Certificeret Tester

Advanced Test Manager Pensum

Dansk version 2012

International Software Testing Qualifications Board

Advanced Niveau Test Manager underarbejdsgruppe: Rex Black (Chair), Judy McKay (Vice Chair), Graham Bath, Debra Friedenberg, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson, Tsuyoshi Yumoto; 2010-2012.

Revisionshistorie

Version	Dato	Bemærkning
ISEB v1.1	04SEP01	ISEB Practitioner Syllabus
ISTQB 1.2E	SEP03	ISTQB Advanced Level Syllabus from EOQ-SG
V2007	12OCT07	Certified Tester Advanced Level syllabus version 2007
D100626	26JUN10	Incorporation of changes as accepted in 2009, separation of each chapters for the separate modules
D101227	27DEC10	Acceptance of changes to format and corrections that have no impact on the meaning of the sentences.
D2011	31OCT11	Change to split syllabus, re-worked LOs and text changes to match LOs. Addition of BOs.
Alpha 2012	09Feb12	Incorporation of all comments from NBs received from October release.
Beta 2012	26Mar12	Incorporation of comments from NBs received on time from Alpha release.
Beta 2012	07APR12	Beta version submitted to GA
Beta 2012	08JUN12	Copy edited version released to NBs
Beta 2012	27JUN12	EWG and Glossary comments incorporated
RC 2012	15AUG12	Release candidate version - final NB edits included
GA 2012	19OCT12	Final edits and cleanup for GA release
DSTB v1.0	5. juni 2014	Dansk udgave frigivet

Indholdsfortegnelse

Bidragsydere

Dette dokument er skrevet af et hovedteam i International Software Testing Qualifidations Board Advanced Niveau underarbejdsgruppen - Advanced Test Manager: Rex Black (Chair), Judy McKay (Vice Chair), Graham Bath, Debra Friedenberg, Bernard Homès, Paul Jorgensen, Kenji Onishi, Mike Smith, Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

Hovedteamet takker reviewteamet og de nationale boards for deres forslag og input.

Da Advanced Niveau pensummet var skrevet færdig bestod Advanced Niveau arbejdsgruppen af følgende medlemmer (i alfabetisk rækkefølge):

Graham Bath, Rex Black, Maria Clara Choucair, Debra Friedenberg, Bernard Homès (Vice Chair), Paul Jorgensen, Judy McKay, Jamie Mitchell, Thomas Mueller, Klaus Olsen, Kenji Onishi, Meile Posthuma, Eric Riou du Cosquer, Jan Sabak, Hans Schaefer, Mike Smith (Chair), Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

Følgende personer deltog i review arbejdet og anden kommentering af samt afstemninger om indholdet:

Chris van Bael, Graham Bath, Kimmo Hakala, Rob Hendriks, Marcel Kwakernaak, Rik Marselis, Don Mills, Gary Mogyorodi, Thomas Mueller, Ingvar Nordstrom, Katja Piroué, Miele Posthuma, Nathalie Rooseboom de Vries, Geoff Thompson, Jamil Wahbeh, Hans Weiberg.

Dokumentet blev formelt frigivet af ISTQB®'s generalforsamling den 19. oktober 2012.

Følgende personer deltog i bearbejdningen til dansk:

- Oversættelse: Hanne Brandt og Mette Bruhn-Pedersen
- Review: Carsten Jørgensen, Lisbeth Hylke Thomsen m.fl.
- Redigering: Ebbe Munk og Klaus Olsen

0 Introduktion til pensum

0.1 Formålet med dette dokument

Dette pensum er grundlaget for den internationale certificering i softwaretest for testmanager på Advanced niveau. ISTQB® stiller dette pensum til rådighed som følger:

1. Til nationale boards med henblik på oversættelse til deres lokale sprog og til at akkreditere kursusudbydere. Nationale boards kan tilpasse beskrivelsen af pensum til deres specifikke sprogbehov og ændre referencer for at tilpasse sig til deres lokale publikationer
2. Til eksamensboards, for at skrive eksamensspørgsmål på det lokale sprog i henhold til læringsmålene i pensummet
3. Til kursusudbydere for at producere kursusmateriale og finde passende undervisningsmetoder
4. Til certificeringskandidater som forberedelse til eksamen - som del af et kursus eller individuelt
5. Til det internationale software- og systemudviklingssamfund for at fremme software- og systemtestprofessionen, og som grundlag for bøger og artikler.

ISTQB® kan give andre tilladelse til at bruge beskrivelsen af pensum til andre formål, hvilket kræver en skriftlig tilladelse.

0.2 Overblik

Det videregående niveau er sammensat af tre pensa:

- Testmanager
- Testanalytiker
- Teknisk testanalytiker.

Det videregående niveaus overbliksdokument [ISTQB_AL_OVIEW] indeholder følgende information:

- Forretningsværdien af hvert pensum
- Sammendrag af hvert pensum
- Relationer mellem pensa
- Beskrivelse af kognitive niveauer (K-niveauer)
- Bilag.

0.3 Eksaminerbare læringsmål

Læringsmålene understøtter forretningsværdierne og bruges til at udforme eksamen til opnåelse af Advanced Test Manager certificering. Generelt er alle dele af dette pensum eksaminerbart på K1 niveau. Det betyder, at kandidaten skal genkende og huske en term eller et koncept. Læringsmålene på K2, K3 og K4 niveau er vist i starten af det relevante kapitel.

0.4 Dansk oversættelse

Som nævnt ovenfor i afsnit 0.1 kan nationale boards tilpasse beskrivelsen af pensum til deres specifikke sprogbehov. Det har vi i DSTB valgt at gøre i nogle passager af dette pensum for at gøre teksten mere læsbar i den danske oversættelse.

1 Testprocessen – 420 min.

Nøgleord

Slutkriterier, testcase, testlukning, testbetingelse, testkontrol, testdesign, testafvikling, testimplementering, testlog, testplanlægning, testprocedure, testscript, testopsummeringsrapport

Læringsmål for testprocessen

1.2 Testplanlægning, -overvågning og -kontrol

TM-1.2.1 (K4) Analyser testbehovene for et system for at planlægge testaktiviteter og arbejdsprodukter, så man opfylder testens formål.

1.3 Testanalyse

TM-1.3.1 (K3) Anvend sporbarhed til at kontrollere at de definerede testbetingelser er fuldstændige og stemmer med testformål, teststrategi og testplan.

TM-1.3.2 (K2) Forklar de faktorer der kan påvirke detaljeringsgraden for specifikation af testbetingelserne. Forklar fordele og ulemper ved at specificere testbetingelser på detaljeret niveau.

1.4 Testdesign

TM-1.4.1 (K3) Anvend sporbarhed til at kontrollere at designede testcases er komplette og i overensstemmelse med de definerede testbetingelser.

1.5 Testimplementering

TM-1.5.1 (K3) Anvend risici, prioritering, testmiljø, dataafhængigheder og begrænsninger til at udarbejde en komplet plan for testafviklingen, som er i overensstemmelse med testformål, teststrategi og testplan.

1.6 Testafvikling

TM-1.6.1 (K3) Anvend sporbarhed til at overvåge testens fremdrift og overensstemmelse med testformål, teststrategi og testplan.

1.7 Evaluering af slutkriterier og rapportering

TM-1.7.1 (K2) Forklar betydningen af en præcis og rettidig indsamling af oplysninger i løbet af testprocessen for at opnå en nøjagtig rapportering og evaluering af slutkriterierne.

1.8 Testlukningsaktiviteter

TM-1.8.1 (K2) Opsummér de fire grupper af testlukningsaktiviteter.

TM-1.8.2 (K3) Gennemgå projektet ved afslutningen for at bedømme processerne og for at finde områder, som kan forbedres.

1.1 Introduktion til pensum

ISTQB® pensum for Foundation-niveauet beskriver den grundlæggende testproces, der omfatter følgende aktiviteter:

- Planlægning og kontrol
- Analyse og design
- Implementering og afvikling
- Evaluering af slutkriterier og rapportering
- Testlukningsaktiviteter.

Pensum for Foundation-niveauet nævner, at selvom aktiviteterne logisk set er sekventielle, kan aktiviteterne i processen overlappe eller finde sted samtidigt. Det er som regel nødvendigt at tilpasse disse hovedaktiviteter inden for rammerne af systemet og projektet.

I dette pensum for Advanced Test Manager betragtes nogle af disse aktiviteter særskilt for at uddybe og optimere processerne yderligere, for bedre at passe til softwareudviklingslivscyklussen, og for at lette effektiv testovervågning og -kontrol. Aktiviteterne behandles nu som følger:

- Planlægning, overvågning og kontrol
- Analyse
- Design
- Implementering
- Afvikling
- Evaluering af slutkriterier og rapportering
- Testlukningsaktiviteter.

1.2 Testplanlægning, overvågning og kontrol

Dette afsnit fokuserer på processerne for planlægning, overvågning og kontrol af test. Som drøftet på Foundation-niveauet er disse aktiviteter en del af testmanager-rollen.

1.2.1 Testplanlægning

For hver testniveau starter testplanlægning ved initieringen af testprocessen for niveauet og fortsætter gennem hele projektet indtil afslutningen af testlukningsaktiviteterne for det niveau. Det indebærer identifikation af de aktiviteter og ressourcer, der er nødvendige for at opfylde teststrategiens mission og formål. Testplanlægning består også i at finde metoder til at indsamle og følge de målinger, der vil blive brugt til at styre projektet, til at afgøre om planen overholdes og til at vurdere opfyldelse af formålene. Ved at vælge de rette metrikker i planlægningsfasen kan man vælge værktøjer, planlægge uddannelse og etablere retningslinjer for dokumentation.

Den valgte strategi (eller strategier) er med til at afgøre hvilke opgaver der skal løses i planlægningsfasen. Hvis man for eksempel anvender en risikobaseret teststrategi (se kapitel 2), kan analysen af risici guide testplanlægningsprocessen. Det gælder både valg af aktiviteter til at afhjælpe de identificerede produktrisici og beredskabsplanlægning. Hvis man finder sandsynlige og potentielt alvorlige fejl relateret til sikkerhed, bør man gøre en betydelig indsats for at udvikle og gennemføre sikkerhedstest. Ligeledes kan testmanageren planlægge mere statisk test (reviews) af designdokumenter, hvis han finder ud af, at alvorlige fejl som regel findes i disse dokumenter.

Oplysninger om risici kan også bruges til at fastsætte prioriteten af de forskellige testaktiviteter. For eksempel, hvor systemets performance udgør en høj risiko, kan performancetest udføres, så snart

integreret kode er tilgængelig. Tilsvarende kan det give mening at planlægge med at der skal laves testchartre og værktøjer til dynamisk test, hvis man følger en reaktiv strategi.

Det er desuden i testplanlægningsfasen at tilgangen til test klart defineres af testmanageren, herunder hvilke testniveauer, der vil blive anvendt. Ligeledes defineres målsætningerne og formålene for hvert niveau, og de testteknikker, der vil blive brugt på hvert testniveau. For eksempel kræver risikobaseret test af visse flyelektroniksystemer en risikovurdering af hvilket niveau af kodedækning der er nødvendigt, og dermed hvilke testteknikker der skal anvendes.

Komplekse relationer kan forekomme mellem testgrundlaget (f.eks. specifikke krav eller risici), testbetingelser og de testcases, der dækker dem. Der eksisterer ofte mange-til-mange-relationer mellem disse arbejdsprodukter. Disse skal forstås for at det er muligt at gennemføre en effektiv testplanlægning, overvågning og kontrol. Beslutninger vedrørende værktøjer kan også afhænge af forståelsen af forholdene mellem arbejdsprodukter.

Der kan også findes relationer mellem arbejdsprodukter produceret af udviklingsteamet og testgruppen. For eksempel kan det være nødvendigt at sporbarhedsmatricen fastholder relationerne mellem de detaljerede elementer i designspecifikationen fra systemets designere, de forretningsmæssige krav fra forretningsanalytikerne og de testarbejdsprodukter, der er defineret af testgruppen. Hvis der skal designes og anvendes lav-niveau testcases, kan der i planlægningsfasen være defineret et krav om, at de detaljerede designdokumenter fra udviklingsteamet skal godkendes, før udformningen af testcases kan begynde. Når man følger en agil livscyklus, kan uformelle informationsudvekslingsmøder blive brugt til at formidle information mellem teams før testen startes.

Testplanen kan også nævne de specifikke funktioner i softwaren, som er indenfor anvendelsesområdet (eventuelt baseret på en risikoanalyse) og identificere funktioner udenfor anvendelsesområdet. Afhængig af det formelle niveau for dokumentation i projektet kan alle funktioner indenfor anvendelsesområdet kombineres med en tilsvarende testdesignspecifikation.

