

1. ĐỊNH GIÁ PHẦN MỀM

Hướng dẫn chung

- Hướng dẫn này dùng để xác định giá trị các phần mềm theo mô hình hướng đối tượng được phát triển mới, hoặc nâng cấp, chỉnh sửa sử dụng nguồn vốn ngân sách nhà nước phục vụ cho ứng dụng công nghệ thông tin trong hoạt động của cơ quan nhà nước.
- Xác định giá trị phần mềm là việc xác định khối lượng công việc cụ thể, phương thức tính toán, kiểm tra trên cơ sở nỗ lực giờ công để thực hiện các trường hợp sử dụng (use- case) quy định trong Biểu đồ use-case theo các chỉ dẫn có liên quan trên nguyên tắc tuân thủ các tiêu chuẩn, quy định về ứng dụng công nghệ thông tin của Việt Nam.
- Các từ ngữ dưới đây được hiểu như sau:
 - “Software - phần mềm” là những chương trình điều khiển các chức năng phần cứng và hướng dẫn phần cứng thực hiện các tác vụ của mình.
 - “Use case”: thể hiện một giao tác cụ thể thuộc tập trình tự các giao tác mà phần mềm thực hiện để cung cấp hay tiếp nhận các giá trị khi một chức năng của phần mềm được sử dụng.
 - “Use case diagram – Biểu đồ use case” dùng mô tả các tác nhân và kết nối giữa tác nhân với các use case nhằm miêu tả chức năng mà phần mềm cung cấp.
 - “Actor - tác nhân” là người hay hệ thống bên ngoài tương tác, trao đổi thông tin với phần mềm.
 - “Activity diagram – Biểu đồ hoạt động” thể hiện quá trình thực hiện của một hay nhiều hoạt động được gắn với một lớp (class) đối tượng dưới tác động của các sự kiện bên ngoài.
 - “Unified Modelling Language (UML) - ngôn ngữ mô hình hóa thống nhất” dùng để đặc tả một phần hay toàn bộ phần mềm với các mô hình nghiệp vụ từ những góc nhìn ở từng mức trừu tượng khác nhau thông qua sử dụng các cấu tử mô hình tạo nên các biểu đồ (diagram) thể hiện các đối tượng gồm cấu trúc dữ liệu, hành vi cũng như cách các đối tượng kết hợp với nhau và được sử dụng để đặc tả khi phát triển, hoặc nâng cấp, chỉnh sửa phần mềm.
 - “Phát triển phần mềm nội bộ” là việc phát triển, gia công, sản xuất phần mềm theo các yêu cầu riêng của khách hàng hoặc người sử dụng nhằm đáp ứng yêu cầu đặc thù của tổ chức và được sử dụng trong nội bộ tổ chức đó.
 - “Chỉnh sửa phần mềm nội bộ” là việc sửa đổi, làm tăng hiệu năng của phần mềm nội bộ đã có nhằm tối ưu hóa khả năng xử lý các yêu cầu của người sử dụng.
 - “Nâng cấp phần mềm nội bộ” là việc chỉnh sửa phần mềm nội bộ với việc tăng cường chức năng-khả năng xử lý của phần mềm đã có nhằm đáp ứng thêm một số yêu cầu của người sử dụng.
- Giá trị phần mềm được xác định là cơ sở cho việc lập chi phí ứng dụng công nghệ thông tin trong hoạt động của cơ quan nhà nước, lập bảng khối lượng mời thầu khi tổ chức lựa chọn nhà thầu, quản lý chi phí trong trường hợp chỉ định thầu hoặc tự thực hiện.
- Nội dung hồ sơ phục vụ cho việc xác định giá trị đối với phần được phát triển, hoặc được nâng cấp, chỉnh sửa của phần mềm bao gồm các tài liệu nêu tại mục 2 Phần II.
- Đối với những xử lý phức tạp của phần mềm nhưng không thể xác định được khối lượng giờ công cụ thể thì có thể tạm xác định và ghi chú là “khối lượng tạm tính”. Khối lượng tạm tính này sẽ được xác định lại khi quyết toán hoặc thực hiện theo quy định cụ thể tại hợp đồng giao nhận thầu phát triển, hoặc nâng cấp, chỉnh sửa phần mềm.
- Đối với những phần mềm có điểm đặc thù mà hướng dẫn xác định chi phí giờ công chưa phù hợp thì các tổ chức, cá nhân có thể tự đưa phương pháp xác định giờ công trên cơ sở phù hợp với hướng dẫn này và phải có thuyết minh cụ thể cách tính.
- Trường hợp sử dụng các tài liệu, hướng dẫn của nước ngoài để thực hiện việc xác định chi phí giờ công cần nghiên cứu, tham khảo hướng dẫn này để bảo đảm nguyên tắc thống nhất về quản lý chi phí phát triển, hoặc nâng cấp, chỉnh sửa phần mềm phục vụ ứng dụng công nghệ thông tin trong hoạt động của cơ quan nhà nước.

