

PHẦN 1: LÝ THUYẾT

1. Khái niệm máy tính, phần cứng, phần mềm

- Máy tính: Là thiết bị điện tử hoạt động dưới sự điều khiển của các chỉ thị được lưu trữ trong bộ nhớ
- Phần cứng: Là các linh kiện, thiết bị điện tử cấu tạo nên máy tính
- Phần mềm: Là chương trình cung cấp chức năng gồm các cấu trúc dữ liệu để xử lý thông tin và các tài liệu mô tả chương trình và cách sử dụng.

2. Phân loại phần mềm

- 2.1. Phần mềm hệ thống: điều hành hoạt động của máy tính, thiết bị, chương trình, trợ giúp các tiện ích (tổ chức tệp tin, nén, dọn ổ đĩa,...)
- 2.2. Phần mềm ứng dụng: 7 loại
 - + *Phần mềm thời gian thực*: thu thập, xử lý dữ liệu thế giới thực, yêu cầu chặt chẽ về thời gian: Thời tiết, chứng khoán,...
 - + *Phần mềm nghiệp vụ*: xử lý những vấn đề nghiệp vụ: quản lý nhân viên, quản lý bán hàng,...
 - + *Phần mềm khoa học kỹ thuật*: thực hiện phép tính để xử lý dữ liệu: CAD, CAM, MATLAB,...
 - + *Phần mềm nhúng*: cài sẵn trong các đồ điện tử mà không cần có sự cài đặt của người sử dụng hay người thứ ba: phần mềm nhúng có trong Tivi, tủ lạnh, máy giặt,...
 - + *Phần mềm dựa trên nền web*: là phần mềm liên quan đến các ứng dụng của web (gmail, DHTML,...)
 - + *Phần mềm máy tính cá nhân*: phần mềm trên chạy trên máy tính cá nhân phục vụ các nhiệm vụ, xử lý khác nhau: Microsoft Office, Adobe Photoshop,...
 - + *Phần mềm trí tuệ nhân tạo*: xử lý các vấn đề phức tạp, không liên quan đến tính toán: mạng trí tuệ nhân tạo, xử lý tín hiệu,...
- 2.3. Phần mềm trung gian giữa phần mềm hệ thống và phần mềm ứng dụng.

3. Công nghệ phần mềm

- *Định nghĩa 1*: CNPM là cách tiếp cận có hệ thống, có kỷ luật, và định lượng được có việc phát triển, hoạt động và bảo trì phần mềm.

- *Định nghĩa 2:* CNPM là ngành khoa học nghiên cứu về việc xây dựng phần mềm có chất lượng, có giá thành hợp lý và trong khoảng thời gian hợp lý.

- *Đối tượng nghiên cứu:*

- + Quy trình công nghệ.
- + Phương pháp xây dựng phần mềm.
- + Công cụ hỗ trợ phát triển phần mềm.

4. Quy trình phát triển phần mềm

Lấy yêu cầu → Phân tích, làm rõ yêu cầu → Lập kế hoạch → Thiết kế → Lập trình → Kiểm thử → Bảo trì

5. Ngôn ngữ lập trình

- Là phương tiện giao tiếp giữa con người với máy tính,
- Thực hiện các nhiệm vụ cụ thể
- Truyền đạt yêu cầu và hướng dẫn cho máy tính để thực hiện công việc cụ thể.

→ Là bước cốt lõi trong tiến trình công nghệ phần mềm

6. Nhận thức sai lầm trong lĩnh vực CNPM

- *Sai lầm của người quản lý:*

- + Chỉ cần mua một cái máy tính mới là đủ.
- + Tăng thêm số lượng lập trình viên là sẽ bắt kịp tiến độ.

- *Sai lầm của khách hàng:*

- + Có thể bổ sung các yêu cầu chi tiết sau.
- + Các yêu cầu đều được điều chỉnh trong quá trình phát triển phần mềm vì phần mềm là mềm dẻo.

- *Sai lầm của người phát triển phần mềm:*

- + Chỉ cần viết xong chương trình và chạy được thì coi như đã hoàn thành công việc
- + Chỉ khi nào phần mềm chạy được thì mới có thể đánh giá được chất lượng của nó

7. Những thất bại trong dự án phần mềm

- Giải thích sai, hiểu sai bài toán.
- Bỏ qua các yếu tố ảnh hưởng đến hệ thống.
- Phân tích sai yêu cầu.