Det kan også være nødvendigt at testmanageren samarbejder med projektets arkitekter for at definere den første testmiljøspecifikation, for at sikre at de nødvendige ressourcer er til rådighed, for at sikre sig at de folk, der skal konfigurere miljøet har forpligtet sig til at gøre det og for at forstå omkostnings-/leveringstidsplaner samt arbejdet med at færdiggøre og levere testmiljøet.

Endelig bør man identificere alle eksterne afhængigheder og deres tilhørende Service Level Agreements (SLA'er). Om nødvendigt må man tage kontakt til disse. Eksempler på afhængigheder er forespørgsler på ressourcer fra eksterne grupper, afhængigheder af andre projekter (hvis arbejdet foregår i et program), eksterne leverandører eller udviklingspartnere, idriftsættelsesteam og databaseadministratorer.

1.2.2 Testovervågning og kontrol

For at en testmanager kan lave en effektiv testkontrol, skal der etableres en testtidsplan og rammemodel for overvågningen, som muliggør opfølgning på testarbejdsprodukter og ressourcer imod planen. Denne rammemodel bør omfatte de specifikke målinger og målsætninger, som er nødvendige for at relatere status af testarbejdsprodukter og aktiviteter til planen og strategiske formål.

For små og mindre komplekse projekter kan det være forholdsvis let at forholde testarbejdsprodukter og aktiviteter til planen og strategiske formål, men generelt er der behov for at definere mere detaljerede formål for at opnå dette. Det kan omfatte målingerne og målsætningerne for at opfylde testformålene og dækningen af testgrundlaget.

Af særlig betydning er behovet for at relatere status af testarbejdsprodukter og aktiviteter til testgrundlaget på en måde, der er forståelig og relevant for projektet og forretningens interessenter. At definere målepunkter og måling af fremdrift på basis af testbetingelserne og grupper af testbetingelser kan bruges som et middel til at opnå dette ved at relatere andre testarbejdsprodukter til testgrundlaget via testbetingelserne.
Korrekt konfigureret sporbarhed, bl.a. muligheden for at rapportere sporbarhedsstatus, gør det komplekse forhold, der eksisterer mellem udviklingsarbejdsprodukter, testgrundlaget og testarbejdsprodukterne mere gennemsigtige og forståelige.

Nogle gange har de detaljerede tiltag og målsætninger, som interessenterne kræver overvåget ikke direkte sammenhæng med systemets funktionalitet eller en specifikation, især hvis der er ringe eller ingen formel dokumentation.

For eksempel kan en interessent fra forretningen være mere interesseret i at etablere dækning af en driftscyklus i forretningen, selvom specifikationen er defineret i forhold til systemets funktionalitet. Inddragelse af interessenter fra forretningen på et tidligt tidspunkt i et projekt kan hjælpe med at definere disse tiltag og målsætninger, som ikke kun kan bidrage til at give bedre kontrol i løbet af projektet, men som også kan bidrage til at drive og påvirke testaktiviteterne gennem hele projektet.

For eksempel kan interessenters foranstaltninger og målepunkter resulterer i strukturering af arbejdsprodukter og/eller testudførelsesplaner for at muliggør en nøjagtig overvågning af testfremdriften op mod disse målepunkter. Disse målepunkter kan også bidrage til sporbarhed for et specifik testniveau og kan sikre informationssporbarhed på tværs af testniveauer.

Testkontrol er en vedvarende aktivitet. Den indebærer sammenligning af faktiske fremdrift med planen og gennemføring af korrigerende handlinger, når det er nødvendigt. Testkontrol sikre at testen sigter mod at opfylde missionen, strategier og formål, herunder revision af testplanlægningsaktiviteterne efter behov. Passende reaktioner på kontroldata afhænger af detaljeret planlægningsinformation.

Indholdet af testplanlægningsdokumenter og testkontrolaktiviteter er dækket i kapitel 2.

1.3 Testanalyse

I stedet for at behandle testanalyse og -design samlet som beskrevet i pensum for Foundation-niveauet betragtes de her i pensum for Advanced Test Manager som separate aktiviteter, selv om de kan gennemføres som parallelle, integrerede, eller iterative aktiviteter.

Testanalyse er den aktivitet, der definerer "hvad" der skal testes i form af testbetingelser. Man kan finde testbetingelserne ved at analysere testgrundlag, testformål og produktrisici. De kan ses som de detaljerede tiltag og målsætninger for at få succes (f.eks. som en del af slutkriterierne) og bør kunne spores tilbage til testgrundlaget og de definerede strategiske mål, bl.a. testformål og andre projekt- eller interessentspecifikke succeskriterier. Testbetingelser bør også kunne spores frem til testdesign og andre testarbejdsprodukter, efterhånden som disse arbejdsprodukter bliver klar.

Testanalyse for et givet testniveau kan udføres, så snart grundlaget for test er etableret for det niveau. Formelle testteknikker og andre generelle analytiske teknikker (f.eks. analytiske risikobaserede strategier og analytiske kravbaserede strategier) kan anvendes til at finde testbetingelserne. Testbetingelser kan angive eller udelade værdier eller variable afhængigt af testniveauet, oplysningerne der foreligger på tidspunktet for analysen og den valgte detaljeringsgrad (dvs. detaljeringsgraden i dokumentationen).

Der er en række faktorer at overveje, når der træffes beslutning om hvor detaljeret testbetingelser skal specificeres, herunder:

- Testniveauet
- Detaljerings- og kvalitetsniveauet af testgrundlaget
- Kompleksiteten af systemer/software
- Projekt- og produktrisici
- Forholdet mellem testgrundlaget, hvad der skal testes og hvordan det skal testes
- Den anvendte softwareudviklingslivscyklus
- Teststyringsværktøjer der skal anvendes
- Det niveau som testdesign og andre testarbejdsprodukter specificeres og dokumenteres på
- Testanalytikerens færdigheder og viden
- Modenhedsgraden i testprocessen og selve organisationen (bemærk at større modenhed kan kræve større detaljeringsgrad eller tillade et lavere detaljeringsniveau)
- Adgang til vejledning fra andre projektinteressenter.

En detaljeret specifikation af testbetingelser kan resultere i et større antal testbetingelser. For eksempel kan du have en enkelt generel testbetingelse, "Test indkøbskurven", for en e-handel applikation. Dog kan dette blive opdelt i flere testbetingelser i et detaljeret testbetingelsesdokument, med én betingelse for hver understøttet betalingsmåde, én betingelse for hvert muligt destinationsland og så videre.

Nogle af fordelene ved at angive testbetingelser på et detaljeret niveau er at det:

- Giver mere fleksibilitet når andre testarbejdsprodukter (f.eks. testcases) skal forbindes til testgrundlaget og testformålene, hvilket giver en bedre og mere detaljeret overvågning og kontrol for en testmanager
- Bidrager til at forebygge fejl da det kan foregå tidligt i projektet så snart der findes et grundlag for test. Potentielt set kan det ske før systemarkitektur og detaljeret design er på plads (Også beskrevet i Foundation pensum).
- Relaterer testarbejdsprodukter til interessenter på en måde som de kan forstå (ofte betyder testcases og andre testarbejdsprodukter intet for interessenter i forretningen og simple målinger som f.eks. antallet af testcases afviklet har ingen betydning i forhold til interessenternes krav til dækning)
- Hjælper med at påvirke og styre ikke blot testaktiviteter men også andre udviklingsaktiviteter
- Muliggør optimering af testdesign, implementering og afvikling sammen med de resulterende arbejdsprodukter ved en mere effektiv dækning af detaljerede tiltag og målsætninger
- Giver grundlaget for klarere horisontal sporbarhed inden for et testniveau.

Nogle af ulemperne ved at angive testbetingelser på et detaljeret niveau er:

- Tidskrævende
- Vedligeholdelse kan blive vanskeligt i et skiftende miljø
- Formalitetsniveauet skal defineres og implementeres i hele teamet.

Specifikation af detaljerede testbetingelser kan være særlig effektivt i følgende situationer:

- Letvægtsmetoder til testdesigndokumentation, såsom tjeklister, bliver brugt til at imødekomme udviklingslivscyklusen, omkostnings- og/eller tidsbegrænsningerne eller andre faktorer
- Få eller ingen formelle krav eller andre udviklingsarbejdsprodukter er tilgængelige som testgrundlag
- Store, komplekse eller høj risikoprojekter kræver et niveau af overvågning og kontrol, der ikke kan leveres ved blot at relatere testcases til udviklingsarbejdsprodukter.

Testbetingelser kan specificeres med færre detaljer, når testgrundlaget let kan relateres direkte til testdesign arbejdsprodukter. Det er mere sandsynligt i følgende tilfælde:

- Test på komponentniveau
- Mindre komplekse projekter hvor der er simple hierarkiske relationer mellem hvad der skal testes, og hvordan det skal testes
- Accepttest hvor usecases kan anvendes til at hjælpe med at definere testcases.

1.4 Testdesign

Testdesign er den aktivitet, der definerer "hvordan" noget skal testes. Det er identifikation af testcases ved trinvis uddybning af testbetingelserne eller testgrundlaget ved hjælp af testteknikker beskrevet i teststrategien og/eller testplanen.

Afhængigt af de metoder, der anvendes til testovervågning, testkontrol og sporbarhed, kan testcases være direkte relateret (eller indirekte relateret via testbetingelserne) til testgrundlaget og de definerede formål. Disse formål omfatter strategiske formål, testformål og andre succeskriterier for projektet eller andre interessenter.

Testdesign for et givet testniveau kan udføres, når man har fundet testbetingelserne og der er tilstrækkelig information til at producere testcases på det niveau man har valgt. For lavere testniveauer er det sandsynligt, at testanalyse og design vil blive gennemført som en integreret aktivitet. For højere testniveauer er det sandsynligt, at testdesignet er en separat aktivitet som følger efter testanalysen.

Det er sandsynligt at nogle opgaver, der normalt opstår under testimplementering, bliver integreret i testdesignprocessen, hvis man anvender en iterativ tilgang. Det gælder f.eks. oprettelsen af testdata. Faktisk kan denne fremgangsmåde optimere dækningen af testbetingelser ved enten at skabe lav-niveau eller høj-niveau testcases som en del af processen.

1.5 Testimplementering

Testimplementering er den aktivitet, hvor test skal organiseres og prioriteres af testanalytikere. I formelt dokumenterede sammenhænge er testimplementering den aktivitet, hvor testdesign implementeres som konkrete testcases, testprocedurer og testdata. Nogle organisationer som følger IEEE 829 [IEEE829] standarden definerer input og deres tilknyttede forventede resultater i testcasespecifikationer og testtrin i testprocedurespecifikationer. Det er mere almindeligt at hver tests input, forventede resultater og testtrin dokumenteres sammen. Testimplementering omfatter også oprettelsen af lagrede testdata (f.eks. i flade filer eller databasetabeller).

Testimplementering indebærer også den sidste kontrol, der sikrer at testgruppen er klar til at testafviklingen kan finde sted. Kontrollen kan omfatte leveringen af det krævede testmiljø, testdata og kode (muligvis afvikle nogle test af testmiljøet og/eller accepttest af koden), og at alle testcases er blevet skrevet, reviewet og er klar til at blive kørt. Det kan også omfatte kontrol af eksplicitte og implicitte startkriterier for testniveauet (se afsnit 1.7). Testimplementering kan også omfatte udvikling af en detaljeret beskrivelse af testmiljøet og testdata.

Detaljeringsgraden og tilhørende kompleksitet af arbejdet under testimplementering kan blive påvirket af detaljer i testarbejdsprodukterne (f.eks. testcases og testbetingelser). I nogle tilfælde omfatter testcases detaljerede beskrivelser af hvordan de skal afvikles trin for trin for at sikre at det gøres pålideligt og ensartet uanset hvem der gør det. Som oftest når testcases arkiveres med henblik på langsigtet genbrug i regressionstest. Hvis der findes lovgivning på området, bør tests dokumentere overholdelsen af gældende standarder (se afsnit 2,9).

Under testimplementering bør rækkefølgen for kørslen af manuelle og automatiske tests indgå i en plan for testafvikling. Testmanagere bør nøje kontrollere for begrænsninger, herunder risici og prioriteringer, der kan kræve at test køres i en bestemt rækkefølge eller på særligt udstyr. Afhængigheder af testmiljøet eller testdata skal være kendt og kontrolleret.

