Contents

A

activity 283 (Hoạt động):

là một đơn vị công việc chính có ngày bắt đầu và ngày kết thúc chính xác, tiêu tốn tài nguyên, chẳng hạn như thời gian sử dụng máy tính hoặc số ngày của con người, và tạo ra các sản phẩm công việc, chẳng hạn như ngân sách, tài liệu thiết kế, lịch trình, mã nguồn hoặc hướng dẫn sử dụng.Một hoạt động bao gồm một tập hợp các nhiệm vụ, một nhiệm vụ là đơn vị công việc nhỏ nhất chịu trách nhiệm giải trình của cấp quản lý.

algorithmic cost estimation model 277 (mô hình thuật toán ước tính chi phí):

là công cụ ước tính, tính toán giá trị của số liệu; ước tính thời lượng và chi phí sau đó có thể được tính toán bằng cách sử dụng mô hình. Nhìn bề ngoài, mô hình thuật toán ước tính chi phí vượt trội hơn so với ý kiến của chuyên gia, bởi vì có thể bị sai lệch và có thể bỏ qua một số khía cạnh của cả sản phẩm đã hoàn thành và sản phẩm mục tiêu. Ngược lại, một mô hình ước tính chi phí theo thuật toán là không thiên vị; mọi sản phẩm đều được xử lý theo cùng một cách.

application composition model 281(Mô hình thành phần ứng dụng):

Là một mô hình của COCOMO II. Mô hình thành phần ứng dụng, dựa trên các điểm đối tượng (tương tự như các điểm chức năng), được áp dụng sớm nhất trong quy trình công việc, khi có sẵn

kiến thức tối thiểu về sản phẩm sẽ được xây dựng. Sau đó, khi có nhiều kiến thức hơn, thì mô hình này không được sử dụng nữa mà là mô hình thiết kế ban đầu sẽ được sử dụng.

B

baseline 284 (đường cơ sở):

Một khi sản phẩm công việc đã được xem xét và thống nhất, nó sẽ trở thành đường cơ sở và chỉ có thể được thay đổi thông qua các thủ tục chính thức.

(VD về việc thay đổi baseline: Sau khi đã quyết định phần nào phải được thay đổi để sửa lỗi, lập trình viên sẽ đóng băng phiên bản hiện tại của phần mềm mà họ sẽ thay đổi. Không có lập trình viên nào khác có thể thực hiện thay đổi đối với bất kỳ phiên bản bị đóng băng. Sau khi lập trình viên bảo trì đã thực hiện các thay đổi và chúng đã được kiểm tra, phiên bản mới được cài đặt, do đó sửa đổi đường cơ sở. Khi một phiên bản mới đã được cài đặt, bất kỳ lập trình viên bảo trì nào khác đều có thể đóng băng phiên bản mới và thực hiện các thay đổi đối với nó. Và phần tạo tác kết quả sẽ trở thành phiên bản đường cơ sở tiếp theo.)

bottom-up approach 277 (Phương pháp tiếp cận từ dưới lên):

Một cách để cố gắng giảm thiểu sai sót do đánh giá tổng thể một sản phẩm là chia sản phẩm thành các thành phần nhỏ hơn. Ước tính về thời lượng và chi phí được thực hiện cho từng thành phần riêng biệt và được kết hợp để cung cấp một con số tổng thể. Cách tiếp cận từ dưới lên này có ưu điểm là ước tính chi phí cho một số thành phần nhỏ hơn thường nhanh hơn và chính xác hơn so với một thành phần lớn. Ngoài ra, quá trình ước tính có thể sẽ chi tiết hơn so với một sản phẩm lớn, nguyên khối.

(Điểm yếu của cách tiếp cận này là một sản phẩm nhiều hơn tổng các thành phần của nó. Với mô hình hướng đối tượng, sự độc lập của các lớp khác nhau giúp cho cách tiếp cận từ dưới lên. Tuy nhiên, sự tương tác giữa các đối tượng khác nhau trong sản phẩm làm phức tạp quá trình ước tính.)

C

COCOMO 278

COCOMO (*Constructive Cost Model - Chi phí xây dựng mô hình*) là một phương pháp được sử dụng để đo kích thước phần mềm.

COCOMO gồm 3 mức, từ mô hình ước tính vĩ mô xử lý sản phẩm tổng thể, mô hình trung gian đến mô hình ước tính vi mô xử lý sản phẩm một cách chi tiết.

Trong đó, mức COCOMO trung gian được áp dụng phổ biến nhất, được áp dụng cho gần như toàn bộ dự án ở những cài đặt đơn giản và thô sơ; nó cũng có thể được áp dụng để tính chi phí ở mức các thành phần của phần mềm.

Bonus: Đặc điểm của COCOMO Trung gian:

+ *Dùng đơn vị KDSI để ước lượng kích thước phần mềm*
+ *Phát triển theo 3 dạng:*
    - *organic (tự nhiên): phát triển từ đầu, không dựa trên framework có sẵn hay phần mềm cũ mà được phát triển tự nhiên*
    - *semi-detached: ngược với organic, có sử dụng phát triển từ PM có sẵn*
    - *embedded: điều khiển thiết bị ngoại vi*

Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng COCOMO?

