

1. Testowanie metodą białej skrzynki to:

Podejście, w którym testy opracowuje się na podstawie znajomości struktury programu/komponentu

2. Na proces oparty na elementach (komponentach) wielokrotnego użycia składają się:

Specyfikacja wymagań, analiza komponentów, modyfikacja wymagań, projekt, budowa i integracja, zatwierdzenie

3. Inspekcje oprogramowania są przykładem:

Statycznej analizy wybranych reprezentacji systemu (np. kodu źródłowego)

4. Przykładem systemu opartego na kontroli wersji jest:

Oprogramowanie Wiki

5. Generyczny model architektury kompilatora jest przykładem

Dziedzinowego modelu odniesienia

6. Elementami procesu projektowania obiektowego (OOD) są: (wybierz poprawne odpowiedzi)

Identyfikacja głównych obiektów systemu

Wyspecyfikowanie interfejsów obiektów

7. W rozproszonym modelu kontroli wersji

Każda stacja robocza przechowuje własną kopię repozytorium

8. Dwa rodzaje modeli architektonicznych charakterystycznych dla dziedziny nazwane są:

Wzorcami architektonicznymi oraz wzorcami dziedzinowymi

9. Testem jednostkowym może być objęta (wybierz poprawne odpowiedzi):

Klasa obiektów

Moduł

Procedura

10. Wzorce projektowe to:

Starannie sprawdzone rozwiązania uniwersalnych problemów

11. Kroki zarządzania zmianą wymagania (wybierz poprawne odpowiedzi):

Analiza problemu i specyfikacja zmiany

Analiza zmiany i ocena kosztów

Implementacja zmiany

12. Testy jednostkowe (ang. unit tests) (wybierz poprawne odpowiedzi):

Są testami defektów

Komponenty systemu testowane są w izolacji

13. Moduł:

Jest częścią składową podsystemu

14. Przykładem dziedzinowego modelu odniesienia jest :

generyczny model architektury kompilatora

15. Wadą modelu sterowania z rozgłaszaniem jest (wybierz poprawne odpowiedzi):

możliwe konflikty w obsłudze zdarzeń

podsystemy nie wiedza czy i kiedy zdarzenie zostało obsłużone

16. Wydanie oprogramowania (wybierz poprawne odpowiedzi):

może być publiczne i prywatne (w zakresie zespołu projektowego)

może być pierwsze i aktualizacyjne

17. Jednym z najbardziej znanych zrębów do projektowania graficznego interfejsu użytkownika jak i webowego nazywa się:

MVC

18. Proces zarządzania ryzykiem składa się z następujących czynności:

Identyfikacja, analiza, planowanie, monitorowanie

19. Celem testowania defektów / usterek oprogramowania jest:

ujawnienie utajonych defektów systemu

20. Dwa podstawowe modele architektury przepływu sterowania między systemami to:

sterowanie wywołanie-powrót oraz sterowanie za pomocą przerwań.

21. Testy wydania:

Mają za zadanie wykazać , że system spełnia swoją specyfikację

Testowany jest kompletny system

Są testami zatwierdzającymi

22. Testy integracyjne:

tworzenie systemu metod niezależnych ścieżek

mogą być wykorzystywane w strategii wstępującej lub zstępującej

23. Niezależna praca programistów nad komponentami systemu wspierana jest przez system zarządzania wersjami poprzez (zaznacz poprawne odpowiedzi):

udostępnienie mechanizmów pozwalających na tworzenie wersji komponentu poprzez połączenie (ang. merge) wersji komponentów znajdujących się w odrębnych sekwencjach wersji

możliwość tworzenia odrębnych sekwencji wersji komponentów (gałęzi)

wprowadzenie publicznego repozytorium oraz prywatnych przestrzeni roboczych

umożliwienie budowania systemu niezależnie od przestrzeni roboczej programisty

24. Inspekcje oprogramowania mogą dotyczyć: (wybierz poprawne odpowiedzi)

projektu systemu

specyfikacji wymagań

kodu źródłowego

25. Ile minimalnie klas równoważności należy zdefiniować dla parametru wejściowego, który jest listą liczb całkowitych. Poprawna liczba danych w liście mieści się w przedziale: "gdzie tu przedział???"

3 (lista pusta, lista zawierająca jedną liczbę, lista zawierająca więcej niż jedną liczbę)

26. Komunikacja pomiędzy komponentami systemu rozproszonego może się odbywać z wykorzystaniem modelu: (wybierz poprawne odpowiedzi)

interakcji bazującej na komunikatach

zdalnego wywołania procedury

27. W klasycznym procesie zarządzania zmianą "zgłoszenie zmiany":

tworzone jest przez rolę "klient"

musi zostać przeanalizowane m.in. pod kątem kosztów wprowadzania zmiany

28. Merge/pull request:

daje możliwość dokonania przeglądu kodu przez zespół

wymagany jest do scalenia (merge) zmian pomiędzy dowolnymi gałęziami

scalenie zmian może wymagać akceptacji zespołu

29. Zaangażowanie klienta w proces zmian w systemie można realizować za pomocą:

mechanizmów umożliwiających zgłaszanie oraz śledzenie propozycji zmian

30. Testy obciążenia to testy:

usterek

31. Alternatywą w przypadku braku możliwości efektywnego zaangażowania klienta w iteracyjny proces wytwarzania oprogramowania, jest:

wykorzystanie systemu zarządzania zmianą (zgłoszeniem zmiany)