Der kan være nogle ulemper ved tidlig testimplementering. Med en Agil livscyklus kan koden for eksempel ændres dramatisk fra iteration til iteration, hvilket gør en stor del af implementeringsarbejdet forældet. Selv uden en livscyklus som er så tilbøjelig til at indeholde forandringer som den Agile, kan enhver iterativ eller inkrementel livscyklus give betydelige ændringer mellem iterationerne, hvilket gør beskrevene tests upålidelige eller genstand for et stort vedligeholdelsesbehov. Det samme gælder for dårligt forvaltede sekventielle livscyklusser, hvor kravene ofte ændres, selv sent i projektet. Inden en omfattende testimplementeringsindsats påbegyndes er det klogt at forstå softwareudviklingslivscyklussen og forudsigeligheden af de softwareegenskaber, der vil være til rådighed i testen.

Der kan være nogle fordele ved tidlig testimplementering. For eksempel giver konkrete test bearbejdede eksempler på, hvordan softwaren skal opføre sig, hvis den er skrevet i overensstemmelse med testgrundlaget. Forretningsdomæneeksperter vil sandsynligvis finde verifikation af konkrete tests nemmere end verifikation af abstrakte forretningsregler, og kan derved finde yderligere svagheder i softwarespecifikationer. Sådanne verificerede tests kan give lysende illustrationer af den påkrævede systemopførsel til softwaredesignere og -udviklere.

1.6 Testafvikling

Testafvikling begynder, når testobjektet er leveret og startkriterierne for testafvikling er opfyldt. Tests skal udformes eller i det mindste defineres før testafvikling. Værktøjer skal være på plads, især for teststyring, håndtering af fejl og (hvis relevant) automatisering af testafvikling. Opfølgning på testresultater, bl.a. opfølgning på målinger, skal virke og de data som indsamles skal forstås af alle teammedlemmer. Testmanageren skal desuden sørge for at der er tilgængelige standarder for testlogning og fejlrapportering. Når testmanageren sikrer at alle disse ting er på plads før testafviklingen starter, kan testen forløbe effektivt.

Tests skal udføres i henhold til testcases, selvom testmanageren bør overveje at tillade et vist spillerum, således at testeren kan dække yderligere interessante testscenarier og adfærd, der observeres under testen. Når man følger en teststrategi, der til dels er reaktiv, skal der afsættes tid til testsession med erfaringsbaserede og fejlbaserede teknikker. Selvfølgelig skal enhver afvigelse, som opdages under en sådan ubeskrevet test, beskrive variationerne fra den skrevne testcase, hvilket er nødvendigt for at kunne reproducere fejlen. Automatiserede tests vil følge deres definerede instruktioner uden at afvige.

Den vigtigste rolle for en testmanager under testafviklingen er at overvåge fremdriften i henhold til testplanen og om nødvendigt at indlede og gennemføre kontroltiltag for at styre testen mod en vellykket afslutning i henhold til mission, formål og strategi. For at gøre det, kan testmanageren bruge sporbarhed fra testresultaterne tilbage til testbetingelserne, testgrundlaget og ultimativt testformålene, og også fra testformålene frem til testresultaterne. Denne proces er beskrevet i detaljer i afsnit 2.6.

1.7 Evaluering af slutkriterier og rapportering

Dokumentation og rapportering til overvågning og kontrol af testfremdrift er detaljeret beskrevet i afsnit 2.6.

Fra testprocessens synspunkt er det vigtigt, at der er egnede processer på plads for at skaffe den nødvendige information til rapportering og til evaluering af slutkriterier.

Definition af kravene til information og metoderne til indsamling indgår i testplanlægning, overvågning og kontrol. Under testanalyse, testdesign, testimplementering og testafvikling bør testmanageren sikre, at medlemmerne i testgruppen giver de nødvendige oplysninger på en nøjagtig og rettidig måde for en opnå en effektiv evaluering og rapportering.

Hyppigheden og det krævede detaljeringsniveau for rapportering er afhængig af projektet og organisationen. Rapporteringen bør forhandles under testplanlægningsfasen og bør drøftes med projektets relevante interessenter.

1.8 Testlukningsaktiviteter

Når testafviklingen er fastsat til at være afsluttet, bør de vigtigste resultater blive opsamlet og enten givet videre til den relevante person eller arkiveret. Samlet set er de testlukningsaktiviteter. Testlukningsaktiviteter inddeles i fire hovedgrupper:

1. **Kontrol af om testen er afsluttet** – sikring af at alt testarbejde faktisk er afsluttet. For eksempel bør alle planlagte test enten være kørt eller bevidst sprunget over, og alle kendte defekter bør enten være rettet og gentestet, udskudt til en fremtidig frigivelse eller accepteret som permanente begrænsninger
2. **Overlevering af testartefakter** – levering af værdifulde arbejdsprodukter til dem, der har brug for dem. For eksempel bør kendte defekter, som enten er udskudt eller accepteret, meddeles til dem, der vil bruge og understøtte brugen af systemet. Test og testmiljøer bør gives til dem, der er ansvarlige for vedligeholdelsestesten. Regressionstestsæt (enten automatiserede eller manuelle) skal dokumenteres og leveres til vedligeholdelsesteamet
3. **Tilegnede erfaringer** - afholdelse eller deltagelse i retrospektive møder, hvor vigtige erfaringer (både fra selve testprojektet og fra hele softwareudviklingslivscyklusen) kan dokumenteres. På møderne lægges der planer, der skal sikre, at gode praksisser kan gentages og dårlige praksisser enten ikke gentages eller hvor udfordringer ikke kan løses, at de er indpasset i projektplaner. Følgende områder skal overvejes:
    a. Var repræsentationen af brugere i analysesessionerne af kvalitetsrisici bred nok? For eksempel, ved en for sen opdagelse af uforudsete defektklynger kunne teamet måske have opdaget, at et bredere udsnit af brugerrepræsentanter burde deltage i analysesessionerne af kvalitetsrisici på fremtidige projekter.
    b. Var estimaterne korrekte? For eksempel kan estimater være betydeligt fejlvurderet og derfor bliver fremtidige estimeringsaktiviteter nødt til at tage højde for det sammen med de underliggende årsager, f.eks. var testen ineffektiv eller var estimatet faktisk lavere end det burde have været
    c. Hvad er tendenserne og resultaterne af årsags- og virkningsanalyse af defekterne? For eksempel, vurdere om sene ændringsanmodninger påvirkede kvaliteten af analysen og udviklingen, kigge efter tendenser, der viser dårlig praksis, f.eks. springe et testniveau over, der ville have fundet defekter tidligere og på en mere omkostningseffektiv måde, for at spare tid. Kontrollere om tendenser i defekter kunne relateres til områder som nye teknologier, personaleændringer, eller manglen på færdigheder.
    d. Er der muligheder for procesforbedringer?
    e. Var der nogen uventede afvigelser fra planen, der skal tages højde for i den fremtidige planlægning?
4. **Arkivering** af resultater, logs, rapporter og andre dokumenter og arbejdsprodukter i konfigurationsstyringssystemet. For eksempel bør både testplanen og projektplanen

opbevares i et planlægningsarkiv med en klar sammenkædning til systemet og versionen, som de blev brugt på.

Disse opgaver er vigtige og bliver tit overset. De bør inddrages i testplanen. Det er almindeligt, at en eller flere af disse opgaver udelades, som regel på grund af for tidlig omplacering eller afskedigelse af projektteamets medlemmer, ressource- eller tidspres på efterfølgende projekter, eller udbrændthed i teamet. På projekter udført under kontrakt, f.eks. kundetilpasset udvikling, bør de nødvendige opgaver stå i kontrakten.

2 Teststyring – 750 minutter

Nøgleord

Testniveauplan, hovedtestplan, produktrisiko, projektrisiko, kvalitetsrisiko, risiko, risikoanalyse, risikovurdering, risikoidentifikation, risikoniveau, risikostyring, risikoreduktion, risikobaseret test, testtilgang, testbetingelser, testkontrol, testdirektør, testestimering, testmanager, testniveau, teststyring, testovervågning, testplan, testpolitik, teststrategi, Wide Band Delphi

Læringsmål for teststyring

2.2 Testledelse i sammenhæng

TM-2.2.1 (K4) Bestem de optimale testaktiviteter ved at analysere interessenter, omstændigheder og softwareprojektets eller softwareprogrammets behov inklusive softwareudviklings-livscyklusmodellen.
TM-2.2.2 (K2) Forstå hvordan softwareudviklingslivscyklusaktiviteter og arbejdsprodukter påvirker test og hvordan test påvirker softwareudviklingslivscyklusaktiviteter og arbejdsprodukter.
TM-2.2.3 (K2) Forklar hvordan man håndterer ledelsesmæssige problemer med erfaringsbaseret test og ikke-funktionel test.

2.3 Risikobaseret test og prioritering af testen

TM-2.3.1 (K2) Forklar de forskellige måder som risikobaseret test forholder sig til risici.
TM-2.3.2 (K2) Forklar ved hjælp af eksempler forskellige teknikker til analyse af produktrisici.
TM-2.3.3 (K4) Analysér, identificér og vurdér de forskellige produktkvalitetsrisici. Opsummér disse risici og deres vurderede risikoniveau set ud fra nøgleinteressenternes perspektiv.
TM-2.3.4 (K2) Beskriv hvordan identificerede kvalitetsrisici kan reduceres og styres i forhold til deres vurderede risikoniveau igennem deres livscyklus og testprocessen.
TM-2.3.5 (K2) Giv eksempler på muligheder for at udvælge og prioritere tests og fordele indsatsen.

2.4 Testdokumentation og andre arbejdsprodukter

TM-2.4.1 (K4) Analyser eksempler på testpolitikker og teststrategier og udarbejd hovedtestplaner, testniveauplaner og andre testarbejdsprodukter som er fuldstændige og i overensstemmelse med disse dokumenter.
TM-2.4.2 (K4) Analysér risici for et givet projekt og vælg passende muligheder for risikostyring (reduktion, beredskab, overførsel og/eller accept).
TM-2.4.3 (K2) Beskriv ved hjælp af eksempler hvordan teststrategier påvirker testaktiviteter.
TM-2.4.4 (K3) Definér normer og skabeloner for testarbejdsprodukter som passer til organisation, livscyklus og projektets behov. Brug skabeloner fra standardiseringsorganisationer, hvis muligt.

2.5 Testestimering

TM-2.5.1 (K3) Estimér alle aktiviteter i testprocessen for et givet projekt ved hjælp af alle relevante estimeringsteknikker.
TM-2.5.2 (K2) Forstå og giv eksempler på hvad der kan påvirke testestimaterne.

2.6 Definition og brug af testmetrikker

TM-2.6.1 (K2) Beskriv og sammenlign typiske testrelaterede metrikker.
TM-2.6.2 (K2) Sammenlign de forskellige dimensioner i overvågning af testfremdrift.
TM-2.6.3 (K4) Analysér og rapporter testresultater i form af tilbageværende risici, status for defekter, status for testafvikling, status for testdækning og tillid for at give indblik og anbefalinger som hjælper projektinteressenterne til at tage beslutninger om frigivelse.

2.7 Den forretningsmæssige værdi af test

TM-2.7.1 (K2) Giv eksempler på hver af de fire kategorier som bestemmer omkostningerne ved en givet kvalitet.

TM-2.7.2 (K3) Estimér værdien af testen på grundlag af kvalitetsomkostningerne og andre kvantitative og kvalitative overvejelser. Kommuniker estimatet til testinteressenterne.

2.8 Distribueret, outsourcet og insourcet test

TM-2.8.1 (K2) Forstå de nødvendige faktorer for at få succes med personalestrategier der benytter distribuerede, outsourcede eller insourcede testgrupper.

2.9 Styring af standarder for softwaretest

TM-2.9.1 (K2) Opsummér kilder til og anvendelser af standarder inden for softwaretest.

2.1 Introduktion

På avanceret niveau sker der en karrieremæssig specialisering for professionelle testere. Dette kapitel beskriver den viden som testprofessionelle har behov for i ledende roller som f.eks. testkoordinator og testmanager. Forskellige organisationer har forskellige definitioner af titler og ansvarsområder for personer i sådanne roller. I dette pensum henviser vi samlet til disse fagfolk som testmanagere.

