**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

---------------o0o---------------

****

**Báo cáo môn học**

**Chủ đề: Đặc tả độ tin cậy và bảo mật**

**GVHD: Lương Mạnh Bá**

**Nhóm sinh viên thực hiện:**

**Vũ Bích Ngọc 20162974**

**Trần Đức Sơn 20163571**

**Nguyễn Huy Thái 20163680**

**HÀ NỘI, 5/2020**

Nội dung

[Mục tiêu của phần này 2](#_Toc37835976)

[I. Đặc tả yêu cầu định hướng theo rủi ro 3](#_Toc37835977)

[II. Đặc tả tính an toàn 6](#_Toc37835978)

[1. Xác định nguy cơ 6](#_Toc37835979)

[2. Đánh giá rủi ro 7](#_Toc37835980)

[3. Phân tích mối nguy hiểm 10](#_Toc37835981)

[4. Giảm thiểu nguy cơ 12](#_Toc37835982)

[III. Đặc tả độ tin cậy 13](#_Toc37835983)

[1. Chỉ số độ tin cậy 15](#_Toc37835984)

[2. Yêu cầu độ tin cậy phi chức năng 17](#_Toc37835985)

[3. Đặc tả độ tin cậy chức năng 21](#_Toc37835986)

[IV. Đặc tả tính an ninh 22](#_Toc37835987)

[V. Đặc tả hình thức 26](#_Toc37835988)

[Những đặc điểm chính 29](#_Toc37835989)

# Mục tiêu của phần này

Mục tiêu của ***phần 12: Đặc tả độ tin cậy và bảo mật*** là giải thích cách đặc tả độ tin cậy và yêu cầu bảo mật một cách quy tắc và không quy tắc. Khi đọc chapter này, bạn có thể:

* hiểu làm thế nào cách tiếp cận theo rủi ro có thể được sử dụng cho định nghĩa và phân tích độ an toàn, độ tin cậy và yêu cầu bảo mật;
* hiểu được làm thế nào những cây lỗi được sử dụng để phân tích rủi ro và tìm ra được những yêu cầu về tính an toàn;
* được giới thiệu về thước đo của những đặc tả về độ tin cậy và làm thế nào chúng được sử dụng để đặc tả yêu cầu về độ tin cậy có thể đo đếm được;
* phân biệt những loại khác nhau về yêu cầu bảo mật có thể được đòi hỏi trong những hệ thống lớn;
* Chú ý đến những lợi thế và bất lợi khi sử dụng những đặc tả đúng luật hoặc về mang tính toán học của hệ thống

Các nội dung chính bao gồm:

* **Đặc tả yêu cầu định hướng theo rủi ro**
* **Đặc tả tính an toàn**
* **Đặc tả độ tin cậy**
* **Đặc tả tính an ninh**
* **Đặc tả hình thức**

## Đặc tả yêu cầu định hướng theo rủi ro

Vào tháng 9 năm 1993, một máy bay hạ cánh xuống sân bay Warsaw ở Ba lan trong cơn bão. Khoảng 9 giây sau khi hạ cánh, những chiếc phanh trong hệ thống phanh được điều khiển bởi máy tính của máy bay không hoạt động. Hệ thống phanh không nhận diện được rằng máy bay đã hạ cánh và tưởng rằng máy bay vẫn ở trên không. Một tính năng an toàn trên máy bay đã dừng hệ thống đẩy ngược đã triển khai, làm cho máy bay chậm dần, bởi vì cái này khá nguy hiểm nếu máy bay ở trên không. Máy bay chạy đến cuối đường bay và đâm vào một ngân hàng trên mặt đất, bốc cháy.

Điều tra về vụ tai nạn cho thấy hệ thống phanh hoạt động đúng như nó được lập trình. Không có lỗi lập trình. Tuy nhiên, đặc tả phần mềm không đầy đủ, chưa tính đến những trường hợp ít gặp phát sinh. Phần mềm hoạt động nhưng hệ thống bị lỗi.

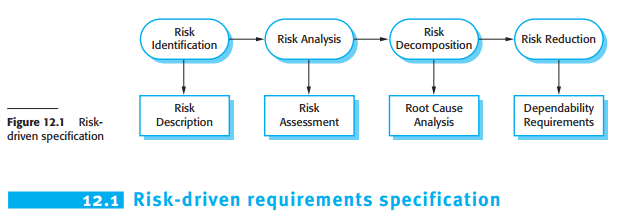
Điều này minh họa rằng độ tin cậy của hệ thống không chỉ phụ thuộc vào kỹ thuật tốt. Nó còn đòi hỏi phải chú ý đến các chi tiết yêu cầu hệ thống và yêu cầu phần mềm phải đảm bảo độ tin cậy và bảo mật của hệ thống. Những yêu cầu về độ tin cậy và bảo mật có hai loại:

1. *Các yêu cầu về chức năng, trong đó xác định các cơ sở kiểm tra và phục hồi cần được đưa vào hệ thống và các tính năng cung cấp bảo vệ chống lại các lỗi hệ thống và các cuộc tấn công bên ngoài.*
2. *Những yêu cầu dạng phi chức năng, xác định độ tin cậy và tính sẵn có của hệ thống*

Khởi đầu khi tạo các yêu cầu về độ tin cậy và yêu cầu bảo mật thường là các quy tắc, chính sách, quy định về kinh doanh cấp cao hoặc tên miền. Đây là những yêu cầu cấp cao được mô tả là “*shall not*” (không nên). Ngược lại, những yêu cầu chức năng thông thường xác định những gì hệ thống sẽ làm, yêu cầu *‘shall not’* định nghĩa hành vi của hệ thống không được chấp nhận. các ví dụ về yêu cầu ‘*shall not’* là

* *“Hệ thống không nên cho phép những người sử dụng được chỉnh sửa quyền truy cập của bất kỳ file nào mà họ không tạo”*
* *“Hệ thống không nên cho phép chế độ đẩy ngược được lựa chọn khi máy bay đang bay (độ an toàn)”*
* *“Hệ thống không nên cho phép kích hoạt đồng thời của nhiều hơn 3 tín hiệu báo động (độ an toàn)”*

Những yêu cầu ‘*shall not’* này không được áp dụng trực tiếp nhưng phải được phân tách thành các yêu cầu chức năng phần mềm cụ thể hơn. Ngoài ra, chúng được thực hiện thông qua các quyết định được thiết kế trong hệ thống như quyết định sử dụng loại thiết bị cụ thể trong hệ thống.



Các yêu cầu về độ tin cậy và bảo mật được coi là các yêu cầu bảo vệ. Chúng chỉ định cách hệ thống tự bảo vệ bản thân khỏi các lỗi bên trong, ngăn chặn các lỗi hệ thống gây thiệt hại cho môi trường của nó, ngăn chặn các tai nạn hoặc tấn công từ môi trường của hệ thống, làm hong hệ thống và tạo điều kiện phục hồi trong trường hợp hỏng hóc. Để nghiên cứu những yêu cầu bảo vệ này, bạn cần hiểu những rủi ro đối với hệ thống và môi trường của nó. Tiếp cận hướng rủi ro đối với đặc tả yêu cầu có tính đến những sự kiện nguy hiểm có thể xảy ra, xác suất xảy ra những sự kiện này, xác suất thiệt hại sẽ xảy ra do sự kiện đó và mức độ thiệt hại gây ra. Các yêu cầu về bảo mật và độ tin cậy sẽ được thiết lập sau đó, dựa trên phân tích nguyên nhân có thể gây ra các sự kiện nguy hiểm.

Tiến trình đặc tả hướng rủi ro chính liên quan đến việc hiểu về việc đối mặt với rủi ro của hệ thống, khám phá các nguyên nhân gốc rễ, và tạo ra các yêu cầu để quản lý những rủi ro này. Những giai đoạn trong tiến trình này bao gồm:

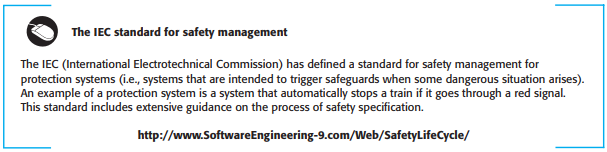
1. ***Xác định Nguy cơ***Những rủi ro tiềm năng tới hệ thống được xác định. Những thứ đó phụ thuộc vào môi trường mà hệ thống sử dụng. Rủi ro có thể phát sinh từ tương tác giữa hệ thống và những điều kiện hiếm xảy ra khi thao tác với môi trường. Tai nạn tại Warsaw mà chúng tối mô tả trước đó xảy ra khi cơn bão tạo ra ngược gió làm cho máy bay bị nghiêng vì vậy (không thường thấy) nó đáp đất bằng một bánh xe.
2. ***Phân tích rủi ro và phân loại***Mỗi một rủi ro được xem xét riêng. Những rủi ro nào nghiêm trọng và không hợp lý được chọn để phân tích thêm. Ở giai đoạn này, rủi ro có thể được loại bỏ vì chúng không có khả năng phát sinh hoặc do phần mềm không thể phát hiện được (ví dụ: phản ứng dị ứng với cảm biến trong hệ thống bơm insulin)
3. ***Phân tách rủi ro***Mỗi rủi ro được phân tích để khám phá ra nguyên nhân gốc rễ của rủi ro đó. Những nguyên nhân đó là lý do tại sao một hệ thống có thể chạy lỗi. Chúng có thể là lỗi phần mềm hoặc phần cứng hoặc lỗ hổng cố hữu từ quá trình quyết định thiết kế hệ thống.
4. ***Giảm thiểu nguy cơ***Đề xuất cách thức các rủi ro có thể được loại bỏ hoặc giảm đi. Điều này góp phần vào các yêu cầu về độ tin cậy của hệ thống xác định các biện pháp phòng vệ chống lại các rủi ro và cách thức quản lý các rủi ro.

Đối với các hệ thống lớn, phân tích rủi ro có thể được cấu trúc thành các giai đoạn (Leveson, 1995), trong đó mỗi giai đoạn xem xét các loại rủi ro khác nhau:

1. ***Phân tích sơ bộ rủi ro***, trong đó xác định các rủi ro chính từ môi trường hệ thống. Chúng độc lập với công nghệ được sử dụng để phát triển hệ thống. Mục đích phân tích rủi ro sơ bộ là phát triển một tập hợp các yêu cầu bảo mật và độ tin cậy ban đầu cho hệ thống.
2. ***Phân tích vòng đời rủi ro***, diễn ra trong quá trình phát triển hệ thống và chủ yếu liên quan đến các rủi ro phát sinh từ các quyết định thiết kế hệ thống. Các công nghệ và kiến trúc hệ thống khác nhau có những rủi ro liên quan riêng. Ở giai đoạn này, bạn nên mở rộng các yêu cầu để bảo vệ chống lại những rủi ro này.
3. ***Phân tích cách thức rủi ro hoạt động***, liên quan đến giao diện người dùng hệ thống và rủi ro từ lỗi vận hành. Một lần nữa, một khi các quyết định đã được đưa ra về thiết kế giao diện người dùng, các yêu cầu bảo vệ hơn nữa có thể phải được thêm vào.