- Quy trình thực hiện dự án không đúng, không phù hợp.
- Người dùng thay đổi ý định, môi trường thay đổi.
- Người quản lý không tốt, không có năng lực.
- Đạo đức, tinh thần trách nhiệm của những người tham gia chưa tốt.

8. Đánh giá chất lượng phần mềm:

- *Tính chức năng*: Khả năng đáp ứng được nhu cầu sử dụng: phù hợp, chính xác, an toàn, tương tác
- *Tính tin cậy*: Khả năng hoạt động ổn định trong những điều kiện cụ thể khác nhau: hoàn thiện, sửa lỗi, phục hồi.
- *Tính khả dụng*: Dễ đọc, dễ hiểu, dễ sử dụng, hấp dẫn người sử dụng.
- *Tính hiệu quả*: hoạt động một cách hợp lý, tương ứng với lượng tài nguyên nó sử dụng, tiết kiệm thời gian.
- *Khả năng bảo trì*: Khả năng dễ sửa lỗi, cải tiến, chỉnh sửa gồm: khả năng phân tích, khả năng thay đổi, tính cân bằng, khả năng kiểm định
- *Tính khả chuyển*: Khả năng thích nghi với nhiều môi trường khác nhau.

9. Quá trình phần mềm, Mô hình quá trình phần mềm

- *Quá trình phần mềm*: Là một tập hợp các quá trình trong khi thực hiện dự án và mục tiêu của nó là xây dựng và phát triển phần mềm
- *Mô hình quá trình phần mềm*: Là một khuôn mẫu, mô tả cách các hoạt động, công việc và nhiệm vụ trong quá trình phần mềm được tổ chức, thực hiện và kiểm soát.

10. Mô hình thác nước

- *Quy trình*: Xác định tính khả thi → Phân tích yêu cầu và lập kế hoạch cho dự án phần mềm → Thiết kế hệ thống → Chi tiết hóa thiết kế hệ thống → Lập trình → Kiểm thử và tích hợp các thành phần → Cài đặt phần mềm → Vận hành phần mềm.

- Ưu điểm:

- + Là mô hình đơn giản.
- + Phù hợp với phần mềm nhỏ, đơn giản.
- + Thực hiện các giai đoạn một cách tuyến tính.
- + Quy định tốt tài liệu cho mỗi giai đoạn
- + Kiểm chứng cẩn thận.

- + Được sử dụng rộng rãi
- + Bảo trì thuận lợi
- + Thích hợp khi yêu cầu được hiểu tốt, hệ lớn phức tạp.

- *Nhược điểm:*

- + Tách biệt giữa các giai đoạn
- + Khó tuân thủ tuần tự
- + Khó đáp ứng yêu cầu
- + Chậm có phiên bản thực hiện.
- + Đòi hỏi khách hàng phải kiên nhẫn.
- + phát hiện sai sót muộn là một thảm họa.
- + Quá nhiều kiểm thử.
- + tài liệu hướng dẫn khó hình dung, khó hiểu đối với khách hàng.

❖ *Mô hình thác nước cải tiến:* cho phép quay lui khi phát hiện lỗi trong các giai đoạn phía trước.

- *Được sử dụng khi:*

- + Xác định sản phẩm ổn định, hiểu rõ các vấn đề kỹ thuật.
- + Đội ngũ dự án đã có kinh nghiệm.
- + Xác định rõ yêu cầu từ khách hàng ngay từ đầu.

11. Mô hình mẫu

- *Quy trình:* Thu thập yêu cầu → Thiết kế nhanh → Xây dựng bản mẫu → Khách hàng đánh giá mẫu thử → Làm mịn → Quay lại thiết kế nhanh để điều chỉnh → Xây dựng sản phẩm.

- *Ưu điểm:*

- + Khách hàng được tương tác sớm với hệ thống.
- + Khách hàng và người phát triển được làm việc với nhau.
- + Xác định chính xác được yêu cầu.
- + Phát hiện được những yêu cầu mới, bất ngờ.
- + Thiết kế và phát triển mềm dẻo qua nhiều vòng lặp.

- *Nhược điểm:*

+ Khách hàng có thể cho rằng nguyên mẫu là hệ thống thực dẫn đến mong đợi không được thực tế.

