**LÝ THUYẾT NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

**Vũ Trọng Khôi - B22DCCN468**

**1. Thế nào là corrective maintenance?**

\* Corrective maintenance - Bảo trì khắc phục: loại bỏ các lỗi còn sót lại mà không thay đổi các yêu cầu kĩ thuật, thành phần chi tiết cũng như nâng cấp.

**2. Thế nào là adaptive maintenance?**

\* Adaptive maintenance - Bảo trì thích nghi: sửa chữa khi có sự thay đổi của môi trường mà phần mềm đang chạy, như phần mềm, phần cứng, sách hoạt động.

**3. Thế nào là perfective maintenance?**

\* Perfective maintenance - Bảo trì hoàn thiện: tu chỉnh để cải thiện tính hiệu quả của sản phẩm, như là thêm chức năng hoặc giảm thời gian phản hồi

**4. Thế nào là refactoring?**

\* Refactoring - Tái cấu trúc: thay đổi thiết kế sản phẩm khi sản phẩm đang được tạo ra, cải tiến và làm tốt hơn chất lượng mã nguồn. Không làm thay đổi chức năng chính, chức năng chung của ứng dụng.

**5. Thế nào là "from scratch"?**

\* From scratch là bắt đầu xây dựng phần mềm từ mốc không có gì, không dựa trên một cơ sở, code có sẵn hay tái sử dụng modul cũ mà là làm mới hoàn toàn

**6. Thế nào là moving target problem?**

\* Moving-target problem là vấn đề xảy ra khi đang phát triển phần mềm: Các yêu cầu của phần mềm đó thay đổi, gây ra nhiều khó khăn cho đội phát triển vả dễ tạo thêm regression fault vấn đề thay đổi mục tiêu.

**7. Thế nào là regression fault?**

\* Regression fault - Lỗi hồi quy: lỗi vô tình được tạo ra một phần của sản phẩm khi ta thay đổi, sửa lỗi một phần khác mà không biết sự kết nối giữa 2 phần đó. Từ đó khiến một phần của sản phẩm không hoạt động.

**8. Thế nào là một episode?**

\* Episode là khoảng thời gian một sản phầm trải qua một vòng đời, lặp lại pha.

**9. Thế nào là một iteration(lặp đi lặp lại)?**

\* Iteration là một chu kì phát triển trong mô hình phát triển lặp, mỗi vòng lặp có thể tạo ra một phiên bản hoàn thiện có thể chạy được. Mỗi lần lặp sản phẩm sẽ dần hoàn thiện, đi tới gần mục tiêu cầu đạt cho tới khi có sản phẩm thỏa mãn mục tiêu của sự lặp

**10. Thế nào là một increasement?**

\* Increasement - Tăng trưởng là quá trình phát triển phần mềm theo từng phần một, mỗi phần phải qua nhiều vòng interation. Sau mỗi vòng, phần mềm sẽ tăng trưởng cả về số lượng (số lượng modul tăng dần) và cả chất lượng (sửa lỗi các modul trước đó)

**11. Thế nào là một artifact?**

\* Artifact - Đồ tạo tác là một bộ phận của một sản phẩm phần mềm như: tài liệu, usecase, hướng dẫn, sơ đồ lớp,... giúp mô tả chức năng, kiến trúc và thiết kế của phần mềm.

- Mỗi artifact khác nhau liên quan với quy trình riêng của nó.

**12. Thế nào là portability?**

\* Portability - Tính khả chuyển của phần mềm là mức độ dễ dàng trong việc chỉnh sửa phần mềm để có thể chạy được trong một hệ thống, môi trường, phần cứng khác mà không phải code lại từ đầu.

**13. Thế nào là reusebility?**

\* Reusebility - Tính sử dụng lại của phần mềm là khả năng sử dụng lại các thành phần của một sản phẩm để phát triển một sản phẩm khác với chức năng, mục đích khác.

**14. Thế nào là milestone?**

\* Milestone là các điểm mốc quan trọng trong phát triển sản phẩm, đánh dấu việc hoàn thành giai đoạn quan trọng hoặc đạt được một mục tiêu.

Có thể là thời điểm khách hàng nhận được sản phẩm hay thời điểm một thành phần của sản phẩm được coi như hoàn thành.

**15. Thế nào là một story?**

\* Story là một tính năng mà khách hàng muốn có cho sản phẩm của mình

**16. Thế nào là refactoring? Câu 4**

**17. Thế nào là concept exploration?**

\* Concept exploration - Tìm hiểu miền ứng dụng: xác định rõ lĩnh vục ứng dụng của phần mềm, là quá trình điều tra để biết, hiểu được những nhu cầu của khách hàng.

