23. Báo cáo thẩm tra tích hợp phần mềm	HR	HR	HR	HR	HR
Kiểm thử /thẩm định lần cuối tổng quát phần mềm					
24. Báo cáo kiểm thử tổng thể phần mềm	HR	HR	HR	HR	HR
25. Báo cáo thẩm định phần mềm	HR	HR	HR	HR	HR
26. Báo cáo thẩm định các chương trình	R	HR	HR	HR	HR
27. Thông tin lưu ý về phiên bản phần mềm sử dụng	HR	HR	HR	HR	HR

Bảng A.1 – Các vấn đề vòng đời và lưu trữ (5.3) *(kết thúc)*

TÀI LIỆU	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4

Các hệ thống được cấu hình bằng các thuật toán/dữ liệu ứng dụng					
28. Chỉ dẫn các yêu cầu ứng dụng	HR	HR	HR	HR	HR
29. Kế hoạch chuẩn bị ứng dụng (xem Chú thích 2)	HR	HR	HR	HR	HR
30. Chỉ dẫn kiểm thử ứng dụng (xem Chú thích 2)	HR	HR	HR	HR	HR
31. Thiết kế và cấu trúc ứng dụng (xem Chú thích 2)	HR	HR	HR	HR	HR
32. Báo cáo thẩm tra chuẩn bị ứng dụng	HR	HR	HR	HR	HR
33. Báo cáo kiểm thử ứng dụng	HR	HR	HR	HR	HR
34. Mã nguồn của các thuật toán/dữ liệu ứng dụng	HR	HR	HR	HR	HR
35. Báo cáo thẩm tra các thuật toán/dữ liệu ứng dụng	HR	HR	HR	HR	HR
Triển khai phần mềm					
36. Kế hoạch triển khai và phát hành phần mềm	R	HR	HR	HR	HR
37. Hướng dẫn triển khai phần mềm	R	HR	HR	HR	HR
38. Thông tin lưu ý về phiên bản phần mềm sử dụng	HR	HR	HR	HR	HR
39. Biên bản triển khai	R	HR	HR	HR	HR
40. Báo cáo thẩm tra việc triển khai	R	HR	HR	HR	HR
Bảo trì phần mềm					
41. Kế hoạch bảo trì phần mềm	R	HR	HR	HR	HR
42. Biên bản thay đổi phần mềm	HR	HR	HR	HR	HR
43. Biên bản bảo trì phần mềm	R	HR	HR	HR	HR
44. Báo cáo thẩm tra việc bảo trì phần mềm	R	HR	HR	HR	HR
Đánh giá phần mềm					
45. Kế hoạch đánh giá phần mềm	R	HR	HR	HR	HR
46. Báo cáo đánh giá phần mềm	R	HR	HR	HR	HR

Chú thích 1: Các tài liệu được kết hợp theo cách khác thì phải tuân theo mục 5.3.2.11 và 5.3.2.12.

Chú thích 2: Tài liệu số 29, 30 và 31 sẽ là HR hay R phụ thuộc vào mức quan trọng được xác định trong quá trình và hạng mục cần thực hiện việc thẩm tra. Ví dụ, dữ liệu chỉ cần được thẩm tra nhưng trong phạm vi hệ thống thì được kiểm thử trong khi đó các đặc tính có tính chức năng hơn thì cần cả thẩm tra và kiểm thử. Trong trường hợp này HR là được xác định nhưng có thể chọn R.

Bảng A.2 – Chỉ dẫn các yêu cầu phần mềm

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Các biện pháp hình thức (dựa trên phương pháp toán học)	D.28	-	R	R	HR	HR
2. Lập mô hình	Bảng A.17	R	R	R	HR	HR
3. Các biện pháp có tính cấu trúc	D.52	R	R	R	HR	HR
4. Các bảng so sánh logic	D.13	R	R	R	HR	HR
Các yêu cầu: 1. Chỉ dẫn các yêu cầu phần mềm sẽ bao gồm một bản mô tả vấn đề bằng ngôn ngữ tự nhiên và mọi diễn giải hình thức hoặc bán hình thức cần thiết phản ánh việc ứng dụng. 2. Bảng thể hiện các yêu cầu bổ sung để xác định rõ ràng và chính xác chỉ dẫn kỹ thuật. Phải lựa chọn một hoặc nhiều kỹ thuật này để thỏa mãn mức toàn vẹn về an toàn phần mềm đang được sử dụng.						

Bảng A.3 – Cấu trúc phần mềm (7.3)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Lập trình phòng thủ	D.14	-	HR	HR	HR	HR
2. Chuẩn đoán và xử lý lỗi	D.26	-	R	R	HR	HR
3. Mã sửa lỗi	D.19	-	-	-	-	-
4. Mã phát hiện lỗi	D.19	-	R	R	HR	HR
5. Lập trình xác nhận hư hỏng	D.24	-	R	R	HR	HR
6. Kỹ thuật túi an toàn	D.47	-	R	R	R	R
7. Lập trình đa chiều	D.16	-	R	R	HR	HR
8. Khối phục hồi	D.44	-	R	R	R	R
9. Phục hồi lùi	D.5	-	NR	NR	NR	NR
10. Phục hồi tiến	D.30	-	NR	NR	NR	NR
11. Cơ chế phục hồi sự cố kiểu thử lại	D.46	-	R	R	R	R
12. Ghi nhớ các trường hợp thực hiện	D.36	-	R	R	HR	HR
13. Trí tuệ nhân tạo - Sửa lỗi	D.1	-	NR	NR	NR	NR
14. Tái cấu hình động phần mềm	D.17	-	NR	NR	NR	NR
15. Phân tích tác động lỗi phần mềm	D.25	-	R	R	HR	HR
16. Suy giảm nhẹ	D.31	-	R	R	HR	HR
17. Ẩn thông tin	D.33	-	-	-	-	-
18. Đóng gói thông tin	D.33	R	HR	HR	HR	HR
19. Giao diện được xác định đầy đủ	D.38	HR	HR	HR	M	M
20. Biện pháp hình thức	D.28	-	R	R	HR	HR
21. Lập mô hình	Bảng A.17	R	R	R	HR	HR
22. Các biện pháp có tính cấu trúc	D.52	R	HR	HR	HR	HR
23. Lập mô hình được hỗ trợ bởi máy tính để hỗ trợ các chương trình chỉ dẫn kỹ thuật và thiết kế	Bảng A.17	R	R	R	HR	HR

Các yêu cầu:

1. Việc kết hợp các kỹ thuật đối với mức toàn vẹn về an toàn phần mềm SIL 3 và SIL 4 như sau:
   a) 1, 7, 19, 22 và một từ 4, 5, 12 hoặc 21;
   b) 1, 4, 19, 22 và một từ 2, 5, 12, 15 hoặc 21.

2. Việc kết hợp các kỹ thuật đối với mức toàn vẹn về an toàn phần mềm SIL 1 và SIL 2 như sau: 1, 19, 22 và một từ 2, 4, 5, 7, 12, 15 hoặc 21.

3. Một số các lực chọn này có thể được xác định tại mức hệ thống.

4. Các mã phát hiện lỗi có thể được sử dụng tuân thủ theo các yêu cầu của EN 50159.

Chú thích: Kỹ thuật/biện pháp số 19 là dành cho các giao diện bên ngoài.

Bảng A.4 – Thiết kế và phát triển phần mềm (7.4)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Các biện pháp hình thức	D.28	-	R	R	HR	HR
2. Lập mô hình	Bảng A.17	R	HR	HR	HR	HR
3. Biện pháp có tính cấu trúc	D.52	R	HR	HR	HR	HR
4. Biện pháp tiếp cận module	D.38	HR	M	M	M	M
5. Các thành phần	Bảng A.20	HR	HR	HR	HR	HR
6. Tiêu chuẩn thiết kế và mã hóa	Bảng A.12	HR	HR	HR	M	M
7. Các chương trình phân tích	D.2	HR	HR	HR	HR	HR
8. Ngôn ngữ lập trình mạnh	D.49	R	HR	HR	HR	HR
9. Lập trình theo cấu trúc	D.53	R	HR	HR	HR	HR
10. Ngôn ngữ lập trình	Bảng A.15	R	HR	HR	HR	HR
11. Tập con ngôn ngữ	D.35	-	-	-	HR	HR
12. Lập trình định hướng đối tượng	Bảng A.22 D.57	R	R	R	R	R
13. Lập trình hướng thủ tục	D.60	R	HR	HR	HR	HR
14. Lập trình mê ta	D.59	R	R	R	R	R

Các yêu cầu:

1. Việc kết hợp các kỹ thuật đối với mức toàn vẹn về an toàn phần mềm SIL 3 và SIL 4 được phê chuẩn như sau: 4, 5, 6, 8 và một từ 1 hoặc 2;

2. Việc kết hợp các kỹ thuật đối với mức toàn vẹn về an toàn phần mềm SIL 1 và SIL 2 được phê chuẩn như sau: 3, 4, 5, 6 và một từ 8, 9 hoặc 10.

3. Lập trình mê ta phải được hạn chế trong việc tạo ra mã nguồn phần mềm trước khi biên dịch.

Bảng A.5 – Thẩm tra và kiểm thử (6.2 và 7.3)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Chứng minh hình thức	D.29	-	R	R	HR	HR
2. Phân tích tĩnh	Bảng A.19	-	HR	HR	HR	HR
3. Kiểm thử và phân tích động	Bảng A.13	-	HR	HR	HR	HR
4. Đo kiểm thử Metrics	D.37	R	R	R	R	R
5. Theo dõi theo vết	D.58	R	HR	HR	M	M
6. Phân tích tác động lỗi phần mềm	D.25	-	R	R	HR	HR
7. Phạm vi kiểm thử đoạn mã	Bảng A.21	R	HR	HR	HR	HR
8. Kiểm thử chức năng và hộp đen	Bảng A.14	HR	HR	HR	M	M
9. Kiểm thử hiệu năng	Bảng A.18	-	HR	HR	HR	HR
10. Kiểm thử giao diện	D.34	HR	HR	HR	HR	HR
Các yêu cầu: 1. Đối với mức toàn vẹn về an toàn phần mềm SIL 3 và SIL 4 thì phê chuẩn việc kết hợp các kỹ thuật 3, 5, 7, 8 và một từ 1, 2 hoặc 6; 2. Đối với mức toàn vẹn về an toàn phần mềm SIL 1 và SIL 2 thì phê chuẩn việc kết hợp các kỹ thuật 5 với một từ 2, 3 hoặc 8. Chú thích 1: Kỹ thuật/Biện pháp 1, 2, 4, 5, 6 và 7 là dành cho các hoạt động thẩm tra. Chú thích 2: Kỹ thuật/Biện pháp 3, 8, 9 và 10 là dành cho các hoạt động kiểm thử						

Bảng A.6 – Tích hợp (7.6)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Kiểm thử chức năng và hộp đen	Bảng A.14	HR	HR	HR	HR	HR
2. Kiểm thử hiệu năng	Bảng A.18	-	R	R	HR	HR

Bảng A.7 – Kiểm thử tổng thể phần mềm (6.2 và 7.7)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Kiểm thử hiệu năng	Bảng A.18	-	HR	HR	M	M
2. Kiểm thử chức năng và hộp đen	Bảng A.14	HR	HR	HR	M	M
3. Lập mô hình	Bảng A.17	-	R	R	R	R
Yêu cầu: 1. Đối với mức toàn vẹn về an toàn phần mềm SIL 1 và SIL 2 thì phê chuẩn việc kết hợp các kỹ thuật 1 và 2.						

Bảng A.8 – Các kỹ thuật phân tích phần mềm (6.3)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Phân tích phần mềm tĩnh	D.13 D.37 Bảng A.19	R	HR	HR	HR	HR
2. Phân tích phần mềm động	Bảng A.13 Bảng A.14	-	R	R	HR	HR
3. Sơ đồ nguyên nhân hậu quả	D.6	R	R	R	R	R
4. Phân tích tình huống hình cây	D.22	R	R	R	R	R
5. Phân tích tác động lỗi phần mềm	D.25	R	R	R	HR	HR
Yêu cầu: 1. Một hoặc nhiều hơn những kỹ thuật này phải được sử dụng để thỏa mãn mức toàn vẹn về an toàn yêu cầu;						

Bảng A.9 – Đảm bảo chất lượng phầm mềm (6.5)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Được chứng nhận theo TCVN ISO 9001	7.1	R	HR	HR	HR	HR
2. Phù hợp với TCVN ISO 9001	7.1	M	M	M	M	M
3. Phù hợp với EN ISO 90003	7.1	R	R	R	R	R
4. Hệ thống chất lượng của tổ chức	7.1	M	M	M	M	M
5. Quản lý cấu hình phần mềm	D.48	M	M	M	M	M
6. Danh mục kiểm tra	D.7	R	HR	HR	HR	HR
7. Theo dõi theo vết	D.58	R	HR	HR	M	M
8. Ghi và phân tích dữ liệu	D.12	HR	HR	HR	M	M
Yêu cầu: 1. Bảng này phải được áp dụng cho tất cả các giai đoạn và các vai trò khác nhau.						

Bảng 10 – Bảo trì phần mềm (9.2)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Phân tích tác động	D.32	R	HR	HR	M	M
2. Ghi và phân tích dữ liệu	D.12	HR	HR	HR	M	M

Bảng 11 – Các kỹ thuật chuẩn bị dữ liệu (8.4)

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Các biện pháp chỉ dẫn dạng bảng	D.68	R	R	R	R	R
2. Ngôn ngữ lập trình chuyên dụng	D.69	R	R	R	R	R
3. Mô phỏng	D.42	R	HR	HR	HR	HR
4. Kiểm thử chức năng	D.42	M	M	M	M	M
5. Danh mục kiểm tra	D.7	R	HR	HR	M	M
6. Kiểm tra Fagan	D.23	-	R	R	R	R
7. Rà soát thiết kế hình thức	D.56	R	HR	HR	HR	HR
8. Chứng minh sự chính xác (của dữ liệu) hình thức	D.29	-	-	-	HR	HR
9. Xem xét từng bước	D.56	R	R	R	HR	HR
Các yêu cầu: 1. Đối với mức toàn vẹn về an toàn phần mềm SIL 1 và SIL 2 thì phê chuẩn việc kết hợp các kỹ thuật 1 và 4; 2. Đối với mức toàn vẹn về an toàn phần mềm SIL 3 và SIL 4 thì phê chuẩn việc kết hợp các kỹ thuật 1, 4, 5 và 7 hoặc 2, 3 và 6. Chú thích: Việc mô tả tham chiếu D.29 là trong các chương trình còn kỹ thuật 8 trong bảng này áp dụng phương pháp chứng minh sự chính xác của dữ liệu hình thức.						

A.2 Các bảng chi tiết

Bảng 12 – Các tiêu chuẩn mã hóa

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Tiêu chuẩn mã hóa	D.15	HR	HR	HR	M	M
2. Hướng dẫn phương thức mã hóa	D.15	HR	HR	HR	HR	HR
3. Các đối tượng bất biến	D.15	-	R	R	HR	HR
4. Các biến bất biến	D.15	-	R	R	HR	HR
5. Giới hạn sử dụng con trỏ	D.15	-	R	R	R	R
6. Giới hạn sử dụng vòng lặp	D.15	-	R	R	HR	HR
7. Các bước nhảy không có điều kiện	D.15	-	HR	HR	HR	HR
8. Quy mô và độ phức tạp giới hạn của các chức năng, chương trình con và các biện pháp	D.38	HR	HR	HR	HR	HR
9. Chiến lược điểm đầu vào/đầu ra cho các chức năng, chương trình con và các biện pháp	D.38	R	HR	HR	HR	HR
10. Số lượng giới hạn của các thông số chương trình con	D.38	R	R	R	R	R
11. Sử dụng giới hạn các biến chung	D.38	HR	HR	HR	M	M
Yêu cầu: 1. Chấp nhận kỹ thuật 3, 4 và có thể được coi như là một phần của trình biên dịch đã được thẩm định.						

Bảng A.13 – Phân tích và kiểm thử động

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Thực hiện kế hoạch kiểm thử từ phân tích giá trị giới hạn biên	D.4	-	HR	HR	HR	HR
2. Thực hiện kế hoạch kiểm thử từ quá trình dự đoán lỗi	D.20	R	R	R	R	R
3. Thực hiện kế hoạch kiểm thử từ quá trình tạo lỗi	D.21	-	R	R	R	R
4. Lập mô hình hiệu năng	D.39	-	R	R	HR	HR
5. Các mức tương đương và kiểm thử phân vùng đầu vào	D.18	R	R	R	HR	HR
6. Kiểm thử dựa trên cấu trúc	D.50	-	R	R	HR	HR
Yêu cầu: 1. Việc phân tích đối với các trường hợp kiểm thử là ở cấp hệ thống con và trên cơ sở chỉ dẫn kỹ thuật và/hoặc các đoạn mã.						

Bảng A.14 – Kiểm thử chức năng/hộp đen

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Thực hiện kế hoạch kiểm thử từ sơ đồ nguyên nhân hậu quả	D.6	-	-	-	R	R
2. Lập mô hình mẫu / mô phỏng	D.43	-	-	-	R	R
3. Phân tích giá trị giới hạn biên	D.4	R	HR	HR	HR	HR
4. Kiểm thử các mức tương đương và phân vùng đầu vào	D.18	R	HR	HR	HR	HR
5. Mô phỏng quá trình	D.42	R	R	R	R	R
Yêu cầu: 1. Sự hoàn chỉnh của việc mô phỏng sẽ phụ thuộc mức độ của mức toàn vẹn về an toàn phần mềm, độ phức tạp và tính ứng dụng.						

Bảng A.15 – Ngôn ngữ lập trình trong chế độ văn bản

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. ADA	D.54	R	HR	HR	HR	HR
2. MODULA-2	D.54	R	HR	HR	HR	HR
3. PASCAL	D.54	R	HR	HR	HR	HR
4. C hoặc C++	D.54 D.35	R	R	R	R	R
5. PL/M	D.54	R	R	R	NR	NR
6. BASIC	D.54	R	NR	NR	NR	NR
7. Assembler	D.54	R	R	R	R	R
8. C#	D.54 D.35	R	R	R	R	R
9. JAVA	D.54 D.35	R	R	R	R	R
10. Danh sách câu lệnh	D.54	R	R	R	R	R

Các yêu cầu:

1. Việc lựa chọn các ngôn ngữ lập trình phải dựa trên các yêu cầu được đưa ra trong mục 6.7 và 7.3.

2. Không yêu cầu điều chỉnh các quyết định được đưa ra để loại trừ các ngôn ngữ lập trình chuyên dụng.

Chú thích 1: Về việc đánh giá sự phù hợp của ngôn ngữ lập trình, xem mục D.54, “Ngôn ngữ lập trình phù hợp”.

Chú thích 2: Nếu ngôn ngữ lập trình chuyên dụng không có trong bảng, nó sẽ không bị loại bỏ một cách tự động. Tuy nhiên, nên thỏa mãn D.54.

Chú thích 3: Các hệ thống chạy thực gắn liền với các ngôn ngữ lập trình đã được lựa chọn cần thiết để chạy các chương trình ứng dụng nên được điều chỉnh việc sử dụng theo mức toàn vẹn về an toàn phần mềm.

Bảng A.16 – Ngôn ngữ dạng biểu đố cho các thuật toán ứng dụng

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Sơ đồ khối chức năng	D.63	R	R	R	R	R
2. Lược đồ hàm tuần tự	D.61	-	HR	HR	HR	HR
3. Sơ đồ bậc thang	D.62	R	R	R	R	R
4. Lược đồ trạng thái	D.64	R	HR	HR	HR	HR

Bảng A.17 – Lập mô hình

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Lập mô hình dữ liệu	D.65	R	R	R	HR	HR
2. Sơ đồ luồng dữ liệu	D.11	-	R	R	HR	HR
3. Sơ đồ luồng kiểm soát	D.66	R	R	R	HR	HR
4. Cơ chế trạng thái hữu hạn hoặc sơ đồ chuyển đổi trạng thái	D.27	-	HR	HR	HR	HR
5. Petri-Nets theo thời gian	D.55	-	R	R	HR	HR
6. Bảng so sánh logic	D.13	R	R	R	HR	HR
7. Các biện pháp hình thức	D.28	-	R	R	HR	HR
8. Lập mô hình hiệu năng	D.39	-	R	R	HR	HR
9. Lập mô hình mẫu / mô phỏng	D.43	-	R	R	R	R
10. Sơ đồ cấu trúc	D.51	-	R	R	HR	HR
11. Sơ đồ chuỗi	D.67	R	HR	HR	HR	HR
Các yêu cầu: 1. Hướng dẫn lập mô hình phải được định nghĩa và được sử dụng. 2. Tối thiểu phải lựa một một kỹ thuật HR.						

Bảng A.18 – Kiểm thử hiệu năng

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Kiểm thử Avalance	D.3	-	R	R	HR	HR
2. Ràng buộc về thời gian phản hồi và bộ nhớ	D.45	-	HR	HR	HR	HR
3. Các yêu cầu về hiệu năng	D.40	-	HR	HR	HR	HR

Bảng A.19 – Phân tích tĩnh

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Phân tích giá trị giới hạn biên	D.4	-	R	R	HR	HR
2. Danh mục kiểm tra	D.7	-	R	R	R	R
3. Phân tích luồng kiểm soát	D.8	-	HR	HR	HR	HR
4. Phân tích luồng dữ liệu	D.10	-	HR	HR	HR	HR
5. Dự đoán lỗi	D.20	-	R	R	R	R
6. Tổng duyệt/ rà soát thiết kế	D.56	HR	HR	HR	HR	HR

Bảng A.20 – Các thành phần

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Che dấu thông tin	D.33	-	-	-	-	-
2. Đóng gói thông tin	D.33	R	HR	HR	HR	HR
3. Giới hạn số lượng tham số	D.38	R	R	R	R	R
4. Giao diện được xác định đầy đủ	D.38	R	HR	HR	M	M

Yêu cầu:

1. Việc che dấu và đóng gói thông tin chỉ là khuyến nghị cao nếu không có chiến lược tổng thể để truy cập dữ liệu.

Chú thích: Kỹ thuật/Biện pháp 4 là dành cho các giao diện nội bộ.

Bảng A.21 – Phạm vi kiểm thử đoạn mã

Tiêu chí phạm vi kiểm thử	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Câu lệnh	D.50	R	HR	HR	HR	HR
2. Nhánh	D.50	-	R	R	HR	HR
3. Điều kiện phức hợp	D.50	-	R	R	HR	HR
4. Luồng dữ liệu	D.50	-	R	R	HR	HR
5. Đường chạy	D.50	-	R	R	HR	HR

Các yêu cầu:

1. Với mọi mức SIL, một thông số định lượng về phạm vi kiểm thử phải được xây dựng để việc kiểm thử được thực hiện. Việc xây dựng này có thể hỗ trợ cho việc tuyên bố về độ tin cậy đạt được và sự cần thiết phải sử dụng các kỹ thuật bổ sung.

