**PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**CHƯƠNG 1 :**

***\* Dữ liệu ( Data ) :***  
- Các định lượng, ký tự hoặc ký hiệu mà máy tính thực hiện các thao tác, có thể được lưu trữ và truyền dưới dạng tín hiệu điện và được ghi lại trên phương tiện ghi từ tính, quang học hoặc cơ học

***\* Dữ liệu lớn ( Big data ) :***

- Là các tập dữ liệu có khối lượng lớn và phức tạp mà các ứng dụng xử lý dữ liệu truyền thống không có khả năng thu thập và xử lý tốt

***\* Thông tin ( Information ) :***

- Là dữ liệu đã được xử lý, đặt trong ngữ cảnh phù hợp, có ích cho người dùng

***\* Hệ thống ( System ) :***

- Là tập hợp các phần tử ( components ) tác động lẫn nhau, phối hợp hoạt động ( word together ) để đạt được các mục tiêu ( objectives ) xác định

***\* Hệ thống thông tin ( Information System ) :***

- Là tập hợp các thiết bị phần cứng, phần mềm, hệ thống truyền thông, được tổ chức lại để xử lý dữ liệu và phân bố thông tin phục vụ cho các mục tiêu của một đơn vị

- Các thành phần cấu thành một hệ thống thông tin bao gồm : phần cứng ( hardware ), phần mềm ( software ), dữ liệu ( data ), quy trình xử lý ( processes ), con người ( people )

***\* Tổ chức quản lý trong công ty :***

- Quản lý chiến lược ( strategical ) : quản trị viên cấp cao

- Quản lý chiến thuật ( tactical ) : quản trị viên cấp trung

- Quản lý tác nghiệp ( operational ) : quản trị viên cấp thấp

- Thực hiện nghiệp vụ : nhân viên thừa hành

***\* Các loại quyết định :***

**Quyết định có cấu trúc :**

- Khi cả 2 yếu tố trên đều rõ ràng

- Ví dụ : xử lý đơn đặt hàng quy định rõ là dựa vào trị giá đơn đặt hàng và hạn mức tín dụng của khách để chấp nhận hay không đơn đặt hàng của khách

- Loại quyết định này có thể lập trình trên máy tính để hỗ trợ ở mức tự động hóa cao

**Quyết định phi cấu trúc :**

- Khi cả 2 yếu tố đều không rõ

- Ví dụ : quyết định khi bỏ phiếu bình chọn bàn thắng đẹp nhất

- Nhiều quyết định phi cấu trúc dựa vào cảm tính và kinh nghiệm ứng xử của người ra quyết định, cũng như tác động của môi trường xung quanh

**Quyết định bán cấu trúc :**

- Khi 1 trong 2 yếu tố là không rõ ràng, như dữ liệu không thể lượng hóa được, hoặc không có thủ tục chặt chẽ, hoặc các tiêu chuẩn ra quyết định không đầy đủ

- Ví dụ : quyết định tung sảm phẩm mới ra thị trường là quyết định bán cấu trúc, vì để ra quyết định này giám đốc phải căn cứ vào lợi nhuận ước tính, xem xét đến mức độ rủi ro, số liệu về dự báo nhu cầu thị trường

***\* Quản trị viên cấp cao :***

- Thường quan tâm đến tổng thể của công ty

- Vạch ra các kế hoạch dài hạn, mang tính chiến lược ( strategic ), các kế hoạch này ảnh hưởng đến sự tồn tại và phát triển của công ty

- Ví dụ : xác định thị trường thâm nhập, cấu trúc vốn

- Các quyết định thường là phi cấu hoặc bán cấu trúc

- Thông tin đa số thu thập từ bên ngoài vượt khỏi tầm kiểm soát của công ty, thường là các thông tin dự báo thị trường, đổi mới công nghệ

***\* Quản trị viên cấp trung :***

- Thường vạch ra các kế hoạch trung hạn vài tháng đến một năm mang tính chiến thuật ( tactical ) nhằm đạt được các mục tiêu kinh doanh đã hoạch định

- Ví dụ : phân phối nguồn tài nguyên trong công ty

- Các quyết định của cấp này thường là bán cấu trúc hoặc có cấu trúc, các thông tin thường thu thập từ bên trong hệ thống và đôi khi cần một ít thông tin ngoài, do thu thập từ bên trong hệ thống và tương lai gần, nên các thông tin chắc chắn và cụ thể hơn

***\* Quản trị viên cấp thấp :***

- Liên quan đến các hoạt động hàng ngày của công ty, các quyết định mang tính tác nghiệp ( operational ) nhằm đảm bảo sử dụng hiệu quả nguồn tài nguyện hiện có để hoàn thành các mục tiêu đã đặt ra

- Ví dụ : kiểm soát tồn kho, tín dụng, thuê xe

- Thông tin hầu như có nguồn gốc từ bên trong công ty, đặc điểm thông tin ở cấp này là chi tiết và chắc chắn

***\* Nhân viên thừa hành :***

- Thực hiện các công việc được giao, như tiếp nhận và nhập số liệu vào hệ thống

- Họ thường chỉ yêu cầu hệ thống thông tin phải vận hành dễ dàng, nhanh chóng và thật chính xác

***\* Phân loại hệ thống thông tin :***

**Hệ thống xử lý giao dịch – TPS :**

- Là hệ thống được xây dựng để xử lý các giao dịch, các công việc hàng ngày

- Thường khối lượng dữ liệu xử lý lớn

- Tuân theo quy trình nghiệp vụ với độ chính xác và an toàn cao

- Các hệ thống này thường yêu cầu mức độ chi tiết cao và dễ vận hành

- Ví dụ : hệ thống bán hàng ở các siêu thị

**Hệ thống tự động hóa công việc văn phòng – OAS và hệ thống tri thức – KWS :**

- OAS : hỗ trợ việc tự động hóa các công việc văn phòng như : thư điện tử, lập lịch, nhắc việc

- KWS : giúp tạo và phát triển những thông tin, kiến thức mới, hệ thống quản lý tri thức tổ chức lại các kiến thức này, giúp các nhân viên có thể chia sẻ kiến thức bất cứ nơi nào và khi nào

- Ví dụ KWS : hệ trợ giúp đào tạo công nhân

**Hệ thống thông tin quản lý – MIS :**

- Cung cấp những công cụ hỗ trợ cho nhà quản lý lấy được các thông tin, phục vụ cho nhu cầu ra quyết định hàng ngày của họ

- Các thông tin cung cấp cho nhà quản lý có thể thực hiện : theo định kỳ, bất cứ lúc nào khi có yêu cầu hoặc tình huống đặc biệt

- Hệ thống này không tồn tại độc lập mà thường bao hàm cả hệ thống TPS, nó phân tích các dữ liệu từ hệ thống TPS đưa lên

- Ví dụ : hệ thống quản lý khách sạn, quản lý kho

**Hệ hỗ trợ ra quyết định – DSS :**

- Sử dụng các mô hình ra quyết định và cơ sở dữ liệu chuyên môn hóa để đưa ra các phương án khác nhau, các mô hình phân tích, mô phỏng cho nhà quản lý

