1 En test med det formål at overbelaste ens system og måle hvornår det holder op med at virke kaldes for en?

* Unittest
* Performancetest
* Integrationstest
* Stresstest
* Betatest
* Systemtest
* Accepttest

2 Når man lader en del af sine brugere afprøve en ny version af ens system samtidig med at man lader resten af brugerne bruge den gamle version og så sammenligner de to gruppers resultater hedder det en:

* Alpha/Beta-test
* Integrations-test
* A/B-test
* Unit-test
* Accept-test
* System-test23

3 Du har et lokalt repo, og en kollega har pushet ændringer op i jeres remote repo. Hvordan får du din kollegas ændringer ned i dit lokale repo?

* git checkout
* git commit
* git import
* git export
* git pull

4 Hvis du skal teste metoden alice() der bruger metoden bob(), men du vil ikke have at testen påvirkes af eventuelle fejl i metoden bob(), så er du nødt til at:

* Skrive en test til den anden metode først.
* Skrive en test-stub.
* Omskrive den anden metode.
* Skrive en test-driver

5 Er git et centraliseret eller et decentraliseret versionsstyringsværktøj?

* Decentraliseret
* Centraliseret

Du har følgende kode:

5 int alice = 10;

46 alice = alice + bob();

47 System.out.println("Alice er nu: " + alice);

Du vil gerne pause udførslen i linje 47, men kun hvis alice nu har værdien 85. Hvad gør du?

* Single stepper til linje 47 og ser om variablen alice indeholder værdien 85.
* Sætter et breakpoint i linje 47 med en condition om kun at stoppe hvis alice indeholder værdien 85.
* Sætter et breakpoint
* Sætter et debugpoint i linje 47 med en condition om kun at stoppe hvis alice indeholder værdien 85.
* Sætter et debugpoint
* Det kan man ikke

Hvem godkender en accepttest?

* Udvikleren
* Projektlederen
* Kunden
* Testeren

En unit-test er en?

* Black box test
* White box test
* Breakpoint test
* Debugging test

Du har følgende kode:

45    int alice = 10;

46    alice = alice + bob();

47    System.out.println("Alice er nu: " + alice);

Når debuggeren har pauset udførslen på linje 46 og man vil udføre linjen og pause i første linje i metoden bob(), så hedder det man gør?

* sætte et breakpoint
* sætte et debugpoint
* step out of
* det kan man ikke
* single-steppe
* step into

Når man vil fortælle debuggeren at den skal pause koden på en bestemt linje, så?

* Sætter man et breakpoint på den linje.
* Sætter et debugpoint på den linje.
* step into
* Tænder man debuggeren.
* step out of
* det kan man ikke

Hvordan får man git til at tracke en ny fil?

* git track
* git addfile
* git begin
* git add

Efter at have oprettet et nyt git repo og oprettet en fil ved navn helloWorld.java, hvilken af følgende kommandoer vil så give en fejl?

* git add
* git status
* git add helloWorld.java
* git commit -m "Foerste commit"

Hvad er filnavnet på den fil man kan bruge til at bede git om at ignorere bestemte filer?

* gitignore.txt
* git.ignore
* .gitignore
* Det kan man ikke.
* ignore.git

Hvornår kan man udføre en unittest?

* Før Analysefasen.
* Efter udviklingsfasen.
* Før Designfasen.
* Før Udviklingsfasen.

Hvordan får man git til at oprette en ny branch med navnet feature43?

* git feature43 branch
* Ingen af de andre muligheder
* Hvad er en branch?
* git commit 'feature43'
* Det kan man ikke
* git branch feature43

Hvornår kan man udføre en accepttest?

* Før Udviklingsfasen.
* efter udviklingsfasen.
* Før Analysefasen.
* Før Designfasen.

Hvornår kan man designe en unittest?

* Før Analysefasen.
* Før Designfasen.
* Efter udviklingsfasen.
* Før Udviklingsfasen.

Hvornår kan man udføre en brugertest?

* Når kunden er kommet med opgaven.
* Når man har lavet designet.
* Når koden er færdig og testet.
* Når koden er færdig, men før man tester.
* Når man har lavet analysen.
* Alle andre svar er rigtige.

Når man laver en alpha-test er de typiske testere:

* Ens chef
* Testerne
* Ens kollegaer
* En gruppe slutbrugere
* Udviklerne
* Kunden

Hvornår kan man udføre en systemtest?

* Før Udviklingsfasen.
* Før Designfasen.
* Efter udviklingsfasen.
* Før Analysefasen

Formålet med at debugge en metode er?

* At finde ud af hvor i koden en fejl opstår.
* At finde ud af om der er fejl i koden.
* At kunne ændre variables værdier i koden.
* At teste koden.

