

KIÊN

acceptance testing 7 (Kiểm thử chấp nhận)

Acceptance Testing (Kiểm thử chấp nhận) là một kiểm thử nhằm xác định hệ thống phần mềm có đạt yêu cầu kỹ thuật hay không. Bằng việc kiểm tra các hành vi của hệ thống qua dữ liệu thực tế, kiểm thử chấp nhận sẽ xác định có hay không việc hệ thống đáp ứng được các tiêu chí lẫn yêu cầu của khách hàng

adaptive maintenance 8 (Bảo trì thích ứng)

Bảo trì thích ứng: là dạng bảo trì thích ứng, thay đổi phần mềm để đáp ứng thay đổi của môi trường như quy định của chính phủ, CPU, công nghệ mới, thu được tiền từ khách hàng do khách hàng phát sinh nên sẽ phải kí hợp đồng để tương thích với nhu cầu của khách hàng

analysis phase 7 (giai đoạn phân tích)

analysis phase(giai đoạn phân tích) hay specification phase(giai đoạn thông số kỹ thuật): Các yêu cầu của khách hàng được phân tích và trình bày dưới dạng tài liệu đặc tả

architectural design 7 (thiết kế kiến trúc)

(thiết kế kiến trúc) kaf trong đó toàn bộ sản phẩm được chia nhỏ thành các thành phần, được gọi là mô-đun

artifact 18 (Tạo tác)

là 1 trong những loại sản phẩm hữu hình được tạo ra trong quá trình phát triển phần mềm. VD use case, sơ đồ lớp... Mỗi artifact mô tả chức năng kiến trúc thiết kế của phần mềm. Mỗi artifact khác nhau so với quá trình phát triển riêng của nó như kế hoạch dự án, business case

attribute 25 (Thuộc tính)

thuộc tính là một đặc tả định nghĩa đặc tính của một đối tượng, phần tử, hay tập tin. Nó còn có thể chỉ đến giá trị cho một thực thể cụ thể.

bug 25 (Lỗi)

Bug là những lỗi phần mềm trong chương trình hoặc hệ thống máy tính làm cho kết quả không chính xác hoặc không hoạt động như mong muốn.

classical paradigm 18 hay structured

structured or classical paradigm (mô hình cấu trúc hay mô hình cổ điển)

structured or classical paradigm (mô hình cấu trúc hay mô hình cổ điển)

Mô hình cổ điển là một mô hình lập trình nhằm mục đích cải thiện độ rõ ràng, chất lượng và thời gian phát triển của một chương trình máy tính bằng cách sử dụng rộng rãi các cấu trúc của luồng điều khiển có cấu trúc như lựa chọn (if / then / else) và lặp lại (while và for), cấu trúc khối, và các chương trình con.

clickware 23

phần mềm nhấp chuột: phần mềm không được nhắm mục tiêu đến một khách hàng hoặc người dùng cụ thể cho đến khi nó được phát triển và có sẵn để mua. Thường được tải xuống trên World Wide Web,

client 23

Khách hàng là cá nhân muốn sản phẩm được xây dựng (phát triển)

HOÀNG DƯƠNG

coding 7

(Mã hóa) Là quá trình hoặc hoạt động viết chương trình máy tính; hoặc quá trình gán mã để phân loại, nhận dạng.

commercial-off-the-shelf (COTS) software 23

(Phần mềm thương mại ra khỏi kệ) Là phần mềm được nhà phát triển bán với giá thành thấp, đáp ứng nhu cầu của rất nhiều người, thu lại được lợi nhuận nhờ việc bán với số lượng lớn. Phần mềm chuyên dụng thương mại có sẵn được thiết kế cho các ứng dụng cụ thể (chẳng hạn như thanh toán hợp pháp hoặc y tế, phân tích hóa học, phân tích thống kê) có thể được sử dụng với ít hoặc không sửa đổi.

contract software 23

(Phần mềm hợp đồng) khách hàng và nhà phát triển là thành viên của các tổ chức hoàn toàn độc lập. Trái với internal software (phần mềm nội bộ) - cả khách hàng lẫn những nhà phát triển đều có thể là một phần của cùng 1 tổ chức.

- Ví dụ, khách hàng có thể là một quan chức cấp cao trong Bộ Quốc phòng và các nhân viên phát triển của một nhà thầu quốc phòng lớn chuyên về phần mềm cho các hệ thống vũ khí.

