

PHÂN TÍCH YÊU CẦU PHẦN MỀM

SẮP XẾP YÊU CẦU THEO THỨ TỰ ƯU TIÊN

Nguyễn Thị Thu Hương
BM CNPM - Khoa CNTT
Email: huongnt@tlu.edu.vn

NỘI DUNG

- Kiểm tra và kiểm chứng
- Sắp xếp yêu cầu theo thứ tự ưu tiên

Kiểm tra và Kiểm chứng (Verification and Validation)

- Các khái niệm cơ bản
- Kiểm chứng yêu cầu
- Kiểm tra các yêu cầu

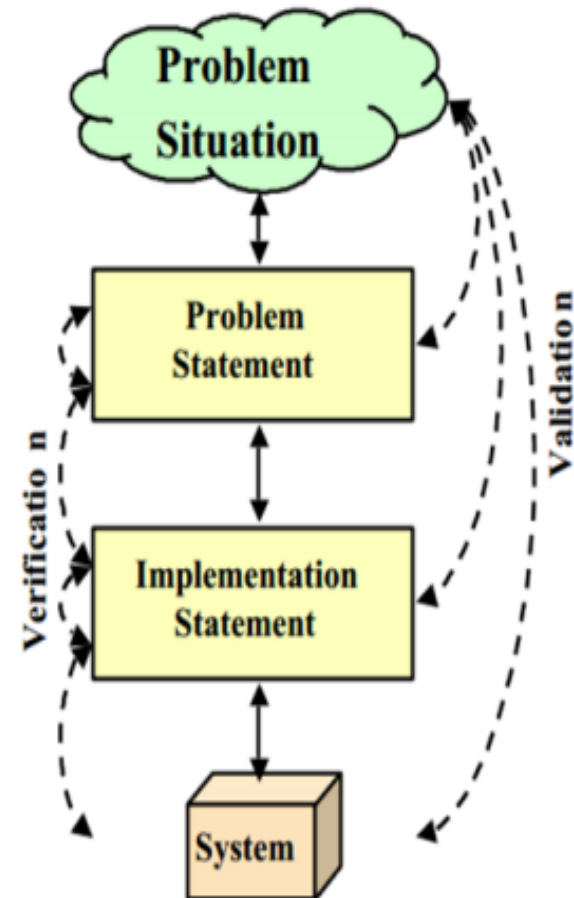
Kiểm tra và kiểm chứng

Kiểm chứng:

- Chúng ta đã xây dựng đúng hệ thống?
- Các vấn đề đặt ra đã nắm bắt được các thuộc tính lĩnh vực?
- Hệ thống đã đáp ứng được các yêu cầu từ các nhân tố tham gia?

Kiểm tra:

- Chúng ta đã xây dựng hệ thống đúng?
- Thiết kế đáp ứng được đặc tả?
- Cài đặt đáp ứng được đặc tả?
- Các mô hình yêu cầu thống nhất với các mô hình khác?



Tiêu chuẩn kiểm tra và kiểm chứng



Sự khác biệt:

D: Những điều luôn luôn đúng trong lĩnh vực ứng dụng.

R: Những điều chúng ta mong là đúng trong lĩnh vực ứng dụng.

S: Sự mô tả các hành vi mà chương trình cần thực hiện để đáp ứng với các yêu cầu

Hai tiêu chuẩn kiểm tra

- Chương trình (P) thực hiện trên một máy tính (C) cụ thể đáp ứng với đặc tả (S)
- Đặc tả (S) được cho trong thuộc tính của lĩnh vực (D) thỏa mãn các yêu cầu (R)

Hai tiêu chuẩn kiểm chứng

- Chúng ta đã xem xét (và hiểu) tất cả các yêu cầu (R) quan trọng?
- Chúng ta đã xem xét (và hiểu) tất cả các thuộc tính lĩnh vực (D) liên quan.

