

# ISTQB™ Foundation Level Prøveeksamen

Svar på følgende spørsmål

For hvert spørsmål er der ETT og BARE ETT rett svar! (Unntak er avmerket spesielt).

## Spørsmål til Kap 1 ("Fundamentals")

**1.1. (K2) Hva er betydningen av prinsippet om at feil opptrer samlet?**

- (a) Flere feil kan rettes ved å rette på en plass.
- (b) Et lite antall moduler inneholder de fleste feil som finnes under test.
- (c) Et avvik (failure) er ofte forårsaket av flere feil i fellesskap.
- (d) Feil finnes statistisk sett oftere i senere testfaser.

**1.2 (K2) Hvilket av de følgende utsagn er FEIL?**

- (a) Å finne få eller ingen feil garanterer bra kvalitet.
- (b) Å finne og rette feil hjelper ikke hvis hele systemet er ubrukelig og ikke oppfylder brukernes behov og forventninger.
- (c) Testing kan ikke bevise at det ikke finnes feil.
- (d) Hvis en gjentar de samme testene igjen og igjen, vil denne samlingen av testtilfelle før eller siden ikke finne nye feil.

**1.3 (K1) Hvilken av de følgende testene gir høyest verdi i et prosjekt?**

- (a) En test som produserer riktige resultater
- (b) En test som finner en feil eller har potensialet for å finne feil
- (c) En test som gir brukeren større tillit
- (d) En test som ikke forsinker leveransen

**1.4 (K2) En større samling testtilfelle finner ingen problemer. Hvilken av de følgende slutninger kan du trekke?**

- (a) Produktet inneholder ingen eller meget få feil.
- (b) Produktet er ikke tilstrekkelig grundig testet.
- (c) Det behøves mer informasjon for å trekke en slutning.
- (d) Testdekningen var altfor lav.

**1.5 (K1) Dette spørsmål er på engelsk grunnet terminologien. Det kan være flere riktige svar.**

**Something that is incorrect in software is called:**

- (a) An error
- (b) A fault
- (c) A failure
- (d) A defect
- (e) A problem

**1.6 (K2) Dette spørsmål er på engelsk grunnet terminologien.  
Hva er sant?**

- (a) A fault causes a bug which may cause a failure
- (b) A mistake causes an error which will cause failures during run time
- (c) An error causes a fault which may cause failures during run time
- (d) An error causes a problem which can cause defects during run time

**1.7 (K2) Hva er formålet med testsluttkriterier i testplanen?**

- 1) Å vite når en er ferdig med å utføre et spesifikt testtilfelle
  - 2) Å sikre at spesifikasjonen av testtilfelle er fullstendig
  - 3) Å sette kriterier som brukes for å generere testinput
  - 4) Å planlegge når en skal stoppe testingen
  - 5) Å ha mulighet til å kontrollere mot slike kriterier når en skal beslutte om en skal avslutte testingen
  - 6) Forbedret kommunikasjon med oppdragsgiveren om testingens omfang
- (a) 1, 4 og 6 er riktig
  - (b) 4, 5 og 6 er riktig
  - (c) 2, 3, 4 og 5 er riktig
  - (d) 1, 3 og 4 er riktig

# Spørsmål til Kap 2 ("Testing under softwarens livssyklus")

## 2.1 (K1) Hva er testnivåene i den generelle V-modellen?

- (a) Komponenttest, Integrasjonstest, Systemtest, Akseptansetest.
- (b) Enhetstest, Komponenttest, Integrasjonstest, Systemtest, Systemintegrasjonstest, Akseptansetest, Test under vedlikehold.
- (c) Komponenttest, Integrasjonstest, Systemtest, Ikke-funksjonell test, Akseptansetest.
- (d) Komponenttest, Systemtest, Beta test, Akseptansetest.