(note: Phần này có thể chỉ trả lời phần trung gian, mức 1 và 3 không học)

Gồm 3 mức COCOMO:

- COCOMO Cơ bản (Basic):

+ Ưu điểm : Áp dụng tốt để tính nhanh và sơ bộ chi phí cho các dự án nhỏ và vừa
+ Nhược điểm : Bị giới hạn chức năng do không cân nhắc đến các yếu tố như ràng buộc phần cứng, chất lượng nhân sự, kinh nghiệm, trình độ kĩ thuật và công cụ sử dụng

- COCOMO Trung gian (Intermediate)

+ Ưu điểm :
    1. Được áp dụng cho gần như toàn bộ dự án ở những cài đặt đơn giản và thô sơ. Nó cũng có thể được áp dụng để tính chi phí ở mức các thành phần của phần mềm
    2. Khắc phục được nhiều nhược điểm so với các phương pháp khác, ví dụ như so với Line of code thì do dùng đơn vị KDSI nên sẽ tránh được sai số đáng kể khi số dòng lệnh thay đổi do phong cách hoặc ngôn ngữ lập trình; cũng trách được việc phụ thuộc vào chuyên gia so với function point
    3. Được đánh giá các tiêu chí dựa trên điểm số được tổng hợp từ các dự án lớn và được coi như một chuẩn đánh giá
+ Nhược điểm :

    1. Sản phẩm có quá nhiều thành phần sẽ khó tính được bằng COCOMO trung gian
    2. Khó áp dụng trong các dự án cỡ lớn

- COCOMO Chi tiết (Detailed)

+ Ưu điểm : Dễ hiểu, trực quan, đặc biệt hữu ích cho người ước lượng hiểu được ảnh hưởng của những yếu tố khác nhau đến chi phí sản phẩm
+ Nhược điểm : Thành công của pp phụ thuộc vào mô hình áp dụng với yêu cầu của khách hàng.

COCOMO tính đến nhiều tiêu chí hơn hay là function point? Giải thích?

Nhiều hơn, COCOMO tính theo 15 tiêu chí, còn function point có 14 tiêu chí.

Ở FP có 14 tiêu chí nhưng mời chuyên gia về để đánh giá, đây là nhược điểm vì điểm có thể bị biến thiên bởi tính cá nhân của các chuyên gia, các chuyên gia khác nhau có thể cho điểm khác nhau.

Còn ở COCOMO thì có một bảng 15 tiêu chí có các thang điểm theo chuẩn, điểm này được tổng hợp dựa trên điểm đánh giá thực tế từ các dự án lớn. Nên sẽ tránh được việc cho điểm dựa trên tính cá nhân của các chuyên gia.

SW development multiplier của COCOMO thì khác gì TCF của function point?

- Để tính độ phức tạp kỹ thuật TCF của FP thì cần chuyên gia cho điểm từ 0-5 cho 14 tiêu chí.

TCF = DI*1% + 0.65, (0.65<=TCF<=1.35)

trong đó DI = tổng điểm 14 tiêu chí

- Còn ở COCOMO thì có một bảng 15 tiêu chí có các thang điểm theo chuẩn, điểm này được tổng hợp dựa trên điểm đánh giá thực tế từ các dự án lớn. Nên sẽ tránh được việc cho điểm dựa trên tính cá nhân của các chuyên gia.

Khi có điểm của 15 tiêu chí thì nhân tất cả lại với nhau, ta được hệ số nhân.

Phương pháp COCOMO thì dựa trên LOC, mà LOC thì lại có nhiều nhược điểm, nhưng tại sao COCOMO thì lại có nhiều ưu điểm như thế ?

Vì COCOMO dùng đơn vị là KDSI (*Nghìn dòng mã nguồn bàn giao*) thay vì LOC *(dòng lệnh)*

Giả sử khi thay đổi ngôn ngữ lập trình (Java->C); hoặc thay đổi phong cách lập trình (Vị trí dấu ngoặc/Cách xuống dòng..); hoặc không thống nhất trong thời điểm thống kê số dòng code;... sẽ làm chênh lệnh số dòng lệnh một cách đáng kể nếu tính theo đơn vị LOC, nhưng với KDSI (nghìn dòng lệnh) thì sự chênh lệnh sẽ nhỏ, không đáng kể.

Ví dụ 600 dòng sau khi thay đổi -> 900 dòng: tăng lên 1 nửa; nhưng dùng KDSI thì làm tròn đều là 1, nên bỏ qua sự biến thiên đáng kể.

Vì thế dùng đơn vị KDSI sẽ tránh được nhiều nhược điểm của LOC.

Phương pháp COCOMO kết hợp với Function Point nhưng tránh lỗi của FP như thế nào?

Ở FP chia làm 3 mức thì ở COCOMO cũng chia là 3 dạng dự án, nhưng chia theo đặc trưng kỹ thuật của dự án chứ không mời chuyên gia, nên tránh được việc bị ảnh hưởng bởi chuyên gia, vì chuyên

gia thì mang tính cá nhân, con người, có thể biến thiên; còn đặc trưng kỹ thuật của dự án thì phụ thuộc vào bản chất của dự án.