2. Đối với phạm vi kiểm thử ở mức SIL 3 hoặc SIL 4 tại mức thành phần thì nên tính toán tuân theo:
   - 2 và 3;
   - 2 và 4;
   - 5

   Hoặc phạm vi kiểm thử tại mức tích hợp thì nên tính toán tuân theo một hoặc nhiều hơn trong số 2, 3, 4, hoặc 5.

3. Tiêu chí phạm vi kiểm thử khác có thể được sử dụng, miễn là có căn cứ. Những tiêu chí này phụ thuộc vào cấu trúc phần mềm (xem Bảng A.3) và ngôn ngữ lập trình (xem Bảng A.15 và Bảng A.16).

4. Mọi đoạn mã mà không khả thi để kiểm thử thì nên chứng minh là đúng khi sử dụng kỹ thuật phù hợp. Ví dụ, phân tích tĩnh từ Bảng A.19.

Chú thích 1: Phạm vi câu lệnh tự động đạt được bởi các yếu tố từ 2 đến 5.

Chú thích 2: Tiêu chí phạm vi kiểm thử trong bảng này được sử dụng để kiểm thử dựa trên cấu trúc (dựa trên đoạn mã, hộp trắng). Các kỹ thuật/biện pháp để kiểm thử chức năng được đưa ra trong Bảng A.14.

Chú thích 3: Một phạm vi lớn thường khó có thể đạt được. Việc sử dụng thực hiện kế hoạch kiểm thử từ giá trị biên (D.4) và kiểm thử các mức tương đương và phân vùng đầu vào (D.18) có khả năng bao phủ đầy đủ để đạt được với một số lượng các kiểm thử nhỏ hơn.

Chú thích 4: Sự khác biệt giữa 2 và 3 phụ thuộc vào thực tế mức ngôn ngữ lập trình và việc sử dụng các điều kiện phức hợp. Khi chỉ sử dụng các điều kiện độc lập, chẳng hạn như kết quả của trình biên dịch, 2 và 3 được coi là như nhau.

Bảng A.22 – Cấu trúc phần mềm định hướng đối tượng

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Truy vết khái niện phạm vi áp dụng cho các loại cấu trúc	-	R	R	R	HR	HR
2. Sử dụng các bộ khung phù hợp, việc kết hợp các loại và các mẫu thiết kế được sử dụng phổ biến	-	R	R	R	HR	HR
3. Thiết kế chi tiết định hướng đối tượng	Bảng A.23	R	R	R	HR	HR
Yêu cầu: 1. Khi sử dụng các bộ khung hiện có và các mẫu thiết kế thì các yêu cầu của phần mềm áp dụng các bộ khung và mẫu thiết kế này. Chú thích 1: Cách tiếp cận định hướng đối tượng thể hiện sự khác biệt về thông tin so với các cách tiếp cận trong chế độ văn bản, danh mục dưới đây sẽ chứa các khuyến nghị mà cần thiết phải xem xét cụ thể: - Việc hiểu các mức phân cấp, và việc nhận dạng các chức năng phần mềm mà sẽ được thực hiện dựa trên sự dẫn chứng của phương pháp đã được đưa ra (gồm cả khi sử dụng một dải phân cấp hiện có). - Kiểm thử dựa trên cấu trúc (Bảng A.13). Việc truy vết từ phạm vi áp dụng đến cấu trúc phân loại là ít quan trọng. Chú thích 2: Đối với một phần của phần mềm dự định, một bộ khung có thể tồn tại từ phần mềm trước đó mà xử lý thành công một nhiệm vụ tương tự mà người xây dựng hiểu được. Thì khuyến nghị sử dụng bộ khung đó.						

Bảng A.23 – Thiết kế chi tiết định hướng đối tượng

KỸ THUẬT/BIỆN PHÁP	Tham khảo	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Các loại chỉ nên có một đối tượng	-	R	R	R	HR	HR
2. Tính kế thừa chỉ được sử dụng nếu lớp tạo ra là sự sàng lọc của lớp cơ sở	-	R	HR	HR	HR	HR
3. Mức độ kế thừa bị giới hạn bởi các tiêu chuẩn mã hóa	-	R	R	R	HR	HR
4. (Các biện pháp) dành quyền hoạt động phải được kiểm soát nghiệm ngặt	-	R	R	R	HR	HR
5. Tính đa kế thừa chỉ được sử dụng với các lớp giao diện	-	R	HR	HR	HR	HR
6. Tính kế thừa từ các lớp ẩn	-	-	-	-	NR	NR
Các yêu cầu: 1. Từng lớp được đặc trưng bởi một trọng trách, ví dụ quan tâm đến việc dữ liệu được kết nối chặt chẽ và việc hoạt động trên trường dữ liệu đó. 2. Việc bảo dưỡng là điều cần thiết để tránh các phụ thuộc xoay vòng giữa các đối tượng.						

Phụ lục B

(Quy định)

Vai trò và trách nhiệm đối của các bên liên quan đối với phần mềm chủ chốt

Bảng B.1: Đơn vị quản lý các yêu cầu

Bảng B.2: Đơn vị thiết kế

Bảng B.3: Đơn vị thực hiện

Bảng B.4: Đơn vị kiểm thử

Bảng B.5: Đơn vị thẩm tra

Bảng B.6: Đơn vị tích hợp

Bảng B.7: Đơn vị thẩm định

Bảng B.8: Đơn vị đánh giá

Bảng B.9: Đơn vị quản lý dự án

Bảng B.10: Đơn vị quản lý cấu hình

Bảng B.1 – Vai trò và trách nhiệm của Đơn vị quản lý các yêu cầu

Vai trò: đơn vị quản lý các yêu cầu
Trách nhiệm 1. Chịu trách nhiệm quy định các yêu cầu phần mềm 2. Sở hữu Chỉ dẫn các yêu cầu phần mềm 3. Phải thiết lập và duy trì việc truy vết từ và đến các yêu cầu mức hệ thống 4. Phải đảm bảo các yêu cầu phần mềm và các chỉ dẫn được quản lý cấu hình và quản lý sự thay đổi, bao gồm trạng thái, phiên bản và tình trạng ưu tiên 5. Phải đảm bảo tính nhất quán và hoàn thiện về Chỉ dẫn các yêu cầu phần mềm (với các yêu cầu của người dùng và môi trường ứng dụng cuối cùng) 6. Phải xây dựng và duy trì các tài liệu về yêu cầu phần mềm
Năng lực chính: 1. Phải có năng lực trong lĩnh vực kỹ thuật về các yêu cầu 2. Phải có kinh nghiệm trong lĩnh vực ứng dụng 3. Phải có kinh nghiệm về các thuộc tính an toàn trong lĩnh vực ứng dụng 4. Phải hiểu vai trò tổng quan của hệ thống và môi trường ứng dụng 5. Phải hiểu về kết quả và các kỹ thuật phân tích 6. Phải hiểu về các quy chuẩn áp dụng 7. Phải hiểu các yêu cầu trong tiêu chuẩn này

Bảng B.2 – Vai trò và trách nhiệm của Đơn vị thiết kế

Vai trò: đơn vị thiết kế
Trách nhiệm 1. Phải chuyển các yêu cầu phần mềm cụ thể thành các giải pháp có thể chấp nhận được 2. Sở hữu các giải phảp tải dữ liệu xuống và cấu trúc 3. Phải xác định hoặc lựa chọn các biện pháp thiết kế và các chương trình hỗ trợ 4. Phải áp dụng các nguyên tắc và các tiêu chuẩn thiết kế phù hợp 5. Phải xây dựng các chỉ dẫn thành phần nều cần 6. Phải duy trì việc theo dõi theo vết các yêu cầu phần mềm quy định 7. Phải xây dựng và duy trì tài liệu thiết kế 8. Phải đảm bảo các tài liệu thiết kế phải được kiểm soát về cấu hình và sự thay đổi
Năng lực chính: 1. Phải có năng lực về kỹ thuật phù hợp với lĩnh vực ứng dụng 2. Phải có năng lực về các nguyên tắc thiết kế an toàn 3. Phải có năng lực về sử dụng các phương pháp phân tích thiết kế và kiểm thử thiết kế 4. Phải có khả năng làm việc trong phạm vi ràng buộc của thiết kế ở trong một môi trường nhất định 5. Phải có khả năng hiểu về phạm vi của sự cố 6. Phải hiểu về tất cả các ràng buộc bị áp đặt bởi nền phần cứng, hệ thống vận hành và các hệ thống tương giao 7. Phải hiểu các nội dung liên quan trong tiêu chuẩn này

Bảng B.3 – Vai trò và trách nhiệm của Đơn vị thực hiện

Vai trò: đơn vị thực hiện
Trách nhiệm 1. Phải chuyển các giải pháp thiết kế thành dữ liệu/mã nguồn/các sản phẩm thiết kế khác 2. Phải chuyển mã nguồn thành mã chạy /sản phẩm thiết kế khác 3. Phải áp dụng các nguyên tắc thiết kế an toàn 4. Phải áp dụng các tiêu chuẩn mã hóa/chuẩn bị dữ liệu quy định 5. Phải tiến hành phân tích để thẩm tra kết quả trung gian 6. Phải tích hợp phần mềm trên máy mục tiêu 7. Phải xây dựng và duy trì tài liệu thực hiện, bao gồm các danh mục, các kiểu dữ liệu và các biện pháp áp dụng 8. Phải duy trì việc theo dõi theo vết thiết kế 9. Phải duy trì dữ liệu/đoạn mã đã được tạo ra hoặc bổ sung sửa đổi phải được kiểm soát cấu hình và sự thay đổi
Năng lực chính: 1. Phải có năng lực về kỹ thuật phù hợp với lĩnh vực ứng dụng 2. Phải có năng lực về ngôn ngữ thực hiện và các chương trình hỗ trợ 3. Phải có khả năng áp dụng các tiêu chuẩn mã hóa và phương thức lập trình quy định 4. Phải hiểu về tất cả các ràng buộc bị áp đặt bởi nền phần cứng, hệ thống vận hành và các hệ thống tương giao 5. Phải hiểu các phần tương ứng trong tiêu chuẩn này

Bảng B.4 – Vai trò và trách nhiệm của Đơn vị kiểm thử

Vai trò: đơn vị kiểm thử
Trách nhiệm 1. Phải đảm bảo rằng hoạt động kiểm thử đã có kế hoạch 2. Phải xây dựng chỉ dẫn kiểm thử (đối tượng và các trường hợp kiểm thử) 3. Phải đảm bảo việc theo dõi theo vết các đối tượng kiểm thử trên cơ sở các yêu cầu phần mềm quy định và việc theo dõi theo vết các trường hợp kiểm thử trên cơ sở các đối tượng kiểm thử quy định 4. Phải đảm bảo thực hiện các công việc kiểm thử đã lên kế hoạch và các công việc kiểm thử quy định 5. Phải xác định các sai lệch so với kết quả dự kiến và ghi lại trong các báo cáo kiểm thử 6. Phải thông tin các sai lệch cho các cơ quan quản lý về sự thay đổi liên quan để đánh giá và ra quyết định 7. Phải nắm bắt được kết quả trong các báo cáo 8. Phải lựa chọn thiết bị kiểm thử phần mềm
Năng lực chính: 1. Phải có năng lực trong lĩnh vực thực hiện kiểm thử, ví dụ đoạn mã, dữ liệu, các yêu cầu phần mềm 2. Phải có năng lực thực hiện các biện pháp/cách tiếp cận thẩm tra và kiểm thử khác nhau và có khả năng tìm ra phương pháp phù hợp nhất trong phạm vi đã nêu 3. Phải có khả năng thực hiện các trường hợp kiểm thử từ các chỉ dẫn cho trước 4. Phải có khả năng tư duy phân tích và kỹ năng phân tích tốt 5. Phải hiểu các nội dung liên quan trong tiêu chuẩn này

Bảng B.5 – Vai trò và trách nhiệm của Đơn vị thẩm tra

Vai trò: đơn vị thẩm tra
Trách nhiệm 1. Phải xây dựng Kế hoạch thẩm tra phần mềm (có thể bao gồm các vấn đề về chất lượng) nêu rõ những nội dung gì cần thẩm tra và loại quá trình (ví dụ, rà soát, phân tích,…) và kiểm thử được yêu cầu như là bằng chứng 2. Phải kiểm tra sự đầy đủ (sự hoàn thiện, sự nhất quán, sự chính xác, sự phù hợp và khả năng truy vết) của các bằng chứng được lưu lại từ các công việc rà soát, tích hợp và kiểm thử với các đối tượng thẩm tra quy định 3. Phải nhận biết được sự bất thường, đánh giá những bất thường này bằng các quy định về rủi ro, ghi lại và thông tin những bất thường này tới các cơ quan quản lý sự thay đổi liên quan để đánh giá và đưa ra quyết định 4. Phải quản lý quá trình thẩm tra (rà soát, tích hợp và kiểm thử) và đảm bảo tính độc lập của các hoạt động như yêu cầu 5. Phải xây dựng và duy trì các bản ghi về các hoạt động thẩm tra 6. Phải xây dựng Báo cáo thẩm tra tuyên bố rõ kết quả của các hoạt động thẩm tra
Năng lực chính: 1. Phải có năng lực trong lĩnh vực thực hiện thẩm tra, ví dụ mã hóa, dữ liệu, các yêu cầu phần mềm 2. Phải có năng lực thực hiện các biện pháp/cách tiếp cận thẩm tra khác nhau và có khả năng tìm ra biện pháp hoặc việc kết hợp các biện pháp phù hợp nhất trong phạm vi đã nêu 3. Phải có khả năng thực hiện các trường hợp thẩm tra từ các chỉ dẫn cho trước 4. Phải có khả năng tư duy phân tích và kỹ năng phân tích tốt 5. Phải hiểu các phần tương ứng trong tiêu chuẩn này

Bảng B.6 – Vai trò và trách nhiệm của Đơn vị tích hợp

Vai trò: đơn vị tích hợp
Trách nhiệm 1. Phải quản lý quá trình tích hợp bằng cách sử dụng các cơ sở phần mềm 2. Phải xây dựng Chỉ dẫn kiểm thử tích hợp phần mềm và phần cứng/phần mềm đối với các thành phần phần mềm trên cơ sở cấu trúc và các chỉ dẫn thành phần của đơn vị thiết kế mà nêu rõ các thành phần đầu vào cần thiết, chuỗi các hoạt động tích hợp và các thành phần được tích hợp tổng hợp 3. Phải xây dựng và duy trì các bản ghi về các hoạt động tích hợp 4. Phải nhận biết các bất thường trong việc tích hợp, ghi lại và thông báo những bất thường này tới các cơ quan quản lý sự thay đổi liên quan để đánh giá và đưa ra quyết định 5. Phải xây dựng báo cáo tích hợp tổng quan hệ thống và thành phần, nêu rõ kết quả của việc tích hợp
Năng lực chính: 1. Phải có năng lực trong lĩnh vực thực hiện tích hợp thành phần, ví dụ mã hóa, nền tảng, dữ liệu, hệ các thống vận hành, các giao diện phần mềm, ngôn ngữ lập trình tương ứng, … 2. Phải có năng lực thực hiện các biện pháp/cách tiếp cận tích hợp khác nhau và có khả năng tìm ra biện pháp hoặc việc kết hợp các biện pháp phù hợp nhất trong phạm vi đã nêu 3. Phải có khả năng hiểu về thiết kế và chức năng yêu cầu ở các mức trung gian khác 4. Phải có khả năng thực hiện các loại kiểm thử tích hợp từ một loạt các chức năng tích hợp 5. Phải có khả năng tư duy phân tích và kỹ năng phân tích tốt hướng tới tổng quan mức hệ thống 6. Phải hiểu các nội dung liên quan trong tiêu chuẩn này

Bảng B.7 – Vai trò và trách nhiệm của Đơn vị thẩm định

Vai trò: đơn vị thẩm định

Trách nhiệm 1. Phải xây dựng cách thức hiểu hệ thống của phần mềm trong môi trường ứng dụng dự định 2. Phải xây dựng kế hoạch thẩm tra và quy định các hoạt động và nhiệm vụ quan trọng đối với công việc thẩm định phần mềm và thỏa thuận kế hoạch này với đơn vị đánh giá 3. Phải rà soát các yêu cầu phần mềm theo môi trường dự định/mục đích sử dụng 4. Phải rà soát phần mềm dựa trên các yêu cầu phần mềm để đảm bảo tất cả những yêu cầu này đều được đáp ứng đầy đủ 5. Phải đánh giá sự phù hợp của quá trình phát triển phần mềm và phần mềm được xây dựng dựa trên các yêu cầu của tiêu chuẩn này, gồm có mức SIL được phân bổ 6. Phải rà soát lại sự chính xác, sự nhất quán và sự đầy đủ của công việc thẩm tra và kiểm thử 7. Phải kiểm tra sự chính xác, sự nhất quán và sự đầy đủ của các trường hợp kiểm thử và và các kiểm thử đã được thực hiện 8. Phải đảm bảo thực hiện tất cả các hoạt động kế hoạch thẩm định 9. Phải rà soát và phân loại tất cả các sai lệch về rủi ro, ghi lại và nộp cho cơ quan có trách nhiệm để quản lý sự thay đổi và đưa ra quyết định 10. Phải đưa ra khuyến nghị về sự phù hợp của phần mềm đối với mục đích sử dụng và chỉ ra mọi ràng buộc ứng dụng nếu cần 11. Phải nắm bắt được các sai khác so với kế hoạch thẩm định 12. Phải thực hiện đánh giá, kiểm tra hoặc rà soát tổng quan dự án (như việc thuyết minh quá trình xây dựng chung) nếu cần trong các giai đoạn xây dựng khác nhau 13. Phải rà soát và phân tích các báo cáo thẩm định liên quan các ứng dụng ưu tiên nếu cần 14. Phải rà soát các giải pháp được xây dựng theo vết các yêu cầu phần mềm 15. Phải đảm bảo các sổ tay nguy hiểm liên quan và các lỗi về sự không phù hợp còn lại phải được rà soát và tất cả các nguy hiểm được loại bỏ bằng một phương thức thích hợp thông qua việc loại bỏ hoặc các biện pháp chuyển đổi/kiểm soát các rủi ro 16. Phải xây dựng báo cáo thẩm định 17. Phải đưa ra sự đồng ý/không đồng ý đối với việc phát hành phần mềm
Năng lực chính: 1. Phải có năng lực trong lĩnh vực thực hiện thẩm định 2. Phải có kinh nghiệm về các thuộc tính an toàn trong lĩnh vực ứng dụng 3. Phải có năng lực thực hiện các biện pháp/cách tiếp cận thẩm định khác nhau và có khả năng tìm ra biện pháp hoặc việc kết hợp các biện pháp phù hợp nhất trong phạm vi đã nêu 4. Phải có khả năng thực hiện các loại bằng chứng thẩm định yêu cầu từ các chỉ dẫn trước đó ảnh hưởng tới ứng dụng dự định 5. Phải có khả năng kết hợp các loại và các nguồn bằng chứng khác nhau và tổng hợp một cách tổng quan về mục đích hoặc các ràng buộc và giới hạn của ứng dụng 6. Phải có khả năng tư duy phân tích và kỹ năng phân tích tốt 7. Phải hiểu tổng quan về phần mềm và và hiểu về môi trường ứng dụng 8. Phải hiểu các phần tương ứng trong tiêu chuẩn này

Bảng B.8 – Vai trò và trách nhiệm của Đơn vị đánh giá

Vai trò: đơn vị đánh giá
Trách nhiệm 1. Phải xây dựng việc hiểu phần mềm hệ thống trong môi trường ứng dụng dự định 2. Phải xây dựng kế hoạch đánh giá và thông tin kế hoạch này cho cơ quan quản lý an toàn và tổ chức khách hàng (các tổ chức hợp đồng của đơn vị đánh giá) 3. Phải đánh giá sự phù hợp của quá trình phần mềm và phần mềm được xây dựng dựa trên các yêu cầu của tiêu chuẩn này, gồm có mức SIL được phân bổ 4. Phải đánh giá năng lực của tổ chức và đội ngũ nhân viên dự án đối với công việc phát triển phần mềm 5. Phải đánh giá các hoạt động thẩm tra, thẩm định và các bằng chứng hỗ trợ 6. Phải đánh giá hệ thống quản lý chất lượng được phê chuẩn đối với công việc phát triển phần mềm 7. Phải đánh giá hệ thống quản lý cấu hình, sự thay đổi và các bằng chứng trong việc sử dụng và ứng dụng 8. Phải xác định và đánh giá mọi sai lệch và rủi ro từ các yêu cầu phần mềm trong báo cáo đánh giá 9. Phải đảm bảo rằng kế hoạch đánh giá được thực hiện 10. Phải thực hiện đánh giá và kiểm tra an toàn về tổng thể quá trình xây dựng nếu cần tại các giai đoạn xây dựng khác nhau 11. Phải đưa ra quan điểm chuyên môn phù hợp với phần mềm được xây dựng đối với mục đích sử dụng mà nêu chi tiết mọi ràng buộc, điều kiện ứng dụng đối với công việc kiểm soát rủi ro nếu cần 12. Phải xây dựng báo cáo đánh giá và duy trì các bản ghi về quá trình đánh giá
Năng lực chính: 1. Phải có năng lực trong lĩnh vực/kỹ thuật thực hiện đánh giá 2. Phải có chứng chỉ của cơ quan quản lý an toàn có thẩm quyền 3. Phải luôn duy trì việc có được đủ kinh nghiệm cao về các nguyên tắc an toàn và việc ứng dụng các nguyên tắc này trong lĩnh vực ứng dụng 4. Phải có khả năng kiểm tra biện pháp hoặc việc kết hợp các biện pháp phù hợp nhất trong phạm vi áp dụng đã nêu 5. Phải có khả năng hiểu về quá trình quản lý chất lượng, kỹ thuật, nguồn nhân lực, an toàn tương ứng trong việc đáp ứng các yêu cầu của tiêu chuẩn này 6. Phải có khả năng sử dụng các biện pháp/cách tiếp cận đánh giá 7. Phải có khả năng tư duy phân tích và kỹ năng phân tích tốt 8. Phải có khả năng kết hợp các loại và các nguồn bằng chứng khác nhau và tổng hợp tổng quan về sự phù hợp đối với mục đích hoặc các ràng buộc và giới hạn về ứng dụng 9. Phải hiểu tổng quan về phần mềm, gồm có hiểu về môi trường ứng dụng 10. Phải có khả năng chứng minh sự đầy đủ của tất cả các quá trình xây dựng (như quá trình thẩm tra, thẩm định, quản lý cấu hình, quản lý chất lượng) 11. Phải hiểu các yêu cầu trong tiêu chuẩn này