## 2.2 (K2) Her er noen utsagn om regresjonstesting. Hva er rett?

- 1) Regresjonstesting gjøres bare under system- og akseptansetest.
  - 2) Regresjonstesting gjøres på alle testnivåer og for funksjonell og ikke-funksjonell testing.
  - 3) Regresjonstesting er fornyet testing av et allerede testet program, etter endringer, for å finne feil som er introdusert eller aktivert som et resultat av endringen.
  - 4) Regresjonstesting omfatter en del av eller alle allerede utførte testtilfelle.
  - 5) Regresjonstest er viktigst ved tradisjonell systemutvikling.
  - 6) Regresjonstest er viktigst ved inkrementell eller trinnvis systemutvikling.
- 
- (a) 2 og 3 er riktig, resten er feil.
  - (b) 1,3,4,6 er riktig, resten feil
  - (c) 1 og 6 er feil, resten riktig
  - (d) 1 og 5 er feil, resten riktig

## 2.3 (K2) Hvilke av de følgende utsagn for testingen i en livssyklusmodell er rett?

- 1) I en iterativ utviklingsmodell blir funksjonelle tester utført for hvert inkrement, mens ikke-funksjonelle tester utføres i siste inkrement.
  - 2) I en iterativ utviklingsmodell blir funksjonelle tester og ytelsestester utført i hvert inkrement, andre ikke-funksjonelle tester blir utført i siste inkrement.
  - 3) For hvert konstruktive trinn finnes der et testtrinn.
  - 4) Analyse og design av testene for et gitt testnivå bør starte når tilhørende utviklingsaktivitet utføres.
  - 5) Testnivåene må tilpasses prosjektets natur og systemarkitekturen.
  - 6) Ved objektorientert systemutvikling er komponenttest spesielt viktig, mens integrasjonstesting kan en se bort fra.
- 
- (a) 1,3,6 er feil, resten riktig
  - (b) 1,2 og 6 er feil, resten riktig

(c) 2 og 6 er feil, resten riktig

(d) 1 og 2 er feil, resten riktig

**2.4 (K2) Hvilke(n) av de følgende oppgavene hører ikke inn under testplanlegging?**

- (a) Å bestemme sluttkriterier for testingen.
- (b) Å implementere organisasjonens testpolicy og/eller teststrategi.
- (c) Å sette i gang korrigerende tiltak.
- (d) Å identifisere testingens mål.
- (e) Å identifisere testdatasett.

**2.5 (K1) Hva er en negativ test?**

- (a) Et testtilfelle som ikke kjørte suksessfullt
- (b) En innføring av et system som feilet pga negativ kundefeedback
- (c) En test for robustheten til systemet
- (d) En test som sjekker brukernes forståelse av systemet i vanskelige situasjoner

**2.6 (K1) Hvilken av disse oppgavene er IKKE en del av systemtesting?**

- (a) Test av funksjonaliteten som sluttbrukerne ser den
- (b) Test av samarbeid mellom komponentene
- (c) Test av systemets ytelse
- (d) Test av systemets sikkerhet

# Spørsmål til Kap 3 Syllabus ("Statistiske teknikker")

## 3.1 (K2) Hvilke karakteristikk hører sammen når det gjelder gjennomganger?

- 1) Inspeksjon
  - 2) Uformell review
  - 3) Walkthrough
  - 4) Teknisk review
- 
- A) Krever bruk av en opplært moderator
  - B) Ledes vanligvis av forfatteren
  - C) Kan brukes til opplæring av folk
  - D) Kan brukes til å finne konsensus om løsninger
- 
- (a) 1 – B og C og D, 2 – C, 3 – B, 4 - A
  - (b) 1 – A, 3 – B, 3 – C, 4 – D
  - (c) 1 og 4 – A, 2 – B, 3 – B og C og D
  - (d) 1 – D, 2 – B, 3 – C, 4 - A

## 3.2 (K2) Hva er hovedoppgavene til en linje- eller prosjektleder i forbindelse med en gjennomgang?

- a) Å sørge for at reviews finner sted samt at deltakerne har tid til å forberede seg og delta
- b) Å kontrollere at deltakerne forbereder seg skikkelig
- c) Å sjekke at dokumentene følger gitte retningslinjer
- d) Å organisere at det finnes kvalifiserte moderatorer og sørge for møterom