- Ở một nhiều quy mô nhỏ hơn, khách hàng có thể là kế toán trong 1 đơn vị và nhà phát triển sinh viên kiếm được thu nhập bằng cách phát triển phần mềm bán thời gian.

corrective maintenance 8

(Bảo trì sửa chữa) Các hoạt động được thực hiện để phát hiện, cô lập, và khắc phục một lỗi để các thất bại trong thiết bị, máy móc, hoặc hệ thống có thể được phục hồi lại trạng thái thể hoạt động bình thường của nó. Tương phản với Preventive maintenance (bảo dưỡng phòng ngừa).

defect 25

(Khiếm khuyết) Là một điều kiện trong phần mềm mà không đáp ứng được yêu cầu hoặc sự kì vọng của người dùng cuối. Nói cách khác lỗi là những sai sót trong coding hoặc trong tư duy của người lập trình dẫn đến trục trặc hoặc kết quả chương trình không chính xác.

design by contract 20

(*Thiết kế theo hợp đồng/đúng luật*) Là một cách tiếp cận để thiết kế phần mềm. Nó quy định rằng các nhà thiết kế phần mềm nên định nghĩa theo đúng luật, một tả chính xác về interface và dễ kiểm tra đối với các thành phần phần mềm, mở rộng định nghĩa thông thường của loại dữ liệu trừu tượng với điều kiện tiên quyết, điều kiện sau và bất biến.

- Các thông số kỹ thuật này được gọi là "contracts", theo một ẩn dụ khái niệm với các điều kiện và nghĩa vụ của hợp đồng kinh doanh.

design document 7

(*Tài liệu thiết kế*) Là loại tài liệu được viết ra để giải thích các điểm quan trọng của thiết kế phần mềm khi kĩ sư chuẩn bị cho việc phát triển, bao gồm: làm gì (What), làm để làm gì (Why), làm như thế nào (How).

Chỉ viết ra những điểm quan trọng và chính xác, dễ hiểu, dễ truyền đạt cho người khác.

design phase 7

(*Giai đoạn thiết kế*) Là giai đoạn thứ 3 trong 6 giai đoạn của mô hình vòng đời (life-cycle) cổ điển. Ở giai đoạn này:

- Các thông số kỹ thuật trải qua 2 quy trình thiết kế liên tiếp trong suốt giai đoạn thiết kế.
- Đầu tiên là thiết kế kiến trúc, trong đó sản phẩm được chia thành các thành phần, được gọi là các module.
- Sau đó, mỗi module được thiết kế, thủ tục này được gọi là detailed design (thiết kế chi tiết).
- Kết quả ghi trong 2 bản design documents (tài liệu thiết kế) mô tả "sản phẩm làm như thế nào"

TUẤN ANH

19. Detailed-design 7 (thiết kế chi tiết):

Detailed design là một trong những level của design phần mềm. Trong quá trình này, người ta sẽ đi thiết kế chi tiết các thành phần và cho output là một bản thiết kế có thể dễ dàng coding.

20. Developer 23 (nhà phát triển):

là những người dùng ngôn ngữ lập trình để sáng tạo, xây dựng và bảo trì các chương trình, phần mềm, ứng dụng. ... Developer thường đóng vai trò quan trọng trong quá trình tạo ra phần mềm

21. development-thenmaintenance model 9(Phát triển và bảo trì mô hình phần mềm):

Một phần mềm sau khi được xây dựng và phát triển thì luôn luôn phải được bảo trì ,sửa lỗi trong khi phát triển và nâng cấp các tính năng

22. Documentation phase 17(giai đoạn tài liệu):

Để giải thích các điểm quan trọng của thiết kế software khi kĩ sư chuẩn bị cho việc phát triển, bao gồm : làm gì (What), làm để làm gì (Why), làm như thế nào (How).

23.Encapsulation 20(Tính đóng gói)

Là 1 trong 4 tính chất cơ bản của OOP,Hiểu một cách đơn giản, "đóng gói" là việc đưa tất cả thông tin, dữ liệu quan trọng vào bên trong một đối tượng (object).

24.Enhancement 8(nâng cao):

Một nâng cao sự thay đổi sản phẩm hoặc nâng cấp mà tăng phần mềm hoặc khả năng phần cứng vượt quá thông số kỹ thuật của khách hàng ban đầu. Cải tiến cho phép phần mềm và phần cứng hiệu suất sản phẩm khả năng mở rộng.

