Chương 9 Lập kế hoạch và Ước tính

Mục tiêu học tập

Sau khi nghiên cứu chương này, bạn sẽ có thể

- Giải thích tầm quan trọng của việc lập kế hoạch.
- Ước tính quy mô và chi phí xây dựng một sản phẩm phần mềm.
- Đánh giá cao tầm quan trọng của việc cập nhật và theo dõi các ước tính.
- Lập kế hoạch quản lý dự án phù hợp với tiêu chuẩn IEEE.

Những thách thức khi xây dựng một sản phẩm phần mềm không có giải pháp dễ dàng. Để kết hợp với nhau một sản phẩm phần mềm lớn cần thời gian và tài nguyên. Và, giống như bất kỳ công trình xây dựng lớn nào khác dự án, lập kế hoạch cẩn thận khi bắt đầu dự án có lẽ là yếu tố quan trọng nhất- yếu tố tant phân biệt thành công và thất bại. Kế hoạch ban đầu này, tuy nhiên, không phương tiện là đủ. Lập kế hoạch, giống như kiểm thử, phải tiếp tục trong suốt quá trình phát triển phần mềm- quy trình cố vấn và bảo trì. Bất chấp nhu cầu lập kế hoạch liên tục, những các hoạt động đạt đến đỉnh điểm sau khi các thông số kỹ thuật đã được phác thảo nhưng trước khi hoạt động thiết kế- quan hệ bắt đầu. Tại thời điểm này trong quá trình, khoảng thời gian có ý nghĩa và ước tính chi phí là được tính toán và lập kế hoạch chi tiết để hoàn thành dự án.

Trong chương này, chúng tôi phân biệt hai loại **quy hoạch này** , quy hoạch tiến hành xuyên suốt dự án và việc lập kế hoạch căng thẳng phải được thực hiện sau khi cụ thể- hàng tấn đã hoàn tất.

9.1 Lập kế hoạch và Quy trình Phần mềm

Lý tưởng nhất là chúng tôi muốn lập kế hoạch cho toàn bộ dự án phần mềm ngay từ đầu của chương trình cess, và sau đó thực hiện theo kế hoạch đó cho đến khi phần mềm mục tiêu cuối cùng đã được chuyển đến khách hàng. Tuy nhiên, điều này là không thể, vì chúng tôi thiếu đủ thông tin trong quá trình ban đầu

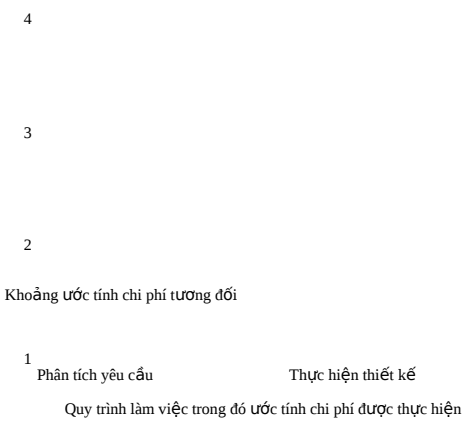
Trang 2

quy trình làm việc để có thể vạch ra một kế hoạch có ý nghĩa cho dự án hoàn chỉnh. Ví dụ, trong quy trình làm việc yêu cầu, bất kỳ loại lập kế hoạch nào (ngoại trừ chỉ cho yêu cầu-bản thân quy trình làm việc của ments) là vô ích.

Có một thế giới khác biệt giữa thông tin do nhà phát triển sử dụng tại cuối quy trình yêu cầu và cuối quy trình phân tích, tương tự như sự khác biệt giữa bản phác thảo thô và bản thiết kế chi tiết. Bởi hết các yêu cầu quy trình làm việc, các nhà phát triển tốt nhất phải có hiểu biết không chính thức về những gì khách hàng cần. Ngược lại, khi kết thúc quy trình phân tích, lúc đó khách hàng ký một tài liệu cho biết chính xác những gì sẽ được xây dựng, các nhà phát triển đánh giá chi tiết về hầu hết (nhưng thường vẫn không phải là tất cả) các khía cạnh của sản phẩm mục tiêu. Đây là điểm sớm nhất trong tại đó có thể xác định chính xác khoảng thời gian và ước tính chi phí.

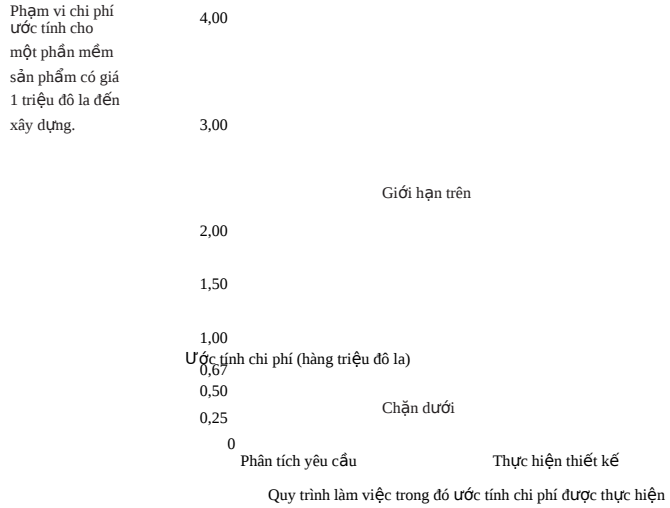
Tuy nhiên, trong một số tình huống, một tổ chức có thể được yêu cầu đưa ra thời hạn và ước tính chi phí trước khi các thông số kỹ thuật có thể được lập. Trong trường hợp xấu nhất, khách hàng có thể nhấn mạnh vào một giá thầu trên cơ sở một hoặc hai giờ thảo luận sơ bộ. Hình 9.1 cho thấy làm thế nào vấn đề này có thể được. Dựa trên một mô hình trong [Boehm và cộng sự, 2000], nó mô tả mối quan hệ nhiều ước tính chi phí cho các quy trình công việc khác nhau của vòng đời. Ví dụ, giả sử rằng, khi một sản phẩm vượt qua kiểm tra chấp nhận ở cuối quy trình triển khai và được giao cho khách hàng, giá của nó là 1 triệu đô la. Nếu một ước tính chi phí có được thực hiện giữa chừng trong quy trình công việc yêu cầu, có khả năng là ở đâu đó trong khoảng ($ 0,25 triệu, $ 4 triệu), như trong Hình 9.2. Tương tự, nếu ước tính chi phí đã được thực hiện giữa chừng trong quá trình phân tích, phạm vi khả năng ước tính sẽ giảm xuống (0,5 triệu đô la, 2 triệu đô la). Hơn nữa, nếu ước tính chi phí đã được thực hiện vào cuối quy trình phân tích, tức là vào thời điểm thích hợp, kết quả có lẽ sẽ nằm trong phạm vi tương đối rộng (0,67 triệu đô la, 1,5 triệu đô la). Tất cả bốn điểm được đánh dấu trên các đường giới hạn trên và dưới trong Hình 9.2, có thang logarit trên trục tung. Mô hình này được gọi là **hình nón của sự không chắc chắn** . Nó là

HÌNH 9.1
Một mô hình cho ước tính người thân phạm vi của một chi phí ước tính cho mỗi vòng đời quy trình làm việc.



Trang 3

HÌNH 9.2



Rõ ràng từ Hình 9.1 và 9.2 rằng ước tính chi phí không phải là một khoa học chính xác; lý do cho điều này được nêu trong Phần 9.2.

Dữ liệu dựa trên mô hình hình nón của sự không chắc chắn đã cũ, bao gồm năm tỷ lệ các vị trí nộp cho Phòng Hệ thống Điện tử của Lực lượng Không quân Hoa Kỳ [Devenny, 1976], và các kỹ thuật ước lượng đã được cải thiện kể từ thời điểm đó. Tuy nhiên, hình dạng tổng thể của đường cong trong Hình 9.1 có thể không thay đổi quá mức. Do đó, màng cứng sớm tion hoặc ước tính chi phí, nghĩa là, một ước tính được thực hiện trước khi các thông số kỹ thuật đã được ký kết do khách hàng bật ra, có thể kém chính xác hơn đáng kể so với ước tính được đưa ra khi đã tích lũy đủ dữ liệu.

Bây giờ chúng ta kiểm tra các kỹ thuật để ước tính thời lượng và chi phí. Giả định thông qua- phần còn lại của chương này là quy trình phân tích đã được hoàn thành; đó là, ước tính và lập kế hoạch có ý nghĩa bây giờ có thể được thực hiện.

9.2 Thời lượng và chi phí ước tính

Ngân sách là một phần không thể thiếu trong bất kỳ kế hoạch quản lý dự án phần mềm nào. Trước khi thiết kế bắt đầu, khách hàng cần biết họ sẽ phải trả bao nhiêu cho sản phẩm. Nếu nhóm phát triển đánh giá thấp chi phí thực tế, tổ chức phát triển có thể mất tiền vào dự án. Mặt khác, nếu nhóm phát triển đánh giá quá cao, thì khách hàng có thể quyết định rằng, trên cơ sở phân tích chi phí-lợi ích hoặc lợi tức đầu tư, không có ích gì khi sản phẩm được xây dựng. Ngoài ra, khách hàng có thể giao công việc cho một tổ chức phát triển khác có ước tính hợp lý hơn. Dù bằng cách nào, nó là rõ ràng rằng ước tính chi phí chính xác là rất quan trọng.

Trên thực tế, có hai loại chi phí liên quan đến việc phát triển phần mềm. Đầu tiên là **chi phí nội bộ** , **chi phí** cho các nhà phát triển; thứ hai là **chi phí bên ngoài** , **giá** mà khách hàng sẽ trả tiền. Chi phí nội bộ bao gồm tiền lương của các nhóm phát triển, quản lý- ers, 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, tiền điện nước và tiền lương của quản lí cấp cao. Mặc dù giá 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 quan trọng. Ví dụ, các nhà phát triển những người thực sự cần công việc có thể được chuẩn bị để tính phí khách hàng bằng giá. Một khác tình huống phát sinh khi hợp đồng được trao trên cơ sở hồ sơ dự thầu. Khách hàng có thể từ chối giá thầu thấp hơn đáng kể so với tất cả các giá thầu khác vì chất lượng của kết quả sản phẩm có lẽ cũng sẽ thấp hơn đáng kể. Do đó, một nhóm phát triển

có thể cố gắng đưa ra một giá thầu sẽ thấp hơn một chút nhưng không đáng kể so với giá thầu tin rằng sẽ là giá thầu của đối thủ cạnh tranh.

Một phần quan trọng khác của bất kỳ kế hoạch nào là ước tính thời gian của dự án. Khách hàng chắc chắn muốn biết khi nào thành phẩm sẽ được giao. Nếu sự phát triển tổ chức không thể giữ đúng lịch trình của mình, thì tốt nhất là tổ chức mất uy tín, tại các điều khoản hình phạt tồi tệ nhất được viện dẫn. Trong mọi trường hợp, những người quản lý chịu trách nhiệm về phần mềm kế hoạch quản lý dự án có rất nhiều việc phải làm. Ngược lại, nếu sự phát triển tổ chức đánh giá quá cao thời gian cần thiết để xây dựng sản phẩm, khi đó sẽ có cơ hội tốt rằng khách hàng sẽ đi nơi khác.

Thật không may, không có nghĩa là dễ dàng để có được một **ước tính chi phí** chính xác và **thời gian ước tính** . Có quá nhiều biến liên quan để có thể xử lý chính xác một trong hai chi phí hoặc thời lượng. Một khó khăn lớn là yếu tố con người. Hơn 40 năm trước, Sackman và đồng nghiệp đã quan sát thấy sự khác biệt lên tới 28-1 giữa các cặp lập trình viên [Sackman, Erikson, và Grant, 1968]. Thật dễ dàng để cố gắng phủ nhận kết quả của họ bằng cách nói rằng thử nghiệm- các lập trình viên ghen tị luôn làm tốt hơn những người mới bắt đầu, nhưng Sackman và các đồng nghiệp của ông đã so sánh các cặp lập trình viên phù hợp. Ví dụ, họ quan sát thấy hai lập trình viên với 10 năm kinh nghiệm về các loại dự án tương tự và đo thời gian chúng thực hiện các tác vụ như mã hóa và gỡ lỗi. Sau đó, họ quan sát, chẳng hạn, hai người mới bắt đầu đã tham gia nghề nghiệp trong cùng một khoảng thời gian ngắn và có trình độ học vấn tương tự. Com- phân tích hiệu suất kém nhất và tốt nhất, họ quan sát thấy sự khác biệt từ 6 đến 1 về kích thước sản phẩm, 8 đến 1 trong thời gian thực thi sản phẩm, 9 đến 1 trong thời gian phát triển, 18 đến 1 trong thời gian mã hóa và 28 đến 1 trong thời gian gỡ lỗi. Một quan sát đặc biệt đáng báo động là màn trình diễn tốt nhất và tồi tệ nhất trên một sản phẩm của hai lập trình viên, mỗi người đã có 11 năm kinh nghiệm. Cũng khi các trường hợp tốt nhất và xấu nhất được loại bỏ khỏi mẫu của Sackman và cộng sự, quan sát thấy có sự khác biệt- Các cuộc cạnh tranh vẫn theo thứ tự từ 5 đến 1. Trên cơ sở những kết quả này, rõ ràng, chúng ta không thể hy vọng để ước tính chi phí hoặc thời lượng phần mềm với bất kỳ mức độ chính xác nào (trừ khi chúng tôi đã nêu chi tiết thông tin liên quan đến tất cả các kỹ năng của tất cả nhân viên, điều này sẽ là bất thường nhất). Nó đã được lập luận rằng, trong một dự án lớn, sự khác biệt giữa các cá nhân có xu hướng triệt tiêu, nhưng đây có lẽ là mơ tưởng; sự hiện diện của một hoặc hai đội rất tốt (hoặc rất tệ) các thành viên có thể gây ra sự sai lệch rõ rệt so với lịch trình và ảnh hưởng đáng kể đến ngân sách.

Một yếu tố con người khác có thể ảnh hưởng đến ước tính là ở một quốc gia tự do, không có cách nào đảm bảo rằng một nhân viên quan trọng sẽ không từ chức trong suốt dự án. Thời gian và tiền bạc sau đó được dành để cố gắng lấp đầy vị trí còn trống và tích hợp người thay thế vào nhóm, hoặc trong việc tổ chức lại các thành viên còn lại trong nhóm để bù đắp cho sự mất mát. Hoặc cách, lịch trình trượt và ước tính không ổn định.

Trang 5

Cơ bản của vấn đề ước tính chi phí là một vấn đề khác: Kích thước của một sản phẩm như thế nào để được đo lường?

9.2.1 Các thước đo về kích thước của một sản phẩm

Số liệu phổ biến nhất cho kích thước của sản phẩm là số dòng mã. Hai đơn vị thường được sử dụng: **dòng mã (LOC)** và **hàng nghìn hướng dẫn nguồn được chuyển giao- tions (KDSI)** . Nhiều vấn đề liên quan đến việc sử dụng các dòng mã [van der Poel và Schach, 1983].