2. QUY TRÌNH THỰC HIỆN PHẦN MỀM

Cách thức triển khai, cài đặt scrum:

- Thu nhập các đặc điểm của sản phẩm (backlog) trong đơn đặt hàng >>Lập team và tìm ra PO(Product Owner)
- Ước lượng đầy các yêu cầu về sản phẩm đầu ra >> Ước lượng công việc cho mỗi team.
- Lên kế hoạch phát triển các vòng lặp sprint >> Xác định khoảng thời gian, mục tiêu, phân tích các yêu cầu(Userstory)
- Lên kế hoạch phát triển các nhiệm vụ của sprint >> Ước lượng thời gian làm từng task, nhận dạng task quan trọng.
- Tạo ra không gian làm việc cộng tác cho tất cả mọi người >> Phân công công việc cho từng thành viên.
- Xây dựng từng sprint, lập trình, kiểm thử và điều chỉnh thời gian để có hiệu quả tốt nhất.
- Báo cáo kết quả >> Đạt được gì? Khó Khăn? Sẽ hoàn thành gì?
- Đánh giá, phản ánh và lặp lại >> Trình bày kết quả, phản hồi và thời hạn của sprint.

3. ƯỚC LƯỢNG VÀ LẬP KẾ HOẠCH

- Cấu trúc phân rã công việc (WBS- Work Break-down Structure) là phân nhóm các công việc cần thực hiện trong dự án, những công việc này xác định phạm vi tổng thể của dự án
- Đây là tài liệu nền tảng trong quản lý dự án vì nó cung cấp cơ sở để lập kế hoạch và quản lý các lịch biểu, chi phí, và những thay đổi
- Tiếp cận tương tự: xem lại WBS của dự án tương tự và sửa đổi cho phù hợp với dự án hiện hành
- Tiếp cận từ trên xuống (top-bottom): bắt đầu với thành phần lớn nhất và sau đó chi nhỏ dần
- Tiếp cận từ dưới lên (bottom-up): bắt đầu từ công việc chi tiết và kết hợp dần thành công việc lớn
- Tiếp cận Mind-Mapping: ghi ra các công việc dưới dạng phi tuyến và sau đó tạo ra WBS

CÁC KHÁI NIỆM VỀ BẢNG PHÂN CÔNG CÔNG VIỆC

- Bảng công việc (WBS) là gì?
 - Là danh sách những chi tiết những gì cần hoàn thành một dự án.
 - Nếu làm WBS tốt, sẽ xác định được các bước để hoàn thành dự án.
 - WBS là cơ sở để ước lượng chi phí. Từ BCV sẽ có một bức tranh chung về kinh phí dự án.
 - WBS là cơ sở để xác định trách nhiệm giữa các cá nhân.
 - WBS là cơ sở để xây dựng lịch trình thực hiện dự án.
- Ai xây dựng WBS?
 - Người quản lý dự án; Khách hàng; thành viên tổ dự án.
 - Người tài trợ dự án; Ban quản lý dự án.

NHỮNG YẾU TỐ CỦA BẢNG CÔNG VIỆC

- Danh sách các sản phẩm: liệt kê theo dạng phân cấp, sản phẩm chính/phụ
- Ước lượng thời gian: theo từng công việc
- Lịch trình thực hiện: thứ tự thực hiện công việc, ngày bắt đầu, ngày kết thúc từng công việc
- Phân bổ lực lượng, tài nguyên.
- Tính chi phí.
- Kiểm soát rủi ro.