2.2 Testledelse i sammenhæng

Det er en central opgave for en leder er at sikre og udnytte ressourcer (medarbejdere, software, hardware, infrastruktur osv.) til at udføre værdiskabende processer. For software og it-chefer er sådanne processer ofte en del af et projekt eller et program, der sigter på at levere software eller et system til intern eller ekstern brug. For testmanagere drejer det sig om de processer, der er indgår i test, især i den grundlæggende testproces beskrevet i pensum for Foundation-niveauet og i kapitel 1 i dette pensum. Testprocesser skaber kun værdi ved at bidrage til projektets eller programmets samlede succes eller ved at forhindre alvorlige afvigelser. Derfor skal testmanageren planlægge og styre testprocesserne i den forbindelse. Testmanageren skal arrangere testprocesserne på passende vis, herunder de tilhørende aktiviteter og arbejdsprodukter, i henhold til de andre interessenters behov og situation, deres aktiviteter (f.eks softwareudviklingslivscyklus hvori test finder sted), og deres arbejdsprodukter (f.eks kravspecifikationer).

2.2.1 Forstå testinteressenter

Folk er testinteressenter når de har en interesse i testaktiviteterne, testarbejdsprodukterne eller kvaliteten af de endelige system eller leverance. Interessenters interesse kan skyldes direkte eller indirekte involvering i testaktiviteter, direkte eller indirekte modtagelse af testarbejdsprodukter, eller direkte eller indirekte påvirkning af kvaliteten af leverancerne produceret af projektet eller programmet.

Testinteressenter varierer afhængig af projektet, produktet, organisationen, og andre faktorer og kan omfatte følgende roller:

- **Udviklere, udviklingsledere og udviklingsansvarlige** udvikler softwaren som skal testes, modtager testresultater og må ofte agere ud fra på disse resultater (f.eks. rette rapporterede fejl)
- **Databasearkitekter, systemarkitekter og designere** designer softwaren, modtager testresultater og må ofte agere ud fra på disse resultater
- **Marketings- og forretningsanalytikere** bestemmer hvilke funktioner der skal være i softwaren og kvalitetsniveauet for dem. De er også ofte involveret i at definere behovet for testdækning, review af testresultater, og de træffer beslutninger baseret på testresultater
- **Topledelsen, produktansvarlige og projektsponsorer** er ofte inddraget i at definere behovet for testdækning, reviewer testresultater og træffer beslutninger baseret på testresultater
- **Projektledere** er ansvarlige for at styre deres projekter. Det kræver en prioritering mellem kvalitet, tidsplan og funktioner. En projektleder arbejder sammen med testmanageren om testplanlægning og testkontrol og skaffer de nødvendige ressourcer til testen.
- **Teknisk support, kundesupport og helpdesk-medarbejdere** hjælper brugerne og kunderne med at få gavn af funktioner og kvalitet i den leverede software
- **Direkte og indirekte brugere** er de slutbrugere, der anvender softwaren og modtager output eller serviceydelser produceret af eller understøttet af softwaren.

For mere om testinteressenter, see kapitel 2 af [Goucher09].

Denne liste af interessenter er ikke komplet. Testmanagere skal finde de specifikke interessenter for deres projekt eller program. Testmanageren skal også forstå hvori relationen mellem interessenten og test består og hvordan testgruppen hjælper med at opfylde interessenters behov. Desuden skal testmanageren registrere de øvrige softwareudviklings-livscyklus-aktiviteter og arbejdsprodukter som gensidigt påvirker testen. Uden dette kan testprocessen ikke nå den optimale egnethed og effektivitet (se sektion 2.2.3).

2.2.2 Yderligere softwareudviklingslivscyklusaktiviteter og arbejdsprodukter

Da softwaretest er en vurdering af kvaliteten af en eller flere arbejdsprodukter fremstillet af andre aktiviteter end testaktiviteter, finder det normalt sted inden for rammerne af et større sæt af softwareudviklingslivscyklusaktiviteter. Testmanageren skal planlægge og lede testaktiviteter med en forståelse af, hvordan de andre aktiviteter og deres arbejdsprodukter påvirker test, som det blev diskuteret i pensum for Foundation-niveauet, og hvordan test påvirker disse andre aktiviteter og deres arbejdsprodukter.

I en organisation som anvender agile praksis udfører udviklere ofte testdrevet udvikling, laver automatiseret komponenttest og integrerer koden kontinuerlig (sammen med testene for den kode) i konfigurationsstyringssystemet. Testmanageren bør samarbejde med den udviklingsansvarlige for at sikre at testerne deltager i disse aktiviteter.

Testere kan foretage review af komponenttestene både for at bidrage med forslag til øget dækning og egnethed af disse test og for at få en dybere forståelse af softwaren og måden den er implementeret. Testere kan evaluere de måder som de kan integrere deres egne automatiserede tests, især funktionel regressionstest, i konfigurationsstyringssystemet. [Crispin09]

Mens det specifikke forhold mellem testaktiviteter, de andre testinteressenter, softwareudviklingslivscyklussens arbejdsaktiviteter og arbejdsprodukter varierer afhængigt af projektet, den valgte softwareudviklingslivscyklus og en række andre faktorer, er test nært forbundet og relateret til følgende:

- Kravudvikling og ledelse. Testmanageren skal tage hensyn til krav under afgrænsning og estimering af testindsats samt forblive opmærksom på ændringer til krav og udøve testkontrolhandlinger for at tilpasse test til disse ændringer. Tekniske testanalytikere og testanalytikere bør deltage i review af krav
- Projektledelse. Testmanageren må i samarbejde med testanalytikere og de tekniske testanalytikere give projektlederen krav vedrørende tidsplan og ressourcer. Testmanageren skal samarbejde med projektlederen for at forstå ændringer i projektplanen og udøver testkontrolhandlinger for at tilpasse testen til disse ændringer
- Konfigurationsstyring, frigivelsesstyring og ændringsstyring. Testmanageren må i samarbejde med testgruppen etablere processer og mekanismer for levering af testobjekter og beskrive dem i en testplan. Testmanageren kan få testanalytikeren og den tekniske testanalytiker til at verificere builds og til at sikre versionskontrol under testeksekvering
- Softwareudvikling og vedholdelse. Testmanageren bør samarbejde med udviklingsansvarlige for at koordinere leveringen af testobjekter inklusiv indhold og datoer for hver frigivelse samt deltage i fejlhåndtering (se kapitel 4)
- Teknisk support. Under testlukningen bør testmanageren samarbejde med den ansvarlige for teknisk support for at sikre overdragelse af testresultater så de, der skal supportere produktet efter frigivelse, er bekendt med kendte afvigelser og workarounds. Desuden bør testmanageren samarbejde med den ansvarlige for teknisk support for at analysere afvigelser i produktion og implementere forbedringer af testprocessen

- Produktion af teknisk dokumentation. Testmanageren bør samarbejde med den ansvarlige for den tekniske dokumentation for at levere testdokumentation i tide samt håndtere defekter fundet i disse dokumenter.

Udover at identificere testinteressenter som beskrevet ovenfor skal testmanageren også finde andre arbejdsprodukter og aktiviteter i softwarens livscyklus som gensidigt påvirker testen. Ellers vil testprocessen ikke nå den optimale egnethed og effektivitet.

2.2.3 Afstemning af testaktiviteter og andre livscyklusaktiviteter

Test bør også være en integreret del af projektet uanset hvilken softwareudviklingsmodel der anvendes:

- Sekventielle modeller som f.eks. vandfaldsmodel, V-model og W-model. I en sekventiel model afsluttes alle arbejdsprodukter og aktiviteter (f.eks. krav, design, implementering, komponenttest, integrationstest, systemtest og accepttest) for en bestemt fase før de næste fase påbegyndes. Testplanlægning, testanalyse, testdesign og testimplementering overlapper med projektplanlægning, forretnings/kravanalyse, software- og databasedesign og programmering. I hvilken grad tingene overlapper afhænger af testniveauet. Testeksekvering følger sekventielt rækkefølgen af testniveauer diskuteret i pensum for Foundation-niveauet og dette pensum
- Iterative og inkrementelle modeller som f.eks. Rapid Application Development (RAD) og Rational Unified Process (RUP). I iterative og inkrementelle modeller grupperes de funktioner, som skal udvikles, sammen (f.eks. i forhold til forretningsprioritet eller risiko) og derefter afvikles de forskellige projektfaser inklusiv arbejdsprodukter og aktiviteter for hver gruppe af funktioner. Faserne kan afvikles sekventielt eller kan overlappe og selve iterationerne kan foregå sekventielt eller kan overlappe. Under initiering af projektet sker overordnet testplaning og testanalyse parallelt med projektplanlægning og forretnings-/kravanalyse. Detaljeret testplanlægning, testanalyse, testdesign og test implementering sker i begyndelse af hver iteration og overlapper. Testeksekvering overlapper ofte testniveauer. Hvert testniveau begynder så tidligt som mulig og kan fortsætte efter højere testniveauer er startet
- Agil som f.eks. SCRUM og Extreme Programming (XP). Disse modeller er iterative livscyklusser hvor iterationerne er meget korte (ofte to til fire uger). Arbejdsprodukterne og aktiviteterne for hver iteration afsluttes før næste iteration starter (dvs. iterationerne er sekventielle). Test foregår ligesom i iterative modeller men med en højere grad af overlap mellem de forskellige testaktiviteter og udviklingsaktiviteter, bl.a. betydelig overlap mellem testeksekvering (på forskellige niveauer) og udviklingsaktiviteter. Alle aktiviteter i en iteration, inklusiv testaktiviteterne, bør afsluttes før næste iteration starter. I et agilt projekt ændres testmanagerens rolle sig typisk fra en direkte ledelsesmæssig rolle til en teknisk autoritet/rådgivende rolle
- Spiral. I spiralmodellen anvendes prototyper tidligt i projektet til at bekræfte gennemførligheden og for at eksperimentere med beslutninger vedrørende design og implementering. Rækkefølgen som prototypeeksperimenter foretages i bestemmes ud fra forretningsprioritet og teknisk risiko. Disse prototyper testes for at bestemme hvilke aspekter af tekniske problemer der endnu ikke er løst. Når de vigtigste problemer er løst fortsætter projektet efter en sekventiel eller iterativ model.

For at kunne tilpasse testaktiviteter inden for livscyklussen skal testmanageren have en detaljeret forståelse af de livscyklusmodeller, der anvendes i organisationen. For eksempel kunne den grundlæggende testproces fra ISTQB tilpasses på systemtestniveau i V-modellen således:

- Systemtest planlægningsaktiviteter forekomme samtidig med projektplanlægning, og test kontrol fortsætter indtil afviklingen af systemtesten og lukning er fuldført

- Systemtestanalyse- og -designaktiviteter sker samtidig med specificering af krav, (overordnet) design af system og arkitektur, og komponentdesign (lav-niveau)
- Systemtestimplementeringsaktiviteter kan starte i løbet af systemdesign, selvom hovedparten af disse aktiviteter typisk vil finde sted samtidig med kodning og komponenttest, og hvor arbejdet relateret til systemtestimplementeringsaktiviteter ofte strækker sig indtil få dage før starten af eksekveringen af systemtesten
- Systemtestafviklingsaktiviteterne begynder, når alle startkriterier for systemtesten er opfyldt (eller frafaldes), hvilket typisk betyder, at komponenttest og ofte også komponentintegrationstest i det mindste er fuldført. Afvikling af systemtest fortsætter indtil slutkriterierne for systemtesten er opfyldt
- Evaluering af slutkriterierne for systemtesten og rapportering af testresultaterne for systemtesten sker løbende igennem afviklingen af systemtesten, generelt med større hyppighed og betydning efterhånden som projektdeadlines rykker nærmere
- Lukningsaktiviteter for systemtesten finder sted efter slutkriterierne for systemtesten er opfyldt og afviklingen af systemtest erklæres for afsluttet, dog kan de nogle gange blive udskudt til efter accepttest og alle projektaktiviteter er afsluttet.

I en iterativ og inkrementel livscyklus udføres de samme opgaver men timingen og i hvilken udstrækning kan variere. For eksempel i stedet for at bygge hele testmiljøet i starten af projektet kan det være mere effektivt kun at bygge den del som behøves i den nuværende iteration. Jo længere frem der planlægges jo mere kan omfanget af den fundamentale testprocess udvides i enhver af de iterative eller inkrementelle livscyklus modeller

Udover planlægningsfaserne som sker i ethvert projekt kan testafvikling og rapportering også være influeret af den livscyklus som gruppen anvender. For eksempel kan det i en iterativ livscyklus være egnet at udarbejde fuldstændige rapporter og udføre erfaringsopsamling sessioner i slutningen af en iteration før den næste iteration starter. Ved at behandle hver iteration som et mini-projekt har gruppen mulighed for at korrigere og tilpasse ud fra de ting der skete i den sidste iteration. Da iterationer kan være korte og tidsbegrænsede kan det give mening at afkorte den tid og energi som bruges på rapportering og vurdering, men opgaverne bør blive udført så de generelle testfremskridt følges og for at finde eventuelle problemområder så hurtigt som muligt. Problemer med processer som opleves i en iteration can nemt påvirke og endda forekomme igen i den næste iteration hvis der ikke gøres noget for at rette op på sagen.

