

Løsningsforslag til eksamen i Software testing 2021

Del 1

(oppgavene er gitt i villkårlig rekkefølge av Inspira)

1 Multiple choice

Which two of the following are typical activities for a test leader?

Select one or more alternatives:

- Gather and report test progress metrics
- Evaluate test results against exit criteria
- Develop system requirements, design specifications and design models
- Acquire and prepares test data
- Handle all test automation duties

2 Multiple choice

Match each testing level with the corresponding test basis:

	System testing	Acceptance testing	Integration testing	Component testing
Program specifications and requirements	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
System architecture	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
User stories	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Programming code	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

3 Multiple Choice

Which of the following statements best describes Black-box design techniques?

Select one alternative:

- A sequence of test cases in executive order, and any associated actions that may be required to set up the initial preconditions and any wrap-up activities post executions.
- A procedure for deriving and/or selecting test cases based on an analysis of the internal structure of a component or a system.
- A procedure for deriving and/or selecting test cases based on an analysis of the specifications, either functional or non-functional, of a component or system without reference to its internal structure.
- A procedure for deriving and/or selecting test cases based on testers' experience, knowledge and intuition.

4 Multiple choice

Which of these is NOT a *non-functional* quality characteristic?

Select one alternative:

- Portability
- Usability
- Maintainability
- Performance
- Security

5 Multiple choice

Which of these basic approaches to a software development project follows the key elements of user centric design process?

Select one alternative:

- Interview users, develop the software, and perform acceptance test
- Interview users, iteratively develop prototypes, and evaluate the software
- Define requirements, develop the software, and perform acceptance test
- Iteratively develop prototypes, perform expert reviews, and integrate found issues

6 Multiple choice

According to the ISTQB Glossary, which of the following is NOT a test type?

Select one alternative:

- White-box testing
- Change-related testing
- Functional testing
- Component testing

7 Multiple choice

In context of Software testing, the concept of *risk* is a product of two factors. Which ones?

Select one or more alternatives:

- Likelihood
- Management
- Budget
- Quality assurance
- Impact

8 Multiple choice

Which of the following terms is related to White-box testing?

Select one alternative:

- Code coverage
- Regression testing
- Configuration
- Accuracy

9 Multiple choice

The black-box design techniques can be used in several of the areas below. However, you should match them so that all the techniques are applied to the most suitable area.

	Use case testing	EP and BVA*	Decision table	State transition testing
Testing the user interaction in a Graphical User Interface	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testing the functionality of a ticket machine	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Testing an access control system based on several binary conditions	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Testing the interest rate return on a savings account	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

* EP and BVA: Equivalence partitioning and boundary value analysis

10 Multiple choice

Which two of the following are typical activities for a tester?

Select one or more alternatives:

- Gather and report test progress metrics
- Evaluate test results against exit criteria
- Handle all test automation duties
- Acquire and prepares test data
- Develop system requirements, design specifications and design models

11 Multiple choice

Which of these describes a *function* test?

Select one alternative:

- Checking the effect of high volumes of traffic in a call-center system
- Checking how easy it is to use the software system
- Measuring how long it takes to upload a file to a cloud-based solution
- Counting if the number of outputs is as expected

Maximum marks: 1

12 Multiple choice

How does *verification* differ from *validation* in the context of testing?

Select one alternative:

- Verification is related to whether the program works according to its specifications and requirements, while validation is related to whether the program satisfies the needs of the users.
- Verification is related to whether the program satisfies the needs of the users, while validation is related to whether the program works according to its specifications and requirements.
- Verification is related to 100% code coverage, while validation is related to 100% use-case coverage.
- They does not differ. Verification and validation mean the same thing in the context of testing.

13 Multiple Choice

Which of the following statements best describes White-box testing techniques?

Select one alternative:

- A sequence of test cases in executive order, and any associated actions that may be required to set up the initial preconditions and any wrap-up activities post executions.
- A procedure for deriving and/or selecting test cases based on an analysis of the internal structure of a component or a system.
- A procedure for deriving and/or selecting test cases based on an analysis of the specifications, either functional or non-functional, of a component or system without reference to its internal structure.
- A procedure for deriving and/or selecting test cases based on testers' experience, knowledge and intuition.