Bảng B.9 – Vai trò và trách nhiệm của Đơn vị quản lý dự án

Vai trò: đơn vị quản lý dự án
Trách nhiệm 1. Phải đảm bảo rằng hệ thống quản lý chất lượng và sự độc lập của các bên mà tuân thủ theo mục 5.1 được xây dựng và các tiến trình thì được kiểm tra trên cơ sở các kế hoạch 2. Phải phân bổ đủ số lượng nguồn nhân lực có năng lực trong dự án để thực hiện các nhiệm vụ quan trọng, gồm các hoạt động an toàn, ảnh hưởng tới tính độc lập của các bên 3. Phải đảm bảo bổ nhiệm đơn vị thẩm định phù hợp cho dự án như được quy định trong tiêu chuẩn này 4. Phải chịu trách nhiệm chuyển giao và triển khai phần mềm và đảm bảo rằng các yêu cầu an toàn từ các bên liên quan cũng được đáp ứng đầy đủ và được chuyển giao 5. Phải cho phép có đủ thời gian để thực hiện chuẩn xác và đáp ứng các nhiệm vụ an toàn 6. Phải chứng thực một phần và toàn bộ các tài liệu chuyển giao về an toàn từ quá trình xây dựng 7. Phải đảm bảo duy trì đầy đủ việc theo dõi theo vết và các bản ghi trong việc đưa ra quyết định liên quan đến an toàn
Năng lực chính: 1. Phải hiểu các yêu cầu quản lý, tổ chức, năng lực và chất lượng trong tiêu chuẩn này 2. Phải hiểu các yêu cầu của quá trình an toàn 3. Phải có khả năng suy xét các lựa chọn khác nhau và hiểu về ảnh hưởng của quyết định đối với tính năng an toàn hoặc việc chọn các lựa chọn 4. Phải hiểu các yêu cầu trong quá trình phát triển phần mềm 5. Phải hiểu các yêu cầu trong các tiêu chuẩn liên quan khác

Bảng B.10 – Vai trò và trách nhiệm của Đơn vị quản lý cấu hình

Vai trò: đơn vị quản lý cấu hình
Trách nhiệm 1. Phải chịu trách nhiệm lên kế hoạch quản lý cấu hình phần mềm 2. Phải sở hữu hệ thống quản lý cấu hình 3. Phải thiết lập tất cả các thành phần phần mềm được xác định rõ ràng và tính độc lập được diễn giải vào trong hệ thống quản lý cấu hình 4. Phải chuẩn bị Thông tin lưu ý về phiên bản phần mềm sử dụng, bao gồm cả các phiên bản của các thành phần phần mềm không phù hợp
Năng lực chính: 1. Phải có năng lực về quản lý cấu hình phần mềm 2. Phải hiểu các yêu cầu trong tiêu chuẩn này

Phụ lục C

(Tham khảo)

Tóm tắt việc kiểm soát các tài liệu

Bảng C.1 đưa ra bản tóm tắt tài liệu

Bảng C.1 – Tóm tắt việc kiểm soát các tài liệu

Giai đoạn	Tài liệu	Quy định cho	Kiểm tra lần 1	Kiểm tra lần 2
Lập kế hoạch	1. Kế hoạch đảm bảo chất lượng phần mềm	a	VER	VAL
	2. Báo cáo thẩm tra đảm bảo chất lượng phần mềm	VER		VAL
	3. Kế hoạch quản lý cấu hình phần mềm	B.10	VER	VAL
	4. Kế hoạch thẩm tra phần mềm	VER		VAL
	5. Kế hoạch thẩm định phần mềm	VAL	VER	
Các yêu cầu phần mềm	6. Chỉ dẫn các yêu cầu phần mềm	REQ	VER	VAL
	7. Chỉ dẫn kiểm thử tổng thể phần mềm	TST	VER	VAL
	8. Báo cáo thẩm tra các yêu cầu phần mềm	VER		VAL
Cấu trúc và thiết kế	9. Chỉ dẫn cấu trúc phần mềm	DES	VER	VAL
	10. Chỉ dẫn thiết kế phần mềm	DES	VER	VAL
	11. Chỉ dẫn giao diện phần mềm	DES	VER	VAL
	12. Chỉ dẫn kiểm thử tích hợp phần mềm	INT	VER	VAL
	13. Chỉ dẫn kiểm thử tích hợp phần cứng / phần mềm	INT	VER	VAL
	14. Báo cáo thẩm tra thiết kế và cấu trúc phần mềm	VER		VAL
Thiết kế thành phần	15. Chỉ dẫn thiết kế thành phần phần mềm	DES	VER	VAL
	16. Chỉ dẫn kiểm thử thành phần phần mềm	TST	VER	VAL
	17. Báo cáo thẩm tra thiết kế thành phần phần mềm	VER		
Thực hiện và kiểm thử thành phần	18. Tải liệu hỗ trợ và mã nguồn phần mềm	IMP	VER	VAL
	19. Báo cáo thẩm tra mã nguồn phần mềm	VER		VAL
	20. Báo cáo kiểm thử thành phần phần mềm	TST	VER	VAL
Tích hợp	21. Báo cáo kiểm thử tích hợp phần mềm	INT	VER	VAL
	22. Báo cáo kiểm thử tích hợp phần cứng / phần mềm	INT	VER	VAL
	23. Báo cáo thẩm tra tích hợp phần mềm	VER		

Bảng C.1 – Tóm tắt việc kiểm soát các tài liệu *(kết thúc)*

Giai đoạn	Tài liệu	Quy định cho	Kiểm tra lần 1	Kiểm tra lần 2
Kiểm thử tổng thể phần mềm / thẩm định lần cuối	24. Báo cáo kiểm thử tổng thể phần mềm	TST	VER	VAL
	25. Báo cáo thẩm định phần mềm	VAL	VER	
	26. Báo cáo thẩm định các chương trình	a	VER	
	27. Thông tin lưu ý về phiên bản phần mềm	a	VER	VAL
Các hệ thống được cấu hình bởi các thuật toán và dữ liệu ứng dụng	28. Chỉ dẫn các yêu cầu ứng dụng	REQ	VER	VAL
	29. Kế hoạch chuẩn bị ứng dụng	REQ hoặc DES	VER	VAL
	30. Chỉ dẫn kiểm thử ứng dụng	TST	VER	VAL
	31. Cấu trúc và thiết kế ứng dụng	DES	VER	VAL
	32. Báo cáo thẩm tra chuẩn bị ứng dụng	VER		
	33. Báo cáo kiểm thử ứng dụng	TST	VER	VAL
	34. Mã nguồn của các thuật toán/dữ liệu ứng dụng	DES	VER	VAL
	35. Báo cáo thẩm tra các thuật toán/dữ liệu ứng dụng	VER		VAL
Triển khai phần mềm	36. Kế hoạch triển khai và phát hành phần mềm	a	VER	VAL
	37. Hướng dẫn triển khai phần mềm	a	VER	VAL
	38. Thông tin lưu ý về phiên bản phần mềm	a	VER	VAL
	39. Các biên bản triển khai	a	VER	VAL
	40. Báo cáo thẩm tra triển khai	VER		
Bảo trì phần mềm	41. Kế hoạch bảo trì phần mềm	a	VER	VAL
	42. Các biên bản thay đổi phần mềm	a	VER	VAL
	43. Các biên bản bảo trì phần mềm	a	VER	VAL
	44. Báo cáo thẩm tra bảo trì phần mềm	a	VER	VAL
Đánh giá phần mềm	45. Kế hoạch đánh giá phần mềm	ASR	VER	VAL
	46. Báo cáo đánh giá phần mềm	ASR	VER	
a chưa xác định rõ vai trò cụ thể				

Phụ lục D

(Tham khảo)

Danh mục các kỹ thuật

D.1 Sử dụng trí tuệ nhân tạo hiệu chỉnh lỗi

Mục đích

Để có thể phản ứng trước các nguy hiểm có thể xảy ra một cách linh hoạt bằng việc kết hợp các biện pháp, các mô hình quá trình và một số kiểu loại phân tích an toàn và độ tin cậy trong thực tế.

Mô tả

Do các quy tắc có thể được phát sinh trực tiếp từ các chỉ dẫn kỹ thuật và được kiểm tra dựa theo những quy tắc này nên các hoạt động dự đoán lỗi (tính toán các xu hướng), hiệu chỉnh lỗi, bảo trì và giám sát cụ thể có thể được hỗ trợ rất hiệu quả bằng các hệ thống dựa trên nền tảng trí tuệ nhân tạo theo nhiều kênh khác nhau của hệ thống. Bằng biện pháp tiếp cận này, có thể tránh được một cách có hiệu quả các lỗi phổ biến nhất được đưa vào trong các chỉ dẫn kỹ thuật do đã có một số các quy tắc thiết kế và thực hiện, đặc biệt khi áp dụng kết hợp các mô hình và các biện pháp theo thuộc tính chức năng hoặc theo thuộc tính mô tả.

Lựa chọn các biện pháp sao cho các lỗi có thể được sửa chữa và tác động của các hư hỏng được giảm thiểu tối đa để đáp ứng độ tin cậy và an toàn yêu cầu.

D.2 Các chương trình phân tích

Mục đích

Để thiết kế ra một chương trình theo cách có thể dễ dàng thực hiện được phân tích chương trình. Sự hoạt động của chương trình phải được kiểm thử toàn diện trên cơ sở phân tích.

Mô tả

Mục đích là để tạo ra các chương trình dễ dàng phân tích khi sử dụng các biện pháp phân tích tĩnh. Để đạt được mục đích này, phải tuân theo các quy tắc lập trình có tính cấu trúc, ví dụ:

- Thành phần luồng điều khiển nên được viết theo cấu trúc hệ thống, đó là các chuỗi, các phép lặp và lựa chọn;
- Các phần hợp thành phải nhỏ;
- Số lượng các đường chạy có thể thực hiện từ đầu đến cuối một phần hợp thành là nhỏ;
- Các mảng chương trình riêng lẻ phải được thiết kế sao cho chúng càng tách rời độc lập càng tốt;
- Sự liên quan giữa các thông số đầu vào và đầu ra nên càng đơn giản càng tốt;

- Không nên sử dụng các tính toán phức tạp làm cơ sở cho các quyết định phân nhánh và vòng lặp;

- Các quyết định phân nhánh và vòng lặp nên có liên quan đơn giản đến các thông số đầu vào của thành phần;

- Giới hạn biên giữa các loại sơ đồ khác nhau phải đơn giản.

D.3 Kiểm thử Avalanche/stress

Mục đích

Để tạo áp lực lên đối tượng kiểm thử một khối lượng công việc cao bất thường, để cho thấy đối tượng kiểm thử chịu được khối lượng công việc bình thường một cách dễ dàng.

Mô tả

Có một loạt các điều kiện kiểm thử khác nhau có thể áp dụng cho kiểm thử Avalanche. Một trong số các điều kiện đó được liệt kê dưới đây:

- Nếu làm việc trong chế độ vòng lặp để kiểm tra trạng thái (polling mode) thì đối tượng kiểm thử nhận được nhiều thay đổi đầu vào hơn trong một đơn vị thời gian so với dưới các điều kiện bình thường;

- Nếu làm việc theo yêu cầu thì số lượng các yêu cầu trong một đơn vị thời gian cho đối tượng kiểm thử sẽ tăng vượt trên các điều kiện bình thường;

- Nếu khối lượng cơ sở dữ liệu đóng vai trò quan trọng thì nó sẽ được tăng vượt quá các điều kiện bình thường;

- Các thiết bị ảnh hưởng được cài đặt ở tốc độ tối đa hoặc tốc độ thấp nhất tương ứng;

- Đối với trường hợp tới hạn, tất cả các yếu tố ảnh hưởng sẽ được cài đặt ở các điều kiện giới hạn cùng thời điểm cho tới mức có thể thực hiện được.

Dưới những điều kiện kiểm thử này, có thể đánh giá được sự hoạt động theo thời gian của đối tượng kiểm thử. Ảnh hưởng của thay đổi khối lượng công việc có thể được giám sát. Kích thước chính xác của các bộ nhớ đệm bên trong hoặc các biến số động, ngăn xếp có thể được kiểm tra.

D.4 Phân tích giá trị giới hạn biên

Mục đích

Để loại bỏ các lỗi phần mềm xuất hiện tại các giới hạn thông số hoặc các giới hạn biên.

Mô tả

Miền đầu vào của chương trình được phân thành một số loại lớp đầu vào. Các kiểm thử nên có cả các giới hạn biên và giá trị giới hạn của các loại này. Những kiểm thử này sẽ kiểm tra xem các giới hạn trong miền đầu vào của chỉ dẫn kỹ thuật có trùng với những giới hạn có trong chương trình. Việc sử dụng giá trị 0, theo hướng chuyển đổi trực tiếp hoặc gián tiếp, thường dễ bị lỗi và đòi hỏi phải chú ý đặc biệt đối với:

- Phép chia cho 0;
- Các kí tự điều khiển không in;
- Phần tử rỗng của ngăn xếp hoặc danh sách;
- Ma trận không;
- Bảng 0.

Thông thường, các giới hạn biên cho đầu vào có sự tương ứng trực tiếp đến các giới hạn cho dải đầu ra. Các trường hợp kiểm thử nên được quy định để ép các giá trị đầu ra theo các giá trị giới hạn của nó. Nếu có thể quy định một trường hợp kiểm thử làm cho đầu ra vượt quá các giá trị giới hạn biên quy định thì cũng xem xét tương tự.

Nếu đầu ra là một chuỗi dữ liệu, ví dụ như bảng in, nên đặc biệt chú ý đến các phần tử đầu, cuối và các danh sách có chứa các phần tử 0, 1 và 2.

D.5 Khôi phục lại

Mục đích

Để tạo ra sự hoạt động theo đúng chức năng trong trường hợp xảy ra một hoặc nhiều sự cố.

Mô tả

Nếu phát hiện ra một sự cố, hệ thống được đưa về trạng thái cài đặt sẵn trước đó, có tính ổn định đã được chứng minh trước đó. Phương pháp này ngầm lưu lại trạng thái cài đặt sẵn một cách thường xuyên ở các điểm kiểm soát được xác định rõ (được gọi là các checkpoint). Phương pháp này có thể được tiến hành toàn diện (đối với cơ sở dữ liệu hoàn chỉnh) hoặc tăng dần (chỉ thay đổi giữa các điểm kiểm soát). Khi đó hệ thống phải phục hồi lại những thay đổi đã xảy ra trong thời gian đó bằng việc sử dụng cách ghi chép hằng ngày (kiểm nghiệm theo vết các hoạt động), phục hồi (tất cả các tác động của những thay đổi được hủy bỏ) hoặc tương tác với bên ngoài (thủ công).

D.6 Sơ đồ nguyên nhân hậu quả

Mục đích

Để lập mô hình dưới dạng sơ đồ chuỗi các tình huống có thể phát sinh trong hệ thống như là kết quả của sự kết hợp các tình huống cơ bản.

Mô tả

Có thể được coi như là sự kết hợp của phân tích sự cố hình cây và tình huống hình cây. Bắt đầu từ một tình huống nghiêm trọng, một sơ đồ nguyên nhân hậu quả sẽ được truy vết về trước và về sau. Theo hướng ngược về trước, nó tương đương với một cây sự cố có tính huống nghiệm trọng là tình huống cao nhất được đưa ra. Theo hướng về sau, các hậu quả có thể phát sinh từ tình huống sẽ được xác định rõ. Sơ đồ có thể có các kí hiệu ở phía trên mô tả các điều kiện cho việc lan truyền dọc theo các nhánh khác nhau từ đỉnh. Cũng có thể bao gồm cả thời gian trễ. Những điều kiện này cũng có thể được mô tả trong các cây sự cố. Đường lan truyền có thể được kết hợp với các ký hiệu logic, để tạo ra sơ đồ chặt chẽ hơn. Một tập hợp các ký hiệu chuẩn được xác định để sử dụng trong các sơ đồ nguyên nhân hậu quả. Có thể sử dụng các sơ đồ để tính toán xác suất của các hậu quả nghiêm trọng nhất định.

D.7 Sử dụng danh mục kiểm tra

Mục đích

Để hỗ trợ quá trình đánh giá tất cả các lĩnh vực quan trọng của hệ thống hơn là đặt ra các yêu cầu cụ thể.

Mô tả

Một tập hợp các vấn đề sẽ được người thực hiện danh mục kiểm tra hoàn chỉnh. Nhiều vấn đề trong đó là đặc tính chung và Đơn vị đánh giá phải làm rõ những vấn đề này có phù hợp nhất với hệ thống cụ thể đang được đánh giá.

Để thích nghi với sự đa dạng mở rộng trong lĩnh vực phần mềm và các hệ thống đang được thẩm định, hầu hết các danh mục kiểm tra đều có các vấn đề có thể áp dụng cho nhiều loại hệ thống. Do đó, có những vấn đề trong danh mục kiểm tra được sử dụng lại không liên quan tới hệ thống đang xử lý và những vấn đề nên bỏ qua. Đối với một hệ thống cụ thể, đều có thể có yêu cầu để bổ sung vào danh mục kiểm tra tiêu chuẩn các vấn đề được dẫn hướng cụ thể đối với hệ thống đang được xử lý.

Trong mọi trường hợp, cần làm rõ việc sử dụng danh mục kiểm tra sẽ phụ thuộc chủ yếu vào kinh nghiệm và kết luận của người kỹ sư lựa chọn và áp dụng danh mục kiểm tra. Do đó, các quyết định được đưa ra bởi người kỹ sư, liên quan tới danh mục kiểm tra được lựa chọn, và mọi vấn đề bổ sung hoặc không cần thiết nên được ghi lại đầy đủ và được kết luận. Mục đích là để đảm bảo có thể rà soát

được việc áp dụng danh mục kiểm tra và sẽ đạt được các kết quả tương tự trừ khi sử dụng các tiêu chí khác nhau.

Ký hiệu để đánh giá trong một danh mục kiểm tra càng ngắn gọn càng tốt. Khi cần phải có một kết luận rộng hơn, nên tham chiếu đến các tài liệu bổ sung để có thể thực hiện được. Nên sử dụng Đạt, không đạt và chưa quyết định được, hoặc một số tập hợp giới hạn các ký hiệu quen thuộc để ghi lại kết quả cho từng vấn đề. Sự ngắn gọn này sẽ làm đơn giản hóa quá trình đi đến kết luận tổng quát cũng như kết quả của việc đánh giá bằng danh mục kiểm tra.

D.8 Phân tích luồng điều khiển

Mục đích

Để phát hiện ra các cấu trúc chương trình sai tiềm ẩn và sơ sài.

Mô tả

Phân tích luồng điều khiển sẽ xác định các khu vực mã chương trình nghi ngờ không hoạt động tốt. Chương trình này sẽ được phân tích để tạo ra một sơ đồ định hướng có thể được phân tích đối với:

- Mã không thể truy cập, ví dụ: các bước nhảy không có điều kiện để lại các khối mã không thể đạt tới (mã thừa);

- Mã ràng buộc, là mã được cấu trúc tốt có sơ đồ kiểm soát rút gọn qua các sự giản lược sơ đồ liên tiếp thành một node độc lập. Mã được cấu trúc sơ sài chỉ có thể được giảm bớt thành một nút chứa vài node.

D.9 Phân tích hư hỏng có nguyên nhân chung

Mục đích

Để xác định rõ các hư hỏng tiềm ẩn trong các hệ thống dự phòng hoặc các hệ thống con dự phòng mà có thể phá hỏng các lợi ích của sự dự phòng do sự xuất hiện của cùng các hư hỏng trong các bộ phận dự phòng tại cùng một thời điểm.

Mô tả

Các hệ thống máy tính dùng để kiểm soát an toàn của một cơ sở thường sử dụng dự phòng về phần cứng và các bộ phận được coi là chiếm đa số. Kỹ thuật này được sử dụng để tránh các hư hỏng thành phần ngẫu nhiên mà có xu hướng chống lại quá trình xử lý dữ liệu chính xác trong hệ thống máy tính.

Tuy nhiên, một số hư hỏng có thể là giống nhau đối với nhiều hơn một thành phần. Ví dụ: nếu một hệ thống máy tính được lắp đặt trong một phòng độc lập, các sai sót trong hệ thống điều hòa có thể giảm

hiệu quả của sự dự phòng. Điều tương tự cũng đúng đối với các tác động bên ngoài khác lên hệ thống máy tính như: hỏa hoạn, lụt lội, nhiễu loạn điện từ, tai nạn máy bay và động đất. Hệ thống máy tính cũng có thể bị ảnh hưởng bởi các sự cố liên quan tới vận hành và bảo trì. Do đó, việc đưa ra các quy trình phù hợp và được lưu trữ tốt cho quá trình vận hành và bảo trì sẽ là rất quan trọng. Việc đào tạo mở rộng cho nhân viên vận hành và bảo trì cũng rất quan trọng.

Các tác động từ bên trong cũng góp phần quan trọng tới các hư hỏng có nguyên nhân chung (CCF – common cause failures). Chúng có thể nảy sinh từ các lỗi thiết kế trong các thành phần chung hoặc giống nhau và các giao diện của chúng, cũng như khi thành phần bị xuống cấp do tuổi thọ. Phân tích CCF phải tìm kiếm trong hệ thống những hư hỏng chung tiềm ẩn. Các biện pháp phân tích CCF gồm kiểm soát chất lượng chung, rà soát thiết kế, thẩm tra và kiểm thử bằng một nhóm độc lập và phân tích các sự cố thực từ kinh nghiệm của những hệ thống tương tự. Tuy nhiên, phạm vi của phân tích sẽ vượt ra khỏi phần cứng. Thậm chí nếu sử dụng “phần mềm khác nhau” trong chuỗi xử lý phức tạp của hệ thống máy tính dự phòng, sẽ có thể có một số sự tương đồng trong các cách tiếp cận phần mềm mà có thể dẫn đến CCF. Ví dụ như các lỗi trong chỉ dẫn kỹ thuật chung.

Khi các CCF không xảy ra chính xác tại cùng một thời điểm, có thể đưa ra các cảnh báo bằng các biện pháp so sánh giữa các chuỗi dự phòng mà giúp phát hiện ra hư hỏng trước khi hư hỏng này tác động chung cho tất cả các chuỗi xử lý. Phân tích CCF nên tính tới kỹ thuật này.

