



Quản lý dự án Công Nghệ Thông Tin

Phần 5. Ước lượng Dự án

NỘI DUNG

- Độ đo phần mềm: LOC, FP và các độ đo dẫn xuất
- Ước lượng, khâu yếu nhất của quản trị dự án

TẦM QUAN TRỌNG

- Ước lượng dự án hiện là khâu yếu nhất hiện nay. Không ước lượng được thì dự án rất dễ vỡ kế hoạch về thời gian và tài chính.
- Thực tế không dự án nào có thể ước lượng chính xác, ước lượng cần được thực hiện nhiều vòng. Mức ước lượng trong giai đoạn xác định có thể sai tới 50-100%, nhưng trong giai đoạn thiết kế tổng thể phải giảm tới 25-50%, còn trong giai đoạn thiết kế chi tiết chỉ còn 10-25%
- Ước lượng chỉ có thể chính xác nếu phân rã được các vấn đề nhỏ hơn, đó là kỹ thuật chia để trị (divide and conquer)

CÁC PHƯƠNG PHÁP ƯỚC LƯỢNG

- **Ước lượng chuyên gia:** các chuyên gia đã có kinh nghiệm triển khai dự án phần mềm, có thể trả lời ngay các ước lượng tuy rằng không phải lúc nào độ chính xác cũng đáng tin cậy
- **Đánh giá bằng kinh nghiệm quá khứ.** Phải có số liệu quá khứ, phải hiểu được tình hình hiện tại
- **Đánh giá bằng các mô hình ước lượng thực nghiệm.** Phải có các tham số về dự án (các độ đo)

ĐỘ ĐO

- Khái niệm độ đo: là các chỉ số đặc trưng cho một khía cạnh nào đó. Trong công nghệ phần mềm có độ đo của phần mềm (software metric/software measure), độ đo của dự án (project metric) và độ đo của quy trình phần mềm (process metric).
- Có độ đo trực tiếp và độ đo gián tiếp. Độ đo trực tiếp là độ đo có thể tình đếm trực tiếp không thông qua các độ đo khác (ví dụ độ đo **LOC** – lines of code), có độ đo gián tiếp là các độ đo tính qua các độ đo khác (ví dụ tỉ lệ lỗi = số lỗi / số dòng mã nguồn)
- Dự án cũng có độ đo, chi phí cho dự án, năng suất của dự án,
- Quy trình phần mềm cũng có độ đo, chẳng hạn tỉ lệ chi phí trung bình cho mỗi giai đoạn phát triển phần mềm đối với quy trình thác nước

ĐỘ ĐO LOC – METRIC HƯỚNG QUY MÔ PHẦN MỀM

- **LOC** (*lines of code*) hay **KLOC** (nghìn dòng lệnh). Độ đo này chỉ có thể chính xác sau khi dự án đã kết thúc. Tuy nhiên bằng kinh nghiệm, hoặc bằng thống kê tương tự có thể ước lượng được khối lượng mã nguồn của một phần mềm trước khi kết thúc dự án.
- LOC sau khi kết thúc dự án sẽ được dùng để ước lượng các dự án tương tự sau này.
- Các độ đo dẫn xuất: số lỗi trên KLOC, chi phí trên KLOC, số tài liệu trên KLOC, năng suất số KLOC /manmonth.
- LOC phụ thuộc vào môi trường lập trình nên khó so sánh giữa các dự án nếu chúng phát triển trên các môi trường lập trình khác nhau

ĐIỂM CHỨC NĂNG (FUNCTION POINT) METRIC HƯỚNG CHỨC NĂNG

- Điểm chức năng (FP) đo độ phức tạp của phần mềm. Quy mô chỉ phản ánh một khía cạnh nhỏ của độ phức tạp, chính chức năng thể hiện độ phức tạp chính xác hơn
- FP được tính qua 5 yếu tố chính và 14 yếu tố phụ. Các yếu tố chính là
 - Số user input (số các thành phần dữ liệu đưa vào), số các input được dùng trong các câu hỏi khác nhau được tính riêng rẽ
 - Số user output (xuất hiện trong các report, các màn hình, các thông báo). Các output trong các câu hỏi khác nhau được kể riêng rẽ
 - Số truy vấn (inquiry) của người sử dụng - số input trong các truy vấn online
 - Số lượng file logic (có thể chỉ là một phần của CSDL, có thể tính như một bảng của CSDL) và các file độc lập
 - Số lượng các giao tiếp ngoài: ngoại vi, các hệ thống thông tin khác mà nó giao tiếp

ĐIỂM CHỨC NĂNG (FUNCTION POINT) METRIC HƯỚNG CHỨC NĂNG

Mỗi yếu tố trên được gán một trọng số, tùy theo ảnh hưởng của mỗi yếu tố và tùy theo mức độ phức tạp: thường tính theo 3 mức là đơn giản, trung bình và phức tạp.