## 3.3 (K2) Hva er hovedforskjellen mellom en formell teknisk review og en inspeksjon?

- a) Inspeksjon krever sjekklister samt start- og sluttkriterier, det er valgfritt ved tekniske reviews.
- b) Inspeksjoner brukes for dokumentasjon mens tekniske reviews brukes for kode.
- c) Tekniske reviews er mindre formelle og kan derfor kjøres med lengre møter.
- d) Tekniske reviews blir bare brukt innenfor utviklingsgruppen, Inspeksjoner kan inkludere en videre krets av interessenter, inklusive kunden.

I eksamen blir det bare tre spørsmål til dette kapittel. Besvar de neste to bare hvis du er ferdig med alt annet!

### **3.4 (K1) Hva er hovedmålet med statistisk analyse?**

- a) Tidlig oppdagelse av feil før testutførelsen.
- b) Tidlig oppdagelse av ikke-funksjonelle feil.
- c) Testing av koden i debuggingsmodus før integrasjon.
- d) Testing av en på forhånd definert tilstand ved testing av tilstandsoverganger.

### **3.5 (K2) Hvilke funksjoner har et verktøy som utfører statistisk analyse?**

- a) Det instrumenterer koden med spesiell kode som oppdager eventuelle hackerangrep.
- b) Det kontrollerer hvilke kodelinjer som er utført under en test og hvilke ikke for på den måten å oppdage "død" kode.
- c) Det kontrollerer om koden har rett versjon.
- d) Det kontrollerer om koden følger en kodestandard og finner sikkerhetshull.

## Spørsmål til Kap 4 ("Test design teknikker")

### 4.1 (K2) Hva er IKKE RIKTIG for ekvivalensklasseinndeling?

- a) Alle ugyldige data for en input er alltid del av samme ekvivalensklasse.
- b) Inputdata blir delt i grupper der en forventer at data i en gruppe oppfører seg på lik måte.
- c) Ekvivalensklasseinndeling som teknikk kan brukes for å sikre seg at både input og output blir dekket med sin datavariasjon.
- d) Ekvivalensklasser kan finnes både for gyldige og for ugyldige data.

### 4.2 (K3) Gitt er følgende programkode

```
If (a = 1 AND b < 100)
  Then ...
End if
```

Hvilket av de følgende utsagn er FEIL når en krever 100% av hvert nevnte dekningskriterium?

- a) For å få programinstruksjonsdekning (statement coverage) trenger en 2 testtilfelle.
- b) For å få beslutningsdekning (decision coverage) trenger en 2 testtilfelle.
- c) For å få programinstruksjonsdekning (statement coverage) trenger en 1 testtilfelle.
- d) For å få programinstruksjonsdekning (statement coverage) trenger en 1 testtilfelle, og for beslutningsdekning (decision coverage) trenger en 2 testtilfelle.

### 4.3 (K3) Gitt er følgende pseudokode

```
Integer a;
  If (a > 1 AND a < 50)
    Then ...
  End if
```

Hvilket sett av testverdier er et resultat av en riktig utført grenseverdianalyse?

- a) 0, 1, 2, 50, 51
- b) -1, 0, 1, 49, 50, 51
- c) 1, 2, 49, 50
- d) -32767, -1, 0, 1, 49, 50, 51, +32767

### 4.4 (K3) Hvilke verdier for å teste input til et felt for personnavn er et typisk resultat av feilgjetting?

- a) "Anna", "Åse"
- b) "George W. Bush", "Osama Bin Laden", "007"



- c) ingen verdi gitt, "Mona Åse Stöckheim-Ænesdal", et navn som er svært langt
- d) "Jon D. Olsen", et navn som er svært langt

**4.5 (K3) Et felt på en skjerm skal fylles inn med et positivt heltall opp til to sifre langt. Hvilket resultat av de nedenfor stående er en riktig utført ekvivalensklasseinndeling?**