**18. Thế nào là business model?**

**19. Thế nào là traceability?**

**\* Traceability - Khả năng truy vết: truy ngược, theo dõi và liên kết giữa các yêu cầu, thiết kế, mã nguồn, kiểm thử để đảm bảo sản phẩm đúng với yêu cầu bàn đầu.**

**20. Thế nào là egoless programming?**

Lập trình không cái tôi là một triêt lí phát triển phần mềm với các nguyên tắc:

* Mỗi lập trình viên khuyến khích đồng đội tìm ra lỗi trong mã nguồn của mình
* Người kiểm tra lỗi nên đưa ra lời khuyên thay vì chế giễu
* Việc tìm ra lỗi là bình thường, là tất yếu trong quá trình phát triển phần mềm.
* Cả đội cùng nhau xây dựng tình năng, sản phẩm và sản phẩm là của cả đội

**21. Thế nào là PM?**

**22. Thế nào là technical leader?**

**23. Thế nào là programming secrectary?**

**24. Thế nào là backup programmer?**

**25. Thế nào là supper programmer?**

**26. Thế nào là một bản thiết kế còn ommision?**

**27. Thế nào là một bản thiết kế còn contradiction?**

**28. Thế nào là một phần mềm COTS?**

**29. Thế nào là SPMP?**

**30. Thế nào là alpha release?**

**31. Thế nào là beta release?**

**32. Thế nào là process?**

**33. Thế nào là workflow?**

**34. Luật Miller trong CNPM nói gì?**

**35. Luật Brooks trong CNPM nói gì?**

**36. Luật Dijkstra trong CNPM nói gì?**

**37. Verification và Validation (V&V) là gì?**

**38. Thế nào là inspection?**

**39. Thế nào walkthrough?**

**40. Thế nào là một moderator trong nhóm inspection? 41. Thế nào là một recorder trong nhóm inspection?**

**42. Mô hình CMM là gì?**

**43. Thế nào là test performance?**

**44. Thế nào là test robustness?**

**45. Thế nào là coin of uncertainty?**

**46. Thế nào là norminal effort?**

**47. Thế nào là phần mềm organic?**

**48. Thế nào là phần mềm embeded?**

**49. Thế nào là phần mềm semi-detached?**

**50. Thế nào là TCF?**

**51. Thế nào là UFP?**

**52. Thế nào là flow trong FFT?**

**53. Thế nào là process trong FFP?**

**54. Tại sao không có pha kiểm thử?**

**55. Tại sao không có pha làm tài liệu?**

**56. Tại sao không có pha lập kế hoạch?**

**57. Nếu không áp dụng các mô hình vòng đời phần mềm thì có phát triển được phần mềm không? Tại sao?**

**58. Tại sao người ta phải dùng nhiều mô hình vòng đời khác nhau để phát triển phần mềm? 59. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu thác nước? 60. Mô hình vòng đời phần mềm kiểu thác nước thì phù hợp với những dự án có đặc điểm gì? 61. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu bản mẫu nhanh? 62. Mô hình vòng đời phần mềm kiểu bản mẫu nhanh thì phù hợp với những dự án có đặc điểm gì?**

**63. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu lặp và tăng trưởng? 64. Mô hình vòng đời phần mềm kiểu lặp và tăng trưởng thì phù hợp với những dự án có đặc điểm gì?**

**65. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu xoắn ốc? 66. Mô hình vòng đời phần mềm kiểu xoắn ốc thì phù hợp với những dự án có đặc điểm gì? 67. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu tiến trình linh hoạt? 68. Mô hình vòng đời phần mềm kiểu tiến trình linh hoạt thì phù hợp với những dự án có đặc điểm gì?**

**69. Trong mô hình tiến trình liên hoạt, luôn có đại diện của khác hàng trong nhóm phát triển thì có ưu điểm gì?**

**70. Nêu ưu điểm, nhược điểm của mô hình nhóm code bình đẳng?**

**71. Mô hình nhóm code bình đẳng thì phù hợp với những dự án có đặc điểm gì? 72. Nêu ưu điểm, nhược điểm của mô hình nhóm code có chef?**

**73. Mô hình nhóm code có chef thì phù hợp với những dự án có đặc điểm gì? 74. Nêu ưu điểm, nhược điểm của kĩ thuật pair programming?**

**75. Kĩ thuật pair programming thì phù hợp với những dự án có đặc điểm gì? 76. Nêu ưu điểm, nhược điểm của kĩ thuật time boxing?**

**77. Nêu ưu điểm, nhược điểm của kĩ thuật stand up meeting?**

**78. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng LOC? 79. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng FFP? 80. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng Function Point? 81. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng COCOMO? 82. Tại sao trong mô hình tiến trình linh hoạt, không cần có pha đặc tả?**

**83. Tại sao trong nhóm walkthrough và inspection, luôn phải có đại diện của workflow tiếp theo?**

**84. Nếu nhóm SQA phát hiện ra ít lỗi, thì có thể kết luận nhóm code giỏi hay nhóm SQA kém? Tại sao?**

**85. Tại sao nói inspection và walkthrough là hướng tài liệu, mà không phải hướng vào người tham gia?**

**86. Quality assurance thì khác gì với testing?**

**87. Tại sao nói function point chịu ảnh hưởng chủ quan của các chuyên gia? 88. COCOMO tính đến nhiều tiêu chí hơn hay là function point? Giải thích? 89. SW development multiplier của COCOMO thì khác gì TCF của function point?**

**4**

**90. TCF của function point thì khác gì hằng số b của FFP? 91. Tại sao nguyên lí Djistra lại đúng?**

**92. Tại sao luật Brook lại đúng?**

**93. Người ta áp dụng luật Miller trong CNPM như thế nào? 94. Phát triển phần mềm thì khác gì sản xuất phần mềm? 95. Test trường hợp sai kiểu dữ liệu đầu vào thì thuộc thể loại test gì?**

**5**