25. Error 25(lỗi):

Là hành động của con người dẫn đến kết quả sai.

26. Ethics (đạo đức) 26:

Đạo đức là một tập hợp các nguyên tắc đạo đức chi phối hành vi của một cá nhân hay nhóm. Do đó, đạo đức máy tính được thiết lập các nguyên tắc đạo đức mà điều chỉnh việc sử dụng máy tính. Một số vấn đề chung về đạo đức máy tính bao gồm quyền sở hữu trí tuệ (như nội dung điện tử có bản quyền), mối quan tâm riêng tư, và làm thế nào các máy tính ảnh hưởng đến xã hội.

27. Failure (thất bại) 25 :

chính là sự khác biệt giữa kết quả thực tế trên màn hình và kết quả mong đợi của một function, hệ thống hoặc service nào đó.

TÂN

fault (Lỗi trong đoạn mã) 25 : Khi lập trình viên mắc lỗi thì sẽ dẫn đến một lỗi trong đoạn mã đó và được gọi là fault.

field (trường) 25 : VD: Trong C ++, một thành viên của một đối tượng tham chiếu đến một thuộc tính ("feild") hoặc một phương thức.

implementation phase (Giai đoạn thực hiện) 7 : là các thành phần khác nhau trải qua quá trình mã hóa và thử nghiệm (thử nghiệm đơn vị) riêng biệt. Sau đó, các thành phần của sản phẩm được kết hợp và kiểm tra tổng thể; đây được gọi là tích hợp. Khi các nhà phát triển hài lòng rằng sản phẩm hoạt động chính xác, nó sẽ được khách hàng kiểm tra (kiểm tra chấp nhận). Giai đoạn triển khai kết thúc khi sản phẩm được khách hàng chấp nhận và được cài đặt trên máy tính của khách hàng.

instance variable (Biến thể) 25: một **biến thể** là một biến được định nghĩa trong một lớp (tức là một biến thành viên), mà mỗi đối tượng khởi tạo của lớp có một bản sao hoặc thể hiện riêng biệt. Một biến thể hiện có những điểm tương đồng với một biến lớp, nhưng không phải là biến tĩnh. Biến thể là một biến được khai báo trong một lớp nhưng bên ngoài các hàm tạo, phương thức hoặc khối.

integration (hội nhập) 7

là 1 khía cạnh bên trong của kỹ thuật phần mềm và các mô hình hình vòng đời lặp đi lặp lại đã được sử dụng để đảm bảo phần mềm không có lỗi

internal software development 23 (Phát triển phần mềm nội bộ): Cả khách hàng và nhà phát triển có thể là một phần của cùng một tổ chức. Đó được gọi là phát triển phần mềm nội bộ. (VD: Ví dụ: khách hàng có thể là chuyên gia tính toán chính của một công ty bảo hiểm và các nhà phát triển một nhóm do phó chủ tịch phụ trách phát triển phần mềm của công ty bảo hiểm đó đứng đầu).

life cycle 6 (Vòng đời): là một mô tả lý thuyết về những việc phải làm, chuỗi các bước thực tế được thực hiện trên một sản phẩm phần mềm cụ thể, từ khi bắt đầu cho đến cuối cùng, được gọi là vòng đời của sản phẩm đó.

(Bonus: Trên thực tế, các giai đoạn của vòng đời của một sản phẩm phần mềm có thể không được thực hiện chính xác như được chỉ định trong mô hình vòng đời, đặc biệt là khi gặp phải tình trạng vượt quá thời gian và chi phí.)

life-cycle model (Mô hình vòng đời): Là mô tả về các bước cần thực hiện khi xây dựng một sản phẩm phần mềm.

Linus's Law 23 (Luật Linus's): là Hệ quả mong đợi của bản chất công khai của các lỗi trong phần mềm nguồn mở.