D.10 Phân tích luồng dữ liệu

Mục đích

Để phát hiện ra các cấu trúc chương trình sai tiềm ẩn và sơ sài.

Mô tả

Phân tích luồng dữ liệu sẽ kết hợp các thông tin thu được từ phân tích luồng điều khiển cùng với các thông tin về các biến được đọc hoặc được gán giá trị trong các phần mã khác nhau. Phân tích có thể kiểm tra:

- Các biến được đọc trước khi được gán giá trí. Chúng rất có khả năng bị lỗi và sẽ làm cho hoạt động lập trình có lỗi nhất định;

- Các biến được gán giá trị nhiều hơn một lần không được đọc. Đây có thể là đoạn mã bị bỏ qua;

- Các biến được gán giá trị nhưng không bao giờ được đọc. Có thể là đoạn mã thừa.

Đây là sự mở rộng đối với phân tích luồng dữ liệu được biết đến như là phân tích luồng thông tin, khi các luồng dữ liệu thực tế (cả trong và giữa các quy trình) được so sánh với dự định thiết kế. Nó được thực hiện bình thường bằng chương trình tính toán khi các luồng dữ liệu dự định được xác định sử dụng diễn giải có cấu trúc đọc được bằng chương trình.

D.11 Sơ đồ luồng dữ liệu

Mục đích

Để mô tả luồng dữ liệu qua một chương trình dưới dạng sơ đồ.

Mô tả

Các sơ đồ luồng dữ liệu sẽ ghi lại cách thức các đầu vào dữ liệu được chuyển đổi thành đầu ra, trong đó từng giai đoạn trong sơ đồ thể hiện một sự chuyển đổi riêng biệt.

Các thành phần cơ bản của sơ đồ luồng dữ liệu bao gồm:

- Các chức năng, được thể hiện bằng các các ô;
- Các luồng dữ liệu, được thể hiện bằng các mũi tên;
- Các kho dữ liệu, được thể hiện bằng các hình hộp mở;
- Đầu vào/đầu ra, được thể hiện bằng các loại hình hộp đặc biệt.

Các sơ đồ luồng dữ liệu sẽ mô tả cách thức đầu vào được chuyển thành đầu ra. Không và không nên bao gồm thông tin điều khiển hoặc thông tin theo chuỗi. Mỗi ô diễn giải có thể được coi như là một hộp đen đứng độc lập chuyển đổi các đầu vào ngay khi chúng sẵn sàng thành các đầu ra.

Một trong các ưu điểm cơ bản của các sơ đồ luồng dữ liệu là chúng thể hiện các chuyển đổi mà không cần phải đưa ra bất kì giả thiết nào về cách thức chúng được thực hiện.

Việc chuẩn bị các sơ đồ luồng dữ liệu được tiếp cận tốt nhất bằng cách xem xét các đầu vào hệ thống và làm việc hướng đến các đầu ra hệ thống. Mỗi ô diễn giải phải thể hiện một sự chuyển đổi riêng biệt – đầu ra của nó nên khác với đầu vào của nó theo một cách nào đó. Không có quy luật cho việc xác định cấu trúc tổng quát của sơ đồ và việc lập cấu trúc sơ đồ luồng dữ liệu sẽ là một trong các lĩnh vực sáng tạo của thiết kế hệ thống. Giống như tất cả các thiết kế, đây là một quá trình lặp đi lặp lại cần sớm được cải tiến trong các giai đoạn để tạo ra sơ đồ cuối cùng.

D.12 Ghi và phân tích dữ liệu

Mục đích

Để thuận lợi cho việc cải tiến quá trình phần mềm bằng cách ghi lại, thẩm định và phân tích dữ liệu liên quan từ vấn đề nhân sự và các dự án riêng biệt. Việc liên quan của dữ liệu được xác định bằng các

mục tiêu chiến lược của tổ chức. Các mục tiêu này có thể hướng tới việc đánh giá một biện pháp phát triển phần mềm cụ thể so với các tuyên bố về nó, ví dụ, về hiệu quả của việc ngăn ngừa thiếu sót.

Mô tả

Việc ghi và phân tích dữ liệu là một phần quan trọng trong việc cải tiến quá trình phần mềm. Việc ghi dữ liệu có giá trị thể hiện một phần quan trọng trong việc hiểu hơn về quá trình phát triển phần mềm và để đánh giá các biện pháp phát triển phần mềm thay thế.

Các bản ghi chi tiết được duy trì trong suốt dự án, cả trong dự án và cả trong từng giai đoạn. Ví dụ: một kỹ sư sẽ được yêu cầu để duy trì các bản ghi có thể có:

- Nỗ lực chi tiết hóa trên các thành phần riêng biệt;
- Việc kiểm thử được thực hiện trên từng thành phần;
- Các quyết định và căn cứ cơ sở của nó;
- Quá trình đạt được các cột mốc trong dự án;
- Các vấn đề và các giải pháp.

Trong suốt và tại thời điểm kết luận dự án, những bản ghi này có thể được phân tích để tạo ra các thông tin đa dạng mở rộng. Đặc biệt, việc ghi lại dữ liệu là rất quan trọng trong việc bảo trì các hệ thống máy tính khi các kỹ sư bảo trì không lúc nào cũng biết được cơ sở căn cứ để thực hiện những quyết định xác định này trong suốt quá trình xây dựng dự án.

Do việc lập kế hoạch sơ sài, việc ghi dữ liệu có xu hướng bị quá dung lượng và bị mờ không rõ nét. Việc này có thể tránh được bằng cách tuân theo nguyên tắc việc ghi dữ liệu nên được định hướng bởi mục tiêu, các câu hỏi, các số đo liên quan đến những yếu tố quan trọng chiến lược của tổ chức.

Để đạt được độ chính xác mong muốn, quá trình ghi và thẩm định dữ liệu nên được tiến hành đồng thời với quá trình phát triển, ví dụ như là một phần của quá trình kiểm soát cấu hình.

D.13 Bảng so sánh logic (bảng thực tế)

Mục đích

Để đưa ra chỉ dẫn kỹ thuật và phân tích rõ ràng, chặt chẽ về sự kết hợp và mối liên hệ logic phức tạp.

Mô tả

Các bảng này liên quan tới các biện pháp sử dụng 2 bảng thông số để mô tả ngắn gọn các mối liên hệ logic giữa các biến toán tử Boolean.

Sự ngắn gọn và đặc tính dạng bảng của cả hai phương pháp sẽ phù hợp để phân tích các kết hợp logic phức tạp được thể hiện trong các đoạn mã.

Cả hai phương pháp trên đều có khả năng thực hiện được nếu sử dụng như là các chỉ dẫn kỹ thuật.

D.14 Lập trình phòng thủ

Mục đích

Để tạo ra các chương trình phát hiện ra các luồng điều khiển, luồng dữ liệu hoặc các giá trị dữ liệu bất thường trong suốt quá trình chạy chương trình và phản ứng lại các chương trình theo một phương thức được xác định trước và có thể chấp nhận được.

Mô tả

Nhiều kỹ thuật có thể được sử dụng trong suốt quá trình lập trình để kiểm tra các sự cố bất thường về dữ liệu và điều khiển. Những kỹ thuật này có thể được áp dụng một cách có hệ thống trong suốt quá trình lập trình một hệ thống để giảm thiểu khả năng xử lý dữ liệu bị lỗi.

Có thể xác định rõ hai khu vực trùng nhau của các kỹ thuật phòng thủ. Phần mềm an toàn khi xảy ra lỗi bản chất được thiết kế để thích nghi với các thiếu sót về mặt thiết kế của nó. Những thiếu sót này có thể do lỗi đơn giản trong thiết kế hoặc mã hóa, hoặc do các yêu cầu bị lỗi. Một số kỹ thuật phòng thủ được liệt kê ra dưới đây:

- Các biến nên được kiểm tra theo dải;
- Nếu cần, các giá trị nên được kiểm tra về sự hợp lý;
- Các thông số cho các quy trình nên được kiểm tra về kiểu loại, đơn vị đo và dải giá trị ở đầu vào quy trình.

Ba khuyến nghị đầu tiên này sẽ giúp đảm bảo các biến được điều khiển là hợp lý cả về mặt chức năng chương trình và mức độ vật lý của biến.

Các thông số chỉ được phép đọc (read-only) và được phép đọc - ghi (read-write) nên được tách riêng biệt và việc truy cập chúng cũng nên được kiểm tra. Các chức năng nên coi tất cả các thông số là chỉ được phép đọc. Các hằng số bằng chữ không nên có khả năng ghi. Việc này sẽ giúp phát hiện các lỗi ghi đè (overwriting) hoặc các lỗi về sử dụng biến.

Phần mềm chấp nhận lỗi được thiết kế cho các hư hỏng đã được xác định trong môi trường của chính nó hoặc sử dụng ngoài phạm vi chính thống hoặc các điều kiện được chờ đợi, và xử lý theo yêu cầu được xác định trước. Kỹ thuật này bao gồm:

- Các biến đầu vào và các biến trung gian với giá trị vật lý nên được kiểm tra về sự hợp lý của nó;

- Tác động của các biến đầu ra nên được kiểm tra, ưu tiên bằng phương pháp giám sát trực tiếp các thay đổi tình trạng hệ thống liên quan;

- Phần mềm nên kiểm tra cấu hình của chính nó. Có thể bao gồm cả kiểm tra sự tồn tại và khả năng truy cập phần cứng kỳ vọng, đồng thời cả sự hoàn thiện của chính phần mềm. Đặc biệt quan trọng đối với tính toàn vẹn về khả năng bảo trì sau các quy trình bảo trì.

Một số các kỹ thuật lập trình phòng thủ như kiểm tra chuỗi luồng điều khiển, cũng như xử lý các hư hỏng từ bên ngoài.

D.15 Tiêu chuẩn mã hóa và hướng dẫn phương thức mã hóa

Mục đích

Để đảm bảo sự sắp xếp đồng bộ các tài liệu thiết kế và đoạn mã được tạo ra, để tập trung vào lập trình thống nhất và để tập trung vào một biện pháp thiết kế tiêu chuẩn để tránh xảy ra các lỗi.

Mô tả

Các tiêu chuẩn mã hóa chính là các quy tắc và các hạn chế về ngôn ngữ lập trình đã biết để tránh các sự cố có khả năng xảy ra khi sử dụng ngôn ngữ lập trình đó.

Nội dung của tiêu chuẩn mã hóa nên có:

- Sự điều chỉnh về ngôn ngữ lập trình;

- Phạm vi và tiêu chuẩn cơ sở khi sẵn có;

Chú thích: Đối với ngôn ngữ lập trình chuyên dụng thì các tiêu chuẩn cơ sở có khả năng không khả dụng.

- Quy trình thay đổi tiêu chuẩn mã hóa;

- Phân tích các sự cố tiềm ẩn và cách xử lý khuyến nghị;

- Các hạn chế để tránh sự cố;

- Tính linh động.

Hướng dẫn phương thức mã hóa xử lý các vấn đề chẳng hạn như việc định dạng và đặt tên các quy ước, và mặc dù mang tính chủ quan cao, cao hơn bất cứ phương thức nào ảnh hưởng tới khả năng đọc đoạn mã. Việc thiết lập một phương thức thống nhất và chung cho một dự án sẽ tạo điều kiện thuận lợi cho việc hiểu và bảo trì các đoạn mã được phát triển từ nhiều hơn một lập trình viên, và sẽ dễ dàng đối với với những người muốn hợp tác trong việc phát triển các chương trình tương tự.

D.16 Lập trình đa chiều

Mục đích

Phát hiện và bảo vệ các sự cố thiết kế phần mềm còn lại trong quá trình thực hiện lập trình, để ngăn chặn các hư hỏng nghiêm trọng liên quan tới an toàn của hệ thống và để duy trì hoạt động với độ tin cậy cao.

Mô tả

Trong lập trình đa chiều, một thông số chương trình cho trước được xử lý N lần theo các cách khác nhau. Các giá trị đầu vào giống nhau sẽ cho ra N lần chạy, và các kết quả tạo ra từ N lần chạy được so sánh với nhau. Nếu các kết quả được coi là đúng, kết quả sẽ được truyền tới các đầu ra của máy tính.

Các lần chạy N có thể chạy song song trên các máy tính riêng biệt, tất cả các phiên bản thay thế có thể chạy trên cùng một máy tính và các kết quả sẽ phụ thuộc vào tính toán bên trong. Các chiến lược tính toán khác nhau có thể được sử dụng trên N phiên bản sẽ phụ thuộc vào các yêu cầu khai thác ứng dụng.

- Nếu hệ thống có một trạng thái an toàn, thì có thể thực hiện được các yêu cầu thỏa thuận hoàn chỉnh (tất cả N lần đều được đồng ý), hoặc sử dụng giá trị đầu ra an toàn sự cố. Đối với các hệ thống đóng ngắt đơn giản, việc tính toán có thể thiên về hướng an toàn. Trong trường hợp này, các hoạt động liên quan tới an toàn sẽ ngắt ra nếu một trong hai phiên bản yêu cầu tác động ngắt. Cách tiếp cận này chủ yếu sử dụng chỉ với 2 phiên bản (N=2);

- Đối với các hệ thống không có trạng thái an toàn, có thể sử dụng các hướng tính toán theo đa số. Đối với các trường hợp không có thỏa thuận tổng hợp, có thể sử dụng các phương pháp xác suất để tăng tối đa cơ hội lựa chọn giá trị đúng, ví dụ như lấy giá trị trung bình, dừng tạm thời đầu ra cho đến khi đạt được yêu cầu…

Kỹ thuật này không loại bỏ hết các sự cố thiết kế phần mềm còn lại, nhưng đưa ra một biện pháp để phát hiện và bảo vệ trước khi nó ảnh hưởng tới an toàn.

Thật không may, các kinh nghiệm và các nghiên cứu phân tích cho thấy lập trình phiên bản thứ N cũng không hiệu quả như mong muốn. Thậm chí nếu sử dụng các thuật toán khác nhau, thì các phiên bản phần mềm đa chiều cũng thường lỗi về các đầu vào tương tự.

Hai lựa chọn thay thế cho lập trình phiên bản thứ N là đa thiết kế và đa chức năng. Đa thiết kế liên quan đến việc sử dụng đa thành phần, từng thành phần được thiết kế theo một phương thức khác nhau nhưng lại thực hiện chức năng tương tự. Đa chức năng liên quan đến việc xử lý các vấn đề tương tự theo các phương thức khác nhau về chức năng. Không kể tới các biện pháp tiếp cận trên, thì không có biện pháp hiệu quả hiện đang sẵn có nào để đánh giá mức đa dạng này.

D.17 Tái cấu hình động

Mục đích

Để duy trì chức năng hệ thống dù có sự cố ở bên trong.

Mô tả

Cấu trúc logic của hệ thống phải sao cho có thể được sắp xếp thành tập hợp con các nguồn sẵn có của hệ thống. Cấu trúc cần phải có khả năng phát hiện ra hư hỏng trong các nguồn vật lý và sau đó sẽ sắp xếp lại cấu trúc logic thành các nguồn bị giới hạn mất chức năng. Mặc dù ý tưởng thường giới hạn thành sự phục hồi các đơn vị phần cứng bị hỏng, nhưng cũng có thể áp dụng đối với các đơn vị phần mềm bị hỏng nếu có đủ “Thời gian vận hành dư” cho phép thử lại phần mềm hoặc nếu có đủ dữ liệu dư để làm cho các hư hỏng ít quan trọng trở nên độc lập và tách biệt.

Mặc dù thường được áp dụng cho phần cứng, kỹ thuật này đang được xây dựng để áp dụng cho phần mềm, và từ đó là toàn bộ hệ thống. Nó phải được xem xét tại giai đoạn thiết kế hệ thống đầu tiên.

D.18 kiểm thử phân vùng đầu vào và các mức tương đương

Mục đích

Để kiểm thử sự phù hợp của phần mềm bằng cách sử dụng ít dữ liệu kiểm thử nhất. Dữ liệu kiểm thử sẽ được lấy bằng cách lựa chọn các phân vùng trong phạm vi đầu vào được yêu cầu để thử hoạt động phần mềm.

Mô tả

Kế hoạch kiểm thử này dựa trên mối liên quan tương ứng giữa các đầu vào mà xác định ra một phân vùng trong phạm vi đầu vào để kiểm thử.

Lựa chọn các trường hợp kiểm thử với mục đích là bao quát tất cả các tập hợp con của phân vùng này. Ít nhất tiến hành một trường hợp kiểm thử ở từng mức tương đương.

Có 2 khả năng cơ bản để phân vùng đầu vào:

- Các mức tương đương có thể được xác định trong chỉ dẫn kỹ thuật. Việc diễn giải chỉ dẫn kỹ thuật có thể hoặc là đầu vào được định hướng, ví dụ: các giá trị được lựa chọn sẽ được xử lý theo cùng phương thức, hoặc định hướng đầu ra, ví dụ: tập hợp các giá trị dẫn đến kết quả chức năng giống nhau;

- Các mức tương đương có thể được xác định ở cấu trúc bên trong chương trình. Trong trường hợp này, các kết quả mức tương đương được xác định từ phân tích chương trình tĩnh, ví dụ như tập hợp các giá trị dẫn đến thực hiện cùng chương trình chạy.

D.19 Mã phát hiện và sửa chữa lỗi

Mục đích

Để phát hiện và sửa chữa các lỗi trong các thông tin nhạy cảm.

Mô tả

Đối với thông tin chứa *n* bít, một khối mã *k* bit được tạo ra cho phép phát hiện các lỗi và sửa chữa các lỗi. Các loại mã khác nhau bao gồm:

- Mã sửa lỗi tuyến tính;
- Mã vòng;
- Mã đa thức;
- Mã băm;
- Mã mã hóa thông tin.

D.20 Dự đoán lỗi

Mô tả

Để loại bỏ các lỗi lập trình phổ biến.

Mô tả

Kinh nghiệm và trực giác kiểm thử kết hợp với sự hiểu biết và sự tìm hiểu về hệ thống được kiểm thử có thể bổ sung thêm một số các trường hợp kiểm thử không được xếp loại vào trong tập hợp các trường hợp kiểm thử được thiết kế từ ban đầu. Các giá trị đặc biệt hoặc sự kết hợp của các giá trị có thể dễ gây ra lỗi. Có thể tìm ra một số trường hợp kiểm thử thú vị từ danh mục kiểm tra. Cũng có thể cân nhắc liệu hệ thống có đủ mạnh. Các nút có thể được ấn hoạt động ở bảng điều khiển trước quá nhanh hoặc quá thường xuyên? Điều gì sẽ xảy ra nếu 2 nút được ấn cùng một lúc?

D.21 Tạo lỗi

Mô tả

Để xác minh liệu tập hợp các trường hợp kiểm thử có phù hợp.

Mô tả

Một số loại lỗi đã biết được đưa vào trong chương trình, và chương trình sẽ thực hiện các trường hợp kiểm thử dưới các điều kiện kiểm thử. Nếu chỉ tìm thấy một số lỗi đã tạo ra thì trường hợp kiểm thử này sẽ không phù hợp. Tỉ lệ tìm thấy các lỗi được tạo so với tổng số lượng các lỗi đã tạo bằng với tỉ lệ các lỗi thực tế được tìm thấy so với tổng số các lỗi. Việc này có thể đánh giá số lượng lỗi còn lại và từ đó là công việc kiểm thử còn lại.

$$\frac{\text{số lượng các lỗi đã tạo được tìm thấy}}{\text{tổng số các lỗi đã tạo ra}} = \frac{\text{số lượng lỗi thực được tìm thấy}}{\text{số lượng các lỗi thực}}$$

Việc phát hiện ra tất cả các lỗi được tạo ra có thể chỉ ra rằng hoặc trường hợp kiểm thử là phù hợp hoặc các lỗi được tạo ra là quá dễ để tìm thấy. Các giới hạn cho phương pháp này là để đạt được các kết quả có thể sử dụng, các loại lỗi cũng như các vị trí tạo lỗi phải tương ứng với sự phân phối thống kê các lỗi thực tế.

D.22 Phân tích tình huống hình cây

Mục đích

Để lập mô hình theo dạng sơ đồ chuỗi các tình huống có thể phát triển trong hệ thống sau khi bắt đầu một tình huống, và từ đó chỉ ra cách thức các hậu quả nghiêm trọng có thể xảy ra.

Mô tả

Ở trên đỉnh của sơ đồ sẽ quy định các điều kiện chuỗi tính huống xảy ra sau đó mà liên quan tới quá trình phát triển của cây sau khi bắt đầu tình huống là mục tiêu của phép phân tích. Bắt đầu từ tình huống đầu tiên, kẻ một đường đến điều kiện đầu tiên trong chuỗi đó. Ở đây, sơ đồ sẽ rẽ thành hai nhánh “yes” và “no”, mô tả cách thức tương lai sẽ diễn ra phụ thuộc vào điều kiện. Đối với những nhánh này, tiếp tục vẽ một đường đến điều kiện tiếp theo tương tự. Tuy nhiên, không phải tất cả các điều kiện đều liên quan tới tất cả các nhánh. Một nhánh sẽ đi đến cuối chuỗi tình huống và mỗi nhánh của cây được cấu trúc theo phương pháp này sẽ thể hiện một hậu quả có thể xảy ra. Cây tình huống có thể được sử dụng để tính toán xác suất của các hậu quả khác nhau dựa trên xác suất và số lượng các điều kiện trong chuỗi tình huống.

D.23 Kiểm tra Fagan

Mục đích

Để phát hiện ra các lỗi trong tất cả các giai đoạn của quá trình xây dựng chương trình.

Mô tả

Thực hiện đánh giá "hình thức" về các tài liệu đảm bảo chất lượng nhằm tìm ra các lỗi và các thiếu sót. Quá trình kiểm tra sẽ bao gồm 5 giai đoạn: lập kế hoạch, chuẩn bị, kiểm tra, thực hiện lại và theo dõi. Mỗi giai đoạn trên đều có mục đích riêng của nó. Quá trình xây dựng hệ thống hoàn chỉnh phải được kiểm tra (chỉ dẫn kỹ thuật, thiết kế, mã hóa và kiểm thử).

D.24 Lập trình xác nhận hư hỏng

Mục đích

Để phát hiện ra các lỗi thiết kế phần mềm còn sót lại trong quá trình chạy chương tình phần mềm.