- Nguồn dữ liệu lấy từ TPS và các nguồn dữ liệu khác bên ngoài công ty

- Quyết định cuối cùng vẫn là con người

- Câu hỏi thường ở dạng WHAT-IF và nhận được câu trả lời kiểu tương tác

- Ví dụ : hệ thống dự báo kinh tế, chuẩn đoán bệnh từ xa

**Hệ chuyên gia – ES và trí tuệ nhân tạo – AI :**

- ES : thu thập và sử dụng các kiến thức của các chuyên gia chuyên ngành ( domain expert ) để giải quyết các vấn đề phức tạp trong thực tế

- AI : là một lĩnh vực của hệ chuyên gia, mục đích là phát triển một máy tính có trí thông minh như con người, có khả năng phân tích, suy diễn

**Hệ hỗ trợ ra quyết định nhóm – GDSS :**

- Cho phép một nhóm các nhà quản lý có thể phối hợp với nhau, cùng nhau làm việc để xây dựng các quyết định mang tính chiến lược

**Hệ trợ giúp điều hành – ESS :**

- Còn gọi là hệ thống thông tin giám đốc ( EIS ), mục đích của hệ thống cung cấp cho giám đốc khả năng truy cập dễ dàng và tức thời thông tin có chọn lọc, có tính quyết định đến việc hoàn thành mục tiêu chiến lược của công ty

***\* Phân tích viên hệ thống :***

- Là người chịu trách nhiệm chính trong việc phân tích các nghiệp vụ, nhận ra các cơ hội để cải tiến, thiết kế và cài đặt hệ thống thông tin đạt được mục tiêu trên

- Để xây dựng hệ thống thành công, phân tích viên cần hiểu rõ phương pháp luận, nắm vững kỹ thuật và thực hiện một cách sáng tạo các bước trong quy trình phát triển hệ thống

***\* Vai trò của phân tích viên :***

- Là một chuyên gia tư vấn : tư vấn về phần cứng, phần mềm, chức năng, cơ sở kỹ thuật. Các tư vấn này phải phù hợp với các yêu cầu kỹ thuật của hệ thống sắp xây dựng

- Là một chuyên gia trợ giúp : là người trung gian giữa khách hàng và lập trình viên, phân tích viên hệ thống phải giải đáp mọi thắc mắc từ cả 2 phía để có thể chuyển những yêu cầu trong thế giới thực hành hệ thống thông tin hoàn chỉnh

- Là tác nhân thay đổi : phân tích viên có khả năng tác động để thay đổi quy trình vận hành nhằm đưa công nghệ thông tin vào ứng dụng cho đơn vị

***\* Các giai đoạn của SDLC ( chu kỳ phát triển hệ thống ) :***

- Lập kế hoạch ( Planning ) : tại sao phải xây dựng hệ thống

- Phân tích ( Analysis ) : hệ thống cần phải làm những gì, ai sẽ là người phân tích, phân tích khi nào, địa điểm phân tích

- Thiết kế ( Design ) : làm thế nào để hệ thống hoạt động

- Thi công ( Implementation ) : lập trình và thử nghiệm hệ thống

- Vận hành và hỗ trợ ( Operation and support ) : đưa hệ thống vào vận hành

***\* Một số kỹ thuật phát triển :***

- JAD ( Joint Application Development ) : kỹ thuật xây dựng nhóm cộng tác trong phát triển ứng dụng, bao gồm nhiều thành viên thuộc nhiều lĩnh vực và cấp độ quản lý

- RAD ( Rapid Application Development ) : kỹ thuật phát triển nhanh ứng dụng, nổi lên từ những năm 1990, nhờ sự xuất hiện các công cụ hỗ trợ như CASE tools, các ngôn ngữ lập trình trực quan ngày càng mạnh mẽ

**\* CASE tools :**

- Viết tắt của Computer Aided System Engineering

- Là các công cụ hỗ trợ cho việc phát triển hệ thống ở nhiều mức độ khác nhau

- Upper CASE tools : hỗ trợ việc mô hình hóa hệ thống và tạo ra thiết kế luận lý cho hệ thống

- Lower CASE tools : giúp phát sinh mã nguồn chương trình từ các mô hình luận lý, đẩy nhanh tốc độ phát triển hệ thống

- Tất cả các thiết kế được lưu trữ trong một kho gọi là CASE repository

***\* Tuyên ngôn của Agile :***

- Cá nhân và sự tương hỗ quan trọng hơn quy trình và công cụ

- Sản phẩm sử dụng được quan trọng hơn tài liệu về sản phẩm

- Cộng tác với khách hàng quan trọng hơn đàm phán hợp đồng

- Phản hồi với sự thay đổi quan trọng hơn bám theo kế hoạch

***\* 12 nguyên tắc của Scrum :***

- Thỏa mãn yêu cầu của khách hàng thông qua việc giao hàng sớm và liên tục

- Chào đón việc thay đổi yêu cầu, thậm chí là những thay đổi yêu cầu muộn

- Giao phần mềm chạy được cho khách hàng một cách thường xuyên

- Nhà kinh doanh và kỹ sự lập trình phải làm việc cùng nhau hàng ngày trong suốt dự án

- Các dự án được xây dựng xung quanh những cá nhân có động lực, cung cấp cho họ môi trường và sự hỗ trợ cần thiết và tin tưởng họ để hoàn thành công việc

- Trao đổi trực tiếp mặt đối mặt là phương pháp hiệu quả nhất để truyền đạt thông tin

- Phần mềm chạy được là thước đo chính của tiến độ

- Phát triển bền vững và duy trì được nhịp độ phát triển liên tục

- Liên tục quan tâm đến kỹ thuật và thiết kế để cải tiến sự linh hoạt

- Sự đơn giản là cần thiết – nghệ thuật tối đa hóa lượng công việc chưa hoàn thành

- Nhóm tự tổ chức

- Thích ứng thường xuyên với sự thay đổi

***\* Ba vai trò :***

- Product Owner ( chủ sản phẩm ) : là người chịu trách nhiệm về sự thành công của dự án, người định nghĩa các yêu cầu và đánh giá cuối cùng đầu ra của các nhà phát triển phần mềm

- Scrum Master : là người có hiểu biết sâu sắc về Scrum và đảm bảo nhóm có thể làm việc hiệu quả với Scrum

- Dev Team ( đội phát triển ) : một nhóm liên chức năng tự quản lý để tiến hành chuyển đổi Product Backlog item thành chức năng của hệ thống

***\* Bốn cuộc họp :***

- Sprint Planning ( Họp kế hoạch Sprint ) : Scrum sử dụng cách thức lập kế hoạch từng phần và tăng dần theo thời gian, việc lập kế hoạch không diễn ra duy nhất một lần trong vòng đời dự án mà được lặp đi lặp lại, thích nghi với các tình hình thực tiễn trong tiến trình đi đến sản phẩm

- Daily Scrum ( Họp Scrum hằng ngày ) : Scrum Master tổ chức cho đội sản xuất họp hằng ngày trong khoảng 15 phút để nhóm phát triển chia sẻ tiến độ công việc cũng như chia sẻ các khó khăn gặp phải trong quá trình phát triển phần mềm suốt một Sprint

