###Chương 1

#### 1.1

Bạn chịu trách nhiệm tự động hóa một thực hành kiến ​​trúc nhiều địa điểm. Chi phí phát triển

phần mềm đã được ước tính là $ 530,000. Khoảng bao nhiêu tiền bổ sung sẽ cần thiết

để bảo trì phần mềm sau giao hàng?

Khoảng 75% tổng ngân sách là bảo trì.

Vì vậy, nếu 25% là phát triển, thì 75% là gấp 3 lần.

530000 \* 3 = 1590000

#### 1.2

Có cách nào để dung hòa định nghĩa thời gian cổ điển về bảo dưỡng không

với định nghĩa hoạt động mà chúng ta đang sử dụng?

Không. Vì "Bảo trì" cổ điển tương đương với bảo trì "gửi sau" hiện đại.

#### 1.3

Bạn là một nhà tư vấn kỹ thuật phần mềm. Trưởng phòng thông tin xăng dầu khu vực

công ty phân phối muốn bạn phát triển một sản phẩm phần mềm sẽ thực hiện tất cả các

chức năng kế toán của công ty và cung cấp thông tin trực tuyến cho nhân viên trụ sở chính

liên quan đến đơn đặt hàng và hàng tồn kho trong các bể chứa khác nhau của công ty. Máy tính là bắt buộc

cho 21 nhân viên kế toán, 15 nhân viên order, 37 lưu kho ... vv .. 4 tuần.

Bạn nói gì với anh ta?

Nếu chỉ có số tiền trên mà yêu cầu như vậy thì không nhận.

Giải thích sự cần thiết của một khung thời gian dài hơn:

- Chi phí sau bảo trì đặc biệt quan trọng

dành thời gian để phát triển một sản phẩm mạnh mẽ.

- Để đảm bảo sản phẩm chất lượng cao, đội ngũ không vội vàng thẳng tay.

vào nhưng, nhưng thực hiện một lượng phân tích đáng kể.

Điều này liên quan đến việc phát hiện sớm các lỗi, bởi vì

chi phí sửa một lỗi sau khi phát hành tăng lên rất nhiều.

Giải thích rằng bạn không thể chỉ cần thêm nhiều nhà phát triển hơn,

- Tháng nam thần thoại.

\*) tính ra có 73 máy tính mà như giá thị trường 1000$ 1 máy thì số tiền 30000$ là ko thể chấp nhận được. số tiền này chỉ đủ để mua bản mềm ban đầu.

#### 1.4

Bạn là một phó đô đốc trong Hải quân Velorian. Nó đã được quyết định để gọi trong một phần mềm

tổ chức phát triển để phát triển phần mềm điều khiển thế hệ mới

tên lửa đối hạm. Bạn phụ trách giám sát dự án. Để bảo vệ chính phủ

của Veloria, bạn bao gồm những điều khoản nào trong hợp đồng với nhà phát triển phần mềm?

-Kiểm tra nghiêm ngặt sẽ cần được thực hiện trước khi sử dụng tại hiện trường.

- Các yêu cầu cần được lập ra một cách chặt chẽ. Không thanh toán nếu không đáp ứng.

-Nhóm nên sử dụng các tiêu chuẩn mã hóa có thể được bên thứ ba xác minh về mã an toàn.

-Chính phủ Valerian sở hữu tất cả tài liệu và hiện vật, ngay cả khi việc phát triển bị hủy bỏ.

-Các tiêu chí chấp nhận phải được quy định: phần mềm phải đáp ứng các thông số kỹ thuật, được cung cấp đúng thời hạn,

trong phạm vi ngân sách, các lỗi được sửa miễn phí trong một năm, tài liệu đầy đủ và hoàn chỉnh, bao gồm mã nguồn.

#### 1.5

Bạn là một kỹ sư phần mềm có công việc giám sát sự phát triển của dự án 1.4.

Liệt kê những cách mà công ty của bạn có thể không đạt được hợp đồng với hải quân.

Phân tích không phản ánh yêu cầu -> Việc thực hiện cũng sẽ không.

Quá nhiều nhà phát triển có thể được chỉ định vào nhóm -> Gian hàng phát triển

Tài liệu chứng minh không đủ cho việc tuân thủ các tiêu chuẩn bảo trì.

#### 1.6

Chín tháng sau khi giao hàng, một sản phẩm được phát hiện có lỗi trong phần mềm. Chi phí sửa chữa 18,900.

Chi phí khoảng bao nhiêu trong giai đoạn phân tích?

154 đô la.

Từ hình 1.6, tỷ lệ chi phí sửa chữa chính sau sửa chữa phân tích là 368 đến 3

bằng 123 đến 1.