+ Người phát triển có sự lựa chọn không tốt như phù hợp cho nguyên mẫu nhưng có thể không phù hợp cho hệ thống thực

+ Nguyên mẫu không giống hoàn toàn so với hệ thống cuối cùng.

- *Được sử dụng khi:*

+ Hệ thống nhỏ.

+ Chi phí thay đổi không quá lớn.

+ Cần sự cấp bách về thời gian triển khai, đưa vào ứng dụng từng thành phần.

+ Khó đặc tả rõ ràng các yêu cầu ngay từ đầu.

+ Thường sử dụng kết hợp với mô hình thác nước.

12. Mô hình tăng dần

- *Quy trình* : Phân tích → Thiết kế → Mã hóa → Kiểm thử (*Xem tập bài giảng chương 3*)

- *Ưu điểm:*

+ Sau mỗi lần tăng vòng có thể chuyển giao kết quả cho khách hàng nên các chức năng của hệ thống được thấy sớm hơn.

+ Các vòng trước đóng vai trò mẫu thử giúp tìm hiểu thêm các yêu cầu ở những vòng tiếp theo.

+ Những chức năng có thứ tự ưu tiên cao thì sẽ được kiểm thử càng kỹ.

- *Nhược điểm:*

+ Phải xác định chức năng đầy đủ và chức năng hoàn chỉnh trước.

+ Phải xác định rõ các giao tiếp (interface).

+ Thời gian hoàn thành cách biệt nhiều.

+ Kiểm tra khó khăn hơn

+ Đòi hỏi kế hoạch và thiết kế tốt, phân chia công việc hợp lý, các nhân viên phải nghiêm túc và công tác tốt.

- *Được sử dụng khi:*

+ Hiểu rõ được yêu cầu nhưng muốn có sự tiến hóa dần của sản phẩm

+ Cần đưa ra sản phẩm với các chức năng cơ bản trước

- + Áp dụng cho những sản phẩm có thời gian phát triển dài hơn 1 năm.

13. Mô hình phát triển nhanh

- Quy trình:

+ Là quy trình phát triển phần mềm gia tăng, tăng dần từng bước, mỗi chu trình phát triển rất ngắn (60-90 ngày).

+ Gồm một số nhóm, mỗi nhóm làm 1 RAD theo các pha: Mô hình nghiệp vụ, Mô hình dữ liệu, Mô hình xử lý, Tạo ứng dụng, Kiểm thử và đánh giá.

- Ưu điểm:

- + Thời gian phát triển giảm nhờ công cụ.
- + Cần ít người phát triển hơn.
- + Nhanh chóng hình dung ra sản phẩm.
- + Sử dụng hiệu quả các framework và công cụ đóng gói.
- + Giảm thiểu rủi ro.

- Nhược điểm:

- + Đối với dự án lớn, cần nhiều người làm việc.
- + Đòi hỏi người phát triển lẫn khách hàng phải làm việc tận tâm, tận lực.
- + Nếu thiếu sự nhiệt tình thì sẽ thất bại.
- + Chỉ phù hợp với một số ứng dụng.
- + Không hợp với các dự án có rủi ro cao.

- Được sử dụng khi:

- + Hệ thống dễ phân chia module và có thể mở rộng.
- + Yêu cầu được biết rõ và hợp lý
- + Người dùng có thể tham gia tốt qua toàn bộ chu kỳ.
- + Thời gian phát triển ngắn (dưới 60 ngày).
- + Áp dụng đối với những hệ thống nhỏ không có tính nghiêm ngặt.

14. Mô hình xoắn ốc

- Quy trình:

- + Xây dựng theo nhiều chu kỳ, càng về sau thì phần mềm càng hoàn chỉnh.

+ 6 vùng công việc: Giao tiếp khách hàng, Lập kế hoạch, Phân tích rủi ro, Kỹ nghệ, Xây dựng và xuất xưởng, Khách hàng đánh giá

- Ưu điểm:

- + Các phiên bản được hoàn thiện dần theo chiều xoắn ốc từ trong ra ngoài
- + Hạn chế rủi ro sớm.
- + Nhận được phản hồi sớm từ khách hàng.
- + Giảm rủi ro về chi phí.

- Nhược điểm:

- + Tốn nhiều chi phí
- + Không thích hợp với nhiều dự án nhỏ và rủi ro thấp.
- + Mô hình phức tạp, khó sử dụng.
- + cần kiến thức chuyên sâu
- + Khó quản lý tiến trình và thuyết phục khách hàng.