Các giai đoạn trên rất cần thiết bởi vì không thể đưa ra tất cả các quyết định về độ tin cậy và bảo mật mà không có thông tin đầy đủ về việc triển khai hệ thống. Yêu cầu bảo mật và độ tin cậy bị ảnh hưởng đặc biệt bởi các lựa chọn công nghệ và quyết định thiết kế. Kiếm tra hệ thống có thể phải được đưa vào để đảm bảo rằng các thành phần của bên thứ ba đã hoạt động chính xác. Các yêu cầu bảo mật có thể phải được sửa đổi vì chúng mâu thuẫn với các tính năng bảo mật được cung cấp bởi một hệ thống sẵn có.

* Ví dụ, một yêu cầu bảo mật có thể là *những người dùng nên tự xác định hệ thống bằng cách sử dụng cụm từ thông qua thay vì một mật khẩu*. Các cụm từ được coi là an toàn hơn mật khẩu. Chúng khó bị kẻ tấn công đoán hoặc sử dụng những hệ thống bẻ khóa mật khẩu tự động. Tuy nhiên, nếu một quyết định được đưa ra sử dụng một hệ thống hiện có chỉ hỗ trợ xác thực bằng mật khẩu, thì yêu cầu bảo mật này không được hỗ trợ. Có thể cần bổ sung chức năng trong hệ thống để bù đắp cho những rủi ro gia tăng khi sử dụng mật khẩu thay vì một cụm từ được thông qua.



Chuẩn IEC (International Electrotechnical Commission) được định nghĩa là một chuẩn quản lý độ an toàn trong hệ thống bảo mật (ví dụ hệ thống được dự định để kích hoạt bảo vệ khi một vài tình huống nguy hiểm phát sinh). Một ví dụ của hệ thống bảo vệ là một hệ thống dừng một cách tự động tàu hỏa khi nó đi quan một tín hiệu đèn đỏ. Những chuẩn này bao gồm các hướng dẫn mang tính mở rộng trong quá trình xử lý của đặc tả an toàn.

## Đặc tả tính an toàn

Hệ thống đặc biệt an toàn (Safety-critical) là một hệ thống mà những lỗi xảy ra có ảnh hưởng với môi trường của hệ thống và gây thương tổn hoặc cái chết cho con người trong môi trường. Mối quan tâm chính của đặc tả an toàn là xác định các yêu cầu sẽ giảm thiểu khả năng xảy ra lỗi hệ thống như vậy. Những hoạt động trong tiến trình đặc tả dựa vào nguy cơ cơ bản, được cho trong Hình 12.1, ứng với quy trình đặc tả an toàn như sau:

1. ***Nhận dạng mối nguy*** Trong đặc tả tính an toàn, đây là quá trình nhận dạng những nguy hiểm khi coi những nguy hiểm này ảnh hưởng đến hệ thống.
2. ***Phân tích nguy cơ***Đây là một quá trình đánh giá mối nguy để quyết định mối nguy hiểm nào là nguy hiểm nhất và/hoặc có khả năng xảy ra nhất. Chúng nên được ưu tiên khi đạt được các yêu cầu an toàn.

***Phân tách rủi ro*** Quá trình này liên quan đến việc nghiên cứu khám phá những sự kiện có thể dẫn đến sự xuất hiện của một mối nguy hiểm. Trong đặc tả tính an toàn, quá trình này được gọi là phân tích nguy cơ.

1. ***Giảm thiểu rủi ro***Quá trình này dựa trên kết quả phân tích mối nguy và dẫn đến việc xác định các yêu cầu an toàn. Những điều này có thể liên quan đến việc đảm bảo rằng mối nguy hiểm không phát sinh hoặc dẫn đến tai nạn hoặc nếu tai nạn xảy ra, thiệt hại sẽ được giảm thiểu.

### Xác định nguy cơ

Trong một hệ thống safety-critical (tạm dịch là đặc biệt an toàn), những rủi ro chính đến từ những mối nguy hiểm có thể dẫn đến tai nạn. Bạn có thể giải quyết vấn đề xác định mối nguy bằng cách xem xét các loại nguy cơ khác nhau, như mối nguy vật lý, mối nguy về điện tử, mối nguy về sinh học, mối nguy về bức xạ, mối nguy về sự cố dịch vụ, v.v.. Mỗi một lớp đó sau đó có thể được phân tích và khám phá ra các mối nguy cụ thể. Sự kết hợp có thể có của các mối nguy hiểm có khả năng gây nguy hiểm cho hệ thống cũng phải được xác định.

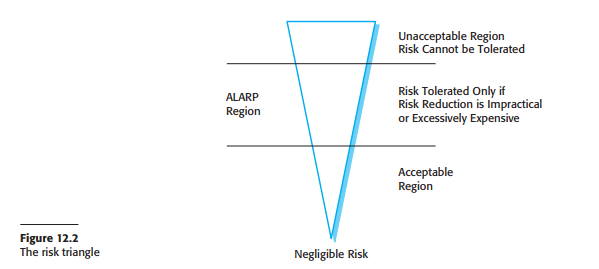
* Ví dụ: Hệ thống bơm insulin mà đã được đề cập trong ví dụ trước là một hệ thống quan trọng về an toàn, bởi vì thất bại có thể gây thương tích hoặc thậm chí tử vong cho người sử dụng hệ thống. Các tai nạn có thể xảy ra khi sử dụng chiếc máy này bao gồm người dùng phải chịu hậu quả lâu dài do kiểm soát lượng đường trong máu kém (các vấn đề về mắt, tim, thận); rối loạn chức năng nhận thức do mức độ đường trong máu thấp; hoặc sự xuất hiện của một số điều kiện y tế khác chẳng hạn như phản ứng dị ứng.
* Một số nguy cơ trong hệ thống bơm insulin:
* Tính toán quá liều insulin (phục vụ thất bại);
* Tính toán dưới mức quy định liều insulin (phục vụ thất bại);
* Lỗi trong hệ thống hiển thị phần cứng (phục vụ thất bại);
* Lỗi nguồn bị mất điện do hết pin (điện);
* Can thiệp điện với các thiết bị y tế khác như máy trợ tim (điện);
* Tiếp xúc cảm biến và bộ truyền động kém do lắp không đúng (vật lý);
* Các bộ phận của máy bị vỡ trong cơ thể bệnh nhân (vật lý);
* Nhiễm trùng gây ra bởi giới thiệu của máy (sinh học);
* Phản ứng dị ứng với các vật liệu hoặc insulin được sử dụng trong máy (sinh học);

Các kỹ sư giàu kinh nghiệm, làm việc với các chuyên gia tên miền và cố vấn an toàn chuyên nghiệp, xác định các mối nguy hiểm từ kinh nghiệm trước đó và từ một phân tích về tên miền ứng dụng. Các kỹ thuật làm việc nhóm như brainstorming (động não) có thể được sử dụng, trong đó một nhóm người trao đổi ý tưởng. Đối với hệ thống bơm insulin, những người có thể tham gia bao gồm bác sĩ, nhà vật lý y tế, kỹ sư, và nhà thiết kế phần mềm.

Những nguy cơ liên quan đến phần mềm thường liên quan đến việc không cung cấp dịch vụ hệ thống hoặc với sự thất bại của các hệ thống giám sát và bảo vệ. Hệ thống giám sát và bảo vệ được bao gồm trong một thiết bị để phát hiện các điều kiện, chẳng hạn như mức pin thấp, có thể dẫn đến hỏng thiết bị.

### Đánh giá rủi ro

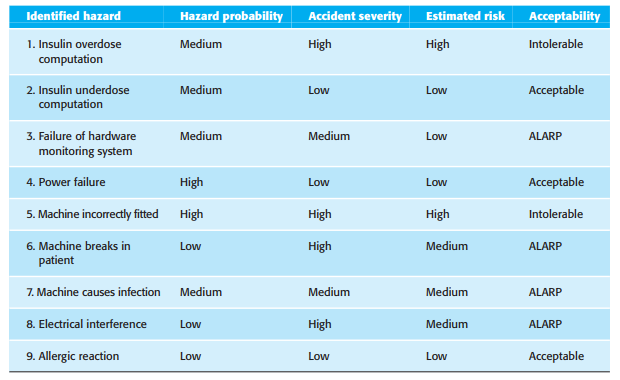
Quá trình đánh giá rủi ro tập trung vào hiểu tính xác suất mà rủi ro (hay nguy cơ) này sẽ xảy ra và hậu quả nếu tai nạn hoặc sự cố liên quan đến nguy cơ đó sẽ xảy ra. Bạn cần thực hiện phân tích này để hiểu liệu rủi ro này có là một mối nguy hại nghiêm trọng tới hệ thống hoặc tài nguyên hay không. Hoạt động phân tích cung cấp một cơ sở để quyết định cách quản lý rủi ro liên quan đến nguy cơ.

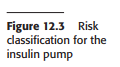


Đối với mỗi mối nguy cơ, kết quả của quá trình phân tích và phân loại là một khẳng định về khả năng chấp nhận được. Điều này được thể hiện dưới dạng rủi ro, trong đó rủi ro có tính đến khả năng xảy ra tai nạn và hậu quả của nó. Có ba loại rủi ro mà bạn có thể sử dụng trong đánh giá nguy cơ:

* **Rủi ro không thể chịu đựng** **(Intolerable risks**) trong các hệ thống đặc biệt an toàn là những nguy cơ đe dọa tính mạng con người. Hệ thống phải được thiết kế sao cho các mối nguy hiểm đó không thể phát sinh hoặc nếu có, các tính năng trong hệ thống sẽ đảm bảo rằng chúng được phát hiện trước khi chúng gây ra tai nạn.
* **Rủi ro thấp đến mức có thể chấp nhận được (ALARP)** là những rủi ro có hậu quả ít nghiêm trọng hoặc nghiêm trọng nhưng có xác suất xảy ra rất thấp. Hệ thống phải được thiết kế sao cho khả năng xảy ra tai nạn do rủi ro được giảm thiểu, tùy thuộc vào các cân nhắc khác như chi phí và vận chuyển.
* **Rủi ro chấp nhận được** là những trường hợp tai nạn liên quan thường dẫn đến các thiệt hại nhỏ. Các nhà thiết kế hệ thống nên thực hiện tất cả các bước có thể để giảm rủi ro có thể chấp nhận được, miễn là những điều này không làm tăng chi phí, thời gian giao hàng hoặc các thuộc tính hệ thống phi chức năng khác

Hình 12.2 (Brazendale and Bell, 1994), lập trình một hệ thống đặc biệt an toàn, cho thấy 3 khu vực. Hình dạng của sơ đồ phản ánh chi phí đảm bảo cho rủi ro không dẫn đến các sự cố hoặc tai nạn. Chi phí thiết kế hệ thống cần được đáp ứng





Để đối phó với những nguy cơ trong tam giác rủi ro. Chi phí cao nhất phát sinh do rủi ro ở đầu biểu đồ, chi phí thấp nhất phát sinh do rủi ro ở đỉnh tam giác.