18900/123 = 153,6585

#### 1.7

Giả sử rằng nó đã được tìm thấy trong giai đoạn thực hiện .. làm thế nào để sửa chữa?

$ 2700

Tỷ lệ là khoảng 7 đến 1,

18900/7 = 2700

#### 1.8

Bạn là chủ tịch của một tổ chức xây dựng phần mềm quy mô lớn. Bạn hiển thị hình 1.6 cho

nhân viên của bạn, thúc giục họ sớm tìm ra lỗi trong vòng đời phần mềm. Ai đó trả lời rằng

Thật không đáng tin cậy khi mong đợi bất kỳ ai sớm loại bỏ lỗi ... nếu đó là lỗi mã hóa. Làm thế nào để bạn trả lời?

60 đến 70 phần trăm tất cả các lỗi được phát hiện trong các dự án lớn là lỗi yêu cầu, phân tích, thiết kế.

Nhiều trong số này chỉ đáng chú ý trong quá trình thực hiện. Mã hóa là biểu hiện của những giai đoạn trước đó.

Bằng cách tăng cường chất lượng ở các giai đoạn trước đó, các lỗi mã hóa sẽ dễ dàng tránh được hơn nhiều.

#### 1.9

Mô tả một tình huống trong đó khách hàng, nhà phát triển và người dùng là cùng một người.

1) Một nhà phát triển trò chơi tạo ra một công cụ cho các dự án trong tương lai. Anh ấy là khách hàng vì chương trình chỉ dành cho anh ấy

hoặc đội của anh ấy. Anh ấy là nhà phát triển khi anh ấy đang tạo ra nó. Anh ấy là người dùng vì anh ấy sẽ tạo ra các trò chơi với nó.

2) Một nhà vật lý tạo ra một bộ tương tự để kiểm tra một lý thuyết.

3) Một người nghiện Linux gửi một yêu cầu kéo.

#### 1.10

Những vấn đề nào có thể phát sinh nếu khách hàng, người dùng, nhà phát triển có cùng mục đích?

Làm thế nào những điều này có thể được giải quyết.

Có thể cảm thấy ít nghiêng về tài liệu hơn vì anh ấy biết mình sẽ sử dụng nó như thế nào.

Giải pháp: Vẫn thực thi các tiêu chuẩn.

Có thể thiếu tính di động vì nó được thiết kế cho một môi trường cụ thể.

Giải pháp: Phát triển như thể nó dành cho một lượng lớn khán giả hơn.

#### 1.11

Những lợi thế tiềm năng nào có được nếu khách hàng, nhà phát triển và khách hàng là người dùng?

Sự hài lòng của cá nhân khi tạo ra thứ gì đó có liên quan có thể làm tăng chất lượng.

Không có rắc rối khi hiểu sai yêu cầu giữa nhà phát triển và khách hàng, và nhu cầu

rõ ràng là hấp dẫn đối với nhà phát triển.

Ngân sách và kỳ vọng hợp lý hơn, vì khách hàng đã quen thuộc với nhu cầu phát triển.

#### 1.12

Tra cứu từ 'hệ thống' trong từ điển. Viết ra những thứ có liên quan đến kỹ thuật phần mềm.

1. một tập hợp hoặc sự kết hợp của các sự vật hoặc bộ phận tạo thành một tổng thể phức hợp hoặc đơn nhất:

2. bất kỳ tập hợp hoặc tập hợp các thành viên tương quan:

3. một tập hợp có trật tự và toàn diện các sự kiện, nguyên tắc, học thuyết, hoặc những thứ tương tự trong một lĩnh vực cụ thể

lĩnh vực kiến ​​thức hoặc tư tưởng:

4. một cơ quan phối hợp của các phương pháp hoặc một kế hoạch hoặc kế hoạch của thủ tục; sơ đồ tổ chức:

5. bất kỳ phương pháp hoặc kế hoạch thủ tục có công thức, thông thường hoặc đặc biệt nào:

6. phương pháp hoặc cách thức sắp xếp hoặc thủ tục có trật tự:

11. một phương pháp hoặc sơ đồ phân loại:

15. Máy vi tính. sự kết hợp hoạt động của phần cứng, phần mềm và các thiết bị truyền thông dữ liệu.

#### 1.13

Đây là ngày đầu tiên của bạn tại công việc đầu tiên của bạn. Người quản lý của bạn đưa cho bạn một danh sách chương trình và nói,

"xem nếu bạn có thể tìm thấy lỗi". Bạn trả lời gì?

"Chương trình định làm gì?". Yêu cầu tài liệu / thông số kỹ thuật.