(Bonus: Luật Linus cho rằng: tất cả các lỗi nếu có đủ cá nhân xem xét kỹ lưỡng mã nguồn của một sản phẩm phần mềm nguồn mở, thì ai đó sẽ có thể xác định được lỗi đó và đề xuất cách khắc phục lỗi đó)

THĂNG

maintenance 10

(*Sự duy trì, bảo quản thiết bị*) Bằng các hoạt động kỹ thuật, bảo trì viên có nhiệm vụ quản lý, tu sửa máy móc nhằm đảm bảo thiết bị luôn ở trong trạng thái làm việc hiệu quả

message 19

(*Thông điệp*) là một phần của công nghệ phần mềm trung gian. Thông điệp là một công nghệ quan trọng. Thông điệp là một cách để truyền dữ liệu qua hoặc một yêu cầu chức năng từ xa từ máy trạm đến máy chủ.

member function 26

(các hàm thành viên)Trong C ++, một thành viên của một đối tượng tham chiếu đến một thuộc tính ("trường fi") hoặc một phương thức. Trong Java, thuật ngữ trường fi được sử dụng để biểu thị một thuộc tính ("biến phiên bản") hoặc một phương thức.

methodology 24

(*cơ sở phương pháp luận*) là một phép biện chứng duy vật đem lại sự nhìn nhận một cách khách quan về các hoạt động phức tạp theo quy luật của sự biện chứng để có thể tiến tới việc xem xét các sự việc

trmaintenance 10

(*Sự duy trì, bảo quản thiết bị*) Bằng các hoạt động kỹ thuật, bảo trì viên có nhiệm vụ quản lý, tu sửa máy móc nhằm đảm bảo thiết bị luôn ở trong trạng thái làm việc hiệu quả

method 19

(*Phương thức*) Là một phương pháp là một chương trình con gắn liền với một lớp học cụ thể được xác định trong mã nguồn của một chương trình.

methodology 24

(*cơ sở phương pháp luận*). là một phép biện chứng duy vật đem lại sự nhìn nhận một cách khách quan về các hoạt động phức tạp theo quy luật của sự biện chứng để có thể tiến tới việc xem xét các sự việc trong chính một mối quan hệ biện chứng.

mistake 25

(*lỗi nhầm lẫn, sai lầm*) là lỗi do lập trình viên gây ra làm ảnh hưởng đến phần mềm

module 7

Module là một thành phần giữ một nhiệm vụ cụ thể nào đó trong 1 phần mềm, một máy tính hay 1 hệ thống cụ thể. Trong lập trình thì module sẽ ở dạng codes, còn đối với một phần mềm như Software thì module được coi như một compiled.

object-oriented paradigm 25

(*mô hình hướng đối tượng*) bao gồm 3 phần :Phân tích hướng đối tượng , Thiết kế hướng đối tượng, Lập trình hướng đối tượng

open-source software 23

(*Phần mềm nguồn mở*) là phần mềm với mã nguồn được công bố và sử dụng một giấy phép nguồn mở. Giấy phép này cho phép bất cứ ai cũng có thể nghiên cứu, thay đổi và cải tiến phần mềm, và phân phối phần mềm ở dạng chưa thay đổi hoặc đã thay đổi

QUÂN

operational definition (of maintenance) 10

Operational definition (of maintenance) (Định nghĩa vận hành của bảo trì): Bảo trì là quá trình xảy ra khi “phần mềm trải qua các sửa đổi đối với code và các tài liệu liên quan do sự cố hoặc nhu cầu cải tiến hoặc thích ứng” [theo ISO / IEC 12207, 1995]. Theo định nghĩa vận hành này, bảo trì xảy ra bất cứ khi nào có lỗi hoặc các yêu cầu thay đổi, bất kể việc này diễn ra trước hay sau khi lắp đặt sản phẩm.

Trong cuốn sách này, thuật ngữ bảo trì sau phân phối đề cập đến định nghĩa của IEEE năm 1990 về bảo trì là bất kỳ thay đổi nào đối với phần mềm sau khi nó đã được phân phối và cài đặt trên máy tính của khách hàng và bảo trì hiện đại hoặc chỉ bảo trì đề cập đến định nghĩa sửa chữa của ISO / IEC 1995 , các hoạt động hoàn thiện hoặc thích ứng được thực hiện bất kỳ lúc nào. Do đó, bảo trì sau phân phối là một tập hợp con của bảo trì (hiện đại).

paradigm 24

Paradigm (mô hình):