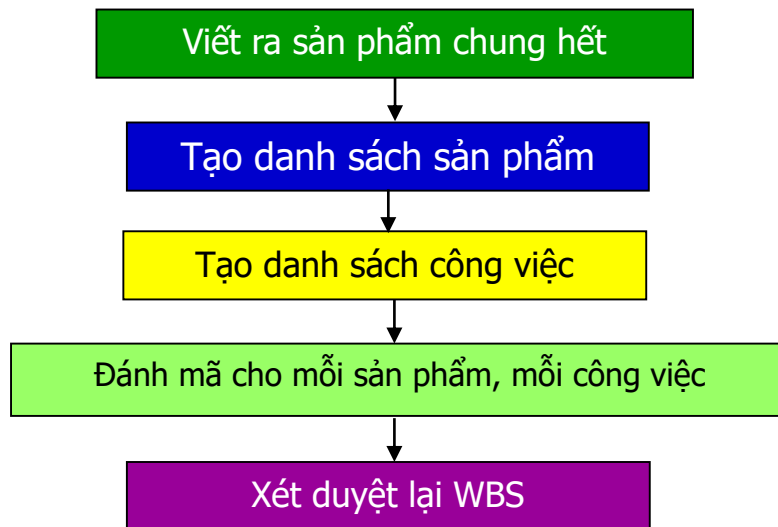
CẤU TRÚC BẢNG CÔNG VIỆC

WBS gồm 2 phần chính:

- Danh sách sản phẩm: DSSP
- Danh sách công việc: DSCV

Cách liệt kê:

- DSSP: mô tả tuần tự từ trên xuống, mức độ phân cấp tùy vào sản phẩm. Tên sản phẩm được mô tả bằng **danh từ**, được đánh mã số.
- DSCV: liệt kê các công việc cụ thể và chi tiết để hoàn thành sản phẩm. Mỗi công việc được mô tả bằng **động từ và một bổ ngữ**, được đánh mã số



4. CHI PHÍ

Chi phí là gì:

- Là những tài nguyên cần phải bỏ ra để đạt đến một mục tiêu cụ thể. Chi phí dùng để trả cho yêu cầu về hàng hóa và dịch vụ.
- Do chi phí của dự án được chi cho việc cung cấp tài nguyên và được sử dụng bất kì nơi đâu trong dự án, nên người quản lí dự án phải hiểu rõ việc quản lí chi phí dự án.
- Những nhà chuyên nghiệp về CNTT biết rằng,
 - Ước lượng chi phí ban đầu cho dự án CNTT thường thấp vì ước lượng dựa trên những yêu cầu chưa đầy đủ và còn mơ hồ.
 - Dự án CNTT thường phát triển những công nghệ mới, cải tiến tiến trình kinh doanh. Bất cứ công nghệ mới, thường chưa được sử dụng nên không kiểm tra trước được (thiếu kinh nghiệm). Vấn đề rủi ro là không thể tránh được. Qua những nhận định trên ta cần quan tâm nhiều đến việc quản lí chi phí.

Quản lý chi phí:

- **Lên kế hoạch tài nguyên:** xác định nguồn tài nguyên nào (con người, thiết bị, vật tư) và số lượng bao nhiêu cho từng công việc của dự án. Liệt kê thành danh sách tài nguyên cụ thể.
- **Ước lượng chi phí:** ước lượng gần đúng cho những tài nguyên cần thiết và tổng hợp cho toàn bộ dự án.
- **Chi phí ngân sách:** đưa ra được làn ranh giới cho từng ngân sách cấp cho từng công việc và đặt ra kế hoạch quản lí.
- **Kiểm tra điều khiển:** điều khiển ngay khi có sự thay đổi trong công việc, có liên quan đến ngân sách, ước lượng lại chi phí để kịp cập nhật, bổ sung và điều phối. Rút ngay những bài học kinh nghiệm.

Phân loại chi phí:

- Trả công lao động (phần lớn)
- Huấn luyện, đào tạo lại.
- Máy móc trang thiết bị làm việc.
- Đi lại, trao đổi.
- Tiện nghi làm việc.
- Văn phòng phẩm
- Thời gian.
- Thu thập thông tin.

Quản trị chi phí dự án:

- Trả công lao động (phần lớn)
 - Huấn luyện, đào tạo lại.
 - Máy móc trang thiết bị làm việc.
 - Đi lại, trao đổi.
 - Tiện nghi làm việc.
 - Văn phòng phẩm
 - Thời gian.
 - Thu thập thông tin.
- Lợi nhuận (profits) là doanh thu trừ chi phí
 - Vòng đời chi phí (Life cycle costing) cho chúng ta nhìn thấy được hình ảnh chi phí của dự án trong quá trình dự án đang thực hiện
 - Phân tích nguồn tiền mặt (cash flow analysis) là phương pháp ước lượng chi phí hàng năm lợi nhuận cho dự án
 - Internal rate of return (IRR) là tỉ giá chiết khấu mà làm cho NPV = 0
 - Chi phí trực tiếp (Direct costs)
 - Chi phí gián tiếp (indirect costs)
 - Chi phí sunk cost (chi phí ẩn) là những chi phí phải bỏ ra trước khi bắt đầu dự án, loại chi phí này không thu lại được
 - Dự trữ (Reserves) là số tiền cần ước lượng để dành vào việc làm giảm rủi ro (rủi ro khó lường trước)