14 Multiple choice

Which two of the following are NOT a review type, according to the ISTQB Glossary?

Select one or more alternatives:

- Technical review
- Walkthrough
- Examination
- Inspection
- Intrusion detection

15 Multiple choice

Static code analysis identifies all but two of the following problems. Which ones?

Select one or more alternatives:

- Violations of programming standards
- Unreachable code
- Deviation between actual and expected output
- Variables that are declared, but never used
- Memory leaks

Maximum marks: 2

16 Multiple choice

Which of the following test activities is the strongest candidate for automation?

Select one alternative:

- Beta testing
- Usability testing
- Regression testing
- Confirmation testing

Maximum marks: 1

Del 2

17. Case 1- Decision table

Tre betingelser:

- Rushtid
- El-bil (hvis det ikke er elbil, er det bensin- og dieslbiler)
- Passasjer

a)

	R1	R2	R3	R4	R5	R6	R7	R8
Elbil?	true	true	true	true	false	false	false	false
Rushtid?	true	true	false	false	true	true	false	false
Passasjer?	true	false	true	false	true	false	true	false
Pris i NOK	5	10	0	0	40	80	25	50

b)

	R1	R2	R3/R4	R5	R6	R7	R8
Elbil?	true	true	true	false	false	false	false
Rushtid?	true	true	false	true	true	false	false
Passasjer?	true	false	-----	true	false	true	false
Pris	5	10	0	40	80	25	50

Prisen som en elbil må betale utenfor rushtiden (dvs. i dette tilfellet ikke betale fordi den er 0) er **ikke avhengig av om det er passasjer eller ikke**. Derfor kan regel 3 og 4 slår sammen.

c)

Test casene

1. El-bil i rushtid med passasjer
2. El-bil utenfor rushtid uten passasjer
3. Diesebil i rushtid med passasjer
4. Bensinbil utenfor rushtid uten passasjer

dekker av reglene R1, R3/R4, R5 og R8. Reglene merket med rødt viser hvilke regler som mangler test caser.

	R1	R2	R3/R4	R5	R6	R7	R8
Elbil?	true	true	true	false	false	false	false
Rushtid?	true	true	false	true	true	false	false
Passasjer?	true	false	-----	true	false	true	false
Pris	5	10	0	40	80	25	50

Ved å legge til følgende test caser blir alle reglene dekket.

R2: Elbil i rushtiden uten passasjer

R6: Ikke-elbil (bensin eller diesebil) i rushtiden uten passasjer

R7: Ikke-elbil utfor rushtiden med passasjer.

18. EP-BVA

- a) **Delmengdene** må være **disjunkte**, dvs. **de må ikke ha felles elementer**.
Alle delmengdene må til sammen utgjøre hele mengden de er utgått fra, dvs. hvis de er delmengder av en større mengde A, må unionen av alle delmengdene utgjøre hele A.
- b) Programmet må **respondere likt overfor alle elementene innen hver ekvivalenspartisjon**.
- c) **Mengden må kunne sorteres**, elementene må kunne stilles i sortert rekkefølge for at vi skal kunne avgjøre hvilket element som er nærmest grensen mellom to partisjoner.

- d) Vi får følgende **4 partisjoner**:

Elbiler, Ikke-elbiler, busser og vogntog.

Mengdene er ikke sorterbare, og ekvivalenspartisjonene har derfor **ikke grenseverdier**.

- e) Vi får følgende **4 partisjoner**:

2 partisjoner for hver av rushtid-periodene 07:30:00 – 9:30:00 og 15:30:00 – 17:30:00

2 partisjoner for periodene mellom rushtidene 9:30:01 – 15:29:59 og 17:30:01 – 7:29:59

dvs. på dagen og fra ettermiddagen gjennom natten til neste morgen.

Tidspunkter er sorterbare, så **vi får grenseverdier** på begge sider av hver partisjonsgrense.

Eks. 07: 29:59 og 07:30:00 osv.