Sau đó, vào những năm 1980, từ mô hình đã trở thành một từ thông dụng chính của thế giới kinh doanh, như trong cụm từ, “Đó là một mô hình hoàn toàn mới”. Vì vậy, ngành công nghiệp phần mềm bắt đầu sử dụng từ mô hình trong cụm từ mô hình hướng đối tượng và mô hình cổ điển (hoặc truyền thống) để có nghĩa là “một phong cách phát triển phần mềm”. Đây là một sự lựa chọn thuật ngữ không may mắn khác, bởi vì một mô thức là một mô hình hoặc một khuôn mẫu.

perfective maintenance 8

Perfective maintenance (bảo trì hoàn hảo) là một trong 2 loại cải tiến (cập nhật phần mềm) trong bảo trì sau phân phối (postdelivery maintenance), là những thay đổi mà khách hàng cho rằng sẽ cải thiện hiệu quả của sản phẩm, chẳng hạn như chức năng bổ sung hoặc giảm thời gian phản hồi.

phase 6

Phase (giai đoạn) là các bước nhỏ được chia ra từ mô hình vòng đời phần mềm (Mô hình vòng đời là mô tả các bước cần được thực hiện khi xây dựng một sản phẩm phần mềm). Hay nói cách khác Phase là các bước cần được thực hiện khi xây dựng một sản phẩm phần mềm.

planning phase 16

Planning phase (Giai đoạn lập kế hoạch) là giai đoạn quan trọng không thể thiếu ngay từ đầu dự án với yêu cầu tìm ra các nhiệm vụ chính phụ cần phải hoàn thành và sắp xếp thời gian hợp lý cho chúng. Không có giai đoạn lập kế hoạch riêng biệt. Thay vào đó, các hoạt động lập kế hoạch được thực hiện trong suốt vòng đời phần mềm. Tuy nhiên, có những lúc các hoạt động lập kế hoạch chiếm ưu thế. Chúng bao gồm việc bắt đầu dự án (lập kế hoạch sơ bộ) và trực tiếp sau khi tài liệu đặc tả đã được khách hàng ký tắt (kế hoạch quản lý dự án phần mềm).

postdelivery maintenance 7

Postdelivery maintenance (Bảo trì sau phân phối) bao gồm tất cả các thay đổi đối với sản phẩm sau khi sản phẩm đã được phân phối và cài đặt trên máy tính của khách hàng và vượt qua kiểm tra chấp nhận. Bảo trì sau phân phối bao gồm bảo trì khắc phục (hoặc sửa chữa phần mềm), bao gồm loại bỏ các lỗi còn sót lại trong khi vẫn giữ nguyên các thông số kỹ thuật, cũng như cải tiến (hoặc cập nhật phần mềm), bao gồm các thay đổi đối với thông số kỹ thuật và việc thực hiện các thay đổi đó.

(Sau phân phối ở đây được hiểu là khi sản phẩm được sử dụng để thực hiện các nhiệm vụ mà nó đã được phát triển.)

process 5

Theo định nghĩa trong ISO 9000 thì Quá trình (*Process*) được định nghĩa là "*tập hợp các hoạt động có quan hệ lẫn nhau hoặc tương tác để biến đổi đầu vào thành đầu ra*". Quá trình phần mềm, tức là phương cách chúng ta sản xuất phần mềm, có các đặc tính và vấn đề riêng biệt của nó, mặc dù nó giống với kỹ thuật truyền thống ở nhiều khía cạnh.

process 5 (theo ngân hàng)

(là phương cách sản xuất ra phần mềm. Gồm những quy trình trong việc tạo ra 1 phần mềm, có thể có nhiều mô hình khác nhau trong 1 process, có thể sử dụng mô hình workflow để làm ra phần mềm)

product 24

Product (sản phẩm) là kết quả cuối cùng của một quá trình (process) sản xuất phần mềm.

program 24

Một program (chương trình) bao gồm mã biên dịch có thể chạy trực tiếp từ hệ điều hành của máy tính.