- Sprint Review ( Họp sơ kết Sprint ) : Cuối Sprint, nhóm phát triển cùng với Product Owner sẽ rà soát lại các công việc đã hoàn tất trong Sprint vừa qua và đề xuất các chỉnh sửa hoặc thay đổi cần thiết cho sản phẩm

- Sprint Retrospective ( Họp cải tiến Sprint ) : Dưới sự trợ giúp của Scrum Master, nhóm phát triển sẽ rà soát lại toàn diện Sprint vừa kết thúc và tìm cách cải tiến quy trình làm việc cũng như bản than sảm phẩm

**CHƯƠNG 2 :**

***\* Các lý do của yêu cầu hệ thống :***

- Cải tiến dịch vụ : để phục vụ tốt hơn cho khách hàng và người sử dụng

- Hiệu quả tốt hơn : truy xuất dữ liệu nhanh hơn, đáp ứng theo sự phát triển của đơn vị, cần cấu hình phần cứng mạnh hơn

- Giảm chi phí : chi phí vận hành và bảo trì

- Thông tin nhiều hơn : cung cấp đầy đủ thông tin, hỗ trợ việc thay đổi yêu cầu

- Kiểm soát tốt hơn : đảm bảo dữ liệu được chính xác và an toàn, mã hóa dữ liệu, quyền truy cập của người dùng

***\* Các bước nghiên cứu sơ bộ :***

**Bước 1 : Hiểu rõ vấn đề :**

- Xác định đúng bản chất của vấn đề và lý do của các yêu cầu hệ thống

- Xác định vấn đề đã nêu có phải là vấn đề có thật hay không

- Xác định rõ phạm vi khảo sát

**Bước 2 : Xác định phạm vi và các ràng buộc :**

- Phạm vi dự án : xác định ranh giới và mức độ cho phép mở rộng, lưu ý hiện tượng dự án bị nới rộng dần

- Các ràng buộc : là các điều kiện mà hệ thống phải thỏa mãn hoặc các kết quả mà hệ thống phải đạt được, phân biệt giữa các loại ràng buộc : hiện tại và tương lai, bên trong và bên ngoài, bắt buộc và mong muốn

**Bước 3 : Tìm hiểu thực tế :**

- Phân tích sơ đồ tổ chức, phỏng vấn, quan sát, thăm dò ý kiến người dùng

**Bước 4 : Xác định tính khả thi :**

- Khả thi về kỹ thuật, kinh tế và vận hành

**Bước 5 : Ước tính thời gian và chi phí :**

- Xác định các thông tin cần thu thập, nguồn của các thông tin này, các khó khăn để thu thập và phân tích các thông tin

- Có cần phỏng vấn, số người phỏng vấn, mất thời gian bao lâu, ai thực hiện phỏng vấn và mất thời gian bao lâu

- Chi phí để phân tích các thông tin và báo cáo cho ban quản lý

- Xác định chi phí và thời gian cho giai đoạn kế tiếp

- Xác định chi phí cho toàn dự án

**Bước 6 : Báo cáo cho ban quản lý :**

- Công việc cuối cùng là lập báo cáo gửi cho ban quản lý

- Báo cáo cần trình bày : đánh giá về yêu cầu hệ thống, ước lượng chi phí và lợi ích, các đề nghị

- Ngoài báo cáo viết, phân tích viên có thể còn được yêu cầu có một buổi trình bày cho ban quản lý của dự án

***\* Khả thi về kỹ thuật :***

- Trả lời câu hỏi : có thể làm được không

- Ý kiến của đội ngũ kỹ thuật : sự quen thuộc đối với ứng dụng, sự hiểu biết về lĩnh vực hoạt động

- Sự quen thuộc về kỹ thuật : mở rộng công nghệ đang dùng, có sẵn thiết bị cần thiết

- Độ tin cậy của phần cứng và phần mềm

- Hiệu suất thích hợp với yêu cầu

- Khả năng đáp ứng yêu cầu phát triển của tương lai

- Độ lớn của dự án : số người tham gia, thời gian và các đặc thù của hệ thống

***\* Khả thi về kinh tế :***

- Trả lời câu hỏi : có nên làm hay không

- Xem xét các lợi ích dự kiến mang lại so với chi phí ước lượng sẽ phải bỏ ra cho việc phát triển, cài đặt và vận hành hệ thống

- So sánh tỉ số Costs / Benefits giữa các giải pháp khác nhau

- Tính toán ROI, NPV

- Chi phí một lần và chi phí duy trì : chi phí phát triển phần cứng, phần mềm, bản quyền, lương, chi phí vận hành hàng năm

- Lợi ích : lợi ích hữu hình và lợi ích vô hình

- Tính toán chi phí và lợi ích theo thời gian

- So sánh với chi phí nếu không xây dựng hệ thống

***\* Khả thi về vận hành :***

- Trả lời câu hỏi : khi dự án được xây dựng xong, có đưa vào vận hành được không

- Cần phân tích những người có ảnh hưởng đến dự án như các nhà quản lý, người dùng, bộ phận máy tính, các tổ chức có liên quan

- Xem xét sự ủng hộ và phản đối của người dùng

- Nhân lực triển khai

- Sự ảnh hưởng đến hiệu suất, khách hàng và hoạt động của công ty

- Lịch trình triển khai có hợp lý không

**CHƯƠNG 3 :**

***\* Phỏng vấn :***

- Được sử dụng nhiều nhất

- Ưu điểm là cho kết quả và trang bị kiến thức cho người phân tích trong một thời gian ngắn

- Các bước thực hiện : xác định đối tượng phỏng vấn, xác định mục tiêu phỏng vấn, chuẩn bị câu hỏi và cấu trúc hỏi, chuẩn bị cho buổi phỏng vấn, thực hiện việc phỏng vấn, lập báo cáo buổi phỏng vấn, đánh giá buổi phỏng vấn

**Bước 1 : Xác định đối tượng phỏng vấn :**

- Chọn đúng người để phỏng vấn : bộ phận điều hành : hiểu mục đích, tổng quan về hệ thống, bộ phận thực hiện : hiểu cách xử lý chi tiết nghiệp vụ

- Dựa vào sơ đồ tổ chức của công ty, tiến hành các buổi phỏng vấn chính thức

- Cần tìm hiểu cấu trúc phi chính thức trong đơn vị

- Phỏng vấn nhiều người cùng lúc : tiết kiệm thời gian, có cơ hội đánh giá mối quan hệ cá nhân, một số thông tin nhạy cảm khó thu thập, trong buổi phỏng vấn đôi khi chỉ xoay quanh một vài người

**Bước 2 : Xác định mục tiêu buổi phỏng vấn :**

- Xác định những lãnh vực cần thảo luận

- Liệt kê các sự kiện muốn thu thập

- Các ý tưởng, đề nghị cần đề xuất

- Nên hạn chế chỉ nên đặt 1 tới 3 mục tiêu trong một buổi phỏng vấn

**Bước 3 : Chuẩn bị câu hỏi :**

- Giúp phân tích viên không bị lệch hướng khi phỏng vấn

- Câu hỏi mở : để người được phỏng vấn tự do trình bày các ý kiến của mình, có thể thu được nhiều thông tin mới