Hvilken af følgende skriver man normalt en unittest, for at teste:

* en metode
* en komponent
* en server
* et system

Hvis du har følgende metode:

public int skatAtBetale(String indkomst) {

        int skatteprocent = 38;

        int fradrag = 48000;

        int skatAtBetale = 0;

        if (Integer.parseInt(indkomst) > fradrag) {

            skatAtBetale = ((Integer.parseInt(indkomst) - fradrag) \*skatteprocent) / 100;

        }

        return skatAtBetale;

    }

Hvilken af følgende ville være et fornuftigt valg af grænseværdier til ækvivalensklasser til en positiv test:

* String.valueOf(Integer.MIN\_VALUE), "0", "48000" og

String.valueOf((Integer.MAX\_VALUE-48000)/38)

* "0", "38", "48000" og String.valueOf((Integer.MAX\_VALUE-48000)/38)
* Integer.MIN\_VALUE, 0, 48000 og (Integer.MAX\_VALUE-48000)/38
* String.valueOf(Integer.MIN\_VALUE), "0", "48000" og

String.valueOf((Integer.MAX\_VALUE-48000)/38)

* String.valueOf(Integer.MIN\_VALUE), "0", "48000",

String.valueOf((Integer.MAX\_VALUE-48000)/38) og

String.valueOf(Integer.MAX\_VALUE)

* "0", "38", "48000" og "abc"
* Integer.MIN\_VALUE, 0, 48000, (Integer.MAX\_VALUE-48000)/38 og

Integer.MAX\_VALUE

Hvis vi med code coverage har opnået en statement coverage på 80% betyder det at? Tip: a < b betyder a mindre end b.

* branch coverage kan man ikke sige noget om og path coverage >= 80%
* branch coverage >= 80% og path coverage kan man ikke sige noget om
* branch coverage >= 80% og path coverage >= 80%
* Man kan ikke sige noget om hverken branch coverage eller path coverage
* branch coverage < 80% og path coverage < 80%

Hvordan opretter man en ny branch med navnet feature43 og skifter til den?

* git checkout --create\_branch feature43
* git checkout -b feature43
* git branch --checkout feature43
* git branch -switch feature43

En accept-test er en?

* Black box test
* Debugging test
* Breakpoint test
* White box test

Hvis du skal implementere en ny feature, men du vil ikke påvirke de andre udvikleres arbejde, så er det en god idé at arbejde i?

* et bestemt tag
* Ingen af de andre valgmuligheder
* main branchen
* master branchen

Når man laver en beta-test er de typiske testere:

* Kunden
* Udviklerne
* Ens kollegaer
* Ens chef
* En gruppe slutbrugere
* Testerne

Hvad kan man se ud af denne traceability matrix (TC=Text case og RS=Krav):

Et billede, der indeholder tekst, skærmbillede, nummer/tal, Font/skrifttype

Automatisk genereret beskrivelse

* Krav nummer 2 bliver ikke testet af en eneste test case.
* Test case nummer 2 er unødvendig og kan slettes.
* Alle svarmulighederne er rigtige.
* Test case nummer 4 er unødvendig, da kravene også bliver testet af andre testcases.

Hvad betyder det hvis man bruger et bibliotek der anvender semantisk versionering og versionsnummeret skifter fra 5.3.1 til 5.4.0?

* Det er en mindre tilføjelse eller ændring, som er bagudkompatibel med den tidligere version
* Man kan ikke regne med at den nye version er kompatibel med brugen af
* den tidligere version.
* Det er en bugfix uden store ændringer.
* Ingen af de andre svar.

git clone kommandoen gør følgende:

* Merger en branch
* Opretter en lokal kopi af repoet, men opretter ikke et working directory
* Opretter en lokal kopi af repoet og opretter et working directory
* Opretter ikke en lokal kopi af repoet, men opretter et working directory

Man kan godt lave en integrationstest uden at have lavet unit tests?

* Sandt
* Spørgsmålet kan ikke besvares.
* Falskt

Du har følgende kode:

45 int alice = 10;

46 alice = alice + bob();

47 System.out.println("Alice er nu: " + alice);

Når debuggeren har pauset udførslen på linje 46 og man vil udføre linjen og resten af denne metode, returnere og stoppe på første linje efter denne metode blev kaldt, så hedder det man gør?

* single-steppe
* sætte et breakpoint
* step into
* sætte et debugpoint
* step out of
* det kan man ikke

Hvis vi med code coverage har opnået en path coverage på 80% betyder det at?

Tip: a < b betyder a mindre end b.