Mô tả

Phương pháp lập trình xác nhận sẽ tuân theo ý tưởng kiểm tra điều kiện cần (trước khi chuỗi các câu lệnh được thực hiện, các điều kiện ban đầu sẽ được kiểm tra về giá trị hiệu lực) và điều kiện đủ (các kết quả được kiểm tra sau khi thực hiện chuỗi các câu lệnh). Nếu điều kiện cần hoặc điều kiện đủ không được đáp ứng, quá trình xử lý sẽ dừng lại với một lỗi.

D.25 SEEA – Phân tích tác động lỗi phần mềm

Mục đích

Để xác định các thành phần phần mềm, mức độ quan trọng của nó, để đưa ra các phương pháp phát hiện các lỗi phần mềm và tăng cường độ mạnh cho phần mềm; để đánh giá độ chính xác cần thiết trên các thành phần phần mềm khác nhau.

Mô tả

Phân tích được thực hiện theo 3 giai đoạn:

- Xác định các thành phần phần mềm chính.

Xác định mức độ phân tích (ở mức độ dòng lệnh đơn giản, tập hợp các lệnh, một thành phần…) cần thiết cho từng thành phần phần mềm từ chỉ dẫn kỹ thuật của nó.

- Phân tích lỗi phần mềm

Kết quả của giai đoạn này là một bảng liệt kê những thông tin sau:

- Tên thành phần;
- Lỗi được xem xét;
- Các hậu quả của lỗi ở cấp module;
- Các hậu quả ở cấp hệ thống;
- Chỉ tiêu an toàn bắt buộc;
- Mức nghiêm trọng của lỗi;
- Các biện pháp phát hiện lỗi được đề xuất;
- Chỉ tiêu bắt buộc nếu thực hiện các biện pháp phát hiện lỗi được đề xuất;
- Mức nghiêm trọng còn lại nếu thực hiện các biện pháp phát hiện lỗi được đề xuất;

- Tổng hợp.

Quá trình tổng hợp sẽ xác định các tình huống không an toàn còn lại và công việc thẩm định cần thiết cho mức độ nghiêm trọng của từng module.

SEEA là sự phân tích theo chiều sâu được tiến hành bởi một nhóm độc lập, là một phương pháp tìm kiếm lỗi mạnh.

D.26 Chuẩn đoán và xử lý sự cố

Mục đích

Để phát hiện ra các sự cố trong hệ thống có thể dẫn đến hư hỏng, từ đó đưa ra cơ sở cho biện pháp xử lý để giảm thiểu tối đa hậu quả của hư hỏng.

Mô tả

Việc phát hiện sự cố là quá trình kiểm tra các tình trạng có lỗi trong hệ thống (do một sự cố trong hệ thống (hệ thống con) tạo ra được kiểm tra như giải thích trước đó). Mục đích chính của việc phát hiện sự cố là ngăn chặn tác động của các kết quả sai. Một hệ thống đưa ra các kết quả đúng hoặc không có kết quả nào được gọi là tự kiểm tra “self-checking”.

Việc phát hiện sự cố dựa trên các nguyên tắc ràng buộc (chủ yếu để phát hiện các sự cố phần cứng) và sự khác nhau (các lỗi phần mềm). Một số loại hướng tính toán cần thiết để quyết định tính chính xác của các kết quả. Các biện pháp đặc biệt có thể áp dụng là lập trình xác nhận, lập trình giá trị N và kỹ thuật túi an toàn và đối với phần cứng bằng cách đưa ra các cảm biến, các vòng lặp điều khiển, các mã kiểm tra lỗi…

Có thể phát hiện sự cố bằng cách kiểm tra trong miền giá trị hoặc trong phạm vi thời gian ở các mức khác nhau, đặc biệt về mặt vật lý (nhiệt độ, hiệu điện thế…), logic (mã phát hiện lỗi), chức năng (các xác nhận) hoặc mức độ từ bên ngoài (các kiểm tra tính khả thi). Các kết quả của những kiểm tra này có thể được lưu lại và được kết hợp với các dữ liệu liên quan để cho phép kiểm tra theo dõi hư hỏng.

Các hệ thống phức tạp được cấu thành từ các hệ thống con. Sự hiệu quả của việc phát hiện, chuẩn đoán và khắc phục sự cố sẽ phụ thuộc vào độ phức tạp của tương tác giữa các hệ thống con mà làm ảnh hưởng tới quá trình lan truyền sự cố.

Việc chuẩn đoán sự cố sẽ cô lập hệ thống con nhỏ nhất có thể được xác định. Các hệ thống con nhỏ hơn cho phép chuẩn đoán chi tiết các sự cố (nhận dạng các tình trạng bị lỗi).

D.27 Cơ chế chuyển đổi trạng thái hữu hạn / sơ đồ chuyển đổi trạng thái

Mục đích

Để xác định hoặc thực hiện cấu trúc điều khiển hệ thống.

Mô tả

Nhiều hệ thống có thể được xác định theo tình trạng, đầu vào và các hoạt động của chúng. Do đó khi ở trạng thái S1 đang nhận đầu vào I, một hệ thống có thể tiến hành hoạt động A và chuyển sang trạng thái S2. Bằng cách xác định các hoạt động của hệ thống cho từng đầu vào ở từng trạng thái, chúng ta có thể xác định được một hệ thống hoàn chỉnh. Mô hình hệ thống tạo ra được gọi là cơ chế chuyển đổi trạng thái hữu hạn - Finite state machine (FSM). Nó cũng thường được vẽ dưới dạng một sơ đồ chuyển đổi trạng thái thể hiện cách thức hệ thống đi từ trạng thái này sang trạng thái khác, hoặc là một ma trận có các kích thước thể hiện tình trạng, đầu vào và các cột ma trận chứa các hoạt động và sẽ tạo ra tình trạng mới từ sự tiếp nhận đầu vào của tình trạng đã biết.

Khi một hệ thống là phức tạp hoặc có cấu trúc tự nhiên, phương pháp này có thể được thể hiện trong một cơ chế chuyển đổi trạng thái hữu hạn (FSM) được phân lớp.

Một quy định kỹ thuật hoặc một thiết kế, được thể hiện là một FSM có thể được kiểm tra về tính hoàn thiện (hệ thống phải hoạt động tốt và tình trạng mới cho từng đầu vào theo mỗi tình trạng), tính thống nhất (chỉ duy nhất một thay đổi trạng thái được xác định cho từng trạng thái / cặp đầu vào) và khả năng tiếp cận (liệu có thể chuyển từ trạng thái này sang trạng thái khác bằng bất kì chuỗi đầu vào nào không). Đây là những đặc tính quan trọng đối với các hệ thống chính và có thể được kiểm tra. Các chương trình để hỗ trợ những kiểm tra này có thể viết ra dễ dàng. Thuật toán cũng cho phép phát sinh tự động các tình huống kiểm thử để thẩm tra quá trình thực hiện FSM hoặc để xây dựng một mô hình FSM.

D.28 Các biện pháp hình thức

Mục đích

Các biện pháp hình thức có liên quan tới các chương trình và các kỹ thuật khắt khe về mặt toán học đối với chỉ dẫn kỹ thuật, thiết kế, thẩm tra các hệ thống phần mềm và phần cứng.

Mô tả

Khắt khe về mặt toán học có nghĩa là các chỉ dẫn kỹ thuật được sử dụng theo các biện pháp hình thức chính là các câu lệnh thức chuẩn về mặt logic toán học và các công việc thẩm tra hình thức chính là các kết luật chặt chẽ có tính logic. Giá trị của các biện pháp hình thức là cung cấp một phương thức để kiểm tra tượng trưng toàn bộ không gian trạng thái của một thiết kế kỹ thuật số (hoặc phần cứng hoặc phần mềm) và thiết lập một sự hoạt động chính xác hoặc đặc tính an toàn chuẩn cho tất cả các đầu vào có thể. Tuy nhiên, việc này hiếm khi được thực hiện trong thực tế (ngoại trừ đối với các thành phần đặc biệt quan trọng trong các hệ thống chủ chốt về an toàn) bởi vì các hệ thống trong thực tế cực kỳ phức tạp.

Một vài biện pháp tiếp cận được sử dụng để vượt qua các không gian trạng thái quy mô lớn gắn liền với các hệ thống thực tế:

- Áp dụng các biện pháp hình thức cho các yêu cầu và các thiết kế ở cấp cao khi mà hầu hết các chi tiết là rất trừu tượng;

- Áp dụng các biện pháp hình thức chỉ cho các thành phần chủ chốt nhất;

- Phân tích các mô hình phần cứng và phần mềm nếu các biến được tạo ra rời rạc và các dải bị giảm mạnh;

- Phân tích các mô hình hệ thống theo phương thức phân cấp mà có khả năng “chia để trị”;

- Tự động hóa trong việc thẩm tra càng nhiều càng tốt.

Mặc dù việc sử dụng các phép toán logic là một chủ đề thống nhất theo các quy tắc của các biện pháp hình thức, không có biện pháp hình thức độc lập nhất. Từng phạm vi áp dụng đều yêu cầu các phương pháp lập mô hình khác nhau và các biện pháp tiếp cận chứng cứ khác nhau. Ngoài ra, trong từng phạm vi áp dụng cụ thể, các giai đoạn khác nhau của vòng đời có thể được xử lý tốt nhất bằng các kỹ thuật và các chương trình khác nhau. Ví dụ, bộ chứng minh lý thuyết có thể được sử dụng tốt nhất để phân tích độ chính xác của mô tả mức chuyển thanh ghi của một mạch biến đổi chuỗi Fourier nhanh, trong khi đó các phương pháp có nguồn gốc số học có thể được sử dụng tốt nhất để phân tích độ chính xác của các sàng lọc thiết kế vào một thiết kế ở mức cổng. Vì vậy có một số lượng lớn các biện pháp hình thức đang được phát triển trên toàn thế giới.

Một số ví dụ về các biện pháp hình thức được mô tả trong các mục dưới đây của phụ lục này. Danh sách các ví dụ được nêu ở đây chưa phải là hoàn chỉnh nhất. Các biện pháp hình thức được mô tả là CSP, CCS, HOL, LOTOS, OBJ, Temporal Logic, VDM, Z Method, B Method và kiểm tra mô hình.

D.28.1 CSP – Quá trình giao tiếp liên tục

Mục đích

CSP là một kỹ thuật để quy định các hệ thống phần mềm hoạt động đồng thời, ví dụ: các hệ thống của quá trình giao tiếp hoạt động đồng thời.

Mô tả

CSP cung cấp một ngôn ngữ lập trình để quy định các hệ thống của các quá trình và làm bằng chứng để thẩm tra việc thực hiện các quá trình thỏa mãn các chỉ dẫn kỹ thuật của nó (được mô tả như là vết – các chuỗi tình huống được phép).

Một hệ thống được lập mô hình như một mạng lưới các quá trình độc lập. Mỗi quá trình được mô tả theo tất cả các hoạt động có thể của nó. Một hệ thống được lập mô hình bằng việc tạo ra các quá trình xảy ra theo chuỗi hoặc đồng thời. Các quá trình có thể giao tiếp thông qua các kênh (đồng bộ hoặc trao đổi dữ liệu), việc liên kết chỉ được tiến hành khi tất cả các quá trình đều sẵn sàng. Có thể lập mô hình thời gian các tình huống tương ứng.

Lý thuyết đằng sau CSP được tích hợp trực tiếp vào trong cấu trúc các bộ vi mạch INMOS, ngôn ngữ lập trình OCCAM cho phép chạy trực tiếp hệ thống được quy định CSP lên mạng lưới các bộ vị mạch.

D.28.2 CCS – Phép tính của các hệ thống giao tiếp

Mục đích

CCS là một phương pháp để mô tả và suy luận về sự hoạt động đồng thời của các hệ thống, các quá trình giao tiếp.

Mô tả

Tương tự như CSP, CCS là một phương pháp tính toán học liên quan tới sự hoạt động của các hệ thống. Thiết kế hệ thống được lập mô hình như một mạng lưới các quá trình độc lập hoạt động theo chuỗi hoặc song song. Các quá trình có thể giao tiếp với nhau qua các cổng (tương tự như các kênh của CSP), việc giao tiếp chỉ diễn ra khi tất cả các quá trình đều sẵn sàng. Không thể lập mô hình cho những quá trình không thể đưa ra được căn cứ. Bắt đầu từ sự mô tả trừu tượng mức độ cao của toàn bộ hệ thống (được coi như là một vết), có thể thực hiện một sự tinh chỉnh dần từng bước thành sự kết hợp của quá trình giao tiếp có toàn bộ hoạt động như được yêu cầu cho toàn bộ hệ thống. Tương tự,

có thể thực hiện công việc theo kiểu từ dưới lên, kết hợp các quá trình và tách rời các đặc tính của hệ thống tạo ra bằng cách sử dụng các quy tắc suy luận liên quan tới các quy tắc tổng hợp.

D.28.3 HOL – Logic cấp cao

Mục đích

Đây là ngôn ngữ lập trình hình thức dự định sẽ là cơ sở đối với việc thẩm tra và chỉ dẫn kỹ thuật phần cứng.

Mô tả

HOL (Logic cấp cao) tham chiếu tới một ký hiệu logic cụ thể và hệ thống hỗ trợ cho máy, những chương trình trên được phát triển tại Phòng kiểm thử máy tính của trường Đại học Cambridge. Dẫn giải logic hầu hết được lấy từ Church's Simple Theory of Types và hệ thống hỗ trợ máy được dựa trên hệ thống LCF (Logic hàm khả tính).

D.28.4 LOTOS

Mục đích

LOTOS là một phương thức để mô tả và tạo cơ sở cho sự hoạt động của các hệ thống có quá trình giao tiếp xảy ra đồng thời.

Mô tả

LOTOS (ngôn ngữ cho việc quy định mệnh lệnh tạm thời) trên cơ sở ngôn ngữ CCS có thêm các tính năng bổ sung từ tính toán CSP và CIRCAL (tính toán mạch) số học liên quan. Nó sẽ khắc phục được nhược điểm của CCS trong việc xử lý các cấu trúc dữ liệu và thể hiện giá trị bằng cách kết hợp một thành phần thứ hai dựa trên ngôn ngữ kiểu loại dữ liệu cơ sở ACT ONE. Tuy nhiên, có thể sử dụng thành phần xác định quá trình của LOTOS cùng với hệ hình thức cho bản mô tả các loại dữ liệu cơ sở.

D.28.5 OBJ

Mục đích

Để đưa ra một chỉ dẫn kỹ thuật hệ thống chính xác với các phản hồi từ người dùng và thẩm định hệ thống trước khi chạy chương trình.

Mô tả

OBJ là một ngôn ngữ chỉ dẫn kỹ thuật đại số. Người sử dụng sẽ quy định các yêu cầu theo dạng hàm toán học đại số. Các vấn đề về sự hoạt động, tính cấu trúc của hệ thống được quy định theo hoạt động

vận hành trên các loại dữ liệu cơ sở (ADT). Một ADT giống như một gói ADA trong đó nhìn thấy được sự hoạt động vận hành trong khi các chi tiết cho việc chạy chương trình bị "ẩn".

Chỉ dẫn kỹ thuật OBJ, và việc chạy chương trình từng bước tuần tự sẽ tuân theo các kỹ thuật chứng minh hình thức tương tự cũng như các biện pháp tiếp cận hình thức khác. Hơn nữa, do phạm vi cấu trúc của chỉ dẫn kỹ thuật OBJ có thể thực hiện bằng máy nên OBJ sẽ hướng thẳng tới việc đạt được sự thẩm định hệ thống từ chính chỉ dẫn kỹ thuật. Quá trình chạy chủ yếu là quá trình đánh giá chức năng thông qua việc thay thế phương trình (viết lại), việc này sẽ tiếp tục cho đến khi đạt được giá trị đầu ra cụ thể. Khả năng chạy chương trình này cho phép người sử dụng cuối cùng của hệ thống được lập ra có được đánh giá "tổng quan" về hệ thống cuối cùng ở giai đoạn chỉ dẫn kỹ thuật hệ thống mà không cần phải quen thuộc với các kỹ thuật quy định hình thức sơ bộ.

Giống như với tất cả các kỹ thuật ADT khác, OBJ chỉ có thể áp dụng cho các hệ thống xảy ra theo tuần tự, hoặc cho các lĩnh vực xảy ra theo chuỗi của các hệ thống xảy ra đồng thời. OBJ được sử dụng rộng rãi cho chỉ dẫn kỹ thuật cho cả các ứng dụng công nghiệp quy mô lớn và nhỏ.

D.28.6 Logic thời gian

Mục đích

Thể hiện một cách trực tiếp các yêu cầu hoạt động, an toàn và minh chứng hình thức những đặc tính này được duy trì trong các bước xây dựng tiếp theo.

Mô tả

Logic vị từ bậc 1 (predicate logic-logic cấp I) theo tiêu chuẩn sẽ không bao hàm khái niệm thời gian trong đó. Logic thời gian sẽ mở rộng logic vị từ bằng việc thêm vào các toán tử cách thức (ví dụ: Henceforth và Eventually). Những toán tử này có thể được sử dụng để đánh giá khả năng xác nhận hệ thống. Ví dụ: các đặc tính về an toàn có thể được yêu cầu để duy trì "hencerforth", trong khi các trạng thái hệ thống mong muốn khác có thể được yêu cầu để đạt được "eventually" từ một số trạng thái ban đầu. Các hàm thời gian được diễn giải theo chuỗi các trạng thái (sự hoạt động). Các yếu tố cấu thành một "trạng thái" sẽ phụ thuộc vào mức độ mô tả được lựa chọn. Sự mô tả này có thể hướng tới toàn bộ hệ thống, một thành phần trong hệ thống hoặc chương trình máy tính. Các khoảng thời gian được định lượng và các ràng buộc sẽ không được xử lý rõ ràng trong logic thời gian. Phải xử lý thời gian tuyệt đối bên ngoài bằng cách thêm vào các trạng thái thời gian bổ sung như một phần của việc xác định trạng thái.

D.28.7 VDM – Biện pháp xây dựng Vienna

Mục đích

Thực hiện và chỉ dẫn kỹ thuật có tính hệ thống các chương trình theo chuỗi.

Mô tả

VDM là biện pháp chỉ dẫn kỹ thuật trên cơ sở toán học và là kỹ thuật để cải tiến việc xử lý theo cách cho phép chứng minh khả năng hoạt động chính xác về chỉ dẫn kỹ thuật.

Biện pháp chỉ dẫn kỹ thuật là dựa trên mô hình mà trạng thái hệ thống được mô hình hóa về mặt cấu trúc lý thuyết được xác định là không đổi (các thuộc tính), và các quá trình vận hành ở trạng thái đó sẽ được mô hình hóa bằng cách quy định các điều kiện cần và đủ đối với trạng thái hệ thống. Các quá trình vận hành có thể được chứng minh để duy trì tính bất biến của hệ thống.

Việc xử lý chỉ dẫn kỹ thuật được thực hiện bằng cách cụ thể hóa trạng thái hệ thống về cấu trúc dữ liệu theo ngôn ngữ mục tiêu và bằng việc sàng lọc các phép toán của một chương trình theo ngôn ngữ mục tiêu. Các bước cụ thể hóa và sàng lọc sẽ làm phát sinh các mệnh đề cần chứng minh việc thiết lập chính xác. Người thiết kế sẽ lựa chọn có tiến hành các mệnh đề cần chứng minh này hay không.

VDM chủ yếu được sử dụng trong giai đoạn chỉ dẫn kỹ thuật nhưng có thể được sử dụng trong các giai đoạn thiết kế và thực hiện hướng tới các mã nguồn. Nó chỉ có thể áp dụng cho các chương trình liên tục hoặc các quá trình xử lý liên tục trong các hệ thống đồng thời.

D.28.8 Biện pháp Z

Mục đích

Z là một diễn giải ngôn ngữ quy định kỹ thuật đối với các hệ thống xảy ra liên tục và là một kỹ thuật thiết kế cho phép người phát triển xử lý từ chỉ dẫn kỹ thuật Z thành thuật toán có thể chạy được theo cách đưa ra các bằng chứng về sự chính xác của nó tương ứng với chỉ dẫn kỹ thuật.

Z chủ yếu được sử dụng trong giai đoạn chỉ dẫn kỹ thuật nhưng phải tìm ra một phương pháp để chuyển từ chỉ dẫn kỹ thuật này thành thiết kế và xử lý. Phương pháp này sẽ phù hợp nhất với quá trình xây dựng các hệ thống liên tục được định hướng theo dữ liệu.

Mô tả

Giống như VDM, kỹ thuật chỉ dẫn kỹ thuật là phương pháp dựa trên mô hình, ở đó trạng thái hệ thống sẽ được lập mô hình theo cấu trúc lý thuyết-tập hợp xác định các giá trị bất biến (các vị từ), và các hoạt động ở trạng thái đó được lập mô hình bằng cách quy định các điều kiện cần và đủ đối với trạng thái hệ thống. Các hoạt động vận hành có thể được phê chuẩn để duy trì các giá trị bất biến của hệ thống từ đó chứng minh tính thống nhất không đổi của nó. Phần chính thức của chỉ dẫn kỹ thuật được phân chia thành các sơ đồ cho phép lập cấu trúc các chỉ dẫn kỹ thuật thông qua việc cải tiến sửa đổi.

Thông thường, một chỉ dẫn kỹ thuật Z là sự kết hợp của Z hình thức và phần nội dung diễn giải thông tin theo ngôn ngữ tự nhiên. (Nội dung hình thức Z có thể quá ngắn, không thể đọc được một cách dễ

dàng và thường cần phải giải thích về mục đích của nội dung này, trong khi ngôn ngữ tự nhiên mang tính thông tin có thể dễ bị sai và không rõ ràng).

Không giống như VDM, Z giống như diễn giải hơn là một phương pháp hoàn chỉnh. Tuy nhiên, một biện pháp kết hợp (được gọi là B) được xây dựng để có thể sử dụng cùng với Z.

D.28.9 Biện pháp B

Mục đích

Giống như VDM, mục đích của biện pháp B là để mô hình hóa một hệ thống hoặc phần mềm hình thức và để chứng minh rằng sự hoạt động của hệ thống hoặc phần mềm tuân thủ các đặc tính được tạo ra chi tiết trong suốt quá trình lập mô hình.

Mô tả

Việc lập mô hình B gọi các phần tử toán từ lý thuyết tập hợp. Một mặt các giá trị bất biến (ví dụ, các vị từ) xác định các đặc tính tĩnh của mô hình. Mặt khác, các phép toán thiết lập các điều kiện sau, do đó sẽ xác định sự hoạt động của nó. Chỉ dẫn kỹ thuật của phần mềm hoặc hệ thống phức hợp có thể được tạo ra bằng cách phân tích mô hình thành các máy được gắn chặt với các đường dẫn ngữ nghĩa khác nhau.