32. Dostarczenie przyrostowe może być zamodelowane w systemie kontroli wersji za pomocą modelu:

Feature branch

33. System rozproszony określany jako skalowalny powinien umożliwiać: (wybierz poprawne odpowiedzi)

geograficzne rozpraszanie komponentów systemu bez zmniejszania wydajności

zarządzanie systemem, zwiększającym swój rozmiar, nawet jeśli jego części znajdują się fizycznie w niezależnych organizacjach

zwiększanie zasobów systemu w odpowiedzi na zwiększającą się liczbę jego użytkowników

34. Celem refaktoryzacji kodu źródłowego jest:

zmiana struktury kodu, bez jego funkcjonalności

35. Model architektury 4+1 uwzględnia następujące perspektywy:

logiczna

procesów

projektowa

36. Jeżeli w systemie Git zatwierdzamy zmiany w plikach, które znajdują się już pod kontrolą systemu to można wykorzystać następujące polecenia:

git add; git commit

git commit -a

37. Według modelu wersjonowania semantycznego, zakładając że wersje systemu mają postać sekwencji X,Y,Z wprowadzenie niekompatybilnych zmian wymaga zwiększenia wartości w sekwencji:

X

38. Wydanie (ang. Release) oprogramowania to:

Wersja oprogramowania, która została dostarczona klientom

39. Model zarządzania repozytorium kodu źródłowego o nazwie „Feature branch” może być efektywnie wykorzystany przy:

Realizacji przyrostowej

Dostarczaniu przyrostowym

40. W modelu „Git flow” pojawienie się nowej wersji w gałęzi master oznacza:

Wydanie produkcyjne

41. Pojęcie „Nieawodność systemu” oznacza:

Prawdopodobieństwo bezawaryjnego działania w ciągu ustalonego czasu w zadanym środowisku w określonym celu

42. Jeżeli w systemie przetwarzania danych możliwe jest rozbiecie procesu na serię niezależnych kroków to organizacji systemu można zastosować wzorec

Potoku

43. Do metod zatwierdzania wymagań można zaliczyć:

Przeglądy wymagań

44. W procesie zarządzania konfiguracją budowanie systemu odbywa się:

Lokalnie w środowisku prywatnym programisty

Z pomocą usługi budowania powiązanej z systemem kontroli wersji

W ramach procesu zatwierdzania zgłoszenia zmiany (change request)

45. Założmy, że budujemy system, w którym wymagane jest istnienie wielu reprezentacji tych samych danych oraz automatyczna ich aktualizacja w odpowiedzi na modyfikację danych. Jaki wzorec architektoniczny można zastosować w tym kontekście?

Model-Widok-Kontroler

46. Test Driven Development (TDD) to:

Iteracyjne podejście do procesu implementacji systemu

47. Założmy, że testujemy system transakcyjny, zaprojektowany do przetwarzania maximum 300 transakcji na sekundę (tx/s) wykonanie testów obciążenia (stres testing) wymaga:

Wykonania serii testów o wzrastającym obciążeniu do poziomu przekraczającego 300 lub powodującego błąd systemu

48. Do strategii testowania strukturalnego należą:

Testowanie ścieżek

Testowanie, którego celem jest uruchomienie wszystkich instrukcji programu lub komponentu

49. Podsystem: (wybierz poprawne odpowiedzi)

Jest systemem na swoich własnych prawach

Posiada dobrze zdefiniowany interfejs

Składa się z modułów

50. W procesie testowania systemu, termin „testy akceptacyjne” oznacza:

Testowanie komponentu z wykorzystaniem rzeczywistych danych w celu zaakceptowania lub odrzucenia danego komponentu

51. Przypadek użycia (element modelu przypadków użycia) opisuje:

Otoczenie systemu

52. W procesie testowania systemu, termin „testy jednostkowe” oznacza:

Indywidualne testowanie komponentów

53. Narzędzia RAD to:

Narzędzia umożliwiające szybką budowę prototypów lub gotowych aplikacji

54. Najważniejszą bazą informacji dla potrzeb planowania projektu jest:

Zbiór wymagań użytkowników

55. Na proces oparty na elementach wielokrotnego użycia składają się:

Specyfikacja wymagań, analiza komponentów, modyfikacja wymagań, projekt, budowa i integracja, zatwierdzania

56. Przykładami narzędzi należących do grupy lower-CASE są (wybierz poprawne odpowiedzi):

Repozytorium wymagań

Edytor UML

57. Do procesu projektowania oprogramowania NIE należy:

Abstrakcyjna specyfikacja

Projektowanie architektury

58.