Quy trình quản lý chi phí

- Hoạch định nguồn lực (Resource Planning)
- Ước lượng chi phí (Cost Estimating)
- Dự thảo chi phí (Cost Budgeting)
- Kiểm soát và điều chỉnh chi phí (Cost Control)

Hoạch định nguồn nhân lực

- Xác định nguồn tài nguyên hữu hình (con người, thiết bị, vật liệu ...) cần thiết để hoàn thành dự án.
- Kế hoạch tài nguyên sẽ phụ thuộc vào bản chất riêng của từng dự án và tổ chức thực hiện dự án đó.
- Vấn đề quan trọng là phải có người có kinh nghiệm, họ đã từng tham gia thực hiện những dự án tương tự,
- Tổ chức phải hỗ trợ việc xác định những nguồn tài nguyên gì là cần thiết
- Để xây dựng được kế hoạch tài nguyên cần cân nhắc như:
 - Những công việc cụ thể nào trên dự án sẽ gặp khó khăn và khó khăn ?
 - Có những phạm vi nào của dự án ảnh hưởng đến tài nguyên?
 - Tổ chức đã thực hiện dự án nào trước đây tương tự như dự án đang thực hiện ? Kỹ năng chuyên môn cá nhân của những người tham gia thực hiện ? có đáp ứng được ?
 - Tổ chức có đủ người, thiết bị, vật tư để thực hiện dự án ?
 - Tổ chức có cần yêu cầu thêm tài nguyên ? (có thể những tài nguyên này phải thuê mượn bên ngoài)
 - Có những chính sách của tổ chức ảnh hưởng đến những tài nguyên cần thiết ?
- Để giải quyết những vấn đề trên cần tham khảo đến: WBS, scope statement, thông tin trước đây và hiện tại, chính sách của tổ chức (policies)...
- Kết quả cuối cùng là phải đưa ra danh sách những tài nguyên được yêu cầu (resource requirements)

Ước lượng chi phí

- Từ danh sách tài nguyên cần thiết, ban quản lý dự án ước lượng chi phí cho những tài nguyên này.
- Phương pháp và công cụ kỹ thuật để ước lượng chi phí dự án
 - Ước lượng phỏng đoán
 - Ước lượng theo ngân sách
 - Ước lượng sau cùng
 - Ước lượng tương tự hay ước lượng từ trên xuống (Analogous estimating/top-down estimating)
 - Ước lượng từ dưới lên (Bottom-up estimating)
 - Mô hình tham số (Parametric modeling)
 - Computerized tools: như bảng tính và một số chương trình quản lý có thể phối hợp để ước lượng chi phí.
 - Function Point Analysis (FPA), Use Case Point Analysis (UCP)

Ước lượng phỏng đoán

- Thường được dùng trong bước đầu để chọn lựa thực hiện dự án.
- Hỗ trợ cho người quản lý dự án và cấp cao hơn đưa ra quyết định và nó thường được ước lượng trước khi dự án thực hiện từ 3 năm hay nhiều hơn.
- Kết quả chi phí này so với chi phí thật là dưới 25% hay trên 75%. Có trường hợp những nhà chuyên nghiệp ước lượng chi phí gấp đôi do lịch sử ước lượng chi phí những dự án CNTT thường tăng cao.

Ước lượng sau cùng

- Có độ chính xác cao hơn so với những cách trên
- Thường được dùng trong giai đoạn dự án được thực hiện và do có được những thông tin đầy đủ hơn .
- Ví dụ nếu dự án cần đặt mua 1000 máy vi tính cá nhân từ những nhà cung cấp bên ngoài trong thời gian 3 tháng tới, khi đó, đã chọn được nhà cung cấp và biết được chi phí phải trả. Thường thì ước lượng này có thể thực hiện được trong vòng một năm trước khi dự án hoàn tất. Tuy nhiên độ chính xác cũng chỉ là dưới 5% hay trên 10%, bởi vì còn yếu tố rủi ro mà ta sẽ bàn đến trong chương sau.