Ranh giới giữa các khu vực trong Hình 12.2 không phải là vấn đề kỹ thuật mà phụ thuộc vào các yếu tố chính trị xã hội. Theo thời gian, xã hội đã trở nên nhiều mối nguy hiểm hơn nên ranh giới đã di chuyển xuống dưới. Mặc dù chi phí tài chính để chấp nhận rủi ro và chi trả cho bất kỳ tai nạn nào có thể thấp hơn chi phí phòng ngừa tai nạn, ý nghĩ phổ biến là dùng tiền đó để giảm khả năng xảy ra lỗi hệ thống, điều này có thể phát sinh chi phí.

* Ví dụ, công ty có thể chi một số tiền nhỏ hơn để làm sạch ô nhiễm trong trường hợp hiếm hoi xảy ra, thay vì cài đặt các hệ thống để ngăn ngừa ô nhiễm. Tuy nhiên, vì công chúng và báo chí sẽ không chấp nhận những tai nạn như vậy, việc khắc phục thiệt hại thay vì ngăn chặn tai nạn không được chấp nhận. Những tình huống như vậy cũng có thể dẫn đến việc phân loại lại rủi ro.
* Ví dụ: các rủi ro được cho là không thể xảy ra (và do đó trong khu vực ALARP) có thể được phân loại lại là không thể chịu đựng được do các sự kiện, chẳng hạn như các cuộc tấn công khủng bố hoặc tai nạn đã xảy ra.

Đánh giá mối nguy liên quan đến việc ước tính xác suất nguy hiểm và mức độ nghiêm trọng của rủi ro. Điều này thường khó khăn vì các mối nguy hiểm và tai nạn là không phổ biến vì vậy các kỹ sư có liên quan có thể không có kinh nghiệm giải quyết các sự cố hoặc tai nạn trước đó. Xác suất và mức độ nghiêm trọng được biểu thị bằng cách sử dụng các thuật ngữ tương đối, chẳng hạn như ‘có thể xảy ra’, ‘không thể xảy ra’, ‘hiếm khi’, và ‘cao’, ‘trung bình’, ‘thấp’. Chỉ có thể định lượng các thuật ngữ này nếu có đủ dữ liệu về tai nạn và sự cố để phân tích thống kê.

Hình 12.3 cho thấy sự phân loại rủi ro đối với các mối nguy được xác định trong phần trước của hệ thống phân phối insulin. Tôi đã phân tách các mối nguy liên quan đến việc tính toán insulin không chính xác thành quá liều insulin và thiếu insulin.

Quá liều insulin có khả năng nghiêm trọng hơn so với việc sử dụng ít liều lượng insulin hơn trong thời gian ngắn. Quá liều insulin có thể dẫn đến rối loạn chức năng nhận thức, hôn mê và cuối cùng là tử vong. Insulin thiếu dẫn đến lượng đường trong máu cao. Trong ngắn hạn, những điều này gây ra mệt mỏi nhưng không nghiêm trọng lắm; về lâu dài, chúng có thể dẫn đến các vấn đề nghiêm trọng về tim, thận và mắt.

Các mối nguy hiểm 4-9 trong Hình 12.3 không liên quan đến phần mềm, tuy nhiên phần mềm vẫn có vai trò trong việc phát hiện mối nguy. Phần mềm giám sát phần cứng sẽ theo dõi trạng thái hệ thống và cảnh báo các sự cố tiềm ẩn. Cảnh báo thường sẽ cho phép phát hiện nguy hiểm trước khi gây ra tai nạn. Ví dụ về các mối nguy hiểm có thể được phát hiện là sự cố mất điện, được phát hiện bằng cách theo dõi pin và một vị trí đặt máy không chính xác có thể được phát hiện bằng các tín hiệu theo dõi từ cảm biến đường trong máu.

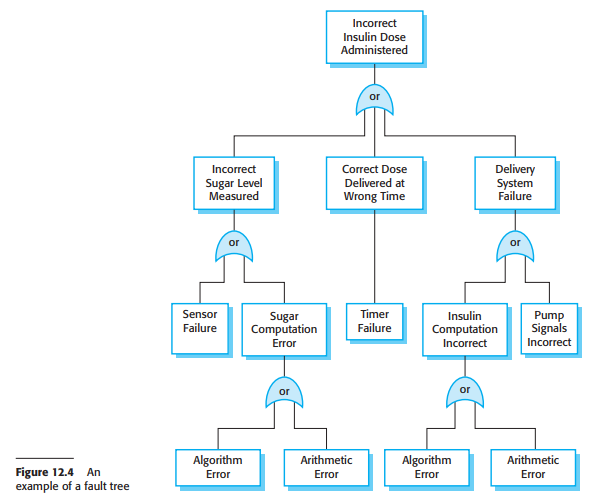
Các phần mềm giám sát trong hệ thống tất nhiên cũng có liên quan đến an toàn. Việc không phát hiện ra mối nguy hiểm có thể dẫn đến tai nạn. Nếu hệ thống giám sát bị lỗi nhưng phần cứng hoạt động chính xác thì đây không phải là một lỗi nghiêm trọng. Tuy nhiên, nếu hệ thống giám sát bị lỗi và lỗi phần cứng không thể được phát hiện, thì điều này có thể gây ra hậu quả nghiêm trọng hơn.

### Phân tích mối nguy hiểm

Phân tích mối nguy là quá trình khám phá nguyên nhân gốc rễ của các mối nguy hiểm trong một hệ thống an toàn. Mục đích là tìm hiểu những sự kiện hoặc sự kết hợp của các sự kiện có thể gây ra lỗi hệ thống dẫn đến nguy hiểm. Để làm điều này, bạn có thể sử dụng phương pháp từ trên xuống hoặc từ dưới lên. Các kỹ thuật suy diễn, từ trên xuống, có xu hướng dễ sử dụng hơn, bắt đầu với mối nguy hiểm và làm việc từ đó đến lỗi hệ thống có thể xảy ra. Các kỹ thuật quy nạp, từ dưới lên bắt đầu với một lỗi hệ thống được đề xuất và xác định những mối nguy hiểm nào có thể xảy ra từ sự thất bại đó.

Các kỹ thuật khác nhau đã được đề xuất khi có thể tiếp cận phân tích hoặc phân tích nguy cơ. Chúng được tóm tắt bởi Storey (1996). Chúng bao gồm đánh giá và danh sách kiểm tra, các kỹ thuật chính thức như phân tích mạng Petri (Peterson, 1981), formal logic (Jahanian và Mok, 1986) và phân tích cây lỗi (Leveson và Stolzy, 1987; Storey, 1996). Khi tôi không có đủ không gian để trình bày tất cả các kỹ thuật này ở đây, tôi tập trung vào một cách tiếp cận được sử dụng rộng rãi để phân tích mối nguy dựa trên fault tree. Để thực hiện phân tích fault tree, bạn bắt đầu với các mối nguy hiểm đã được xác định. Đối với mỗi mối nguy hiểm, sau đó bạn làm việc ngược trở lại để khám phá các nguyên nhân có thể gây ra mối nguy hiểm đó. Bạn đặt mối nguy hiểm ở vị trí root của cây và xác định các trạng thái hệ thống có thể dẫn đến nguy cơ đó. Đối với mỗi trạng thái này, bạn xác định các trạng thái hệ thống tiếp theo có thể dẫn đến chúng. Bạn tiếp tục phân tách cho đến khi bạn đạt được (các) nguyên nhân gốc của rủi ro. Các mối nguy hiểm mà chỉ có thể phát sinh từ sự kết hợp của nhiều nguyên nhân thường ít có khả năng dẫn đến tai nạn hơn các mối nguy hiểm với chỉ một nguyên nhân gốc rễ.

Hình 12.4 là một fault tree cho các mối nguy liên quan đến phần mềm trong hệ thống phân phối insulin có thể dẫn đến một liều insulin được cung cấp không đúng liều lượng. Trong trường hợp này, tôi đã hợp nhất quá liều insulin và thiếu liều lượng insulin vào một mối nguy hiểm duy nhất, đó là ‘liều insulin không đúng liều lượng’. Điều này làm giảm số lượng fault tree cần thiết. Tất nhiên, khi bạn chỉ định cách phần mềm sẽ phản ứng với mối nguy hiểm này, bạn phải phân biệt giữa việc sử dụng insulin quá liều và thiếu lượng insulin cần thiết. Chúng không nghiêm trọng như nhau trong thời gian ngắn, dùng quá liều nghiêm trọng hơn.



Từ hình 12.4, bạn có thể thấy rằng:

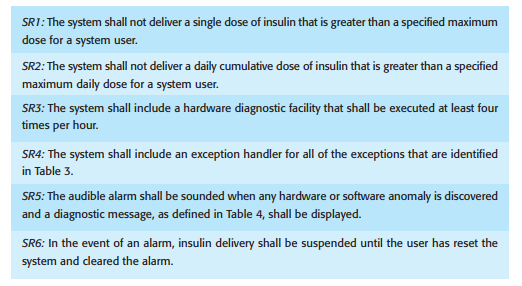
1. Có ba nguyên nhân có thể dẫn đến việc sử dụng sai liều lượng insulin. Lượng insulin được sử dụng sai liều lượng do quá trình tính toán lượng đường trong máu của người bệnh sai. Hệ thống phân phối có thể không đáp ứng chính xác với các lệnh chỉ định lượng insulin được tiêm. Hoặc cũng có thể tính toán chính xác liều lượng nhưng được sử dụng vào thời điểm quá sớm hoặc quá muộn.
2. Nhánh trái của fault tree, liên quan đến việc đo lượng đường trong máu không chính xác, xem xét điều này có thể xảy ra như thế nào. Điều này có thể xảy ra do cảm biến cung cấp đầu vào để tính toán mức đường đã thực hiện lỗi hoặc do việc tính toán mức đường trong máu đã hoạt động không chính xác. Mức đường được tính từ một số thông số đo, chẳng hạn như độ dẫn của da. Tính toán không chính xác có thể xuất phát từ thuật toán không chính xác hoặc lỗi số học do sử dụng số dấu phẩy động.
3. Nhánh trung tâm của cây liên quan đến các vấn đề về thời gian và kết luận rằng những điều này chỉ có thể xảy ra do lỗi hẹn giờ hệ thống.
4. Nhánh phải của cây kiểm tra các nguyên nhân có thể gây ra lỗi liên quan đến lỗi hệ thống phân phối. Những điều này có thể là kết quả của việc tính toán không đúng yêu cầu insulin hoặc do không gửi được tín hiệu chính xác đến máy bơm cung cấp insulin. Một lần nữa, một tính toán không chính xác có thể xảy ra do lỗi thuật toán hoặc lỗi số học.

Cây fault tree cũng được sử dụng để xác định các vấn đề phần cứng tiềm ẩn. hardware fault tree có thể cung cấp cái nhìn sâu sắc về các yêu cầu cho phần mềm để phát hiện và có thể khắc phục những vấn đề này. Ví dụ, liều insulin không được dùng với tần suất rất cao, không quá hai hoặc ba lần mỗi giờ và đôi khi ít thường xuyên hơn mức này. Do đó, dung lượng bộ xử lý có sẵn để chạy các chương trình chẩn đoán và tự kiểm tra. Các lỗi phần cứng như cảm biến, bơm hoặc lỗi hẹn giờ có thể được phát hiện và cảnh báo trước khi chúng có ảnh hưởng nghiêm trọng đến bệnh nhân.