Ví dụ: Kiểm tra và kiểm chứng

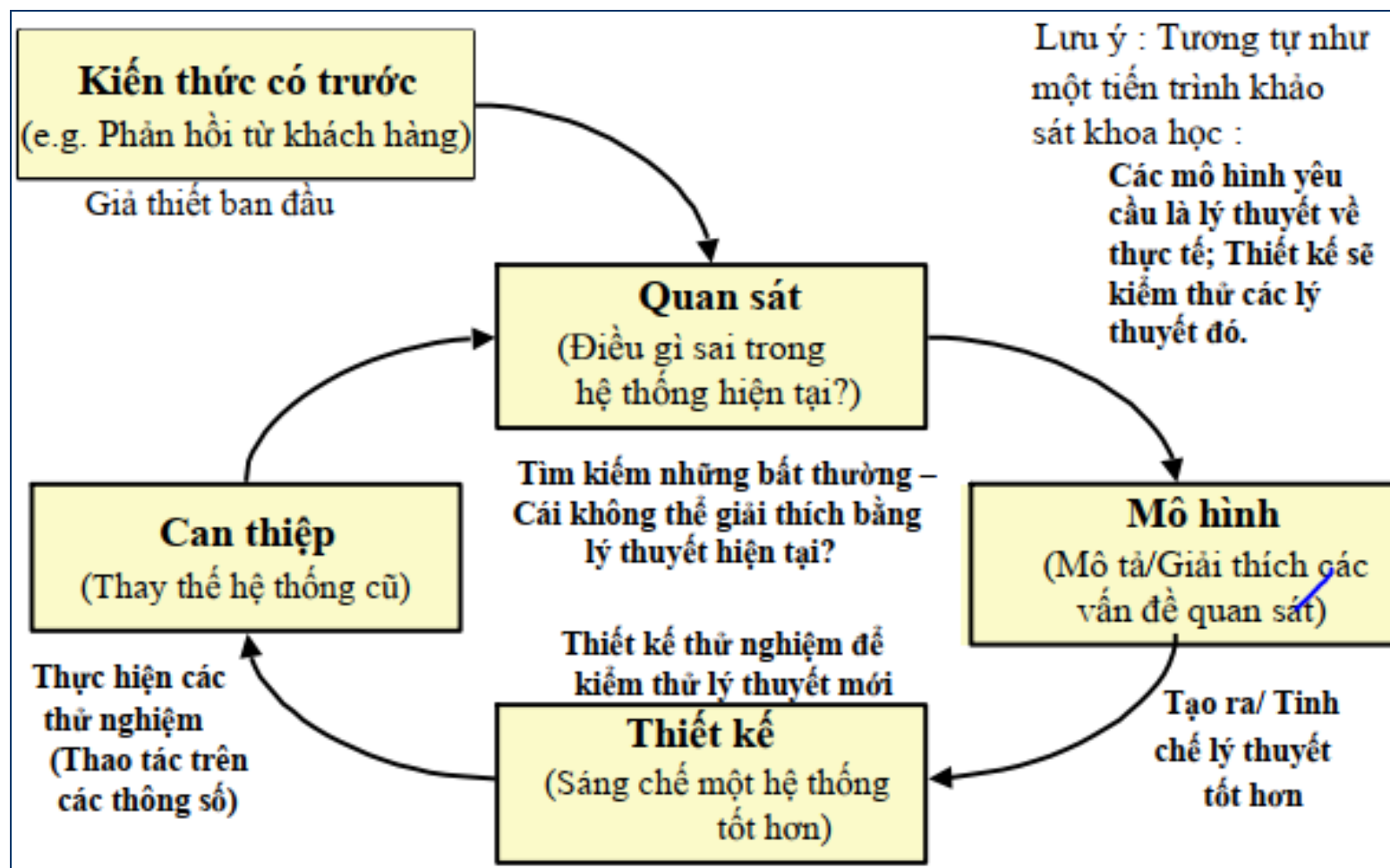
- **Requirement R:** “Phản lực chỉ có thể xảy ra khi máy bay đang chạy trên đường băng”
- **Domain Properties D:**
 - ✓ Xung lực bánh xe xảy ra khi và chỉ khi các bánh xe bật ra
 - ✓ Các bánh xe bật ra khi và chỉ khi nó chạy trên đường băng
- **Specification S:** Phản lực có thể xảy ra khi và chỉ khi có xung lực bánh xe
- **Kiểm tra:**
 - ✓ Phần mềm cho máy bay (P) thực thi trên máy tính trong buồng lái của máy bay (C) có hoàn toàn chính xác như đặc tả (S)?
 - ✓ Đặc tả (S) trong ngữ cảnh của giả thuyết (D) có đáp ứng yêu cầu (R)?
- **Kiểm chứng:**
 - ✓ Giả thuyết của chúng ta (D) về lĩnh vực có thật chính xác? Có thiếu sót gì không?
 - ✓ Yêu cầu (R) có thật sự cần thiết? Có thiếu sót gì không?

Đặc điểm phần mềm

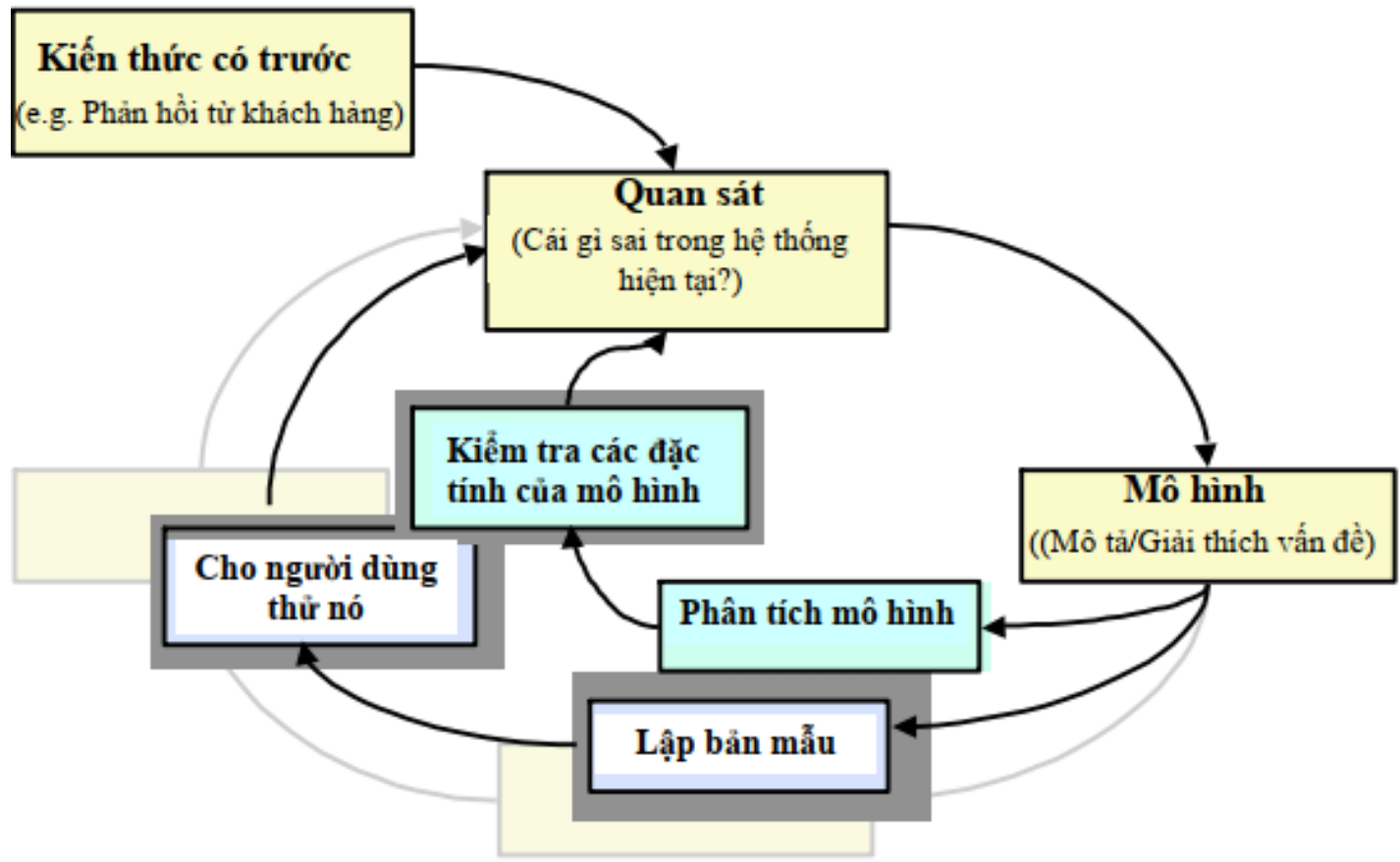
- Vô hình, mơ hồ, trừu tượng
- Không có quy luật tự nhiên nào bên trong
- Các hành vi phần mềm
- Không có các ràng buộc tự nhiên nào trong các phần mềm phức tạp
- Phần mềm hoàn toàn có thể thực hiện một công việc lặp đi lặp lại