- Câu hỏi đóng : câu trả lời đã được giới hạn, lấy ngay thông tin cụ thể

- Cấu trúc hỏi hình tháp ( pyramid structure ) : bắt đầu bằng câu hỏi đóng, mở rộng vấn đề bằng các câu hỏi mở, kết thúc buổi phỏng vấn là một câu hỏi mang tính tổng quát

- Cấu trúc hỏi hình phễu ( funnel structure ) : bắt đầu bằng câu hỏi tổng quát, thu hẹp dần các vấn đề bằng các câu hỏi đóng, kết thúc buổi phỏng vấn là các câu hỏi mang tính đặc thù

- Cấu trúc hỏi hình kim cương ( diamond structure ) : bắt đầu bằng câu hỏi đóng thông thường để lấy số liệu, mở rộng vấn đề bằng các câu hỏi mở, khi đã đạt được mục tiêu tổng quát sẽ tiếp tục thu hẹp vào các vấn đề chuyên sâu bằng các câu hỏi đóng

- Kết thúc buổi phỏng vấn là các câu hỏi chuyên biệt

**Bước 4 : Chuẩn bị cho buổi phỏng vấn :**

- Xác định ngày giờ cụ thể

- Xác định địa điểm phỏng vấn ( tại ngay chỗ làm việc hay ở phòng họp )

- Thông báo nhắc nhở

- Gởi trước cho người được phỏng vấn các câu hỏi chủ yếu và các yêu cầu về tài liệu để họ chuẩn bị

- Đọc trước các tài liệu liên quan đến công ty và các nghiệp vụ cần phỏng vấn

**Bước 5 : Thực hiện cuộc phỏng vấn :**

- Tự giới thiệu

- Mô tả tổng quát dự án

- Giải thích mục tiêu của buổi phỏng vấn

- Đặt các câu hỏi theo thứ tự đã chuẩn bị

- Chú ý lắng nghe cẩn thận

- Tổng kết các điểm chính

- Chuẩn bị cho cuộc phỏng vấn kế tiếp

**Bước 6 : Lập báo cáo buổi phỏng vấn :**

- Trong buổi phỏng vấn nên ghi chép nhanh và ngắn gọn các thông tin quan trọng

- Nếu sử dụng các phương tiện ghi âm, phải báo trước với người được phỏng vấn

- Ghi lại cẩn thận các thông tin đã ghi nháp thành báo cáo chính thức

- Gởi báo cáo này cho người được phỏng vấn để họ xác nhận lại các hiểu biết của chúng là chính xác

**Bước 7 : Đánh giá buổi phỏng vấn để rút kinh nghiệm :**

***\* Bản câu hỏi :***

- Hữu hiệu khi cần thu thập thông tin từ nhiều người, giảm thời gian khảo sát

- Nhiều người có thể trả lời thẳng thắn

- Thường có chất lượng không cao do : mẫu thiết kế không tốt, không hiểu rõ câu hỏi, tâm lý điền chiếu lệ, khó tìm hiểu các khái niệm nghiệp vụ mới

- Bản câu hỏi bắt đầu bằng tiêu đề giải thích ngắn gọn mục đích

- Tiếp theo là các chỉ dẫn tổng quát về cách ghi phiếu

- Ngắn gọn và than thiện, hướng dẫn rõ ràng, sắp xếp câu hỏi theo thứ tự hợp lý, tránh các câu hỏi có thể gây hiểu nhầm, hạn chế các câu hỏi mở, hạn chế các câu hỏi nhạy cảm, nhóm các câu hỏi có liên quan vào một phân nhóm

***\* Khảo sát tài liệu :***

- Tìm hiểu các tài liệu liên quan đến hệ thống hiện tại như các quy định của công ty

- Cần có bản sao của các mẫu biểu thực đang sử dụng

- Khảo sát các biểu trống, biểu mẫu đã điền số liệu, các phần mềm đang sử dụng

- Cần lưu ý một số tài liệu đã bị lỗi thời

***\* Quan sát :***

- Rất hữu hiệu để hiểu rõ các quy trình nghiệp vụ

- Nên đặt thêm các câu hỏi thích hợp để hiểu một cách đầy đủ về hệ thống

- Quan sát tất cả các bước xử lý, ghi nhận xem xét tất cả các mẫu biểu, báo cáo

- Trao đổi với từng người để đánh giá tính đầy đủ, kịp thời, chính xác và hữu ích của các mẫu biểu và báo cáo

- Chú ý hiệu ứng Hawthrne : năng suất làm việc thường tăng lên khi một người biết rằng họ đang bị quan sát

***\* Nghiên cứu :***

- Qua báo, tạp chí, sách, internet : các nhà cung cấp phần cứng và phần mềm, các hang cung cấp tin, các nhóm tin, hội nghị chuyên đề, hội thảo, tham gia các nơi đang sử dụng hệ thống tương tự

***\* Lấy mẫu :***

- Là việc chọn lọc có hệ thống các phần tử tiêu biểu trong một tập hợp cần khảo sát

- Có thể sử dụng để chọn người phỏng vấn, gỏi bản câu hỏi, chọn tài liệu khảo sát

- Mẫu thích hợp ( convenience sampling )

- Mẫu theo mục đích ( purpose sampling )

- Mẫu ngẫu nhiên đơn giản ( simple random sampling )

- Mẫu ngẫu nhiên phức tạp ( complex random sampling )

**CHƯƠNG 4 :**

***\* Thực thể ngoại ( external entity ) :***

- Biểu diễn các đối tượng nằm ngoài hệ thống nhưng có tương tác với hệ thống như cung cấp dữ liệu cho hệ thống hoặc nhận kết quả từ hệ thống

- Các thực thể có thể là con người, tổ chức hoặc một hệ thống khác

- Còn gọi là tác nhân đầu cuối : thực thể ngoại cung cấp dữ liệu cho hệ thống là nguồn, thực thể ngoại nhận dữ liệu từ hệ thống là đích, có thể vừa là nguồn vừa là đích

- Ký hiệu là hình chữ nhật

- Tên thực thể ngoại là danh từ mô tả đối tượng tương tác với hệ thống

- Một thực thể ngoại có thẻ được vẽ lại nhiều lần trong sơ đồ để tránh các dòng dữ liệu cắt nhau nên có thêm dấu gạch chéo ở góc hình chữ nhật để phân biệt

Khách hàng

Khách hàng

……………………

***\* Quy trình xử lý ( process ) :***

- Quy trình xử lý biểu diễn cho một công việc xử lý dữ liệu trong hệ thống

- Trong sơ đồ DFD, quy trình xử lý giống như một hộp đen, không cho thấy các chi tiết xử lý bên trong của hệ thống

- Chi tiết xử lý sẽ được trình bày trong phần đặc tả quá trình

- Ký hiệu : hình chữ nhật tròn góc ( hoặc hình tròn ), mỗi quy trình xử lý sẽ có một số nhận dạng gọi là ID, tên quy trình xử lý nên bắt đầu bằng một động từ

***\* Dòng dữ liệu ( data flow ) :***

- Biểu diễn đường đi của dữ liệu từ thành phần này đến thành phần khác trong hệ thống

