

CÂU HỎI ÔN TẬP KIỂM TRA GIỮA KỲ NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Biên soạn: GVC. ThS. Hoàng Thị Hà

Bộ môn: CNPM – Khoa CNTT

Email: htha@vnua.edu.vn

CHƯƠNG 1: GIỚI THIỆU CHUNG

1. Hãy trình bày những khái niệm cơ bản của lĩnh vực công nghệ phần mềm:
 - a. Công nghệ phần mềm (Software engineering)
 - b. Khoa học máy tính (Computer Science)
 - c. Phần mềm chuyên nghiệp (professional software)
 - d. Tiến trình phần mềm (software processes)
2. Sự khác nhau giữa công nghệ phần mềm (software Engineering) và khoa học máy tính (computer science) là gì?
3. Sự khác nhau giữa công nghệ phần mềm (software Engineering) và công nghệ hệ thống (software Engineering) là gì?
4. Thách thức mà ngành công nghệ phần mềm đang phải đối mặt là gì?
5. Tại sao phần mềm chuyên nghiệp không chỉ là chương trình máy tính?
6. Thế nào là phần mềm chung (generic software)
7. Thế nào là phần mềm đặt hàng (custom software)
8. Sự khác nhau quan trọng giữa việc phát triển phần mềm dùng chung (generic software) và phần mềm đặt hàng (custom software) là gì?
9. Bốn thuộc tính quan trọng mà bất cứ 1 phần mềm chuyên nghiệp (professional software) nào cũng cần phải có là gì? Gợi ý thêm 4 thuộc tính khác mà các phần mềm nên có.
10. Có bao nhiêu loại ứng dụng phần mềm? Là những loại ứng dụng nào?
11. Stand-alone applications là gì?
12. Ứng dụng loại Interactive transaction-based applications là gì?
13. Ứng dụng loại Embedded control systems là gì?

14. Ứng dụng loại Batch processing systems là gì?

15. Entertainment systems là gì?

16. Systems for modeling and simulation là gì?

17. Data collection systems là gì?

18. Systems of systems là gì?

(Sinh viên tham khảo thêm link:

1 <http://www.cs.unb.ca/~wdu/cs3043/a1ans.htm>

2. <http://sce2.umkc.edu/BIT/burris/pl/software-process/qanda.html>

CHƯƠNG 2: QUY TRÌNH PHẦN MỀM (Software processes)

1. Trình bày những pha(giai đoạn) chính trong vòng đời phần mềm (software life-cycle) và mô tả ngắn gọn các hoạt động và sản phẩm tạo ra của mỗi pha.
2. Sự khác nhau giữa software life-cycle và software process model là gì?
3. Tiến trình phần mềm(Software processes) là gì? Có những kiểu mô hình tiến trình chung nào?
4. Trình bày mô hình tiến trình thác nước(waterfall model). Ưu nhược điểm của mô hình? Với những kiểu dựa án nào ta nên sử dụng mô hình thác nước. Hãy giải thích tại sao?
5. Trình bày mô hình tiến trình phát triển tăng dần (incremental development). Ưu nhược điểm của mô hình? Với những kiểu dựa án nào ta nên sử dụng mô hình này. Hãy giải thích tại sao?
6. Trình bày mô hình tiến trình tái sử dụng (reuse-oriented development). Ưu nhược điểm? Với những kiểu dựa án nào ta nên sử dụng mô hình này.
7. Giải thích tại sao phát triển tăng dần là cách tiếp cận hiệu quả nhất để phát triển hệ thống phần mềm nghiệp vụ (business software systems). Tại sao mô hình này ít phù hợp với công nghệ hệ thống thời gian thực?
8. Mô hình tiến trình phần mềm nào là phù hợp khi thực hiện "Thiết kế giao diện người dùng"?
9. Thế nào là các tiến trình theo kế hoạch (Plan-driven processes) ? Loại dự án phù hợp với tiến trình này?
10. Ưu điểm của phát triển phần mềm theo hướng tái sử dụng (Reuse-oriented software engineering) là gì?
11. Đưa ra các nguyên nhân cho câu trả lời của bạn dựa trên loại hệ thống được phát triển, đề xuất mô hình quy trình phần mềm chung phù hợp nhất để quản lý sự phát triển của các hệ thống :
 - a) Hệ thống điều khiển phanh bó cứng trong xe hơi
 - b) Hệ thống kế toán mới một trường đại học thay thế cho phần mềm kế toán đang tồn tại
 - c) Hệ thống lập kế hoạch du lịch tương tác giúp người dùng lập kế hoạch hành trình du lịch
12. CASE là gì? Nêu tầm quan trọng của CASE trong phát triển phần mềm.

Tham khảo : <http://sce2.umkc.edu/BIT/burrise/pl/software-process/qanda.html>

CHƯƠNG 3: PHÁT TRIỂN PHẦN MỀM LINH HOẠT

(Agile software development)