- Được sử dụng khi:

- + Phần mềm có quy mô lớn, dự án lớn, phức tạp có khả năng kiểm soát rủi ro
- + Những dự án có rủi ro từ trung bình đến cao
- + Những dự án cần nhiều thời gian và có thể gặp nhiều rủi ro
- + Các yêu cầu chưa xác định rõ ràng.

❖ **Mô hình xoắn ốc WinWin**

- Ưu điểm:

- + Khắc phục được nhược điểm của mô hình xoắn ốc ban đầu
- + Là mô hình tiến trình hướng rủi ro, luôn đánh giá rủi ro
- + Khách hàng có phần mềm thỏa mãn yêu cầu.
- + Người phát triển có kinh phí thỏa đáng và thời gian hợp lý

- Các giai đoạn trong chu kỳ:

- + Xác định những người liên quan đến hệ thống và yêu cầu của từng người dùng.
- + Xác định điều kiện thỏa mãn yêu cầu.
- + Lên kế hoạch.

- + Xác định rủi ro.
- + Định nghĩa mức độ tiếp theo của sản phẩm và quá trình.
- + Thông qua định nghĩa của sản phẩm và quá trình.
- + Xem lại và nhận xét.

15. Yêu cầu chức năng nghiệp vụ

- Các chức năng của phần mềm tương ứng với công việc có thật trong thế giới thực
- Có 4 chức năng chính:
 - + Lưu trữ.
 - + Tra cứu.
 - + Tính toán.
 - + Kết xuất.

16. Yêu cầu chức năng hệ thống

- Là các chức năng phải phát sinh thêm khi tiến hành các công việc trên máy tính.
- Không tương thích với các công việc trong thế giới thực
- *Một số chức năng thông dụng:*
 - + Phân quyền sử dụng giữa các người dùng.
 - + Sao lưu, backup, phục hồi thông tin.
 - + Định cấu hình thiết bị, ngày giờ làm việc.
 - + Mô phỏng hoạt động ở thế giới thực
 - + Báo động, nhắc nhở người dùng.

17. Tài liệu SRS

- *Khái niệm:* Là tài liệu chỉ mô tả về chức năng và các ràng buộc (WHAT); Không mô tả về phương thức cài đặt; Không phải là tài liệu thiết kế.
- Một tài liệu SRS cần thể hiện được các thành phần sau:
 - + Các yêu cầu chức năng.
 - + Các yêu cầu phi chức năng.
 - + Các ràng buộc khi thiết kế và cài đặt.
 - + Giao diện.

18. Tại sao phải ước lượng năng lực

Để xác định thời gian, chi phí và nguồn lực để thực hiện dự án phần mềm.

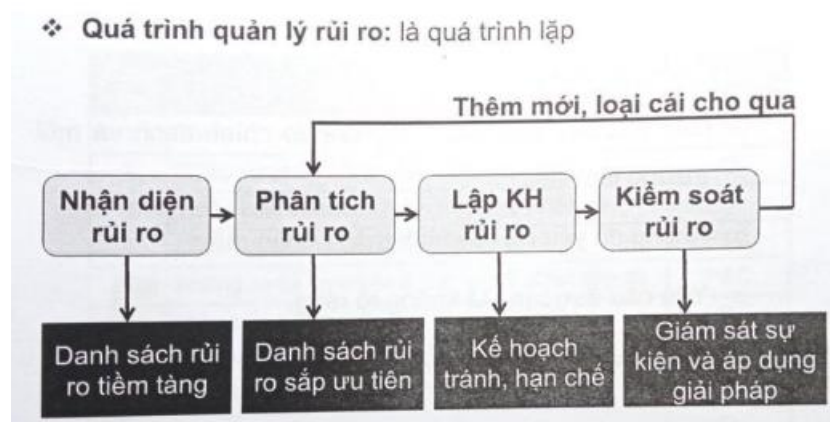
19. Khái niệm rủi ro

Rủi ro dùng để chỉ một hay nhiều sự việc chưa nhưng có khả năng xảy ra trong tương lai tác động đến dự án, và khi sự việc đó xảy ra thường xuyên sẽ gây ảnh hưởng lớn đến giá thành, chất lượng và thời gian hoàn thành dự án.