#### 1.14

Bạn phụ trách phát triển sản phẩm trong vấn đề 1.1.

Bạn sẽ sử dụng mô hình oo hay cổ điển?

Cách tiếp cận oo sẽ phù hợp hơn để xử lý các yêu cầu có thể thay đổi.

Các đối tượng có thể được sử dụng để mô hình hóa môi trường và cung cấp một hệ thống mô-đun

sẽ cho phép hậu họa dễ dàng hơn.

Thay vì triển khai thành phần c9 nếu một sản phẩm phần mềm, các nhà phát triển quyết định sử dụng

so sánh COTS với các thông số kỹ thuật tương tự như c9. Ưu nhược điểm là gì?

Nhược điểm:

- Là một sản phẩm mã nguồn đóng, có thể không phù hợp với mô-đun để phát triển liên tục.

Khi các yêu cầu thay đổi, mô-đun COTS này sẽ không.

- Có thể không tích hợp trong sạch.

- Có thể tốn nhiều tiền hơn là chỉ viết mã

- Có thể phải trả thêm chi phí cho khách hàng nếu họ phải mua phần mềm này

- Tin tưởng công ty sẽ tiếp tục tồn tại và ủng hộ sản phẩm

#### 1.16

Thay vì triển khai thành phần c37 của một sản phẩm phần mềm, các nhà phát triển quyết định sử dụng

một thành phần mã nguồn mở có cùng thông số kỹ thuật với c37. Ưu nhược điểm là gì?

Ưu và nhược điểm phần lớn phụ thuộc vào chất lượng và sự hỗ trợ tích cực của mô-đun đó.

Thuận lợi:

- Nếu thành phần phổ biến, họ sẽ thực hiện một cách tiếp cận tiêu chuẩn hóa hơn.

- Có thể đã được xem xét kỹ lưỡng bởi một số lượng lớn người thử nghiệm.

- Có thể được duy trì tích cực theo cách tiếp tục mở rộng và hỗ trợ

bảo trì sau giao hàng.

- Có thể tiết kiệm tiền bạc, thời gian

- Nếu cần, nhóm có thể tiếp quản việc phát triển thành phần ngay cả khi bản gốc

người đóng góp chấm dứt.

Nhược điểm:

- Nếu chất lượng linh kiện kém, ưu điểm sẽ biến mất.

- Phong cách mã hóa có thể khác đáng kể so với phần còn lại của sản phẩm

- Có thể không tích hợp sạch sẽ với phần còn lại của sản phẩm.

#### 1.17

Đối tượng P gọi phương thức m1 của đối tượng Q ....

Dễ dàng tạo ra một hệ thống các đối tượng phụ thuộc nhiều vào nhau và

khả năng di động và khả năng tái sử dụng được cho là của OOP biến mất.

#### 1.18

Có chính xác không khi nói rằng, theo Định luật Linus, tất cả phần mềm nguồn mở

có chất lượng cao?

Không- chỉ là nó có tiềm năng. Chất lượng của mã nguồn mở hoàn toàn phụ thuộc

về mức độ quan tâm đến dự án đó và số tiền hỗ trợ.