Ví dụ

	Tham số đo	Số đo	ĐG	TB	PT	
1	Số input	25	3	4	6	100
2	Số output	30	4	5	7	150
3	Số inquiry	20	3	4	6	120
4	Số file	10	7	10	15	70
5	Số tương tác ngoài	10	5	7	10	70
	Tổng					510

ĐIỂM CHỨC NĂNG (FUNCTION POINT)

14 YẾU TỐ ĐIỀU CHỈNH PHỤ

1. Hệ thống đòi hỏi backup và khôi phục tin cậy
2. Đòi hỏi dữ liệu truyền thông
3. Có các chức năng phân tán
4. Hiệu năng là điều quan trọng
5. Yêu cầu sử dụng môi trường nặng
6. Hệ thống đòi hỏi dữ liệu on-line
7. Khi đòi hỏi dữ liệu online, cần nhiều màn hình dữ liệu hoặc nhiều xử lý
8. Master file được cập nhật online
9. Input, output, file, và tính toán online phức tạp
10. Quá trình xử lý bên trong phức tạp
11. Mã được thiết kế để dùng lại
12. Việc chuyển đổi và cài đặt được tinh ngay trong thiết kế
13. Hệ thống được thiết kế để có thể cài đặt nhiều lần cho các tổ chức khác nhau
14. Ứng dụng được thiết kế để dễ thay đổi và làm dễ dàng sử dụng cho người dùng

Mỗi Fi được từ 0 tới 5 điểm tùy theo mức độ

$$FP = \text{Điểm của các yếu tố chính} \times [0.65 + \sum F_i / 100]$$

MÔ HÌNH ƯỚC LƯỢNG THỰC NGHIỆM

- Hầu hết các mô hình thực nghiệm đều phải có đầu vào từ một độ đo độ phức tạp của dự án có thể là LOC hay FP
- Mô hình chính $E = A + B \times V^C$ trong đó E là công sức có thể đo bằng người x tháng, A, B và C là một hằng số đặc trưng cho mô hình và V là LOC hay FP. Mô hình này cho thấy công sức không tỉ lệ tuyến tính theo độ phức tạp
- Một vài ước lượng
 - Waston Felix $E = 5.2 + KLOC^{0.91}$
 - Baley Basil $E = 5.5 + 0.73 KLOC^{1.16}$
 - Matson Barret $E = 585.7 + 15.12 \times FP$

MÔ HÌNH ƯỚC LƯỢNG THỰC NGHIỆM COCOMO

- COCOMO : Constructive Cost Model (Barry Boehm- 1981)
- COCOMO II (1996) có phân mức đối với các dự án: dự án ở mức định hướng (làm bản mẫu, xem xét công nghệ, giao diện, hiệu năng), dự án ở mức trước khi thiết kế (khi yêu cầu phần mềm đã ổn định và kiến trúc đã được xác lập) và dự án ở tầng sau kiến trúc (khi phần mềm được xây dựng)

BA MÔ HÌNH CƠ BẢN CỦA COCOMO

- Mô hình 1. Mô hình COCOMO cơ sở tính công sức phát triển phần mềm (và chi phí) xem như hàm của kích cỡ chương trình được diễn đạt theo số dòng mã ước lượng.
- Mô hình 2. COCOMO trung bình tính công sức phát triển phần mềm như một hàm của kích cỡ chương trình và một tập các "hướng dẫn chi phí" bao hàm cả các đánh giá chủ quan về sản phẩm, phần cứng, nhân sự và các thuộc tính dự án.
- Mô hình 3. COCOMO nâng cao tổ hợp của tất cả các đặc trưng của phiên bản trung bình với việc đánh giá của ảnh hưởng của hướng dẫn chi phí lên từng bước (phân tích, thiết kế ...) của tiến trình kỹ nghệ phần mềm.

COCOMO

- Phương trình COCOMO cơ bản có dạng sau:

$$E = a \times \text{KLOC}^b, \quad D = c \times E^d$$

Trong đó E là công sức được áp dụng theo người/tháng, D là thời gian phát triển theo người/tháng, và KLOC là số được ước lượng về số dòng mã phải bàn giao

- Mô hình cơ sở được mở rộng để xem xét một tập các "thuộc tính hướng dẫn chi phí" thuộc tính sản phẩm, các thuộc tính phần cứng, các thuộc tính nhân viên và các thuộc tính dự án. Boehm đưa vào nhân tố điều chỉnh công sức (EAF). Giá trị điển hình cho EAF lấy trong miền từ 0.9 đến 1.4.
- Phương trình COCOMO trung bình có dạng

$$E = a \times \text{KLOC}^b \times \text{EAF}$$

COCOMO

- Thực tế triển khai, COCOMO được chi tiết hoá rất nhiều, người ta xây dựng các phiên bản phụ thuộc vào điều kiện cụ thể, một số các tham số của COCOMO phải dựa vào số liệu quá khứ. Đã có một số phần mềm ước lượng
- Người ta tính nỗ lực theo từng giai đoạn (thiết kế sơ bộ, thiết kế chi tiết, lập trình và kiểm thử module, kiểm thử). Tham số đầu vào của COCOMO là chi phí hàng tháng cho nhân viên tùy theo từng loại như người phân tích, người lập trình, người kiểm thử, nhân viên hành chính, nhân viên làm tài liệu, mỗi loại đó đều có một trọng số để điều chỉnh