Generel information om hvordan test kan tilpasses andre livscyklusaktiviteter kan beskrives i teststrategien (se sektion 2.4.2). Testmanageren bør tilpasse hvert testniveau og hvilken som helst kombination af softwareudviklingslivscyklus og testproces til projektet i løbet af testplanlægning og/eller projektplanlægning.

Afhængig af organisationens, projektets og produktets behov kan yderligere testniveauer udover niveauerne defineret i pensum for Foundation-niveauet være nødvendige, som f.eks.:

- Hardware-software integrationstest
- Systemintegrationstest
- Test af interaktioner mellem egenskaber
- Kunde produkt integrationstest.

Hvert testniveau bør have følgende elementer klart defineret:

- Testformål med realistiske mål
- Testomfang og testelementer
- Testbasis sammen med måder som man kan måle dækningen af testbasis (dvs. sporbarhed)
- Start- og slutkriterier

- Testleverancer, bl.a. rapportering af resultater
- Anvendte testteknikker sammen med en måde som man kan opnå en passende dækning med disse teknikker
- Relevante målinger og metrikker i forhold til testformål, start- og slutkriterier og rapporterng af resultater (inklusiv måling af dækning)
- Testværktøjer som kan anvendes til bestemte testopgaver (hvor og hvornår)
- Ressourcer (f.eks. testmiljøer)
- Hvem der er ansvarlige i og uden for testgruppen
- Overholdelse af standarder for organisationen, lovkrav og andre standarder.

Som beskrevet senere i dette kapitel er den bedste praksis at definere disse elementer sammenhængende på tværs af alle testniveauer for at undgå spild og farlige mangler på tværs af forskellige niveauer af lignende test.

2.2.4 Styring af ikke-funktionel test

Manglende planlægning af den ikke-funktionelle test kan gøre at alvorlige problemer med kvaliteten i et system først opdages efter frigivelse. Dog er mange typer af ikke-funktionelle test dyre, så testmanageren skal vælge hvilke ikke-funktionelle test skal udføres baseret på risiko og begrænsninger. Desuden er der mange forskellige typer af ikke-funktionelle test, hvoraf nogle muligvis ikke er egnet for en givet applikation.

Da testmanageren ikke nødvendigvis har tilstrækkelig ekspertise til at håndtere alle de planlægningsmæssige hensyn, skal testmanageren uddelegere noget af testplanlægningsansvaret til de tekniske testanalytikere (og i nogle tilfælde testanalytikere) som er ansvarlig for de ikke-funktionelle testaktiviteter. Testmanageren kan bede analytikerne om at overveje følgende generelle faktorer:

- Interessenters krav
- Nødvendig værktøj
- Testmiljø
- Organisatoriske faktorer
- Datasikkerhed.

For yderligere oplysninger se kapitel 4 i pensum for Advanced Technical Test Analyst [ISTQB ATTA SYL].

En anden vigtig overvejelse for testmanageren er hvordan man kan integrere ikke-funktionelle test i softwareudviklingslivscyklussen. En almindelig fejltagelse er at vente, indtil alle funktionelle tests er afviklet før den ikke-funktionelle test startes, hvilket kan resultere i at kritiske, ikke-funktionelle defekter opdages sent. I stedet bør ikke-funktionelle test prioriteres og rangordnes efter risiko. Der findes som regel måder som ikke-funktionelle risici kan afbødes i løbet af de tidlige niveauer af test eller endda under udvikling. For eksempel kan brugervenlighedsreview af brugergrænseflade-prototyper under systemdesign være effektiv til at finde brugervenligheds-defekter. Sådanne defekter kan skabe betydelige tidsmæssige problemer, hvis de først opdages i slutningen af systemtesten.

I iterative livscyklusser kan hastigheden af forandringer og iterationer gøre det svært at fokusere på visse ikke-funktionelle test, der kræver at avancerede testrammer bygges. Testdesign- og implementeringsaktiviteter, der tager længere tid end de den afsatte tid i en enkelt iteration bør organiseres som arbejdsaktiviteter uden for iterationerne.

2.2.5 Styring af erfaringsbaseret test

Mens erfaringsbaseret test giver fordele ved effektivt at finde defekter, som andre testteknikker nogle gange ikke kan finde og ved at tjene som en kontrol af fuldstændigheden af disse teknikker, giver det også nogle udfordringer i forhold til teststyring. Testmanageren bør være opmærksom på udfordringerne såvel som fordelene ved de erfaringsbaserede teknikker, især udforskende test. Det er vanskeligt at bestemme den opnåede dækningen ved en sådan test på grund af den typiske letvægtsmæssige logning og den minimale forudgående forberedelse af test. Reproducerbarhed af testresultaterne kræver særlig ledelsesmæssig opmærksomhed, især når der er inddraget flere testere.

En måde at håndtere erfaringsbaseret test på, især udforskende test, er at opdele testarbejdet i små, 30 til 120 minutters perioder undertiden kaldet testsessioner. Denne tidsafgrænsning begrænser og fokuserer det arbejde, der skal gøres i løbet af en session og giver en grad af overvågning samt planlægning. Hver session dækker et testcharter, som testmanageren formidler skriftligt eller mundtligt til testeren. Testcharteret angiver testbetingelserne, der skal dækkes i testsession, hvilket yderligere bidrager til at fastholde fokus og forhindre overlapning, hvis flere folk foretager udforskende test samtidig.

En anden teknik til at styre erfaringsbaseret test er ved at integrere en sådan selvstyret og spontan test i mere traditionelle foruddesignede testsessioner. For eksempel kan der gives tilladelse (og afsættes tid) til at testere udforsker ud over de eksplicitte trin, input, og forventede resultater i deres prædefinerede test. Testere kan også tildeles sådanne selvstyrende testsessioner som en del af deres daglige test, før, under eller efter en dag hvor foruddefinerede tests afvikles. Hvis sådanne testsessioner finder defekter eller interessante områder for fremtidig test, kan de foruddefinerede tests opdateres.

I begyndelsen af den udforskende session konstaterer og udfører testeren de nødvendige opsætningsopgaver for at kunne afvikle testene. Under sessionen lærer testeren om hvordan applikationen som testes anvendelses, designer og udfører test ifølge den teknik, der anvendes, og hvad han har lært om programmet, undersøger eventuelle fejl, og nedfælder resultaterne af testen i en log. (Hvis det er nødvendigt at kunne repetere testene, bør testerne også logge testinput, handlinger og begivenheder.) Efter sessionen kan en debriefing finde sted, som sætter retningen for efterfølgende sessioner.

2.3 Risikobaseret test og andre strategier for prioritering og tildeling af ressourcer

En universel teststyringsudfordring er den rette udvælgelse, tildeling og prioritering af tests. Det vil sige ud af en næsten uendelig række af testbetingelser og kombinationer af betingelser der kan dækkes, skal testgruppen vælge en begrænset række betingelser, bestemme og tildele en passende indsats for at dække hver betingelse med testcases, og prioritere rækkefølgen af testcases på en måde der optimerer egnetheden og effektiviteten af testarbejdet som skal udføres. Identifikation og analyse af risici sammen med andre faktorer kan anvendes af testmanageren til at hjælpe med at løse dette problem, selvom mange interagerende begrænsninger og variabler kan kræve en kompromisløsning.

2.3.1 Risikobaseret test

Risiko er muligheden for et negativt eller uønsket resultat eller begivenhed. Risici eksisterer, når nogle problemer kan opstå, som vil mindske kunders, brugers, deltageres, eller interessenters opfattelse af produktets kvalitet eller projektets succes. Hvis den primære effekt af problemerne påvirker produktets

kvalitet, kaldes problemerne kvalitetsrisici, produktrisici, eller produktkvalitetsrisici. Hvis den primære effekt af problemerne påvirker projektets succes, kaldes problemerne projektrisici eller planlægningsrisici.

I risikobaseret test finder og vurderer man kvalitetsrisici i en risikoanalyse af produktkvaliteten sammen med interessenterne. Derefter designer, implementerer og udfører testgruppen test for at afbøde kvalitetsrisici. Kvalitet omfatter samtlige funktioner, adfærd, karakteristika og attributter, der påvirker kunders, brugeres og interessenters tilfredshed. Derfor er en kvalitetsrisiko en beskrivelse af en potentiel situation, hvor der kan findes kvalitetsproblemer i et produkt. Eksempler på kvalitetsrisici for et system er f.eks. forkerte beregninger i rapporter (en funktionel risiko med hensyn til med nøjagtighed), langsomt svar på brugerinput (en ikke-funktionel risiko med hensyn til effektivitet og svartid), og problemer med at forstå skærme og felter (en ikke-funktionel risiko med hensyn til brugervenlighed og forståelsesegnethed). Når testen afslører defekter, har test mindsket kvalitetsrisikoen ved at skabe bevidsthed om defekter og muligheder for at håndtere dem inden frigivelse. I den situation hvor testen ikke finder defekter, har den test reduceret kvalitetsrisikoen ved at vise, at systemet fungerer korrekt under de testede betingelser.

Risikobaseret test bruger produktkvalitetsrisici til at vælge testbetingelser, fordele testindsatsen for disse betingelser, og til at prioritere de resulterende testcases. Der findes en række teknikker til risikobaseret test som varierer betydeligt både hvad angår typen og mængden af indsamlet dokumentation og niveauet af formalitet som anvendes. Uanset om det er eksplicit eller implicit, har risikobaseret test til formål at anvende test til at nedsætte det generelle niveau af kvalitetsrisiko og især at mindske denne risiko til et acceptabelt niveau.

Risikobaseret test består af fire hovedaktiviteter:
- Risikoidentifikation
- Risikovurdering
- Risikoreduktion
- Risikostyring.

Disse aktiviteter overlapper hinanden. De følgende underafsnit vil diskutere hver af disse aktiviteter.

For at være mest egnet bør risikoidentifikation og -vurdering omfatte repræsentanter for alle grupper af projekt- og produktinteressenter, dog sker det at realiteterne i et projekt gør at nogle interessenter optræder som stedfortrædere for andre interessenter. For eksempel kan et lille udsnit af kunderne hjælpe med at finde defekter, der har stor betydning for deres brug af softwaren i softwareudvikling til massemarkedet. I dette tilfælde er gruppen af potentielle kunder hele den mulige kundebase. På grund af deres særlige ekspertise med hensyn til produktkvalitetsrisici og afvigelser, bør testere inddrages aktivt i risikoidentifikation og risikovurderingsprocessen.

2.3.1.1 Identifikation af risici

Interessenter kan finde risici gennem en eller flere af følgende teknikker:
- Ekspert interviews
- Uafhængige vurderinger
- Brug af risikoskabeloner
- Post-projektevaluering
- Risikoworkshops
- Brainstorming
- Checklister
- Brug af erfaringer fra tidligere.

Der er størst chance for at finde mange produktkvalitetsrisici ved at inddrage et bredt udsnit af interessenter i processen.

Risikoidentifikation skaber ofte biprodukter, dvs. finder problemer der ikke har med produktkvaliteten at gøre. Det kan f.eks. være generelle spørgsmål om produktet, problemer i produktet, problemer i projektet eller problemer i krav- og designspecifikationer. Projektrisici er ikke det primære fokus i risikobaseret test, men viser sig tit mens man leder efter kvalitetsrisici. Styring af projektrisiko er vigtig for al test, ikke kun for risikobaseret test. Projektrisici diskuteres nærmere i afsnit 2.4.

2.3.1.2 Vurdering af risici

Når risikoidentifikation er overstået, kan risikovurdering begynde. Risikovurdering er undersøgelse af de fundne risici. Risikovurdering indebærer specifikt kategorisering af hver risiko og fastsættelse af sandsynlighed og effekt ved hver risiko. Risikovurdering kan også omfatte vurdering eller tildeling af andre egenskaber for hver risiko, f.eks. hvem der er ejer af risikoen.

Kategoriseringen af risici består i at give hver risiko en passende kategori, f.eks. performance, pålidelighed, funktionalitet osv. Nogle organisationer bruger ISO 9126 standardens [ISO9126] (som bliver erstattet af ISO 25000 standarden [ISO25000]) kvalitetskarakteristika til kategorisering. Mange organisationer bruger andre metoder til kategorisering. Den checkliste, der benyttes til risikoidentifikation, bruges ofte også til at kategorisere risici. Der findes generelle lister for kvalitetsrisiko, og mange organisationer tilpasser disse checklister til egne behov. Når checklister anvendes som grundlag for risikoidentifikation sker kategoriseringen af risiko ofte som en del af identifikationen.

