**Chương 8**

**Non-Functional Requirements (NFRs) & Requirements Prioritization**

**(Các yêu cầu phi chức năng và yêu cầu ưu tiên)**

Department of Software Engineering, (**Bộ môn công nghệ phần mềm)**

College of Information & Communication Technology, Can Tho University **(Khoa công nghệ thông tin và truyền thông đại học Cần Thơ)**

Tran Van Hoang

**(Các yêu cầu phi chức năng là gì)**

## n IEEE definition (Định nghĩa chuẩn IEEE)

q **Non-functional** requirements in software system engineering, a software requirement that describes not what the software will do, but how the software will do it, for example, software performance requirements, software external interface requirements, design constraints, and software quality attributes.

Yêu cầu phi chức năng trong kỹ thuật hệ thống phần mềm, yêu cầu phần mềm mô tả không phải phần mềm sẽ làm gì, nhưng phần mềm sẽ làm như thế nào, ví dụ như yêu cầu về hiệu năng phần mềm, yêu cầu giao diện phần mềm bên ngoài, các ràng buộc về thiết kế và các thuộc tính chất lượng phần mềm.

**(Các yêu cầu phi chức năng là gì)**

q Nonfunctional requirements are difficult to test.

Yêu cầu phi chức năng rất khó để kiểm tra.

q Therefore, they are usually evaluated subjectively.

Do đó, chúng thường được đánh giá một cách chủ quan.

**(Các yêu cầu phi chức năng là gì)**

## n Functional vs. Non-Functional

## Chức năng và phi chức năng

q **Functional** requirements describe what the system should do.

Yêu cầu chức năng mô tả những gì mà hệ thống nên làm

n functions that can be captured in use cases.

Về chức năng có thể bắt các lỗi trong trường hợp sử dụng.

n behaviors that can be analyzed by drawing sequence diagrams, state charts, etc.

Về hành vi có thể được phân tích bằng cách vẽ các sơ đồ tuần tư, biểu đồ trạng thái, vv.

**(Các yêu cầu phi chức năng là gì)**

## n Functional vs. Non-Functional

## Chức năng và phi chức năng

q **Non-functional** requirements are global constraints on a software system.

Các yêu cầu phi chức năng là những ràng buộc được sử dụng toàn cầu cho một hệ thống phần mềm (Dịch không xác nghĩa)

n e.g. development costs, operational costs, performance, reliability.

Ví dụ như: chi phí phát triển, chi phí hoạt động, hiệu suất, độ tin cậy.

n Maintainability, portability, robustness etc.

Khả năng bảo hành, tính khả chuyển, sự vững mạnh , vv

**(Các yêu cầu phi chức năng là gì)**

n Often known as software qualities, or just the “**ilities**”.

Thường được gọi là yêu cầu mềm hay “ilities” (Dịch không xác nghĩa).

n Usually cannot be implemented in a single module of a program.

Thông thường không thể thực hiện trong một mô- đun duy nhất cảu một chương trình.

**(Các yêu cầu phi chức năng là gì)**

### n NFRs for a computer game.

### Yêu cầu phi chức năng cho một hệ thống máy tính

1. Can run on any operating system.

**Có thể chạy trên bất kỳ hệ điều hành nào.**

1. The password should be encrypted for safety.

**Mật khẩu phải được mã hóa để an toàn.**

1. Software can run on any internet explorer, firefox, google chrome.

**Phần mềm có thể chạy trên bất kỳ trình duyệt nào như explorer, firefox, google chorme.**

**(Các yêu cầu phi chức năng là gì)**

1. Must be free of defects.

**Phải không có lỗi.**

1. Every video must have an unique identification.

**Mỗi một phim phải có một định dạng duy nhất.**

1. All requests must return a response in < 2 seconds.

**Tất cả phải trả lại phản hồi trong vòng 2 giây.**

7. …

**(Các yêu cầu phi chức năng là gì)**

## n The challenge of NFRs

## Các thách thức của yêu cầu phi chức năng

q Hard to model.

Khó để mô hình hóa

q Usually stated informally, and so are:

Thông thường cần một kết quả chính thức, và thế thì: (không xác nghĩa)

n difficult to implement during development

Khó thực hiện trong quá trình phát triển

n difficult to evaluate for the customer prior to delivery

Khó khăn để đánh giá cho khách hàng trước khi giao sản phẩm.