- Tạo mã nguồn chỉ là một phần nhỏ trong tổng số nỗ lực phát triển phần mềm. Có vẻ hơi xa vời rằng thời gian cần thiết cho các yêu cầu, phân tích, quy trình thiết kế, thực hiện và kiểm tra công việc (bao gồm lập kế hoạch và tài liệu- hoạt động tation) chỉ có thể được biểu thị dưới dạng một hàm của số dòng mã trong sản phẩm cuối cùng.
- Việc triển khai cùng một sản phẩm bằng hai ngôn ngữ khác nhau dẫn đến các phiên bản có sự khác biệt không có số dòng mã. Ngoài ra, với các ngôn ngữ như Lisp hoặc với nhiều ngôn ngữ không thủ tục 4GLs (Phần 15.2), khái niệm về một dòng mã không được định nghĩa.
- Thường không rõ chính xác cách đếm các dòng mã. Chỉ nên thực thi các dòng mã được đếm hoặc định nghĩa dữ liệu là tốt? Và những bình luận có nên được tính không? Nếu không, có một nguy cơ là các lập trình viên sẽ miễn cưỡng dành thời gian cho những gì họ nhận thấy là những nhận xét "không có lợi", nhưng nếu các nhận xét được tính, thì nguy cơ ngược lại

là các lập trình viên sẽ viết hàng loạt nhận xét để cố gắng tăng
năng suất. Ngoài ra, những gì về đếm các câu lệnh ngôn ngữ kiểm soát công việc? Một xác suất khác-
lem là cách tính các dòng đã thay đổi hoặc các dòng đã xóa — trong quá trình nâng cao sản phẩm-
uct để cải thiện hiệu suất của nó, đôi khi số lượng dòng mã bị giảm.
Việc sử dụng lại mã (Phần 8.1) cũng làm phức tạp việc đếm dòng: Nếu mã được sử dụng lại được sửa đổi,
nó được tính như thế nào? Và, điều gì sẽ xảy ra nếu mã được kế thừa từ một lớp cha (Phần 7.8)? Trong
ngắn gọn, số liệu rõ ràng đơn giản của các dòng mã là bất cứ điều gì ngoại trừ thẳng cho-
phường để đếm.

- Không phải tất cả các mã được triển khai đều được chuyển đến máy khách. Nó không phải là hiếm đối với một nửa mã bao gồm các công cụ cần thiết để hỗ trợ nỗ lực phát triển.
- Giả sử rằng một nhà phát triển phần mềm sử dụng trình tạo mã, chẳng hạn như trình tạo báo cáo, trình tạo màn hình hoặc trình tạo giao diện người dùng đồ họa (GUI). Sau một vài phút của hoạt động thiết kế từ phía nhà phát triển, công cụ này có thể tạo ra hàng nghìn dòng mã.
- Số lượng dòng mã trong sản phẩm cuối cùng chỉ có thể được xác định khi sản phẩm đã hoàn thành. Do đó, ước tính chi phí dựa trên các dòng mã là nguy hiểm gấp đôi. Để bắt đầu quá trình ước tính, số dòng mã trong thành phẩm phải được ước tính. Sau đó, ước tính này được sử dụng để ước tính chi phí sản phẩm. Không chỉ có sự không chắc chắn trong mọi kỹ thuật chi phí, mà nếu đầu vào cho bản thân công cụ ước tính chi phí không chắc chắn cũng không chắc chắn (nghĩa là số dòng mã trong sản phẩm chưa được chế tạo), thì độ tin cậy của ước tính chi phí kết quả là không chắc là rất cao.

Bởi vì số lượng dòng mã không đáng tin cậy, các chỉ số khác phải được xem xét.
Một cách tiếp cận thay thế để ước tính kích thước của sản phẩm là sử dụng các chỉ số dựa trên

Trang 6

các đại lượng đo lường có thể được xác định sớm trong quá trình phần mềm. Ví dụ,
van der Poel và Schach [1983] đã đưa ra **số liệu FFP** để ước tính chi phí của trung bình
quy mô sản phẩm xử lý dữ liệu. Ba yếu tố cấu trúc cơ bản của xử lý dữ liệu
sản phẩm là các tệp, luồng và quy trình của nó; tên FFP là một từ viết tắt được hình thành từ
các chữ cái đầu của các phần tử đó. Một *tệp* được định nghĩa là một tập hợp các
hồ sơ liên quan thường trú trong sản phẩm; giao dịch và các tệp tạm thời là
bị loại trừ. Một *dòng chảy* là một giao diện dữ liệu giữa các sản phẩm và môi trường, chẳng hạn như một
màn hình hoặc một báo cáo. Một *quá trình* là một thao tác logic hoặc số học chức năng xác định
dữ liệu; các ví dụ bao gồm sắp xếp, xác thực hoặc cập nhật. Với số lượng tệp *Fi* , luồng
Fl , và xử lý *Pr* trong một sản phẩm, kích thước *S* và chi phí *C của* nó được đưa ra bởi

$$S \quad Fi \quad Fl \quad Pr \qquad (9,1)$$

$$C\,d \qquad S \qquad (9.2)$$

trong đó *d* là hằng số thay đổi giữa các tổ chức. Hằng số *d* là một
chắc chắn về **hiệu quả** (**năng suất**) của quá trình phát triển phần mềm trong đó
cơ quan. Kích thước của một sản phẩm đơn giản là tổng số tệp, luồng và
quy trình, một số lượng có thể được xác định sau khi thiết kế kiến trúc hoàn thành. Các
khi đó chi phí tỷ lệ với quy mô, hằng số tỷ lệ *d* được xác định bởi
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-
sự. Không giống như các chỉ số dựa trên số dòng mã, chi phí có thể được ước tính trước
mã hóa bắt đầu.

Tính hợp lệ và độ tin cậy của chỉ số FFP đã được chứng minh bằng cách sử dụng
mẫu bao gồm một loạt các ứng dụng xử lý dữ liệu quy mô trung bình. Không may,
chỉ số không bao giờ được mở rộng để bao gồm cơ sở dữ liệu, một thành phần thiết yếu của nhiều dữ liệu-
sản phẩm chế biến.

Một số liệu tương tự, nhưng được phát triển độc lập, cho kích thước của sản phẩm đã được phát triển
của Albrecht [1979] dựa trên các điểm chức năng; Số liệu của Albrecht dựa trên số lượng
các mục đầu vào *Inp* , các mục đầu ra *Out* , các yêu cầu *Inq* , các tệp chính *Maf* , và các giao diện *Inf* . Trong nó
dạng đơn giản nhất, số **điểm chức năng** *FP* được cho bởi phương trình

$$FP = 4\,inch + 5\,Hết + 4\,Inq + 10\,Maf + 7\,Inf \quad (9,3)$$

Bởi vì đây là thước đo kích thước của sản phẩm, nó có thể được sử dụng để ước tính chi phí và ước tính năng suất.

Công thức (9.3) là đơn giản hóa của phép tính ba bước. Đầu tiên, không điều chỉnh các điểm chức năng được tính:

1. Mỗi thành phần của sản *phẩm— Inp* , *Out* , *Inq* , *Maf* và *Inf*— phải được phân loại như đơn giản, trung bình hoặc phức tạp (xem Hình 9.3).
2. Mỗi thành phần được gán một số điểm chức năng tùy thuộc vào cấp độ của nó. Đối với ví dụ, một đầu vào trung bình được gán bốn điểm chức năng, như được phản ánh trong phương trình (9.3), nhưng một đầu vào đơn giản chỉ được gán ba, trong khi một đầu vào phức tạp được gán sáu hàm-tion điểm. Dữ liệu cần thiết cho bước này xuất hiện trong Hình 9.3.
3. Các điểm chức năng được gán cho mỗi thành phần sau đó được tổng hợp lại, mang lại kết **quả không điểm chức năng vừa đủ (*UFP*)** .

Trang 7

HÌNH 9.3 Bảng của điểm chức năng các giá trị.

	Mức độ phức tạp		
Thành phần	**Đơn giản**	**Trung bình cộng**	**Phức tạp**
Mục đầu vào	3	4	6
Mục đầu ra	4	5	7
Yêu cầu	3	4	6
Tệp chính	7	10	15
Giao diện	5	7	10

HÌNH 9.4 Kỹ thuật các yếu tố cho điểm chức năng tính toán.

1. Truyền thông dữ liệu
2. Xử lý dữ liệu phân tán
3. Tiêu chí hoạt động
4. Phần cứng được sử dụng nhiều
5. Tỷ lệ giao dịch cao
6. Nhập dữ liệu trực tuyến
7. Hiệu quả của người dùng cuối
8. Cập nhật trực tuyến
9. Các phép tính phức tạp
10. Khả năng tái sử dụng
11. Dễ dàng cài đặt
12. Dễ dàng hoạt động
13. Tính di động
14. Khả năng bảo trì

Thứ hai, hệ số **phức tạp kỹ thuật (*TCF*)** được tính toán. 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; tập hợp đầy đủ các yếu tố là được thể hiện trong Hình 9.4. Mỗi yếu tố trong số 14 yếu tố này được gán một giá trị từ 0 (“không hiện tại hoặc không ảnh hưởng ”) đến 5 (“ ảnh hưởng mạnh mẽ xuyên suốt ”). 14 số kết quả được tính tổng, mang lại tổng mức độ ảnh hưởng (*DI*). Các *TCF* sau đó được đưa ra bởi

$$TCF = 0,65 + 0,01 * DI \quad (9,4)$$

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

Thứ ba, *FP* , số điểm chức năng, được cung cấp bởi

FP *UFP* *TCF* **(9,5)**

Các thử nghiệm để đo lường tỷ lệ năng suất phần mềm đã cho thấy sự phù hợp hơn khi sử dụng func-
điểm tốt hơn so với việc sử dụng KDSI. Ví dụ, Jones [1987] đã tuyên bố rằng ông đã quan sát thấy các lỗi

Trang 8

HÌNH 9.5
Một sự so sanh của nhà lắp ráp và Sản phẩm Ada [Jones, 1987]. (© 1987 IEEE.)

	Phiên bản Assembler	Phiên bản Ada
Kích thước mã nguồn	70 KDSI	25 KDSI
Chi phí phát triển	$ 1,043,000	$ 590,000
KDSI mỗi người-tháng	0,335	0,211
Báo cáo chi phí mỗi nguồn	$ 14,90	$ 23,60
Điểm chức năng mỗi người-tháng	1,65	2,92
Chi phí cho mỗi điểm chức năng	$ 3.023	$ 1.170

vượt quá 800 phần trăm đếm KDSI, nhưng *chỉ* [nhấn mạnh thêm] 200 phần trăm đếm-
điểm chức năng ing, một nhận xét tiết lộ nhất.

Để cho thấy sự vượt trội của các điểm chức năng so với các dòng mã, Jones [1987] trích dẫn
ví dụ trong hình 9.5. Cùng một sản phẩm được mã hóa cả trong trình lắp ráp và Ada
và kết quả được so sánh. Đầu tiên, hãy xem xét KDSI mỗi người-tháng. Chỉ số này cho chúng ta biết rằng
mã hóa trong trình hợp dịch rõ ràng là hiệu quả hơn 60% so với mã hóa trong Ada,
là sai lầm đáng kể. Các ngôn ngữ thế hệ thứ ba như Ada đã thay thế trình hợp dịch một cách đơn giản
vì nó hiệu quả hơn nhiều khi viết mã bằng ngôn ngữ thế hệ thứ ba. Bây giờ hãy xem xét
chỉ số thứ hai, chi phí cho mỗi báo cáo nguồn. Lưu ý rằng một tuyên bố Ada trong sản phẩm này là
tương đương với 2,8 câu lệnh hợp ngữ. Sử dụng báo cáo chi phí trên mỗi nguồn làm thước đo
hiệu quả một lần nữa ngụ ý rằng nó hiệu quả hơn để viết mã trong trình hợp dịch hơn là trong Ada. Tuy nhiên,
khi điểm chức năng mỗi người-tháng được coi là thước đo hiệu quả lập trình,
sự vượt trội của Ada so với trình lắp ráp được phản ánh rõ ràng.

Mặt khác, cả điểm chức năng và chỉ số FFP của các phương trình (9.1) và (9.2)
cùng một điểm yếu: Việc bảo trì sản phẩm thường được đo lường không chính xác. Khi nào
một sản phẩm được duy trì, có thể thực hiện các thay đổi lớn đối với sản phẩm mà không làm thay đổi
số lượng tệp, luồng và quy trình hoặc số lượng đầu vào, đầu ra, yêu cầu, bản gốc
tệp và giao diện. Các dòng mã không tốt hơn về mặt này. Đối với một trường hợp cực đoan, nó là
Có thể thay thế mọi dòng sản phẩm bằng một dòng hoàn toàn khác mà không cần thay đổi-
ing tổng số dòng mã.

Ít nhất 40 biến thể và phần mở rộng cho các điểm chức năng của Albrecht đã được đề xuất
[Maxwell và Forselius, 2000]. Các điểm chức năng Mk II được đưa ra bởi Symons [1991]
để cung cấp một cách chính xác hơn để tính toán các điểm chức năng chưa được điều chỉnh (*UFP*). Các
phần mềm được phân tách thành một tập hợp các giao dịch thành phần, mỗi giao dịch bao gồm một đầu vào,
một quá trình và một đầu ra. Giá trị của *UFP* sau đó được tính toán từ các đầu vào, quá trình này,
và kết quả đầu ra. Điểm chức năng Mk II được sử dụng rộng rãi trên toàn thế giới.

9.2.2 Các kỹ thuật ước tính chi phí

Bất chấp những khó khăn khi ước tính kích thước, điều cần thiết là các nhà phát triển phần mềm
chỉ cần làm những gì tốt nhất có thể để có được ước tính chính xác về cả thời gian và dự án
ect chi phí, trong khi tính đến càng nhiều càng tốt các yếu tố có thể ảnh hưởng đến
ước tính. Chúng bao gồm trình độ kỹ năng của nhân sự, mức độ phức tạp của dự án,
quy mô của dự án (chi phí tăng theo quy mô nhưng nhiều hơn so với tuyến tính), sự quen thuộc của
nhóm phát triển với khu vực ứng dụng, phần cứng mà sản phẩm sẽ trở thành

Trang 9

chạy và tính khả dụng của các công cụ CASE. Một yếu tố khác là hiệu ứng thời hạn. Nếu một dự án phải được hoàn thành vào một thời gian nhất định, nỗ lực tính bằng người-tháng lớn hơn nếu không có ràng buộc được đặt vào thời gian hoàn thành; do đó, chi phí càng lớn. Điều này cho thấy rằng thời lượng và chi phí không độc lập; thời hạn càng ngắn, nỗ lực càng lớn và do đó, càng lớn chi phí.

Từ danh sách trước, không có nghĩa là toàn diện, ước tính rõ ràng là vấn đề khó khăn. Một số cách tiếp cận đã được sử dụng, với thành công lớn hơn hoặc ít hơn.