* Statement coverage >= 80% og branch coverage >= 80%
* Statement coverage kan man ikke sige noget om og branch coverage >=80%
* Statement coverage < 80% og branch coverage < 80%
* Statement coverage >= 80% og branch coverage kan man ikke sige noget om
* Man kan ikke sige noget om hverken statement coverage eller branch coverage

Hvis vi med code coverage har opnået en branch coverage på 80% betyder det at?

Tip: a < b betyder a mindre end b.

* statement coverage >= 80% og path coverage >= 80%
* statement coverage kan man ikke sige noget om og path coverage >= 80%
* Man kan ikke sige noget om hverken statement coverage eller path coverage
* statement coverage >= 80% og path coverage kan man ikke sige noget om
* statement coverage < 80% og path coverage < 80%

En betatest kommer ofte før en alphatest?

* Sandt
* Falskt

Når debuggeren har pauset udførslen på en bestemt linje, så kan man?

* Ændre værdierne i de variable der er deklareret, men ikke kigge i dem.
* Kigge i de variable der er deklareret og ændre deres værdier.
* Kigge i de variable der er deklareret, men ikke ændre deres værdier.
* Man kan hverken kigge i variable der er deklareret eller ændre deres værdier

Hvordan omdøber man den nuværende branch fra "bugfix22" til "feature43"?

* git branch -m feature43
* git reflog feature43
* git branch -D feature43
* git remote rn feature43

Hvornår kan man designe en integrationstest?

* Før Designfasen.
* Efter udviklingsfasen.
* Før Udviklingsfasen.
* Før Analysefasen.

En unittest består af tre dele.

* en udførdel, en kontrollérdel og en debugdel.
* en setupdel, en udførdel og en kontrollérdel.
* en startdel, en mellemdel og en slutdel.
* en analysedel, en designdel og en implementérdel.

Du har følgende kode:

45 int alice = 10;

46 alice = alice + bob();

47 System.out.println("Alice er nu: " + alice);

Når debuggeren har pauset udførslen på linje 46 og man vil udføre linjen og pause i næste linje, så hedder det man gør?

* step out of
* single-steppe
* det kan man ikke
* sætte et debugpoint
* step into
* sætte et breakpoint

I en formel testcase skal der blandt andet være:

* Test case ID, Preconditions, Postconditions, testkoden, forventet resultat og dato/tidspunkt for udførsel.
* Test case ID, Preconditions, Postconditions, test procedure, forventet resultat og dato/tidspunkt for udførsel.
* Test case ID, Preconditions, Postconditions, ækvivalensklasser, forventet resultat og dato/tidspunkt for udførsel.
* Test case ID, Preconditions, Postconditions, blodtype, forventet resultat og dato/tidspunkt for udførsel.

Når man laver en unit-test er de typiske testere:

* Ens kollegaer
* Ens chef
* Kunden
* Udviklerne
* En gruppe slutbrugere
* Testerne

Hvis man bruger TDD (Test driven development) så:

* Skriver man en masse kode, hvorefter man skriver store tests der tester koden.
* Skriver man få linier kode, hvorefter man skriver små tests der tester koden.
* Skriver man store tests, hvorefter man skriver en masse kode, der får testen til at køre uden fejl.
* Skriver man små tests, hvorefter man skriver få linier kode, der får testen til at køre uden fejl.

Hvornår kan man designe en accepttest?

* Efter udviklingsfasen.
* Før Designfasen
* Før Analysefasen.
* Før Udviklingsfasen.

Hvis du vil oprette en ny fil og så tilføje den til et remote repo, så er den korrekte rækkefølge:

* opret fil --> git add --> git commit --> git push
* opret fil --> git pull --> git push --> git add
* opret fil --> git push --> git commit --> git fetch
* opret fil --> git commit --> git add --> git push

Hvornår kan man udføre en integrationstest?

* Efter udviklingsfasen.
* Før Udviklingsfasen.
* Før Analysefasen.
* Før Designfasen.

Hvis man i et eksisterende udviklingsprojekt der bruger git, pludselig mister adgang til github, så

* Har hver udvikler kun de sidste nye ændringer lokalt
* Har hver udvikler ingenting af koden
* Har hver udvikler hele ændringshistorikken lokalt

Hvilken git kommando bruges til at oprette et lokalt tomt git repository i den nuværende mappe?

* git log
* git init --empty
* git reset
* git init

I konfigurationsstyring styrer man blandt andet versioner af projektet, aftaler, standarder og projektets struktur.