**(Các yêu cầu phi chức năng là gì)**

q Hard to make them measurable requirements

Khó để đo lường các yêu cầu cho chúng(Không xác nghĩa)

n We’d like to state them in a way that we can measure how well they’ve been met.

Chúng ta muốn nêu rõ chúng theo một cách mà chúng ta có thể đo được mức độ chúng được đáp ứng như thế nào.

**Ví dụ yêu cầu phi chức năng**

### n Interface requirements

### Yêu cầu dao diện

q How will the new system interface with its environment?

Giao diện hệ thống mới với môi trường của nó như thế nào?

n User interfaces

Giao diện người dùng.

n Interfaces with other systems

Dao diện với các hệ thống khác.

### n Performance requirements

### Yêu cầu về hiệu năng

**Ví dụ yêu cầu phi chức năng**

q Time/space bounds

Giới hạn không gian và thời gian.

n response time, throughput and available storage space

Thời gian đáp ứng, thông lượng và không gian lưu trữ có sẵn.

n e.g. ”the system must handle 1,000 transactions per second”

Ví dụ hệ thống phải xử lí được 1000 giao dịch mỗi giây

n Query response time must be fast. (not measurable)

Thời gian đáp ứng truy vấn phải được nhanh chóng (Không thể đo lường)

**Ví dụ yêu cầu phi chức năng**

### n Reliability

### Độ bền

q The availability of components.

Sự sẵn có của các thành phần.

q Integrity of information maintained and supplied to the system.

Tính toàn vẹn cảu thông tin được duy trì và cung cấp cho hệ thống.

n e.g. "system must have less than 1hr downtime per three months“.

ví dụ. "Hệ thống phải có thời gian chết ít hơn 1 giờ trong ba tháng".

n e.g. The system must never crash. (not feasible)

ví dụ: Hệ thống không bao giờ sụp đổ (không khả thi)

**Ví dụ yêu cầu phi chức năng**

### n Security

### Tính bảo mật

q One or more requirements about protection of your system and its data.

Một hoặc nhiều yêu cầu về bảo vệ hệ thống và dữ liệu của hệ thống

n E.g. permissible information flows, or who can do what

Ví dụ. Các luồng thông tin được phép, hoặc ai có thể làm được

**Ví dụ yêu cầu phi chức năng**

### n Portability

### Tính khả chuyển

q It must be possible to port the system to the Linux platform with < 12 person-months of effort.

Nó phải có khả năng đưa hệ thống vào nền tảng Linux với <12 tháng để người sử dụng nỗ lực.

### n Economic requirements

### Các yêu cầu kinh tế

q e.g. restrictions on immediate and/or long-term costs.

Ví dụ: Các hạn chế về chi phí trước mặt hoặc dài hạn.

**Ví dụ yêu cầu phi chức năng**

**Ví dụ yêu cầu phi chức năng**

### n Operating requirements

### Yêu cầu trong quá trình vận hành

q Physical constraints (size, weight)

Trở ngại vật lí về kích thước trọng, lượng

q Personnel availability & skill level

Sự sẵn có của nhân lực và trình độ

q Accessibility for maintenance

Khả năng tiếp cận để bảo vệ

q Environmental conditions

Điều kiện môi trường

**Ví dụ yêu cầu phi chức năng**

### n Lifecycle requirements

### Yêu cầu vòng đời

q “Future-proofing”

Trong tương lai

n Maintainability

Khả năng bảo trì

n Enhanceability

Tính năng nâng cấp

q Limits on development

Các giới hạn về mặt phát triển

n E.g. development time limitations

Ví dụ: Hạn chế về thời gian phát triển.

**Ví dụ yêu cầu phi chức năng**

n Resource availability

Nguồn lực có sẵn

n Methodological standards

Tiêu chuẩn phương pháp luận

Requirements Prioritization

**Yêu cầu ưu tiên**

### n Why Prioritization is needed?

### Tại sau ưu tiên là cần thiết

n Cost-Value Approach

Tiếp cận chi phí, giá trị

q Sorting Requirements by cost/value

Phân loại yêu cầu chi phí/ giá trị

q Estimating Relative Costs/Values using AHP

Ước tính chi phí tương đối / giá trị sử dụng AHP

**Tại sao cần độ ưu tiên**

### n When customer expectations are high and timelines are short, you need to make sure the product delivers the most critical or valuable functionality as early as possible.