18. Quản lý rủi ro

- Quản lý rủi ro là quá trình phát hiện, phân tích, xử lý và kiểm soát các rủi ro có thể có trong một dự án
- Quản lý rủi ro không đảm bảo sự thành công mà chỉ làm tăng khả năng thành công của dự án.
- Đảm bảo rằng sự ảnh hưởng của rủi ro đến giá thành, chất lượng và thời gian hoàn thành dự án là tối thiểu.

19. Quá trình quản lý rủi ro



20. Top 10 rủi ro

- 1. Thiếu kỹ năng quản lý dự án hiệu quả.
- 2. Thiếu sự phân công trách nhiệm quản lý.
- 3. Thiếu kiến thức và kỹ năng quản lý nhân sự.
- 4. Quản lý sự thay đổi không tốt.
- 5. Không có kế hoạch hoặc kế hoạch chưa đầy đủ.
- 6. Hiểu sai yêu cầu.
- 7. Thời hạn hoàn thành.

- 8. Người dùng thiếu tinh thần trách nhiệm.
- 9. Thiếu yêu cầu rõ ràng.
- 10. Người quản lý thiếu kỹ năng ứng xử.

21. Kiến trúc phần mềm

Kiến trúc phần mềm của một chương trình là cấu trúc kiến tạo nên hệ thống phần mềm bao gồm:

- + Các thành phần của phần mềm.
- + Các thuộc tính của từng thành phần
- + Mối liên hệ giữa các thành phần

22. Vai trò của kiến trúc phần mềm

- Kiến trúc phần mềm không phải là mô tả hoạt động, mà là mô hình phân hoạch theo những cách nhìn khác nhau (chức năng, tiến trình, tĩnh hay động,...)
- Giúp kỹ sư hệ thống:
 - + Phân tích khả năng đáp ứng yêu cầu hệ thống
 - + Nghiên cứu các giải pháp thay thế.
 - + Giảm rủi ro của việc phát triển phần mềm
 - + Sử dụng lại.

23. Tài liệu kiến trúc phần mềm

- Một tài liệu mô tả kiến trúc phần mềm có các thành phần sau:
 - + Ngữ cảnh của hệ thống và kiến trúc
 - + Mô tả khung hình kiến trúc.
 - + tài liệu thể hiện mối quan hệ giữa các khung hình.

24. Các tiêu chí đánh giá kiến trúc phần mềm

- Khả năng sửa lỗi.
- Khả năng thao tác
- Độ tin cậy.
- Khả năng mang chuyển.
- Khả năng sẵn sàng.

- Sự bảo mật.
- Khả năng tái sử dụng.

25. Khái niệm đối tượng

- Là khái niệm mô tả các sự vật/thực thể trong thế giới thực
- Cho phép người phát triển phần mềm biểu diễn các khái niệm thế giới thực trong thiết kế phần mềm của họ.
- Các đối tượng duy trì các quan hệ giữa chúng.

❖ Tính chất của đối tượng:

Đối tượng gồm: Trạng thái (tập hợp thuộc tính) + Hành vi (tập hợp phương thức) + Định danh.

26. Khái niệm lớp

Là tập hợp các đối tượng có cùng thuộc tính và phương thức.

27. Khái niệm hướng đối tượng

- Lấy đối tượng làm trung tâm
- Hệ thống bao gồm tập hợp các đối tượng và quan hệ giữa các đối tượng
- Các đối tượng trao đổi bằng thông điệp.
- Không sử dụng biến toàn cục.

28. Tính trừu tượng

Nhóm các đối tượng có các thuộc tính tương tự nhau thành một lớp. Loại bỏ các thuộc tính cá biệt, chỉ giữ lại các thuộc tính chung nhất.

29. Tính bao đóng

Gộp các thuộc tính và phương thức trong một đối tượng, đồng thời giới hạn các truy xuất các thuộc tính đó → Quy định phạm vi truy cập cho các thành viên.

30. Tính kế thừa

- Tạo lớp mới dựa trên lớp cũ.
- Tái sử dụng thuộc tính, phương thức.

31. Tính đa hình

- Khả năng đại diện cho công việc
- Giao việc cho cha, con có thể thay thế.
- Gọi cùng một phương thức nhưng ở các đối tượng khác nhau sẽ cho ra kết quả khác nhau.