Ước lượng từ trên xuống

- Sử dụng con số chi phí thực sự của một dự án đã thực hiện trước đây mà tương tự như dự án cần ước lượng, và xem đó như lời góp ý của chuyên gia.
- Thường cho kết quả ước lượng thấp hơn những phương pháp khác và độ chính xác cũng kém hơn .
- Tuy nhiên, nó có độ tin cậy và thuyết phục hơn, thêm vào đó, khi ước lượng có thể chia dự án ra từng phần và so sánh những phần đó với dự án tương tự, sau đó tăng thêm hay giảm đi.

→ Phương pháp này không hiệu quả khi ước lượng các sản phẩm như phần mềm, công nghệ, thiết bị mới...

Ước lượng từ trên xuống

- Chia nhỏ công việc (WBS) để ước lượng
- Ở những công việc nhỏ có thể do nhóm nhỏ hay cá nhân thực hiện, nhóm hay cá nhân sẽ ước lượng chi phí , sau đó tổng hợp những bước nhỏ này thành những công việc lớn rồi toàn bộ dự án

→ Phương pháp này chính xác hơn, nhưng tốn công sức và chi phí ước lượng thường cao hơn thực tế.

Mô hình tham số:

- Dùng đặc tính của dự án (tham số) đặt trong một mô hình toán học để ước lượng chi phí dự án.
- Mô hình COCOMO (Constructive Cost Model, Barry Boehm): mô hình tham số thông dụng, để ước lượng chi phí cho các dự án phát triển phần mềm, dựa trên các tham số như số dòng lệnh, số lượng nhập xuất dữ liệu, số tập tin phải bảo trì, số tập tin phải cập nhật.
- Hiện nay có Mô hình cải tiến COCOMO II

Công thức cocomo

- Tác giả Barry Boehm
- $E = aKLOC^b + EAF$
- $D = cE^d$
- Hệ số EAF (effort adjustment factor) thay đổi từ 0.9 -1.4 tùy ngữ cảnh dự án

Loại dự án	Hệ số a	Hệ số b	Hệ số c	Hệ số d
Nhỏ	2.4	1.05	2.5	0.38
Vừa	3.0	1.12	2.5	0.35
Lớn	3.6	1.2	2.5	0.32

Kỹ thuật ước tính chi phí cho phần mềm

- Function Point Analysis (FPA)
 - Ước tính độ lớn phần mềm (software size) dựa trên số lượng chức năng phân phối yêu cầu chức năng của người dùng
 - Tính dựa trên data Input, output, enquiry, transaction,...
- Use case Point Analysis (UCP)

Type of Component	Complexity of Components			Total
	Low	Average	High	
External Inputs	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
External Outputs	___ x 4 = ___	___ x 5 = ___	___ x 7 = ___	
External Inquiries	___ x 3 = ___	___ x 4 = ___	___ x 6 = ___	
Internal Logical Files	___ x 7 = ___	___ x 10 = ___	___ x 15 = ___	
External Interface Files	___ x 5 = ___	___ x 7 = ___	___ x 10 = ___	
Total Number of Unadjusted Function Points				=====

Dự thảo chi phí (cost budgeting)

- Phân bổ chi phí toàn bộ (được ước tính ở giai đoạn 4.2) vào từng công việc cụ thể để thiết lập một đường ranh giới về chi phí (cost baseline) giúp cho việc đánh giá các hoạt động của dự án
- Sau đây là bảng dự toán ngân sách cho dự án Business Systems Replacement cho FY97

Budget Category	Estimated Codes
Headcount (employee)	13
Compensation	\$1.008.500
Consultant/Purchase service	\$424.500
Travel	\$25.000
Depreciation (trượt giá)	\$91.000
Rents/Leases	\$98.000
Other Supplies & Expense	\$153.000
Total Costs	\$1.800.000

Kiểm soát và điều chỉnh chi phí

- Giám sát việc thực thi về chi phí để phát hiện những điều khác biệt với kế hoạch
- Bảo đảm rằng tất cả những thay đổi hợp lý đều phải được ghi nhận vào cost baseline một cách chính xác
- Ngăn chặn những thay đổi không đúng, không thích hợp và những thay đổi trái phép (unauthorized)
- Thông báo sự thay đổi đến những người liên quan có thẩm quyền

Một số khái niệm dùng trong kỹ thuật evm

- The Planned Value (PV), trước đây gọi là Budgets Cost Work Scheduled (BCWS): là chi phí để thực hiện hoàn tất một công việc theo kế hoạch.
- The Actual Cost (AC) hay Actual Cost of Work Performed (ACWP): là chi phí thực để hoàn thành công việc.
- The Earned Value (EV) hay Budgeted Cost of Work Performed (BCWP): là phỏng đoán giá trị của công việc thực sự hoàn thành, được tính bằng phần trăm công việc thực sự hoàn thành nhân với chi phí theo kế hoạch.