### Giảm thiểu nguy cơ

Khi các rủi ro tiềm ẩn và nguyên nhân gốc rễ của chúng đã được xác định, bạn có thể rút ra các yêu cầu an toàn để quản lý rủi ro và đảm bảo rằng các sự cố hoặc tai nạn không xảy ra. Có ba chiến lược có thể bạn có thể sử dụng:

1. ***Tránh nguy hiểm*** Hệ thống được thiết kế sao cho nguy cơ không thể xảy ra.
2. ***Phát hiện và loại bỏ mối nguy*** Hệ thống được thiết kế sao cho các mối nguy được phát hiện và vô hiệu hóa trước khi chúng gây ra tai nạn.
3. ***Hạn chế thiệt hại*** Hệ thống được thiết kế để giảm thiểu hậu quả của tai nạn.





Thông thường, các nhà thiết kế hệ thống quan trọng thường sử dụng kết hợp các phương pháp này. Trong một hệ thống quan trọng về an toàn, các mối nguy hiểm không thể chịu đựng được có thể được xử lý bằng cách giảm thiểu xác suất của chúng và thêm một hệ thống bảo vệ cung cấp một bản sao lưu an toàn.

* Ví dụ, trong một hệ thống điều khiển nhà máy hóa chất, hệ thống sẽ cố gắng phát hiện và tránh áp suất vượt quá trong lò phản ứng. Tuy nhiên, cũng có thể có một hệ thống bảo vệ độc lập theo dõi áp suất và mở van xả nếu phát hiện áp suất cao. Trong hệ thống phân phối insulin, trạng thái an toàn là trạng thái tắt máy trong đó không tiêm insulin. Đây không phải là mối đe dọa đối với sức khỏe của bệnh nhân tiểu đường nếu xét trong một khoảng thời gian ngắn. Xem xét các lỗi phần mềm có thể dẫn đến một liều insulin không chính xác, có thể rút ra một số loại lỗi:

1. ***Lỗi số học*** Xảy ra trong quá trình tính toán số học. Thông số kỹ thuật cần xác định tất cả các lỗi số học có thể xảy ra và nêu rõ rằng một trình xử lý ngoại lệ phải được đưa vào cho mỗi lỗi có thể xảy ra. Thông số kỹ thuật cần đặt ra hành động được thực hiện cho từng lỗi này. Hành động an toàn mặc định là tắt hệ thống phân phối và kích hoạt cảnh báo cảnh báo.
2. ***Lỗi thuật toán*** Khó khăn hơn do không có ngoại lệ chương trình rõ ràng được xử lý. Loại lỗi này có thể được phát hiện bằng cách so sánh liều insulin cần thiết được tính toán với liều được phân phối trước đó. Nếu nó cao hơn nhiều, điều này có thể có nghĩa là lượng insulin được tính toán không chính xác. Hệ thống cũng có thể theo dõi trình tự liều lượng. Sau khi một số liều trên trung bình đã được phân phối, có thể xuất hiện cảnh báo để giới hạn liều lượng tiếp theo sau đó được phân phối để cân bằng.

Một số yêu cầu an toàn thu được đối với phần mềm bơm insulin được thể hiện trong Hình 12.5. Đây là những yêu cầu của người dùng và chúng sẽ được thể hiện chi tiết hơn trong đặc tả yêu cầu hệ thống. Trong Hình 12.5, các tham chiếu đến Bảng 3 và 4 liên quan đến các bảng được bao gồm trong tài liệu yêu cầu, chúng không được hiển thị ở đây.

## Đặc tả độ tin cậy

Như đã thảo luận trong Chương 10, độ tin cậy tổng thể của một hệ thống phụ thuộc vào độ tin cậy phần cứng, độ tin cậy của phần mềm và độ tin cậy của các nhà quản trị hệ thống. Phần mềm hệ thống phải tính đến điều này. Cũng như bao gồm các yêu cầu bù cho lỗi phần mềm, cũng có thể có các yêu cầu về độ tin cậy liên quan để giúp phát hiện và phục hồi từ các lỗi phần cứng và lỗi vận hành.

Độ tin cậy khác với an toàn và bảo mật ở chỗ nó là thuộc tính hệ thống có thể đo lường được. Nghĩa là, có thể chỉ định mức độ tin cậy được yêu cầu, theo dõi hoạt động của hệ thống theo thời gian và kiểm tra xem độ tin cậy cần thiết đã đạt chưa. Ví dụ, một yêu cầu về độ tin cậy có thể là các lỗi hệ thống yêu cầu khởi động lại không nên xảy ra nhiều hơn tần suất một lần mỗi tuần.

Mỗi khi xảy ra lỗi như vậy, nó có thể được ghi lại và bạn có thể kiểm tra xem mức độ tin cậy cần thiết đã đạt được chưa. Nếu chưa, bạn có thể sửa đổi yêu cầu về độ tin cậy của mình hoặc gửi yêu cầu thay đổi để giải quyết các vấn đề hệ thống cơ bản. Bạn có thể quyết định chấp nhận mức độ tin cậy thấp hơn do chi phí thay đổi hệ thống cần thiết để cải thiện độ tin cậy lớn hoặc vì việc khắc phục sự cố có thể có tác dụng phụ bất lợi, chẳng hạn như hiệu suất hoặc thông lượng hệ thống bị thấp đi.

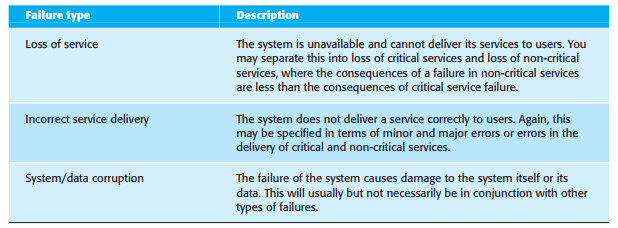
Ngược lại, cả an toàn và bảo mật đều là về việc tránh các tình huống không mong muốn, thay vì để một mức độ an toàn hoặc bảo mật đạt được cấp độ mong muốn. Ngay cả một tình huống như vậy trong vòng đời của một hệ thống có thể không được chấp nhận và nếu nó xảy ra, các thay đổi hệ thống phải được thực hiện. Thật vô nghĩa khi đưa ra những tuyên bố như lỗi hệ thống sẽ dẫn đến ít hơn 10 vết thương mỗi năm. Ngay sau khi một chấn thương xảy ra, vấn đề hệ thống phải được khắc phục.

Các yêu cầu về độ tin cậy có hai loại:

1. ***Các yêu cầu phi chức năng***, xác định số lượng lỗi có thể chấp nhận được trong quá trình sử dụng bình thường của hệ thống hoặc thời gian mà hệ thống không có sẵn để sử dụng. Đây là những yêu cầu về độ tin cậy có thể định lượng được.
2. ***Các yêu cầu chức năng***, xác định các chức năng của hệ thống và phần mềm để tránh, phát hiện hoặc chịu đựng các lỗi trong phần mềm và do đó đảm bảo rằng các lỗi này không dẫn đến lỗi hệ thống.

Yêu cầu độ tin cậy định lượng dẫn đến các yêu cầu hệ thống chức năng liên quan. Để đạt được một số mức độ tin cậy cần thiết, các yêu cầu về chức năng và thiết kế của hệ thống cần chỉ định các lỗi được phát hiện và các hành động cần thực hiện để đảm bảo rằng các lỗi này không dẫn đến lỗi hệ thống.

Quá trình đặc tả độ tin cậy có thể dựa trên quy trình đặc tả kỹ thuật theo rủi ro chung được thể hiện trong Hình 12.1:

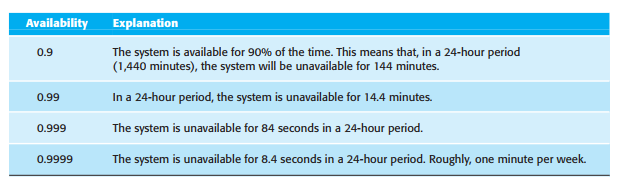
1. ***Xác định rủi ro*** Ở giai đoạn này, bạn xác định các loại lỗi hệ thống có thể dẫn đến thiệt hại kinh tế dưới một số loại. Ví dụ: hệ thống thương mại điện tử có thể không khả dụng để khách hàng đặt hàng hoặc lỗi làm hỏng dữ liệu có thể cần thời gian để khôi phục cơ sở dữ liệu hệ thống từ bản sao lưu và chạy lại các giao dịch đã được xử lý. Danh sách các loại lỗi có thể xảy ra, được cho trong Hình 12.6, có thể được sử dụng làm điểm bắt đầu để xác định rủi ro.
2. ***Phân tích rủi ro*** Liên quan đến việc ước tính chi phí và hậu quả của các loại lỗi phần mềm khác nhau và chọn các lỗi có hậu quả cao để phân tích thêm.
3. ***Phân tách rủi ro*** Ở giai đoạn này, bạn thực hiện phân tích nguyên nhân gốc rễ của các lỗi hệ thống nghiêm trọng và có thể xảy ra. Tuy nhiên, điều này có thể là không thể ở giai đoạn yêu cầu vì nguyên nhân gốc có thể phụ thuộc vào quyết định thiết kế hệ thống. Bạn có thể phải quay lại hoạt động này trong quá trình thiết kế và phát triển.
4. ***Giảm rủi ro*** Ở giai đoạn này, bạn nên tạo ra các thông số kỹ thuật về độ tin cậy định lượng, đưa ra xác suất chấp nhận được của các loại thất bại khác nhau. Tất nhiên, những điều này nên tính đến chi phí thất bại. Bạn có thể sử dụng các xác suất khác nhau cho các dịch vụ hệ thống khác nhau. Bạn cũng có thể tạo ra các yêu cầu về độ tin cậy chức năng. Một lần nữa, điều này có thể phải đợi cho đến khi quyết định thiết kế hệ thống được đưa ra



### Chỉ số độ tin cậy

Có hai số liệu quan trọng được sử dụng để chỉ định độ tin cậy cộng với số liệu bổ sung được sử dụng để chỉ định thuộc tính hệ thống có liên quan. Việc lựa chọn số liệu phụ thuộc vào loại hệ thống đang được chỉ định và các yêu cầu của miền ứng dụng. Các số liệu là:

1. ***Xác suất thất bại theo yêu cầu (POFOD)*** Nếu bạn sử dụng số liệu này, bạn xác định xác suất rằng nhu cầu dịch vụ từ hệ thống sẽ dẫn đến lỗi hệ thống. Vì vậy, POFOD = 0,001 có nghĩa là có 1 / 1.000 khả năng xảy ra lỗi khi có nhu cầu.
2. ***Tỷ lệ xảy ra lỗi (ROCOF)*** Số liệu này đưa ra số lỗi hệ thống có thể xảy ra có thể xảy ra liên quan đến một khoảng thời gian nhất định (ví dụ: một giờ) hoặc với số lần thực hiện hệ thống. Trong ví dụ trên, ROCOF là 1 / 1.000. Đối ứng của ROCOF là thời gian trung bình đến thất bại (MTTF), đôi khi được sử dụng làm thước đo độ tin cậy. MTTF là số đơn vị thời gian trung bình giữa các lỗi hệ thống được quan sát. Do đó, ROCOF của hai lần thất bại mỗi giờ ngụ ý rằng thời gian trung bình cho sự thất bại là 30 phút.
3. ***Tính khả dụng (AVAIL)*** Tính khả dụng của một hệ thống phản ánh khả năng cung cấp dịch vụ của nó khi được yêu cầu. AVAIL là xác suất mà một hệ thống sẽ hoạt động khi có nhu cầu về dịch vụ. Do đó, khả dụng là 0,9999, có nghĩa là, trung bình, hệ thống sẽ khả dụng với 99,99% thời gian hoạt động. Hình 12.7 cho thấy mức độ sẵn có khác nhau có ý nghĩa gì trong thực tế.