1. Phán đoán chuyên môn bằng phép tương tự

Trong phần **đánh giá chuyên môn bằng** kỹ thuật loại suy, một số chuyên gia được tham khảo ý kiến. An chuyên gia đưa ra ước tính bằng cách so sánh sản phẩm mục tiêu với sản phẩm đã hoàn thành với mà chuyên gia đã tích cực tham gia và ghi nhận những điểm giống và khác nhau. Đối với ví dụ rất nhiều, 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 phải trực tuyến thu thập dữ liệu. Bởi vì tổ chức đã quen thuộc với loại sản phẩm sẽ được phát triển, chuyên gia giảm thời gian và nỗ lực phát triển xuống 15 phần trăm. Tuy nhiên, người dùng đồ họa giao diện hơi phức tạp; điều này làm tăng thời gian và nỗ lực lên 25 phần trăm. cuối cùng sản phẩm mục tiêu phải được phát triển bằng ngôn ngữ mà hầu hết các thành viên trong nhóm không quen thuộc, do đó tăng thời gian lên 15 phần trăm và nỗ lực lên 20 phần trăm. Kết hợp ba con số này, chuyên gia quyết định rằng sản phẩm mục tiêu sẽ có thêm 25% thời gian và nỗ lực nhiều hơn 30 phần trăm so với phần trước. Bởi vì sản phẩm trước đó đã 12 tháng để hoàn thành và yêu cầu 100 người-tháng, sản phẩm mục tiêu được ước tính là mất 15 tháng và tiêu thụ 130 người-tháng.

Hai chuyên gia khác trong tổ chức so sánh hai sản phẩm giống nhau. Một con-ám chỉ rằng sản phẩm mục tiêu sẽ mất 13,5 tháng và 140 tháng. Cai khac đưa ra các số liệu của 16 tháng và 95 người-tháng. Làm thế nào có thể dự đoán của ba chuyên gia này được hòa giải? Một kỹ thuật là kỹ thuật **Delphi** : Nó cho phép các chuyên gia để đi đến thống nhất mà không cần họp nhóm, điều này có thể không tác dụng phụ mong muốn của một thành viên thuyết phục làm lung lay nhóm. Trong kỹ thuật này, các chuyên gia làm việc độc lập. Mỗi ước tính 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 đang sản xuất 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 chấp nhận được. Không có cuộc họp nhóm nào diễn ra trong quá trình lặp lại quá trình.

Định giá bất động sản thường được thực hiện trên cơ sở đánh giá của chuyên gia bằng phép loại suy. An thẩm định viên đến định giá bằng cách so sánh một ngôi nhà với những ngôi nhà tương tự đã được bán gần đây. Giả sử rằng căn nhà A được định giá, căn nhà B bên cạnh vừa được bán với giá $ 205,000, và căn nhà C trên đường tiếp theo được bán cách đây 3 tháng với giá 218.000 đô la. Người thẩm định có thể suy luận như sau: Nhà A có nhiều phòng tắm hơn nhà B và sân rộng 5000 feet vuông lớn hơn. Nhà C có diện tích tương đương với nhà A, nhưng mái của nó kém. Mặt khác, nhà C có một bể sục. Sau khi suy nghĩ cẩn thận, người thẩm định có thể đi đến con số $ 215,000 cho ngôi nhà A.

Trong trường hợp sản phẩm phần mềm, đánh giá của chuyên gia bằng phép loại suy kém chính xác hơn so với thực tế định giá bất động sản. Nhớ lại rằng chuyên gia phần mềm đầu tiên của chúng tôi đã tuyên bố rằng việc sử dụng một lan-guage sẽ tăng thời gian lên 15 phần trăm và nỗ lực lên 20 phần trăm. Trừ khi chuyên gia có

Trang 10

một số dữ liệu đã được xác thực mà từ đó có thể xác định được ảnh hưởng của mỗi sự khác biệt (khả năng không thể xảy ra), lỗi gây ra bởi những gì có thể được mô tả chỉ là phỏng đoán sẽ dẫn đến trong ước tính chi phí không chính xác một cách vô vọng. Ngoài ra, trừ khi các chuyên gia may mắn với tổng số thu hồi (hoặc đã lưu giữ hồ sơ chi tiết), những hồi ức của họ về các sản phẩm đã hoàn thành có thể đủ không chính xác để làm mất hiệu lực dự đoán của họ. Cuối cùng, các chuyên gia là con người và, do đó, có những thành kiến có thể ảnh hưởng đến dự đoán của họ. Đồng thời, kết quả của ước tính của một nhóm chuyên gia nên phản ánh kinh nghiệm tập thể của họ; nếu điều này là rộng đủ, kết quả cũng có thể chính xác.

2. Phương pháp tiếp cận từ dưới lên

Một cách để cố gắng giảm thiểu các lỗi do đánh giá tổng thể một sản phẩm là phá vỡ sản phẩm thành các thành phần nhỏ hơn. Ước tính thời lượng và chi phí được thực hiện cho từng thành phần riêng biệt và kết hợp để cung cấp một con số tổng thể. Từ **dưới lên này phương pháp tiếp cận** có lợi thế là ước tính chi phí cho một số thành phần nhỏ hơn đồng minh nhanh hơn và chính xác hơn so với một đồng minh lớn. Ngoài ra, quá trình ước tính có thể sẽ chi tiết hơn so với một sản phẩm lớn, nguyên khối. Điểm yếu của điều này phương pháp tiếp cận 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 cách tiếp cận từ dưới lên. Tuy nhiên, tương tác giữa các đối tượng khác nhau trong sản phẩm trình bày quá trình ước tính.

3. Mô hình ước tính chi phí theo thuật toán

Trong cách tiếp cận này, một số liệu, chẳng hạn như các điểm chức năng hoặc số liệu FFP, được sử dụng làm đầu vào cho một mô hình xác định giá thành sản phẩm. 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. Bề ngoài, một **mô hình ước tính chi phí theo thuật toán** vượt trội hơn so với ý kiến của chuyên gia, bởi vì con người chuyên gia, như đã chỉ ra trước đây, thường có thành kiến và có thể bỏ qua một số khía cạnh của cả sản phẩm đã hoàn thành và 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. Điều nguy hiểm với một mô hình như vậy là ước tính của nó chỉ tốt như các giả định cơ bản. Ví dụ, cơ bản mô hình điểm chức năng là giả định rằng mọi khía cạnh của sản phẩm đều được thể hiện trong năm đại lượng ở bên phải của phương trình (9.3) và 14 hệ số kỹ thuật. Một vấn đề nữa là thường cần một lượng đáng kể đánh giá chủ quan trong quyết định những giá trị nào cần gán cho các tham số của mô hình. Ví dụ, thường xuyên nó không rõ liệu một yếu tố kỹ thuật cụ thể của mô hình điểm chức năng có nên được đánh giá hay không a 3 hoặc a 4.

Nhiều mô hình ước tính chi phí theo thuật toán đã được đề xuất. Một số dựa trên toán học- các lý thuyết biểu tượng về cách phần mềm được phát triển. Các mô hình khác dựa trên thống kê; số lượng lớn các dự án được nghiên cứu và xác định các quy tắc thực nghiệm từ dữ liệu. Hỗn hợp mô hình kết hợp các phương trình toán học, mô hình thống kê và đánh giá của chuyên gia. Mô hình lai quan trọng nhất là COCOMO của Boehm, được mô tả chi tiết trong Mục 9.2.3. (Xem Chỉ trong trường hợp bạn muốn biết Hộp 9.1 để thảo luận về từ viết tắt COCOMO.)

Trang 11

Chỉ trong trường hợp bạn muốn biết Hộp 9.1

COCOMO là một từ viết tắt được hình thành từ hai chữ cái đầu tiên của mỗi từ trong COnstructive COst MOdel. Bất kỳ mối liên hệ nào với Kokomo, Indiana, hoàn toàn là ngẫu nhiên.

Các *MO* trong COCOMO là viết tắt của "mô hình", vì vậy cụm từ *mô hình COCOMO* không nên đã sử dụng. Cụm từ đó thuộc cùng danh mục với "máy ATM" và "số PIN", cả hai trong số đó đã được Sở Thông tin Dự phòng cấp báo mộng.

9.2.3 COCOMO trung gian

COCOMO thực sự là một loạt ba mô hình, từ một mô hình macroestimation mà coi toàn bộ sản phẩm thành một mô hình vi vết bệnh xử lý sản phẩm một cách chi tiết. Trong phần này, mô tả được đưa ra về COCOMO trung gian, có cấp độ trung bình là phức tạp và chi tiết. COCOMO được mô tả chi tiết trong [Boehm, 1981]; tổng quan là trình bày trong [Boehm, 1984].

Thời gian phát triển máy tính sử dụng COCOMO trung gian được thực hiện trong hai giai đoạn. Đầu tiên, một ước tính sơ bộ về nỗ lực phát triển được cung cấp. Hai tham số phải là giao phối: chiều dài của sản phẩm trong KDSI và chế độ phát triển của sản phẩm, một thước đo về mức độ khó 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).

Từ hai tham số này, có thể tính được **nỗ lực danh nghĩa** . Ví dụ, nếu dự án được đánh giá là đơn giản về cơ bản (hữu cơ), sau đó là nỗ lực danh nghĩa (trong người-tháng) được cho bởi phương trình

$$\text{Nỗ lực danh nghĩa } 3{,}2\ (\text{KDSI})^{1{,}05} \text{ người-tháng} \qquad (9{,}6)$$

Các hằng số 3.2 và 1.05 là các giá trị phù hợp nhất với dữ liệu ở chế độ hữu cơ sản phẩm được Boehm sử dụng để phát triển COCOMO trung gian.

Ví dụ: nếu sản phẩm được xây dựng là sản phẩm hữu cơ và ước tính khoảng 12.000 người được phân phối báo cáo nguồn (12 KDSI), thì nỗ lực danh nghĩa là

$$3{,}2\ (12)^{1{,}05} \qquad 43 \text{ người-tháng}$$

(nhưng hãy đọc Chỉ trong trường hợp bạn muốn biết Hộp 9.2 để biết nhận xét về giá trị này).

Tiếp theo, giá trị danh nghĩa này phải được nhân với 15 **nỗ lực phát triển phần mềm số nhân** . Các số nhân này và giá trị của chúng được cho trong Hình 9.6. Mỗi hệ số có thể có tối đa sáu giá trị. Ví dụ: hệ số phức tạp của sản phẩm được gán cho các giá trị 0,70, 0,85, 1,00, 1,15, 1,30 hoặc 1,65, tùy theo liệu các nhà phát triển đánh giá độ phức tạp của dự án như rất thấp, thấp, danh nghĩa (trung bình), cao, rất cao hoặc cực cao. Như có thể thấy từ Hình 9.6, tất cả 15 số nhân đều nhận giá trị 1,00 khi tương ứng-tham số ing là danh nghĩa.

Boehm cung cấp các nguyên tắc để giúp nhà phát triển xác định xem thông số thực sự nên được đánh giá danh nghĩa hoặc cho dù đánh giá thấp hơn hoặc cao hơn. Ví dụ, xem xét lại hệ số phức tạp của mô-đun. Nếu các hoạt động điều khiển của mô-đun về cơ bản bao gồm một chuỗi các cấu trúc của lập trình có cấu trúc (chẳng hạn như **if - then - else** , **do - while** , **case**), thì độ phức tạp được đánh giá là *rất thấp* . Nếu các toán tử này là lồng vào nhau, thì xếp hạng *thấp* . Việc thêm các bảng quyết định và điều khiển liên mô-đun làm tăng đánh giá đến *danh nghĩa* . Nếu các toán tử được lồng vào nhau nhiều, với các vị từ ghép và hàng đợi và ngăn xếp, khi đó xếp hạng *cao*. Sự hiện diện của mã hóa quay lại và đệ quy và

Chỉ trong trường hợp bạn muốn biết Hộp 9.2

Một phản ứng đối với giá trị của nỗ lực danh nghĩa có thể là, "Nếu nỗ lực 43 tháng của người cần thiết để tạo ra 12.000 hướng dẫn nguồn được phân phối, sau đó trung bình mỗi chương trình-mer đang tạo ra ít hơn 300 dòng mã mỗi tháng — tôi đã triển khai nhiều hơn điều đó trong một đêm! ”

Một sản phẩm 300 dòng thường chỉ là: 300 dòng mã. Ngược lại, một Sản phẩm 12.000 dòng phải trải qua tất cả các quy trình làm việc của vòng đời. Nói cách khác, tổng nỗ lực của 43 tháng người được chia cho nhiều hoạt động, bao gồm cả viết mã.

HÌNH 9.6 Hệ số nhân nỗ lực phát triển phần mềm COCOMO trung gian [Boehm, 1984]. (© 1984 IEEE)

	Xếp hạng					
Trình điều khiển chi phí	**Rất thấp**	**Thấp**	**Trên danh nghĩa**	**Cao**	**Rất cao, cực cao**	
Thuộc tính sản phẩm						
Độ tin cậy cần thiết của phần mềm	0,75	0,88	1,00	1,15	1,40	
Kích thước cơ sở dữ liệu		0,94	1,00	1,08	1.16	
Độ phức tạp của sản phẩm	0,70	0,85	1,00	1,15	1,30	1,65
Thuộc tính máy tính						
Giới hạn thời gian thực hiện			1,00	1.11	1,30	1,66

Hạn chế lưu trữ chính			1,00	1,06	1,21	1.56
Sự biến động của máy ảo *		0,87	1,00	1,15	1,30	
Thời gian quay vòng máy tính		0,87	1,00	1,07	1,15	
Thuộc tính nhân sự						
Năng lực của nhà phân tích	1,46	1.19	1,00	0,86	0,71	
Ứng dụng trải nghiệm	1,29	1.13	1,00	0,91	0,82	
Khả năng lập trình viên	1,42	1.17	1,00	0,86	0,70	
Trải nghiệm máy ảo *	1,21	1.10	1,00	0,90		
Kinh nghiệm ngôn ngữ lập trình	1,14	1,07	1,00	0,95		
Thuộc tính dự án						
Sử dụng các phương pháp lập trình hiện đại	1,24	1.10	1,00	0,91	0,82	
Sử dụng các công cụ phần mềm	1,24	1.10	1,00	0,91	0,83	
Lịch trình phát triển bắt buộc	1,23	1,08	1,00	1,04	1.10	

* Đối với một sản phẩm phần mềm nhất định, máy ảo bên dưới là phức hợp của phần cứng và phần mềm (hệ điều hành, cơ sở dữ liệu hệ thống quản lý) nó kêu gọi hoàn thành nhiệm vụ của mình.

xử lý ngắt ưu tiên cố định đẩy xếp hạng lên *rất cao* . Cuối cùng, nhiều tài nguyên lập lịch với các mức độ ưu tiên thay đổi động và kiểm soát mức vi mã đảm bảo rằng đánh giá là *rất cao*. Các xếp hạng này áp dụng cho các hoạt động kiểm soát. Một mô-đun cũng phải được đánh giá từ quan điểm của các hoạt động tính toán, các hoạt động phụ thuộc vào thiết bị, và các hoạt động quản lý dữ liệu. Để biết chi tiết về tiêu chí tính toán từng 15 số nhân, tham khảo [Boehm, 1981].

