HOC VIÊN CÔNG NGHÊ BƯU CHINH VIÊN THÔNG

**KHOA:** CÔNG NGHỆ THÔNG TIN 1

**NGÂN HÀNG CÂU HỎI THI THỰC HÀNH - VẤN ĐÁP**

**Phần lí thuyết**

**Câu hỏi lí thuyết Kỹ nghệ phần mềm**

1. Thế nào là corrective maintenance?

2. Thế nào là adaptive maintenance?

3. Thế nào là perfective maintenance?

4. Thế nào là refactoring?

5. Thế nào là "from scratch"?

6. Thế nào là moving target problem?

7. Thế nào là regession fault?

8. Thế nào là một episode?

9. Thế nào là một iteration?

10. Thế nào là một increasement?

11. Thế nào là một artifact?

12. Thế nào là portability?

13. Thế nào là reusebility?

14. Thế nào là milestone?

15. Thế nào là một story?

16. Thế nào là refactoring?

17. Thế nào là concept exploration?

18. Thế nào là business model?

19. Thế nào là traceability?

20. Thế nào là egoless programming?

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?

3

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