POFOD nên được sử dụng như một thước đo độ tin cậy trong các tình huống khi lỗi theo yêu cầu có thể dẫn đến lỗi hệ thống nghiêm trọng. Điều này áp dụng không phân biệt tần suất của các nhu cầu. Ví dụ, một hệ thống bảo vệ theo dõi lò phản ứng hóa học và tắt phản ứng nếu nó quá nóng nên có độ tin cậy được chỉ định bằng POFOD. Nói chung, yêu cầu đối với một hệ thống bảo vệ là không thường xuyên vì hệ thống này là tuyến phòng thủ cuối cùng, sau khi tất cả các chiến lược phục hồi khác đã thất bại. Do đó, POFOD 0,001 (1 lỗi trong 1.000 yêu cầu) có vẻ rủi ro, nhưng nếu chỉ có hai hoặc ba yêu cầu trên hệ thống, thì có lẽ bạn sẽ không bao giờ thấy lỗi hệ thống.

ROCOF là số liệu thích hợp nhất để sử dụng trong các tình huống trong đó nhu cầu về các hệ thống được thực hiện thường xuyên thay vì không liên tục. Ví dụ: trong một hệ thống xử lý một số lượng lớn giao dịch, bạn có thể chỉ định ROCOF là 10 lỗi mỗi ngày. Điều này có nghĩa là bạn sẵn sàng chấp nhận rằng trung bình 10 giao dịch mỗi ngày sẽ không hoàn thành thành công và sẽ phải bị hủy. Ngoài ra, bạn có thể chỉ định ROCOF là số lần thất bại trên 1.000 giao dịch.

Nếu thời gian tuyệt đối giữa các lần thất bại là quan trọng, bạn có thể chỉ định độ tin cậy là thời gian trung bình giữa các lần thất bại.

* Ví dụ: nếu bạn chỉ định độ tin cậy cần thiết cho một hệ thống có giao dịch dài (chẳng hạn như hệ thống thiết kế có sự trợ giúp của máy tính), bạn nên chỉ định độ tin cậy với thời gian trung bình dài đến thất bại. MTTF phải dài hơn nhiều so với thời gian trung bình mà người dùng làm việc trên các mô hình của mình mà không lưu kết quả của họ. Điều này có nghĩa là người dùng sẽ khó có thể mất việc do lỗi hệ thống trong bất kỳ phiên nào.

**Để đánh giá độ tin cậy của một hệ thống, bạn phải nắm bắt dữ liệu về hoạt động của nó. Dữ liệu cần thiết có thể bao gồm:**

1. Số lượng lỗi hệ thống đưa ra một số yêu cầu cho các dịch vụ hệ thống. Điều này được sử dụng để đo POFOD.
2. Thời gian hoặc số lượng giao dịch giữa các lần lỗi hệ thống cộng với tổng thời gian đã trôi qua hoặc tổng số giao dịch. Điều này được sử dụng để đo ROCOF và MTTF.
3. Thời gian sửa chữa hoặc khởi động lại sau khi hệ thống bị lỗi dẫn đến mất dịch vụ. Điều này được sử dụng trong việc đo lường sẵn có. Tính khả dụng không chỉ phụ thuộc vào thời gian giữa các lần hỏng mà còn phụ thuộc vào thời gian cần thiết để hệ thống hoạt động trở lại.

Đơn vị thời gian có thể được sử dụng là thời gian lịch hoặc thời gian xử lý hoặc đơn vị riêng biệt, chẳng hạn như số lượng giao dịch. Trong các hệ thống dành nhiều thời gian chờ đợi để đáp ứng yêu cầu dịch vụ, chẳng hạn như hệ thống chuyển mạch điện thoại, đơn vị thời gian nên sử dụng là thời gian xử lý. Nếu bạn sử dụng thời gian theo lịch, thì điều này sẽ bao gồm thời gian hệ thống không làm gì cả.

### Yêu cầu độ tin cậy phi chức năng

Các yêu cầu về độ tin cậy phi chức năng là các thông số kỹ thuật định lượng về độ tin cậy và tính sẵn có của một hệ thống, được tính bằng một trong các số liệu được mô tả trong phần trước. Độ tin cậy định lượng và đặc điểm kỹ thuật sẵn có đã được sử dụng trong nhiều năm trong các hệ thống đặc biệt an toàn nhưng chỉ hiếm khi được sử dụng trong các hệ thống quan trọng trong kinh doanh. Tuy nhiên, khi ngày càng có nhiều công ty yêu cầu phục vụ 24/7 từ hệ thống của họ, có khả năng các kỹ thuật như vậy sẽ ngày càng được sử dụng nhiều hơn.

**Có một số lợi thế trong việc đưa ra các đặc tả kỹ thuật về độ tin cậy định lượng:**

Quá trình quyết định mức độ tin cậy cần thiết giúp làm rõ những gì các bên liên quan thực sự cần. Nó giúp các bên liên quan hiểu rằng có rất nhiều loại lỗi hệ thống khác nhau và rõ ràng họ nhận ra rằng mức độ tin cậy cao là rất tốn kém để đạt được.

Nó cung cấp một cơ sở để đánh giá khi nào ngừng thử nghiệm một hệ thống. Bạn dừng lại khi hệ thống đã đạt được mức độ tin cậy cần thiết.

Nó là một phương tiện để đánh giá các chiến lược thiết kế khác nhau nhằm cải thiện độ tin cậy của một hệ thống. Bạn có thể đưa ra đánh giá về cách mỗi chiến lược có thể dẫn đến mức độ tin cậy cần thiết.

Nếu cơ quan quản lý phải phê duyệt một hệ thống trước khi nó đi vào hoạt động (ví dụ: tất cả các hệ thống quan trọng đối với an toàn bay trên máy bay đều được quy định), thì bằng chứng cho thấy mục tiêu độ tin cậy cần thiết đã được đáp ứng là quan trọng đối với chứng nhận hệ thống.

Để thiết lập mức độ tin cậy hệ thống cần thiết, bạn phải xem xét các tổn thất liên quan có thể xảy ra do lỗi hệ thống. Đây không chỉ đơn giản là tổn thất tài chính, mà còn tổn thất về mặt danh tiếng cho một doanh nghiệp. Mất danh tiếng có nghĩa là khách hàng sẽ đi nơi khác. Mặc dù tổn thất ngắn hạn từ lỗi hệ thống có thể tương đối nhỏ, nhưng tổn thất dài hạn có thể đáng kể hơn nhiều.

* Ví dụ: nếu bạn cố gắng truy cập một trang web thương mại điện tử và thấy rằng nó không phục vụ, bạn có thể sẽ tìm những gì bạn muốn ở nơi khác thay vì chờ hệ thống sẵn sàng. Nếu điều này xảy ra nhiều hơn một lần, bạn có thể sẽ không mua sắm tại trang web đó nữa.

Vấn đề với việc chỉ định độ tin cậy bằng các số liệu như POFOD, ROCOF và AVAIL có thể xác định quá mức độ tin cậy và do đó phải chịu chi phí phát triển và xác nhận cao. Lý do cho điều này là các bên liên quan hệ thống cảm thấy khó khăn để chuyển kinh nghiệm thực tế của họ thành các thông số kỹ thuật định lượng. Họ có thể nghĩ rằng POFOD = 0,001 (1failure trong 1.000 nhu cầu) đại diện cho một hệ thống tương đối không đáng tin cậy. Tuy nhiên, nếu nhu cầu về một dịch vụ là không phổ biến, thì nó lại thực sự thể hiện mức độ tin cậy rất cao.

Nếu bạn chỉ định độ tin cậy như một số liệu, rõ ràng điều quan trọng là phải đánh giá rằng mức độ tin cậy cần thiết đã đạt được. Bạn thực hiện đánh giá này như là một phần của thử nghiệm hệ thống. Để đánh giá độ tin cậy của một hệ thống theo thống kê, bạn phải quan sát một số lỗi. Ví dụ, nếu bạn có POFOD 0,0001 (1 lỗi trong 10.000 yêu cầu), thì bạn có thể phải thiết kế các thử nghiệm thực hiện 50 hoặc 60 nghìn yêu cầu trên một hệ thống và quan sát thấy một số lỗi. Thực tế có thể không thể thiết kế và thực hiện số lượng thử nghiệm này. Do đó, việc xác định quá mức độ tin cậy dẫn đến chi phí thử nghiệm rất cao.

Khi bạn chỉ định tính khả dụng của một hệ thống, bạn có thể gặp vấn đề tương tự. Mặc dù mức độ sẵn sàng rất cao dường như là mong muốn, nhưng hầu hết các hệ thống đều có các nhu cầu rất không liên tục

* Ví dụ: một hệ thống kinh doanh sẽ chủ yếu được sử dụng trong giờ làm việc bình thường và một con số khả dụng đơn lẻ không thực sự phản ánh nhu cầu của người dùng. Bạn cần tính sẵn sàng cao khi hệ thống đang được sử dụng ở bất kỳ lúc nào. Tất nhiên, tùy thuộc vào loại hệ thống, có thể không có sự khác biệt thực tế thực tế giữa tính khả dụng là 0,999 và tính khả dụng là 0,9999.

Một vấn đề cơ bản với việc xác định quá mức là trên thực tế có thể không thể chứng minh rằng mức độ tin cậy hoặc tính sẵn sàng rất cao đã đạt được.

* Ví dụ, giả sử một hệ thống được thiết kế để sử dụng trong một ứng dụng quan trọng về an toàn và do đó được yêu cầu không bao giờ thất bại trong toàn bộ vòng đời của nó. Giả sử rằng 1.000 bản sao của hệ thống sẽ được cài đặt và hệ thống được thực thi 1.000 lần mỗi giây. Tuổi thọ dự kiến của hệ thống là 10 năm. Do đó, tổng số lần thực hiện hệ thống là khoảng 3 \* 1014. Không có lý do nào để xác định rằng tỷ lệ xảy ra lỗi phải là 1/1015 thực thi (điều này cho phép một số yếu tố an toàn) vì bạn không thể kiểm tra hệ thống đủ lâu để xác thực mức độ tin cậy này.

Do đó, các tổ chức phải thực tế về việc có đáng để đặc tả và xác định mức độ tin cậy rất cao hay không. Mức độ tin cậy cao được chứng minh rõ ràng trong các hệ thống mà hoạt động đáng tin cậy là rất quan trọng, chẳng hạn như hệ thống chuyển mạch điện thoại hoặc khi sự cố hệ thống có thể dẫn đến thiệt hại kinh tế lớn. Họ có thể không được áp dụng cho nhiều loại hình kinh doanh hoặc hệ thống khoa học. Các hệ thống như vậy có yêu cầu độ tin cậy khiêm tốn, vì chi phí thất bại chỉ đơn giản là xử lý sự chậm trễ và việc phục hồi từ những điều này là đơn giản và tương đối rẻ tiền.