COCOMO II 281

COCOMO II là một bản sửa đổi lớn của COCOMO năm 1981. COCOMO II có thể xử lý nhiều loại kỹ thuật kỹ thuật phần mềm hiện đại, bao gồm hướng đối tượng, các mô hình vòng đời khác nhau, tạo mẫu nhanh, ngôn ngữ thế hệ thứ tư, tái sử dụng và phần mềm COTS. COCOMO II phức tạp hơn đáng kể so với COCOMO ban đầu.

cone of uncertainty? (Mô hình nón của sự không chắc chắn):

Là biểu đồ hình nón của sự không chắc chắn. Trong quản lý dự án, Nón của sự không chắc chắn mô tả sự phát triển của lượng trường hợp không chắc chắn tốt nhất trong một dự án. Khi bắt đầu một dự án, tương đối ít người biết về sản phẩm hoặc kết quả công việc, và do đó, các ước tính có độ không chắc chắn lớn.

mô tả sự phát triển của lượng trường hợp không chắc chắn tốt nhất trong một dự án. Khi bắt đầu một dự án, thông tin tương đối ít được biết về sản phẩm hoặc kết quả công việc, và do đó, các ước tính về có độ không chắc chắn lớn. Vì trong một số tình huống, một tổ chức có thể được yêu cầu đưa ra các ước tính về thời lượng và chi phí trước khi có thể đưa ra các trích dẫn cụ thể. Trong trường hợp xấu nhất, khách hàng có thể nhất quyết trả giá trên cơ sở một hoặc hai giờ thảo luận sơ bộ. Khi ấy ta có tương đối ít thông tin về kết quả công việc và sản phẩm, nên sẽ dẫn đến sự không chắc chắn.

cost 271 (chi phí nội bộ / chi phí cho các developer):

Chi phí nội bộ bao gồm tiền lương của các nhóm phát triển, người quản lý và nhân viên hỗ trợ tham gia vào dự án; chi phí của phần cứng và phần mềm để phát triển sản phẩm; và chi phí chung như tiền thuê nhà, điện nước và tiền lương của quản lý cấp cao. Mặc dù giá cả thường dựa trên chi phí cộng với tỷ suất lợi nhuận, nhưng trong một số trường hợp, các yếu tố kinh tế và tâm lý là quan trọng.

cost estimate 271 (ước tính chi phí) :

Ước tính chi phí là ước tính chi phí của một chương trình, dự án hoặc hoạt động. Dự toán chi phí là sản phẩm của quá trình lập dự toán chi phí. Dự toán chi phí có một giá trị tổng duy nhất và có thể có các giá trị thành phần có thể xác định được.

D

Delphi technique 276 (kỹ thuật Delphi) :

+ Nó cho phép các chuyên gia đi đến thống nhất mà không cần họp nhóm, điều này có thể gây ra tác dụng phụ không mong muốn là một thành viên thuyết phục làm lung lay nhóm.

+ Trong kỹ thuật này, các chuyên gia làm việc độc lập
+ Mỗi thứ tạo ra một ước tính và cơ sở lý luận cho ước tính đó. Sau đó những ước tính và hợp lý này được phân phối cho tất cả các chuyên gia, những người hiện đưa ra ước tính thứ hai. -> Quá trình ước tính và phân phối này tiếp tục cho đến khi các chuyên gia có thể đồng ý trong phạm vi dung sai được chấp nhận.

documentation 291 (tài liệu):

+ Sự phát triển của một sản phẩm phần mềm đi kèm với rất nhiều tài liệu.

+ Tài liệu là một khía cạnh thiết yếu của nỗ lực sản xuất phần mềm. Theo một nghĩa rất thực tế, sản phẩm là tài liệu, bởi vì không có tài liệu thì sản phẩm không thể được duy trì. Lập kế hoạch cho nỗ lực tài liệu đến từng chi tiết, và sau đó đảm bảo rằng kế hoạch được tuân thủ, là một thành phần quan trọng của quá trình sản xuất phần mềm thành công.

duration 282 (khoảng thời gian) :

là khoảng thời gian các nhà phát triển phần mềm sử dụng dự đoán ước tính để hoàn thành những mục tiêu trong sản phẩm. Việc theo dõi cẩn thận các dự đoán phải được thực hiện trong suốt quá trình phát triển, bất kể các kỹ thuật mà các dự đoán đã được thực hiện.

duration estimate 271 (ước tính thời gian):

là quá trình ước tính các khoảng thời gian cần thiết để hoàn thành công việc với mức độ huy động nguồn lực nhất định.

E

early design model 281 (mô hình thiết kế ban đầu):

Mô hình thiết kế ban đầu sử dụng Điểm chức năng chưa điều chỉnh (UFP) làm thước đo kích thước. Mô hình này được sử dụng ở giai đoạn đầu của dự án phần mềm khi không có đủ thông tin về kích thước sản phẩm chưa được phát triển

Thế nào là phần mềm embedded? :

Là một trong ba chế độ của COCOMO trung gian. Nó là những dự án lớn, phức tạp.