Bestemmelse af risikoniveauet involverer typisk for hver risiko en vurdering af sandsynligheden for at risikoen indtræffer og effekt hvis den skulle indtræffe. Sandsynligheden for at risikoen indtræffer er sandsynligheden for at det potentielle problem er til stede i det system, som skal testes. Med andre ord er sandsynligheden en vurdering af den tekniske risiko. Faktorer der påvirker sandsynligheden for produkt- og projektrisici:

- Kompleksitet af teknologi og projektorganisation
- Spørgsmål til personaleforhold og uddannelse for forretningsanalytikere, designere og programmører
- Konflikter i gruppen
- Kontraktmæssige problemer med leverandører
- Geografisk spredning af udviklingsorganisationen
- Gamle systemer versus nye tilgange
- Værktøjer og teknologi
- Dårlig organisatorisk eller teknisk ledelse
- Tid, ressourcer, budget eller pres fra ledelsen
- Mangel på kvalitetssikringsaktiviteter tidligere i processen
- Høj ændringshastighed
- Mange defekter tidligere i projektet
- Grænseflade- og integrationsproblemer.

Effekten hvis risikoen indtræffer er alvoren hvormed brugerne, kunderne eller andre interessenter påvirkes. Faktorer, der påvirker effekten af projekt- og produktrisici:

- Brugsfrekvensen af den berørte funktion
- Hvor kritisk funktionen er for at opnå et forretningsmål
- Skade på virksomhedens omdømme
- Tab af forretning
- Potentielle finansielle, miljømæssige eller sociale tab. Potentielt erstatningsansvar.
- Civilretslige eller strafferetslige sanktioner

- Tab af licens
- Mangel på fornuftige omgåelser
- Synligheden af afvigelser førende til negativ omtale
- Personsikkerhed.

Risikoniveauet kan enten vurderes kvantitativt eller kvalitativt. Hvis sandsynlighed og effekt kan fastslås kvantitativt, kan man gange de to værdier sammen for at beregne et kvantitativ risikoprioritetsnummer. Men typisk kan risikoniveauet kun fastslås kvalitativt. Det vil sige at man omtaler sandsynligheden som værende meget høj, høj, medium, lav eller meget lav, men man kan ikke udtrykke sandsynligheden som en procentsats med nogen reel præcision. På samme måde kan man tale om at effekten er meget høj, høj, medium, lav eller meget lav, men man kan ikke udtrykke effekten i finansielle termer på en fuldstændig eller præcis måde. Den kvalitative vurdering af risikoniveauer skal ikke ses som værende ringere end kvantitative metoder. Når kvantitative vurderinger af risikoniveauer bruges uhensigtsmæssigt, kan resultaterne faktisk vildlede interessenter omkring omfanget af deres opfattelse og håndtering af risici. Kvalitative vurderinger af risikoniveauer kombineres ofte ved at gange eller lægge sammen for at skabe en samlet risikoscore. Denne samlede risikoscore kan udtrykkes som et risikoprioritetsnummer, men bør fortolkes som en kvalitativ, relativ værdi på en ordinal skala.

Hertil kommer at ved medmindre risikoanalysen er baseret på omfattende og statistisk valide risikodata vil risikoanalysen baseres på interessenternes subjektive opfattelser af sandsynlighed og effekt. Projektledere, programmører, brugere, forretningsanalytikere, arkitekter og testere har typisk forskellige opfattelser og dermed muligvis forskellige meninger om risikoniveauet for hver risikoelement. Processen for risikoanalyse bør omfatte en eller anden måde at opnå konsensus eller, i værste fald, at fastsætte et risikoniveau som der er enighed om (ved ledelsens beslutning eller ved at tage gennemsnittet, medianen, eller mode-niveau for risikoelementet). Risikoniveauerne bør kontrolleres for en god fordeling på hele skalaen for at risikoværdierne giver relevant vejledning i form af testrækkefølge, prioritering og fordeling af indsats. Ellers kan de ikke anvendes som en rettesnor for de risiko-reducerende aktiviteter.

2.3.1.3 Risikoreduktion

Risikobaseret test starter med en analyse af kvalitetsrisiko (identificering og vurdering af produktkvalitetsrisici). Denne analyse er grundlaget for den overordnede hovedtestplan og de andre testplaner. Som anført i planen eller planerne designes, implementeres og afvikles test for at dække disse risici. Indsatsen ved at udvikle og gennemføre en test er proportional med risikoniveauet, hvilket betyder, at mere omhyggelige testteknikker (f.eks. parvis test) anvendes til større risici, mens mindre omhyggelige testteknikker (f.eks. ækvivalenspartitionering og tidsbegrænset, udforskende test) anvendes til mindre risici. Desuden er prioriteten for udvikling og afvikling af testen baseret på risikoniveauet. Nogle sikkerhedsrelaterede standarder (f.eks. FAA DO-178B/ED 12B, IEC 61508) foreskriver testteknikker og dækningsgrad baseret på risikoniveauet.
Derudover har risikoniveauet betydning for beslutninger vedrørende anvendelsen af review af projektarbejdsprodukter inklusiv testarbejdsprodukter, testernes grad af uafhængighed og erfaringsniveau samt omfanget af bekræftelsestest (gentest) og regressionstest.

I løbet af projektet bør testgruppen være bevidst om yderligere oplysninger, der ændrer sættet af kvalitetsrisici og/eller risikoniveauet ved kendte kvalitetsrisici. Der bør løbende foretages regelmæssige justering af kvalitetsrisikoanalysen, hvilket medfører tilpasninger af testene. Disse tilpasninger bør som minimum ske ved større projektmilepæle. Justeringer omfatter identificering af nye risici, revurdering af niveauet af eksisterende risici og vurdering af effektiviteten af de risiko-reducerende aktiviteter. For eksempel hvis en risikoidentifikations- og vurderingssession fandt sted på grundlag af kravspecifikationen under kravfasen, bør en revurdering af risici finde sted, når designspecifikationen er afsluttet. Et andet eksempel er, hvis der i løbet af testen af en komponent viser sig at være betydeligt flere end det forventede antal defekter, kan man konkludere, at

sandsynligheden for at der var defekter i dette område var højere end ventet, og man justere dermed sandsynligheden og det overordnede risikoniveau opad. Det kan betyde en øget test for denne komponent.

Produktkvalitetsrisici kan også reduceres før testafviklingen starter. For eksempel hvis det konstateres at der er problemer med kravene under identifikation af risici, kan projektgruppen gennemføre grundige reviews af kravspecifikationerne som en afbødende handling. Dette kan reducere risikoniveauet, hvilket kan resultere i at færre test er nødvendige for at afbøde den tilbageværende kvalitetsrisiko.

2.3.1.4 Risikostyring i livscyklussen

Ideelt set sker risikostyring gennem hele livscyklussen. Hvis en organisation har et testpolitikdokument og/eller teststrategidokument, så skal disse beskrive den generelle proces for hvordan produkt- og projektrisici styres i test, samt vise hvordan risikostyring er integreret i og påvirker alle faser af test.

I en moden organisation hvor projektgruppen er meget bevidst om risiko, finder risikostyring sted på mange niveauer, og ikke kun i test. Vigtige risici addresseres ikke kun tidligere i specifikke test, men også på tidligere testniveauer. Hvis performance f.eks. er en central kvalitetsrisiko, så kan performancetesten starte allerede under komponent- og integrationstesten. Modne organisationer nøjes ikke med blot at finde risici, men finder også kilderne til disse risici og konsekvensen af dem, hvis de indtræffer. For de defekter, der opstår, anvendes underliggende årsagsanalyse til bedre at forstå kilder til risiko og til at gennemføre procesforbedringer, der i første omgang forhindrer fejl. Reduktion af risici foretages igennem hele softwareudviklingslivscyklussen. Risikoanalysen omfatter alle informationer den overvejer relateret arbejdsaktiviteter, analyse af systemets adfærd, omkostningsbaseret risikovurdering, produktrisikoanalyse, slutbrugerrisikoanalyse, og ansvarsrisikoanalyse. Risikoanalyse gennemsyrer test og testgruppen deltager i og påvirker en risikoanalyse for hele programmet.

De fleste risikobaserede testmetoder omfatter også teknikker til at bestemme rækkefølgen af samt prioritere testen og sikrer dermed at de vigtigste områder dækkes tidligt og at de vigtigste defekter findes i løbet af testafviklingen. I nogle tilfælde afvikles alle høj-risiko test før alle lav-risiko test og testene afvikles i stringent rækkefølge (ofte kaldet "dybde-først"). I andre tilfælde anvendes en stikprøvetilgang til at uddrage test fra alle identificerede risikoelementer, så udvælgelsen vægter risiko, men hvert risikoelement dækkes mindst én gang (ofte kaldet "bredde-først").

Uanset om risikobaseret test følger dybde-først eller bredde-først er det muligt at den afsatte tid til test opbruges før alle tests er afviklet. Risikobaseret test muliggør at testere kan rapportere til ledelsen i forhold til tilbageværende risikoniveau på dette tidspunkt, hvilket giver ledelsen mulighed for at beslutte om testen forlænges eller den resterende risiko overføres til brugerne, kunderne, helpdesk/teknisk support og/eller driftspersonale.

Ved at anvende de mest sofistikerede risikobaserede testteknikker, som ikke behøver at være de mest formelle eller omfattende teknikker, i løbet af testafviklingen kan projektdeltagere, projekt- og produktledere, topledere, erfarne ledere og projektinteressenter overvåge og kontrollere softwareudviklingslivscyklussen, herunder foretage beslutning om frigivelse baseret på det tilbageværende risikoniveau. Det kræver at testmanageren rapporterer testresultater i forhold til risiko på en måde som hver enkelt testinteressent kan forstå.

2.3.2 Risikobaserede testteknikker

Der findes en række teknikker til risikobaseret test. Nogle af disse teknikker er yderst uformelle. Eksempler herpå er tilgange, hvor testeren analyserer kvalitetsrisici under udforskende test

[Whittaker09]. Dette kan hjælpe med at guide testen, men kan føre til en overdreven fokus på sandsynligheden for defekter, i stedet for deres effekt, og omfatter ejheller tværorganisatorisk input fra interessenter. Desuden er sådanne tilgange subjektive og afhængige af den enkelte testers færdigheder, erfaring og præferencer. Derfor opnår man sjældent det fulde udbytte af risikobaseret test med disse tilgange.

I et forsøg på at opnå fordelene ved risikobaseret test og samtidig minimere omkostningerne anvender mange fagfolk letvægtstilgange til risikobaseret test. Disse tilgange sammenfletter lydhørheden og fleksibilitet fra de uformelle tilgange med styrken og opbyggelsen af konsensus fra de mere formelle tilgange. Eksempler på letvægtstilgange omfatter Pragmatisk Risk Analysis and Management (PRAM) [Black09], Systematic Software Testing (SST) [Craig02], og Product Risk Management (PRISMA) [vanVeenendaal12]. Udover de sædvanlige egenskaber for risikobaseret test har disse teknikker typisk følgende attributter:

- Udviklet over tid baseret på erfaring med risikobaseret test i erhverslivet, især i brancher, hvor effektivitet er vigtig
- Baseret på omfattende involvering af en tværfaglig gruppe af interessenter, der både repræsenterer forretningsmæssige og tekniske perspektiver, i den indledende risikoidentifikation og vurdering
- Optimalt hvis de indføres i løbet af de tidligste faser af et projekt, hvor mulighederne for at afbøde kvalitetsrisici er størst, og hvor de vigtigste arbejdesprodukter og biprodukter af risikoanalysen kan bidrage til at påvirke specifikationen og implementeringen af produktet på en måde, der minimerer risiko
- Anvender det genererede output (risikomatrix eller risikoanalysetabel) som grundlag for testplanen og testbetingelserne og dermed alle de efterfølgende teststyrings- og analyseaktiviteter
- Understøtter rapportering af testresultater i form af tilbageværende risiko til alle niveauer af testinteressenter.