###Chương 2

#### 2.1 Trình bày nghiên cứu trường hợp nhỏ của Winburg trong Phần 2.2 và 2.3 bằng cách sử dụng mô hình thác nước. Có phải đây là hiệu quả hơn hay kém hơn so với mô hình cây tiến hóa? Giải thich câu trả lơi của bạn. Trong mô hình thác nước: (null) -> Yêu cầu -> Phân tích -> Thiết kế -> Thực hiện ^ --------- | ^ ------- | ^ ------- | Mô hình này cho thấy rằng các bước đã được tuân theo và các vòng phản hồi được sử dụng, nhưng không nắm bắt thứ tự của các sự kiện như được thấy trong mô hình cây tiến hóa. Không đưa ra một hình ảnh rõ ràng về điều gì đã thực sự xảy ra trong quá trình phát triển.  Chọn mô hình cây tiến hóa. Nhưng tóm lại tùy vào bài toán thì mỗi sơ đồ lại có các điểm mạnh và yếu khác nhau. #### 2.2 Giả sử rằng lập trình viên trong nghiên cứu điển hình Winburg mini đã sử dụng các số chính xác đơn từ đầu. Vẽ cây tiến hóa kết quả. #### 2.3 Mối liên hệ giữa Luật Miller và sự sàng lọc từng bước là gì? Định luật Miller nói rằng con người có thể tập trung vào khoảng 7 sự việc. Dự án phần mềm lớn có nhiều hơn 7 thành phần khác nhau. Các nhà phát triển có thể tạo ra các thứ tự sự việc ưu tiên để dễ dàng sử dụng, quản lý. #### 2.4 Tinh chỉnh từng bước có tương ứng với lặp lại hoặc tăng dần không? Có, vì nó liên quan đến việc từng bước xem xét các yêu cầu khác, ác định rõ các yêu cầu của mỗi bước. #### 2,5 Quy trình làm việc, cấu tạo phần phần mềm và đường cơ sở có liên quan với nhau như thế nào? Quy trình làm việc thì mô tả một trong những hoạt động cốt lõi của sự phát triển. 5 quy trình làm việc cốt lõi là: yêu cầu, phân tích, thiết kế, triển khai và kiểm tra. #### 2.6 Mối liên hệ giữa mô hình thác nước và mô hình lặp lại và tăng dần là gì? Đối với mỗi gia số a ... n, các giai đoạn của mô hình thác nước tương ứng với quy trình công việc cốt lõi có thể được thực hiện trong công suất thay đổi. Ví dụ: Phần tăng thêm a có thể tập trung nhiều vào các yêu cầu và phân tích, trong khi Phần tăng dần d liên quan đến việc thực hiện và kiểm tra. Kiểm tra là một quy trình làm việc, nếu được thực hiện đúng cách, được đảm bảo sẽ được thực hiện trong mọi bước tăng. #### 2.7 Giả sử bạn phải xây dựng một tích để xác định căn bậc hai của 9384.2034 đến bốn thập phân. Một khi sản phẩm đã được thực hiện và thử nghiệm, nó sẽ bị vứt bỏ. Cái mà bạn sẽ sử dụng mô hình vòng đời? Đưa ra lý do cho câu trả lời của bạn. Mã và sửa. Các yêu cầu, phân tích và thiết kế là quá mức cần thiết và không cần quan tâm đến tài liệu sẽ hỗ trợ sau bảo trì sau này nữa. #### 2,8 Bạn là một nhà tư vấn kỹ thuật phần mềm và đã được gọi bởi phó chủ tịch cho tài chính của một tập đoàn sản xuất lốp xe và bán chúng thông qua chuỗi cửa hàng bán lẻ lớn của cửa hàng. Cô ấy muốn tổ chức của bạn tạo ra một sản phẩm sẽ giám sát hàng tồn kho của công ty, bắt đầu với việc mua nguyên liệu thô và theo dõi lốp xe khi chúng được sản xuất, được phân phối cho các cửa hàng riêng lẻ và bán cho khách hàng. Bạn sẽ sử dụng tiêu chí nào trong việc lựa chọn mô hình vòng đời cho dự án? Khách hàng có nhiều nhu cầu cụ thể. Ban đầu, tôi sẽ sử dụng, tạo mẫu nhanh để đảm bảo rằng sản phẩm đang phát triển thể hiện yêu cầu thực tế của nó. Sau đó, sẽ đưa ra phân tích hiệu quả để nhanh chóng hoàn thành sản phẩm demo #### 2.9 Liệt kê những rủi ro liên quan đến việc phát triển phần mềm của Bài toán 2.8. Bạn sẽ cố gắng như thế nào giảm thiểu từng rủi ro? Làm việc thông minh: Có thể phản ứng nhanh hơn với những thay đổi của sản phẩm hoặc thậm chí mang lại những yêu cầu thay đổi này sớm hơn thông qua tương tác với khách hàng thường xuyên. Sau bảo trì: Những thay đổi được thực hiện đối với những yêu cầu có thể là gánh nặng cho tương lai của sản phẩm. Cố gắng làm việc với khách hang để giảm thiểu số lượng các thành phần phải tahy đổi khi đã làm xong. Đảm bảo rằng tất cả yêu cầu sử dụng giao diện tiêu chuẩn hóa như ATAPI hoặc POSIX điều đó sẽ tạo điều kiện thuận lợi cho việc áp dụng các giải pháp thay thế hoặc ít nhất là mã nguồn mở để các nhà phát triển có thể tiếp quản mã nếu cần. #### 2.10 Việc bạn phát triển sản phẩm kiểm soát kho hàng cho công ty săm lốp đã thành công đến mức tổ chức quyết định rằng nó phải được thực hiện lại như một gói để bán cho nhiều loại các tổ chức khác nhau sản xuất và bán sản phẩm thông qua các nhà