- a) Ugyldige klasser = ikke verdi gitt / ikke heltall / null eller negativt tall / over 99. Gyldige klasser = verdi gitt / heltall / verdi fra og med 1 til og med 99.
- b) Ugyldige klasser = kommatall / ikke tall / ledende nuller / ledende pluss / null eller negativt tall / over 99. Gyldige klasser = verdi gitt / heltall / verdi fra og med 1 til og med 99.
- c) Ugyldige klasser = ingen verdi gitt / kommatall / null eller negativt tall / over 99. Gyldige klasser = verdi gitt / heltall / verdi fra og med 1 til og med 99.
- d) Ugyldige klasser = kommatall / ikke tall / ledende nuller / ledende pluss / null eller negativt tall / over 100. Gyldige klasser = verdi gitt / heltall / verdi fra og med 2 til og med 99.

**4.6 (K3) Et felt på en skjerm skal fylles inn med en tekst på opp til 10 tegn lengde. Hvilket resultat av de nedenfor stående er en gyldig ekvivalensklasseinndeling?**

- a) Ugyldige klasser = ikke gitt verdi / lengre enn 11 posisjoner. Gyldige klasser = verdi gitt / lengde mellom 1 og 10 tegn.
- b) Ugyldige klasser = ikke gitt verdi / lengre enn 10 posisjoner. Gyldige klasser = verdi gitt / lengde mellom 1 og 10 tegn.
- c) Ugyldige klasser = ikke gitt verdi / andre tegn enn a til z / lengre enn 10 posisjoner. Gyldige klasser = verdi gitt / lengde mellom 1 og 10 tegn / bare bokstaver fra a til z.
- d) Ugyldige klasser = ikke gitt verdi / lengre enn 10 posisjoner. Gyldige klasser = verdi gitt / en klasse for hver av lengdene fra og med 1 til og med 10.

#### 4.7 (K2) Hva er typiske white-box teknikker, sortert stigende etter sin grad av testdekning?

- (a) programinstruksjonsdekning – grenseverdidekning – beslutningsdekning – multiplert beslutningsdekning
- (b) forgreningsdekning – programinstruksjonsdekning – beslutningsdekning – betingelsesdekning
- (c) programinstruksjonsdekning – beslutningsdekning - betingelsesdekning – forgreningsdekning
- (d) programinstruksjonsdekning – beslutningsdekning – multiplert betingelsesdekning

#### 4.8 (K3) Gitt er følgende programkode

**Info:** (FOR... END FOR er en løkke. Alt i mellom de to stikkord utføres like mange ganger som det er elementer i tabellen som du som tester lar programmet lese).

```
BEGIN
  INDEX := 0;
  FOR ALL ELEMENTS IN THE TABLE BELOW DO THIS:
    INCREMENT THE INDEX "INDEX" BY 1;
    IF (AMOUNT (TABLE(INDEX)) > 0)
      THEN PAY
    END IF;
  END FOR;
END
```

Hvor mange ganger må en minst starte dette programmet for 100% programinstruksjonsdekning (statement coverage)?

- (a) 1
- (b) 2
- (c) 4
- (d) mange

**4.9 (K3) For programmet gitt ovenfor:** Hvor mange ganger må en starte programmet minst for 100% stidekning (path coverage)?

- (a) 1
- (b) 2
- (c) 4
- (d) et stort antall

#### 4.10 (K3) Gitt er følgende programkode

```
IF (A > B)
  THEN DO X;
END IF
IF (C IS True)
  THEN DO Y;
END IF
```

Hvor mange testtilfelle behøver en minst for 100% av følgende testdekninger?