Để xem cách này hoạt động như thế nào, Boehm [1984] đưa ra ví dụ về com- dựa trên bộ vi xử lý phần mềm xử lý thông tin liên lạc cho một mạng lưới chuyển tiền điện tử mới có độ tin cậy cao- làm việc, với hiệu suất, lịch trình phát triển và các yêu cầu về giao diện. Sản phẩm này- uct phù hợp với mô tả của chế độ nhúng và ước tính có 10.000 nguồn được phân phối hướng dẫn (10 KDSI) dài, vì vậy nỗ lực phát triển danh nghĩa được đưa ra bởi

$$\text{Nỗ lực danh nghĩa } 2{,}8\ (\text{KDSI})^{1{,}20} \qquad \textbf{(9,7)}$$

Trang 13

HÌNH 9.7 Trung gian COCOMO số nhân nỗ lực xếp hạng cho bộ vi xử lý thông tin liên lạc phần mềm [Boehm, 1984]. (© 1984 IEEE)

Trình điều khiển chi phí	Tình hình	Xếp hạng	Cố gắng Hệ số nhân
Độ tin cậy cần thiết của phần mềm	Hậu quả tài chính nghiêm trọng lỗi phần mềm	Cao	1,15
Kích thước cơ sở dữ liệu	20.000 byte	Thấp	0,94
Độ phức tạp của sản phẩm	Xử lý thông tin liên lạc	Rất cao	1,30
Giới hạn thời gian thực hiện	Sẽ sử dụng 70% thời gian có sẵn	Cao	1.11
Hạn chế lưu trữ chính	45 nghìn trên 64 nghìn cửa hàng (70%)	Cao	1,06
Máy ảo biến động	Dựa trên thương mại phần cứng vi xử lý	Trên danh nghĩa	1,00
Thời gian quay vòng máy tính	Thời gian quay vòng trung bình 2 giờ	Danh nghĩa	1,00
Năng lực của nhà phân tích	Các nhà phân tích cao cấp giỏi	Cao	0,86
Ứng dụng trải nghiệm	3 năm	Trên danh nghĩa	1,00
Khả năng lập trình viên	Lập trình viên cao cấp giỏi	Cao	0,86
Trải nghiệm máy ảo	6 tháng	Thấp	1.10
Kinh nghiệm ngôn ngữ lập trình	12 tháng	Trên danh nghĩa	1,00
Sử dụng chương trình hiện đại thực hành	Hầu hết các kỹ thuật được sử dụng qua 1 năm	Cao	0,91
Sử dụng các công cụ phần mềm	Ở máy tính mini cơ bản mức công cụ	Thấp	1.10
Lịch trình phát triển bắt buộc	9 tháng	Trên danh nghĩa	1,00

(Một lần nữa, các hằng số 2.8 và 1.20 là các giá trị phù hợp nhất với dữ liệu được nhúng trên sản phẩm.) Bởi vì dự án được ước tính dài 10 KDSI, nỗ lực danh nghĩa là

$$2{,}8\ (10)^{1{,}20} \qquad 44 \text{ người-tháng}$$

Nỗ lực phát triển ước tính có được bằng cách nhân nỗ lực danh nghĩa với
15 nhân nỗ lực phát triển phần mềm. Xếp hạng của các hệ số này và giá trị của chúng
ues được cho trong Hình 9.7. Sử dụng các giá trị này, tích của các số nhân được tìm thấy là
1,35, do đó, nỗ lực ước tính cho dự án là

1,35 44 59 người-tháng

Con số này sau đó được sử dụng trong các công thức bổ sung để xác định chi phí đô la, phát triển
lịch trình, phân phối giai đoạn và hoạt động, chi phí máy tính, chi phí bảo trì hàng năm và
các hạng mục liên quan khác; để biết thêm chi tiết, xem [Boehm, 1981]. COCOMO trung gian đã hoàn thành
mô hình ước tính chi phí theo thuật toán, cung cấp cho người dùng hầu như mọi hỗ trợ có thể hình dung được trong
lập kế hoạch dự án.

COCOMO trung gian đã được xác nhận đối với một mẫu rộng gồm 63 dự án
bao gồm nhiều lĩnh vực ứng dụng khác nhau. Kết quả của việc áp dụng trung gian
COCOMO đối với mẫu này là các giá trị thực tế nằm trong khoảng 20% so với dự đoán
giá trị khoảng 68 phần trăm thời gian. Những nỗ lực để cải thiện độ chính xác này không có ý nghĩa gì
bởi vì trong hầu hết các tổ chức, dữ liệu đầu vào cho COCOMO trung gian nói chung là
chính xác chỉ trong khoảng 20 phần trăm. Tuy nhiên, độ chính xác thu được bằng cách trải nghiệm
các nhà ước tính có kinh nghiệm đã đặt COCOMO trung gian ở vị trí tiên tiến của ước tính chi phí
nghiên cứu trong những năm 1980; không có kỹ thuật nào khác chính xác nhất quán.

Trang 14

Vấn đề chính đối với COCOMO trung gian là đầu vào quan trọng nhất của nó là
số dòng mã trong sản phẩm mục tiêu. Nếu ước tính này không chính xác, thì mọi
dự đoán của mô hình có thể không chính xác. Bởi vì khả năng rằng những dự đoán của
COCOMO trung gian hoặc bất kỳ kỹ thuật ước tính nào khác có thể không chính xác, quản lý
phải giám sát tất cả các dự đoán trong suốt quá trình phát triển phần mềm.

9.2.4 COCOMO II

COCOMO được đưa ra vào năm 1981. Vào thời điểm đó, mô hình vòng đời duy nhất được sử dụng là
Mô hình thác nước. Hầu hết phần mềm được chạy trên máy tính lớn. Các công nghệ như máy khách – máy chủ
và hướng đối tượng về cơ bản là không xác định. Theo đó, COCOMO đã không
thăm dò bất kỳ yếu tố nào trong số này. Tuy nhiên, khi các công nghệ mới hơn bắt đầu được chấp nhận
thực hành kỹ thuật phần mềm, COCOMO bắt đầu trở nên kém chính xác hơn.

COCOMO II [Boehm và cộng sự, 2000] 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 được mô tả trong Chương 2, tạo mẫu nhanh
(Phần 11.13), ngôn ngữ thế hệ thứ tư (Phần 15.2), sử dụng lại (Phần 8.1) và COTS
phần mềm (Mục 1.11). COCOMO II vừa linh hoạt vừa tinh vi. Thật không may, để
đạt được mục tiêu này, COCOMO II phức tạp hơn đáng kể so với COCOMO ban đầu.
Do đó, độc giả muốn sử dụng COCOMO II nên nghiên cứu [Boehm et al., 2000]
chi tiết; chỉ tổng quan về sự khác biệt chính giữa COCOMO II và trung gian
COCOMO được đưa ra ở đây.

Đầu tiên, COCOMO trung gian bao gồm một mô hình tổng thể dựa trên các dòng mã
(KDSI). Mặt khác, COCOMO II bao gồm ba mô hình khác nhau. các **ứng dụng**
mô hình thành phần , dựa trên các điểm đối tượng (tương tự như các điểm chức năng), được áp dụng tại
quy trình công việc sớm nhất, khi có sẵn kiến thức tối thiểu về sản phẩm sẽ được chế tạo.
Sau đó, khi có nhiều kiến thức hơn, **mô hình thiết kế ban đầu** được sử dụng; mô hình này là
dựa trên điểm chức năng. Cuối cùng, khi các nhà phát triển có thông tin tối đa, các **hậu**
mô hình kiến trúc được sử dụng. Mô hình này sử dụng các điểm chức năng hoặc các dòng mã (KDSI). Các
đầu ra từ COCOMO trung gian là ước tính chi phí và quy mô; đầu ra từ mỗi
ba mô hình của COCOMO II là một loạt các ước tính chi phí và kích thước. Theo đó, nếu nhất
ước tính khả năng của nỗ lực là E , sau đó mô hình thành phần ứng dụng trả về phạm vi
(0,50 E , 2,0 E) và mô hình hậu kiến trúc trả về phạm vi (0,80 E , 1,25 E). Điều này phản ánh
độ chính xác ngày càng tăng của sự tiến triển của các mô hình COCOMO II.

Điểm khác biệt thứ hai nằm ở mô hình nỗ lực cơ bản của COCOMO:

$$\text{Nỗ lực } a \quad (\text{kích thước})^{b} \qquad \textbf{(9,8)}$$

trong đó a và b là hằng số. Trong COCOMO trung gian, số mũ b có ba dif-
giá trị ferent, tùy thuộc vào chế độ của sản phẩm được xây dựng có phải là hữu cơ hay không (b = 1,05),

semidetached (b = 1,12), hoặc nhúng (b = 1,20). Trong COCOMO II, giá trị của b thay đổi từ 1,01 đến 1,26, tùy thuộc vào nhiều thông số khác nhau của mô hình. Bao gồm các sự quen thuộc với các sản phẩm thuộc loại đó, mức độ thành thục của quy trình (Phần 3.13), mức độ rủi ro giải quyết (Phần 2.7) và mức độ hợp tác nhóm (Phần 4.1).

Sự khác biệt thứ ba là giả định liên quan đến việc tái sử dụng. COCOMO trung gian giả định rằng số tiền tiết kiệm được do tái sử dụng tỷ lệ thuận với số lượng tái sử dụng. COCOMO II có tính đến rằng những thay đổi nhỏ đối với phần mềm được sử dụng lại phải chịu mức độ lớn không tương xứng chi phí (bởi vì mã đã được hiểu chi tiết cho dù chỉ là một thay đổi nhỏ và chi phí của thử nghiệm một mô-đun đã sửa đổi là tương đối lớn).

Trang 15

Thứ tư, hiện có 17 trình điều khiển chi phí nhân, thay vì 15 trình điều khiển trung gian COCOMO. Bảy trong số các trình điều khiển chi phí là mới, chẳng hạn như khả năng tái sử dụng bắt buộc trong các sản phẩm trong tương lai, doanh thu nhân sự hàng năm và liệu sản phẩm có đang được phát triển tại nhiều địa điểm hay không.

COCOMO II đã được hiệu chỉnh bằng cách sử dụng 83 dự án từ nhiều lĩnh vực khác nhau. Mô hình này vẫn còn quá mới để có nhiều kết quả về độ chính xác của nó và, ngang bằng ticular, mức độ mà nó là một cải tiến so với người tiền nhiệm của nó, bản gốc (1981) COCOMO.

9.2.5 Thời lượng theo dõi và ước tính chi phí

Trong khi sản phẩm đang được phát triển, nỗ lực phát triển thực tế phải liên tục so với dự đoán. Ví dụ: giả sử rằng chỉ số ước tính được sử dụng bởi các nhà phát triển phần mềm đã dự đoán rằng **thời lượng** của quy trình phân tích sẽ kéo dài 3 tháng và cần 7 người-tháng nỗ lực. Tuy nhiên, đã 4 tháng trôi qua và 10 người-tháng nỗ lực đã được sử dụng, nhưng các thông số kỹ thuật không có nghĩa là hoàn thành. Những sai lệch kiểu này có thể coi là một cảnh báo sớm rằng có điều gì đó đã xảy ra sai và hành động sửa chữa phải được thực hiện. Vấn đề có thể là kích thước của sản phẩm- uct bị đánh giá thấp nghiêm trọng hoặc nhóm phát triển không đủ năng lực như ban đầu được cho là. Dù lý do là gì, sẽ có thời lượng nghiêm trọng và chi phí cao hơn- và ban quản lý phải thực hiện hành động thích hợp để giảm thiểu các tác động.

Theo dõi cẩn thận các dự đoán phải được thực hiện trong suốt quá trình phát triển, không thể phụ thuộc vào các kỹ thuật mà các dự đoán đã được thực hiện. Sự sai lệch có thể là do các chỉ số dự đoán kém, phát triển phần mềm không hiệu quả, kết hợp cả hai, hoặc một số lý do khác. Điều quan trọng là phát hiện sớm những sai lệch và có biện pháp xử lý ngay lập tức. hành động sửa chữa. Ngoài ra, điều cần thiết là phải liên tục cập nhật các dự đoán trong điều kiện thông tin bổ sung khi nó có sẵn.

Bây giờ các chỉ số để ước tính thời lượng và chi phí đã được thảo luận, các thành phần của kế hoạch quản lý dự án phần mềm được mô tả.

9.3 Các thành phần của Kế hoạch Quản lý Dự án Phần mềm

Kế hoạch quản lý dự án phần mềm có ba thành phần chính: công việc phải thực hiện, nguồn lực để thực hiện nó và số tiền để trả cho tất cả. Trong phần này, ba các thành phần của kế hoạch được thảo luận. Thuật ngữ này được lấy từ [IEEE 1058, 1998], được thảo luận chi tiết hơn trong Phần 9.4.

Phát triển phần mềm yêu cầu *tài nguyên* . Các **nguồn lực** chính cần có là con người ai sẽ phát triển phần mềm, phần cứng chạy phần mềm và hỗ trợ phần mềm như hệ điều hành, trình soạn thảo văn bản và phần mềm kiểm soát phiên bản (Phần 5.9).

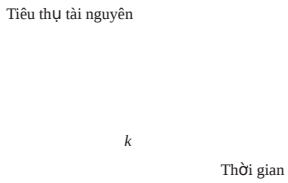
Việc sử dụng các nguồn lực như nhân sự thay đổi theo thời gian. Norden [1958] đã chỉ ra rằng đối với các dự án lớn, **phân phối Rayleigh** là một sự xấp xỉ tốt về cách tiêu thụ tài nguyên, R_c , thay đổi theo thời gian, t , nghĩa là,

$$R_c \quad \frac{t}{k_2} \, e^{t_2/2k_2} \qquad 0\ t \qquad \textbf{(9.9)}$$

Tham số k là hằng số, thời gian tiêu thụ ở mức cao nhất và e = 2.71828. . . , cơ số của logarit Naperian (tự nhiên). Một đường cong Rayleigh điển hình được thể hiện trong Hình 9.8.

Trang 16

HÌNH 9.8
Rayleigh
đường cong hiển thị
tài nguyên như thế nào
tiêu dùng
thay đổi theo thời gian.



Tiêu thụ tài nguyên bắt đầu nhỏ, tăng nhanh đến đỉnh và sau đó giảm chậm hơn tỷ lệ. Putnam [1978] đã nghiên cứu khả năng ứng dụng các kết quả của Norden vào việc phát triển phần mềm- và nhận thấy rằng nhân sự và tiêu thụ tài nguyên khác được mô hình hóa với một số mức độ chính xác của phân bố Rayleigh.

Do đó, không đủ trong một kế hoạch phần mềm chỉ đơn thuần nói rằng ba chương trình cấp cao- người có ít nhất 5 năm kinh nghiệm là bắt buộc. Những gì cần thiết là một cái gì đó như tiếp theo:

> Ba lập trình viên cao cấp có ít nhất 5 năm kinh nghiệm trong lập trình thời gian thực là cần thiết, hai để bắt đầu 3 tháng sau khi dự án bắt đầu, thứ ba để bắt đầu 6 tháng sau cái đó. Hai sẽ bị loại bỏ khi bắt đầu thử nghiệm sản phẩm, thứ ba khi giao hàng sau bắt đầu bảo trì.

Thực tế là nhu cầu nguồn lực phụ thuộc vào thời gian không chỉ áp dụng cho nhân sự mà còn đến thời gian sử dụng máy tính, phần mềm hỗ trợ, phần cứng máy tính, tiện nghi văn phòng và thậm chí là du lịch. Do đó, kế hoạch quản lý dự án phần mềm là một hàm của thời gian.