### Khi kỳ vọng của khách hàng cao và thời gian ngắn, bạn cần đảm bảo sản phẩm mang lại chức năng quan trọng nhất hoặc có giá trị càng sớm càng tốt.

n Prioritization is a way to deal with competing demands for limited resources.

### Ưu tiên là cách giải quyết nhu cầu cạnh tranh đối với các nguồn lực hạn chế

**Tại sao cần độ ưu tiên**

n Prioritization is a critical strategy for agile projects.

Ưu tiên là một chiến lược quan trọng cho các dự án nhanh.

**Khái niệm cơ bản về độ ưu tiên**

n **Need to select what to implement**

Cần chọn những gì để thực hiện

q Customers (usually) ask for way too much.

Khách hàng thường yêu cầu rất nhiều

q Balance time-to-market with amount of functionality.

Cân bằng thời gian đưa ra thị trường với số lượng các chức năng

q Decide which features go into the next release.

Quyết định thêm tính năng nào vào phiên bản tiếp theo

**Khái niệm cơ bản về độ ưu tiên**

n **For each requirement/feature, ask:**

Đối với mỗi yêu cầu/ tính năng cần hỏi:

q How important is this to the customer?

Quan trọng với khách hàng như thế nào

q How much will it cost to implement?

Bao nhiêu chi phí để có thể thực hiện

q How risky will it be to attempt to build it?

Có những nguy cơ nào trong qua trình xây dựng nó

**Cách tiếp cận chi phí – giá trị**

## n Calculate return on investment (ROI)

## Tính toán lợi nhuận trên vốn đầu tư

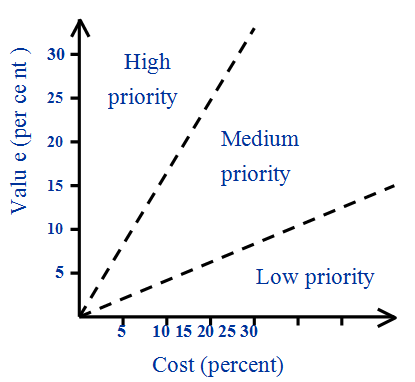
q Assess each requirement’s importance to the project as a whole.

Đánh giá tầm quan trọng đối với mỗi dự án nói chung.

q Assess the relative cost of each requirement.

Đánh giá các chi phí tương đối đối với mỗi yêu cầu.

q Compute the cost-value trade-off:



Tính toán gia rị thương mại

**Ước tính chi phí và giá trị**

## n Two approaches:

## Hai phương pháp

q Absolute scale (e.g. dollar values)

Quy mô tuyệt đối (Ví dụ: Giá trị đồng đô)

n Requires much domain experience.

Yêu cầu kinh nghiệm trên các vùng miền

q Relative values (e.g. less/more; a little, somewhat, very)

Giá trị tương đối (Ví dụ: ít/ nhiều, một ít, hơi ít, rất nhiều)

n Much easier to elicit.

Dễ dàng hơn trong gợi nhớ

n Prioritization becomes a sorting problem .

Ưu tiên trở thành một cách phân loại

**Ước tính chi phí và giá trị**

## n Comparison Process – options

## So sánh và lựa chọn

q Basic sorting - for every pair of requirements (i, j), ask if i>j?

Phân loại cơ bản - cho mỗi cặp yêu cầu (i, j), hãy hỏi nếu i> j?

n E.g. bubblesort - start in random order, and swap each pair if out of order.