Khái niệm	Công thức
Giá trị thu được (EV)	$EV = PV \times X\%$ Ngày hoàn thành
Chi phí phát sinh (CV = Cost Variance)	$CV = EV - AC$
Biến động Lịch (SV = Schedule Variance)	$SV = EV - PV$
Chỉ số thực hiện chi phí (CPI =Cost Performance Index)	$CPI = EV/AC$
Chỉ số thực hiện lịch (SPI =Schedule performance index)	$SPI = EV/PV$
Ước tính tại thời điểm hoàn tất (EAC =Estimate at completion)	$EAC = BAC/CPI$
Ước tính thời gian hoàn tất (Estimate time to complete)	Ước tính thời gian ban đầu/SPI

NHẮC LẠI CÁC KHÁI NIỆM CƠ BẢN

User Story là gì?

User Story là một bản tóm tắt nhu cầu người dùng. Thông thường, user story do khách hàng, hoặc đại diện của khách hàng, người thực sự hiểu nghiệp vụ và nắm bắt được chính xác yêu cầu của mình đối với nhóm phát triển. Story không đơn thuần là công cụ requirement (Cohn), mà còn là công cụ để giao tiếp, chốt kết dính và cái “phanh hãm” trong phát triển. Scrum quy định Product Owner sở hữu các story (thông qua product backlog), nhưng đó không phải công việc đơn thuần của Product Owner (ví dụ như ở một vài cách làm khác, “BA làm requirement” v.v.).

User Story Point là gì?

Đó là đại lượng chỉ độ lớn tương đối của các user story trong cùng một dự án. Trong một phiên hoạch định trước Sprint, nhóm phát triển dùng Scrum Poker để đánh giá độ lớn bé các story này, và ghi các giá trị đó lên mỗi user story card.

Agile Estimation là gì?

Là cách thức ước lượng độ lớn của story theo cách linh hoạt. Sử dụng Scrum Poker, nhóm sẽ đánh giá các story dựa theo sự so sánh với các story mẫu (là các story dễ hiểu đối với nhóm, gán giá trị khởi đầu để làm “mốc” đánh giá cho các story khác).

Trước khi Sprint 1 diễn ra, Nhóm Scrum cộng tác trong buổi họp Kế hoạch Phát hành (Release Planning) để xác định những tính năng nào sẽ có trong bản phát hành, thời điểm nào sẽ phát hành sản phẩm. Khi đó nhóm sẽ phải ước tính cho tất cả các story được xác định tham gia vào release tới.

Velocity là gì?

Là tốc độ burn được bao nhiêu điểm (point) trong một Sprint. Ví dụ Sprint 1 nhóm burn được 45 point, Sprint 2 được 51, Sprint 3 được 48 thì tốc độ trung bình được tính:

$$V = (45+51+48) / 3 = 48.$$

Giả sử mọi thứ không đổi, một release được ước tính ban đầu có độ lớn 480 point thì nhóm phải trải qua khoảng $480/48 = 10$ Sprint.

Lưu ý: velocity chỉ có giá trị tương đối, hỗ trợ việc ước tính, giá trị tuyệt đối của nó không có ý nghĩa gì. Cấp quản lý về cơ bản không thể căn cứ vào velocity của nhóm từ Sprint trước để “ép tiến độ”, nếu chưa tính tới các yếu tố khác như focus factor, sự biến động về nhóm, sự thay đổi về công nghệ v.v.

Focus factor là gì?

Focus factor là tỉ lệ thời gian sản xuất thực tế của nhóm dành cho các story (sau khi trừ đi các thời gian họp hành, học tập, giải lao, ốm đau v.v.).

Ví dụ một ngày làm việc 8 tiếng, có 15 phút họp chính thức, 45 phút thảo luận về design, 30 phút đọc sách kĩ thuật, 30 phút trao đổi về các yêu cầu, 30 phút commit code lên repository, 30 phút viết log dự án; thời gian còn lại là làm việc trên các story (design, test, code) thì hệ số tập trung có thể là:

$$FF = 1.0 - (15+45+30+30+30+30)/8*60 = 62.5 \%$$

Một nhóm càng ít mature (nhóm mới, nhóm “ô hợp”, hoặc va phải công nghệ lạ lắm v.v.) thì hệ số tập trung càng thấp. Cần xác định được hệ số tập trung thì mới biết được capacity thực tế của nhóm từ đó ước tính được tốc độ thực tế của nhóm. Nhiều người chỉ đặt FF ở mức 50% (Kniberg) ngay cả khi nhóm đã tương đối mature. Theo quan sát của riêng cá nhân tôi (không có dữ liệu đầy đủ), các nhóm ở Hà Nội thường phải chịu một ff lớn là khoảng $7/8 = 87.5\%$ (ngày làm việc 8 tiếng thì chịu sức ép sản xuất 7 tiếng; đây có thể là nguyên nhân dẫn đến tình trạng overtime phổ biến hiện nay).