- Ký hiệu : đường thẳng có mũi tên ở một đầu để biểu diễn chiều di chuyển của dữ liệu, mỗi dòng dữ liệu có một mã số nhận dạng khi sử dụng CASE tool, tên dòng dữ liệu nên là một danh từ thể hiện nội dung dữ liệu di chuyển trên đó

- Ký hiệu mũi tên ở hai đầu chỉ dòng dữ liệu có thể di chuyển cả 2 chiều, với cấu trúc dữ liệu bên trong giống nhau, thường xảy ra với các dòng dữ liệu ra vào kho dữ liệu

- Tên dòng dữ liệu không nên đặt trùng nhau để tránh nhầm lẫn khi mô tả từ điển dữ liệu sau này

- Có thể không ghi tên cho các dòng dữ liệu ra vào kho dữ liệu

***\* Kho dữ liệu ( data store ) :***

- Biểu diễn nơi lưu trữ dữ liệu, để các quy trình xử lý khác nhau trong hệ thống sử dụng

- Kho dữ liệu sau này sẽ được thiết kế để chuyển thành cơ sở dữ liệu của hệ thống

- Ký hiệu hình chữ nhật mở một đầu bên phải hoặc 2 đường song song, một kho dữ liệu có một số nhận dạng gọi là ID, tên kho dữ liệu là một danh từ thể hiện nội dung lưu trữ trong kho

***\* Các bước xây dựng sơ đồ DFD :***

- Từ các tài liệu nghiệp vụ xác định các thực thể ngoại, các kho dữ liệu cần lưu trữ, các chức năng của hệ thống, các dòng dữ liệu quan trọng

- Xây dựng sơ đồ ngữ cảnh ( context diagram )

- Xây dựng sơ đồ DFD mức 0

- Xây dựng sơ đồ DFD ở các mức chi tiết

- Kiểm tra sơ đồ ở các cấp để phát hiện lỗi

- Xây dựng sơ đồ DFD vật lý từ sơ đồ DFD luận lý

- Phân hoạch sơ đồ DFD vật lý

**CHƯƠNG 6 :**

***\* Các lý do để tự phát triển phần mềm :***

- Thỏa mãn các yêu cầu riêng của công ty : sản phẩm đóng gói thường không đáp ứng được

- Hạn chế việc phải thay đổi các quy trình nghiệp vụ hoặc chính sách kinh doanh : nếu thay đổi quá nhiều người ta chọn giải pháp tự phát triển

- Đáp ứng yêu cầu giao tiếp với hệ thống đang tồn tại : ví dụ, hệ thống nhân sự phải giao tiếp được với hệ thống kế toán đang hoạt động

- Phù hợp với công nghệ đang sử dụng tại công ty : đang sử dụng phần cứng, công nghệ cũ, chưa thể nâng cấp, không thể sử dụng các sản phẩm thương mại

- Phát triển nguồn lực nội bộ : tạo lợi thế cạnh tranh, đáp ứng nhanh các tình huống khác nhau

***\* Các lý do để mua phần mềm đóng gói :***

- Chi phí thấp, thời gian triển khai nhanh hơn

- Độ tin cậy và hiệu suất đã được kiểm chứng, đội ngũ nhân viên kỹ thuật cần ít hơn

- Nhận được sự nâng cấp sau này từ phía nhà cung cấp phần mềm

- Chia sẻ được kinh nghiệm từ các công ty khác cùng sử dụng phần mềm này

***\* Các bước để mua gói phần mềm :***

**Bước 1 : Xác định các yêu cầu hệ thống thông tin :**

- Nhận diện các tính năng chính của hệ thống : các CSF phải được thể hiện đầy đủ

- Ước lượng khối lượng hiện tại và sự gia tăng của hệ thống trong tương lai : xác định khối lượng giai dịch và xử lý hiện tại, dự đoán sự thay đổi cho 3 đến 5 năm tới

- Xác định các ràng buộc về phần cứng : ràng buộc về phần cứng hiện tại, các yêu cầu về nâng cấp hoặc đầu tư mới

- Chuẩn bị hồ sơ mời thầu : RFP ( Request For Proposal ) : là văn bản viết chính thức gởi cho các nhà cung cấp, mô tả đầy đủ các tính năng yêu cầu ( thường khi ta chưa xác định sản phẩm hay gói phần mềm nào sẽ dùng ), dựa vào văn bản này các nhà cung cấp sẽ chuẩn bị hồ sơ dự thầu với các giải pháp và sản phẩm của họ, RFQ ( Request For Quotation ) : tương tự như RFP nhưng dùng trong trường hợp ta đã xác định được sản phẩm, chủ yếu yêu cầu chào giá

**Bước 2 : Xác định các nhà cung cấp :**

- Liên hệ với các nhà cung cấp có năng lực : cần lưu ý đối với nhà quản lý đây là một sự đầu tư, cần quan tâm đến khả năng bảo trì và nâng cấp hệ thống sau này

- Gởi RFP ( hoặc RFQ ) cho họ

- Việc xác định các nhà cung cấp có thể dựa vào : các công ty tư vấn về CNTT, những người đã sử dụng các phần mềm tương tự, các báo, tạp chí có uy tín

**Bước 3 : Đánh giá lựa chọn :**

- Đánh giá từng phần hồ sơ dự thầu

- Chấm điểm và xếp hạng cho từng hạng mục, thang điểm đã được lập lúc chuẩn bị hồ sơ mời thầu : xếp hạng theo giá trị tương đối so sánh với mục đó của những người khác, hạng 1,2,3, sử dụng thang tỷ lệ cố định từ 0 đến 10, phương pháp nhị phân, 1 : đáp ứng, 0 : không đáp ứng

- Thường đánh giá trên các tiêu chí : kỹ thuật, quản lý ( kế hoạch, nhân sự, tình huống ), tài chính, thông tin về công ty ( công việc quá khứ )

- Một số việc thực hiện khi đánh giá : tìm hiểu thêm thông tin về những người đã dùng phần mềm này, chạy thử nghiệm chương trình, kiểm tra benchmark

**Bước 4 : Mua phần mềm :**

- Khi mua phần mềm ta không là chủ sở hữu của phần mềm mà chỉ mua bản quyền sử dụng : cần lưu ý license của các phần mềm khác có sử dụng trong phần mềm ta mua

- Có thể thực hiện thuê phần mềm bằng hợp đồng thuê

- Cần quan tâm thương lượng các hợp đồng bảo trì

**Bước 5 : Cài đặt phần mềm :**

- Thời gian cài đặt phụ thuộc vào kích cỡ và độ phức tạp của hệ thống

- Chiến lược cài đặt cần hoạch định trước, tránh không làm ảnh hưởng nhiều đến công việc nghiệp vụ hàng ngày của công ty

- Chuẩn bị đầy đủ các việc : cấu hình hệ thống, đào tạo người sử dụng, chuyển đổi dữ liệu

***\* On – premises :***

- Phần mềm lưu trữ dữ liệu tại chỗ

- Phần mềm được cài đặt và hoạt động từ chính máy chủ và hệ thống máy tính của doanh nghiệp, có thể tận dụng những nguồn tài nguyên máy tính của công ty