Nogle af disse teknikker (f.eks SST) er baseret på at have kravspecifikationer som et input til risikoanalysen og kan derfor kun bruges når der findes kravspecifikationer. Andre teknikker (f.eks PRisMa og PRAM) lægger op til at der anvendes en blanding af risiko-baseret og krav-baseret strategi som anvender krav og/eller andre specifikationer som et input til risikoanalysen, men som også kan fungere udelukkende baseret på input fra interessenterne. Anvendelsen af krav som input hjælper med at sikre en god dækning af kravene. Testmanageren må sikre at man ikke overser væsentlige risici der ikke umiddelbart kan udledes af kravene, især for ikke-funktionelle områder. Når der er gode, prioriterede krav til rådighed som input, så er der typisk også en stærk sammenhæng mellem risikoniveau og proriteringen af et krav.

Mange af disse teknikker ser også brugen af risikoidentificerings- og vurderingsprocessen som en måde at skabe enighed om testtilgangen blandt interessenter. Dette er en stor fordel, men kræver samtidig at interessenterne afsætter tid til at deltage i enten gruppebrainstorming eller individuelle interviews. Hvis interessenter ikke deltager tilstrækkeligt vil der være mangler i risikoanalysen. Selvfølgelig er interessenter ikke altid enige om niveauet af risiko. I så fald skal personen, som styrer arbejdet med kvalitetsrisikoanalysen, arbejde kreativt og positivt med interessenterne for at opnå så stor enighed som muligt. En person, som styrer analyse af kvalitetsrisiko, bør have de samme færdigheder som en uddannet moderator.

Ligesom mere formelle teknikker tillader letvægtsteknikker brug af vægtning af faktorer relateret til sandsynlighed og effekt for at understrege forretningsmæssige eller tekniske risici (f.eks. afhængigt af niveauet af test). Letvægts teknikker (i modsætning til mere formelle teknikker):

- bruger kun to faktorer: sandsynlighed og effekt
- bruger enkle, kvalitative vurderinger og måleenheder.

På grund af den letvægtsmæssige karakter af disse tilgange er de fleksible, kan anvendes på en række af forskellige områder, og er til at anvende for grupper med folk med forskellige niveauer af erfaring og færdigheder (selv med ikke-tekniske og uerfarne folk).

I den formelle og tunge ende af skalaen har testmanageren en række valgmuligheder:

- Faremomentanalyse, som udvider analysenprocessen til at omfatte forudgående aspekter for at finde de farer, der ligger til grund for hver enkelt risiko
- Omkostninger ved eksponering fastsættes ud fra en vurdering af risikoen for hver kvalitetsrisiko: 1) sandsynligheden for en afvigelse for risikoelementet, målt i procent, 2) det økonomiske tab ved at en typisk afvigelse sker i produktion, og 3) omkostningen ved at teste for sådanne afvigelser
- Failure Mode and Effect Analysis (FMEA) og dens varianter [Stamatis03] identificerer kvalitetsrisici, deres potentielle årsager, og deres sandsynlige effekt, hvorefter man tildeler en score for alvorsgrad, prioritet og sandsynlighed for at de opdages
- Quality Function Deployment (QFD), som er en teknik til styring af kvalitetsrisiko med implikationer for test, som specifikt er rettet mod kvalitetsrisici, der opstår på grund af en fejlagtig eller utilstrækkelig forståelse af kundernes eller brugernes behov
- Fault Tree Analysis (FTA), hvor forskellige, faktisk konstaterede afvigelser (fra test eller fra produktion) eller potentielle afvigelser (kvalitetsrisici) underkastes en analyse af underliggende årsager. Analysen starter med de defekter, der ville kunne være grund til afvigelsen, og fortsætter derefter med fejltagelser eller defekter der kunne være grund til disse defekter, indtil de forskellige underliggende årsager er fundet.

Hvilke specifikke teknikker der anvendes til risikobaseret test, og hvor formelle de er, afvejes i forhold til projekt, proces og produkt forhold. For eksempel en uformel tilgang, som f.eks. Whittakers udforskende teknik kan anvendes til en opgradering eller nødløsning. I agile projekter er analysen af kvalitetsrisiko fuldt integreret i den tidlige periode af hvert sprint og dokumentation af risici er en del af opfølgningen på user stories. Systemer af systemer kræver risikoanalyse af hvert system såvel som af det samlede system af systemer. Personsikkerhedskritiske og missionskritiske projekter kræver højere niveauer af formalitet og dokumentation.

Input, processer og output til risikobaseret test vil som regel blive bestemt ud fra den valgte teknik. Fælles input omfatter viden fra interessenter, specifikationer og historiske data. Fælles processer omfatter risikoidentifikation, risikovurdering og risikostyring. Almindelige output omfatter en liste over kvalitetsrisici (med tilhørende risikoniveau og den anbefalede fordeling af testindsats), konstaterede defekter i inputdokumenter såsom specifikationer, spørgsmål eller problemer relateret til risikoelementerne samt projektrisici, som påvirker testen eller projektet som helhed.

Normalt er den mest kritiske succesfaktor for risikobaseret test at den rigtige gruppe af interessenter inddrages i risikoidentifikation og -vurdering. Alle interessenter har hver især deres egen forståelse af, hvad produktkvalitet er og sæt af prioriteter og bekymringer vedrørende kvaliteten. Dog kan interessenter som oftest opdeles i to brede kategorier: forretningsinteressenter og tekniske interessenter.

Forretningsinteressenter omfatter blandt andet kunder, brugere, driftspersonale, helpdesk og teknisk support personale. Disse interessenter forstår kunden og brugeren og kan dermed finde risici og vurdere effekten ud fra et forretningsmæssigt synspunkt.

Tekniske interessenter omfatter blandt andet udviklere, arkitekter, databaseadministratorer og netværksadministratorer. Disse interessenter forstår de underliggende måder, software kan fejle på, og kan dermed finde risici og vurdere sandsynligheden ud fra et teknisk perspektiv.

Nogle interessenter har både et forretningsmæssigt og et teknisk perspektiv. For eksempel områdeeksperter med en rolle inden for test eller forretningsanalyse har et bredere syn på disse risici på grund af deres tekniske og forretningsmæssige ekspertise.

Det at lede efter risikoelementer kan skabe en betydelig liste af risici. Det er nødvendigt at interessenterne kan argumentere for at et risikoelement eksisterer. Så snart én interessent opfatter noget som en risiko for kvaliteten af systemet, er det et risikoelement. Dog er det vigtigt, at interessenterne opnår enighed om deres vurdering af risikoniveauet. For eksempel i letvægtstilgange, der bruger sandsynlighed og effekt som vurderingsfaktorer, må man finde en fælles måde at rangordne sandsynlighed og effekt på som en del af processen. Alle interessenter, herunder testgruppen, skal bruge den samme skala, og skal kunne enes om en enkelt sandsynlighed og effekt score for hver kvalitetsrisikoelement.

Testmanageren skal inddrage interessenterne i arbejdet med risikobaseret test, hvis den skal lykkes på langt sigt. Den tværgående gruppe skal se nytten af risikoanalyse for at sikre at teknikken bruges løbende. Derfor må testmanageren forstå interessenternes behov og forventninger og hvor meget tid de kan bruge på processen.

Det er en stor fordel for testmanageren hvis interessenterne er godt engageret i analyseprocessen af kvalitetsrisiko. Fordelen er, at interessenterne på projekter med uklare eller manglende krav stadig kan finde risici, hvis de blot guides af en ordentlig checkliste. Denne fordel ses efter implementering af risikobaseret test, når testgruppens effektivitet i at finde defekter forbedres. Dette er et resultat af, at der anvendes et mere fuldstændigt testgrundlag som i dette tilfælde er listen over kvalitetsrisikoelementer.

I risikobaseret test bør testgruppen i løbet af testlukningen måle, om den var i stand til at høste fordelene. I mange tilfælde indebærer dette at besvare nogle eller alle af følgende fire spørgsmål ved hjælp af metrikker og samtale med gruppen:

- Afslørede testgruppen en større procentdel af væsentlige defekter end af mindre væsentlige defekter?
- Fandt testgruppen de fleste vigtige defekter tidligt i testafviklingsperiode?
- Var testgruppen i stand til at forklare testresultaterne til interessenterne i forhold til risiko?
- Var de test som testgruppen sprang over (hvis nogen) associeret med et lavere niveau af risiko end de afviklede test?

I de fleste tilfælde giver en vellykket risikobaseret test et bekræftende svar på alle fire spørgsmål. På lang sigt bør testmanageren fastsætte procesforbedringsmål for disse metrikker og samtidig bestræbe sig på at forbedre effektiviteten af analyseprocessen af kvalitetsrisiko. Selvfølgelig kan andre metrikker og succeskriterier anvendes ved risikobaseret test, og testmanageren bør nøje overveje forholdet mellem disse metrikker, testgruppens strategiske mål og den adfærd, der vil opstå som følge af ledelse baseret på et bestemt sæt af metrikker og succeskriterier.

2.3.3 Andre teknikker til udvælgelse af test

Mens mange testmanagere har risikobaseret test som et element i deres teststrategi, er der mange som også omfatter andre teknikker.

En af de mest anvendte alternative teknikker til udvikling og prioritering af testbetingelser er kravbaseret test. Review af tvetydigheder identificerer og fjerner uklarheder i krav (som tjener som testgrundlag), ofte ved hjælp af en checkliste over normalt forekommende defekter i krav (se [Wiegers03]). Kravbaseret test kan anvende flere teknikker, f.eks. review af uklarheder, analyse af testbetingelser og graftegning af årsag og virkning.

Analysen af testbetingelser består i en nærlæsning af kravspecifikationen for at finde de testbetingelser som skal dækkes [Craig02]. Hvis kravene er prioriteret kan prioriteten bruges til at fordele indsats og prioritere testcases. Hvis der ikke er tildelt en prioritet, er det vanskeligt at bestemme en passende indsats og rækkefølgen af test uden at kombinere kravbaseret test med risikobaseret test.

Årsags-virkningsgraftegning drøftes i pensum for Advanced Test Analyst i forbindelse med dækning af kombinationer af testbetingelser som en del af testanalyse. Men årsags-virkningsgraftegning har en bredere anvendelse, idet den kan bruges til at reducere et meget stort problem testmæssigt til et overskueligt antal testcases og stadig give 100% funktionel dækning af testgrundlaget. Årsags-virkningsgraftegning kan også finde mangler i testgrundlaget under testcasedesign. Det kan afsløre defekter tidligt i softwarens udviklingslivscyklus, når testdesignet starter på grundlag af et udkast til kravene. Det er en kompleks opgave at tegne disse grafer, og det er en væsentlig hindring for at bruge årsags-virkningsgraftegninger. Der findes værktøjer som understøtter denne fremgangsmåde.

En generel hindring for kravbaseret test er, at kravspecifikationerne ofte er tvetydige, umulige at teste, ufuldstændige eller ikke-eksisterende. Det er Ikke alle organisationer, som er motiveret til at løse disse problemer, så testere konfronteret med sådanne situationer skal vælge en anden teststrategi.

En anden metode, der undertiden bruges til at øge brugen af de eksisterende krav er skabelsen af brugs- eller operationelle profiler. Det er en modelbaseret tilgang som anvender en blanding af usecases, brugere (nogle gange kaldet personaer), input og output til at gengive hvordan systemet anvendes i den virkelige verden. Det gør det muligt at teste funktionalitet, brugervenlighed, tværoperationalitet, pålidelighed, datasikkerhed og performance.

I løbet af testanalyse og -planlægning kan testgruppen finde brugerprofilerne og forsøger at dække dem med testcases. Brugerprofilen er en tilnærmning baseret på de tilgængelige oplysninger om hvordan softwaren realistisk set bruges. Det betyder, ligesom med risikobaseret test, at brugsprofiler ikke perfekt afbilleder den virkelige verden. Men hvis der er tilstrækkelig med information og input fra interessenter, så vil modellen være tilstrækkelig brugbar (se [Musa04]).

Nogle testmanagere anvender metodiske tilgange som f.eks. checklister til at bestemme hvad der skal testes, i hvor stort et omfang og i hvilken rækkefølge. For meget stabile produkter kan en liste over de vigtigste funktionelle og ikke-funktionelle områder, som skal testes, kombineret med et lager af eksisterende testcases være tilstrækkeligt. Checklisten angiver retningslinjer for fordelingen af indsats og rækkefølgen af test, som normalt baseres på typen og mængden af ændringer, der er lavet. Sådanne tilgange kan være mindre anvendelige, når de anvendes til test af mere end småændringer.

Sidst men ikke mindst er en anden almindelig metode at anvende en reaktiv tilgang. I en reaktiv tilgang udføres meget få testanalyse-, testdesign- eller testimplementeringsopgaver før testafviklingen. Testgruppen fokuserer på at reagere på det faktiske leverede produkt. Yderligere test fokuserer på de klynger af defekter, som opdages undervejs. Prioritering og fordeling foregår helt dynamisk. Den reaktive tilgang kan fungere som et supplement til andre tilgange, men når der udelukkende anvendes en reaktiv tilgang har man en tendens til at gå let henover store dele af applikationen, der er vigtige, men som ikke ser ud til at indeholde et stort antal fejl.