QUÝ

property 25

- **Property** là một thành viên (member) của một class, interface. Nó là mở rộng của một trường (field). **Property** cho phép bạn truy cập vào một trường hoặc thay đổi giá trị của trường đó, mà không cần thiết phải truy cập trực tiếp vào trường.

quality 17

- Chất lượng sản phẩm phần mềm (tiếng Anh: Software **quality**) là khả năng đáp ứng toàn diện nhu cầu của người dùng về tính năng cũng như công dụng được nêu ra một cách tường minh hoặc không tường minh **trong** những ngữ cảnh xác định.

regression fault 20

- Là một lỗi vô tình được đưa vào một bộ phận của sản phẩm do hậu quả của việc thực hiện một thay đổi đường như không liên quan đến một bộ phận khác của sản phẩm

requirements phase 7

- **requirements phase** hay còn gọi là Giai đoạn yêu cầu. Trong giai đoạn yêu cầu, khái niệm được khám phá và tìm hiểu, đồng thời các yêu cầu của khách hàng được đưa ra.

responsibility-driven design 20

- Là **Thiết kế theo hướng uy tín** là quá trình thiết kế các lớp bằng cách gán các phần tử chịu trách nhiệm được xác định rõ ràng cho mỗi lớp. Quá trình này có thể được sử dụng để xác định lớp nào nên triển khai phần nào của một chức năng ứng dụng.

retirement 8

- **retirement** - Sự nghỉ hưu . Nghỉ hưu xảy ra khi sản phẩm bị loại bỏ khỏi dịch vụ. Điều này xảy ra khi chức năng được cung cấp bởi sản phẩm không còn có ích cho tổ chức khách hàng.

send a message 26

- là quá trình đầu tiên tìm phương thức mà một đối tượng thực hiện một thông điệp và sau đó gọi phương thức đó.

shrink-wrapped software 23

- Phần mềm Shrinkwrap có nghĩa là các ứng dụng phần mềm máy tính 'có sẵn' thường có sẵn cho tất cả những người mua và người được cấp phép quan tâm theo các điều khoản và điều kiện tiêu chuẩn

software 24(Phần mềm)

- Phần mềm không chỉ bao gồm mã ở dạng máy có thể đọc được mà còn bao gồm tất cả các tài liệu là thành phần nội tại của mọi dự án. Phần mềm bao gồm tài liệu đặc tả, tài liệu thiết kế, các loại tài liệu pháp lý và kế toán, kế hoạch quản lý dự án phần mềm, và các tài liệu quản lý khác cũng như tất cả các loại sổ tay hướng dẫn.

ĐẠT

software crisis 4

- Khủng hoảng phần mềm là một thuật ngữ được sử dụng trong những ngày đầu của khoa học máy tính nói về khó khăn trong việc viết các chương trình máy tính hữu ích và hiệu quả trong thời gian yêu cầu

software depression 5

software engineering 2

- Kỹ nghệ phần mềm – Software Engineering (SE)
- SE là sự thiết lập và sử dụng các nguyên tắc công nghệ đúng đắn để được phần mềm một cách kinh tế, vừa tin cậy vừa làm việc hiệu quả trên các máy tính thực
- SE là bộ môn tích hợp cả quy trình, các phương pháp, các công cụ để phát triển phần mềm máy tính

software project management plan 7

- Mô tả những đề xuất phát triển phần mềm một cách chi tiết và đầy đủ

software repair 8

- Sửa chữa phần mềm là các hoạt động nhằm khắc phục các lỗi, hư hỏng liên quan tới phần mềm

specification document 7

- Tài liệu đặc tả là một hợp đồng giữa khách hàng và nhà phát triển. Nó xác định chính xác những gì sản phẩm phải làm và những ràng buộc đối với sản phẩm

specification phase 7

- Có nghĩa là giai đoạn đặc điểm kỹ thuật

- Các hoạt động trong giai đoạn bao gồm khám phá, nắm bắt, biểu hiện và xem xét các yêu cầu

state variable 25

- Biến trạng thái là một trong những tập hợp các biến được sử dụng để mô tả "trạng thái" toán học của một hệ thống động lực. Một cách trực quan, trạng thái của một hệ thống mô tả đủ về hệ thống để xác định hành vi trong tương lai của nó trong trường hợp không có bất kỳ lực bên ngoài nào ảnh hưởng đến hệ thống. Mô hình bao gồm các phương trình vi phân bậc nhất được ghép nối được cho là ở dạng biến trạng thái.
- Hoặc là: biến trạng thái của hệ thống là những tham số quan tâm có thể được sử dụng để tìm tất cả các tham số khác của hệ thống và kiến thức của chúng cho phép chúng ta biết về trạng thái hiện tại hoặc tương lai của hệ thống tổng thể

structured paradigm 18

- Lập trình có cấu trúc là một mô hình lập trình nhằm mục đích cải thiện độ rõ ràng, chất lượng và thời gian phát triển của một chương trình máy tính bằng cách sử dụng rộng rãi các cấu trúc của luồng điều khiển có cấu trúc như lựa chọn (if / then / else) và lặp lại (while và for), cấu trúc khối, và các chương trình con.