Công việc phải làm được chia thành hai loại. Đầu tiên là công việc tiếp tục trong suốt dự án và không liên quan đến bất kỳ quy trình phát triển phần mềm cụ thể nào. Công việc như vậy được gọi là một **chức năng của dự án** . Ví dụ như quản lý dự án và kiểm soát chất lượng. Thứ hai là công việc liên quan đến quy trình làm việc cụ thể trong quá trình phát triển sản phẩm; như là công việc được gọi là một *hoạt động* hoặc một *nhiệm vụ* . Một **hoạt động** là một đơn vị lớn công việc mà có chính xác ngày bắt đầu và ngày kết thúc; tiêu tốn tài nguyên, chẳng hạn như thời gian sử dụng máy tính hoặc số ngày của con người; và kết quả là **các sản phẩm công việc** , chẳng hạn như ngân sách, tài liệu thiết kế, lịch trình, nguồn mã hoặc hướng dẫn sử dụng. Ngược lại, 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. Do đó có ba loại công việc trong kế hoạch quản lý dự án phần mềm: các chức năng dự án được thực hiện trong suốt dự án, hoạt động (đơn vị công việc chính), và nhiệm vụ (đơn vị công việc nhỏ).

Trang 17

Một khía cạnh quan trọng của kế hoạch liên quan đến việc hoàn thành các sản phẩm công việc. Ngày mà một sản phẩm công việc được coi là hoàn thành được gọi là một **cột mốc quan trọng** . Để xác định xem một tác phẩm sản phẩm thực sự đã đạt đến một cột mốc quan trọng, trước tiên sản phẩm đó phải vượt qua một loạt các **bài đánh giá được** thực hiện bởi các thành viên trong nhóm, ban quản lý hoặc khách hàng. Một cột mốc tiêu biểu là ngày mà thiết kế được hoàn thành và vượt qua đánh giá. Sau khi một sản phẩm làm việc đã được xem xét và đã đồng ý, nó trở thành **đường cơ sở** và chỉ có thể được thay đổi thông qua các thủ tục chính thức, như được mô tả trong Phần 5.10.2.

Trên thực tế, có nhiều thứ hơn đối với một sản phẩm làm việc chứ không chỉ đơn thuần là sản phẩm đó. Một **gói công việc-tuổi tác** xác định không chỉ sản phẩm công việc mà còn xác định các yêu cầu về nhân sự, thời hạn, nguồn lực, tên của cá nhân chịu trách nhiệm và tiêu chí nghiệm thu đối với sản phẩm công việc. **Tiền** của khóa học là một thành phần quan trọng của kế hoạch. Một ngân sách chi tiết phải được tính toán và số tiền được phân bổ, 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.

9.4 Khung kế hoạch quản lý dự án phần mềm

Có nhiều cách để lập một kế hoạch quản lý dự án. Một trong những điều tốt nhất là IEEE Tiêu chuẩn 1058 [1998]. Các thành phần của kế hoạch được thể hiện trong Hình 9.9.

- Tiêu chuẩn được đưa ra bởi đại diện của nhiều tổ chức lớn tham gia phát triển phần mềm. Đầu vào đến từ cả ngành công nghiệp và toàn cầu- mối quan hệ, và các thành viên của nhóm làm việc và nhóm đánh giá đã có nhiều năm kinh nghiệm lập kế hoạch quản lý dự án. Tiêu chuẩn kết hợp điều này kinh nghiệm.
- Kế hoạch quản lý dự án IEEE được thiết kế để sử dụng với tất cả các loại sản phẩm phần mềm- uv. Nó không áp đặt một mô hình vòng đời cụ thể hoặc quy định một phương pháp luận cụ thể. Kế hoạch về cơ bản là một khuôn khổ, các nội dung trong đó được điều chỉnh bởi từng tổ chức- cho một miền, nhóm phát triển hoặc kỹ thuật cụ thể.
- Khung kế hoạch quản lý dự án IEEE hỗ trợ cải tiến quy trình. Đối với ví dụ, nhiều phần của khuôn khổ phản ánh các lĩnh vực quy trình chính của CMM (Phần 3.13) chẳng hạn như quản lý cấu hình và số liệu.
- Khung kế hoạch quản lý dự án IEEE là lý tưởng cho Quy trình hợp nhất. Đối với ví dụ, một phần của kế hoạch được dành cho việc kiểm soát các yêu cầu và một phần khác dành cho rủi ro quản lý, cả hai khía cạnh trung tâm của Quy trình hợp nhất.

Mặt khác, mặc dù tuyên bố được đưa ra trong Tiêu chuẩn IEEE 1058 [1998] rằng Kế hoạch quản lý dự án IEEE có thể áp dụng cho các dự án phần mềm thuộc mọi quy mô, một số các phần không liên quan đến phần mềm quy mô nhỏ. Ví dụ, phần 7.7 của kế hoạch khung có tiêu đề là "Kế hoạch quản lý nhà thầu phụ", nhưng nó chỉ là tất cả những chưa từng nghe đến nhà thầu phụ được sử dụng trong các dự án quy mô nhỏ.

Theo đó, bây giờ chúng tôi trình bày khung kế hoạch theo hai cách khác nhau. Đầu tiên, đầy đủ khung được mô tả trong Phần 9.5. Thứ hai, một phiên bản được viết tắt một chút của khung- công việc được sử dụng trong Phụ lục F cho một kế hoạch quản lý cho một dự án quy mô nhỏ, MSG Nghiên cứu điển hình nền tảng.

HÌNH 9,9 IEEE dự án sự quản lý

1. Tổng quan
   1.1 Tóm tắt dự án
      1.1.1 Mục đích, phạm vi và mục tiêu
      1.1.2 Các giả định và ràng buộc

khung kế hoạch.

Trang 19

9.5 Kế hoạch quản lý dự án phần mềm IEEE

Bản thân khung **kế hoạch quản lý dự án phần mềm IEEE** (SPMP) bây giờ được mô tả chi tiết. Các số và tiêu đề trong văn bản tương ứng với các mục trong Hình 9.9. Các các thuật ngữ khác nhau được sử dụng đã được định nghĩa trong Phần 9.3.

1. Tổng quan.

1.1 Tóm tắt dự án.

1.1.1 Mục đích, phạm vi và mục tiêu. Một mô tả ngắn gọn được đưa ra về mục đích và phạm vi của sản phẩm phần mềm sẽ được phân phối, cũng như các mục tiêu của dự án. Kinh doanh nhu cầu được bao gồm trong tiểu mục này.

1.1.2 Các giả định và ràng buộc. Bất kỳ giả định nào làm cơ sở cho dự án là được nêu ở đây, cùng với các ràng buộc, chẳng hạn như ngày giao hàng, ngân sách, tài nguyên và hiện vật được tái sử dụng.

1.1.3 Các sản phẩm của dự án. Tất cả các mặt hàng sẽ được giao cho khách hàng được liệt kê ở đây, cùng với ngày giao hàng.

1.1.4 Lịch trình và tóm tắt ngân sách. Lịch trình tổng thể được trình bày ở đây, cùng với ngân sách tổng thể.

1.2 Sự phát triển của kế hoạch quản lý dự án. Không có kế hoạch nào có thể được đúc bằng bê tông. Kế hoạch quản lý dự án, giống như bất kỳ kế hoạch nào khác, yêu cầu cập nhật liên tục theo trải nghiệm và thay đổi trong cả tổ chức khách hàng và phát triển phần mềm hoặc- ganization. Trong phần này, các thủ tục và cơ chế chính thức để thay đổi kế hoạch là được mô tả, bao gồm cả cơ chế để đặt bản thân kế hoạch quản lý dự án theo điều khiển hình tượng.

2 Tài liệu tham khảo. Tất cả các tài liệu được tham chiếu trong kế hoạch quản lý dự án được liệt kê ở đây.

3 Định nghĩa và từ viết tắt. Thông tin này đảm bảo rằng ban quản lý dự án kế hoạch sẽ được mọi người hiểu theo cùng một cách.

4 Tổ chức dự án.

4.1 Các giao diện bên ngoài. Không có dự án nào được xây dựng trong môi trường chân không. Các thành viên dự án phải tương tác với tổ chức khách hàng và các thành viên khác trong tổ chức của họ. Ngoài ra, các nhà thầu phụ có thể tham gia vào một dự án lớn. Hành chính và quản lý- ranh giới rial giữa dự án và các thực thể khác này phải được thiết lập.

4.2 Cấu trúc bên trong. Trong phần này, cấu trúc của tổ chức phát triển chính nó được mô tả. Ví dụ, nhiều tổ chức phát triển phần mềm được chia thành hai loại nhóm: nhóm phát triển làm việc trên một dự án và nhóm hỗ trợ cung cấp các chức năng hỗ trợ, chẳng hạn như quản lý cấu hình và đảm bảo chất lượng, trên cơ sở toàn tổ chức. Ranh giới hành chính và quản lý giữa nhóm dự án và các nhóm hỗ trợ cũng phải được xác định rõ ràng.

4.3 Vai trò và trách nhiệm. Đối với mỗi chức năng của dự án, chẳng hạn như đảm bảo chất lượng, và đối với mỗi hoạt động, chẳng hạn như thử nghiệm sản phẩm, cá nhân chịu trách nhiệm phải được xác định.

5 Kế hoạch quy trình quản lý.

5.1 Kế hoạch khởi nghiệp.

Trang 20

5.1.1 Kế hoạch ước tính. Các kỹ thuật được sử dụng để ước tính thời gian và chi phí của dự án là được liệt kê ở đây, cũng như cách các ước tính này được theo dõi và, nếu cần, được sửa đổi trong khi dự án đang được tiến hành.

5.1.2 Kế hoạch nhân sự. Số lượng và loại nhân sự cần thiết được liệt kê cùng nhau với thời lượng mà chúng cần thiết.

5.1.3 Kế hoạch mua lại nguồn lực. Cách đạt được các nguồn lực cần thiết, bao gồm phần cứng, phần mềm, hợp đồng dịch vụ và dịch vụ quản trị, được đưa ra ở đây.

5.1.4 Kế hoạch đào tạo nhân viên dự án. Tất cả các khóa đào tạo cần thiết để hoàn thành thành công dự án được liệt kê trong tiểu mục này.

5.2 Kế hoạch làm việc.

5.2.1 Hoạt động công việc. Trong tiểu mục này, các hoạt động công việc được chỉ định, cho đến mức độ nhiệm vụ nếu phù hợp.

5.2.2 Phân bổ lịch trình. Nói chung, các gói công việc phụ thuộc lẫn nhau và hơn nữa phụ thuộc vào các sự kiện bên ngoài. Ví dụ: quy trình triển khai sau quy trình thiết kế và thử nghiệm sản phẩm trước. Trong tiểu mục này, mô tả liên quan- dencies được chỉ định.

5.2.3 Phân bổ nguồn lực. Các tài nguyên khác nhau được liệt kê trước đây được phân bổ cho các chức năng, hoạt động và nhiệm vụ thích hợp của dự án.

5.2.4 Phân bổ ngân sách. Trong tiểu mục này, ngân sách tổng thể được chia nhỏ ở chức năng, hoạt động và mức nhiệm vụ của dự án.

5.3 Kế hoạch kiểm soát.

5.3.1 Kế hoạch kiểm soát các yêu cầu. Như được mô tả trong Phần B của cuốn sách này, trong khi sản phẩm phần mềm đang được phát triển, các yêu cầu thường xuyên thay đổi. Các cơ chế

được sử dụng để giám sát và kiểm soát các thay đổi đối với các yêu cầu được đưa ra trong phần này.

5.3.2 Kế hoạch kiểm soát lịch trình. Trong phần phụ này, các cơ chế để đo lường progress được liệt kê, cùng với mô tả về các hành động sẽ được thực hiện nếu tiến độ thực tế bị chậm chậm tiến độ kế hoạch.

5.3.3 Kế hoạch kiểm soát ngân sách. Điều quan trọng là chi tiêu không được vượt quá mức số tiền nhận được. Cơ chế kiểm soát để giám sát khi chi phí thực tế vượt quá chi phí ngân sách, cũng như các hành động cần thực hiện nếu điều này xảy ra, được mô tả trong tiểu mục này.

5.3.4 Kế hoạch kiểm soát chất lượng. Các cách thức mà chất lượng được đo lường và kiểm soát được mô tả trong tiểu mục này.

5.3.5 Kế hoạch báo cáo. Để giám sát các yêu cầu, lịch trình, ngân sách và chất lượng, cơ chế báo cáo cần phải có. Các cơ chế này được mô tả trong tiểu mục này.

5.3.6 Kế hoạch thu thập số liệu. Như đã giải thích trong Phần 5.5, không thể quản lý quá trình phát triển mà không đo lường các chỉ số liên quan. Các chỉ số được thu thập được liệt kê trong tiểu mục này.

5.4 Kế hoạch quản lý rủi ro. Các rủi ro phải được xác định, ưu tiên, giảm thiểu và được theo dõi. Tất cả các khía cạnh của quản lý rủi ro được mô tả trong phần này.

5.5 Kế hoạch kết thúc dự án. Các hành động cần thực hiện sau khi dự án hoàn thành, bao gồm phân công lại nhân viên và lưu trữ hiện vật, được trình bày ở đây.

6 Kế hoạch quy trình kỹ thuật.

6.1 Mô hình quy trình. Trong phần này, mô tả chi tiết được đưa ra về vòng đời mô hình sẽ được sử dụng.

6.2 Phương pháp, công cụ và kỹ thuật. Các phương pháp luận phát triển và chương trình các ngôn ngữ ghép hình sẽ được sử dụng được mô tả ở đây.

6.3 Kế hoạch cơ sở hạ tầng. Các khía cạnh kỹ thuật của phần cứng và phần mềm được mô tả trong chi tiết trong phần này. Các hạng mục cần được đề cập bao gồm hệ thống máy tính (cứng-thiết bị, hệ điều hành, mạng và phần mềm) được sử dụng để phát triển phần mềm sản phẩm, cũng như các hệ thống máy tính mục tiêu mà sản phẩm phần mềm sẽ được chạy trên đó và các công cụ CASE sẽ được sử dụng.

6.4 Kế hoạch nghiệm thu sản phẩm. Để đảm bảo rằng sản phẩm phần mềm đã hoàn thành sẽ vượt qua kiểm tra chấp nhận của nó, các tiêu chí chấp nhận phải được đưa ra, khách hàng phải đồng ý với các tiêu chí bằng văn bản, và các nhà phát triển sau đó phải đảm bảo rằng các tiêu chí này thực sự được đáp ứng. Theo cách đó ba giai đoạn này của quá trình chấp nhận sẽ được thực hiện được mô tả trong phần này.

7 Các kế hoạch quy trình hỗ trợ.

7.1 Kế hoạch quản lý cấu hình. Trong phần này, mô tả chi tiết là đưa ra các phương tiện mà tất cả các hiện vật được đặt dưới sự quản lý cấu hình.

7.2 Kế hoạch kiểm tra. Kiểm tra, giống như tất cả các khía cạnh khác của phát triển phần mềm, cần cẩn thận lập kế hoạch.

7.3 Kế hoạch tài liệu. Mô tả tài liệu các loại, có hoặc không được giao cho khách hàng khi kết thúc dự án, được bao gồm trong phần này.

7.4 Kế hoạch đảm bảo chất lượng. Tất cả các khía cạnh của đảm bảo chất lượng, bao gồm cả kiểm tra, tiêu chuẩn và đánh giá, được bao gồm trong phần này.