bán lẻ của riêng họ. Sản phẩm mới do đó phải có tính di động và dễ dàng thích ứng với phần cứng và / hoặc hệ điều hành mới. Các tiêu chí bạn sử dụng trong việc lựa chọn mô hình vòng đời cho dự án này sẽ khác với những tiêu chí đó như thế nào trong câu trả lời của bạn cho Vấn đề 2.8? Trong khi tôi chọn Agile cho Vấn đề 2.8, các yêu cầu khắt khe hơn của vấn đề này có thể tốt hơn được phục vụ với một mô hình chính thức chuẩn hơn. Nhắc lại và khuyến khích đưa ra các tiêu chí cho phép các nhà phát triển tập trung cho các các nền tảng quan trọng nhất. Ngoài ra, để phục vụ nhu cầu của một số lượng lớn khách hàng, mỗi bước phát triển có thể sử dụng phân tích yêu cầu sâu hơn với khách hàng mới (hoặc hiện tại) để tinh chỉnh sản phẩm và thu hút nhiều đối tượng hơn. #### 2.11 Mô tả loại sản phẩm sẽ là ứng dụng lý tưởng cho phần mềm nguồn mở sự phát triển. Để phần mềm nguồn mở thành công, nó thường phải thu hút những cá nhân đóng góp vào mã nguồn mở. Hệ điều hành, cơ sở dữ liệu, trình biên dịch, trình soạn thảo văn bản, "core utils", thư viện luồng,vv…. tất cả đều hấp dẫn các nhà phát triển. Các nhân viên hệ điều hành có thể dành nhiều ngày để gửi thư để tranh cãi chi tiết Các ứng dụng khoa học cũng được hưởng lợi rất nhiều. Nhiều nhà vật lý và toán học đang viết mã trong một bối cảnh học thuật, chứ không phải là một môi trường thương mại. Phần mềm là sản phẩm phụ của quá trình nghiên cứu và nó có ý nghĩa đối với chúng ta và sẽ được phát hành miễn phí. Bằng cách mở mã nguồn, tiến bộ khoa học được khuyến khích và tạo điều kiện cho khả năng phát triển. #### 2.12 Bây giờ hãy mô tả loại tình huống phát triển phần mềm nguồn mở là không phù hợp. Nếu bản thân việc bán phần mềm, thay vì các dịch vụ hỗ trợ, là sản phẩm, thì nó không thể được phát hành miễn phí. Về mặt kỹ thuật, phần mềm nguồn mở vẫn có thể được bán, nhưng có rất ít điểm nếu người mua và có quyền bán mã trên web. #### 2,13 Mô tả loại sản phẩm sẽ là một ứng dụng lý tưởng cho một quy trình nhanh. Từ cuốn sách, các dự án nhỏ hơn như ứng dụng web, ứng dụng di động và giao diện hoạt động tốt nhất. Agile có vẻ thích hợp cho các dự án sáng tạo, thoáng qua, nơi yêu cầu ít nghiêm ngặt hơn. Tuy nhiên, các công ty như USAA đang thách thức quan điểm đó với việc thúc đẩy việc áp dụng rộng rãi. #### 2,14 Bây giờ mô tả loại tình huống mà một quy trình nhanh là không phù hợp. Chúng ta không thể làm việc nhanh khi làm việc trong một sản phẩm quy mô lớn với lượng lớn tài liệu nghiêm ngặt và các mô-đun nhỏ đang được phát triển bởi một số lượng lớn các lập trình viên. Ví dụ, nhanh nhẹn có vẻ không phù hợp với VMware. #### 2,15 Mô tả loại sản phẩm sẽ là ứng dụng lý tưởng cho mô hình vòng đời xoắn ốc. Có thể sự phát triển của các dự án nghiên cứu nội bộ tiên tiến (do đó rủi ro), chẳng hạn như các kiến trúc HPC mới của IBM. Họ sẽ giám sát chặt chẽ tiến độ để đảm bảo rằng tiền không bị đổ vào một dự án vô ích. #### 2,16 Bây giờ mô tả loại tình huống mà mô hình vòng đời xoắn ốc là không phù hợp. Trong bất kỳ phần mềm hợp đồng nào, việc chấm dứt hợp đồng không được giải quyết dễ dàng. #### 2.17 Mô tả rủi ro vốn có khi sử dụng mô hình vòng đời thác nước. Sản phẩm cuối cùng có thể không phải là "những gì nhà phát triển muốn", mặc dù có vẻ như nhà phát triển đã đáp ứng những gì được yêu cầu. Mô hình không có khả năng thích ứng nhanh chóng để đáp ứng các yêu cầu thay đổi và chấp nhận các cách tiếp cận mới. #### 2,18 Mô tả rủi ro vốn có khi sử dụng mô hình vòng đời mã và sửa. Không có khả năng đáp ứng nhu cầu của bất kỳ ai và hoàn toàn không được mở rộng hoặc sửa đổi. #### 2,19 Mô tả rủi ro vốn có khi sử dụng mô hình vòng đời nguồn mở. Tình huống: Quá nhiều nhánh được tạo và người dùng cũng như cộng tác viên gặp khó khăn khi không biết nên sử dụng cái nào. Cộng đồng trở nên rời rạc và mất tầm nhìn rõ ràng về sản phẩm; một số phiên bản có các tính năng mà những người khác không có, v.v. Hãy xem: số lượng lớn các bản phân phối Linux được quan tâm đặc biệt. #### 2,20 Mô tả rủi ro vốn có trong việc sử dụng các quy trình nhanh. #### 2,21 Mô tả rủi ro vốn có khi sử dụng mô hình vòng đời xoắn ốc. Nó được xây dựng dựa trên phân tích rủi ro, nhưng điều này chỉ hữu ích nếu nhóm đủ nhạy bén để phát hiện những rủi ro này. Nó không tự động tiết lộ rủi ro.