Có hai loại mô hình hệ hình thức B chính sau:

- The former, nhằm mục đích phát triển phần mềm: trong trường hợp này, mục tiêu là để tạo ra một chương trình mà tuân thủ chỉ dẫn kỹ thuật của nó. Mô hình này bao gồm các máy trừu tượng (không nhất thiết phải xác định) và việc cải tiến từng bước các máy này, dẫn tới xác định các lệnh thực hiện được viết bằng mã giả gọi là “Bo”. Sau đó mã giả này có thể được chuyển đổi tự động thành các ngôn ngữ lập trình mục tiêu.

- The latter, nhằm mục đích mô hình hóa các hệ thống và trong trường hợp này sẽ nói về tình huống B: mục đích là để quy định rõ ràng và mạch lạc một hệ thống mà đáp ứng đầy đủ các đặc tính chi tiết. Trong mô hình phải thể hiện cả hệ thống và môi trường của hệ thống. Động lực học của hệ thống được mô hình hóa bởi các “tình huống”, và các kỹ thuật cải tiến được sử dụng cho các tương tác chính xác giữa hệ thống và môi trường của hệ thống.

Một tập hợp các mệnh đề cần chứng minh (các biểu thức logic mà cần phải được chứng minh chính thức từ các giả thuyết mà được trích ra từ mô hình hình thức B) được tạo ra một cách tự động. Những mệnh đề cần chứng minh này bảo đảm:

- Sự tồn tại của dữ liệu mà đáp ứng các đặc tính động và tĩnh của mô hình;

- Các phép tính (chạy động mô hình) theo bất biến;

- Làm mịn các phép tính và dữ liệu (và mã giả Bo nếu cần) không mâu thuẫn với chỉ dẫn kỹ thuật được viết trong các máy trừu tượng;

- Gọi từng phép tính trong phạm vi nội dung của điều kiện trước;

- Chương trình đó sẽ kết thúc trong trường hợp lập mô hình phần mềm (đặc biệt kết thúc từng vòng lặp).

Các mệnh đề cần chứng minh khác cũng phải được tạo ra, ví dụ thẩm tra việc tràn số nguyên.

D.28.10 Kiểm tra mô hình

Mục đích

Đưa ra một mô hình của một hệ thống, tự động kiểm thử liệu mô hình này có thỏa mãn chỉ dẫn kỹ thuật đề ra.

Mô tả

Kiểm tra mô hình là quá trình kiểm tra liệu cấu trúc đã xây dựng có phải là mô hình của công thức logic trước đó. Khái niệm này là tổng quát và áp dụng cho tất cả các loại logic và các cấu trúc phù hợp. Một vấn đề kiểm tra mô hình đơn giản sẽ kiểm thử liệu công thức được đưa ra trong logic mệnh đề có phù hợp với cấu trúc đã xây dựng hay không.

Một hạng mục quan trọng của mô hình là sẽ kiểm tra các phương pháp được xây dựng theo thuật toán để thẩm tra các hệ thống hình thức. Việc kiểm tra này sẽ đạt được bằng việc thực hiện thẩm tra liệu cấu trúc này, thường được xây dựng từ thiết kế phần cứng hoặc phần mềm, có thỏa mãn chỉ dẫn kỹ thuật hình thức, điển hình là công thức logic thời gian.

Việc kiểm tra mô hình thường áp dụng chủ yếu cho các thiết kế phần cứng. Đối với phần mềm, do tính không giải được (xem lý thuyết tính toán) cách tiếp cận này không thể là thuật toán đầy đủ, điển hình nó có thể không chứng minh hoặc hoặc bác bỏ một đặc tính trước đó.

Cấu trúc này thường được đưa ra như là một bản mô tả mã nguồn trong ngôn ngữ lập trình mô tả phần cứng công nghiệp hoặc ngôn ngữ lập trình mục đích đặc biệt. Chẳng hạn như một chương trình tương ứng với một cơ chế chuyển đổi trạng thái hữu hạn, ví dụ, một sơ đồ hướng dẫn gồm có các nút (hoặc các đỉnh) và các cạnh. Một tập hợp các mệnh đề con gắn liền với từng nút, sẽ công bố các thành phần bộ nhớ hợp lại làm một. Những nút này sẽ thể hiện trạng thái của hệ thống, các cạnh thể hiện cách chuyển đổi có thể xảy ra mà có thể thay thế trạng thái, trong khi đó các mệnh đề con thể hiện các đặc tính cơ bản mà xử lý các điểm chạy chương trình.

Chính thức, vấn đề này có thể được kết luận như sau: đưa ra một đặc tính mong muốn, diễn đạt như là một công thức logic thời gian p, và một cấu trúc M với trạng thái ban đầu s, quyết định if. Nếu M là hữu hạn, chẳng hạn như tồn tại ở trong phần cứng, thì việc kiểm tra mô hình sẽ làm giảm việc tìm kiếm sơ đồ.

D.29 Chứng minh hình thức

Mục đích

Sử dụng các mô hình lý thuyết, toán học và các quy tắc để có thể chứng minh tính đúng đắn của chương trình mà không cần chạy chương trình.

Mô tả

Một số lượng các xác nhận được đưa ra ở các vị trí khác nhau trong chương trình, và chúng được sử dụng như là các điều kiện cần và đủ cho các đường chạy khác nhau trong chương trình. Việc chứng minh bao gồm việc thể hiện chương trình đã chuyển đổi các điều kiện cần và đủ theo một bộ nguyên tắc logic, và thể hiện chương trình đã hoàn thành.

Một số các biện pháp hình thức được mô tả trong phụ lục này, như CCS, CSP, HOL, LOTOS, OBJ, Logic thời gian, VDM và Z. Bản mô tả những phương pháp này có thể được tìm thấy trong phần D.28 “Các biện pháp hình thức”.

B.30 Phục hồi tiến

Mục đích

Để tạo ra sự hoạt động theo đúng chức năng chuẩn trong việc thể hiện một hoặc nhiều sự cố.

Mô tả

Nếu phát hiện ra một sự cố, trạng thái hiện tại của hệ thống sẽ được điều khiển để đạt được một trạng thái mà sẽ không đổi trong một khoảng thời gian sau đó. Khái niệm này đặc biệt phù hợp với các hệ thống thời gian thực có dữ liệu nhỏ và tỉ lệ thay đổi trạng thái nội bộ nhanh. Giả thiết rằng ít nhất một phần trạng thái hệ thống có thể bị tác động đến môi trường và chỉ một phần các trạng thái hệ thống là bị ảnh hưởng (bắt buộc) bởi môi trường.

D.31 Suy giảm nhẹ

Mục đích

Để duy trì thêm tính khả dụng của các chức năng hệ thống quan trọng bằng cách loại bỏ các chức năng ít quan trọng hơn cho dù có xảy ra hư hỏng.

Mô tả

Kỹ thuật này đưa ra sự ưu tiên cho các chức năng khác nhau được thực hiện trong hệ thống. Thiết kế sau đó sẽ đảm bảo việc nếu như không có đủ các nguồn lực để tiến hành tất cả các chức năng của hệ thống, khi đó các chức năng được ưu tiên cao hơn sẽ được tiến hành theo thứ tự đến các chức năng ưu tiên thấp hơn. Ví dụ: các chức năng ghi lại lỗi và tình huống có thể ít ưu tiên hơn các chức năng điều khiển hệ thống. Việc điều khiển hệ thống sẽ tiếp tục nếu phần cứng liên quan đến việc ghi lỗi bị hỏng.

Ví dụ khác sẽ là hệ thống tín hiệu khi rơi vào tình huống mất thông tin liên lạc với trung tâm điều khiển, thiết bị tín hiệu dọc đường sẽ tự động thiết lập các tuyến khả dụng để dẫn đường cho các đoàn tàu ưu tiên cao nhất. Đây chính là sự suy giảm nhẹ bởi vì các đoàn tàu ở trên các tuyến ưu tiên sẽ có khả năng thông qua khu vực bị tác động bởi việc mất thông tin liên lạc với trung tâm điều khiển còn các đoàn tàu tàu khác chẳng hạn như các đoàn tàu dồn hoặc chạy trên các đường nhánh thì không thể qua được.

D.32 Phân tích tác động

Mục đích

Để xác định rõ tác động của một thay đổi hoặc một cải tiến hệ thống phần mềm đến các thành phần khác trong hệ thống phần mềm đó cũng như các hệ thống khác.

Mô tả

Trước khi thực hiện một cải tiến hoặc bổ sung sửa đổi phần mềm, phải tiến hành một phân tích để xác định rõ các tác động của thay đổi hoặc cải tiến lên phần mềm và đồng thời để xác định rõ các hệ thống và thành phần phần mềm bị tác động.

Sau khi hoàn chỉnh phân tích, sẽ yêu cầu một quyết định liên quan tới việc thẩm tra lại hệ thống phần mềm. Việc thẩm tra lại sẽ phụ thuộc vào số lượng các thành phần bị tác động, mức quan trọng của các thành phần bị tác động và bản chất của thay đổi. Các quyết định có thể là:

- Chỉ các thành phần bị thay đổi cần phải thẩm tra lại;
- Tất cả các thành phần bị tác động được xác định rõ phải thẩm tra lại;
- Toàn bộ hệ thống được thẩm tra lại.

D.33 Đóng gói / giấu thông tin

Mục đích

Để tăng độ tin cậy và khả năng bảo trì phần mềm.

Mô tả

Dữ liệu mà có khả năng truy cập mở rộng tới tất cả các thành phần phần mềm có thể được thay đổi bất ngờ hoặc thay đổi không đúng từ bất kỳ thành phần nào trong số các thành phần này. Mọi thay đổi đến cấu trúc dữ liệu này có thể yêu cầu việc kiểm tra chi tiết về mã lệnh và các thay đổi mở rộng.

Việc ẩn thông tin là cách cơ bản nhất cho việc giảm thiểu những khó khăn này. Các cấu trúc dữ liệu chính sẽ bị "ẩn" và chỉ có thể được tiếp cận thủ công thông qua một loạt các quy trình truy cập xác định. Điều này cho phép các cấu trúc bên trong được thay đổi hoặc các quy trình khác được thêm vào mà không có tác động đến sự hoạt động chức năng của phần mềm còn lại. VÍ dụ, danh sách tên gọi có thể có các quy trình truy cập: Insert, Delete, và Find. Các quy trình tiếp cận và các cấu trúc dữ liệu bên trong có thể được viết lại (ví dụ: để sử dụng một phương pháp soi tìm kiếm khác hoặc để lưu lại các tên trong một ổ cứng) mà không tác động đến sự hoạt động logic của phần mềm còn lại đang sử dụng những quy trình này.

Ý tưởng về một loại dữ liệu cơ sở sẽ được hỗ trợ trực tiếp trong một số ngôn ngữ lập trình, nhưng nguyên tắc cơ bản đều có thể được áp dụng cho dù sử dụng bất cứ ngôn ngữ lập trình nào.

D.34 Kiểm thử giao diện

Mục đích

Để chứng minh các giao diện của các chương trình con không có bất kì lỗi nào hoặc mọi lỗi dẫn đến hư hỏng trong ứng dụng cụ thể của phần mềm hoặc để phát hiện ra tất cả các lỗi có thể liên quan.

Mô tả

Có thể sử dụng một số mức độ chi tiết hoặc mức độ hoàn thiện của kiểm thử. Các mức độ quan trọng nhất được đang được kiểm thử là:

- Tất cả các biến giao diện ở các vị trí tới hạn của chúng;
- Tất cả các biến giao diện ở từng giá trị tới hạn với các biến giao diện khác ở các giá trị bình thường;
- Tất cả các giá trị trong phạm vi của từng biến giao diện với các biến giao diện khác ở các giá trị bình thường;
- Tất cả các giá trị của tất cả các biến trong quá trình kết hợp (kiểm thử này chỉ khả thi đối với các giao diện nhỏ);
- Các điều kiện kiểm thử được quy định liên quan tới từng lệnh gọi chương trình con.

Những kiểm thử này chỉ đặc biệt quan trọng nếu các giao diện của nó không có chứa các xác nhận phát hiện ra các giá trị thông số sai. Những kiểm thử này cũng quan trọng sau khi tạo ra cấu hình mới của các chương trình con đã được tạo ra trước đó.

D.35 Tập con ngôn ngữ

Mục đích

Để giảm thiểu xác suất tạo ra các sự cố lập trình và tăng xác suất phát hiện mọi sự cố còn lại.

Mô tả

Ngôn ngữ sẽ được kiểm tra để xác định rõ các cấu trúc lập trình dễ bị lỗi hoặc khó phân tích, ví dụ: sử dụng các phương pháp phân tích tĩnh. Một tập con ngôn ngữ sẽ được xác định để loại trừ những cấu trúc này.

D.36 Ghi nhớ các trường hợp thực hiện

Mục đích

Để buộc phần mềm về chế độ an toàn khi sự cố nếu thực hiện theo các đường chạy không được phép.

Mô tả

Trong quá trình cấp phép cho các câu lệnh, một bản ghi sẽ được tạo ra đối với tất cả các chi tiết liên quan trong từng lần chạy chương trình. Trong quá trình hoạt động bình thường, mỗi lần chạy chương trình sẽ được so sánh với tập hợp các lần chạy được phép. Nếu chúng khác nhau, một hành động an toàn sẽ được đưa ra.

Bản ghi lại việc thực hiện có thể là tập hợp các đường chạy quyết định tới quyết định độc lập (DD paths) hoặc chuỗi các truy cập độc lập đến các mảng, các bản ghi hoặc các phần, hoặc tất cả.

Có thể có các biện pháp khác nhau để lưu trữ các đường chạy chương trình. Các biện pháp mã hóa Hash có thể được sử dụng để lập sơ đồ chuỗi hoạt động thành một số lượng lớn duy nhất hoặc chuỗi các số. Trong quá trình hoạt động bình thường, giá trị lần chạy phải được kiểm tra theo các trường hợp đã được lưu trước khi xuất hiện bất cứ đầu ra vận hành nào.

Do các kết hợp có thể của các đường chạy quyết định đến quyết định trong một chương trình là rất lớn, việc coi các chương trình thành một tổng thể là không khả thi. Trong trường hợp này, có thể áp dụng kỹ thuật ở mức thành phần.

D.37 Đo kiểm thử Metrics

Mục đích

Để dự đoán trước các thuộc tính của các chương trình từ các đặc tính của chính phần mềm hơn là dự đoán từ quá trình xây dựng hoặc từ lịch sử quá trình kiểm thử phần mềm.

Mô tả

Những mô hình này sẽ tính toán một số đặc tính cấu trúc phần mềm và sẽ tạo mối liên hệ với thuộc tính mong muốn như độ tin cậy hoặc độ phức tạp. Các chương trình phần mềm sẽ được yêu cầu để tính toán hầu hết các tham số. Một số các thước đo có thể áp dụng như sau:

- Độ phức tạp lý thuyết theo sơ đồ: đại lượng này có thể áp dụng sớm trong vòng đời để đánh giá các quy đổi, và được dựa trên độ phức tạp của sơ đồ điều khiển chương trình, thể hiện bằng số lặp (đánh giá độ tin cậy của chương trình);

- Số lượng các cách thức để kích hoạt một thành phần nhất định (khả năng truy cập): càng nhiều cách thức để đánh giá thành phần, càng có khả năng hiệu chỉnh lỗi;

- Khoa học phần mềm: phương pháp này sẽ tính toán độ dài của chương trình bằng cách đếm số lượng các toán tử và toán hạng. Phương pháp này sẽ đưa ra một đại lượng về độ phức tạp và tính toán các nguồn lực phát triển;

- Số lượng các đầu vào và các đầu ra trong mỗi thành phần: giảm thiểu tối đa số lượng các điểm đầu vào/đầu ra là một đặc tính quan trọng trong các kỹ thuật thiết kế và lập trình có tính cấu trúc.

D.38 Tiếp cận module

Mục đích

Phân chia phần mềm thành các phần hoàn chỉnh nhỏ hơn để giới hạn độ phức tạp của phần mềm.

Mô tả

Biện pháp tiếp cận module hoặc module hóa có chứa một số các nguyên tắc đối với các giai đoạn thiết kế, mã hóa và bảo trì một dự án phần mềm. Những nguyên tắc này thay đổi theo biện pháp thiết kế được sử dụng trong giai đoạn thiết kế. Hầu hết các biện pháp đều có những nguyên tắc sau:

- Một module/thành phần phải đáp ứng một nhiệm vụ hoặc chức năng được xác định độc lập;

- Các liên kết giữa các module/thành phần phải bị giới hạn và xác định một cách chặt chẽ, tính gắn kết trong một module/thành phần phải mạnh;

- Phải xây dựng việc tập hợp các chương trình con có một số mức module/thành phần;

- Các chương trình con phải có duy nhất 1 cổng vào và 1 cổng ra;

- Các module/thành phần phải liên kết với các module/thành phần khác thông qua các giao diện của chúng. Khi sử dụng các biến chung hoặc phổ biến, chúng phải có cấu trúc tốt, việc truy cập phải được kiểm soát và phải đưa ra cơ sở cho việc sử dụng chúng trong từng trường hợp;

- Tất cả các giao diện module/thành phần phải được lưu lại đầy đủ;

- Mọi các giao diện module/thành phần phải chứa số lượng nhỏ nhất các thông số cần thiết đối với chức năng của các module/thành phần;

- Phải quy định một hạn chế phù hợp số lượng các thông số, thường là 5.

D.39 Lập mô hình hiệu năng

Mục đích

Để đảm bảo khả năng làm việc của hệ thống là đủ để đáp ứng các yêu cầu được quy định.

Mô tả

Chỉ dẫn các yêu cầu bao gồm các yêu cầu về lưu lượng và phản hồi các yêu cầu đối với các chức năng cụ thể, có thể được kết hợp với các ràng buộc khi sử dụng các nguồn lực tổng hợp trong hệ thống. Thiết kế hệ thống đề xuất sẽ được so sánh theo các yêu cầu đã đưa ra, bằng cách:

- Xác định một mô hình cho các quá trình xử lý trong hệ thống, và các tương tác của chúng;

- Xác định rõ việc sử dụng các nguồn lực của từng quá trình, ví dụ: thời gian cho bộ xử lý, dải giao tiếp, thiết bị lưu trữ…

- Xác định việc phân bổ các yêu cầu cho hệ thống trong các điều kiện trung bình và xấu nhất;

- Tính toán thời gian lưu thông, phản hồi trung bình và xấu nhất cho từng chức năng hệ thống.

Đối với các hệ thống đơn giản, có thể có một giải pháp mang tính phân tích, trong khi đối với các hệ thống phức tạp hơn lại yêu cầu một số dạng mô phỏng để thu được các kết quả chính xác.

Trước khi lập mô hình chi tiết, có thể sử dụng đơn vị kiểm tra “dự tính nguồn lực” đơn giản hơn, kiểm tra này sẽ tính tổng các yêu cầu về nguồn lực trong tất cả các quá trình xử lý. Nếu các yêu cầu vượt quá khả năng của hệ thống được thiết kế thì thiết kế sẽ không khả thi. Thậm chí nếu thiết kế vượt qua kiểm tra này, việc mô hình hóa tính năng hoạt động vẫn có thể cho thấy các độ trễ liên tiếp và các thời gian phản hồi xuất hiện do thiếu các nguồn lực. Để tránh tình huống này, các kỹ sư thường thiết kế các

hệ thống chỉ sử dụng một phần của toàn bộ nguồn lực (ví dụ: 50 %) để sao cho giảm được xác suất thiếu các nguồn lực.

D.40 Các yêu cầu về hiệu năng

Mục đích

Để thiết lập việc thỏa mãn các yêu cầu về hiệu năng của một phần mềm.

Mô tả

Một phân tích được tiến hành trên cả hệ thống và Chỉ dẫn các yêu cầu phần mềm để xác định rõ tất cả các yêu cầu hiệu năng chung và cụ thể, rõ ràng và tiềm ẩn.

Từng yêu cầu về hiệu năng được kiểm tra lần lượt để xác định:

- Chỉ tiêu thành công đạt được;
- Liệu có thể tìm ra một biện pháp theo chỉ tiêu thành công;
- Độ chính xác có thể của những biện pháp như vậy;
- Các giai đoạn dự án có thể đánh giá các biện pháp;
- Các giai đoạn dự án có thể tạo ra các biện pháp.

Khả năng thực hiện từng yêu cầu hoạt động sau đó được phân tích để đưa ra được một danh sách các yêu cầu hoạt động, chỉ tiêu thành công và các biện pháp có thể. Mục đích chính là:

a) Mỗi yêu cầu hiệu năng được tích hợp với ít nhất một biện pháp.

b) Khi có thể, các biện pháp chính xác và có hiệu quả được lựa chọn có thể được sử dụng ngay khi có thể trong quá trình phát triển.

c) Xác định rõ các yêu cầu hiệu năng chủ yếu, không bắt buộc và chỉ tiêu thành công.

d) Khi có thể, phải đưa ra được ưu điểm về khả năng sử dụng một biện pháp độc lập cho nhiều hơn một yêu cầu về hiệu năng.

D.41 Kiểm thử xác xuất

Mục đích

Để đạt được giá trị định lượng về các đặc tính độ tin cậy của phần mềm được điều tra. Giá trị này có thể đề cập tới các mức độ độ tin cậy và mức độ quan trọng liên quan và:

a) Xác xuất hư hỏng trên mỗi yêu cầu.

b) Xác suất hư hỏng trong một khoảng thời gian nhất định.

c) Xác suất chứa lỗi trong đó.

Từ những giá trị này, có thể tìm ra các thông số khác như:

- Xác suất của việc hoạt động không hư hỏng;
- Xác suất tồn tại;
- Tính sẵn sàng;
- MTBF của tỉ lệ hư hỏng;
- Xác suất của việc hoạt động an toàn.

Mô tả

Các xem xét đánh giá mang tính xác xuất dựa trên kiểm thử xác suất hoặc kinh nghiệm về vận hành. Thường thì số lượng các trường hợp kiểm thử của các trường hợp hoạt động được theo dõi là rất lớn.

Để tạo thuận lợi cho việc kiểm thử, thường sử dụng các chương trình hỗ trợ tự động. Các chương trình này liên quan tới các chi tiết của quy định dữ liệu kiểm thử và giám sát đầu ra kiểm thử. Các kiểm thử quy mô lớn được thực hiện trên các máy chủ lớn với phạm vi mô phỏng quá trình phù hợp. Dữ liệu kiểm thử được lựa chọn cả theo quan điểm hệ thống và quan điểm ngẫu nhiên. Quan điểm hệ thống liên quan tới việc kiểm soát toàn bộ quá trình kiểm thử, ví dụ: đảm bảo hồ sơ dữ liệu kiểm thử. Việc lựa chọn ngẫu nhiên sẽ lấy các trường hợp kiểm thử cụ thể ở mức độ chi tiết.