7.5 Kế hoạch đánh giá và đánh giá. Chi tiết về cách thức tiến hành đánh giá được trình bày trong phần này.

7.6 Kế hoạch giải quyết vấn đề. Trong quá trình phát triển một sản phẩm phần mềm, lems là tất cả nhưng chắc chắn sẽ phát sinh. Ví dụ, một bản đánh giá thiết kế có thể đưa ra mức độ quan trọng lỗi trong quy trình phân tích yêu cầu thay đổi lớn đối với hầu hết tất cả các cấu phần đã đã hoàn thành. Trong phần này, cách các vấn đề như vậy được xử lý được mô tả.

7.7 Kế hoạch quản lý nhà thầu phụ. Phần này có thể áp dụng khi con-máy kéo là để cung cấp các sản phẩm công việc nhất định. Cách tiếp cận để lựa chọn và quản lý các nhà thầu sau đó xuất hiện ở đây.

7.8 Kế hoạch cải tiến quy trình. Các chiến lược cải tiến quy trình được bao gồm trong

phần này.

8 Các kế hoạch bổ sung. Đối với một số dự án nhất định, các thành phần bổ sung có thể cần xuất hiện trong kế hoạch. Về khuôn khổ IEEE, chúng xuất hiện ở cuối kế hoạch. Bổ sung các thành phần có thể bao gồm kế hoạch bảo mật, kế hoạch an toàn, kế hoạch chuyển đổi dữ liệu, cài đặt kế hoạch và kế hoạch bảo trì sau phân phối của dự án phần mềm.

9.6 Lập kế hoạch Kiểm tra

Một thành phần của SPMP thường bị bỏ qua là **lập kế hoạch kiểm tra** . Như mọi người khác hoạt động phát triển phần mềm, kiểm thử phải được lập kế hoạch. SPMP phải bao gồm nguồn lực để kiểm tra và lịch trình chi tiết phải chỉ ra rõ ràng việc kiểm tra sẽ được thực hiện trong mỗi quy trình làm việc.

Nếu không có một kế hoạch thử nghiệm, một dự án có thể trở nên tồi tệ theo một số cách. Ví dụ, trong sản phẩm-kiểm tra uct (Mục 3.7.4), nhóm SQA phải kiểm tra xem mọi khía cạnh của đặc điểm kỹ thuật tài liệu, như đã được khách hàng ký tên, đã được triển khai trong sản phẩm hoàn chỉnh. A cách tốt để hỗ trợ nhóm SQA trong nhiệm vụ này là yêu cầu sự phát triển phải được theo dõi-có thể (Mục 3.7). Nghĩa là, phải có thể kết nối từng câu lệnh trong đặc tả tài liệu cho một phần của thiết kế và mỗi phần của thiết kế phải được phản ánh rõ ràng trong mật mã. Một kỹ thuật để đạt được điều này là đánh số mỗi câu lệnh trong đặc tả ghi lại và đảm bảo rằng những con số này được phản ánh trong cả thiết kế và kết quả mã. Tuy nhiên, nếu kế hoạch kiểm tra không chỉ rõ rằng điều này phải được thực hiện, thì rất khó xảy ra các tạo tác phân tích, thiết kế và mã sẽ được dán nhãn thích hợp. Do đó, khi thử nghiệm sản phẩm cuối cùng cũng được thực hiện, nhóm SQA sẽ rất khó xác định-của tôi rằng sản phẩm là sự triển khai đầy đủ các thông số kỹ thuật. Trên thực tế, traceabil-ity nên bắt đầu với các yêu cầu; mỗi câu lệnh trong phần tạo tác yêu cầu (hoặc mỗi câu lệnh một phần của mẫu thử nhanh) phải được kết nối với một phần của hiện vật phân tích.

Một khía cạnh mạnh mẽ của việc kiểm tra là danh sách chi tiết các lỗi được phát hiện trong quá trình kiểm tra-sự. Giả sử rằng một nhóm đang kiểm tra các thông số kỹ thuật của một sản phẩm. Như đã giải thích trong Phần 6.2.3, danh sách các lỗi được sử dụng theo hai cách. Đầu tiên, số liệu thống kê lỗi từ cuộc kiểm tra này phải được so sánh với số liệu thống kê lỗi trung bình tích lũy từ đặc điểm kỹ thuật trước đó thanh tra. Sự sai lệch so với các định mức trước đây cho thấy các vấn đề trong dự án. Thứ hai, thống kê lỗi từ việc kiểm tra thông số kỹ thuật hiện tại phải được chuyển đến kiểm tra thiết kế và mã của sản phẩm. Rốt cuộc, nếu có một số lượng lớn các lỗi của một loại cụ thể, có thể không phải tất cả chúng đều được phát hiện trong quá trình kiểm tra thông số kỹ thuật và việc kiểm tra thiết kế và mã cung cấp thêm cơ hội cho định vị bất kỳ lỗi nào còn lại của loại này. Tuy nhiên, trừ khi kế hoạch thử nghiệm nêu rõ rằng chi tiết của tất cả các lỗi phải được ghi lại cẩn thận, không chắc chắn rằng nhiệm vụ này sẽ được thực hiện.

Một cách quan trọng để kiểm tra mô-đun mã được gọi là kiểm tra hộp đen (Phần 15.11) trong đó mã được thực thi với các trường hợp thử nghiệm dựa trên các thông số kỹ thuật. Các thành viên của Nhóm SQA đọc qua các thông số kỹ thuật và vẽ các trường hợp thử nghiệm để kiểm tra xem mã tuân theo tài liệu đặc điểm kỹ thuật. Thời điểm tốt nhất để vẽ các trường hợp thử nghiệm hộp đen là ở cuối của quy trình phân tích, khi các chi tiết của tài liệu đặc tả vẫn còn mới trong tâm trí của các thành viên của nhóm SQA đã kiểm tra họ. Tuy nhiên, trừ khi kế hoạch thử nghiệm tuyên bố rõ ràng rằng các trường hợp thử nghiệm hộp đen sẽ được chọn vào lúc này, trong tất cả các xác suất chỉ có một số trường hợp thử nghiệm hộp đen sẽ được ném cùng nhau sau này. Đó là, một giới hạn số lượng các trường hợp thử nghiệm sẽ được lắp ráp nhanh chóng chỉ khi áp lực bắt đầu lắp từ nhóm lập trình cho nhóm SQA phê duyệt các mô-đun của mình để chúng có thể được tích hợp vào toàn bộ sản phẩm. Kết quả là, chất lượng của sản phẩm nói chung bị ảnh hưởng.

Do đó, mọi kế hoạch thử nghiệm phải chỉ rõ thử nghiệm nào được thực hiện, khi nào được thực hiện và nó được thực hiện như thế nào. Một kế hoạch kiểm tra như vậy là một phần thiết yếu của phần 7.2 của SPMP. Nếu không có nó, chất lượng của sản phẩm tổng thể chắc chắn sẽ bị ảnh hưởng.

9.7 Lập kế hoạch các dự án hướng đối tượng

Giả sử mô hình cổ điển được sử dụng. Từ quan điểm khái niệm, sản phẩm kết quả nói chung là một đơn vị lớn, mặc dù nó bao gồm các mô-đun riêng biệt. Ngược lại, sử dụng

Trang 23

các thành phần độc lập nhỏ hơn, cụ thể là các lớp. Điều này làm cho việc lập kế hoạch đáng kể dễ dàng hơn, trong đó ước tính chi phí và thời lượng có thể được tính toán dễ dàng và chính xác hơn cho các đơn vị nhỏ hơn. Tất nhiên, các ước tính phải tính đến việc một sản phẩm nhiều hơn chứ không chỉ là tổng các phần của nó. Các thành phần riêng biệt không hoàn toàn độc lập; họ có thể gọi nhau, và những tác động này không được bỏ qua.

Các kỹ thuật ước tính chi phí và thời gian được mô tả trong chương này có áp dụng không chuyển sang mô hình hướng đối tượng? COCOMO II (Phần 9.2.4) được thiết kế để xử lý công nghệ phần mềm hiện đại, bao gồm cả hướng đối tượng, nhưng còn các chỉ số trước đó thì sao chẳng hạn như các điểm chức năng (Phần 9.2.1) và COCOMO trung gian (Phần 9.2.3)? bên trong trường hợp COCOMO trung gian, cần có những thay đổi nhỏ đối với một số nhân chi phí [Pittman, 1993]. Ngoài ra, các công cụ ước tính của mô hình cổ điển dường như hoạt động hợp lý trên các dự án hướng đối tượng — với điều kiện là không sử dụng lại. Tái sử dụng đi vào mô hình hướng đối tượng theo hai cách: tái sử dụng các thành phần hiện có trong phát triển và sản xuất có chủ ý (trong dự án hiện tại) các thành phần được tái sử dụng trong các sản phẩm sau này. Cả hai hình thức tái sử dụng đều ảnh hưởng đến quá trình lập dự toán. Sử dụng lại trong thời gian phát triển rõ ràng làm giảm chi phí và thời gian. Các công thức đã được xuất bản hiển thị tiết kiệm như một chức năng của việc tái sử dụng này [Schach, 1994], nhưng những kết quả này liên quan đến mô hình. Hiện tại, không có thông tin về chi phí và thời gian thay đổi như thế nào khi việc sử dụng lại được sử dụng trong quá trình phát triển sản phẩm hướng đối tượng.

Bây giờ chúng ta chuyển sang mục tiêu tái sử dụng các phần của dự án hiện tại. Nó có thể mất khoảng thời gian gấp ba lần để thiết kế, triển khai, kiểm tra và ghi lại thành phần có thể tái sử dụng như một thành phần không thể sử dụng tương tự [Pittman, 1993]. Ước tính chi phí và thời lượng phải đã được sửa đổi để kết hợp lượng lao động bổ sung này và SPMP nói chung phải được điều chỉnh để kết hợp hiệu quả của nỗ lực tái sử dụng. Do đó, hai hoạt động tái sử dụng hoạt động trong hướng ngược nhau. Việc tái sử dụng các thành phần hiện có làm giảm nỗ lực tổng thể trong việc phát triển- nhập một sản phẩm hướng đối tượng, trong khi thiết kế các thành phần để tái sử dụng trong sản phẩm trong tương lai ucts làm tăng nỗ lực. Người ta kỳ vọng rằng, về lâu dài, số tiền tiết kiệm được do tái sử dụng các lớp học sẽ lớn hơn chi phí của những phát triển ban đầu và đã có một số bằng chứng ủng hộ điều này [Lim, 1994].

9.8 Yêu cầu đào tạo

Khi chủ đề **đào tạo** được nêu ra trong các cuộc thảo luận với khách hàng, một phản ứng chung là, “Chúng tôi không cần phải lo lắng về việc đào tạo cho đến khi sản phẩm hoàn thành, sau đó chúng tôi có thể đào tạo người dùng. ” Đây là một nhận xét hơi đáng tiếc, ngụ ý rằng nó chỉ có người dùng yêu cầu đào tạo. Trên thực tế, các thành viên của nhóm phát triển cũng có thể cần đào tạo, bắt đầu- được đào tạo về lập kế hoạch và lập dự toán phần mềm. Khi phát triển phần mềm mới các kỹ thuật, chẳng hạn như kỹ thuật thiết kế mới hoặc quy trình thử nghiệm, được sử dụng, đào tạo phải được được cung cấp cho mọi thành viên trong nhóm bằng cách sử dụng kỹ thuật mới.

Việc giới thiệu mô hình hướng đối tượng có những hệ quả đào tạo chính. Các giới thiệu các công cụ phần cứng hoặc phần mềm như máy trạm hoặc môi trường tích hợp (xem Phần 15.24.2) cũng yêu cầu đào tạo. Các lập trình viên có thể cần được đào tạo về hoạt động- hệ thống của máy được sử dụng để phát triển sản phẩm cũng như trong quá trình triển khai- ngôn ngữ tation. Việc đào tạo chuẩn bị tài liệu thường xuyên bị bỏ qua, bằng chứng là bởi chất lượng kém của quá nhiều tài liệu. Các nhà khai thác máy tính chắc chắn yêu cầu một số

Trang 24

loại đào tạo để có thể chạy sản phẩm mới; họ cũng có thể yêu cầu đào tạo thêm nếu phần cứng mới được sử dụng.

Việc đào tạo bắt buộc có thể đạt được theo một số cách. Cách dễ nhất và ít nhất đột phá là đào tạo nội bộ, bởi nhân viên hoặc chuyên gia tư vấn. Nhiều công ty cung cấp nhiều khóa đào tạo khác nhau, và các trường cao đẳng thường cung cấp các khóa đào tạo vào buổi tối. Các khóa học dựa trên World Wide Web là một lựa chọn thay thế khác.

Khi nhu cầu đào tạo đã được xác định và lập kế hoạch đào tạo, kế hoạch phải được tích hợp vào SPMP.

9.9 Tiêu chuẩn tài liệu

Sự phát triển của một sản phẩm phần mềm đi kèm với nhiều loại **tài liệu-tation** . Jones nhận thấy rằng 28 trang tài liệu được tạo ra trên mỗi 1000 hướng dẫn-tions (KDSI) cho một sản phẩm thương mại nội bộ của IBM có kích thước khoảng 50 KDSI và khoảng 66 trang trên mỗi KDSI cho một sản phẩm phần mềm thương mại có cùng kích thước. Hệ điều hành IMS / 360 Phiên bản 2.3 có kích thước khoảng 166 KDSI và 157 trang tài liệu cho mỗi KDSI đã được sản xuất. Tài liệu thuộc nhiều loại khác nhau, bao gồm lập kế hoạch, xe đẩy, tài chính và kỹ thuật [Jones, 1986a]. Ngoài các loại tài liệu này, bản thân mã nguồn là một dạng tài liệu; nhận xét trong mã cấu thành tài liệu thêm.

Một phần đáng kể của nỗ lực phát triển phần mềm được tiếp thu bởi các tài liệu-tation. Khảo sát 63 dự án phát triển và 25 dự án bảo trì sau giao hàng cho thấy rằng, cứ mỗi 100 giờ dành cho các hoạt động liên quan đến mã, thì 150 giờ đã được sử dụng về các hoạt động liên quan đến tài liệu [Boehm, 1981]. Đối với các sản phẩm TRW lớn, tỷ lệ lượng thời gian dành cho các hoạt động liên quan đến tài liệu đã tăng lên 200 giờ cho mỗi 100 mã-giờ liên quan [Boehm và cộng sự, 1984].

Các tiêu chuẩn là cần thiết cho mọi loại tài liệu. Ví dụ, tính đồng nhất trong tài liệu thiết kế làm giảm sự hiểu lầm giữa các thành viên trong nhóm và hỗ trợ Nhóm SQA. Mặc dù nhân viên mới phải được đào tạo về các tiêu chuẩn tài liệu, không cần đào tạo thêm khi nhân viên hiện tại chuyển từ dự án này sang dự án khác trong tổ chức. Từ quan điểm của bảo trì sau giao hàng, quy trình mã hóa thống nhất-dards hỗ trợ các lập trình viên bảo trì trong việc hiểu mã nguồn. Tiêu chuẩn hóa là thậm chí còn quan trọng hơn đối với hướng dẫn sử dụng, bởi vì chúng phải được đọc bởi nhiều cá nhân, một số ít là chuyên gia máy tính. IEEE đã phát triển một tiêu chuẩn cho hướng dẫn sử dụng (Tiêu chuẩn IEEE 1063 cho Tài liệu Người dùng Phần mềm).