* Sandt
* Falskt

Debug er:

Vælg én svarmulighed

en lokalisering af hvilke dele af koden, der skal afprøves

en automatiseret testtype

en metode til hurtig test

en lokalisering af fejl i koden

Fordelen ved exploratory testing er:

Vælg én svarmulighed

den er hurtigere og lettere at gennemføre istedet for en planlagt test

hvis den anvendes som supplement til en planlagt test, kan der fanges uforudsete fejl

hvis den anvendes som supplement til en planlagt test, kan antallet af planlagte testcases reduceres

ingen af øvrige svar

Antag du har et bibliotek, der anvender semantisk versionering med versionsnummer 6.1.0. Hvis der bliver foretaget et bugfix, hvad kommer versionsnummeret til at blive?

Vælg én svarmulighed

6.1.1

7.0.0

6.2.0

6.1.0.1

Hvad menes der med begrebet coverage?

Vælg én svarmulighed

omfanget af kildekoden, der opfylder de funktionelle krav

omfanget af kildekoden, der afprøves med en specifik afprøvning

omfanget af kildekoden, der afprøver de ikke-funktionelle krav

omfanget af dokumentatationen, der beskriver afprøvningen

Formålet med en usability test er at:

Vælg én svarmulighed

verificere at brugeren kan anvende programmet

verificere at programmet kan udføre brugerens opgave

validere at programmets funktion stemmer overens med brugerens behov

validere at brugerens kompetancer i forbindelse med anvendelse af programmet

Under udvikling af et program, kan der foretages afprøvning:

Vælg én svarmulighed

alle de tre øvrige

før programmeringen

under programmeringen

efter programmeringen

En automatiseret test har blandt andet følgende fordel:

Vælg én svarmulighed

den finder flere fejl end manuelle test

god til load/performance test

den forbedrer udviklingsprocessen

den er altid hurtigere end manuel test

I en integrationstest bruges en stub til:

Vælg én svarmulighed

at permanent erstatte en kodedel

at simulere en kodedel indtil denne færdigprogrammeres

at teste en kodedel

at klargøre kodedelen til kunden

En unit test kan benyttes til:

Vælg én svarmulighed

at teste ikke-funktionelle krav

at teste en metodes mulige gennemløb

at teste en metodes output givet specifikke parametre op imod forventede resultater

at teste sammenhængen mellem flere metoder, således at kommunikationen mellem disse kan evalueres

GIT anvendes primært til?

Vælg én svarmulighed

versionsstyring

syntaksvalidering

fejlfinding

afprøvning

Hvornår anvendes versionsstyring?

Vælg én svarmulighed

Under programmering

Før programmering

Efter programmering

I alle situationer

n test case har til formål:

Vælg én svarmulighed

at teste brugerens brug af systemet

at definere ikke funktionelle krav

at teste tilfældige preconditions op imod forventede postconditions

at definere afprøvningsdata samt de tilhørende forventede uddata

En ikke-funktionel test kan benyttes til:

Vælg én svarmulighed

at teste en software enhed

at evaluere slutbrugerens forventninger til produktet

at teste en metodes input-output

at teste sammenhængen mellem flere metoder, således at de returnerer konsistente uddata

Hvis man før et program udføres i en debugger sætter et breakpoint på en kodelinje, så:

Vælg én svarmulighed

debuggeren pauser koden på den angivne linje

afventer debuggeren en ekstra kommando, før udførslen starter

debuggeren afslutter programmet på den angivne linje

ingen af de øvrige svar

Den største forskel mellem vandfaldsmodellen og iterativ udvikling er:

Vælg én svarmulighed

vandfaldsmodellen bliver hurtigere gennemført, fordi der udføres færre steps

den iterative model kan rette op på fejl introduceret tidligere i udviklingsforløbet

i den iterative model er afprøvning ikke nødvendig

der er ingen forskel på de to metoder

De primære nøglord i Gherkin er:

Vælg én svarmulighed

Given, When, How

Given, When, Then

Given, When, Why

Given, Why, Cucumber

En white-box test bruges til:

Vælg én svarmulighed

test af en metodes afviklingshastighed

test af at alle mellemresultater i koden er korrekte

test af en metodes forgreninger, løkker og undtagelser

test af javas bytecode

Fordelen ved agil udvikling er:

Vælg én svarmulighed

hurtig omstilling efter brugerønsker

minimal dokumentation

minimal kode

hurtig afprøvning

når man ønsker at afprøve funktionaliteten af et program, så anvender man en:

Vælg én svarmulighed

blackbox test?

whitebox test?

colour test?

metode test?

Hvad bruges konfigurationsstyring til?

Vælg én svarmulighed

at holde styr på kildekoden

at styre aftaler, procedurer, standarder, kontrakter m.m. i forbindelse med projektudførsel

at teste preconditions i en test case

at styre aftaler, procedurer, standarder, kontrakter m.m. i forbindelse med idrifttagning