Những loại dự án này có bản chất khá phức tạp và liên quan đến 300 KDLOC trở lên. Đội ngũ thực hiện các dự án này không cần thiết phải có nhiều kinh nghiệm, các lập trình viên mới vào nghề hoặc chưa có kinh nghiệm cũng có thể làm việc trong các dự án này. Tuy nhiên, trong khi phát triển các dự án này, người dùng cần tuân theo các ràng buộc nghiêm ngặt (phần cứng, phần mềm, con người, thời hạn) và nó phải đáp ứng các yêu cầu nghiêm ngặt của người dùng. Ví dụ về các dự án này là các hệ thống phần mềm được sử dụng trong công nghệ điện tử hàng không và quân sự.

efficiency 273 (hiệu quả)

hằng số d là thước đo hiệu quả (năng suất) của quá trình phát triển phần mềm trong tổ chức đó. Chi phí tỷ lệ thuận với quy mô, hằng số tỷ lệ d được xác định bằng bình phương nhỏ nhất phù hợp với dữ liệu chi phí liên quan đến các sản phẩm do tổ chức đó phát triển trước đó.

expert judgment by analogy 276 (Phán đoán của chuyên gia bằng phép tương tự)

Trong phần đánh giá chuyên môn bằng phép tương tự, một số chuyên gia được tham khảo ý kiến. Một chuyên gia đưa ra ước tính bằng cách so sánh sản phẩm mục tiêu với các sản phẩm đã hoàn thành mà chuyên gia đã tham gia tích cực và lưu ý những điểm giống và khác nhau. Ví dụ: một chuyên gia có thể so sánh sản phẩm mục tiêu với một sản phẩm tương tự được phát triển cách đây 2 năm mà dữ liệu được nhập ở chế độ hàng loạt, trong khi sản phẩm mục tiêu là sản phẩm thu thập dữ liệu trực tuyến.

external cost 271 (chi phí bên ngoài)

là mức giá mà khách hàng sẽ phải trả.

Khách hàng có thể từ chối một giá thầu có thể thấp hơn đáng kể so với tất cả các giá thầu khác với lý do rằng chất lượng của sản phẩm kết quả có thể cũng sẽ thấp hơn đáng kể. Do đó, nhóm phát triển có thể cố gắng đưa ra giá thầu thấp hơn một chút, nhưng không quá thấp so với giá thầu của đối thủ cạnh tranh.

F

FFP metric 273 (số liệu FFP)

số liệu FFP. Đây là 1 phương pháp dùng để ước lượng kích thước của sản phẩm dựa trên 3 yếu tố cấu trúc cơ bản của một sản phẩm là File, Flow, Process.

Trong đó: - File là một tập hợp các bản ghi liên quan đến logic hoặc vật lý trong sản phẩm, là các giao dịch hoặc các tệp tạm thời bị loại trừ

- Flow là kết nối giữa dữ liệu sản phẩm và môi trường, chẳng hạn như màn hình hoặc báo cáo

- Process là một lôgic được xác định theo chức năng hoặc thao tác số học về dữ liệu ; các ví dụ bao gồm sắp xếp, xác thực hoặc cập nhật.

Thế nào là Flow trong FFP?

Luồng (flow) trong FFP là giao diện dữ liệu giữa sản phẩm và môi trường, chẳng hạn như màn hình hoặc báo cáo.

Thế nào là process trong FFP?

Quy trình (process) trong FFP là một thông số để tính kích thước sản phẩm bằng FFP, là một thao tác logic hoặc số học được xác định về mặt chức năng đối với dữ liệu; các ví dụ bao gồm sắp xếp, xác thực hoặc cập nhật.

Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng FFP?

- Ưu điểm :

+ Chính xác hơn LOC, ổn định hơn (nhiều tham số hơn)

+ Áp dụng ở thời điểm sớm hơn LOC (ở cuối pha thiết kế)

- Nhược điểm :

+ Phụ thuộc vào hằng số b

+ Nó chỉ tính số lượng chứ không tính định thức

+ Chưa cover được hết đặc trưng của dự án

+ Vẫn có biến thiên (như số lượng file)

FFP thì có ưu điểm gì hơn so với LOC ?

- Ưu điểm :

+ Chính xác hơn LOC, ổn định hơn (nhiều tham số hơn)

+ Áp dụng ở thời điểm sớm hơn LOC (ở cuối pha thiết kế)

function point (FP) (Điểm chức năng) 273

Một điểm chức năng (FP) là một thành phần của phát triển phần mềm giúp để xấp xỉ chi phí phát triển ban đầu trong quá trình này. Nó là một quá trình trong đó xác định các chức năng cần thiết

và tính phức tạp của họ trong một phần của phần mềm để ước tính kích thước và phạm vi của phần mềm sau khi hoàn thành.

Một điểm chức năng có một số lợi ích, bao gồm gia tăng năng suất và giảm nguy cơ lạm phát của mã tạo ra. điểm chức năng có thể được bắt nguồn từ yêu cầu của phần mềm và có thể được ước tính trong giai đoạn đầu của phát triển phần mềm, trước khi dòng thực tế của mã có thể được xác định. Số lượng các điểm chức năng trong một mã phụ thuộc vào chức năng phức tạp.

Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng Function Point?

- Ưu điểm :

+ Độ chính xác cao hơn, ổn định hơn FFP

+ Cover được nhiều tiêu chí hơn FFP

+ Được tính toán bởi các chuyên gia dẫn đến khách quan hơn

+ Áp dụng ở cuối pha thiết kế

- Nhược điểm : Đánh giá phụ thuộc dựa vào chuyên gia (mang sự chủ quan)

Tại sao nói function point chịu ảnh hưởng chủ quan của các chuyên gia?