Ví dụ. Bong bóng sắp xếp - bắt đầu theo thứ tự ngẫu nhiên, và trao đổi mỗi cặp nếu không theo thứ tự

n requires n\*(n-1)/2 comparisons

Đòi hỏi n\*(n-1)/2 so sánh

**Ước tính chi phí và giá trị**

q Construct a Binary Sort Tree

Xây dựng một cây nhị phân theo:

n Requires O(nlogn) comparisons

Yêu cầu O (nlogn) số lượng các so sánh

**Một số phiền phức**

### n Hard to quantify differences

### Khó để định lượng được sự khác biệt

q Easier to say “*x is more important than y*”…

Dễ dàng hơn khi nói là x quan trọng hơn y

q …than to estimate by how much

Nhưng khó hơn để ước tính là dễ hơn bao nhiêu

### n Not all requirements comparable

### Không phải tất các các yêu cầu đều so sánh được

**Một số phiền phức**

q E.g. different level of abstraction

Ví dụ mức độ trừu tượng khác nhau

### n Requirements may not be independent

### Yêu cầu có thể không được độc lập

q No point selecting between X and Y if they are mutually dependent

Không có điểm để lựa chọn giữa x và y nếu chúng phụ thuộc nhau

**Một số phiền phức**

### n Stakeholders may not be consistent

### Các bên liên quan có thể không phù hợp

q E.g. If X > Y, and Y > Z, then presumably X > Z?

Ví dụ; Nếu X > Y và Y>Z nhưng sau đó có thể X>Z

### n Stakeholders might not agree

### Không có sự đồng thuận giữa các bên liên quan

q Different cost/value assessments for different types of stakeholder

Đánh giá chi phí / giá trị khác nhau cho các loại bên liên quan khác nhau

**Thứ tự ưu tiên**

### n Group Requirements into a hierarchy

### Yêu cầu nhóm thành một hệ thống

q E.g. A goal tree

Ví dụ: Một cây mục tiêu

q E.g. A NFR tree

Ví dụ: Một cây yêu cầu phi chức năng

### n Only make comparisons between branches of a single node:

### Chỉ thực hiện so sánh giữa các nhánh của một nút duy nhất: (Dịch cho dễ hiểu là chỉ so sánh giữa các nhánh có cùng nút cha)

Better system

Comparison set 1

Serve more passengers

Minimize costs Improve safety

Comparison set 3 Comparison set 2

Add new tracks

Increase more frequent

Increase safe distance

Clearer signalling

**Quá trình phân cấp**

### n Create n x n matrix (for n requirements)

### Tạo ma trận nxn đối với một yêu cầu n

q For cell (x,y) in the matrix enter:

Đối với ô (x,y) trong ma trận nhập:

n 1 - if x and y are of equal value

Nếu x và y là giá trị như nhau

n 3 - if x is slightly more preferred than y

Nếu x có nhiều ưu đãi hơn y một chút

n 5 - if x is strongly more preferred than y

Nếu x được ưu tiên nhiều hơn so với y

**Quá trình phân cấp**

n 7 - if x is very strongly more preferred than y

Nếu x được ưu tiên rất nhiều so với y

n 9 - if x is extremely more preferred than y

Nếu x được ưu tiên tuyệt đối so với y

n (use the intermediate values, 2, 4, 6, 8 if compromise needed)

Sử dụng các giá trị trung gian là 2 4 6 8 nếu cần thiết trong lựa chọn

q …and for (y,x) enter the reciprocal

Và cho x và y vào so sánh

**Quá trình phân cấp**

### n Estimate the eigenvalues:

### Ước tính giá trị riêng

q E.g. “averaging over normalized columns”