- (a) Programinstruksjonsdekning: 1, forgreningsdekning: 1
- (b) Programinstruksjonsdekning: 2, forgreningsdekning: 2
- (c) Programinstruksjonsdekning: 2, forgreningsdekning: 1
- (d) Programinstruksjonsdekning: 1, forgreningsdekning: 2

#### 4.11 (K1) Black box tests

- (a) Lages ut fra spesifikasjonen
- (b) Lages ut fra programkoden
- (c) Lages ut fra en strukturell modell av systemet
- (d) Lages ut fra algoritmene som er bruk

#### 4.12 (K1) Hvor er feilgjetting best å bruke?

- (a) Som første metode for å lage testtilfelle
- (b) Etter at mer formelle testteknikker har blitt brukt
- (c) Av uavhengige testere
- (d) Etter at systemet har vært i bruk en stund

**4.13 (K3) Billettprisen er avhengig av høyden på en person: Under 1,10m er det gratis. Fra og med 1,10m og inntil 1,40m er det halv pris. Over 1,40m er det full pris. Hvilket datasett dekker de gyldige ekvivalensklassene?**

- (a) 0,9 – 1,2 – 1,4
- (b) -1,0 – 0,9 – 1,2 – 2,0
- (c) 0,0 – 1,1 – 1,4
- (d) 1,0 – 1,2 – 2,1

Variant: Hvis grensene var: halv pris fra og med 1,20m til og med 1,50 m, hva er rett svar da?

## Spørsmål til Kap 5 ("Testledelse")

**5.1 (K1) Hvilket av de følgende alternativene viser virkelige ulemper med uavhengig testing?**

- (a) Uavhengige testere trenger ekstra opplæring og koster alltid mer
- (b) Utviklertest og uavhengig test vil overlappe og dermed kaste bort ressurser.
- (c) Utviklere kan miste sin sans for ansvar og uavhengige testere kan bli til en flaskehals.
- (d) Uavhengige testere vil bli en flaskehals og de lager problemer for feilhåndteringen.

**5.2 (K1) Hvilken av de følgende alternativer beskriver arbeidsfordelingen mellom testleder og tester?**

- (a) Testleder planlegger, overvåker og styrer testaktivitetene, mens testeren lager og utfører testene.
- (b) Testleder planlegger, organiserer og styrer testaktivitetene, mens testeren lager, prioriterer og utfører testene.
- (c) Testleder planlegger testaktivitetene og velger standardene som skal følges, mens testeren velger verktøyene og styrer deres bruk.
- (d) Testleder planlegger og organiserer testingen, mens testeren prioriterer og utfører testene.

**5.3 (K1) Hvilken av de følgende alternativer er typiske tester oppgaver (i motsetning til testleder)?**

- (a) Sette opp konfigurasjonsstyring av testmateriale; Review tester utviklet av andre.
- (b) Planlegge og initiere spesifisering, forberedelse, implementasjon og utføring av tester og overvåke og styre utføringen.
- (c) Beslutte hva som skal automatiseres, til hvilken grad, og hvordan.
- (d) Forberede og hente inn test data; Review tester utviklet av andre.

**5.4 (K2) Hvilken av de følgende alternativer er typiske prosjektrisikoer?**

- (a) Potensielle områder som lett kan feile i et system.
- (b) Leverandørproblemer, organisatoriske faktorer og kvaliteten av design, kode og tester.
- (c) Forsinkelser og spesielt komplekse områder i det leverte produkt.
- (d) Lav kvalitet av krav, design, kode og tester, og områder der produktet oftest feiler.

### 5.5 (K2) Hvilken av de følgende alternativer er typiske produktrisikoer?

- (a) Problemer med å definere de rette krav og potensielle områder med mange feil.
- (b) Politiske problemer og forsinkelser av spesielt komplekse områder av produkter.
- (c) Produktet feiler ofte samt følgene for kundene og/eller brukerne.
- (d) Lav kvalitet av krav, design, kode og tester, og områder med lav ytelse.