32. Ước lượng năng lực theo hướng tiếp cận dạng Top – Down

- *Quy trình*: Xác định bản chất chức năng và áp dụng của chức năng → Xác định kích thước của dự án sẽ được ước lượng năng lực.

- *Ưu điểm*:

- + Nhanh.
- + Không cần xác định hoạt động chi tiết.
- + Ít tốn chi phí.
- + Có thể dự đoán ở mọi mức độ chi tiết của dự án.
- + Cho thấy tổng chi phí dự kiến.

33. Ước lượng năng lực theo hướng tiếp cận dạng Bottom – Up

- *Quy trình*: Xác định module cần phát triển và phân loại → Xác định năng lực viết code trong bình cho các module → Tính tổng năng lực cần → Điều chỉnh giá trị ước lượng

- *Ưu điểm*:

- + Chính xác
- + Có sự đóng góp của đội dự án.
- + Có sự phân tích chi tiết.
- + Cung cấp khả năng quản lý, điều khiển dự án hiệu quả.

34. Ngôn ngữ lập trình cấu trúc

- Ngôn ngữ lập trình cấu trúc là một phương pháp lập trình có các đặc trưng cơ bản sau:

- + Tuần tự
- + Chọn, rẽ nhánh
- + Lặp

- Ví dụ: Pascal, C, PHP, ...

35. Kiểm thử phần mềm

- *Khái niệm*: Là quá trình thực thi phần mềm với mục tiêu tìm ra lỗi, đảm bảo phần mềm chạy đúng.

- *Nội dung của kiểm thử* gồm:

- + Kế hoạch kiểm thử.
- + Phương pháp và chiến lược kiểm thử

- + Các chiến lược kiểm thử.
- + Các kỹ thuật kiểm thử.

36. Sai sót (Error)

- Chỉ sự không nhất quán giữa các giá trị tính toán, quan sát, đo đạc với giá trị thực hay về mặt lý thuyết.
- Error là một đại lượng dùng để đo sự khác nhau giữa giá trị kết quả đầu ra thực tế của chương trình với giá trị chính xác mà người ta mong đợi
- Error còn dùng để chỉ hành động của con người tạo ra chương trình có chứa lỗi.

37. Lỗi (Fault, Defect, Bug)

- Chỉ điều kiện gây nên sự thất bại của hệ thống trong việc thực hiện chức năng được yêu cầu.
- Là nguyên nhân cơ bản dẫn đến các hoạt động sai của hệ thống.

38. Hỏng hóc (Failure)

- Chỉ sự bất lực trong việc thực hiện chức năng của hệ thống
- Sự thất bại của hệ thống xảy ra khi thực hiện một chức năng yêu cầu thì hành động của hệ thống khác sự mong đợi
- Sự thất bại có thể do các yếu tố chức năng hoặc phi chức năng.

38. Các mức độ kiểm thử

- Kiểm thử đơn vị:

- + Là mức độ kiểm tra từng thành phần
- + Do lập trình viên tạo ra
- + Các test được tạo ra từ kinh nghiệm của lập trình viên.

- Kiểm thử tích hợp:

- + Là mức độ kiểm tra các thành phần được tích hợp lại với nhau thành các hệ thống con
- + Kiểm tra các module có được ghép nối phù hợp không và kiểm tra giao tiếp giữa chúng

- Kiểm thử hệ thống:

- + Là mức độ kiểm thử toàn bộ hệ thống.
- + kiểm tra hoạt động của toàn bộ hệ thống phần mềm sau khi kết nối các hệ thống con lại với nhau
- + Kiểm tra dựa vào tài liệu đặc tả yêu cầu.

- *Kiểm thử chấp nhận:*

- + Là mức độ kiểm tra chấp nhận.
- + Kiểm tra xem hệ thống có thỏa mãn yêu cầu của khách hàng không.
- + Kiểm tra trên các dữ liệu thật.

39. Kiểm thử hộp đen

- Còn được gọi là kiểm thử chức năng
- Là kiểm tra chỉ cần biết dữ liệu đầu vào, kết quả mong đợi mà không tham khảo cấu trúc bên trong của thành phần hoặc hệ thống.
- Thường áp dụng cho những module lớn

40. Kiểm thử hộp trắng

- Còn gọi là kiểm thử cấu trúc
- Là kiểm tra dựa trên sự phân tích cấu trúc bên trong của thành phần hoặc hệ thống.
- Thường áp dụng cho những module nhỏ.