→ *Khi phần mềm lỗi thì các việc cần làm là gì?*

Chu trình điều tra



Chu trình điều tra nhanh



Các kỹ thuật kiểm chứng

- Lập bản mẫu (Prototyping)
- Phân tích mô hình (Model Analysis)
- Kiểm duyệt (Inspection)

Lập bảng mẫu

- “Một bản mẫu phần mềm là một kiến trúc được cài đặt cụ thể trước để các khách hàng, người dùng hoặc nhà phát triển có thể hiểu rõ thêm về một vấn đề hay giải pháp của nó.”

[Davis 1990]

- “Lập bản mẫu là tiến trình xây dựng mô hình làm việc của hệ thống”.

[Agresti 1986]

Hướng tiếp cận lập các bản mẫu

- Bản mẫu trình diễn
 - Dùng để chứng minh khái niệm; giải thích các đặc tính thiết kế; etc.
 - Giải thích, minh họa và thông báo – sau đó bỏ đi
- Bản mẫu thăm dò
 - Dùng để xác định vấn đề, thu thập nhu cầu, làm rõ mục tiêu, so sánh các lựa chọn thiết kế
 - Không hình thức, không cấu trúc và cũng được bỏ đi.
- Bản mẫu thử nghiệm
 - Khai thác các đặc tính kỹ thuật; kiểm tra sự thích hợp của một kỹ thuật
 - Thông thường không bao gồm người dùng/khách hàng
- Bản mẫu tiến triển (Ví dụ: “bản mẫu vận hành”, “hệ thống lái máy bay”):
 - Được phát triển khi thấy tiến trình tiếp diễn sẽ tương thích với hệ thống
 - “Bản mẫu (Prototype)” như là một phân phối sớm, để tiếp tục cải tiến

Các loại bản mẫu

- Bản mẫu dùng thử: Là bản mẫu sẽ bị bỏ đi khi thu được các kiến thức mong muốn.
- Bản mẫu tiến hoá: Là bản mẫu được phát triển dần trong quá trình xây dựng hệ thống.

Phân tích mô hình

- Kiểm tra:
 - “Mô hình có định dạng chuẩn?”
 - Các thành phần trong mô hình thống nhất với những cái khác?
- Kiểm chứng:
 - Dựng hoạt cảnh của mô hình trên các ví dụ nhỏ
 - Các thay đổi hình thức
 - Các câu hỏi ‘Cái gì sẽ xảy ra nếu ...?’ (‘What if’):
 - Nguyên nhân về hậu quả của những yêu cầu ngoại lệ;
 - Nguyên nhân về sự tác động của những thay đổi có thể
 - “Hệ thống có khi nào thực hiện như sau ...”
 - Khảo sát trạng thái
 - Ví dụ: sử dụng kiểm tra mô hình để phát hiện ra sự đáp ứng của một số đặc tính

Kiểm duyệt

- Kiểm duyệt đã được chứng minh là có hiệu quả trong việc phát hiện các lỗi của phần mềm.
- Kiểm duyệt khác các kỹ thuật kiểm chứng ở cấu trúc của nó

Cấu trúc của kiểm duyệt

Checklist

- ✓ Dùng 1 danh sách các câu hỏi kiểm tra/vấn đề

Walkthrough

- ✓ Một người phải có mặt trong từng bước kiểm tra sản phẩm

Round Robin

- ✓ Mỗi reviewer lần lượt nêu ra một vấn đề

Tốc độ Review

- ✓ Mỗi reviewer có 3 phút để xem xét lại 1 vấn đề, sau đó chuyển sang cho người kế tiếp
- ✓ Tốt cho việc đánh giá khả năng hiểu thấu vấn đề!