- Yêu cầu phải có bản sao phần mềm được cấp phép bởi một nhà cung cấp phần mềm

***\* Ưu điểm của On – premises :***

- Toàn quyền kiểm soát : đây được coi là một trong những lợi ích lớn nhất của On – premises, giải thích cho việc tại sao On – premises được chọn lựa bởi nhiều công ty lớn, giúp bảo vệ dữ liệu riêng của công ty

- Bảo vệ hệ thống tốt hơn : mặc dù Cloud cũng có thể phù hợp với những quy định bảo mật nghiêm ngặt và quyền kiểm soát dữ liệu, tuy nhiên khả năng kiểm soát và tính linh hoạt của On – premises có thể giúp cho các doanh nghiệp trong việc cài đặt các phương thức bảo mật để bảo vệ dịch vụ của họ khỏi sự tấn công từ bên ngoài

***\* Nhược điểm của On – premises :***

- Tốc độ và khả năng mở rộng : cần một kế hoạch thật tốt trước mọi thay đổi vì sẽ tốn nhiều thời gian đánh giá, đặt hàng, triển khai phần cứng cho hạ tầng

- Sự hỗ trợ công nghệ : khi sử dụng On – premises, các công ty thường ít khi nhận được sự trợ giúp về công nghệ, đặc biệt là khi các ứng dụng được thiết kế để đáp ứng những nhu cầu đặc biệt của riêng họ

- Chi phí nâng cấp : On – premises đòi hỏi nhiều chi phí để nâng cấp phần cứng, điều đó là bất cứ sai lầm nào cũng có thể dẫn đến sự tốn kém lớn về chi phí

***\* Cloud – based :***

- Điện toán đám mây, còn được gọi là SaaS ( Software – as – a – Service )

- Là mô hình cấp phép và phân phối phần mềm trên cơ sở người dùng đăng ký sử dụng

- Nhà cung cấp làm chủ phần lắp đặt và giúp khách hàng truy cập được phần mềm thông qua Internet

- Dữ liệu được lưu trữ tập trung tại máy chủ của nhà cung cấp

***\* Ưu điểm của Cloud :***

- Giảm vi phạm bản quyền phần mềm : lợi ích lón nhất của SaaS là nó sẽ giảm hoặc ngăn chặn hẳn sự xâm phạm trái phép về bản quyền

- Giải chi phí hoạt động : chỉ tốn chi phí thuê bao, những chi phí khác liên quan đến nâng cấp sẽ được nhà cung cấp SaaS chịu, công ty cũng tiết kiệm nhiều chi phí cho nhân sự IT để cài đặt và quản lý phần mềm

- Được cung cấp một số tiện ích từ nhà cung cấp : lưu trữ dữ liệu, chia sẻ, hợp tác

- Khả năng mở rộng : được thực hiện bởi nhà cung cấp, chủ doanh nghiệp chỉ lo kế hoạch nâng cấp

***\* Nhược điểm của Cloud :***

- Thiếu sự kiểm soát : phần mềm quản lý ngay tại công ty đem lại mức độ kiểm soát cao hờn vì Cloud được quản lý bởi một bên thứ ba

- Hạn chế trong số lượng các ứng dụng : trong khi SaaS ngày một trở nên phổ biến hơn, vẫn còn những loại ứng dụng mà nhà cung cấp chưa thể đáp ứng được cho người dùng

- Cần mạng để kết nối : SaaS được xây dựng trên nền tảng web vì vậy công ty cần chắc chắn rằng có kết nối mạng ổn định, nếu kết nối bị mất, sẽ không thể truy cập vào phần mềm hay dữ liệu của hệ thống

**CHƯƠNG 7 :**

***\* Các nguyên tắc thiết kế :***

- Hiểu biết đầy đủ các nghiệp vụ : phải hiểu rõ nghiệp vụ, xác định hệ thống sẽ hỗ trợ như thế nào cho từng cá nhân, phòng ban, có thể bắt đầu từ FDD

- Sử dụng tối đa hiệu ứng đồ họa : nhiều người khuyên rằng nên thiết kế tương tự giao diện các ứng dụng của Microsoft vì hiệu quả, dễ học

- Hiểu rõ đối tượng người dùng : thường chia 3 cấp : Beginner, Intermediacate và Expert

- Suy nghĩ theo cách của người dùng

- Sử dụng prototype để đạt được sự chấp nhận của người dùng

- Thiết kế giao diện có tính toàn diện : từ giao diện chính có thể tương tác với các chức năng trong hệ thống

- Tiếp tục quán trình phản hồi, lập tài liệu thiết kế giao diện

***\* Bộ nguyên lý Norman :***

- Được tạo bởi Donald A.Norman trong lúc nghiên cứu về vấn đề tương tác giữa con người với máy tính trong dự án “ Design of Everyday Things ” – DOET

- Thường được các nhà thiết kế sử dụng để tích hợp và đánh giá xem giao diện mà mình thiết kế có thực sự hữu ích và giúp cho việc tương tác giữa người với máy tính dễ dàng hơn hay không

- Hoàn tất những yêu cầu từ phía người dùng

- Sử dụng cả “ Knowledge in the Head ” và “ Knowledge in the World ”

- Đơn giản hóa tác vụ của người dùng, làm mọi thứ rõ ràng nhất có thể

- Thu hẹp khoảng cách thực thi và đánh giá, sắp xếp một cách đúng đắn

- Khai thác sức mạnh từ các biện pháp hạn chế, thiết kế phải tính đến lỗi, chuẩn hóa khi cần thiết

***\* Các kinh nghiệm của Nielsen :***

- Được Jakob Nielsen giới thiệu năm 1995

- Là tập các nguyên lý mà ông đã nghiên cứu và chứng minh rằng sẽ tốt cho một giao diện khi thiết kế theo các quy tắc này

- Các quy tắc do Nielsen đưa ra cũng khá giống với bộ nguyên lý Norman, chỉ khác một chút về cách diễn đạt và cách người ta đánh giá nó

- Phù hợp sự chờ đợi : phù hợp thế giới thực, nhất quán và chuẩn, trợ giúp và tài liệu

- Quản lý lỗi : tránh lỗi, giảm tải những gì người dùng cần nhớ, thông báo, kiểm tra và phục hồi lỗi bị mắc phải

- Người sử dụng làm chủ : người sử dụng điều khiển UI và tự do, trực quan trạng thái hệ thống, mềm dẻo và hiệu quả

- Thiết kế đơn giản : thẩm mỹ và đơn giản

**Các kinh nghiệm của Nielsen :**

***\* Phù hợp với thế giới thực :***

- Nguyên tắc “ Speak the user’s language ”

- Tránh từ chuyên môn, từ chuyên môn có thể sử dụng khi users là chuyên gia trong lĩnh vực

- Khi user được quyền đặt tên trong UI thì họ phải được tự do lựa chọn tên, tránh giới hạn độ dài và nội dung

***\* Nhất quán :***

- Còn gọi là “ Principle of Least Surprise ” : không được làm users ngạc nhiên với cách mà lệnh và đối tượng giao diện hoạt động