19. Case 3 State transition

- a) Antall tilstander: 8
Antall overganger: 11

Korteste vei som besøker alle tilstander: **1 -> 2b -> 1 -> 2a -> 3 -> 4 -> 5 -> 6->7.**
Tilstandsdekning er 8/8 = 100%
Overgangsdekning er 8/11= 72,7%

- b) Vei med høyest overgangsdekning
1 -> 2b -> 1 -> 2a -> 2a -> 3 -> 4 -> 5 -> 6 -> 7

Overgangsdekning er 9/11= 81,8%
Tilstandsdekning: 8/8 = 100%

- c) Vi trenger 2 test caser, f. eks.:
1: 1->2b->1->2a->2a->-3->4->5->6->7
2: 1->2b->-3->4->6->7

- d) **Hvis du har 100% overgangsdekning er du garantert å ha 100% tilstandsdekning.**

Del 3

(oppgavene er gitt i villkårlig rekkefølge av Inspira)

1. Formålet med **Entry-kriteriene** er å avgjøre om materialet eller programmet **er klart** for å testes/«reviewes». Her kan det stilles konkrete krav til mange ulike ting som feiltetthet, testomgivelser, eller tidligere utført testing og/eller aktiviteter osv.

Exit-kriteriene skal avgjøre på **når man kan avslutte testingen** i en fase, på et nivå, eller av hele programmet.

Testlederen er ansvarlig for å bestemme **både entry- og exit-kriteriene**, og dette gjøres i ideelt sett **planleggingsfasen**. Under overvåkingen av testingsprosessen må testlederen vurdere om kriteriene skal forandres.

Det er **testlederen** som **sjekker kriteriene** ved inngangen/utgangen av hver testfase/testaktivitet.

2. **Personas** brukes til å beskrive en **fiktiv representant for sluttbrukeren** av systemet. Den er **basert på verifisert informasjon** (statistikk, intervjuer osv.), og **ikke antagelser**.

Personas brukes **under utviklingen** av et program, ideelt sett **fra planleggingsfasen til ferdig produkt**, for i størst mulig grad å kunne **sikre at programmet møter sluttbrukerens behov**.

Eks. Lisa er 70 år, har dårlig syn og svak motorikk grunnet leddgikt. Hun har liten erfaring med data. Osv.

Liten erfaring med data:

Programmet bør være veldig brukervennlig, lett å forstå og lett å lære. De viktigste funksjonene bør være intuitive og gjenkjennelige andre programmer.

Dårlig syn:

Det bør legges vekt på gode kontraster i bildet, tekst som kan forstørres og eventuelt muligheter for at programmet kobles opp mot assisterende verktøy som kan lese teksten høyt for henne. God brukerveiledning, som gjerne leses høyt bør også tas med.

Dårlig motorikk:

Muligheter for tastaturalternativer som kan erstatte bruk av mus.

3.

Komponenttesting/Unittesting bør automatiseres, ellers vil det bli utfordrende å kjøre regresjons testing. Lite egnet for manuell testing.

Integrasjonstesting er også **godt egnet for automatisering**, noe som blant annet gjøres i Continuous Integration, men kan også gjøres manuelt.

Systemtesting kan automatiseres, men skal det være kostnadseffektivt, må samme test kunne kjøres flere ganger. Hvis programmet forandrer seg mye underveis, må man vurdere hvorvidt det er kostnadseffektivt. Mye av denne testingen gjøres i dag manuelt.

Akseptansetesting egner seg minst for automatisering fordi det er nettopp brukerens opplevelse som skal testes. Lite automatisert testing på dette nivået.

Oppsummert. Automatiseringsgraden er omvendt proporsjonalt med testingsnivåene, mest egnet på de laveste nivåene, og minst egnet på de høyeste.

4. **Exploratory testing** er en type **erfaringsbasert testing** som er basert på **testerens egen erfaring, intuisjon og evner**. Testene lages «*on the fly*», og **kan ta alle tilgjengelige metoder i bruk**, gjerne også andre testingsteknikker, f.eks. Black-box-teknikker, **uten** nødvendigvis å ha **tilgang til** den samme **testbasisen** som kreves for å ta i bruk mere formelle metoder. Exploratory testing brukes ofte som **et supplement etter at annen og mer formell testing er utført**, eller **når testerens ikke har tilgang til testbasis** som programspesifikasjoner, krav og programkode. Teknikken **brukes også under tidspress** når man ikke har tid til å utføre mer formell testing.