Có một số bước bạn có thể thực hiện để tránh tình trạng quá tin cậy của hệ thống:

1. Chỉ định các yêu cầu về tính khả dụng và độ tin cậy cho các loại lỗi khác nhau. Cần có xác suất thất bại nghiêm trọng thấp hơn so với những thất bại nhỏ.
2. Chỉ định các yêu cầu về tính khả dụng và độ tin cậy cho các dịch vụ khác nhau một cách riêng biệt. Thất bại ảnh hưởng đến các dịch vụ quan trọng nhất phải được chỉ định là ít có khả năng xảy ra hơn so với những dịch vụ chỉ có tác động cục bộ. Bạn có thể quyết định giới hạn đặc tả độ tin cậy định lượng đối với các dịch vụ hệ thống quan trọng nhất.
3. Quyết định xem bạn có thực sự cần độ tin cậy cao trong hệ thống phần mềm hay không hoặc liệu các mục tiêu phụ thuộc hệ thống tổng thể có thể đạt được theo những cách khác hay không. Ví dụ: bạn có thể sử dụng các cơ chế phát hiện lỗi để kiểm tra các đầu ra của hệ thống và có các quy trình tại chỗ để sửa lỗi. Sau đó, có thể không cần mức độ tin cậy cao trong hệ thống tạo ra các đầu ra.

Để minh họa điểm sau này, hãy xem xét các yêu cầu về độ tin cậy đối với hệ thống ATM ngân hàng phân phối tiền mặt và cung cấp các dịch vụ khác cho khách hàng. Nếu có sự cố ATM phần cứng hoặc phần mềm, thì những điều này dẫn đến các mục không chính xác trong cơ sở dữ liệu tài khoản của khách hàng. Những điều này có thể tránh được bằng cách chỉ định mức độ tin cậy phần cứng và phần mềm rất cao trong ATM.

Tuy nhiên, các ngân hàng có nhiều năm kinh nghiệm về cách xác định và sửa các giao dịch tài khoản không chính xác. Họ sử dụng các phương pháp kế toán để phát hiện khi mọi thứ đã đi sai. Hầu hết các giao dịch thất bại đơn giản có thể bị hủy, dẫn đến không có tổn thất cho ngân hàng và sự bất tiện của khách hàng nhỏ. Do đó, các ngân hàng chạy mạng ATM chấp nhận rằng sự cố ATM có thể có nghĩa là một số lượng nhỏ giao dịch không chính xác nhưng họ nghĩ rằng sẽ chọn cách khắc phục những điều này khi nó xảy ra thay vì phải chịu chi phí rất cao trong việc tránh các giao dịch bị lỗi.

Đối với ngân hàng (và đối với khách hàng của ngân hàng), tính khả dụng của mạng ATM quan trọng hơn so với việc giao dịch ATM cá nhân có thất bại hay không. Thiếu tính khả dụng có nghĩa là nhu cầu nhiều hơn về dịch vụ truy cập, sự không hài lòng của khách hàng, chi phí kỹ thuật để sửa chữa mạng, v.v. Do đó, đối với các hệ thống dựa trên giao dịch, như hệ thống ngân hàng và thương mại điện tử, trọng tâm của đặc tả độ tin cậy thường là chỉ định tính khả dụng của hệ thống.

Để chỉ định tính khả dụng của mạng ATM, bạn nên xác định các dịch vụ hệ thống và chỉ định tính khả dụng cần thiết cho từng dịch vụ này. Đó là:

* dịch vụ cơ sở dữ liệu tài khoản khách hàng;
* các dịch vụ riêng lẻ được cung cấp bởi ATM như ‘tiền mặt’ ‘cung cấp thông tin tài khoản’,... v.v.

Ở đây, dịch vụ cơ sở dữ liệu là quan trọng nhất vì sự thất bại của dịch vụ này có nghĩa là tất cả các máy ATM trong mạng đều không hoạt động. Do đó, bạn nên xác định điều này để có mức độ sẵn sàng cao. Trong trường hợp này, một con số chấp nhận được về tính khả dụng của cơ sở dữ liệu (bỏ qua các vấn đề như bảo trì và nâng cấp theo lịch trình) có thể là khoảng 0,9999, trong khoảng thời gian từ 7 giờ sáng đến 11 giờ tối. Điều này có nghĩa là thời gian xuống dưới một phút mỗi tuần. Trong thực tế, điều này có nghĩa là rất ít khách hàng sẽ bị ảnh hưởng và chỉ dẫn đến sự bất tiện của khách hàng nhỏ.

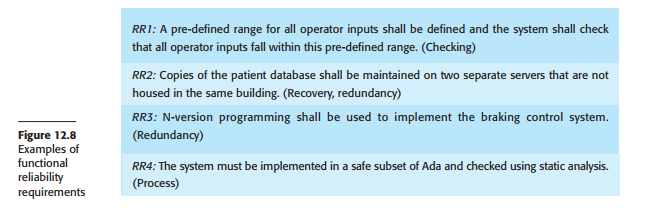
Đối với một ATM riêng lẻ, tính khả dụng tổng thể phụ thuộc vào độ tin cậy cơ học và thực tế là nó có thể hết tiền mặt. Các vấn đề phần mềm có thể có ít ảnh hưởng hơn các yếu tố như vậy. Do đó, mức độ sẵn sàng thấp hơn cho phần mềm ATM là chấp nhận được. Do đó, tính khả dụng chung của phần mềm ATM có thể được chỉ định là 0,999, điều đó có nghĩa là máy có thể không khả dụng trong khoảng từ một đến hai phút mỗi ngày.

Để minh họa đặc tả độ tin cậy dựa trên thất bại, hãy xem xét các yêu cầu về độ tin cậy đối với phần mềm điều khiển trong bơm insulin. Hệ thống này cung cấp insulin một số lần mỗi ngày và theo dõi đường huyết của người dùng nhiều lần mỗi giờ. Do việc sử dụng hệ thống không liên tục và hậu quả thất bại là nghiêm trọng, nên chỉ số độ tin cậy phù hợp nhất là POFOD (xác suất thất bại theo yêu cầu).

Ví dụ: Có hai loại lỗi có thể xảy ra trong bơm insulin:

1. Lỗi phần mềm thoáng qua có thể được sửa chữa bằng các hành động của người dùng như đặt lại hoặc hiệu chỉnh lại máy. Đối với các loại thất bại này, giá trị POFOD tương đối thấp (giả sử 0,002) có thể được chấp nhận. Điều này có nghĩa là một lỗi có thể xảy ra trong mỗi 500 yêu cầu được thực hiện trên máy. Điều này là khoảng 3,5 ngày một lần, bởi vì lượng đường trong máu được kiểm tra khoảng năm lần mỗi giờ.
2. Lỗi phần mềm vĩnh viễn yêu cầu nhà sản xuất cài đặt lại phần mềm. Xác suất của loại thất bại này sẽ thấp hơn nhiều. Khoảng tối thiểu một năm một lần, vì vậy POFOD không quá 0,00002.

Tuy nhiên, việc không cung cấp insulin không có ý nghĩa an toàn ngay lập tức, vì vậy các yếu tố thương mại hơn là các yếu tố an toàn chi phối mức độ tin cậy cần thiết. Chi phí dịch vụ cao vì người dùng cần sửa chữa và thay thế nhanh chóng. Đó là lợi ích của nhà sản xuất để hạn chế số lượng lỗi vĩnh viễn cần sửa chữa.



### Đặc tả độ tin cậy chức năng

Đặc tả độ tin cậy chức năng liên quan đến việc xác định các yêu cầu định nghĩa các ràng buộc và tính năng góp phần vào độ tin cậy của hệ thống. Đối với các hệ thống mà độ tin cậy đã được chỉ định một cách định lượng, các yêu cầu chức năng này có thể cần thiết để đảm bảo đạt được mức độ tin cậy cần thiết. Có ba loại yêu cầu về độ tin cậy chức năng cho một hệ thống:

1. ***Yêu cầu kiểm tra*** Các yêu cầu này xác định kiểm tra các đầu vào cho hệ thống để đảm bảo rằng các đầu vào ngoài phạm vi không chính xác hoặc nằm ngoài phạm vi được phát hiện trước khi chúng được hệ thống xử lý.
2. ***Yêu cầu khôi phục*** Những yêu cầu này nhằm giúp hệ thống phục hồi sau khi xảy ra lỗi. Thông thường, các yêu cầu này liên quan đến việc duy trì các bản sao của hệ thống và dữ liệu của nó và chỉ định cách khôi phục dịch vụ hệ thống sau khi gặp sự cố.
3. ***Yêu cầu dự phòng*** Những yêu cầu này chỉ định các tính năng dự phòng của hệ thống để đảm bảo rằng một lỗi thành phần duy nhất không dẫn đến mất hoàn toàn dịch vụ. Tôi sẽ thảo luận chi tiết hơn trong chương tiếp theo.

Ngoài ra, các yêu cầu về độ tin cậy có thể bao gồm các yêu cầu trong tiến trình xử lý về vấn đề tin cậy. Đây là những yêu cầu để đảm bảo rằng xử lý tốt, giảm số lượng lỗi trong hệ thống, được sử dụng trong quá trình phát triển. Một số ví dụ về độ tin cậy chức năng và các yêu cầu xử lý được thể hiện trong Hình 12.8.

Không có quy tắc đơn giản để có được các yêu cầu về độ tin cậy chức năng. Trong các tổ chức phát triển các hệ thống đặc biệt quan trọng, thường có kiến thức tổ chức về các yêu cầu độ tin cậy có thể có và cách thức các tác động này đến độ tin cậy thực tế của một hệ thống. Các tổ chức này có thể chuyên về các loại hệ thống cụ thể như hệ thống kiểm soát đường sắt, do đó, các yêu cầu về độ tin cậy có thể được sử dụng lại trên một loạt các hệ thống.