Kiểm tra và kiểm chứng độc lập

- Kiểm tra và kiểm chứng được thực hiện bởi một nhà thầu riêng
- ✓ Để đáp ứng kiểm tra và kiểm chứng độc lập thì cần một quan điểm kỹ thuật độc lập.
- ✓ Chi phí khoảng giữa 5% đến 15% của chi phí phát triển
- ✓ Các khảo sát chỉ ra tăng gấp 5 lần lợi nhuận trên vốn đầu tư:
 - Lỗi phát hiện sớm, rẻ hơn để sửa, rẻ hơn để thử lại
 - Đặc tả rõ hơn
 - Các nhà phát triển thích dùng các thực nghiệm tốt hơn

Kiểm tra và kiểm chứng độc lập (tiếp)

- **3 kiểu của sự độc lập:**

Độc lập quản lý:

- Phân chia trách nhiệm từ việc phát triển các phần mềm
- Có thể quyết định khi nào và ở đâu cần tập trung nỗ lực kiểm tra và kiểm chứng

Độc lập tài chính:

- Phân chia chi phí và nguồn tài chính
- Không mạo hiểm phân phối nguồn tài nguyên khi sắp sửa gặp khó khăn

Độc lập kỹ thuật:

- Tổ chức nhân sự riêng biệt, để tránh thành kiến của các nhà phân tích
- Dùng những công cụ và kỹ thuật khác nhau

Các kỹ thuật kiểm tra

- Thực hiện lưu vết đặc tả
- Kiểm thử
- Kiểm duyệt mã lệnh
- Phân tích mã lệnh

Kết luận: Kiểm tra và kiểm chứng

Kiểm chứng (Validation) để xác định đã giải quyết đúng vấn đề.

- ✓ Lập bản mẫu – cho phản hồi của khách hàng sớm
- ✓ Kiểm duyệt – các chuyên gia lĩnh vực đọc đặc tả một cách kỹ lưỡng
- ✓ Phân tích mô hình – phân tích toán học các mô hình của bạn

...cộng thêm các buổi họp & giao tiếp thường xuyên với các đối tác

Kiểm tra (Verification) các bước trong công nghệ là đúng

- ✓ Kiểm tra tính nhất quán – các mô hình có thống nhất với cái khác?
- ✓ Khả năng lưu vết – làm cho việc thiết kế/viết mã lệnh/kiểm thử phản ánh đúng các yêu cầu?

Kết luận: Kiểm tra và kiểm chứng

Sử dụng kiểm tra và kiểm chứng thích hợp :

- Lấy thông tin phản hồi từ khách hàng sớm nếu mô hình mới chỉ là phác thảo.
- Phân tích và kiểm tra tính nhất quán nếu mô hình đã là bản đặc tả
- Kiểm tra và kiểm chứng độc lập nếu hệ thống cần an toàn nghiêm ngặt

Sắp xếp yêu cầu ưu tiên

Requirements Prioritization

Sắp xếp yêu cầu ưu tiên

Tại sao cần sắp xếp thứ tự ưu tiên yêu cầu

- Cần cân bằng các yêu cầu khác nhau để đạt được sự kết hợp tốt nhất

Định hướng theo Chi phí –Giá trị (Cost-Value Approach)

- Sắp xếp các yêu cầu theo tiêu chí Chi phí/Giá trị
- Đánh giá quan hệ Chi phí/Giá trị (dùng AHP)

Cơ sở của sự ưu tiên

Cái gì cần được chọn để cài đặt

- Khách hàng (thường) hỏi quá nhiều về cách thức
- Cân đối giữa thời gian tiếp thị với tổng số các chức năng
- Quyết định đặc tính nào sẽ được phát hành kế tiếp

Đối với mỗi yêu cầu/đặc tính, cần hỏi:

- Nó quan trọng thế nào với khách hàng?
- Chi phí để cài đặt nó là bao nhiêu ?
- Sẽ có rủi ro nào khi cố gắng thực hiện nó?

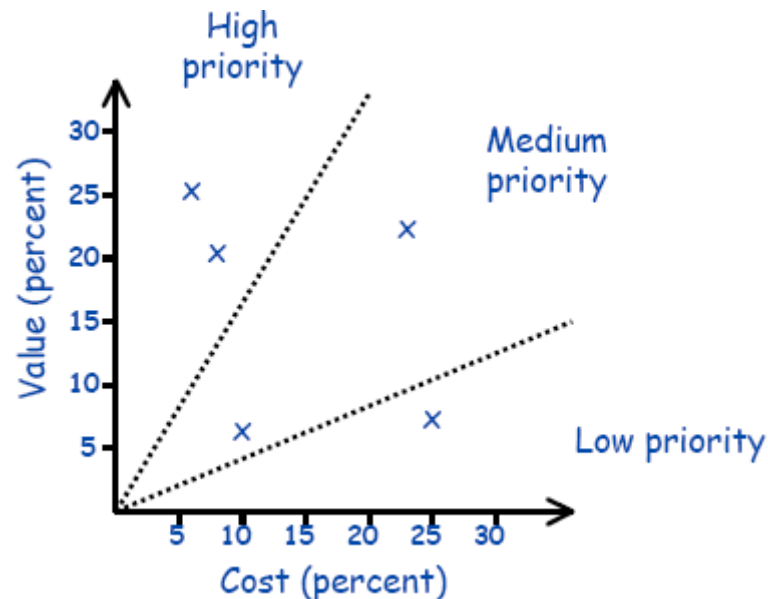
Thực thi khẩn cấp:

- Một số yêu cầu bắt buộc “phải”
 - Một số yêu cầu nên dứt khoát loại bỏ
- Điều này sẽ dẫn đến một phạm vi các “yêu cầu hợp lý”

Tiếp cận theo Chi phí/Giá trị (Cost-Value)

Tính toán lợi nhuận trên vốn đầu tư

- Đánh giá tổng thể tầm quan trọng của mỗi yêu cầu đối với dự án
- Ước lượng quan hệ về chi phí của mỗi yêu cầu
- Tính toán sự thỏa hiệp giữa chi phí và giá trị



Ước lượng Chi phí/Giá trị

➤ Hai cách tiếp cận

- Định mức tuyệt đối: Đòi hỏi phải có kinh nghiệm chuyên môn
- Các giá trị liên quan: Dễ dàng để làm rõ hơn;

Sắp thứ tự ưu tiên dựa trên sự sắp xếp các vấn đề

➤ Quá trình so sánh – chọn lựa

- Cơ sở để sắp xếp – với mỗi cặp yêu cầu (i,j) , xét $i > j$? Chẳng hạn bubblesort – bắt đầu với thứ tự ngẫu nhiên và hoán đổi mỗi cặp nếu sai thứ tự cần $n \cdot (n-1)/2$ bước so sánh
- Dựng Cây thứ tự nhị phân (Binary Sort Tree): Cần $O(n \log n)$ bước so sánh
- Dựng cây phủ tối thiểu (Minimal Spanning Tree): Với mỗi cặp (R_i, R_{i+1}) : tính khoảng cách giữa chúng, cần $n-1$ bước so sánh

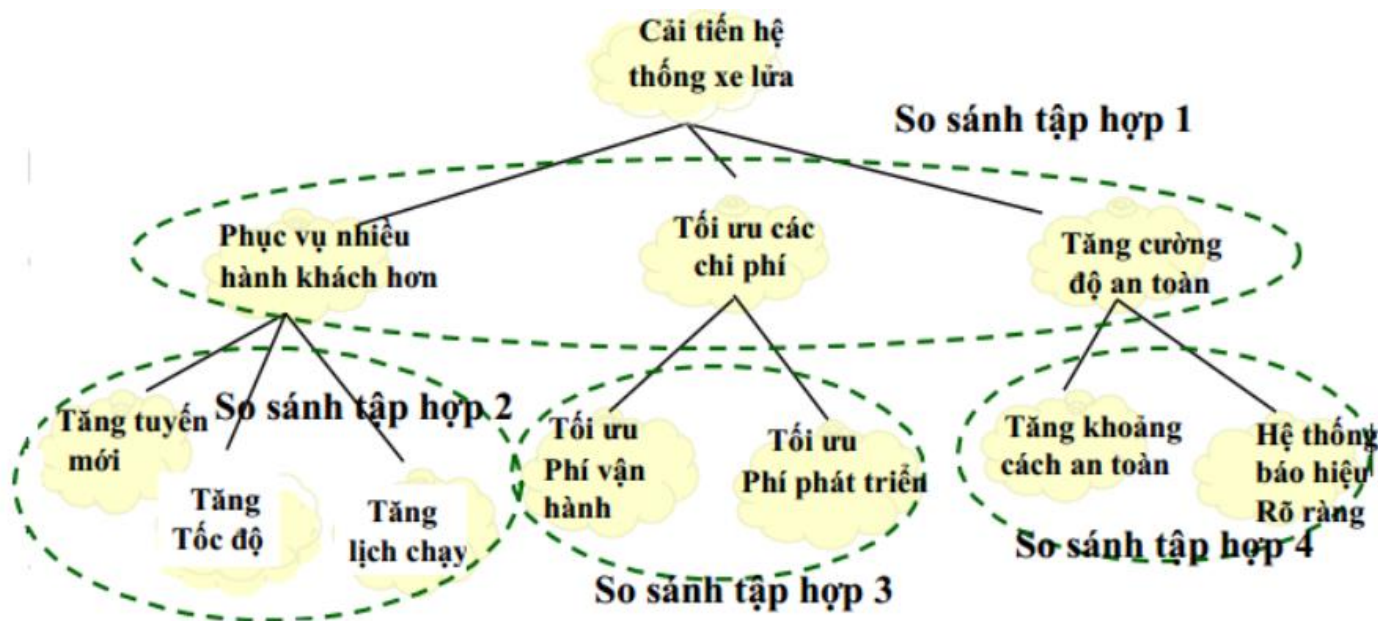
Một số vấn đề phát sinh

- **Khó để xác định mức độ chênh lệch**
 - Dễ dàng để nói “x thì quan trọng hơn y”...
 - ...hơn là ước lượng sự quan trọng nhiều như thế nào.
- **Không phải mọi yêu cầu đều có thể so sánh được**
 - Ví dụ: mức độ trừu tượng khác nhau
 - Ví dụ chức năng chủ yếu và những mở rộng của khách hàng
- **Các yêu cầu có thể không độc lập**
 - Không có điểm chọn lựa giữa X và Y nếu chúng phụ thuộc lẫn nhau
- **Các đối tác có thể không kiên định**

Ví dụ Nếu $X > Y$, và $Y > Z$, thì có lẽ $X > Z$?
- **Các đối tác có thể không thống nhất**
 - Có sự đánh giá về chi phí/giá trị khác nhau với những dạng đối tác khác nhau.