FP mời chuyên gia đánh giá cả 2 mức: mức 1 (xem dự án đơn giản hay phức tạp) và mức 2 (độ phức tạp kỹ thuật). Cùng 1 tiêu chí nhưng các chuyên gia khác nhau có thể đánh giá, cho điểm ở mức độ khác nhau. Khi kết hợp ý kiến của chuyên gia khác nhau cũng ra kết quả khác nhau.

Trong cách tính về FP, các chuyên gia sẽ đánh giá về độ phức tạp của UFP và đánh giá thang điểm kỹ thuật của 14 yếu tố của TCF từ đầu đến cuối đều là do các chuyên gia đánh giá để tính

→ FP chịu ảnh hưởng chủ quan của các chuyên gia

I

IEEE software project management plan 286 Kế hoạch quản lý dự án phần mềm

Người quản lý dự án phần mềm chuẩn bị một tài liệu trên cơ sở quyết định cuối cùng trong quá trình lập kế hoạch dự án. Tài liệu này được gọi là Tài liệu Kế hoạch Quản lý Dự án Phần mềm hoặc tài liệu SPMP.

Tài liệu SPMP là một tài liệu được tổ chức tốt có chứa kế hoạch chi tiết của dự án.

Nó sẽ có thông tin chi tiết về mục tiêu dự án, ước tính dự án, lịch trình dự án, nguồn lực dự án, nhân sự dự án, kế hoạch quản lý rủi ro, giám sát dự án, kiểm soát dự án và các hoạt động sai sót khác.

internal cost 271 Chi phí nội bộ

là chi phí mà các developer phải trả. Nó bao gồm lương cho team phát triển, quản lý và các thành viên tham gia giúp đỡ cho dự án; chi phí cho phần cứng và phần mềm để phát triển phần mềm; cuối cùng là chi phí ở mức trên như là tiền thuê, các tiện ích, và lương cho các quản lý cấp cao

L

lines of code (LOC) 272 (LOC đếm tổng số dòng mã nguồn trong một dự án)

"**dòng mã** "(LOC) là bất kỳ dòng văn bản nào trong mã không phải là chú thích hoặc dòng trống, trong bất kỳ trường hợp nào về số lượng câu lệnh hoặc đoạn câu lệnh trên dòng. LOC rõ ràng bao gồm tất cả các dòng chứa tệp tiêu đề chương trình, khai báo bất kỳ biến nào và các câu lệnh thực thi và không thực thi. Vì Lines of Code (LOC) chỉ tính khối lượng mã, bạn chỉ có thể sử dụng nó để so sánh hoặc ước tính các dự án sử dụng cùng một ngôn ngữ và được mã hóa bằng cùng một tiêu chuẩn mã hóa.

Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng LOC?

Thuận lợi :

- Số liệu được sử dụng nhiều nhất trong ước tính chi phí.
- Các thay thế của nó có nhiều vấn đề so với số liệu này.
- Nó rất dễ dàng trong việc ước tính những nỗ lực.

Nhược điểm:

- Tăng năng suất trong khi viết nhiều dòng mã hơn.
- Mã kém hiệu quả hơn.
- Nó không xem xét sự phức tạp mà chỉ dựa trên 1 thông tin là số dòng lệnh
- Phụ thuộc ngôn ngữ lập trình;
- Phụ thuộc phong cách lập trình;
- Thời điểm sớm nhất có thể áp dụng pp này là cuối pha cài đặt => khó khả thi, chỉ phù hợp với những dự án nhỏ, đơn giản
- Không thống nhất trong cách tính số dòng lệnh;

Thế nào là thousands of lines of code (KLOC)

KLOC (hàng nghìn dòng mã) là thước đo truyền thống để đánh giá độ lớn của một chương trình máy tính hoặc thời lượng hoặc bao nhiêu người để viết nó. Mã được đo thường là mã nguồn . Vì ngôn ngữ nguồn cấp cao hơn (chẳng hạn như C ++) biên dịch thành nhiều dòng mã máy hơn so với ngôn ngữ cấp thấp hơn (chẳng hạn như ngôn ngữ hợp dịch), một KLOC của các câu lệnh C ++ sẽ tạo ra một chương trình lớn hơn (theo số byte s) hơn một KLOC của các câu lệnh ngôn ngữ hợp ngữ.

KLOC đã được sử dụng như một thước đo sơ bộ về năng suất của lập trình viên, như trong "Bạn có thể viết bao nhiêu dòng mã một ngày?" Tuy nhiên, biện pháp này không xem xét hiệu quả của mã. Nhiều yếu tố khác rõ ràng ảnh hưởng đến năng suất.

Sai sót trên mỗi KLOC là một thước đo phổ biến được sử dụng làm mục tiêu hoặc để đánh giá chất lượng mã.