Testowanie obciążenia zakłada:

Testowanie systemu powyżej poziomu maksymalnego obciążenia aż do momentu awarii

59. Do zalet architektury współdzielonego repozytorium danych można zaliczyć (wybierz poprawne odpowiedzi):

Jednolity system danych

Ujednolicenie mechanizmów zarządzania bezpieczeństwem danych

60. Metoda ścieżki krytycznej to:

Graficzna reprezentacja aktywności projektu w postaci sieci aktywności

61. Aktywność procesu inżynierii wymagań, na podstawie rezultatów której, można podjąć decyzję o kontynuacji prac nad systemem, nazywa się:

Studium wykonalności

62. Podsystem:

Jest wyposażony w interfejs

Jest częścią składową modułów systemu

63. Klasy równoważności:

Klasa powiązanych danych wejściowych, w ramach której program zachowuje się w porównywalny sposób dla każdego jej elementu

64. Testowanie metodą czarnej skrzynki to:

Podejście, w którym testy wprowadza się ze specyfikacji programu/komponentu

65. Przykładami narzędzi należących do grupy Upper-case są (wybierz poprawne odpowiedzi):

Debugger

Kompilator

66. Testowanie wydajności zakłada:

Testowanie systemu obciążonego maksymalnie

67. Do zalet modelu architektury zdarzeniowej nazywanego modelem rozgłoszenia(ang. Broadcast) można zaliczyć (wybierz poprawne odpowiedzi):

Umożliwia implementacje szybkich odpowiedzi systemu na zdarzenia

Prostota ewolucji systemu

Możliwość integrowania podsystemów pracujących na różnych platformach i komunikujących się za pomocą sieci komputerowej

68. Podstawowe modele architektury repozytorium danych to:

model repozytorium i model zdecentralizowany

69. Skrót CASE oznacza:

Computer-Aided Software Engineering

70. Narzędzia CASE dzielą się na:

upper-case i lower-case

71. Elementami trójkąta kompromisu (ang. Tradeoff triangle) są:

zasoby, harmonogram, cechy systemu

72. Dwa podstawowe modele architektury związane z kontrolą przepływu sterowania pomiędzy podsystemami to:

model kontroli scentralizowanej oraz model kontroli sterowanej zdarzeniami

73. Model sterowania, w którym sterowanie zaczyna się na wierzchołku hierarchii i przez wywołanie podprogramów przechodzi do najniższych poziomów drzewa wywołań nazywa się:
modelem wywołanie- powrót

74. Metoda szacowania kosztów zwana Metoda Punktów Funkcyjnych należy do kategorii:
metod parametrycznych

75. Wzorzec projektowy to:
mechanizm umożliwiający ponowne użycie wiedzy nt. problemu i sposobu jego rozwiązania

76. Model sterowania, w którym zdarzenie jest przesyłane do wszystkich podsystemów nazywa się:
modelem rozgłaszania

77. Modelowanie pojęciowe oznacza:
Procesy myślowe towarzyszące pracy nad oprogramowaniem

78. Określenie zakresu przedsięwzięcia w fazie strategicznej oznacza:
określenie systemów, organizacji i użytkowników, z którymi tworzony system ma współpracować

79. Decyzje, które powinny być podjęte w fazie strategicznej to m.in.:
wybór modelu, narzędzi CASE, stopnia wykorzystania gotowych komponentów

80. Faza strategiczna jest nazywana również:
studium osiągalności

81. Co to jest proces inżynierii oprogramowania:
zestaw działań, których celem jest wytworzenie lub ewolucja oprogramowania

82. W procesie testowania systemu, termin „testy systemowe” oznacza:

Testowanie systemu ze szczególnym uwzględnieniem nowych komponentów

83. Przy testach jednostkowych zewnętrzne zależności

Można symulować (stubs)

Można pozorować (mocks)

84. Wiele kopii repozytorium kontroli wersji może istnieć w systemach kontroli wersji typu:

Rozproszonego

85. Do zalet modelu architektury o nazwie repozytorium należą:

Łatwa integracja nowych podsystemów

Możliwość efektywnej implementacji przechowywania dużej ilości danych

86. Operacja idempotentna to operacja:

Której wielokrotne wykonanie daje taki sam rezultat

87. Do zalet modelu warstwowego architektury systemu można zaliczyć:

Prostota podmiany warstwy (przy zachowaniu jej interfejsu)

Umożliwienie ukrycia zależności od konkretnej platformy

Ułatwienie przyrostowego procesu tworzenia oprogramowania

88. W modelu interakcji zdalnej bazującym na przesyłaniu komunikatów:

Wywołanie zdalnego komponentu imituje lokalne wywołanie procedury

Warstwa pośrednia przekazuje wywołania do zdalnego komponentu

89. Do czynności procesu ciągłej integracji (continuous integration) należą:

Implementacja oprogramowania

Automatyczne uruchamianie testów

90. Do pożądaných cech zaawansowanego systemu automatyzującego proces budowy systemu należą:

Raportowanie rezultatów

Automatyzacja testów

Minimalizacja rekompilacji

Integracja z systemem zarządzania wersjami