Phân cấp thứ tự ưu tiên

- **Nhóm các yêu cầu theo một cấu trúc phân cấp**
Cây mục tiêu (A goal tree) hoặc Cây NFR (NFR-Non function requirements tree).
- **Chỉ thực hiện sự so sánh giữa các nhánh của cùng một nút:**



Đánh giá chi phí/giá trị

Analytic Hierarchy Process (AHP) - *Adapted from Karlsson & Ryan 1997*

Lập ma trận $n \times n$ (cho n yêu cầu)

- Với mỗi phần tử (x,y) trong ma trận, nhập:
 - 1 – nếu x bằng y
 - 3 – nếu x lớn hơn y một chút
 - 5 – nếu x lớn hơn y nhiều
 - 7 – nếu x rất lớn hơn y
 - 9 – nếu x cực kỳ lớn hơn y
 - (dùng các giá trị trung gian 2,4,6,8 nếu cần)
- ...và với phần tử (y,x) thì nhập giá trị nghịch đảo.

Đánh giá giá trị eigenvalues:

- Ví dụ: “trung bình trên các cột đã bình quân hóa”
 - Tính tổng mỗi cột
 - Chia mỗi phần tử trong ma trận với số tổng cột của nó
 - Tính tổng mỗi hàng
 - Chia mỗi tổng hàng với số hàng
- Cuối cùng sẽ có giá trị cho mỗi yêu cầu:
 - ...cung cấp ước lượng phần trăm trên tổng giá trị dự án

Ví dụ

Cho ma trận đánh giá yêu cầu như sau:

	Req1	Req2	Req3	Req4
Req1	1	1/3	2	4
Req2	3	1	5	3
Req3	1/2	1/5	1	1/3
Req4	1/4	1/3	3	1

Hãy tính giá trị cho mỗi yêu cầu

Ví dụ

Các bước thực hiện

- Bước 1: Tính tổng các phần tử trên mỗi cột
- Bước 2: Chia mỗi phần tử của cột cho tổng cột tương ứng
- Bước 3: Tính tổng các hàng
- Bước 4: Chia mỗi tổng nhận được cho số yêu cầu

Ví dụ

Kết quả

	Req1	Req2	Req3	Req4
Req1	1	1/3	2	4
Req2	3	1	5	3
Req3	1/2	1/5	1	1/3
Req4	1/4	1/3	3	1



Bình quân hóa các cột

	Req1	Req2	Req3	Req4
Req1	0.21	0.18	0.18	0.48
Req2	0.63	0.54	0.45	0.36
Req3	0.11	0.11	0.09	0.04
Req4	0.05	0.18	0.27	0.12



Tổng các hàng

sum	sum/4
1.05	0.26
1.98	0.50
0.34	0.09
0.62	0.16



Kết quả

Req1 - 26% of the cost
Req2 - 50% of the cost
Req3 - 9% of the cost
Req4 - 16% of the cost

Ví dụ

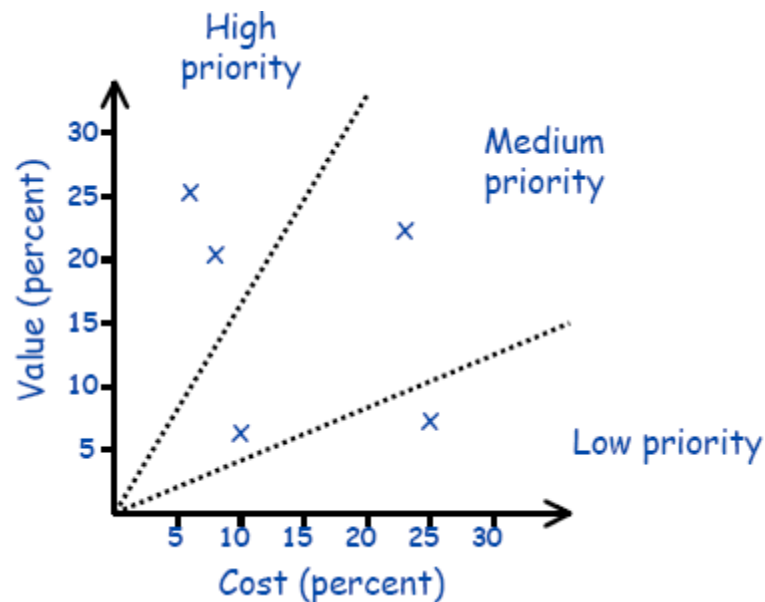
Cho ma trận đánh giá các yêu cầu như sau:

	Req1	Req2	Req3	Req4	Req5
Req1	1	1/2	4	1/3	2
Req2	2	1	1/5	4	1/2
Req3	1/4	5	1	3	1/4
Req4	3	1/4	1/3	1	5
Req5	1/2	2	4	1/5	1