M

milestone 284

Ngày mà một sản phẩm công việc được coi là hoàn thành được gọi là một **mốc** quan trọng. Để xác định liệu một sản phẩm làm việc có thực sự đạt đến một mốc quan trọng hay không, trước tiên nó phải vượt qua một loạt các bài đánh giá được thực hiện bởi các thành viên trong nhóm, quản lý hoặc khách hàng.

money 284

Một gói công việc xác định không chỉ sản phẩm công việc mà còn xác định các yêu cầu của nhân viên, thời gian, nguồn lực, tên của cá nhân chịu trách nhiệm và tiêu chí chấp nhận cho sản phẩm công việc. Tất nhiên, **tiền** là một thành phần quan trọng của kế hoạch. Một ngân sách chi tiết phải được vạch ra và phân bổ tiền, như một chức năng của thời gian, cho các chức năng và hoạt động của dự án. Vấn đề làm thế nào để lập một kế hoạch sản xuất phần mềm sẽ được giải quyết tiếp theo.

N

Thế nào là nominal effort ?

Nỗ lực danh nghĩa, được tính dựa trên 2 thông số: độ dài của sản phẩm trong KDSI và chế độ phát triển của sản phẩm, thước đo mức độ khó khăn nội tại của việc phát triển sản phẩm

đó. Có ba chế độ: hữu cơ (nhỏ và đơn giản), semidetached (kích thước trung bình) và nhúng (phức tạp)

Số lượng lao động sẽ được yêu cầu để hoàn thành một nhiệm vụ. Nó được đo bằng đơn vị người-tháng.

$$\text{Nominal effort} = 3.2 \times (\text{KDSI})^{1.05} \text{ person-months}$$

Nghìn dòng mã được cung cấp *(KDSI).*

O

Thế nào là phần mềm organic ?

Là một trong ba chế độ của COCOMO trung gian. Nó là những dự án nhỏ và đơn giản.

Organic có nghĩa là tự nhiên, nguyên chất; thì phần mềm organic là các phần mềm được phát triển từ đầu, không dựa trên framework có sẵn, hay phần mềm cũ,... mà được phát triển tự nhiên từ ban đầu.

Loại dự án organic không quá cồng kềnh và bao gồm 50 KDLOC (một kg dòng mã được phân phối trở xuống). Nó đòi hỏi đội ngũ có kinh nghiệm trước cùng với sự hiểu biết sâu sắc về dự án phần mềm. Các loại dự án này dễ dàng phát triển và không bị giới hạn về thời gian và ví dụ của các dự án đó là hệ thống kinh doanh, hệ thống quản lý bảng lương, hệ thống quản lý hàng tồn kho, vv.

P

planning 268 (lập kế hoạch)

Để tạo ra một sản phẩm phần mềm lớn cần có thời gian và nguồn lực. Và, giống như bất kỳ dự án xây dựng lớn nào khác, lập kế hoạch cẩn thận khi bắt đầu dự án có lẽ là yếu tố quan trọng nhất giúp quyết định thành công hay thất bại.

Tuy nhiên, việc lập kế hoạch lúc ban đầu này không có nghĩa là đủ mà nó giống như kiểm thử, phải tiếp tục trong suốt quá trình phát triển và bảo trì phần mềm. Các kế hoạch diễn ra liên tục, kéo dài cho đến khi các tài liệu cụ thể đã được vạch ra nhưng phải trước khi các hoạt động thiết kế bắt đầu. Tại thời điểm này trong quá trình, các ước tính chi phí và thời lượng có ý nghĩa được tính toán và lập một kế hoạch chi tiết để hoàn thành dự án.

post architecture model 281 (mô hình hậu kiến trúc)

Khi các nhà phát triển có được lượng thông tin tối đa, mô hình hậu kiến trúc sẽ được đưa vào sử dụng. Mô hình này sử dụng các điểm chức năng hoặc các dòng mã (KDSI).

Đầu ra từ COCOMO trung gian là ước tính chi phí và quy mô; đầu ra từ mỗi mô hình trong số ba mô hình của COCOMO II là một loạt các ước tính về chi phí và quy mô. Theo đó, nếu ước tính khả năng nỗ lực là E, thì mô hình thành phần ứng dụng trả về phạm vi (0,50E, 2,0 E) và mô hình hậu kiến trúc trả lại phạm vi (0,80E, 1,25E). Điều này phản ánh mức độ chính xác ngày càng tăng của sự phát triển của các mô hình COCOMO II

price 271 (mức giá)

Mặc dù giá cả thường dựa trên chi phí cộng với tỷ suất lợi nhuận, trong một số trường hợp, các yếu tố kinh tế và tâm lý là rất quan trọng. Ví dụ, các nhà phát triển đang rất cần công việc có thể chuẩn bị tính phí khách hàng. Một tình huống khác nảy sinh khi hợp đồng được trao trên cơ sở hồ sơ dự thầu. Khách hàng có thể từ chối một giá thầu có thể thấp hơn đáng kể so với tất cả các giá thầu khác với lý do rằng chất lượng của sản phẩm kết quả có thể cũng sẽ thấp hơn đáng kể. Do đó, nhóm phát triển có thể cố gắng đưa ra giá thầu sẽ thấp hơn một chút, nhưng không đáng kể, thấp hơn giá thầu của đối thủ cạnh tranh.

productivity 273 (năng suất)

Ta có công thức: C = d x S (C là cost và S là size)

Hằng số d là thước đo hiệu quả (năng suất) của quá trình phát triển phần mềm trong tổ chức đó. Kích thước của một sản phẩm đơn giản là tổng số tệp, quy trình, một số lượng có thể được xác định sau khi thiết kế kiến trúc hoàn thành. Khi đó, chi phí tỷ lệ thuận với quy mô, hằng số tỷ lệ d được xác định bằng bình phương nhỏ nhất đối với dữ liệu chi phí liên quan đến các sản phẩm do tổ chức đó phát triển trước đó.

project function 283 (chức năng dự án)

Một công việc có thể được chia thành 2 loại. First is work that continues throughout the project and does not relate to any specific workflow of software development. Such work is termed a project function. Ví dụ như quản lý dự án và kiểm soát chất lượng

Có ba loại công việc trong một kế hoạch quản lý dự án phần mềm: các chức năng dự án được thực hiện trong suốt dự án, các hoạt động (các đơn vị công việc chính) và các nhiệm vụ (các đơn vị công việc nhỏ).