1. Thế nào là “Rapid software development processes” ?
2. Các đặc điểm cơ bản của tiếp cận “Rapid software development processes”
3. Thế nào là phương pháp phát triển phần mềm agile (“Agile methods”)?
4. Hãy trình bày tuyên ngôn của các phương pháp Agile?
5. Các phương pháp Agile được đề xuất vào thời gian nào?
6. Những loại phần mềm nào nên sử dụng Các phương pháp Agile? Vì sao?
7. Phân biệt phát triển phần mềm theo “plan-driven” và phát triển phần mềm theo agile?
8. Mô hình Phát triển phần mềm tăng dần có sử dụng được agile không?
9. Hãy kể tên các phương pháp phát triển phần mềm Agile
10. Những yếu tố nào cần quan tâm trước khi quyết định sử dụng phương pháp phát triển phần mềm theo agile?
11. Phương pháp “Extreme programming (XP)” được đề xuất năm nào?
12. Nêu đặc điểm chính của Extreme programming (XP)?
13. Trình bày chu trình phát hành sản phẩm “The extreme Programming”
14. Thế nào là “Test-first development”?
15. Thế nào là “Pair programming” ?
16. Ưu điểm của « Pair programming”?
17. Các đặc điểm chính của pha kiểm thử (Test) trong XP
18. Tìm hiểu SCRUM?
19. Trình bày quy trình phát triển phần mềm theo SCRUM
20. Anh/chị hiểu thế nào về Product Backlog, Sprint trong SCRUM,
21. Các cuộc họp hằng ngày trong SCRUM có đặc điểm gì?
22. Khảo sát và mô tả ngắn các công cụ hỗ trợ các phương pháp Agile?
23. Tại sao các phương pháp Agile lại khó cho việc bảo trì

Chương 4: YÊU CẦU PHẦN MỀM

(Requirement Software)

1. Yêu cầu phần mềm là gì ?
2. Kỹ nghệ yêu cầu(requirements engineering (RE) là gì ?
3. Phân biệt giữa yêu cầu người dùng (User requirements) và yêu cầu hệ thống(system requirements). Lý do vì sao các yêu cầu phần mềm cần phải mô tả ở 2 mức khác nhau
4. Phân biệt yêu cầu chức năng(Functional requirements) và yêu cầu phi chức năng(non-functional requirements). Lấy ví dụ minh họa.
5. Trình bày các loại yêu cầu phi chức năng
6. Hiểu thế nào về yêu cầu phi chức năng đối với sản phẩm(*Product requirements*)
7. Hiểu thế nào về yêu cầu phi chức năng đối với tổ chức (*Organizational requirements*)
8. Hiểu thế nào về yêu cầu phi chức năng bên ngoài (*External requirements External requirements*)
9. Thế nào là tài liệu yêu cầu phần mềm(software requirements document, software requirements specification or SRS)
10. Trình bày cấu trúc của tài liệu SRS theo chuẩn IEEE, 1998
11. Trình bày cấu trúc mô tả 1 yêu cầu chức năng chuẩn IEEE, 1998(xem trang 98, TA phiên bản 9)
12. Hãy mô tả yêu cầu chức năng Tra cứu điểm theo mẫu trên
13. Trình bày tiến trình xác định yêu cầu
14. Trình bày tiến trình khám phá và phân tích yêu cầu.
15. Giải thích từng giai đoạn trong quá trình khám phá và phân tích yêu cầu
16. Khám phá và hiểu yêu cầu từ các (stakeholders) là một quá trình khó vì sao?
17. Để khám phá yêu cầu có những cách tiếp cận nào?
18. Thế nào là “Requirements validation ». Trình bày các loại kiểm tra yêu cầu
19. Tại sao cần phải quản lý yêu cầu ?

Chương 5: THIẾT KẾ PHẦN MỀM

1. Trình bày các hoạt động chính trong quy trình thiết kế phần mềm và kết quả đầu ra của các hoạt động. Mối quan hệ giữa các hoạt động.
2. Thế nào là thiết kế kiến trúc phần mềm(Architectural design)
3. Thế nào là thiết kế dữ liệu (data design)?
4. Thế nào là thiết kế thành phần(component design)
5. Khi mô tả 1 hệ thống phần mềm, tại sao phải thiết kế kiến trúc hệ thống trước khi hoàn tất đặc tả yêu cầu?
6. (Sinh viên tham khảo thêm
https://www.academia.edu/19170420/p2_software_engineering)

CHƯƠNG 6 : KIỂM THỬ PHẦN MỀM

1. Hãy nêu khái niệm về kiểm thử phần mềm ?
2. Mục tiêu của kiểm thử phần mềm là gì ?
3. Thế nào là Validation Testing”
4. Thế nào là Verification Testing
5. Trình bày mô hình tiến trình kiểm thử phần mềm (A model of the software testing process)
6. Các phần mềm thương mại, thường phải đi qua các giai đoạn kiểm thử nào?
7. Development testing là gì? Ai là người thực hiện kiểm thử ở giai đoạn này
8. Release testing là gì? Ai là người thực hiện kiểm thử ở giai đoạn này ?
9. User testing là gì Ai là người thực hiện kiểm thử ở giai đoạn này ?
10. Unit testing là gì ? Ai thực hiện kiểm thử, dữ liệu kiểm thử là dữ liệu thật hay dữ liệu giả lập ?
11. *Component testing* testing là gì ? Ai thực hiện kiểm thử, dữ liệu kiểm thử là dữ liệu thật hay dữ liệu giả lập ?
12. *System testing* là gì ? Ai thực hiện kiểm thử, dữ liệu kiểm thử là dữ liệu thật hay dữ liệu giả lập ?
13. Thế nào là kiểm thử tự động(automated testing)
14. Hãy kể tên một vài công cụ hỗ trợ kiểm thử tự động phổ biến?
15. Test cases là gì?

16. Hãy nêu cấu trúc của test case?
17. Thế nào là kiểm thử hộp đen (black box testing)
18. Thế nào là kiểm thử hộp trắng (White box testing)
19. Thế nào là “*Alpha testing*”
20. Thế nào là « *Beta testing* »?
21. Thế nào là “*Acceptance testing*”
22. Kiểm thử có thể khẳng định chương trình không có lỗi không ?