ƯỚC LƯỢNG CHI PHÍ

Các bước tính chi phí

Công thức để tính chi phí như sau:

$$\text{Chi phí} = \text{REP} / \text{PM} / \text{FF}$$

$$\text{Thời gian phát hành} = \text{REP} / \text{EV} \text{ (số Sprint)}$$

Trong đó:

REP: Release Estimated Points = Số point ước tính của release

PM: Point – Man = quy đổi 1 point tương ứng man-day

EV: Estimated Velocity = Tốc độ ước tính

FF: Focus Factor = Hệ số tập trung

Một quy trình ước tính chi phí cơ bản sẽ trải qua các bước sau đây:

Xác định focus factor > Xác định estimated velocity > Xác định độ quan trọng và cam kết release > Ước tính chi phí

Các chi tiết của từng bước được thảo luận kĩ hơn ở bên dưới.

Xác định focus factor

Dựa vào dữ liệu thực tế (nếu nhóm đã có sự cộng tác trước đó), tính chất của dự án, năng lực hiện có của nhóm và các tham số khác để xác định focus factor. Nếu có ít thông tin, có thể lựa chọn con số an toàn là 50%, sau đó làm mịn lại ở Sprint tiếp theo.

Số liệu FF sẽ ảnh hưởng đến capacity như thế nào?

Giả sử FF = 50%. Nhóm bạn có tổng cộng 9 developer, làm việc 5 ngày/1 tuần, Sprint 2 tuần. Vậy là bạn có $9 \times 5 \times 2 = 90$ man-day. Nhưng FF=50% nên chỉ dùng có 45 man-day cho sản xuất, còn lại là các việc hành chính, học tập, giải trí v.v. Capacity thực sự để tính tốc độ là 45 man-day.

Xác định estimated velocity (EV)

Có một số tình huống cho việc ước tính velocity như sau:

Tình huống 1: Dự án đã chạy được một số Sprint (qua quá trình pilot, hoặc chạy thật):

Chỉ cần đếm và đo tốc độ trung bình. Các dự án inhouse, RnD có thể rơi vào tình huống này. Dễ.

Tình huống 2: Dự án mới, cần ước tính velocity (để tính được chi phí)

Cách 1: chạy pilot (hoặc calibration – tùy cách bạn gọi) một Sprint hoặc mini-Sprint (độ dài rút ngắn xuống 1 tuần hoặc ít hơn) để có dữ liệu. Cách này luôn luôn thực hiện được. Dữ liệu empirical luôn là dữ liệu thật nhất. Dĩ nhiên là bạn phải phân tích kĩ các dữ liệu đo đếm được trước khi ra quyết định cuối cùng (Count>Calculate>Judge).

Cách 2: phân tích dữ liệu lịch sử. Nếu dự án mới không quá khác so với các dự án trước đó, bạn có thể lấy dữ liệu cũ để dùng cho dữ liệu mới. Nếu dự án mới tinh, nhóm mới tinh, bạn không thể dùng được cách này.

Giả sử trước đó bạn không dùng story point để ước lượng, bạn sẽ phải quy đổi từ đơn vị cũ sang đơn vị mới. Ví dụ, trước đó chức năng “Login” được thực hiện với 5 man-day, giờ đây bạn xác định story “Login” là 1 point thì có quy đổi 1 point = 5 man-day. Nếu bạn chuyển từ waterfall sang agile, bạn có thể thực hiện quá trình calibration để biết được giá trị quy đổi thực sự. Cách làm là: chạy một mini-Sprint để pilot, đo và quy đổi (cách 1).

Còn nếu trước đó bạn đã dùng point để đo thì không có gì để bạn, biết rồi!

Xác định độ quan trọng và xác định cam kết

Tới đây bạn đã có: FF, Capacity, EV, quy đổi Point-Man_day (PM). Cần phải xác định thêm tổng Story point cần burn để có được ước tính man-day cho một release.