Ví dụ giá trị trung bình trên cột bình thường

1. Calculate the sum of each column.

Tính tổng của mỗi cột

1. Divide each element in the matrix by the sum of it’s column.

Chia phần tử trong ma trận với tổng của cột chứa nó

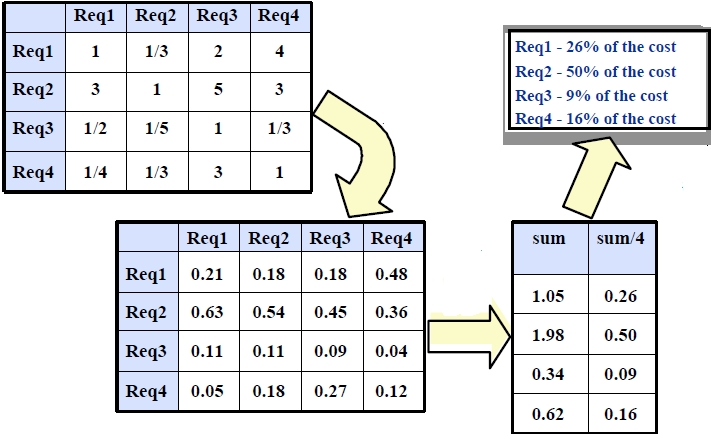
**Quá trình phân cấp**

1. Calculate the sum of each row.

Tính tổng của mỗi hàng

1. Divide each row sum by the number of rows.

Chia mỗi hàng với tổng số hàng

Normalise columns

Result

Sum the rows

**Đồ thị ROI**

## n Do AHP process twice:

## Thực hiện AHP hai lần

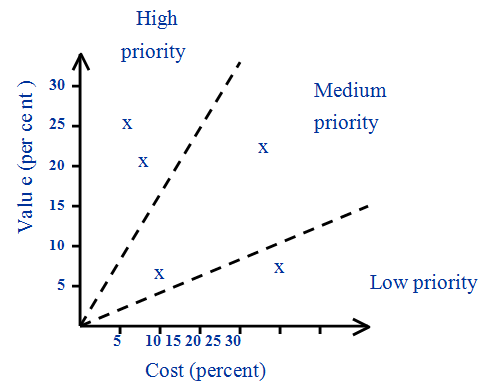
q Once to estimate relative value

Lúc ước tính giá trị tương đối

q Once to estimate relative cost

Lúc ước lượng chi phí tương đối

## n Use results to calculate ROI ratio:

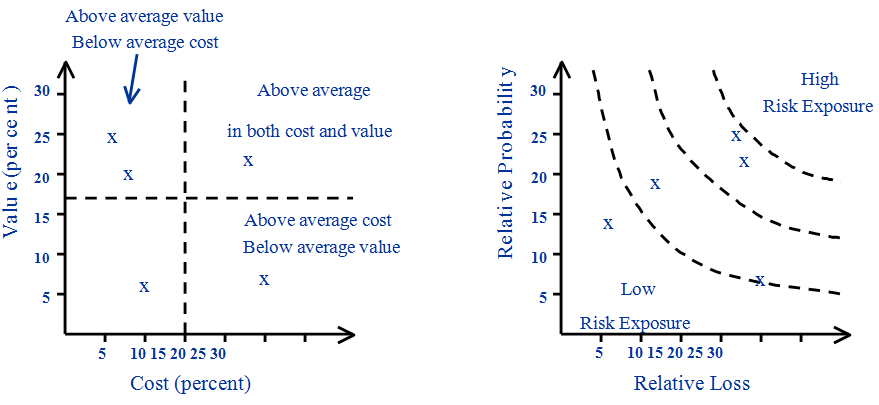


## Sử dụng kết quả để ước tính tỉ lệ ROI

**Các tiêu chí lựa chọn khác**

## n ROI ratio is not the only way to group requirements

## Tỉ lệ ROI không phải là cách duy nhất để nhóm các yêu cầu



**Các kỹ thuật khác**

n **In or out**

Trong hay ngoài

q Very simple

Rất đơn giản

q A group of stakeholders work down a list of requirements and make a binary decision: is it in, or is it out?

Một nhóm các bên liên quan làm việc theo một danh sách các yêu cầu và đưa ra quyết định nhị phân: đó là nó, hay là nó

**Các kỹ thuật khác**

q Keep referring to the project’s business objectives to make this judgment.

Giữ lại mục tiêu kinh doanh của dự án để đưa ra quyết định.

n **$100**

n **MoSCoW**

## n Three-level scale

Karl Wiegers and Joy Beatty, Software Requirements, Chapter 16. pp.313-327

Karl Wiegers và Joy Beatty, Yêu cầu phần mềm, Chương 16. Trang.313-32

**Tài liệu tham khảo**

[1] Prof Steve Easterbrook, lecture notes, University of Toronto, Canada.

[1] GS. Steve Easterbrook, bài giảng, Đại học Toronto, Canada.

[2] Karl Wiegers and Joy Beatty, Software Requirements, Chapter 16. pp.313-327

[2] Karl Wiegers và Joy Beatty, Yêu cầu phần mềm, Chương 16. Trang.313-327

# Q&A