R

Rayleigh distribution 282 (Phân phối Rayleigh)

Phân phối Rayleigh là một phân phối của hàm mật độ xác suất liên tục. Phân phối Rayleigh là một ước lượng rất tốt về cách tiêu thụ tài nguyên (Rc) theo công thức

$$R_c = \frac{t}{k^2} e^{-t^2/2k^2} \qquad 0 \le t < \infty$$

resources (tài nguyên) 282:

Thuật ngữ Resources có nghĩa là Tài nguyên được ngành Công nghệ thông tin (CNTT) sử dụng để chỉ tất cả những vật mang tin có thể được tiếp cận trong không gian vật chất như một cuốn sách, một bài báo…; đồng thời, hiện hữu trong không gian điện tử như một tập tin máy tính, một sưu tập số…

review (đánh giá) 284:

Một đánh giá tình trạng của sản phẩm hoặc trạng thái của dự án để xác định sự khác biệt từ kế hoạch (plan) và để đưa ra cải tiến. Ví dụ bao gồm xem xét quản lý (management review), xem xét (review), xem xét kỹ thuật (technical review), kiểm tra (inspection), và walkthrough.

software development effort multipliers (SPMP) 278:

Là hệ số nhân nỗ lực phát triển phần mềm.

Ví dụ

Phần mềm xử lý giao tiếp dựa trên bộ vi xử lý cho mạng chuyển tiền điện tử với độ tin cậy, hiệu năng, lịch phát triển và các yêu cầu giao diện cao (Microprocessor-based communications processing software for electronic funds transfer network with high reliability, performance, development schedule, and interface requirements)

Bước 1. Ước lượng chiều dài của phần mềm sản phẩm o 10,000 câu lệnh được chuyển giao (10 KDSI)

Bước 2. Ước lượng chế độ phát triển o Chế độ phức tạp ("nhúng"-"embedded")

Bước 3. Tính công sức danh nghĩa o (Công sức danh nghĩa)Nominal effort = 2.8 ´ (10) 1.20 = 44 person-months Bước 4.

Nhân giá trị danh nghĩa với 15 lần chi phí phát triển phần mềm o Product of effort multipliers = 1.35 (Software development effort multipliers)

S

Thế nào là phần mềm semi-detached?

Là một trong ba chế độ của COCOMO trung gian. Nó là những dự án có kích thước vừa phải.(medium size).

Dự án này nằm giữa các dự án organic và embedded. Tương tự, sự phức tạp của những dự án này nằm giữa dự án hữu cơ và dự án nhúng, nơi có thể gửi ít hơn 300 KLOC dòng mã. Nó đòi hỏi một

người có kinh nghiệm trung bình và thời hạn trung bình để sản xuất sản phẩm. Những dự án này có thể liên quan đến hệ điều hành, thiết kế cơ sở dữ liệu, thiết kế trình biên dịch, v.v.

(Hoàng Dương note)

Là một dạng phát triển phần mềm của COCOMO Trung gian.

Ngược lại với phần mềm organic - tức là các phần mềm được xây dựng tự nhiên, từ đầu; phần mềm semi-detached là chỉ các phần mềm được phát triển dựa trên các phần mềm cũ, các phần mềm đã có sẵn mà không phải được xây dựng từ đầu.

T

task (nhiệm vụ) 283:

một nhiệm vụ là đơn vị công việc nhỏ nhất chịu trách nhiệm giải trình của cấp quản lý. Do đó có ba loại của công việc trong kế hoạch quản lý dự án phần mềm: các chức năng của dự án được thực hiện trong suốt dự án, hoạt động (đơn vị công việc chính), và nhiệm vụ (đơn vị công việc nhỏ).

technical complexity factor (TCF) 274:

Đây là một thước đo của ảnh hưởng của 14 yếu tố kỹ thuật, chẳng hạn như tỷ lệ giao dịch cao, tiêu chí hiệu suất (đối với ví dụ, thông lượng hoặc thời gian phản hồi) và cập nhật trực tuyến;

Mỗi yếu tố trong số 14 yếu tố này được gán một giá trị từ 0 (“không hiện tại hoặc không có dòng chảy nào ”) đến 5 (“ dòng chảy mạnh xuyên suốt ”). 14 số kết quả được tính tổng, mang lại tổng mức độ lạm phát (DI). TCF sau đó được cung cấp bởi:

TCF = 0.65 + 0.01* DI

Vì DI có thể thay đổi từ 0 đến 70, TCF thay đổi từ 0,65 đến 1,35.

TCF của function point thì khác gì hằng số b của FFP?