Làm theo cách của Scrum: Dựa theo tầm quan trọng của story, Nhóm Scrum (PO, SM, DevTeam) quyết định trong release tới có bao nhiêu story. Cộng gộp các story point tương ứng với mỗi Story lại sẽ có độ lớn của dự án (tính tới release đó). Gọi giá trị này là REP (release-estimated-point).

Ước tính chi phí (theo man-day) và thời gian phát hành

Chi phí = REP /PM/FF

Thời gian phát hành = REP /EV (số Sprint)

Ví dụ: Nhóm 9 người với FF là 50%, tốc độ ước tính là 50 point/Sprint_2_tuần, quy đổi PM=5 (tức 1 point tương ứng 5 man-day), release 1.0 tới cần 20 story với tổng cộng 200 point (REP = 500) thì:

Chi phí = $500/5/0.5 = 200$ man-day.

Thời gian = $500/50 = 10$ Sprint = 5 tháng.

Vậy là theo ước tính này, dự án sẽ cán đích release 1.0 sau 5 tháng với chi phí là 200 man-day.

Nếu bạn chỉ cho mỗi 1 man-day là 50\$/ngày công (bao gồm mọi chi phí tiền lương, máy móc, phụ cấp v.v.) thì chi phí ước tính cho dự án là $200 \times 25 = 5000\$$.

5. PHÂN CÔNG THỰC HIỆN

STT	Thành viên	Công việc	Chi tiết công việc
1	Nguyễn Ngọc Thành (thiết kế quy trình)	Thiết kế quy trình nghiệp vụ cho cả Sprint 1 và Sprint 2,3	<p>Tìm hiểu công việc cần làm, tìm hiểu quy trình nghiệp vụ trên các phần mềm hiện có (tương tự), quy trình nghiệp vụ thực tế, tham gia thảo luận cùng nhóm trưởng, đưa ra một quy trình gần đúng với thực tế nhất để nhóm bắt đầu làm việc.</p> <p>Trình bày nghiệp vụ lại với các thành viên, đặc biệt là bạn thiết kế cơ sở dữ liệu và thiết kế giao diện để bạn có thể thiết kế dữ liệu/ giao diện tốt nhất.</p>

2	Phạm Hồng Sơn (thiết kế csdl)	Thiết kế cơ sở dữ liệu cả Sprint 1 và Sprint 2,3	Nhận quy trình từ thành viên sau đó nhận thiết kế cơ sở dữ liệu. Tìm hiểu framework ERP để thiết kế cơ sở dữ
3	Đoàn Minh Tài (thiết kế giao diện)	Thiết kế giao diện cả Sprint 1 và Sprint 2,3 Kiểm lỗi chức năng	Nhận quy trình sao đó tìm hiểu lập trình view cho các chức năng, sao đó trình bày phối hợp với bạn kết nối cơ sở dữ liệu. Khi các thành viên làm đến đâu thì tham gia kiểm lỗi phần mềm.
4	Tô Thế Tân (thiết kế nghiệp vụ & kết nối cơ sở dữ liệu)	Thiết kế quy trình tổ chức kết nối cơ sở dữ liệu cho cả Sprint 1 và Sprint 2,3	Tham gia thiết kế quy trình cùng các thành viên. Lập trình kết nối cơ sở dữ liệu. Lập trình ở mức view. Tạo các cơ sở dữ liệu mà các bạn có thể làm chưa đúng.
5	Nguyễn Văn Thiện (lập trình xử lý)	Phát triển hoàn thiện chức năng cho các Sprint Fix bug các Sprint	Tham gia thiết kế quy trình. Nhận giao diện từ các thành viên sau đó lập trình hoàn thiện chức năng. Lập trình xử lý để các chức năng hoàn thiện.
6	Phạm Thanh Phú (nhóm trưởng, kiểm thử)	Phân chia công việc Thiết kế quy trình nghiệp vụ Kiểm lỗi chức năng cho các Sprint 1, 2, 3	Phân công công việc, nhận việc thông báo lại với các thành viên. Kiểm tra tiến độ công việc Tham gia thiết kế quy trình/ nghiệp vụ. Kiểm thử phần mềm

6. QUY TRÌNH THỰC HIỆN MỘT CHỨC NĂNG CỤ THỂ TRONG DỰ ÁN

+ Phân tích yêu cầu

+ Xác định chắc năng

+ Thiết kế dữ liệu

+ Hiện thực chức năng

- Tạo table
- Tạo store procedure
- Tạo service
- Update service
- Code
- Khai báo trên phần mềm chức năng hiển thị

+ Kiểm thử