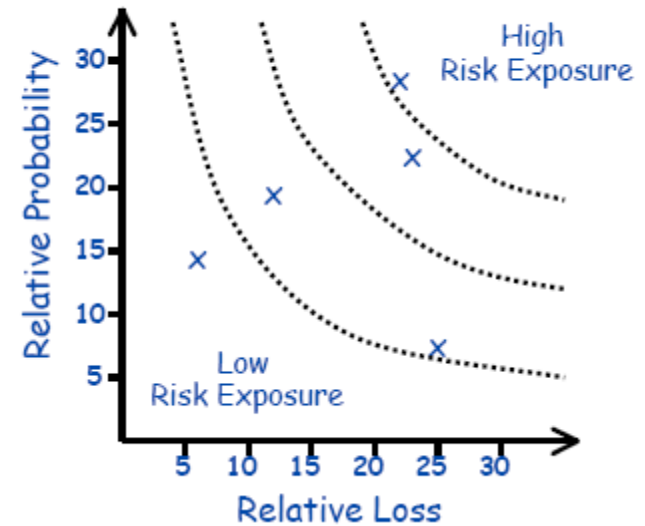
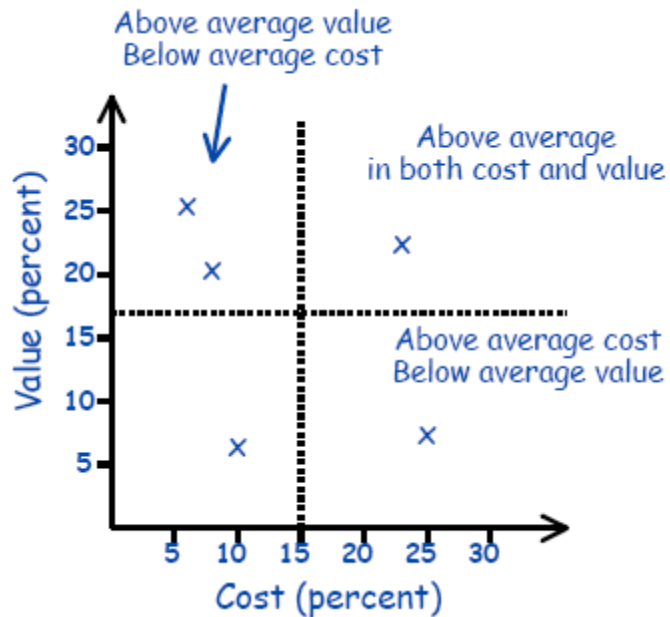
Thực hiện các bước ước lượng lợi nhuận cho mỗi yêu cầu

Vẽ đồ thị lợi nhuận trên vốn đầu tư (ROI graph)

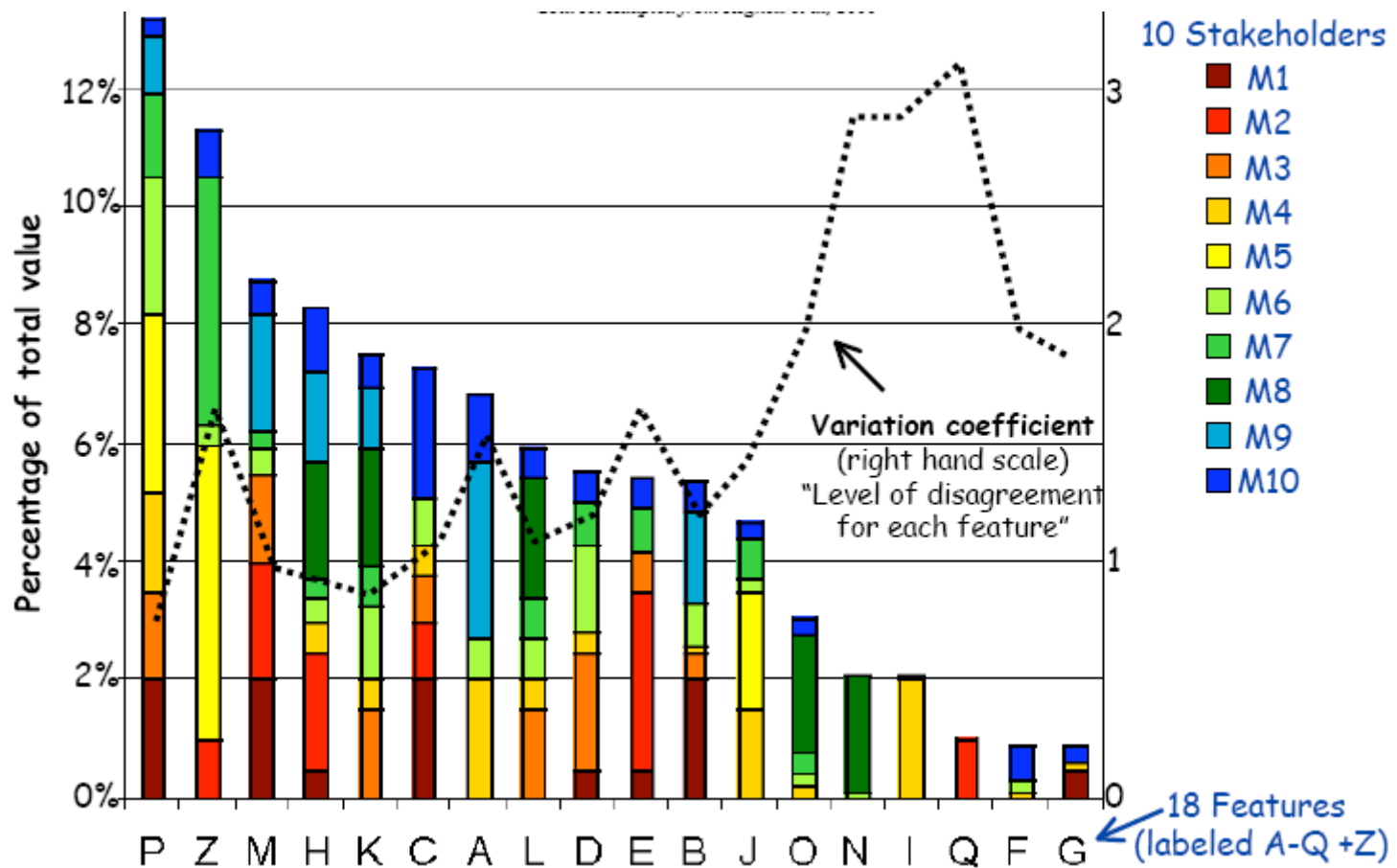
- Thực hiện quá trình AHP hai lần: để đánh giá quan hệ Giá trị và đánh giá quan hệ Chi phí
- Dùng kết quả để tính toán tỷ số ROI (Return On Investment):