Các khai thác kiểm thử độc lập, các quá trình thực hiện kiểm thử và các giám sát kiểm thử được xác định bằng các chương trình hỗ trợ kiểm thử chi tiết như mô tả ở trên. Các điều kiện quan trọng khác được đưa ra thông qua quá trình đáp ứng các điều kiện toán học tiên quyết để có thể đánh giá kiểm thử theo quan điểm của mục đích kiểm thử như dự định.

Các giá trị mang tính xác suất về sự hoạt động của mọi đối tượng kiểm thử cũng có thể được tìm ra từ kinh nghiệm vận hành. Miễn là các điều kiện tương tự được đáp ứng và các thuật toán tương tự có thể được áp dụng như khi đánh giá các kết quả kiểm thử.

D.42 Mô phỏng quá trình

Mục đích

Để kiểm thử chức năng phần mềm, cùng với giao diện của nó với thế giới bên ngoài, mà không cho phép thay đổi thế giới thực theo bất kì cách nào.

Mô tả

Việc tạo ra một hệ thống chỉ nhằm mục đích kiểm thử mà giả lập sự hoạt động của hệ thống phải được kiểm soát bởi hệ thống được kiểm thử.

Việc mô phỏng có thể chỉ sử dụng phần mềm hoặc bằng việc kết hợp của phần mềm và phần cứng. Mô phỏng phải:

- Đưa ra tất cả các đầu vào của hệ thống được kiểm thử mà sẽ tồn tại khi hệ thống được cài đặt;
- Phản hồi lại các đầu ra từ hệ thống theo cách thể hiện nguyên bản thiết bị được kiểm soát;
- Có các quy định đối với các đầu vào của các toán tử để đưa ra mọi nhiễu loạn đi kèm mà hệ thống kiểm thử được yêu cầu xử lý.

Khi phần mềm đang được kiểm thử, việc mô phỏng có thể là mô phỏng phần cứng dự định cùng với các đầu vào và đầu ra của nó.

D.43 Lập mô hình mẫu / mô phỏng

Mục đích

Để kiểm tra tính khả thi của việc chạy hệ thống dựa trên các ràng buộc đưa ra. Để thông tin việc diễn giải hệ thống của người quy định tới khách hàng, để tránh hiểu sai.

Mô tả

Một tập con các chức năng hệ thống, các ràng buộc và các yêu cầu hoạt động sẽ được lựa chọn. Một mô hình mẫu được xây dựng bằng cách sử dụng các chương trình cấp cao. Ở giai đoạn này, các ràng buộc như máy tính mục tiêu, ngôn ngữ chạy, quy mô chương trình, khả năng bảo trì, độ tin cậy và tính sẵn có không cần thiết phải xem xét. Mô hình mẫu được đánh giá dựa trên tiêu chí của khách hàng và các yêu cầu hệ thống có thể được sửa đổi theo quan điểm của đánh giá này.

D.44 Khối phục hồi

Mục đích

Để làm tăng khả năng chương trình thực hiện chức năng dự định của nó.

Mô tả

Một số các phần chương trình khác nhau được viết ra thường là độc lập, mỗi phần dự định thực hiện cùng chức năng mong muốn. Chương trình cuối cùng được cấu trúc từ những phần này. Phần đầu tiên được gọi là phần chính, sẽ được thực hiện đầu tiên. Phần này sẽ được chạy sau khi kiểm thử nghiệm thu kết quả nó tính toán. Nếu kiểm thử được thông qua thì kết quả sẽ được chấp nhận và

được đưa vào các phần sau đó của hệ thống. Nếu nó không thông qua, mọi tác động phụ của phần đầu sẽ được khôi phục lại và phần thứ hai, được gọi là phần thay thế đầu tiên sẽ được chạy. Phần này cũng được chạy sau khi kiểm thử nghiệm thu và được xem xét như phần đầu tiên. Có thể đưa ra các phần thay thế thứ hai, thứ ba và nhiều hơn nếu được yêu cầu.

D.45 Các ràng buộc về thời gian phản hồi và bộ nhớ

Mục đích

Để đảm bảo hệ thống sẽ đáp ứng các yêu cầu theo thời gian và các yêu cầu của bộ nhớ.

Mô tả

Chỉ dẫn các yêu cầu cho hệ thống và phần mềm sẽ bao gồm các yêu cầu về bộ nhớ và phản hồi cho các chức năng cụ thể, có thể được kết hợp với các ràng buộc khi sử dụng toàn bộ nguồn lực hệ thống. Một phân tích được thực hiện để xác định rõ các yêu cầu về sự phân bổ trong các điều kiện trung bình và xấu nhất. Phân tích này yêu cầu sự đánh giá việc sử dụng nguồn lực và thời gian còn lại cho từng chức năng hệ thống. Các đánh giá này có thể đạt được theo một số cách, ví dụ: so sánh với hệ thống hiện có hoặc lập mô hình thử và tạo ra điểm tham chiếu thời gian các hệ thống quan trọng.

D.46 Cơ chế phục hồi sự cố kiểu thử lại

Mục đích

Để cố gắng phục hồi chức năng từ một điều kiện lỗi bị phát hiện bằng cơ chế thử lại.

Mô tả

Trong tình huống có lỗi hoặc sự cố được phát hiện, các công việc sẽ được thực hiện để phục hồi tình huống bằng cách chạy lại cùng các mã lệnh. Quá trình phục hồi bằng cách thử lại có thể hoàn chỉnh như một quy trình khởi động lại và bắt đầu lại hoặc như một nhiệm vụ tái lập kế hoạch và tái khởi động lại, sau một hoạt động theo dõi cho nhiệm vụ hoặc dùng chạy phần mềm. Các kỹ thuật thử lại chủ yếu được sử dụng trong khi có sự cố về giao tiếp hoặc khôi phục lỗi, và các điều kiện cho việc thử lại có thể phát sinh từ lỗi giao thức giao tiếp (kiểm tra tổng quát…) hoặc khi hết thời gian phản hồi xác nhận giao tiếp.

D.47 Túi an toàn

Mục đích

Để bảo vệ chống lại các sự cố có trong quy định kỹ thuật còn lại và các sự cố hoạt động về phần mềm ảnh hưởng có hại tới an toàn.

Mô tả

Túi an toàn là một phương pháp giám sát từ bên ngoài, được thực hiện trên một máy tính độc lập đối với các đặc tĩnh thuật khác nhau. Túi an toàn chỉ đơn thuần liên quan tới việc đảm bảo máy tính chính thực hiện các hoạt động an toàn, không nhất thiết phải đúng. Túi an toàn sẽ giám sát liên tục máy tính chính. Túi an toàn sẽ ngăn không cho hệ thống đi vào trạng thái không an toàn. Thêm vào đó nếu phát hiện ra máy tính chính đang đi vào một trạng thái nguy hiểm tiềm ẩn, hệ thống phải được đưa về trạng thái an toàn bằng túi an toàn hoặc máy tính chính.

D.48 Quản lý cấu hình phần mềm

Mục đích

Quản lý cấu hình phần mềm nhằm đảm bảo tính thống nhất của các nhóm sản phẩm xây dựng chuyển giao khi những sản phẩm này thay đổi. Nói chung, việc quản lý cấu hình áp dụng cho cả việc xây dựng phần cứng và phần mềm.

Mô tả

Quản lý cấu hình phần mềm là một kỹ thuật được sử dụng xuyên suốt toàn bộ quá trình xây dựng. Về bản chất, kỹ thuật này yêu cầu việc ghi lại quá trình phát triển của từng phiên bản sản phẩm “có ý nghĩa” và từng mối liên quan giữa các phiên bản khác nhau của các sản phẩm chuyển giao khác nhau. Các bản ghi tạo ra sẽ cho phép người xây dựng xác định các tác động lên các sản phẩm chuyển giao khác trong trường hợp có thay đổi của một sản phẩm chuyển giao (đặc biệt là các thành phần của nó). Đặc biệt, các hệ thống hoặc các hệ thống con có thể được xây dựng lại một cách tin cậy từ các tập hợp thống nhất các phiên bản của thành phần.

D.49 Ngôn ngữ lập trình mạnh

Mục đích

Giảm xác suất xảy ra các sự cố bằng cách sử dụng một ngôn ngữ lập trình mà cho phép kiểm tra ở mức độ cao bằng chương trình biền dịch.

Mô tả

Những ngôn ngữ lập trình như vậy thường cho phép xác định các dạng dữ liệu được xác định theo người sử dụng từ các loại dữ liệu ngôn ngữ lập trình cơ bản (như Integer, Real). Những dạng dữ liệu này có thể được sử dụng chính xác theo cùng cách thức như các loại cơ bản, nhưng bắt buộc phải có các kiểm tra chặt chẽ để đảm bảo sử dụng đúng dạng dữ liệu. Những kiểm tra này được bắt buộc áp dụng lên toàn bộ chương trình, kể cả khi nếu nó được xây dựng từ các chương trình biên dịch riêng.

Các kiểm tra này cũng đảm bảo số lượng và kiểu loại tham số quy trình phù hợp kể cả khi được tham chiếu từ các thành phần chuyển dịch riêng.

Các ngôn ngữ lập trình mạnh cũng hỗ trợ cho những lĩnh vực khác của việc sử dụng kỹ thuật phần mềm tốt, như các cấu trúc kiểm soát dễ phân tích (ví dụ IF…Then…ELSE, DO…WHILE…) mà tạo ra các chương trình có cấu trúc tốt.

Các ví dụ điển hình về các ngôn ngữ lập trình mạnh là: Pascal, Ada, Modula 2.

D.50 Kiểm thử dựa trên cấu trúc

Mục đích

Để áp dụng các kiểm thử sử dụng các thành phần con nhất định trong cấu trúc chương trình.

Mô tả

Dựa trên phân tích chương trình, một bộ dữ liệu đầu vào sẽ được lựa chọn sao cho phần lớn các thành phần chương trình sẽ được sử dụng. Các thành phần chương trình được sử dụng có thể thay đổi đa dạng phụ thuộc vào mức độ rõ ràng được yêu cầu:

- Các câu lệnh: Đây là kiểm thử ít rõ ràng nhất do có thể thực hiện tất cả các câu lệnh mã hóa mà không cần chạy các nhánh của câu lệnh điều kiện;

- Các nhánh: Tất cả các vế của từng nhánh nên được kiểm tra. Việc này có thể không thực tế đối với một số loại mã phòng vệ;

- Các điều kiện kết hợp: Từng điều kiện trong nhánh điều kiện kết hợp (ví dụ: chạy các liên kết được nối bằng AND/OR);

- LCSAJ (một chuỗi mã thẳng và bước nhảy) là bất kỳ chuỗi sắp xếp các câu lệnh mã hóa, bao gồm các bước nhảy có điều kiện được kết thúc bởi một bước nhảy. Nhiều giao thức phụ tiềm ẩn sẽ có thể không khả thi do các ràng buộc bắt buộc về dữ liệu đầu vào từ việc thực hiện các mã trước đó;

- Luồng dữ liệu: Các giao thức hoạt động được lựa chọn trên cơ sở việc sử dụng dữ liệu, ví dụ: giao thức có cùng các biến được đọc và ghi;

- Sơ đồ Call graph: Một chương trình chứa đựng các thủ tục con mà có thể gọi ra các thủ tục khác. Sơ đồ Call graph là cây viện dẫn thủ tục con trong chương trình. Các kiểm thử được thiết kế để bao quát tất cả các viện dẫn trong cây;

- Giao thức toàn diện: Thực hiện tất cả các giao thức có thể qua mã lệnh. Việc kiểm thử hoàn chỉnh thường không khả thi do số lượng rất lớn các giao thức tiềm ẩn.

D.51 Sơ đồ cấu trúc

Mục đích

Để đưa ra cấu trúc của một chương trình theo dạng sơ đồ.

Mô tả

Các sơ đồ cấu trúc là sự diễn giải bổ sung cho sơ đồ luồng dữ liệu. Các sơ đồ này mô tả hệ thống được lập trình và sơ đồ cấu trúc các bộ phận và hiển thị theo sơ đồ như một cây. Chúng ghi lại cách thức các yếu tố có thể được thực hiện trong một sơ đồ luồng dữ liệu như một sơ đồ cây chứa các đơn vị chương trình.

Một biểu đồ cấu trúc đưa ra các mối liên quan giữa các đơn vị chương trình mà không bao gồm bất kì thông tin nào về thứ tự kích hoạt các đơn vị này. Sử dụng 3 ký hiệu đưới đây để vẽ:

a) Tam giác thể hiện tên của đơn vị.

b) Mũi tên nối những tam giác này.

c) Mũi tên vòng, thể hiện tên của dữ liệu được đi qua và từ các yếu tố trong sơ đồ cấu trúc. Thông thường, mũi tên vòng được vẽ song song với mũi tên nối các tam giác trong sơ đồ.

Từ bất kì sơ đồ luồng dữ liệu phức tạp nào thì đều có thể tạo ra một số lượng các sơ đồ cấu trúc khác nhau.

Các sơ đồ cấu trúc lập ra từ các sơ đồ luồng dữ liệu thể hiện cấu trúc hệ thống ở mức đầu tiên, khi đó mỗi ô trong sơ đồ cấu trúc thể hiện một ý tưởng trong sơ đồ luồng dữ liệu. Đương nhiên là các mức độ sâu hơn cũng có thể được mô tả bằng cách sử dụng cùng kỹ thuật.

D.52 Các biện pháp có tính cấu trúc

Mục đích

Mục đích chính của các biện pháp có tính cấu trúc là tăng cường chất lượng của việc phát triển phần mềm bằng cách tập trung chú ý vào những bộ phận đầu tiên trong vòng đời. Các phương pháp nhằm đạt được điều này thông qua các quy trình, các dẫn giải ngắn gọn và mang tính trực giác (được hỗ trợ bởi máy tính) để xác định sự tồn tại của các yêu cầu và các tính năng hoạt động theo một thứ tự logic và yêu cầu có tính cấu trúc.

Mô tả

Một loạt các phương pháp cấu trúc đã có. Một số như SSADM, LBMS được thiết kế cho các chức năng xử lý dữ liệu truyền thống và các chức năng xử lý chuyển đổi, trong khi những phương pháp

khác (MASCOT, JSD, Yourdon thời gian thực) được định hướng thiên về kiểm soát quá trình và ứng dụng thời gian thực (có xu hướng thiên về các vấn đề quan trọng về an toàn).

Các phương pháp cấu trúc chủ yếu là các "chương trình mang tính tư duy" để nhận thức một cách có hệ thống và phân chia một vấn đề hay một hệ thống. Những tính năng chính này là:

- Thứ tự tư duy mang tính logic, phân nhỏ một vấn đề lớn thành các giai đoạn có thể quản lý được;
- Nhận dạng toàn bộ hệ thống, bao gồm môi trường cũng như hệ thống được yêu cầu;
- Phân chia dữ liệu và chức năng trong hệ thống được yêu cầu;
- Danh mục kiểm tra, ví dụ: danh mục các loại vấn đề cần phải xác định;
- Mức sử dụng trí tuệ thấp – đơn giản, trực giác, thực dụng.

Các diễn giải hỗ trợ có xu hướng ngắn gọn trong việc xác định các thành phần của vấn đề và hệ thống (ví dụ: các quá trình và các dòng chảy của dữ liệu), nhưng các chức năng xử lý được những thành phần này thực hiện lại có xu hướng được thể hiện bằng các diễn giải mang tính thông tin. Tuy nhiên, một số phương pháp lại sử dụng một phần các dẫn giải chính thống (toán học). (ví dụ: JSD sử dụng các thể hiện thông thường; Yourdon, SOM và SDL khai thác các cơ chế trình trạng hữu hạn). Quá trình rút gọn này không chỉ giảm bớt mức độ dễ bị hiểu nhầm mà còn đưa ra phạm vi cho việc xử lý tự động.

Những lợi ích khác của việc diễn giải có tính cấu trúc là khả năng nhìn thấy được của nó, cho phép người sử dụng kiểm tra quy định kỹ thuật hoặc thiết kế bằng trực giác, dựa trên năng lực của người đó chứ không phải là sự hiểu biết không được tuyên bố.

D.53 Lập trình theo cấu trúc

Mục đích

Để thiết kế và chạy chương trình theo cách có thể thực hiện được phân tích thành phần phần mềm. Phân tích này nên có khả năng tìm ra được tất cả các sự hoạt động của thành phần chính.

Mô tả

Thành phần phần mềm nên có độ phức tạp về cấu trúc là nhỏ nhất. Nên tránh việc phân nhánh phức tạp. Các ràng buộc lặp đi lặp lại và phân nhánh (nếu có thể) nên liên quan đơn giản đến các thông số đầu vào. Thành phần phần mềm này nên được chia thành các module nhỏ phù hợp, và sự tương tác của các module này nên được làm cho rõ ràng. Các tính năng của ngôn ngữ lập trình khuyến khích biện pháp tiếp cận trên nên được sử dụng ưu tiên hơn là sử dụng các tính năng được cho là tỏ ra hiệu quả hơn, ngoại trừ trường hợp ưu tiên hoàn toàn sự hiệu quả.

D.54 Ngôn ngữ lập trình phù hợp

Mục đích

Để hỗ trợ các yêu cầu của tiêu chuẩn này nhiều nhất có thể, đặc biệt trong việc lập trình phòng thủ, lập trình mạnh, lập trình theo cấu trúc và xác nhận khả năng. Ngôn ngữ lập trình được chọn nên hướng tới việc mã hóa dễ dàng thẩm tra được với tối thiểu công việc và tạo thuận lợi cho việc xây dựng, thẩm tra vào bảo trì chương trình.

Mô tả

Ngôn ngữ nên được xác định đầy đủ và rõ ràng. Ngôn ngữ nên định hướng tới người sử dụng hoặc vấn đề hơn là định hướng tới máy. Các ngôn ngữ được sử dụng rộng rãi hoặc các ngôn ngữ con của chúng được ưu tiên hơn là các ngôn ngữ có mục đích đặc biệt.

Để bổ sung cho các tính năng đã được tham chiếu, ngôn ngữ nên cung cấp:

- Cấu trúc khối;
- Kiểm tra thời gian chuyển dịch;
- Kiểm tra loại thời gian chạy và phạm vi giới hạn mảng;
- Kiểm tra thông số.

Ngôn ngữ lập trình nên khuyến khích:

- Sử dụng các thành phần nhỏ và có khả năng quản lý;
- Giới hạn việc truy cập dữ liệu trong các thành phần nhất định;
- Xác định các phạm vi phụ có thể thay đổi;
- Mọi loại cấu trúc hạn chế lỗi khác.

Ngôn ngữ nên được hỗ trợ từ một chương trình biên dịch phù hợp, các thư viện của các thành phần hiện có phù hợp, bộ xử lý và các chương trình đối với việc kiểm soát và phát triển phiên bản phần mềm.

Các tính năng gây khó khăn cho việc thẩm tra và do đó nên tránh là:

- Các bước nhảy không điều kiện không bao gồm các phép gọi thủ tục hoặc hàm;
- Vòng lặp, phép truy hồi;

- Con trỏ, khối hoặc mọi loại biến, đối tượng động;
- Can thiệp xử lý ở mức độ mã nguồn;
- Nhiều đầu vào hoặc đầu ra các vòng lặp, các khối hoặc các chương trình con;
- Triển khai hoặc khai báo các biến ẩn;
- Các bản ghi khác nhau và tương đương;
- Các thông số theo quy trình.

Các ngôn ngữ cấp thấp sẽ thể hiện các vấn đề theo đặc tính định hướng máy, đặc biệt là các ngôn ngữ lắp ghép.

D.55 Petri-Nets theo thời gian

Mục đích

Để lập mô hình các vấn đề liên quan tới tính năng hoạt động của hệ thống, để truy cập và có thể tăng cường độ an toàn và các yêu cầu về vận hành qua phân tích và thiết kế lại.

Mô tả

Petri-Nets thuộc loại mô hình lý thuyết sơ đồ mà phù hợp với việc thể hiện thông tin và luồng điều khiển trong các hệ thống có các hoạt động đồng thời và hoạt động không đồng bộ.

Petri-Net là một mạng lưới các vị trí và chuyển đổi. Các vị trí có thể là “có” hoặc “không”. Chuyển đổi là “cho phép” khi tất cả các vị trí đầu vào được xác định là có. Khi được cho phép, nó sẽ được phép (nhưng không bắt buộc) chuyển thành “hoạt động”. Nếu nó hoạt động, các kí hiệu đầu vào sẽ được loại bỏ, và mỗi vị trí đầu ra từ trình biên dịch sẽ lại được ký hiệu thay thế.

Các nguy cơ tiềm ẩn được thể hiện như là các trạng thái cụ thể (các đánh dấu) trong mô hình. Mạng Petri-Nets mở rộng cho phép các chỉ số theo thời gian của hệ thống được mô hình hóa. Mặc dù mạng Petri-Nets “phổ thông” tập trung vào các lĩnh vực luồng điều khiển, một vài mở rộng được đề xuất để kết hợp luồng dữ liệu vào trong mô hình.

D.56 Tổng duyệt / rà soát thiết kế

Mục đích

Để phát hiện ra các lỗi trong một số sản phẩm của quá trình xây dựng sớm nhất và kinh tế nhất có thể.

Mô tả

Ban tiêu chuẩn IEC/TC 56 đã ban hành Hướng dẫn rà soát thiết kế hình thức *(Guide on Formal Design Reviews)*, có sự mô tả tổng quát về các cách rà soát thiết kế hình thức, các đối tượng của chúng, chi tiết các cách rà soát thiết kế khác nhau, thành phần của nhóm rà soát thiết kế, các nhiệm vụ và trách nhiệm liên quan. Tài liệu của IEC cũng đưa ra các hướng dẫn tổng quát cho việc lập kế hoạch và quản lý rà soát thiết kế hình thức, cũng như các chi tiết cụ thể liên quan tới vai trò của các chuyên gia độc lập trong nhóm rà soát thiết kế.

IEC khuyến nghị "Việc rà soát thiết kế hình thức phải được thực hiện đối với tất cả các quá trình/sản phẩm mới, các ứng dụng mới, các xem xét cải tiến các sản phẩm hiện có và các quá trình sản xuất chế tạo ảnh hưởng tới chức năng, hiệu năng, độ an toàn, độ tin cậy, khả năng sẵn có để kiểm tra khả năng bảo trì, tính sẵn sàng, khả năng chi phí và các đặc điểm khác ảnh hưởng tới quá trình/sản phẩm cuối cùng, người sử dụng hoặc người mua hàng".