- Hằng số b của FFP: là thước đo độ hiệu quả được tính dựa trên bình phương tối thiểu phù hợp với chi phí liên quan đến sản phẩm được tổ chức phát triển trước đó

- TCF của function point: là hệ số phức tạp kĩ thuật. Nó là thước đo của 14 yếu tố kỹ thuật,chẳng hạn như tỷ lệ giao dịch cao, tiêu chí hiệu suất (ví dụ: thông lượng hoặc thời gian phản hồi) và cập nhật trực tuyến; Mỗi yếu tố trong số 14 yếu tố này được gán một giá trị từ 0 (“không có hoặc không có ảnh hưởng”) đến 5 (“ảnh hưởng mạnh mẽ xuyên suốt”). sau đó tổng hợp để đưa ra mức độ ảnh hưởng(DI).

TCF = 0.65 + 0.01 * DI

trong đó DI là mức độ ảnh hưởng, DI là tổng của 14 yếu tố kỹ thuật bên trên

test planning 288 :

là việc lên kế hoạch các hoạt động cần được thực hiện để đạt được mục đích của việc thử nghiệm. Đây là 1 thành phần thường bị bỏ qua trong SPMP,nhưng như mọi hoạt động khác trong phát triển phần mềm, kiểm thử cũng phải được lập kế hoạch. SPMP phải bao gồm nguồn lực để kiểm tra và lịch trình chi tiết chỉ ra rõ ràng việc kiểm tra sẽ được thực hiện trong mỗi quy trình làm việc.

thousand delivered source instructions (KDSI) (nghìn dòng mã được cung cấp) 272:

là 1 đơn vị dùng để ước lượng kích thước phần mềm, đây là 1 đơn vị được dùng trong phương pháp COCOMO.

KDSI thì có ưu nhược điểm gì so với KLOC?

KDSI: Kilo Delivered Source Instructions (Nghìn mã nguồn được bàn giao)
KLOC: Kilo Line of Code (Nghìn dòng lệnh)
Cả 2 đều làm giảm sự biến thiên dòng lệnh so với LOC do đều tính theo đơn vị nghìn dòng.

Ưu điểm :

- Có khả năng đánh giá chính xác hơn về năng suất của 1 LTV vì KDSI có quan tâm đến hiệu quả của code còn KLOC chỉ đánh giá việc năng suất chứ ko đánh giá đến hiệu quả của dòng code đó.
- Tránh được sự không thống nhất trong cách tính số dòng lệnh: Ví dụ những dòng code comment thì theo KLOC có thể vẫn tính vào tổng số dòng lệnh, tuy nhiên KDSI chỉ tính những dòng code được bàn giao, không tính phần cmt. Vậy nên sẽ không có trường hợp không thống nhất trong cách tính.

Nhược điểm:

- Nếu sản phẩm có quá nhiều thành phần sẽ gây ra khó khăn trong việc tính toán.

training (Đào tạo) 290

là 1 quá trình học các kĩ năng cần thiết để làm 1 công việc cụ thể hay hoạt động nào đó. Hoạt động đào tạo thường được ngụ ý rằng chỉ có người dùng mới cần phải đào tạo. Nhưng trong thực tế, các thành viên của nhóm phát triển cũng có thể cần phải đào tạo, bắt đầu bằng việc đào tạo về lập kế hoạch và ước lượng phần mềm. Khi có các kĩ thuật phát triển phần mềm mới , chẳng hạn như kỹ thuật thiết kế mới hoặc quy trình thử nghiệm mới được sử dụng, đào tạo phải được cung cấp để cho mọi thành viên trong nhóm biết cách sử dụng kỹ thuật mới.Khi nhu cầu đào tạo đã được xác định và kế hoạch đào tạo đã được lập ra, kế hoạch đó phải được tích hợp vào SPMP.

U

unadjusted function points (UFP) (Điểm chức năng chưa được điều chỉnh):

UFP là 1 thành phần để tính toán FP(function points) .Đây là thước đo dùng để tính toán độ phức tạp các thành phần của sản phẩm.UFP được thực hiện qua 3 bước:

1.Mỗi thành phần của sản phẩm — Input, Output, Inquiry, Master file và Interface — phải được phân loại thành 3 mức đơn giản, trung bình hoặc phức tạp

2.Mỗi thành phần được gán một số điểm chức năng tùy thuộc vào cấp độ của nó.

3.Các điểm chức năng được gán cho mỗi thành phần sau đó được tổng hợp lại, tạo ra các điểm chức năng chưa được điều chỉnh (UFP).

W

work package 284 (Gói công việc) :

là 1 tập hợp bao gồm work product và các thành phần quan trọng để cấu tạo nên 1 work product là yêu cầu của nhân viên, thời hạn, nguồn lực,tên của cá nhân chịu trách nhiệm và tiêu chí chấp nhận cho 1 work product

work product (Sản phẩm công việc)283

là 1 kết quả được tạo ra từ các hoạt động. Hoạt động là một đơn vị công việc chính có ngày bắt đầu và ngày kết thúc; tiêu tốn tài nguyên, chẳng hạn như thời gian sử dụng máy tính hoặc số ngày làm việc và kết quả là các work product, ví dụ như ngân sách, tài liệu thiết kế, lịch trình, mã nguồn hoặc hướng dẫn sử dụng.