###Chương 3

3.1

Xác định các thuật ngữ quy trình phần mềm và Quy trình thống nhất

"Quy trình phần mềm" mô tả toàn bộ các phương pháp, mô hình phong cách sống,

kỹ thuật, công cụ và cá nhân.

"Quy trình hợp nhất" thực sự đề cập đến một tập hợp con của quy trình phần mềm;

phương pháp hướng đối tượng chính hiện đang được sử dụng.

Đó là một hệ thống quy mô lớn, phần lớn chỉ áp dụng cho các sản phẩm phần mềm lớn.

Tuy nhiên, 5 quy trình làm việc cốt lõi có thể áp dụng cho hầu hết tất cả:

Yêu cầu, Phân tích, Thiết kế, Thực hiện và Kiểm tra.

3.2

Trong bối cảnh kỹ thuật phần mềm, thuật ngữ mô hình có nghĩa là gì?

Mô hình là một tập hợp các biểu đồ UML đại diện cho một hoặc nhiều khía cạnh của

phần mềm được phát triển. Thông qua bản chất đồ họa của nó, các liên kết nhanh chóng và

truyền đạt thông tin chính xác.

3,3

Một giai đoạn của Quy trình hợp nhất có nghĩa là gì?

Một giai đoạn là một thể hiện riêng biệt của sự gia tăng, nhưng với các tiêu đề

"Giai đoạn khởi động", "Giai đoạn xây dựng", "Giai đoạn xây dựng" và "Giai đoạn chuyển tiếp".

3,4

Phân biệt rõ ràng giữa mơ hồ, mâu thuẫn và không đầy đủ.

Tính mơ hồ: ngôn ngữ mà ý nghĩa chính xác có thể không rõ ràng.

Bao gồm ngôn ngữ chẳng hạn như "nó", trong đó ngôn ngữ mà nó mong đợi sẽ giải thích trực quan "nó" là gì.

Các tính từ như "to", "duyên dáng", và "phiền phức" cũng mang tính tham vọng.

Tính không đầy đủ: Thiếu các chi tiết có liên quan.

Cả hai điều trên đều là thuộc tính cá thể của ngôn ngữ tự nhiên.

Mâu thuẫn: Các tuyên bố thực tế khác nhau.

3.5

Xem xét quy trình làm việc yêu cầu và quy trình phân tích.

Sẽ có ý nghĩa hơn nếu kết hợp hai hoạt động này thành một quy trình làm việc hơn là xử lý chúng một cách riêng biệt?

Không, vì địa chỉ 2 nhu cầu khác nhau. Mục đích của Yêu cầu là tạo ra sự hiểu biết

giữa khách hàng và công ty phát triển. Vì mục đích này, việc sử dụng các ngôn ngữ tự nhiên là phù hợp

và quan tâm đến sản phẩm -is-, thay vì-như thế nào- nó là gì.

Quy trình phân tích chuyển đổi kết quả của giai đoạn yêu cầu thành sự hiểu biết chính xác.

Quy trình làm việc này đã mô tả chính xác các thông số kỹ thuật, chức năng của sản phẩm, cũng như những công cụ CASE nào,

kỹ thuật và phương pháp luận sẽ được sử dụng.

3.6

Nhiều thử nghiệm được thực hiện trong quy trình triển khai hơn bất kỳ quy trình làm việc nào khác.

Sẽ tốt hơn nếu chia dòng công việc này thành hai dòng công việc riêng biệt, một dòng kết hợp

không kiểm tra các khía cạnh, các khía cạnh khác tất cả các thử nghiệm?

Việc tách nó ra có thể làm mất đi tầm quan trọng của việc thử nghiệm được thực hiện cùng với

tất cả các quy trình làm việc.