Là một phần của quá trình lập kế hoạch, các tiêu chuẩn phải được thiết lập cho tất cả các tài liệu để được sản xuất trong quá trình sản xuất phần mềm. Các tiêu chuẩn này được kết hợp trong SPMP.

Trường hợp tiêu chuẩn hiện có sẽ được sử dụng, chẳng hạn như Tiêu chuẩn ANSI / IEEE cho Phần mềm Tài liệu Thử nghiệm [ANSI / IEEE 829, 1991], tiêu chuẩn được liệt kê trong phần 2 của SPMP (tài liệu tham khảo). Nếu một tiêu chuẩn được viết đặc biệt cho nỗ lực phát triển, sau đó nó xuất hiện trong phần 6.2 (phương pháp, công cụ và kỹ thuật).

Tài liệu là một khía cạnh thiết yếu của nỗ lực sản xuất phần mềm. Trong rất thực tế cảm giác, sản phẩm *là* tài liệu, bởi vì không có tài liệu, 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 sản xuất phần mềm thành công.

Trang 25

9.10 CASE Công cụ lập kế hoạch và ước tính

Một số công cụ có sẵn để tự động hóa COCOMO trung gian và COCOMO II. Đối với tốc độ tính toán khi giá trị của một tham số được sửa đổi, một số triển khai COCOMO trung gian đã được triển khai bằng các ngôn ngữ bảng tính như Lotus 1-2-3 và Excel. Để phát triển và cập nhật kế hoạch, một trình xử lý văn bản là điều cần thiết.

Các công cụ thông tin quản lý cũng rất hữu ích cho việc lập kế hoạch. Ví dụ, giả sử rằng một tổ chức phần mềm lớn có 150 lập trình viên. Công cụ lập kế hoạch có thể giúp các nhà lập kế hoạch theo dõi những lập trình viên nào đã được giao cho các nhiệm vụ cụ thể và những có sẵn cho dự án hiện tại.

Các loại thông tin quản lý tổng quát hơn cũng là cần thiết. Một số thương mại- Các công cụ quản lý sẵn có quan trọng có thể được sử dụng để hỗ trợ việc lập kế hoạch và ước tính- quá trình ing và giám sát toàn bộ quá trình phát triển. Chúng bao gồm MacProject và Microsoft Project.

9.11 Kiểm tra Kế hoạch Quản lý Dự án Phần mềm

Như đã chỉ ra ở đầu chương này, một lỗi trong quản lý dự án phần mềm- kế hoạch cố vấn có thể có tác động tài chính nghiêm trọng đối với các nhà phát triển. Điều quan trọng là tổ chức phát triển không đánh giá quá cao hoặc đánh giá thấp chi phí của dự án hoặc thời hạn của nó. Vì lý do này, toàn bộ SPMP phải được kiểm tra bởi nhóm SQA trước khi ước tính được cung cấp cho khách hàng. Cách tốt nhất để kiểm tra kế hoạch là kiểm tra kế hoạch.

Nhóm kiểm tra kế hoạch phải xem xét SPMP một cách chi tiết, đặc biệt chú ý đến ước tính chi phí và thời gian. Để giảm rủi ro hơn nữa, bất kể các chỉ số được sử dụng, thời hạn và ước tính chi phí phải được tính toán độc lập bởi một thành viên của SQA nhóm ngay sau khi các thành viên của nhóm lập kế hoạch đã xác định được ước tính của họ.

Chương Ôn tập

Chủ đề chính của chương này là tầm quan trọng của việc lập kế hoạch trong quy trình phần mềm (Phần 9.1). Một thành phần quan trọng của bất kỳ kế hoạch quản lý dự án phần mềm nào là ước tính thời lượng và chi phí (Mục 9.2). Một số chỉ số được đưa ra để ước tính kích thước của sản phẩm, bao gồm cả chức năng điểm (Mục 9.2.1). Tiếp theo, các số liệu khác nhau để ước tính chi phí được mô tả, đặc biệt là COCOMO (Mục 9.2.3) và COCOMO II (Mục 9.2.4). Như được mô tả trong Phần 9.2.5, điều cần thiết là để theo dõi tất cả các ước tính. Ba thành phần chính của kế hoạch quản lý dự án phần mềm — công việc được thực hiện, các nguồn lực để thực hiện nó và số tiền phải trả cho nó - được giải thích trong Phần 9.3. Một SPMP cụ thể, tiêu chuẩn IEEE, được nêu trong Phần 9.4 và được mô tả chi tiết trong Phần 9.5. Tiếp theo, hãy làm theo các phần về kiểm tra lập kế hoạch (Phần 9.6), lập kế hoạch các dự án hướng đối tượng (Phần 9.7), và các yêu cầu đào tạo và tiêu chuẩn tài liệu và ý nghĩa của chúng đối với quá trình lập kế hoạch (Mục 9.8 và 9.9). Các công cụ của CASE để lập kế hoạch và ước tính được mô tả trong Phần 9.10. Chương kết luận bằng tài liệu về việc thử nghiệm kế hoạch quản lý dự án phần mềm (Phần 9.11).

Đối với Thêm nữa đọc hiểu

Tác phẩm bốn tập của Weinberg [Weinberg, 1992; Năm 1993; Năm 1994; 1997] cung cấp thông tin chi tiết về nhiều khía cạnh của quản lý phần mềm, cũng như [Bennatan, 2000] và [Reifer, 2000]. Tháng 9– Số tháng 10 năm 2005 của *IEEE Software* có một số bài báo về quản lý phần mềm, đặc biệt là [Royce, 2005] và [Venugopal, 2005]; có các bài báo bổ sung trong số tháng 5 đến tháng 6 năm 2008. Cách

các nhà quản lý định nghĩa thành công được giải thích trong [Procaccino và Verner, 2006]. Các cơ chế được sử dụng bởi proj- Các nhà quản lý ect để giám sát và kiểm soát các dự án phát triển phần mềm được thảo luận trong [McBride, 2008].

Để biết thêm thông tin về Tiêu chuẩn IEEE 1058 cho Kế hoạch Quản lý Dự án Phần mềm, stan- Bản thân dard nên được đọc kỹ [IEEE 1058, 1998]. Sự cần thiết phải lập kế hoạch cẩn thận được mô tả trong [McConnell, 2001].

Tác phẩm kinh điển của Sackman được mô tả trong [Sackman, Erikson, and Grant, 1968]. Một chi tiết hơn nguồn là [Sackman, 1970]. Tác động của kiến thức chuyên môn của lập trình viên đối với lập trình cặp được mô tả trong [Arisholm, Gallis, Dybå, và Sjøberg, 2007].

Phân tích cẩn thận về các điểm chức năng, cũng như các cải tiến được đề xuất, xuất hiện trong [Symons, Năm 1991]. Điểm mạnh và điểm yếu của các điểm chức năng được trình bày trong [Furey và Kitchenham, 1997]. Điểm lớp, một phần mở rộng của điểm chức năng cho các lớp, được giới thiệu trong [Costagliola, Ferrucci, Tortora, và Vitiello, 2005].

Cơ sở lý thuyết cho COCOMO trung gian, cùng với đầy đủ chi tiết để triển khai-
đề cập đến nó, xuất hiện trong [Boehm, 1981]. COCOMO II được mô tả trong [Boehm và cộng sự, 2000]. Cách của
nâng cao các dự đoán COCOMO được trình bày trong [Smith, Hale và Parrish, 2001]. Một phần mở rộng của
COCOMO đối với các dòng sản phẩm phần mềm xuất hiện trong [In, Baik, Kim, Yang, và Boehm, 2006].

Briand và Wüst [2001] mô tả cách ước tính nỗ lực phát triển cho hướng đối tượng
các sản phẩm. Việc ước tính cả kích thước và khuyết tật của các sản phẩm phần mềm hướng đối tượng được mô tả trong
[Cartwright và Shepperd, 2000].

Dữ liệu năng suất phần mềm cho nhiều sản phẩm xử lý dữ liệu kinh doanh khác nhau được trình bày trong
[Maxwell và Forselius, 2000]; đơn vị của năng suất được sử dụng là điểm hàm trên giờ. Khác
các thước đo năng suất được thảo luận trong [Kitchenham và Mendes, 2004]. Lỗi khi ước tính soft-
nỗ lực về kho được phân tích trong [Jorgensen và Moløkken-Østvold, 2004]. Phê bình về một
quy trình nghiên cứu để so sánh các mô hình ước lượng được đưa ra trong [Myrtveit, Stensrud và Shepperd,
2005]. Mô hình xác suất dự đoán nỗ lực phát triển phần mềm xuất hiện trong [Pendharkar, Sub-
ramanian, và Rodger, 2005]. Phân tích chi phí vượt mức cho các sản phẩm phần mềm được xây dựng bằng
các mô hình vòng đời khác nhau xuất hiện trong [Moløkken-Østvold và Jorgensen, 2005]. Có một hiệu quả
yêu cầu quy trình làm việc có thể có tác động tích cực đến năng suất; điều này được hiển thị trong [Damian và
Chisan, 2006]. Tác động của hình nón của sự không chắc chắn đối với ước tính lịch trình được phân tích trong [Little,
Năm 2006]. Một đánh giá toàn diện về 304 nghiên cứu ước tính chi phí phát triển trên 76 tạp chí là trước
được gửi trong [Jorgensen và Shepperd, 2007]. Một cách tiếp cận dựa trên bằng chứng để lựa chọn một
mô hình ước tính chi phí cho một dự án nhất định được mô tả trong [Menzies và Hihn, 2006].

Điều khoản quan trọng

hoạt động *283*
ước tính chi phí theo thuật toán
mô hình *277*
thành phần ứng dụng
mô hình *281*
đường cơ sở *284*
phương pháp tiếp cận từ dưới lên *277*
COCOMO *278*
COCOMO II *281*
hình nón của sự không chắc chắn *269*
giá *271*
dự toán chi phí *271*
Kỹ thuật Delphi *276*
tài liệu *291*
thời lượng *282*
ước tính thời lượng *271*
thiết kế ban đầu mô hình *281*
hiệu quả *273*
phán đoán chuyên môn bằng phép loại suy *275*
chi phí bên ngoài *271*
FFP số liệu *273*
điểm chức năng (FP) *273*
Dự án phần mềm IEEE
kế hoạch quản lý *286*
chi phí nội bộ *271*
dòng mã (LOC) *272*
cột mốc *284*
tiền *284*
nỗ lực danh nghĩa *278*
lập kế hoạch *268*
mô hình hậu kiến trúc *281*
giá *271*
năng suất *273*
chức năng dự án *283*
Phân phối Rayleigh *282*
tài nguyên *282*
đánh giá *284*
nỗ lực phát triển phần mềm
số nhân (SPMP) *278*

Trang 27

nhiệm vụ *283*
yếu tố phức tạp kỹ thuật
(TCF) *274*
lập kế hoạch kiểm tra *288*
ngàn giao
hướng dẫn nguồn
(KDSI) *272*
đào tạo *290*
các điểm chức năng chưa được điều chỉnh
(UFP) *273*
gói công việc *284*
sản phẩm công việc *283*

Các vấn đề

9.1 Tại sao bạn nghĩ rằng một số tổ chức phần mềm hoài nghi coi các *cột mốc* là *cối xay* ?
(Gợi ý: Tra nghĩa bóng của *cối xay* trong từ điển.)

9.2 Bạn là kỹ sư phần mềm tại Pretoriuskop Software Developers. Một năm trước, người quản lý của bạn
đã thông báo rằng sản phẩm tiếp theo của bạn sẽ bao gồm 8 tệp, 48 luồng và 91 quy trình.

(i) Sử dụng số liệu FFP, xác định kích thước của nó.

(ii) Đối với các nhà phát triển phần mềm Pretoriuskop, hằng số *d* trong phương trình (9.2) đã được xác định-
được khai thác là $ 1021. Chỉ số FFP dự đoán chi phí nào?

(iii) Sản phẩm gần đây đã được hoàn thành với chi phí $ 135,200. Điều này cho bạn biết về điều gì
năng suất của nhóm phát triển của bạn?

9.3 Một sản phẩm mục tiêu có 8 đầu vào đơn giản, 3 đầu vào trung bình và 11 đầu vào phức tạp. Có 57 aver-
đầu ra tuổi, 9 câu hỏi đơn giản, 13 tệp chính trung bình và 18 giao diện phức tạp. Mục đích
các điểm chức năng chưa được điều chỉnh (*UFP*).

9.4 Nếu tổng mức độ ảnh hưởng đối với sản phẩm của Vấn đề 9.3 là 47, hãy xác định số
điểm chức năng.

9.5 Bạn nghĩ tại sao mặc dù có nhược điểm của nó, các dòng mã (LOC hoặc KDSI) lại được sử dụng rộng rãi như vậy
như một thước đo về kích thước sản phẩm?

9.6 Bạn chịu trách nhiệm phát triển sản phẩm nhúng 62-KDSI trên danh nghĩa ngoại trừ kích thước cơ sở dữ liệu được đánh giá rất cao và việc sử dụng các công cụ phần mềm thấp. Sử dụng trung gian COCOMO, nỗ lực ước tính tính bằng tháng người là bao nhiêu?

9.7 Bạn phụ trách phát triển hai sản phẩm chế độ hữu cơ 31-KDSI. Cả hai đều là danh nghĩa trong mọi khía cạnh ngoại trừ sản phẩm P1 có độ phức tạp cực cao và sản phẩm P2 có độ phức tạp cực thấp sự phức tạp. Để phát triển sản phẩm, bạn có hai nhóm tùy ý. Đội A có rất cao khả năng của nhà phân tích, kinh nghiệm ứng dụng và khả năng của lập trình viên. Đội A cũng có kinh nghiệm máy ảo và kinh nghiệm ngôn ngữ lập trình. Đội B được đánh giá rất thấp trên tất cả năm thuộc tính.

(i) Tổng nỗ lực (tính bằng người-tháng) là bao nhiêu nếu đội A phát triển sản phẩm P1 và đội B phát triển sản phẩm P2?

(ii) Tổng nỗ lực (tính theo tháng) là bao nhiêu nếu đội B phát triển sản phẩm P1 và đội A phát triển sản phẩm P2?

(iii) Việc phân công cán bộ nào trong hai cách trước đây hợp lý hơn? Là trực giác của bạn được hỗ trợ bởi các dự đoán của COCOMO trung gian?

9.8 Bạn chịu trách nhiệm phát triển một sản phẩm chế độ hữu cơ 48-KDSI được coi là danh nghĩa trong mọi sự tôn trọng.

(i) Giả sử chi phí là 10.100 đô la một người / tháng, thì dự án ước tính là bao nhiêu Giá cả?

(ii) Toàn bộ nhóm phát triển của bạn từ chức khi bắt đầu dự án. Bạn đủ may mắn để có thể thay thế đội danh nghĩa bằng một đội rất giàu kinh nghiệm và có năng lực, nhưng chi phí cho mỗi người-tháng sẽ tăng lên $ 13,400. Bạn mong đợi bao nhiêu tiền được (hay mất) do thay đổi nhân sự?