Tiêu chuẩn chọn lựa khác

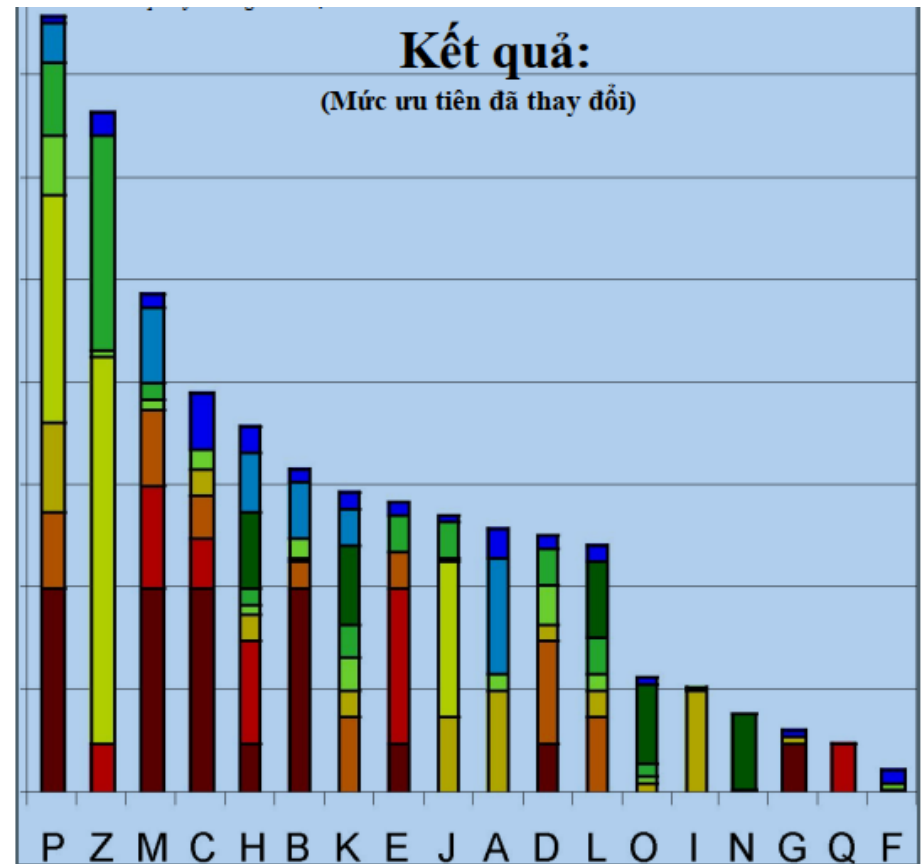
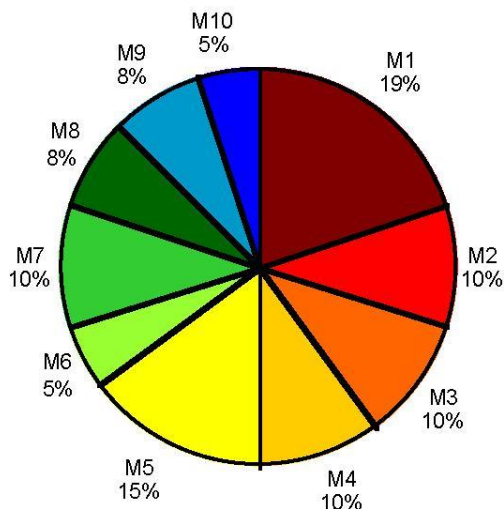


Minh họa giá trị từ các đối tác



Tính trọng số mỗi đối tác

- Trọng số của mỗi đối tác
 - Để phản ánh sự tin cậy?
 - Để phản ánh kích thước của kết quả bình chọn ?
- Ví dụ:



Cơ sở giải quyết mâu thuẫn

Đàm phán

- ... Là thăm dò sự cộng tác:
 - ✓ Một thành viên cố gắng tìm một sự thỏa thuận có thể đáp ứng cho tất cả các thành viên khác.
- Cũng được biết như:
 - ✓ Hành vi hợp nhất
 - ✓ Thiết lập thương lượng
- Khác với:
 - ✓ phân phối/cạnh tranh sự thỏa hiệp

Cạnh tranh

- Là phóng to chính bạn :
 - ✓ Không quan tâm đến mức độ hài lòng của các thành viên khác.
 - ✓ Nhưng không nhất thiết phải gây thù hằn!
- Cực điểm đến:
 - ✓ Khi tất cả thành công cho một người được trả giá bởi những người khác

Cơ sở giải quyết mâu thuẫn (tiếp)

Giải pháp “người thứ ba” (Third party)

- Các thành viên yêu cầu hỗ trợ từ một nguồn bên ngoài
 - ✓ Theo quy luật sách vở, chỉ rõ tác giả hoặc ném đồng xu.
 - ✓ Có thể dùng khi các phương pháp đàm phán và cạnh tranh đều thất bại.
- Khiếu kiện: người đại diện của mỗi thành viên phải đến tham dự.
- Không khiếu kiện: một quyết định được đưa ra bởi một nhân tố khác hơn những người đại diện.
- May rủi: e.g. ném đồng xu

BÀI TẬP

- Hoàn thiện các nội dung trong báo cáo
- Căn chỉnh định dạng và rà soát nội dung báo cáo để nộp