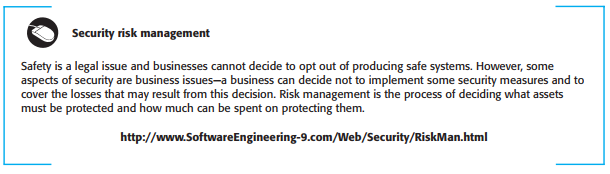
## Đặc tả tính an ninh

Các đặc điểm kỹ thuật của các yêu cầu bảo mật cho các hệ thống có điểm chung với các yêu cầu an toàn. Việc chỉ định chúng một cách định lượng là không thực tế và các yêu cầu bảo mật thường là ‘sẽ không phải là các yêu cầu xác định hành vi hệ thống không được chấp nhận thay vì chức năng hệ thống được yêu cầu. Tuy nhiên, bảo mật là một vấn đề thách thức hơn an toàn, vì một số lý do:

1. Khi xem xét sự an toàn, bạn có thể cho rằng môi trường mà hệ thống được cài đặt không phải là thù địch. Không ai đang cố gắng gây ra một sự cố liên quan đến an toàn. Khi xem xét bảo mật, bạn phải cho rằng các cuộc tấn công vào hệ thống là có chủ ý và kẻ tấn công có thể có kiến thức về các điểm yếu của hệ thống.
2. Khi xảy ra lỗi hệ thống gây rủi ro cho sự an toàn, bạn tìm kiếm các lỗi hoặc thiếu sót đã gây ra lỗi. Khi các cuộc tấn công có chủ ý gây ra lỗi hệ thống, việc tìm ra nguyên nhân gốc có thể khó khăn hơn vì kẻ tấn công có thể cố gắng che giấu nguyên nhân của sự cố.
3. Thường có thể chấp nhận tắt hệ thống hoặc làm giảm dịch vụ hệ thống để tránh sự cố liên quan đến an toàn. Tuy nhiên, các cuộc tấn công vào một hệ thống có thể được gọi là tấn công từ chối dịch vụ, nhằm mục đích tắt hệ thống. Tắt hệ thống có nghĩa là cuộc tấn công đã thành công.
4. Các sự kiện liên quan đến an toàn không được tạo ra bởi một kẻ thù thông minh. Kẻ tấn công có thể thăm dò hệ thống phòng thủ của hệ thống trong một loạt các cuộc tấn công, sửa đổi các cuộc tấn công khi chúng tìm hiểu thêm về hệ thống và các phản ứng của nó.

Những khác biệt này có nghĩa là các yêu cầu bảo mật thường phải rộng hơn các yêu cầu an toàn. Yêu cầu an toàn dẫn đến việc tạo ra các yêu cầu hệ thống chức năng cung cấp bảo vệ chống lại các sự kiện và lỗi có thể gây ra lỗi liên quan đến an toàn. Họ chủ yếu quan tâm đến việc kiểm tra các vấn đề và thực hiện các hành động nếu những vấn đề này xảy ra. Ngược lại, có nhiều loại yêu cầu bảo mật bao gồm các mối đe dọa khác nhau mà một hệ thống phải đối mặt. Firesmith (2003) đã xác định 10 loại yêu cầu bảo mật có thể được bao gồm trong một đặc tả hệ thống:

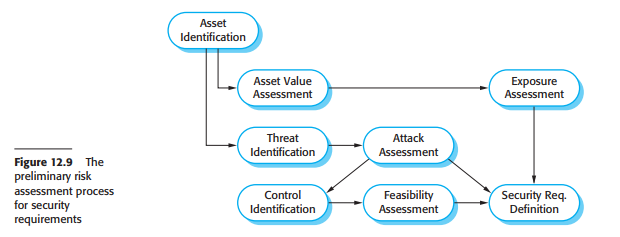
1. Yêu cầu nhận dạng xác định liệu một hệ thống có nên xác định thông tin người dùng của mình hay không trước khi tương tác với họ.
2. Yêu cầu xác thực chỉ định cách xác thực người dùng.
3. Yêu cầu ủy quyền xác định các đặc quyền và quyền truy cập của người dùng được xác định.
4. Yêu cầu về khả năng miễn dịch xác định cách hệ thống tự bảo vệ chống lại virus, worm và các mối đe dọa tương tự.
5. Yêu cầu toàn vẹn xác định cách tránh mất mát dữ liệu.
6. Yêu cầu phát hiện xâm nhập xác định cơ chế nào sẽ được sử dụng để phát hiện các cuộc tấn công trên hệ thống.
7. Yêu cầu không thoái thác xác định rằng một bên trong giao dịch không thể từ chối sự tham gia của mình vào giao dịch đó.
8. Yêu cầu về quyền riêng tư xác định cách bảo mật dữ liệu được duy trì.
9. Yêu cầu kiểm toán bảo mật xác định cách sử dụng hệ thống có thể được kiểm toán và kiểm tra.
10. Yêu cầu bảo mật bảo trì hệ thống xác định cách ứng dụng có thể ngăn các thay đổi được ủy quyền vô tình đánh bại các cơ chế bảo mật của nó.



Tất nhiên, bạn sẽ không thấy tất cả các loại yêu cầu bảo mật này trong mọi hệ thống. Các yêu cầu cụ thể phụ thuộc vào loại hệ thống, tình hình sử dụng và người dùng mong đợi.

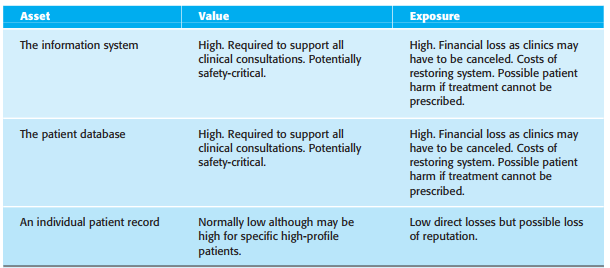
Quá trình phân tích và đánh giá rủi ro được thảo luận trong Phần 12.1 có thể được sử dụng để xác định các yêu cầu bảo mật hệ thống. Có 3 giai đoạn trong quá trình này:

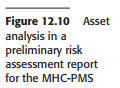
1. ***Phân tích rủi ro sơ bộ*** Ở giai đoạn này, các quyết định về yêu cầu hệ thống chi tiết, thiết kế hệ thống hoặc công nghệ thực hiện chưa được thực hiện. Mục đích của quá trình đánh giá này là để đạt được các yêu cầu bảo mật cho toàn bộ hệ thống.
2. ***Phân tích rủi ro vòng đời***Đánh giá rủi ro này diễn ra trong vòng đời phát triển hệ thống sau khi lựa chọn thiết kế được đưa ra. Các yêu cầu bảo mật bổ sung có tính đến các công nghệ được sử dụng trong việc xây dựng các quyết định thiết kế và triển khai hệ thống.
3. ***Phân tích rủi ro xử lý*** Quá trình này xem xét các rủi ro do các cuộc tấn công độc hại gây ra cho hệ thống vận hành của người dùng, có hoặc không có kiến thức nội bộ về hệ thống.



Các quy trình đánh giá và phân tích rủi ro được sử dụng trong đặc tả yêu cầu bảo mật là các biến thể của quy trình đặc tả kỹ thuật theo rủi ro chung được thảo luận trong Phần 12.1. Một quy trình yêu cầu bảo mật theo hướng rủi ro được thể hiện trong Hình 12.9. Điều này có vẻ khác với quy trình hướng đến rủi ro trong Hình 12.1, nhưng tôi chỉ ra cách mỗi giai đoạn tương ứng với các giai đoạn trong quy trình chung bằng cách đưa hoạt động quy trình chung vào ngoặc. Các giai đoạn của quá trình là:

1. Xác định tài sản, trong đó tài sản hệ thống có thể yêu cầu bảo vệ được xác định. Bản thân hệ thống hoặc các chức năng hệ thống cụ thể có thể được xác định là tài sản cũng như dữ liệu liên quan đến hệ thống (xác định rủi ro).
2. Đánh giá giá trị tài sản, nơi bạn ước tính giá trị của các tài sản được xác định (phân tích rủi ro).
3. Đánh giá tiếp xúc, nơi bạn đánh giá các tổn thất tiềm năng liên quan đến từng tài sản. Điều này cần tính đến những tổn thất trực tiếp như mất cắp thông tin, chi phí phục hồi và mất danh tiếng (phân tích rủi ro).
4. Xác định mối đe dọa, nơi bạn xác định các mối đe dọa đối với tài sản hệ thống (phân tích rủi ro).
5. Đánh giá cuộc tấn công, nơi bạn phân tách từng mối đe dọa thành các cuộc tấn công có thể được thực hiện trên hệ thống và các cách thức có thể xảy ra trong các cuộc tấn công này. Bạn có thể sử dụng attack tree (cây tấn công) (Schneier, 1999) để phân tích các cuộc tấn công có thể. Chúng tương tự như cây bị lỗi khi bạn bắt đầu với một mối đe dọa ở gốc cây và xác định các cuộc tấn công nguyên nhân có thể xảy ra và cách chúng có thể được thực hiện (phân hủy rủi ro).
6. Nhận dạng kiểm soát, nơi bạn đề xuất các kiểm soát có thể được đưa ra để bảo vệ tài sản. Các kiểm soát là các cơ chế kỹ thuật, chẳng hạn như mã hóa, mà bạn có thể sử dụng để bảo vệ tài sản (giảm rủi ro).
7. Đánh giá khả thi, nơi bạn đánh giá tính khả thi kỹ thuật và chi phí của các biện pháp kiểm soát được đề xuất. Không có giá trị kiểm soát đắt tiền để bảo vệ tài sản mà không có giá trị cao (giảm rủi ro).
8. Định nghĩa yêu cầu bảo mật, trong đó thông tin về phơi nhiễm, các mối đe dọa và đánh giá kiểm soát được sử dụng để rút ra các yêu cầu bảo mật hệ thống. Đây có thể là các yêu cầu đối với cơ sở hạ tầng hệ thống hoặc hệ thống ứng dụng (giảm rủi ro).





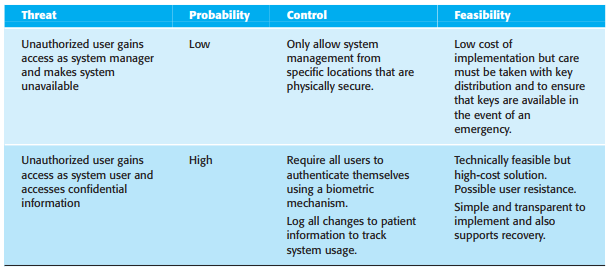
Một đầu vào quan trọng cho quá trình quản lý và đánh giá rủi ro là chính sách bảo mật của tổ chức. Chính sách bảo mật của tổ chức áp dụng cho tất cả các hệ thống và nên đặt ra những gì nên và không nên cho phép.

* Ví dụ: một khía cạnh của chính sách an ninh quân sự có thể nêu rõ 'Người đọc chỉ có thể kiểm tra các tài liệu có phân loại giống hoặc dưới mức độ kiểm tra của người đọc.' Điều này có nghĩa là nếu người đọc đã được xem xét đến mức 'bí mật', họ có thể truy cập các tài liệu được phân loại là 'bí mật', 'bí mật' hoặc 'mở' nhưng không phải là tài liệu được phân loại là 'bí mật hàng đầu'.

Chính sách bảo mật đặt ra các điều kiện phải luôn được duy trì bởi một hệ thống bảo mật và do đó giúp xác định các mối đe dọa có thể phát sinh. Các mối đe dọa là bất cứ điều gì có thể đe dọa an ninh kinh doanh. Trong thực tế, các chính sách bảo mật thường là các tài liệu không chính thức xác định cái gì được phép và cái gì không. Tuy nhiên, Bishop (2005) thảo luận về khả năng diễn đạt các chính sách bảo mật bằng ngôn ngữ chính thức và tạo kiểm tra tự động để đảm bảo rằng chính sách này được tuân thủ.