3.7

â € œĐộ chính xác là trách nhiệm của nhóm SQA. Hãy thảo luận về tuyên bố này.

Các lập trình viên quan tâm đến việc thực hiện các hiện vật của các giai đoạn trước;

công việc của họ không phải là đảm bảo rằng việc thực hiện là chính xác.

3.8

Bảo trì là hoạt động quan trọng nhất của sản xuất phần mềm và khó thực hiện nhất.

Tuy nhiên, nó bị nhiều chuyên gia phần mềm coi thường và bảo trì

các lập trình viên thường được trả ít hơn các nhà phát triển. Bạn có nghĩ rằng điều này là hợp lý? Nếu không,

bạn sẽ cố gắng thay đổi nó như thế nào?

Nó không hợp lý bởi vì bảo trì là những gì giữ cho một sản phẩm phần mềm có liên quan và hữu ích.

Tất cả công việc của quá trình phát triển ban đầu sẽ bị vô hiệu nếu không thực hiện bảo trì.

Bảo trì không chỉ đơn thuần là sửa lỗi mà còn cung cấp các tính năng mới và hỗ trợ bổ sung

environenments.

3,9

Tại sao bạn nghĩ rằng, như đã nêu trong Phần 3.9, nghỉ hưu thực sự là một sự kiện hiếm?

Phần mềm có xu hướng được dựa vào nhiều hơn thay vì ít hơn. Nó thường là mong muốn cho

phần mềm tiếp tục đáng tin cậy và phù hợp hơn là bị bỏ rơi để mua thứ gì đó mới.

3,10

Do hỏa hoạn tại Phần mềm của Elmerâ € ™, tất cả tài liệu về sản phẩm sẽ bị phá hủy ngay trước đó

nó được giao. Tác động của việc thiếu tài liệu là gì?

Chỉ còn lại mã để tìm hiểu cách hoạt động của phần mềm. Tất cả sẽ phải phân tích kỹ lưỡng

để hiểu hoạt động bên trong: thực tế là không thể thực hiện bảo trì hậu kỳ.

Ngoài ra, khi mất các yêu cầu và phân tích, không thể xác định được mức độ hiệu quả của

nhu cầu được thực hiện.

3,11

Bạn vừa mua Antedeluvian Software Developers, một tổ chức trên bờ vực

phá sản vì công ty đang ở mức trưởng thành 1. Bước đầu tiên bạn sẽ làm là gì

khôi phục tổ chức để sinh lời?

Cố gắng thực hiện một số phương pháp kỹ thuật phần mềm.

Trong ngắn hạn, quá trình chuyển đổi sẽ hỗn loạn và khó thực thi

kỹ thuật mà không bị gián đoạn. Có thể sử dụng một số cấp độ của các quy trình nhanh trong thời gian này; của chúng

khả năng thích ứng và giao tiếp thường xuyên có thể giúp tất cả các bên liên quan phản ứng tốt hơn và nhanh chóng

nội bộ hóa các chính sách mới và các tình huống đột xuất.

3,12

Phần 3.13 nói rằng việc giới thiệu môi trường CASE trong tổ chức là rất ít ý nghĩa

tions ở cấp độ trưởng thành 1 hoặc 2. Giải thích tại sao điều này là như vậy.

Các công cụ CASE thúc đẩy chất lượng và năng suất, nhưng còn hơn thế nữa trong các môi trường thực thi các tiêu chuẩn và

phương pháp luận. Các công cụ CASE thường liên quan đến các thực thể như kiểm soát phiên bản, UML và giao diện trực tuyến

bộ kiểm tra bị thiếu trong môi trường cấp 1 & 2.

3,13

Tác dụng của việc giới thiệu các công cụ CASE (trái ngược với môi trường) trong tổ chức-

tions với mức độ trưởng thành thấp?

Các công cụ CASE có thể rất quan trọng trong việc cấu trúc các môi trường không chuẩn này. Triển khai kiểm soát phiên bản,

bộ thử nghiệm, v.v. có thể dẫn đến sự phát triển ổn định và hiệu quả hơn.

3,14

Mức độ trưởng thành 1, mức ban đầu, đề cập đến sự thiếu vắng của quản lý kỹ thuật phần mềm tốt-

thực hành đề cập. Sẽ không tốt hơn nếu SEI gắn nhãn cấp ban đầu là

mức trưởng thành 0?

Tôi hiểu rằng bạn đang ngụ ý rằng họ "chưa đủ 1 cấp độ trưởng thành", nhưng liệu họ có

bắt đầu đếm ở 0 hoặc 1 phần lớn là không đáng kể miễn là hệ thống đánh số được biết đến.