Việc kiểm thử tổng duyệt mã nguồn sẽ có một nhóm kiểm thử tổng duyệt lựa chọn một tập hợp nhỏ các trường hợp kiểm thử trên hồ sơ, đại diện cho tập hợp các đầu vào và các đầu ra tương ứng mong muốn cho chương trình. Dữ liệu kiểm thử sau đó được theo dõi thủ công qua cổng logic của chương trình.

D.57 Lập trình định hướng đối tượng

Mục đích

Để cho phép tạo ra các mẫu thử nhanh chóng, để dễ dàng sử dụng lại các thành phần phần mềm đã có, để ẩn được thông tin, để giảm khả năng gây lỗi trong suốt vòng đời, để giảm các công việc cần thiết trong giai đoạn bảo trì, để chia nhỏ các vấn đề phức tạp thành các vấn đề nhỏ dễ dàng quản lý, để giảm các sự phụ thuộc giữa các thành phần phần mềm, để tạo ra các ứng dụng có thể mở rộng dễ dàng hơn.

Mô tả

Lập trình định hướng đối tượng là một cách nghĩ cơ bản mới về phần mềm dựa trên các nền tảng đã có sẵn trong thế giới thực hơn là dựa trên nền tảng của những tính toán. Lập trình định hướng đối tượng sẽ tổ chức phần mềm như là một tập hợp các đối tượng mà kết hợp cả cấu trúc dữ liệu và tính năng hoạt động. Việc làm này trái ngược với việc lập trình thông thường khi mà cấu trúc dữ liệu và tính năng hoạt động chỉ được liên kết lỏng lẻo với nhau.

Đối tượng: một đối tượng sẽ bao gồm một vùng dữ liệu riêng và tập hợp các phép toán trên đối tượng – được gọi là các biện pháp. Các biện pháp có thể công khai hoặc bí mật. Không thành phần phần mềm nào được phép đọc hoặc thay đổi dữ liệu riêng của một đối tượng một cách trực tiếp. Mỗi thành phần phần mềm khác phải sử dụng các biện pháp công khai trên đối tượng đó để đọc và ghi dữ liệu trong các vùng dữ liệu riêng của đối tượng này.

Loại đối tượng: bằng cách quy định loại đối tượng (thường theo dạng xác định kiểu loại), có thể cho phép tạo ra một số lượng các đối tượng cùng loại, ví dụ: tất cả các quá trình tạo ra đều có vùng dữ liệu riêng và các biện pháp được xác định trong loại đối tượng đó.

Tính (đa) kế thừa: một loại đối tượng có thể kế thừa vùng dữ liệu riêng và các biện pháp của một (hoặc nhiều) trong các loại cao cấp (các loại đối tượng ở nhánh trên trong phân cấp), được cho phép thêm vào một số dữ liệu riêng để bổ sung cho các biện pháp hoặc để cải tiến tính năng thực hiện của các biện pháp để lại. Sử dụng sự tính đa kế thừa có thể tạo ra nhiều cây loại đối tượng.

Tính đa thức: cùng phép toán có thể cho kết quả khác nhau ở trên các loại đối tượng khác nhau, ví dụ: phép toán ghi đối tượng cuối sẽ ghi các giá trị cho đối tượng cuối đó và phép toán ghi đối tượng tệp sẽ ghi các giá trị cho đối tượng tệp đó.

Nhược điểm: ngôn ngữ lập trình định hướng đối tượng có thể dẫn tới một yêu cầu bổ sung cho các nguồn lực với một tác động tiêu cực tới hiệu năng của hệ thống.

D.58 Theo dõi theo vết

Mục đích

Mục đích của việc theo dõi theo vết là đảm bảo thể hiện tất cả các yêu cầu có thể được đáp ứng chính xác và không tạo ra tài liệu không thể theo dõi theo vết.

Mô tả

Việc theo dõi theo vết theo các yêu cầu phải là công việc chính trong quá trình thẩm định một hệ thống và phải đưa ra các phương pháp để cho phép chứng minh nó trong suốt các giai đoạn của vòng đời.

Việc theo dõi theo vết phải được xem xét có thể áp dụng cho cả các yêu cầu về chức năng và không chức năng và phải đề cập cụ thể tới:

a) Truy vết các yêu cầu theo thiết kế hoặc các đối tượng khác thỏa mãn nó.

b) Truy vết các đối tượng thiết kế theo các đối tượng thực hiện mà tạo ra chúng.

c) Truy vết các yêu cầu và các đối tượng thiết kế theo các đối tượng hoạt động và bảo trì được yêu cầu áp dụng trong quá trình sử dụng an toàn và chính xác hệ thống.

d) Truy vết các yêu cầu, các đối tượng thiết kế, hoạt động, vận hành và bảo trì theo các kế hoạch thẩm tra và kiểm thử và các chỉ dẫn kỹ thuật quyết định khả năng chấp nhận nó.

e) Truy vết các kế hoạch thẩm tra và kiểm thử và các chỉ dẫn kỹ thuật theo các báo cáo kiểm thử hoặc theo các báo cáo khác ghi lại các kết quả của việc ứng dụng chúng.

Khi các yêu cầu, các đối tượng thiết kế hoặc các đối tượng khác được triển khai thành một số các tài liệu riêng biệt, việc truy vết phải được duy trì trong cấu trúc tài liệu và theo yêu cầu phân cấp.

Đầu ra của quá trình theo dõi theo vết phải tùy thuộc vào việc quản lý cấu hình hình thức.

D.59 Lập trình mê ta (siêu lập trình)

Mục đích

Kỹ thuật lập trình mê ta cho phép các lập trình viên làm được nhiều việc hơn trong cùng một khoảng thời gian khi họ muốn thực hiện ghi tất cả các mã bằng thủ công.

Mô tả

Lập trình mê ta là việc ghi các chương trình trên máy tính mà các chương trình này sẽ ghi hoặc điều khiển các chương trình khác (hoặc chính nó) chẳng hạn như dữ liệu của chúng hoặc sẽ thực hiện một phần công việc trong thời gian biên dịch được thực hiện theo cách khác lúc chạy chương trình.

Ngôn ngữ được viết trong kỹ thuật lập trình mê ta được gọi là siêu ngôn ngữ. Ngôn ngữ của các chương trình mà được điều khiển được gọi là ngôn ngữ đối tượng. Khả năng của một ngôn ngữ lập trình mà trở thành siêu ngôn ngữ của chính nó được gọi là sự phản xạ hoặc tính phản xạ.

Sự phản xạ là một tính năng ngôn ngữ có giá trị để làm thuận lợi cho việc sử dụng lập trình mê ta. Việc có ngôn ngữ lập trình như kiểu loại dữ liệu bậc 1 (chẳng hạn như ngôn ngữ lập trình Lisp) cũng là rất hữu ích. Ngôn ngữ lập trình chung sẽ dẫn ra một thiết bị lập trình mê ta trong phạm vi một ngôn ngữ, theo hướng những ngôn ngữ này sẽ hỗ trợ nó.

Lập trình mê ta thường làm việc theo một trong hai cách. Cách đầu tiên là làm lộ ra bên trong của động cơ đang chạy bằng mã lập trình qua các giao diện lập trình ứng dụng (APIs). Cách tiếp cận thứ 2 là chạy động các biểu thức chuỗi mà chứa các câu lệnh lập trình. Dó đó, “chương trình có thể ghi chương trình”. Mặc dù đều có thể sử dụng cả hai cách tiếp cận trên nhưng xu hướng là sử dụng một trong hai cách.

D.60 Lập trình hướng thủ tục

Mục đích

Quy định các bước chương trình sẽ thực hiện để đạt được trạng thai mong muốn.

Mô tả

Lập trình hướng thủ tục dựa trên khái niệm về câu lệnh quy trình. Các quy trình, được hiểu là các thói quen, các chương trình con, các biện pháp hoặc các chức năng (không nên nhầm với các hàm toán

học, nhưng nó cũng tương tự như các hàm được sử dụng trong lập trình chức năng) đơn giản bao gồm việc thực hiện một loạt các bước tính toán. Bất cứ quy trình nào được đưa ra thì đều có thể được gọi tại bất cứ thời điểm nào trong khi chạy chương trình, bao gồm các quy trình khác hoặc chính nó.

D.61 Lược đồ hàm tuần tự

Mục đích

Mô tả các thuật toán chương trình dưới dạng lược đồ.

Mô tả

Các thành phần của lược đồ hàm tuần tự cho phép phân vùng một nhóm các thuật toán ứng dụng vào trong một tập hợp các bước và các quá trình chuyển đổi được liên kết với nhau bằng các đường dẫn trực tiếp. Ứng với từng bước là một tập hợp các hoạt động, và ứng với từng chuyển đổi là một điều kiện chuyển đổi. Vì các yếu tố của lược đồ hàm tuần tự yêu cầu lưu trữ thông tin trạng thái, nên chỉ những nhóm các thuật toán ứng dụng mà có thể được cấu trúc bằng việc sử dụng các yếu tố này mới chính là các khối chức năng.

Xem mục 2.6 trong tiêu chuẩn EN 61131-3:2003.

D.62 Sơ đồ bậc thang

Mục đích

Mô tả chương trình dưới dạng sơ đồ.

Mô tả

Xem mục 4.2 trong tiêu chuẩn EN 61131-3:2003.

D.63 Sơ đồ khối chức năng

Mục đích

Mô tả chức năng giữa các biến đầu vào và các biến đầu ra dưới dạng sơ đồ.

Mô tả

Xem mục 4.3 trong tiêu chuẩn EN 61131-3:2003.

D.64 Lược đồ trạng thái hoặc sơ đồ trạng thái

Mục đích

Mô tả hoạt động của một hệ thống dưới dạng sơ đồ.

Mô tả

Lược đồ trạng thái hay sơ đồ đồ trạng thái được sử dụng để mô tả hoạt động của một hệ thống. Các sơ đồ trạng thái mô tả các trạng thái có thể của một đối tượng chẳng hạn như các tình huống xảy ra. Trên từng sơ đồ thường sẽ biểu diễn các đối tượng của một loại độc lập và vết trạng thái khác nhau của các đối tượng trong hệ thống.

Sơ đồ trạng thái có thể được sử dụng để biểu diễn dưới dạng biểu đồ các cơ chế chuyển đổi trạng thái hữu hạn. Sơ đồ này được giới thiệu bởi Taylor Booth trong cuốn sách “lý thuyết máy tự động và máy tuần tự”. Một dạng biểu diễn khác là thông qua bảng chuyển đổi trạng thái.

Hình thức cơ bản của sơ đồ chuyển đổi trạng thái đối với một cơ chế chuyển đổi trạng thái là một sơ đồ chỉ dẫn.

D.65 Lập mô hình dữ liệu

Mục đích

Để tạo ra một mô hình dữ liệu.

Mô tả

Việc lập mô hình dữ liệu trong khoa học máy tính là quá trình tạo ra một mô hình dữ liệu bằng cách áp dụng các bản mô tải mô hình dữ liệu hình thức sử dụng các kỹ thuật lập mô hình dữ liệu.

Một mô hình dữ liệu trong kỹ thuật phần mềm là một mô hình cơ bản mô tả cách thức truy cập và thể hiện dữ liệu. Các mô hình dữ liệu được sử dụng chủ yếu để xác định các đối tượng dữ liệu và mối quan hệ giữa các đối tượng dữ liệu trong phạm vi quan tâm. Một số ứng dụng điển hình của các mô hình cơ sở dữ liệu bao gồm hỗ trợ việc xây dựng cơ sở dữ liệu và tạo điều kiện cho việc trao đổi dữ liệu trong một lĩnh vực quan tâm đặc biệt. Các mô hình dữ liệu được quy định theo ngôn ngữ lập mô hình dữ liệu.

D.66 Sơ đồ / biểu đồ luồng điều khiển

Mục đích

Mô tả hoạt động của hệ thống dưới dạng sơ đồ.

Mô tả

Trong khoa học máy tín, sơ đồ luồng điều khiển hay biểu đồ luồng điều khiển (CFG) là một cách diễn đạt, sử dụng các diễn giải biểu đồ, tất cả các đường chạy mà có thể được nghiên cứu kỹ lưỡng qua một chương trình trong quá trình chạy chương trình. Từng nút trong sơ đồ biểu diễn một khối cơ bản, ví dụ, phần tuyến tính của đoạn mã mà không có bất kỳ bước nhảy hay các mục tiêu nhảy nào; các mục tiêu nhảy bắt đầu một khối và các bước nhảy kết thúc một khối. Các cạnh trực tiếp được sử dụng để thể hiện các bước nhảy theo luồng điều khiển. Trong hầu hết các thể hiện, có 2 khối được kí hiệu đặc biệt: khối đầu vào, thông qua các cổng vào điều khiển vào trong sơ đồ luồng, và khối đầu ra, thông qua các nhánh luồng điều khiển.

Sơ đồ luồng điều khiển là cần thiết đối với việc tối ưu hóa chương trình biên dịch và các chương trình phân tích tĩnh.

Tính năng với cũng là một đặc tính sơ đồ hữu ích khác trong việc tối ưu hóa. Nếu một khối / sơ đồ con không được kết nối với sơ đồ con mà chứa khối đầu vào thì khối đó sẽ không chạy được trong bất cứ lượt chạy chương trình nào, và vì vậy nó là mã không chạy được (mã thừa); nó có thể được loại bỏ an toàn. Nếu khối đầu ra không chạy được từ khối đầu vào thì nó sẽ hiển thị một vòng lặp vô hạn. Một lần nữa mã chết và một số vòng lặp vô hạn có thể xảy ra ngay khi lập trình viên không viết mã rõ ràng theo cách: tối ưu hóa giống như việc sao chép (lan truyền) hằng số và gán hằng số theo tiến trình nhảy có thể nhập nhiều khối cơ bản thành một, làm cho các cạnh bị loại bỏ khỏi sơ đồ luồng điều khiển, ví dụ có thể làm mất kết nối các bộ phận của sơ đồ.

Thuật ngữ

Những thuật ngữ này được sử dụng phổ biến khi nói đến các sơ đồ luồng điều khiển.

Khối đầu vào

Khối mà qua đó tất cả các luồng điều khiển đi vào sơ đồ.

Khối đầu ra

Khối mà qua đó tất cả các luồng điều khiển ra khỏi sơ đồ.

D.67 Sơ đồ chuỗi

Mục đích

Mô tả sự tương tác giữa các quá trình hoặc các thành phần dưới dạng sơ đồ.

Mô tả

Sơ đồ chuỗi là một dạng sơ đồ tương tác, mà qua đó cho thấy cách thức các quá trình hoặc các thành phần hoạt động cùng nhau và theo thứ tự.

D.68 Các biện pháp chỉ dẫn dạng bảng

Mục đích

Mục đích là để đưa ra một biện pháp đã được tiêu chuẩn hóa trong việc xác định các chức năng hướng theo dữ liệu của hệ thống.

Mô tả

Diễn giải dạng bảng chẳng hạn như các bảng điều khiển tín hiệu là biện pháp chuẩn chỉ trong việc ghi lại các yêu cầu lắp đặt cụ thể đối với một hệ thống tín hiệu đường sắt.

Kỹ thuật này là phù hợp khi các kiểu loại mối quan hệ giữa các thành phần của hệ thống được tiêu chuẩn hóa.

Ưu điểm: định dạng của bảng trong từng lĩnh vực có thể đáp ứng như là một danh mục kiểm tra trong quá trình thẩm tra.

D.69 Ngôn ngữ lập trình chuyên dụng

Mục đích

Mục đích là để đưa ra biện pháp quy định chức năng của một hệ thống hướng dữ liệu bằng cách sử dụng các khái niệm và các thuật ngữ dễ dàng sử dụng bởi các kỹ sư ứng dụng là những người không quen với ngôn ngữ lập trình truyền thống.

Mô tả

Một ngôn ngữ lập trình chuyên dụng điển hình kết hợp các cấu trúc điều khiển cũng giống như các ngôn ngữ lập trình cao cấp truyền thống với các toán tử mà chuyên dụng cho từng kiểu loại hệ thống.

Kỹ thuật này là phù hợp khi các các quyết định Boolean cần được quy định, nhưng có thể cũng được ứng dụng ở những vị trí khác.

Ưu điểm: linh hoạt, cho phép tạo ra dữ liệu trong các trường hợp bất thường mà không thể dự đoán trước khi hệ thống được thiết kế từ gốc.

D.70 Ngôn ngữ mô hình hóa hợp nhất - UML

Mục đích

Để biểu diễn các chương trình phần mềm và các tạo tác liên quan theo phương thức mà cho phép làm giảm độ phức tạp bằng các biện pháp cơ bản. Bằng cách cho phép lập mô hình một thiết kế hiện có hay đã lên kế hoạch theo nhiều loại sơ đồ, ngôn ngữ mô hình hóa hợp nhất làm thuận lợi cho việc

đánh giá các đặc tính kỹ thuật chính của thiết kế trên nền của các biểu diễn ở các cấp chi tiết phù hợp. UML thường được sử dụng trong quá trình phát triển hướng mô hình, được hỗ trợ bằng các sản phẩm thương mại. Phương thức phát triển này nhằm mục đích nâng cao chất lượng phần mềm và năng suất của những người phát triển bằng việc sử dụng các ngôn ngữ mô hình hóa cấp cao.

Mô tả

UML là một ngôn ngữ mô hình hóa đa năng tiêu chuẩn, bắt nguồn từ việc sử dụng các ngôn ngữ chỉ dẫn phần mềm định hướng bằng sơ đồ. Dựa vào truyền thống này, UML đã tái sử dụng nhiều khái niệm và biện pháp của các ngôn ngữ lập trình có trước. Những mô hình này được lập theo một hoặc nhiều loại sơ đồ, được phân loại như là các sơ đồ cấu trúc và các sơ đồ hoạt động, the latter cũng bao gồm bốn loại sơ đồ được phân loại như là các sơ đồ tương giao.

Các sơ đồ cấu trúc:

- Sơ đồ gói (nhóm phần tử): thể hiện nội dung của các mối quan hệ giữa các gói khác nhau, từng gói thì chứa các phần tử của mô hình liên quan.

- Sơ đồ lớp: quy định các loại đối tượng với các tính năng khác nhau của chúng và mối quan hệ của chúng với các loại đối tượng khác, dựa trên sự thích nghi của các sơ đồ thực thể kết hợp.

- Sơ đồ đối tượng: thể hiện cách thức các đối tượng khác nhau liên quan đến nhau.

- Sơ đồ cấu trúc hỗn hợp: thể hiện cấu trúc bên trong của một bộ phân loại (chẳng hạn như một lớp hoặc một thành phần) và các điểm tương giao của nó với các bộ phận khác của hệ thống.

- Sơ đồ thành phần: thể hiện các thành phần mà bao gồm hệ thống, các giao diện bên ngoài, các tương tác và mối quan hệ của các thành phần.

- Sơ đồ triển khai: quy định cách thức phần mềm được phân phối qua các nền tảng chạy chương trình.

Các sơ đồ hoạt động:

- Sơ đồ hoạt động: mô tả hoạt động của thuật toán, bằng việc sử dụng sự thích nghi của sơ đồ truyền thống mà cho phép việc lập mô hình chuyển đổi dữ liệu và chạy đồng thời.

- Sơ đồ chuyển đổi trạng thái: mô tả các hoạt động hướng theo tình huống bằng biện pháp cơ chế chuyển đổi trạng thái hữu hạn (sơ đồ chuyển đổi trạng thái).

- Sơ đồ chức năng: đưa ra cách nhìn bao quát (từ trên xuống) cách sử dụng của hệ thống cũng như cách nhìn hệ thống từ bên ngoài. Sơ đồ này hiển thị những chức năng của hệ thống hoặc các lớp

và tương tác của hệ thống với thế giới bên ngoài. Sơ đồ cũng được dùng trong quá trình phân tích hệ thống để nắm bắt được yêu cầu của hệ thống và hiểu được sự hoạt động của hệ thống.

- Sơ đồ tương tác (sơ đồ giao tiếp, sơ đồ tương tác tổng quan, sơ đồ chuỗi, sơ đồ thời gian): mô tả các tình huống bao gồm các hoạt động được thực hiện bởi các đối tượng giao tiếp.

Trong khi UML là ngôn ngữ lập mô hình tổng quát, các giải thích chuyên dụng có thể được tạo ra bằng các mẫu. Bằng việc cải tiến các khái niệm UML tiêu chuẩn, các mẫu có thể tạo ra các diễn giải bằng cách sử dụng các sự mở rộng được xác định trong mẫu đó. Bằng cách này, UML được sử dụng như là nền tàng để xác định các ngôn ngữ lập trình chuyên dụng.

D.71 Ngôn ngữ lập trình chuyên dụng

Mục đích

Để thể hiện các chương trình phần mềm và các tạo tác liên quan theo một ngôn ngữ phù hợp cho một lĩnh vực cụ thể.

Mô tả

Ngôn ngữ lập trình chuyên dụng (DSL) là một ngôn ngữ lập trình, chỉ dẫn kỹ thuật hoặc lập mô hình được tạo ra chuyên để xử lý các vấn đề theo một lĩnh vực ứng dụng cụ thể hoặc lĩnh vực vấn đề hoặc với một kỹ thuật cụ thể. Ngôn ngữ này dựa trên các khái niệm và các tính năng liên quan tới lĩnh vực này. Các ngôn ngữ chuyên dụng cũng được biết đến như là các ngôn ngữ lập trình với mục đích đặc biệt trải ngược với các ngôn ngữ lập trình với mục đích chung hoặc các ngôn ngữ lập mô hình như Java và UML.

Một trong những lợi ích quan trọng của các ngôn ngữ lập trình chuyên dụng là để diễn đạt và xử lý các vấn đề trong một lĩnh vực cụ thể mà không cần hiểu về lập mô hình, chỉ dẫn kỹ thuật, lập trình. Vì một hậu quả, các chương trình, các chỉ dẫn kỹ thuật hoặc các mô hình có thể được tạo ra ở cấp cao hơn, có thể từ người sử dụng cuối cùng. Bằng cách đưa ra các cấu trúc phù hợp với lĩnh vực này và các biện pháp có thể để tạo ra đoạn mã tự động, ngôn ngữ lập trình chuyên dụng cũng làm tăng năng suất làm việc của lập trình viên và chất lượng sản phẩm tạo ra. Việc tạo mã được thực hiện chủ yếu giống như bộ tạo ứng ứng sử dụng DSL như là đầu vào.