Từ phân tích rủi ro cho hệ thống thông tin bệnh viện, bạn có thể rút ra các yêu cầu bảo mật. Một số ví dụ về các yêu cầu này là:

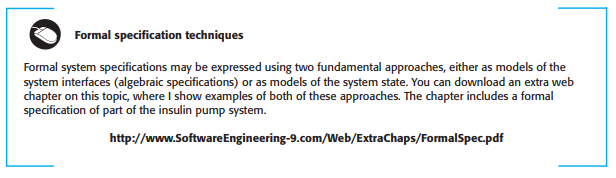
1. Thông tin bệnh nhân sẽ được tải xuống, khi bắt đầu phiên khám, từ cơ sở dữ liệu đến khu vực an toàn trên máy khách hệ thống.
2. Tất cả thông tin bệnh nhân trên máy khách hệ thống sẽ được mã hóa.
3. Thông tin bệnh nhân sẽ được tải lên cơ sở dữ liệu khi phiên khám kết thúc và bị xóa khỏi máy khách.
4. Nhật ký của tất cả các thay đổi được thực hiện đối với cơ sở dữ liệu hệ thống và người khởi tạo các thay đổi này sẽ được duy trì trên một máy tính riêng biệt từ máy chủ cơ sở dữ liệu.



Hai yêu cầu đầu tiên là thông tin bệnh nhân liên quan đến nhau được tải xuống máy cục bộ để các cuộc tham vấn có thể tiếp tục nếu máy chủ cơ sở dữ liệu bệnh nhân bị tấn công hoặc không khả dụng. Tuy nhiên, thông tin này phải được xóa để người dùng sau này của máy khách không thể truy cập thông tin. Yêu cầu thứ tư là một yêu cầu phục hồi và kiểm toán. Điều đó có nghĩa là các thay đổi có thể được phục hồi bằng cách phát lại nhật ký thay đổi và có thể phát hiện ra ai đã thực hiện các thay đổi. Trách nhiệm này không khuyến khích việc lạm dụng hệ thống của nhân viên được ủy quyền.

## Đặc tả hình thức

Các phương thức có tính hình thức là các cách tiếp cận dựa trên toán học để phát triển phần mềm, xác định một mô hình chính thức của phần mềm. Sau đó, bạn có thể hình thức phân tích mô hình này và sử dụng nó làm cơ sở cho một đặc tả hệ thống chính thức. Về nguyên tắc, có thể bắt đầu với một mô hình chính thức cho phần mềm và chứng minh rằng một chương trình được phát triển phù hợp với mô hình đó, do đó loại bỏ các lỗi phần mềm do lỗi lập trình.

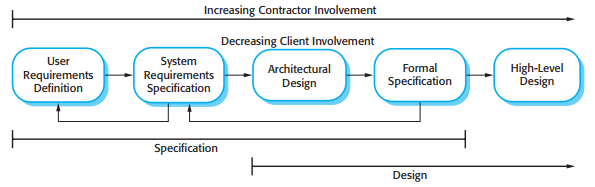


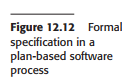
Điểm khởi đầu cho tất cả các quy trình phát triển chính thức là một mô hình hệ thống chính thức, phục vụ như một đặc tả hệ thống. Để tạo mô hình này, bạn dịch các yêu cầu người dùng của hệ thống, được thể hiện bằng ngôn ngữ tự nhiên, sơ đồ và bảng biểu sang ngôn ngữ toán học có định nghĩa chính thức về ngữ nghĩa. Đặc tả chính thức là một mô tả rõ ràng về những gì hệ thống nên làm. Sử dụng các phương thức được hỗ trợ bằng tay hoặc công cụ, bạn có thể kiểm tra xem hành vi của chương trình có phù hợp với đặc điểm kỹ thuật không.

Thông số kỹ thuật chính thức không chỉ cần thiết cho việc xác minh thiết kế và triển khai phần mềm. Chúng là cách chính xác nhất để đặc tả các hệ thống, và do đó giảm phạm vi hiểu lầm. Hơn nữa, việc xây dựng một đặc tả chính thức buộc phải phân tích chi tiết các yêu cầu và đây là một cách hiệu quả để khám phá các vấn đề về yêu cầu. Trong một đặc tả ngôn ngữ tự nhiên, các lỗi có thể được che giấu bằng sự thiếu chính xác của ngôn ngữ. Đây không phải là trường hợp nếu hệ thống được chỉ định chính thức.

Thông số kỹ thuật chính thức thường được phát triển như một phần của quy trình phần mềm dựa trên kế hoạch, trong đó hệ thống được chỉ định hoàn toàn trước khi phát triển. Các yêu cầu hệ thống và thiết kế được xác định chi tiết và được phân tích và kiểm tra cẩn thận trước khi bắt đầu thực hiện. Nếu một đặc tả chính thức của phần mềm được phát triển, điều này thường xuất hiện sau khi các yêu cầu hệ thống đã được chỉ định nhưng trước khi thiết kế hệ thống chi tiết. Có một vòng phản hồi chặt chẽ giữa đặc tả yêu cầu chi tiết và đặc điểm kỹ thuật chính thức.

Hình 12.12 cho thấy các giai đoạn của đặc tả phần mềm và giao diện của nó với thiết kế phần mềm trong quy trình phần mềm dựa trên kế hoạch. Vì tốn kém để phát triển các thông số kỹ thuật chính thức, bạn có thể quyết định giới hạn việc sử dụng phương pháp này đối với các thành phần quan trọng đối với hoạt động của hệ thống. Bạn xác định những điều này trong thiết kế kiến trúc của hệ thống.



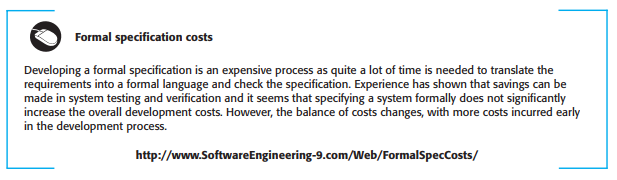


Những lợi thế của việc phát triển một đặc tả hình thức và sử dụng điều này trong một quy trình phát triển hình thức là:

1. Khi bạn phát triển một đặc tả hình thức một cách chi tiết, bạn sẽ phát triển sự hiểu biết sâu sắc và chi tiết về các yêu cầu hệ thống. Ngay cả khi bạn không sử dụng đặc tả trong quy trình phát triển chính thức, phát hiện lỗi yêu cầu là một đối số mạnh để phát triển một đặc tả hình thức (Hall, 1990). Các vấn đề yêu cầu được phát hiện sớm thường rẻ hơn nhiều để sửa chữa so với nếu chúng được tìm thấy ở các giai đoạn sau trong quá trình phát triển.
2. Vì đặc tả được thể hiện bằng ngôn ngữ với ngữ nghĩa được xác định có tính hình thức, bạn có thể tự động phân tích nó để khám phá sự không nhất quán và không đầy đủ.
3. Nếu bạn sử dụng một phương thức như phương thức B, bạn có thể chuyển đổi đặc tả chính thức thành một chương trình thông qua một chuỗi các phép biến đổi bảo toàn chính xác. Do đó, chương trình kết quả được đảm bảo để đáp ứng đặc điểm kỹ thuật của nó.
4. Chi phí kiểm tra chương trình có thể được giảm vì bạn đã xác minh chương trình theo thông số kỹ thuật của chương trình.

Mặc dù có những ưu điểm nêu trên, các phương pháp hình thức đã có tác động hạn chế trong việc phát triển phần mềm thực tế, ngay cả đối với các hệ thống quan trọng. Do đó, có rất ít kinh nghiệm trong cộng đồng phát triển và sử dụng các thông số kỹ thuật hệ thống chính thức. Các đối số được đưa ra để chống lại việc phát triển một đặc tả hệ thống chính thức là:

1. Chủ sở hữu vấn đề và chuyên gia tên miền không thể hiểu một đặc tả chính thức để họ không thể kiểm tra xem nó có đại diện chính xác cho yêu cầu của họ không. Các kỹ sư phần mềm, người hiểu đặc tả chính thức, có thể không hiểu miền ứng dụng nên họ cũng không thể chắc chắn rằng đặc tả chính thức là sự phản ánh chính xác các yêu cầu hệ thống.
2. Khá dễ dàng để định lượng chi phí tạo ra một đặc điểm kỹ thuật đặc tả hình thức, nhưng khó hơn để ước tính mức tiết kiệm chi phí có thể có do việc sử dụng nó. Kết quả là, các nhà quản lý không sẵn sàng chấp nhận rủi ro khi áp dụng phương pháp này.
3. Hầu hết các kỹ sư phần mềm chưa được đào tạo để sử dụng các ngôn ngữ đặc tả kiểu hình thức. Do đó, họ không muốn đề xuất sử dụng chúng trong các quy trình phát triển.
4. Rất khó để mở rộng các cách tiếp cận hiện tại đối với đặc điểm kỹ thuật đặc tả hình thức lên đến các hệ thống rất lớn. Khi đặc tả hình thức được sử dụng, phần lớn là để chỉ định phần mềm lõi kernel quan trọng hơn là các hệ thống hoàn chỉnh.
5. Đặc tả hình thức không tương thích với các phương pháp phát triển nhanh.



# Những đặc điểm chính

* Phân tích rủi ro là một hoạt động quan trọng trong đặc điểm kỹ thuật của các yêu cầu về bảo mật và độ tin cậy. Nó liên quan đến việc xác định các rủi ro có thể dẫn đến tai nạn hoặc sự cố. Các yêu cầu hệ thống sau đó được tạo ra để đảm bảo rằng những rủi ro này không xảy ra và nếu có, chúng sẽ không dẫn đến sự cố hoặc tai nạn.
* Cách tiếp cận theo hướng hướng đến nguy hiểm có thể được sử dụng để hiểu các yêu cầu an toàn cho hệ thống. Bạn xác định các mối nguy tiềm ẩn và phân hủy chúng (sử dụng các phương pháp như phân tích cây lỗi) để khám phá nguyên nhân gốc rễ của chúng. Sau đó, bạn chỉ định các yêu cầu để tránh hoặc phục hồi từ những vấn đề này.
* Yêu cầu độ tin cậy có thể được xác định một cách định lượng trong đặc tả yêu cầu hệ thống. Các số liệu về độ tin cậy bao gồm xác suất thất bại theo yêu cầu (POFOD), tỷ lệ xảy ra lỗi (ROCOF) và tính khả dụng (AVAIL).
* Điều quan trọng là không xác định quá mức độ tin cậy của hệ thống vì điều này dẫn đến chi phí bổ sung không cần thiết trong quá trình phát triển và xác nhận.
* Yêu cầu bảo mật khó xác định hơn yêu cầu an toàn vì kẻ tấn công hệ thống có thể sử dụng kiến thức về các lỗ hổng hệ thống để lên kế hoạch tấn công hệ thống và có thể tìm hiểu về các lỗ hổng từ các cuộc tấn công không thành công.
* Để chỉ định các yêu cầu bảo mật, bạn nên xác định các tài sản sẽ được bảo vệ và xác định cách sử dụng các kỹ thuật và công nghệ bảo mật để bảo vệ các tài sản này.
* Các phương pháp hình thức phát triển phần mềm dựa trên một đặc tả hệ thống được thể hiện dưới dạng mô hình toán học. Phát triển một đặc tả hình thức có lợi ích chính là kích thích kiểm tra và phân tích chi tiết các yêu cầu hệ thống.