- Những gì tương tự phải có hình dáng và hành vi tương tự

- Loại nhất quán quan trọng khác là cách diễn đạt thông qua toàn bộ UI

- Nhất quán về thứ tự lệnh và đối số : thứ tự danh từ - động từ trong GUI : user chọn đối tượng sau đó chọn lệnh tác động trên đối tượng, thứ tự động từ - danh từ : kích hoạt lệnh trước sau đó chọn đối số

- Các loại nhất quán : nhất quán trong : nhất quán ngay trong ứng dụng, nhất quán ngoài : nhất quán giữa các ứng dụng cùng platform, nhất quán ẩn dụ : nhất quán trong việc chọn ẩn dụ

***\* Trợ giúp và tài liệu :***

- Người dùng thường không đọc trợ giúp và tài liệu cho đến thời điểm trước khi sử dụng

- Cẩm nang sử dụng và hướng dẫn trực tuyến là quan trọng

- Trợ giúp cần phải có các khả năng sau : tìm kiếm theo chủ đề, phù hợp ngữ cảnh, hướng nhiệm vụ, cụ thể, ngắn gọn

***\* Người sử dụng làm chủ :***

- Nguyên tắc “ Clearly Marked Exits ”, phải cung cấp khả năng Undo

- Các thao tác cần nhiều thời gian thì phải có khả năng hủy, tất cả các hộp thoại nên có phím Cancel ( kết thúc )

***\* Trực quan trạng thái hệ thống :***

- Nguyên tắc “ Feedback ”, luôn thông báo users về trạng thái của hệ thống

- Các dấu hiệu : thay đổi hình dạng con chạy, highlight, thanh trạng thái và chỉ báo

- Không nên cài đặt dày đặc phản hồi

- Khuyến cáo sử dụng : hành động < 0.1s : cảm giác xảy ra tức thì, hành động xảy ra trong khoảng 0.1 – 1.0s : người sử dụng nhận ra hành động nhưng không cần phản hồi, hành động trong khoảng 1 – 5s: hiển thị busy, hành động kéo dài trên 5s : hiển thị progress bar

***\* Mềm dẻo và hiệu quả :***

- Tạo “Shortcuts”

- Cung cấp các “ đường tắt ” để thực hiện các thao tác thường xuyên : phím lệnh cấp tốc ( Ctrl + C, Ctrl + B, Ctrl + O), viết tắt dòng lệnh, style, bookmarks, lịch sử thực hiện

***\* Tránh lỗi :***

- Tránh lỗi : chọn bằng chuột thay vì nhập bàn phím, ẩn các lệnh không cần thiết

- Các loại lỗi chính : lỗi mô tả : xảy ra khi hai hành động tương tự nhau, lỗi thu hút : khi hai hành động có giai đoạn bắt đầu khởi động như nhau, lỗi phương thức : ví dụ với text editor vi, ở insert mode, các phím ký tự được chèn vào tập tin văn bản

***\* Thông báo lỗi :***

- Chính xác, ví dụ không dùng “ Cannot open file ” mà nên sử dụng “ Cannot file named paper.docx ”, nói bằng ngôn ngữ của người sử dụng

- Đưa ra hỗ trợ mang tính xây dựng : tại sao xảy ra, loại bỏ bằng cách nào

- Lịch sự : không sử dụng “ fatal error ” và “ illegal ”

***\* Giảm tải những gì người dùng cần nhớ :***

- Nguyên tắc “ Mimimize Memory Load ”, sử dụng thực đơn, không sử dụng dòng lệnh

- Sử dụng ComboBox, không sử dụng TextBox nơi chọn lệnh

- Sử dụng những lệnh chung những nơi có thể ( ví dụ Open, Save, Copy Paste )

- Tất cả các thông tin cần thiết phải được nhìn thấy

***\* Thẩm mỹ và đơn giản :***

- Nguyên tắc “ Less is More ”, bỏ đi các thông tin, đặc trưng đồ họa và các tính chất xa lạ, không cần thiết, lựa chọn màu và font phù hợp

- Nhóm bằng dấu cách, gán các điều khiển hợp lý, sử dụng ngôn ngữ phù hợp, bố trí hợp lý

***\* Sự đơn giản :***

- Sự hoàn thiện đạt được khi không còn gì để bổ sung và không còn gì đáng phải bỏ đi

- Nguyên tắc “ When it doubt, leave it out ”

- Nguyên tắc “ Keep it simple, stupid ” ( KISS )

- Các kỹ thuật để đạt được sự đơn giản : sự rút gọn, tính đều, nhiệm vụ kép

***\* Nguyên tắc phối màu 60 – 30 – 10 :***

- Luật 60 – 30 – 10 là công cụ hiệu quả khi bạn sáng tạo các ý tưởng thiết kế nhất là kết hợp với bánh xe màu, nó làm màu sắc trở nên đẹp và hài hòa

- Trang trí một đối tượng với ba màu sắc khác nhau thường tạo ra ấn tượng tốt về thị giác, mỗi màu sắc đã mang trong nó một ý nghĩa và khi kết hợp với nhau tạo ra sự đa dạng

- Với ba màu người ta thường phân ra : màu chủ đạo 60%, màu cấp 2 30%, màu nhấn 10%

***\* Quy trình thiết kế GUI của Galitz :***

- Bước 1 : Nhận biết ai là người sử dụng

- Bước 2 : Hiểu rõ các chức năng nghiệp vụ

- Bước 3 : Hiểu rõ nguyên lý thiết kế màn hình tốt

- Bước 4 : Phát triển thực đơn hệ thống và lược đồ dẫn đường

- Bước 5 : Lựa chọn loại cửa sổ phù hợp

- Bước 6 : Lựa chọn các điều khiển phần cứng phù hợp

- Bước 7 : Lựa chọn các Controls trên màn hình phù hợp

- Bước 8 : Viết text và thông điệp rõ rang

- Bước 9 : Cung cấp phản hồi, hướng dẫn và hỗ trợ hiệu quả

- Bước 10 : Cung cấp khả năng quốc tế hóa và khả năng sử dụng rộng rãi

- Bước 11 : Tạo lập đồ họa, biểu tượng và ảnh có ý nghĩa

- Bước 12 : Chọn màu phù hợp

- Bước 13 : Tổ chức và bố trí cửa sổ và các trang màn hình

- Bước 14 : Kiểm thử hệ thống

**CHƯƠNG 9 :**

***\* Kiến trúc và đặc điểm của hệ thống :***

- Hiệu năng ( Performance ) : định vị các chức năng quan trọng và giảm thiểu giao tiếp

- Bảo mật ( Security ) : sử dụng kiến trúc phân tầng với các phần quan trọng được đặt ở các lớp bên trongs

- An toàn ( Safety ) : định vị các thao tác liên quan đến an toàn trong một số ít các hệ thống con

- Tính sẵn sàng ( Availability ) : có thể thay thế, cập nhật các component mà không phải dừng hệ thống, đảm bảo hệ thống hoạt động liên tục

- Tính dễ bảo trì ( Maintainability ) : dùng các component nhỏ, chi tiết, có thể thay thế được