SƠ ANH

system 24

System (Software system) (Hệ thống phần mềm) là hệ thống các thành phần kết hợp với nhau dựa trên phần mềm, từ đó tạo thành một phần của hệ thống máy tính. Nó bao gồm các chương trình riêng biệt, các file cấu hình để cài đặt chương trình đó, tài liệu hệ thống để miêu tả cấu trúc, tài liệu hướng dẫn cho người sử dụng

systems analysis (Phân tích hệ thống) là quá trình nghiên cứu hệ thống hoặc các bộ phận của nó. Đây là 1 kĩ thuật giải quyết vấn đề giúp cải thiện hệ thống và đảm bảo rằng các thành phần của nó hoạt động hiệu quả

systems design (Thiết kế hệ thống) 24

là 1 quá trình thiết lập kế hoạch để phát triển hệ thống mới hoặc thay thế hệ thống hiện thời bằng cách xác định các thành phần của mô đun của để đáp ứng các yêu cầu cụ thể. Trước khi lập kế hoạch phải hiểu rõ hệ thống cũ cẩn thận và xác định những máy tính tốt nhất để vận hành hiệu quả

technique (kĩ thuật) 25

là những phương pháp hoặc thủ tục trong thiết kế, phát triển, ghi chép tài liệu và bảo trì phần mềm hoặc là để quản lí những hoạt động trên. Có 2 loại kĩ thuật phần mềm : 1 là cho những người làm chương trình, 2 là dành cho những người quản lí

temporal definition (of maintenance) 9

Đây là 1 định nghĩa được dùng trong phương pháp bảo trì cổ điển. Nó phụ thuộc vào thời điểm cài đặt phần mềm.

- Nếu lỗi được sửa trước khi được cài đặt thì được xác định là trong giai đoạn phát triển cổ điển.
- Nếu lỗi được sửa trước khi được cài đặt thì được xác định là trong giai đoạn phát bảo trì cổ điển

testing phase (giai đoạn kiểm tra) 17 là 1 giai đoạn trong chu trình phát triển phần mềm nơi sẽ tập trung vào việc điều tra và khám phá. Trong suốt giai đoạn này, nhà phát triển sẽ tìm hiểu xem phần mềm của họ có hoạt động đúng với yêu cầu của khách hàng hay không

traditional paradigm (Mô hình truyền thống) 25

Đây là 1 loại mô hình yêu cầu mỗi giai đoạn của chu trình phát triển phần mềm phải được hoàn thành toàn bộ trước khi giai đoạn tiếp theo bắt đầu. Nó giống như là 1 pipeline mà mỗi giai đoạn hoàn thành sẽ dẫn đến phần đầu của giai đoạn tiếp theo và sản phẩm ở phần cuối của pipeline

unit testing (Kiểm tra đơn vị)

một phương pháp kiểm tra phần mềm trong đó từng đơn vị mã nguồn riêng lẻ — tập hợp một hoặc nhiều mô-đun chương trình máy tính cùng với dữ liệu điều khiển liên quan, quy trình sử dụng và quy trình vận hành — được kiểm tra để xác định xem chúng có phù hợp để sử dụng hay không

user (computing)

Là người sử dụng các dịch vụ máy tính và dịch vụ mạng. 1 người dùng thường sẽ có 1 tài khoản và được định danh trong hệ thống bằng 1 cái tên do người dùng tự đặt ra.

KIÊN

validation 17(Thẩm định)

Validation là 1 quá trình đánh giá chuyên sâu được thực hiện thường khi đưa ra sản phẩm tới tay khách hàng

verification 17 (xác minh)

Verification là một quá trình đánh giá các sản phẩm làm việc trung gian của một vòng đời phát triển phần mềm để kiểm tra xem liệu rằng chúng ta có đi đúng hướng để tạo ra sản phẩm cuối cùng.

(là việc xác định xem workflow đã được thực hiện đúng hay chưa, việc này được thực hiện ở cuối mỗi workflow)

waterfall model 7 (Mô hình thác nước)

waterfall model: Mô hình thác nước là một phương pháp quản lý dự án dựa trên qui trình thiết kế tuần tự và liên tiếp.