### Chương 4

4.1

Bạn sẽ tổ chức một nhóm để phát triển một dự án trả lương như thế nào? Giải thich câu trả lơi của bạn.

Vì một hệ thống trả lương đã được thiết lập khá chặt chẽ vào thời điểm này, bản chất khám phá của

đội dân chủ dường như ít áp dụng hơn. Agile có thể được sử dụng, nhưng trưởng nhóm lập trình viên có thể là lý tưởng.

Khả năng rò rỉ thông tin và các lỗ hổng bảo mật khác làm cho việc kiểm tra trở nên đặc biệt quan trọng.

Có ai đó xử lý các trường hợp thử nghiệm, tài liệu, v.v. sẽ rất tuyệt, trong khi người đứng đầu tập trung vào

ứng dụng được thiết kế.

4.2

Bạn sẽ tổ chức một nhóm phát triển phần mềm thông tin liên lạc quân sự hiện đại như thế nào-

đồ? Giải thich câu trả lơi của bạn.

Đây sẽ là một trường hợp hoàn hảo cho đội dân chủ. Trong một tình huống hoàn hảo, bạn sẽ có

một loạt các nhà phát triển có kỹ năng từ nhiều lĩnh vực và kinh nghiệm. Các chuyên gia phát thanh,

xử lý tín hiệu, lập trình hệ thống cấp thấp .. Môi trường cần thúc đẩy giao tiếp cởi mở

và chia sẻ các ý tưởng và cân nhắc về các phương pháp tiếp cận mới.

4.3

Phát biểu Luật Brooksâ € ™ s. Giải thích lý do tại sao nó được giữ.

Như được trình bày trong 'The Mythical Man-Month' của Brook, việc thêm nhiều nhà phát triển sẽ làm tăng số lượng

của những người để cập nhật tốc độ và giao tiếp. Nhiều vấn đề về giao diện và kiểm soát phiên bản hơn phát sinh.

4.4

Bạn vừa thành lập một công ty phần mềm mới. Tất cả nhân viên của bạn đều tốt nghiệp đại học gần đây-

điểm nhấn; đây là công việc lập trình đầu tiên của họ. Có thể thực hiện các nhóm dân chủ trong

tổ chức, và nếu có, làm thế nào?

Không lý tưởng khi không có nhà phát triển có kinh nghiệm, nhưng một nhóm dân chủ có vẻ hợp lý.

Môi trường có thể là một phần mở rộng của điều đó trong khuôn viên trường, nhưng nghiêm ngặt hơn.

Vì mỗi nhân viên sẽ tiếp tục các mối quan tâm và lĩnh vực nghiên cứu khác nhau trong thời gian đi học, điều này

có thể cho phép họ nói ra những ý tưởng có thể không thành hiện thực nếu chúng được đóng gói thành những

các nhiệm vụ.

4,5

Nhóm lập trình sinh viên được tổ chức như một nhóm dân chủ. Điều gì có thể được suy luận về

các học sinh trong đội?

4,6

Một nhóm lập trình sinh viên được tổ chức như một nhóm lập trình trưởng. Điều gì có thể được suy luận

về các học sinh trong đội?

4,7

Để so sánh hai tổ chức nhóm khác nhau, TO1 và TO2, trong một công ty phần mềm lớn,

thí nghiệm sau đây được đề xuất. Cùng một sản phẩm phần mềm sẽ được xây dựng bởi hai

các đội, một đội được tổ chức theo TO1 và đội kia theo TO2. Công ty ước tính

rằng mỗi đội sẽ mất khoảng 18 tháng để xây dựng sản phẩm. Đưa ra ba lý do tại sao điều này

thử nghiệm là không thực tế và không có khả năng mang lại kết quả có ý nghĩa.

4.8

Công ty bạn sở hữu vừa tiếp quản một đối thủ cạnh tranh nhỏ hơn và bạn phát hiện ra rằng một trong những

lập trình viên của họ là một siêu lập trình viên. Làm thế nào để bạn đảm bảo rằng cô ấy không rời đi và lấy

một công việc trong một công ty khác?

4,9

Tại sao các nhóm cho các quy trình nhanh phải dùng chung một máy tính?

4,10

Sự khác biệt giữa nhóm dân chủ và nhóm nguồn mở là gì?

4,11

Bạn sẽ tổ chức một nhóm nguồn mở như thế nào?

4,12

Bạn có muốn làm việc trong một tổ chức sử dụng các nhóm đồng bộ hóa và ổn định không? Giải thích

câu trả lời của bạn.

4,13

Những tổ chức nhóm nào tuân theo Pâ € “CMM?