***\* Kiến trúc DBMS tập trung ( Centralized DBMS ) :***

- Kết hợp tất cả mọi thành phần vào một máy chủ lớn, bao gồm phần mềm DBMS, phần cứng, các chương trình ứng dụng và phần mềm xử lý giao diện người dùng

***\* Kiến trúc File – Server :***

- File – Server kết nối với các máy trạm ( workstation ) qua hệ thống mạng

- Cơ sở dữ liệu nằm trên file – server , DBMS và các ứng dụng chạy trên các máy trạm

- Nhược điểm : nghẽn mạng, có DBMS trên từng máy trạm, việc kiểm soát sự truy cập đồng thời, phục hồi dữ liệu và tính toàn vẹn rất phức tạp

***\* Kiến trúc client – server :***

- Các mô hình hệ thống phân tán chỉ ra cách dữ liệu và các xử lý được phân tán trên nhiều component như thế nào : có thể được cài đặt trên một máy đơn

- Tập hợp các server độc lập cung cấp các dịch vụ cụ thể ví như in ấn, quản trị dữ liệu

- Tập hợp các máy trạm triệu gọi các dịch vụ này

- Hệ thống mạng cho phép người dùng truy cập vào các server

***\* Data Warehouse là gì :***

- Là một cơ sở dữ liệu được thiết kế nhằm vào mục tiêu truy vấn và phân tích hơn là xử lý các giao dịch

- Thường chứa các dữ liệu lịch sử được lấy từ các dữ liệu giao dịch và có thể bao gồm dữ liệu từ nhiều nguồn khác nhau

- Môi trường của data warehouse bao gồm cả giải pháp ETL, OLAP, data mining, các công cụ phân tích phía client và các ứng dụng khác quản lý việc thu thập và chuyển dữ liệu đến người dùng

***\* Các đặc điểm của data warehouse :***

- Subject Oriented : được thiết kế để giúp phân tích dữ liệu, được xây dựng xoay quanh một chủ đề nào đó, ví dụ Sales giúp trả lời các câu hỏi như : các khách hang tốt nhất trong năm là ai

- Integrated : data warehouse sẽ lấy dữ liệu từ nhiều nguồn khác nhau và tích hợp thành một dạng xác định nào đó, phải giải quyết những vấn đề như xung đột về tên, sự không nhất quán giữa các đơn vị đo lường

- Nonvolatile : một khi dữ liệu đã được đưa vào data warehouse thì nó không thể thay đổi, mục đích của data warehouse là giúp ta phân tích những gì đã xảy ra

- Time Variant : để phát hiện các xu hướng trong hoạt động nghiệp vụ cần một lượng dữ liệu lớn, quan tâm đến sự biến động của dữ liệu theo thời gian, khác với hệ thống OLTP do yêu cầu về hiệu suất nên dữ liệu lịch sử thường chuyển sang các kho lưu trữ khác

**CHƯƠNG 10 :**

***\* CMMI ( Capability Maturity Model Integration ) là gì :***

- Là khung ( framework ) mô tả các thành phần chính yếu của một quá trình phát triển phần mềm hiệu quả

- Khi tuân thủ tiêu chuẩn này sẽ làm đơn vị phát triển phần mềm đạt các mục tiêu về chi phí, thời gian, đảm bảo chức năng và chất lượng sản phẩm

***\* Các mức của CMMI :***

Managed (4)

Defined (3)

Optimising (5)

Initial (1)

Repeatable (2)

1. Đến (2) : Disciplined process
2. Đến (3) : Standard, consistent process
3. Đến (4) : predictable process
4. Đến (5) : Continuously improving process

**BÀI TẬP PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**Bài tập về sơ đồ GANTT :**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên công việc | Công việc cần thực hiện trước | Thời gian hoàn thành  ( tuần ) | Phân công nhân sự |
| A: Thực hiện phỏng vấn | - | 3 | … |
| B : Xử lý kết quả phỏng vấn | A | 4 | …. |
| C : Đọc báo cáo của công ty | - | 4 | …. |
| D : Phân tích dòng dữ liệu | B,C | 8 | …. |
| E : Giới thiệu prototype | B,C | 5 | …. |
| F : Xem xét phản ứng về prototype | E | 3 | …. |
| G : Phân tích chi phí / lợi ích | D | 3 | …. |
| H : Chuẩn bị kế hoạch đề xuất | G,F | 2 | …. |
| I : Trình bày kế hoạch đề xuất | H | 2 | …. |

**Bài tập về sơ đồ PERT :**

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Task | Must Follow | Expected Time  ( Days ) |
| Draw data flow | P | - | 9 |
| Draw decision tree | Q | P | 12 |
| Revise tree | R | Q | 3 |
| Write up project | S | R,Z | 7 |
| Organize data dict | T | P | 11 |
| Do output prototype | X | - | 8 |
| Revise ouput design | Y | X | 14 |
| Design database | Z | T,Y | 5 |

**Bài tập về sơ đồ PERT :**

|  |  |  |
| --- | --- | --- |
| Công việc | Công việc trước | Thời gian |
| A | - | 4 |
| B | - | 5 |
| C | - | 1 |
| D | A | 6 |
| E | C | 4 |
| F | B | 2 |
| G | D,F | 15 |
| H | E,F | 14 |
| I | G | 4 |
| J | H | 3 |

**Bài tập về bảng quyết định :**

Ví dụ ( tính chiết khấu ) :

Công ty K có chính sách chiết khấu như sau :

- Tổng giá trị đơn hàng ( công ty khuyến khích đặt đơn hàng lớn thay vì đặt nhiều đơn hàng nhỏ vì dễ phân phối và lập thời biểu vận chuyển ) : tiền chiết khấu là 3% đối với các đơn hàng trên 5000$

- Nếu giao hàng trong vòng 50km sẽ được chiết khấu 2% ( do giảm chi phí phân phối )

- Trong trường hợp đã chiết khấu 3% ( quy tắc 1 ) sẽ chỉ chiết khấu 1% nếu giao hàng trong vòng 50km

- Khách hàng đạt 10000$ ( tổng tiền hàng mua trong vòng 12 tháng qua ) sẽ được chiết khấu thêm 2%

Chính sách chiết khấu của công ty DigiSun :

Đối với các cửa hàng đại lý :

- Đơn hàng nhỏ hơn 2000$ chiết khấu 4%, từ 2000$ đến <5000$ chiết khấu 6%, từ 5000$ trở lên chiết khấu 10%

- Chiết khấu them 5% nếu chỉ giao đến một địa chỉ

- Các đơn >=10000$ đều phải báo cáo giám đốc

Đối với khách khác :

- Đơn hàng > 3000$ chiết khấu 4%

- Giao hàng trong vòng 50km chiết khấu thêm 5%

- Nếu không được hưởng các chiết khấu trên những khách có doanh thu năm > 100000$ sẽ được chiết khấu 3%

- Các đơn => 5000$ đều phải báo cáo giám đốc

**Bài tập về các dạng chuẩn ( xem chi tiết tại chương 8 )**

PhieuNhap ( MSPN, NgayNhap, MSNCC, TenNCC, MSHH, TenHH, SL, ĐG)

Hãy chuyển sang dạng chuẩn 3