9.9 Bạn chịu trách nhiệm phát triển phần mềm cho một sản phẩm sử dụng một bộ phát triển mới đã chọn các thuật toán để tính toán các tuyến đường tiết kiệm chi phí nhất cho một công ty vận tải đường bộ lớn. Sử dụng

Trang 28

COCOMO trung gian, bạn xác định rằng giá thành của sản phẩm sẽ là $ 470,000. Tuy nhiên, như một séc, bạn yêu cầu một thành viên trong nhóm của bạn ước tính nỗ lực bằng cách sử dụng các điểm chức năng. Bà ấy báo cáo rằng chỉ số điểm chức năng dự đoán chi phí là 985.000 đô la, lớn hơn gấp đôi so với dự đoán COCOMO của bạn. Bạn làm gì bây giờ?

9.10 Chứng tỏ rằng phân bố Rayleigh [phương trình (9.9)] đạt giá trị lớn nhất khi $t = k$. Tìm thấy mức tiêu thụ tài nguyên tương ứng.

9.11 Kế hoạch bảo trì sau giao hàng sản phẩm được coi là "thành phần bổ sung" của IEEE kế hoạch quản lý dự án phần mềm. Ghi nhớ rằng mọi sản phẩm tầm thường đều được duy trì và chi phí bảo trì sau giao hàng trung bình là khoảng gấp đôi hoặc ba lần chi phí của việc phát triển sản phẩm, điều này có thể được biện minh như thế nào?

9.12 Tại sao các dự án phát triển phần mềm tạo ra quá nhiều tài liệu?

9.13 (Dự án kỳ hạn) Xem xét dự án Chocoholics Anonymous được mô tả trong Phụ lục A. Tại sao không thể ước tính chi phí và thời gian hoàn toàn dựa trên thông tin trong Phụ lục A?

9.14 (Các bài đọc về Kỹ thuật phần mềm) Người hướng dẫn của bạn sẽ phân phối các bản sao của [Costagliola, Ferrucci, Tortora, và Vitiello, 2005]. Bạn có bị thuyết phục bởi việc xác thực theo kinh nghiệm của điểm lớp không?

Người giới thiệu

[Albrecht, 1979] AJ A LBRECHT , "Đo lường năng suất phát triển ứng dụng," *Kỷ yếu của Hội nghị chuyên đề phát triển ứng dụng SHARE / GUIDE của IBM* , Monterey, CA, tháng 10 năm 1979, trang 83–92.

[ANSI / IEEE 829, 1991] *Tài liệu Kiểm tra Phần mềm* , ANSI / IEEE 829-1991, Quốc gia Hoa Kỳ Viện Tiêu chuẩn, Viện Kỹ sư Điện và Điện tử, New York, 1991.

[Arisholm, Gallis, Dybå, và Sjoberg, 2007] E. Một RISHOLM , H. G ALLIS , T. D YBA , VÀ Dik S JØBERG , "Đánh giá lập trình theo cặp dựa trên độ phức tạp của hệ thống và kiến thức chuyên môn của lập trình viên," *Giao dịch IEEE về Kỹ thuật phần mềm* **33** (tháng 2 năm 2007), trang 65–86.

[Bennatan, 2000] EM B ENNATAN , *Đúng giờ trong ngân sách: Thực tiễn quản lý dự án phần mềm và Kỹ thuật* , xuất bản lần thứ 3, John Wiley và Sons, New York, 2000.

[Boehm, 1981] BW B OEHM , *Kinh tế kỹ thuật phần mềm*, Prentice Hall, Englewood Cliffs, NJ, 1981.

[Boehm, 1984] BW B OEHM , "Kinh tế kỹ thuật phần mềm", *Giao dịch IEEE trên phần mềm Kỹ thuật* **SE-10** (tháng 1 năm 1984), trang 4–21.

[Boehm và cộng sự, 1984] BW B OEHM , MH P ENEDO , ED S TUCKLE , RD Wvania , VÀ AB P YSTER , "Môi trường phát triển phần mềm để cải thiện năng suất", *IEEE Computer* **17** (Tháng 6 1984), trang 30–44.

[Boehm và cộng sự, 2000] BW B OEHM , C. A BTS , AW B ROWN , S. C HULANI , BK C LARK , E. H OROWITZ , R. M ADACHY , D. R EIFER , VÀ B. S TEECE , *Ước tính chi phí phần mềm với COCOMO II* , Prentice Hall, Upper Saddle River, NJ, 2000.

[Briand và Wüst, 2001] LC B RIAND VÀ J. W ÜST , "Nỗ lực phát triển mô hình hóa trong đối tượng-Hệ thống định hướng sử dụng thuộc tính thiết kế, " *Giao dịch IEEE về kỹ thuật phần mềm* **27** (Tháng 11 năm 2001), trang 963–86.

[Cartwright và Shepperd, 2000] M. C ARTWRIGHT VÀ M. S HEPPERD , "Một cuộc điều tra thực nghiệm về một Hệ thống Phần mềm Hướng Đối tượng, " *Giao dịch IEEE về Kỹ thuật Phần mềm* **26** (Tháng 8 2000), trang 786–95.

[Costagliola, Ferrucci, Tortora và Vitiello, 2005] G. C OSTAGLIOLA , F. F ERRUCCI , G. T ORTORA , VÀ G. V ITIELLO , "Điểm lớp: Phương pháp tiếp cận để ước tính kích thước của các hệ thống hướng đối tượng," *Giao dịch IEEE về Kỹ thuật phần mềm* **31** (tháng 1 năm 2005), trang 52–74.

Trang 29

[Damian và Chisan, 2006] D. D AMIAN VÀ J. C HISAN , "Một nghiên cứu thực nghiệm về Rela-mối quan hệ giữa các Quy trình Kỹ thuật Yêu cầu và Các Quy trình khác Dẫn đến Khoản thanh toán trong Năng suất, Chất lượng và Quản lý Rủi ro, " *Giao dịch IEEE về Kỹ thuật Phần mềm* **32** (Tháng 7 năm 2006), trang 433–53.

[Devenny, 1976] T. D EVENNY , "Một nghiên cứu khám phá về ước tính chi phí phần mềm tại điện tử Bộ phận Hệ thống, "Luận văn số GSM / SM / 765–4, Viện Công nghệ Không quân, Dayton, OH, Năm 1976.

[Furey và Kitchenham, 1997] S. F UREY VÀ B. K ITCHENHAM , "Các điểm chức năng," *Phần mềm IEEE* **14** (Tháng 3 - Tháng 4 năm 1997), trang 28–32.

[IEEE 1058, 1998] "Tiêu chuẩn IEEE cho các kế hoạch quản lý dự án phần mềm." IEEE Std. 1058-1998, Viện Kỹ sư Điện và Điện tử, New York, 1998.

[In, Baik, Kim, Yang và Boehm, 2006] HP I N , J. B AIK , S. K IM , Y. Y ANG , AND B. B OEHM , "A Quality-Mô hình ước tính chi phí dựa trên vòng đời của dòng sản phẩm ", *Thông báo của ACM* **49** (Tháng 12 năm 2006), trang 85–88.

[Jones, 1986a] C. J ONES , *Năng suất lập trình*, McGraw-Hill, New York, 1986.

[Jones, 1987] C. J ONES , Thư gửi biên tập viên, *IEEE Computer* **20** (tháng 12 năm 1987), tr. 4.

[Jorgensen và Moløkken-Østvold, 2004] M. J ORGENSEN và K. M OLØKKEN -Ø STVOLD , "Lý do Lỗi ước tính nỗ lực phần mềm: Tác động của vai trò người trả lời, Phương pháp tiếp cận thu thập thông tin, và Phương pháp Phân tích Dữ liệu, " *Giao dịch IEEE về Kỹ thuật Phần mềm* **30** (tháng 12 năm 2004), trang 993–1007.

[Jorgensen và Shepperd, 2007] M. J ORGENSEN VÀ M. S HEPPERD , "Đánh giá có hệ thống về phần mềm Nghiên cứu ước tính chi phí phát triển, " *Giao dịch IEEE về Kỹ thuật phần mềm* **32** (Tháng 1 2007), trang 33–53.

[Kitchenham và Mendes, 2004] B. K ITCHENHAM VÀ E. M ENDES , "Đo lường năng suất phần mềm-để cập đến Sử dụng nhiều biện pháp kích thước, " *Giao dịch IEEE về Kỹ thuật phần mềm* **30** (Tháng 12 2004), trang 1023–35.

[Lim, 1994] WC L IM , "Ảnh hưởng của việc tái sử dụng đối với chất lượng, năng suất và kinh tế," *Phần mềm IEEE* **11** (tháng 9 năm 1994), trang 23–30.

[Little, 2006] T. L ITTLE , "Ước tính lịch trình và sự không chắc chắn xung quanh hình nón của sự bất định-t chắc chắn, " *IEEE Software* **23** (tháng 5 - tháng 6 năm 2006), trang 48–54.

[Maxwell và Forselius, 2000] KD M AXWELL VÀ P. F ORSELIUS , "Benchmarking Software Devel-opment Productivity, " *IEEE Software* **17** (Tháng 1 - Tháng 2 năm 2000), trang 80–88.

[McBride, 2008] T. M C B RIDE , "Cơ chế quản lý dự án phát triển phần mềm," *Tạp chí Hệ thống và Phần mềm* **81** (tháng 12 năm 2008), trang 2386–95.

[McConnell, 2001] S. M C C ONNELL , "Chín đại tội của việc lập kế hoạch dự án," *IEEE Software* **18** (Tháng 11 - tháng 12 năm 2001), trang 5–7.

[Menzies và Hihn, 2006] T. M ENZIES AND J. H IHN , "Ước tính chi phí dựa trên bằng chứng để tốt hơn-Phần mềm Chất lượng, " *IEEE Software* **23** (tháng 7 đến tháng 8 năm 2006), trang 64–66.

[Moløkken-Østvold và Jorgensen, 2005] K. M OLØKKEN -Ø STVOLD VÀ M. J ORGENSEN , "A Com-tạm dừng các lần vượt qua dự án phần mềm — Mô hình phát triển linh hoạt so với tuần tự, " *IEEE Giao dịch trên Kỹ thuật phần mềm* **31** (tháng 9 năm 2005), trang 754–66.

[Myrtveit, Stensrud và Shepperd, 2005] I. M YRTVEIT , E. S TENSRUD , VÀ M. S HEPPERD , "Độ tin cậy

và Tính hợp lệ trong Nghiên cứu So sánh của Mô hình Dự đoán Phần mềm, " *Giao dịch IEEE trên Phần mềm Kỹ thuật* **31** (tháng 5 năm 2005), trang 380–91.

[Norden, 1958] PV N ORDEN , "Sự phù hợp đường cong cho một mô hình nghiên cứu và phát triển ứng dụng Lập kế hoạch, " *Tạp chí Nghiên cứu và Phát triển số* **2** *của IBM* (tháng 7 năm 1958), trang 232–48.

Trang 30

[Pendharkar, Subramanian, và Rodger, 2005] PC P ENDHARKAR , GH S UBRAMANIAN , VÀ JA R ODGER , "Mô hình xác suất để dự đoán nỗ lực phát triển phần mềm," *IEEE Transac- về Kỹ thuật phần mềm* **31** (tháng 7 năm 2005), trang 615–24.

[Pittman, 1993] M. P ITTMAN , "Bài học kinh nghiệm trong quản lý phát triển hướng đối tượng," *IEEE Phần mềm* **10** (tháng 1 năm 1993), trang 43–53.

[Procaccino và Verner, 2006] JD P ROCACCINO VÀ JM V ERNER , "Công nghiệp nhanh nhẹn như thế nào Thực tiễn Phát triển Phần mềm? " *Tạp chí Hệ thống và Phần mềm* **79** (tháng 11 năm 2006), trang 1541–51.

[Putnam, 1978] LH P UTNAM , "Một Giải pháp Thực nghiệm Chung cho Định cỡ Phần mềm Macro và Ước tính vấn đề, " *Giao dịch IEEE trên Kỹ thuật phần mềm* **SE-4** (tháng 7 năm 1978), trang 345–61.

[Reifer, 2000] DJ R EIFER , "Quản lý phần mềm: Tốt, xấu và xấu," *IEEE Soft- kho* **17** (tháng 3 - tháng 4 năm 2000), trang 73–75.

[Royce, 2005] W. R OYCE , "Phong cách quản lý phần mềm thành công: Chỉ đạo và cân bằng," *IEEE Phần mềm* **22** (tháng 9 - tháng 10 năm 2005), trang 40–47.

[Sackman, 1970] H. S ACKMAN , *Giải quyết vấn đề Con người – Máy tính: Thực nghiệm Đánh giá Thời gian- Chia sẻ và xử lý hàng loạt*, Auerbach, Princeton, NJ, 1970.

[Sackman, Erikson và Grant, 1968] H. S ACKMAN , WJ E RIKSON , AND EE G RANT , "Exploratory Nghiên cứu thử nghiệm so sánh hiệu suất lập trình trực tuyến và ngoại tuyến, " *Communica- phần của ACM* **11** (tháng 1 năm 1968), trang 3-11.

[Schach, 1994] SR S CHACH , "Tác động kinh tế của việc tái sử dụng phần mềm đối với việc bảo trì," *Tạp chí của Bảo trì phần mềm: Nghiên cứu và Thực hành* **6** (tháng 7 đến tháng 8 năm 1994), trang 185–96.

[Smith, Hale và Parrish, 2001] RK S MITH , JE H ALE , AND AS P ARRISH , "Một nghiên cứu thực nghiệm Sử dụng các mẫu phân công nhiệm vụ để cải thiện độ chính xác của ước tính nỗ lực phần mềm, " *IEEE Giao dịch trên Kỹ thuật phần mềm* **27** (tháng 3 năm 2001), trang 264–71.

[Symons, 1991] CR S YMONS , *Định cỡ và Ước tính phần mềm: Mk II FPA* , John Wiley và Sons, Chichester, Vương quốc Anh, 1991.

[van der Poel và Schach, 1983] KG VAN DER P OEL VÀ SR S CHACH , "Một số liệu phần mềm cho chi phí Ước tính và Đo lường hiệu quả trong phát triển hệ thống xử lý dữ liệu, " *Tạp chí Hệ thống và Phần mềm* **3** (tháng 9 năm 1983), trang 187–91.

[Venugopal, 2005] C. V ENUGOPAL , "Bộ mục tiêu duy nhất: Mô hình mới cho sự thành công của IT Megaproject," *Phần mềm IEEE* **22** (tháng 9 - tháng 10 năm 2005), trang 48–53.

[Weinberg, 1992] GM W EINBERG , *Quản lý phần mềm chất lượng: Tư duy hệ thống* , Vol. 1, Dorset Nhà, New York, 1992.

[Weinberg, 1993] GM W EINBERG , *Quản lý phần mềm chất lượng: Đo lường bậc nhất* , Tập. 2, Nhà Dorset, New York, 1993.

[Weinberg, 1994] GM W EINBERG , *Quản lý phần mềm chất lượng: Hành động đồng thời* , Vol. 3, Dorset Nhà, New York, 1994.

[Weinberg, 1997] GM W EINBERG , *Quản lý phần mềm chất lượng: Dự đoán thay đổi* , Vol. 4, Nhà Dorset, New York, 1997.

Trang 31

Trang này cố ý để trống