Stikkord:

- Tidspress
- Mangelfull testbasis
- Som et supplement til mer formell testing

5. **Hovedforskjellen** er at **statiske teknikker ikke kjører programkoden under test**, mens **dynamiske teknikker krever at programkoden blir kjørt.**

Det betyr at de **statiske** kan tas i bruk **allerede i planleggingsfasen** av prosjektet (reviews) og brukes fortrinnsvis frem til og i implementasjonsfasen (statisk analyse),

De **dynamiske** testteknikkene brukes **på de fire testnivåene** komponenttesting, interaksjonstesting, systemtesting og akseptansetesting, samt testing under drift.

Statiske og dynamiske testingsteknikker er **komplementære**, dvs. de **finner forskjellige typer feil**, (de **dynamiske oppdager feil**, mens de **statiske** i større grad **finner årsaken til dem**) og man **bør derfor bruke begge deler**.

Del 4

Essay

(I dette løsningsforslaget har vi bare kommentert hvordan testprinsippene kan integreres, ikke besvart selve oppgaven)

Hvis programmet vi skal utvikle er selvalgt, vil vi spesielt være bevisst på **P7**, og stille oss spørsmålet **om applikasjonen virkelig trengs** og være til nytte for sluttbrukeren.

Testing er **kontekstavhengig**, **P6**, slik at vi må tilpasse testingen de ressursene vi har og risikoen knyttet til programmet. Siden dette er et studentprosjekt og alle er både testere og utviklere, vil testingen være **til dels uformell**.

Vi kan ikke teste alt, **P2**, og derfor må vi **prioritere det som er mest kritisk**. Vi vil utføre en risikoanalyse som så vil ligge til grunn for hva vi velger å prioritere, **P6**.

Vi vil bruke **personas** til å beskrive sluttbrukeren, slik at er **sikre på å utvikle noe som vil bli brukt**, **P7**.

Vi vil ta i bruk **statiske teknikker** som gjør at **testingen kan starte tidlig** i utviklingsforløpet, **P3**. Denne typen testing er **veldig kostnadseffektiv**, og vil gjøre at vi både sparer tid og ressurser, samt sikrer bedre kvalitet på sluttproduktet fordi **vi får luket ut feil tidlig**.

P3: Ulike typer reviews av

- Ide til prosjekt
- Brukerhistoriene
- Brukerkrav
- Funksjonelle kra
- Systemkravene
- Programkode

Utfører uformelle review der gruppemedlemmene sjekker hverandres arbeid, herunder også programkoden.

Tekniske review, der der gruppen sammen gjennomgår og diskuterer tekniske løsninger.

Ulike **walkthrough** der den som har laget noe presenterer det for de andre i teamet.

Bruker Statisk analyse av programmeringskode for å sikre konsistent kodekvalitet og kodelstil.

Statisk analyse kan i tillegg **avdekke cluster** av feil i koden, **P4**.

Implementering: Skrive **unit-tester parallelt med implemneringen**, **TDD**.

I de tidlige fasene av utviklingsarbeidet (før implementasjonen) vil vi samtidig **planlegge test casene** til tilsvarende testnivåene:

- Brukerhistoriene -> Akseptansetester
- Systemspesifikasjoner -> Systemtester
- Systemarkitektur -> Integrasjonstester
- Under utviklingen av koden følge prinsippene for TDD.

Vi vil videre **utføre tester på alle de 4 testnivåene**. Sidene vi i forkant har forberedt test-casene på de ulike nivåene, vil vi spare tid i gjennomføringen av den.

Vi vil være bevist **P5, the pesticide paradox** og **varierte testene** og ikke minst **hva som testes**.

Selv om vi har testet grundig og vi ikke lenger finner noen feil, er vi **beviste på at det kan være gjenstående feil, P1**. Vi vil derfor lage et opplegg for rapportering av feil, der brukerne kan melde inn feil de oppdager, slik at feil kan rettes også etter at programmet er tatt i bruk.