### 5.6 (K2) Hva er sant om produktrisikoe

- 1) Produktrisikoe blir brukt for å beslutte hva en skal teste tidlig.
  - 2) Produktrisikoe blir brukt for å beslutte hvilke verktøy en skal bruke for testingen.
  - 3) Produktrisikoe blir brukt for å beslutte testingens omfang.
  - 4) Produktrisikoe blir brukt for å beslutte hvor mye en skal teste.
  - 5) Produktrisikoe blir brukt for å beslutte hvilke testteknikker en skal bruke.
- 
- (a) 1, 3 & 4 er riktig, 2 & 5 er feil
  - (b) 1, 2, & 4 er riktig, 3 & 5 er feil
  - (c) 1, 3, 4 & 5 er riktig, 2 er feil
  - (d) 4, & 5 er riktig, 1, 2 & 3 er feil

### 5.7 (K1) Hva er det viktigste punktet som mangler i innholdet til følgende feilrapport?

Dato, godkjenninger, status, omfang, alvor, prioritet, referanser til testtilfelle som er feilet.

- (a) Tittel
- (b) Problemnummer
- (c) Forfatter
- (d) Skjermkopier eller systemlogg

### 5.8 (K1) Hvilken av de følgende er den opprinnelige årsaken til et problem?

- (a) Det var ikke nok tid til å implementere løsningen i samsvar med kravene.
- (b) En beregning var feil, noe som førte til feil resultat.
- (c) Beregningen av kvadratroten var 10 % for lav.
- (d) En test ble feilaktig evaluert som rett, men den var egentlig feil.

# Spørsmål til Kap 6 ("Testverktøy")

## 6.1 (K1) Hva er en testramme?

- (a) Et annet navn for et testtilfelle
- (b) Et strukturert mal for testspesifikasjoner
- (c) Et bibliotek som brukes for å kjøre komponenttester
- (d) Et verktøy som samler og forvalter testresultater

## 6.2 (K1) Her er noen påstander om verktøy for statistisk analyse. Hvilken er riktig?

- (a) Verktøy for statistisk analyse kan hente måledata om koden
- (b) Verktøy for statistisk analyse må utføre testtilfelle for å kunne gjøre målinger i koden.
- (c) Verktøy for statistisk analyse blir best brukt i systemtest fasen.
- (d) Verktøy for statistisk analyse er verdifulle verktøy for regresjonstesting.

## 6.3 (K1) Et verktøy som manipulerer filer eller ekstraherer data fra database tilhører følgende kategori testverktøy:

- (a) Review support verktøy
- (b) Test roboter
- (c) Testledelsesverktøy
- (d) Verktøy for test spesifisering

## 6.4 (K2) Hva er de vanlige vanskelighetene ved testautomatisering?

### 6.4.1: Hva er de vanlige unnskyldningene for vanskeligheter med testautomatisering? 😊

- 1) Verktøyet understøtter ikke alle fasiliteter som brukes i grensesnittet
  - 2) Vedlikehold av testscriptene er for vanskelig og uoversiktlig
  - 3) Utgiftene for verktøylisensen er for dyre
  - 4) Læringen av testverktøyets scriptspråk tar for mange ressurser
- 
- (a) 1 & 4 er riktig, alle andre feil
  - (b) 1,2 & 4 er riktig, 3 er feil
  - (c) 2 og 3 er riktig, alle andre er feil
  - (d) 3 er riktig, alle andre feil

## Løsninger

- 1.1 b
- 1.2 a
- 1.3 b
- 1.4 c
- 1.5  $b+d+e$
- 1.6 c
- 1.7 b

- 2.1 a
- 2.2 d
- 2.3 b
- 2.4  $c+e$
- 2.5 c
- 2.6 b

- 3.1 b
- 3.2 a
- 3.3 a
- 3.4 a
- 3.5 d

- 4.1 a
- 4.2 a
- 4.3 c
- 4.4 c
- 4.5 a
- 4.6 b
- 4.7 d
- 4.8 a
- 4.9 d
- 4.10 d
- 4.11 a
- 4.12 b
- 4.13 d (også ved ny definisjon av grensene er dette rett svar)

- 5.1 c
- 5.2 a
- 5.3 d
- 5.4 b
- 5.5 c
- 5.6 c
- 5.7 c
- 5.8 a

- 6.1 c
- 6.2 a
- 6.3 d
- 6.4 b
- 6.4.1 tenk selv!

Og i en av oppgavene er det lagt inn en alvorlig kunstig feil. Finner du den, så gi beskjed til oss!