2.3.4 Prioritering af test og allokering af indsats i test processen

Testmanageren skal, uanset hvilken teknik eller endnu bedre mix af teknikker der anvendes, sørge for at teknikkerne inkorporeres i projektet og testprocesser. For eksempel i sekventielle livscyklusser (f.eks. V-modellen) udvælger testgruppen test, allokerer testindsats og prioriterer i første omgang

testene i løbet af kravfasen med efterfølgende justeringer, mens iterative eller agile livscyklusser kræver en iteration-for-iteration tilgang. Testplanlægning og kontrol skal overvejes og afstemmes i forhold til udviklingen i risici, krav og/eller brugsprofiler.

I testanalyse, -design og -implementering anvendes den tildeling af indsats og prioritering som blev bestemt i testplanlægningen. Et almindelig brud i testprocessen er, at man omhyggeligt analyserer og/eller modellerer, hvorefter man glemmer at anvende disse informationer til at styre testprocessen efterfølgende. Dette brud sker typisk under design og implementering.

I testafviklingen bør prioriteringen fastsat under testplanlægning følges, selvom det er vigtigt at opdatere prioriteringen periodisk baseret på oplysninger erhvervet efter planen oprindeligt blev skrevet. Ved evaluering og rapportering af testresultater og status i forhold til slutkriterier skal testmanageren også evaluere og rapportere i forhold til de risici, krav, brugsprofiler, checklister og andre vejledninger, der anvendes til at udvælge og prioritere tests. Hvis det er nødvendigt med yderligere testprioritering bør det ske med udgangspunkt i metoden for testprioritering.

Som en del af rapportering af resultater og slutkriterier kan testmanageren evaluere i hvilken grad testen er gennemført. Det bør omfatte en sporing mellem testcases og opdagede fejl og det relevante testgrundlag. For eksempel i risikobaseret test kan testerne undersøge den tilbageværende rest af risiko efterhånden som test afvikles og der findes defekter. Dette understøtter brugen af risikobaseret test til at fastsætte det rette øjeblik til at frigive. Testrapportering skal beskrive de dækkede risici og de risici som stadig er åbne samt realiserede fordele og fordele som endnu ikke er opnået. For at se et eksempel på indberetning af testresultater baseret på dækning af risiko se [Black03].

Sidst men ikke mindst bør testmanageren under testlukning evaluere metrikker og succeskriterier, som afspejler testinteressenternes behov og forventninger, herunder kundernes og brugernes behov og forventninger med hensyn til kvalitet. Testgruppen kan først hævde at den opfylder sit mål, når testen opfylder disse behov og forventninger.

2.4 Testdokumentation og andre arbejdsprodukter

Dokumentation fremstilles ofte som en del af teststyringsaktiviteter. Mens de specifikke navne og omfanget af teststyringsdokumenterne ofte varierer, er det følgende almindelige typer teststyringsdokumenter, der findes i organisationer og i projekter:

- Testpolitik: beskriver organisationens formål og mål med test
- Teststrategi: beskriver organisationens generelle testmetoder uanset projekt
- Hovedtestplan (eller projekttestplan): beskriver anvendelsen af teststrategien i et bestemt projekt
- Testniveauplan (eller fasetestplan): beskriver de bestemte aktiviteter, der skal udføres inden for hvert testniveau.

Måden som disse dokumenter rent fysisk er organiseret kan variere fra sammenhæng til sammenhæng. I nogle organisationer og på nogle projekter er de kombineret i et enkelt dokument. I andre kan de måske findes som adskilte dokumenter. I nogle tilfælde kan indholdet af disse dokumenttyper findes som intuitive, ubeskrevne eller traditionelle testmetoder. Større og mere formelle organisationer og projekter har typisk alle disse dokumenttyper som skriftlige arbejdsprodukter, mens mindre og mere uformelle organisationer og projekter typisk har færre af denne type skriftlige arbejdsprodukter. Dette pensum beskriver dokumenttyperne hver for sig. I praksis vil det være den organisatoriske og projektmæssige sammenhæng der bestemmer den korrekte brug af de forskellige typer.

2.4.1 Testpolitik

Testpolitikken beskriver hvorfor organisationen tester. Den angiver de overordnede formål for test, som organisationen ønsker at opnå. Denne politik bør være udarbejdet af organisationens erfarne folk inden for teststyring i samarbejde med erfarne ansvarlige for grupperne af testinteressenter.

I nogle tilfælde supplerer testpolitikken en bredere kvalitetspolitik eller er en del af den. Denne kvalitetspolitik beskriver de overordnede ledelsesværdier og -mål i relation til kvalitet.

Hvor der findes en skriftlig testpolitik, kan det være et kort høj-niveau-dokument, der:

- Opsummerer organisationens udbytte af test
- Giver en definition af test, f.eks. at den skal opbygge en tillid til, at systemet fungerer som tilsigtet, og reducere niveauet af kvalitetsrisiko (se afsnit 2.3.1)
- Beskriver hvordan det evalueres hvor effektivt og hvor egnet testen er til at opfylde formålene
- Skitserer en typisk testproces, som måske tager udspring i ISTQBs grundlæggende testproces
- Angiver hvordan organisationen vil forbedre sine testprocesser (se kapitel 5).

Testpolitikken kan omhandle testaktiviteter for både nyudvikling og vedligeholdelse. Den kan også referere til interne og/eller eksterne standarder for testarbejdsprodukter og testterminologi, der skal bruges i hele organisationen.

2.4.2 Teststrategi

Teststrategien beskriver organisationens generelle testmetoder. Det er måden som test anvendes til produkt- og projektrisikostyring, opdelingen af testen i niveauer og de høj-niveau aktiviteter der er knyttet til testen (den samme organisation kan have forskellige strategier for forskellige situationer som f.eks. forskellige livscyklusser for softwareudvikling, forskellige niveauer af risiko eller forskellige lovgivningsmæssige krav). Teststrategien og processen og aktiviteter, der er beskrevet i den, bør være i overensstemmelse med testpolitikken. Den skal angive generiske start- og slutkriterier for testen som gælder for organisationen eller for et eller flere programmer.

Som nævnt ovenfor beskriver teststrategier generelle testmetoder som typisk omfatter:

- Analytiske strategier, som f.eks. risikobaseret test, hvor testgruppen analyserer testgrundlaget for at finde testbetingelser, der skal dækkes. F.eks. i kravbaseret test udledes testbetingelser af kravene i testanalysen, derefter designes og implementeres test som dækker disse betingelser. Efterfølgende afvikles testene ofte i en rækkefølge som fastsættes ud fra prioriteten på de krav som testene dækker. Rapportering af testresultater sker i forhold til kravenes status, f.eks. krav som er testet og godkendt, krav som er testet men fejlet, krav som ikke er fuldtud testet, krav hvor testen er blokeret osv.
- Modelbaserede strategier, som f.eks. operationel profilering, hvor testgruppen laver en model (ud fra virkelige eller forventede situationer) af miljøet som systemet eksisterer i, de input og betingelser som systemet udsættes for og hvordan systemet bør fungere. F.eks. i modelbaseret performancetest af en hurtigt voksende mobilapplikation kunne man lave modeller af indgående og udgående netværkstraffik, aktive og inaktive brugere og belastning som resultat af processeringen baseret på nuværende brug og estimeret vækst over tid. Desuden kan man lave modeller som tager højde for hardware, software, datakapacitet, netværk og infrastruktur i det nuværende produktionsmiljø. Man kan også lave modeller for idéelle, forventede og minimum mængder af processeringskapacitet, svartider og ressourceallokering
- Metodebaseret strategier, som f.eks. kvalitetskarakteristikbaseret, hvor testgruppen anvender et sæt af forudbestemte testbetingelser såsom en kvalitetsstandard (f.eks. ISO 25000

[ISO2500] som erstatter ISO 9126 [ISO9126]), en checkliste eller samling af logiske testbetingelser som kan være relateret til et bestemt domæne, applikation eller testtype (f.eks. datasikkerhedstest) og disse testbetingelser anvendes fra en iteration til den næste eller fra en frigivelse til den næste. F.eks. i vedligeholdelsestest af en simpel og stabil e-handelswebsite kunne testere anvende en liste som finder nøglefunktioner, attributter og links for hver side. Hver gang der laves en ændring til websitet ville testerne så dække de relevante elementer på checklisten

- Proces-eller standardkompatible strategier, som f.eks. systemer anvendt i medicinalindustrien som skal overholde standarder fra de godkendende myndigheder (f.eks. US Food and Drug Administration i USA), hvor testgruppen følger et sæt processer defineret af en komité eller et udvalg som fastsætter standarder eller en anden gruppe af eksperter, som omhandler dokumentation, korrekt identifikation og brug af testgrundlag og et eller flere testorakeler, og organiseringen af testgruppen. I projekter som anvender teknikker fra Scrum Agile til at styre hver iteration analyserer testerne f.eks. user stories, der beskriver særlige funktioner. Testerne estimerer testindsatsen for hver enkelt funktion som en del af planlægningsprocessen for iterationen. De finder også testbetingelser (ofte kaldet godkendelseskriterier) for hver user story, de udfører test der dækker disse betingelser, og de rapporterer status for hver user story (ikke testet, fejlet, eller bestået) i løbet af testafviklingen
- Reaktive strategier, som f.eks. at anvende defekt-baseret angreb hvor testgruppen venter med at designe og implementere test indtil softwaren leveres, og dermed reagerer på det faktiske system som skal testes. For eksempel ved at anvende udforskende test til at teste en menu-baseret applikation hvor der laves et sæt af testcharters som svarer til funktionerne, valg af menuer og skærmbillederne. Hver tester tildeles et sæt af testcharters, som de så bruger til at strukturere deres sessioner af udforskende test. Testerne rapporterer periodisk resultaterne af testsessionerne til testmanageren, som på baggrund af resultaterne kan ændre på charterne
- Konsultative strategier, som f.eks. brugerstyret test, hvor testgruppen sætter sin lid til input fra en eller flere nøgleinteressenter til at fastsætte hvilke testbetingelser der skal dækkes. For eksempel ved outsourcet kompabilitetstest af en web-baseret applikation kan en virksomhed give serviceudbyderen, som der outsources til, en prioriteret liste over browser versioner, software til beskyttelse mod diverse ting, operativsystemer, typer af forbindelser og andre konfigurationsindstillinger som de ønsker evalueret i forhold til deres applikation. Testserviceudbyderen kan derefter anvende teknikker såsom parvis test (for områder af høj prioritet) og ekvivalens partitionering (for områder af lavere prioritet) til at finde testcases
- Regressionsteststrategier, som f.eks. omfattende automatisering, hvor test gruppen anvender forskellige teknikker til at håndtere risikoen for regression, især automatisering af funktionelle og/eller ikke-funktionelle test på en eller flere niveauer. For eksempel hvis en web-baseret applikation regressionstestes kan testeren anvende et GUI-baseret værktøj til testautomatisering til at automatisere usecases for de almindelige anvendelser af applikationen samt undtagelserne. Disse test afvikles hver eneste gang applikationen ændres.

Forskellige strategier kan kombineres. Den specifikt valgte strategi skal passe til organisationens behov og muligheder, og organisationerne kan skræddersy strategier, så de passer til bestemte funktioner og projekter.

Teststrategien kan beskrive de testniveauer, der skal udføres. I disse tilfælde skal den give en vejledning for startkriterierne og slutkriterierne på hvert niveau og relationerne mellem niveauerne (f.eks. opdelingen af målene for testdækning).

Teststrategien kan også beskrive:

- Integrationsprocedurer
- Testspecifikationsteknikker

- Uafhængighed af test (som kan variere afhængigt af niveauet)
- Obligatoriske og valgfrie standarder
- Testmiljøer
- Testautomatisering
- Mulighed for genbrug af softwarearbejdsprodukter og testarbejdsprodukter
- Gentest og regressionstest
- Testkontrol og -rapportering
- Testmåleresultater og -metrikker
- Fejlhåndtering
- Konfigurationsstyringstilgang for test-delprodukter
- Roller og ansvarsområder.

Der kan være nødvendigt at definere både kortsigtede og langsigtede teststrategier. Forskellige teststrategier passer til forskellige organisationer og projekter. En omfattende strategi kan være relevant når det drejer sig om sikkerhedskritiske eller datakritiske applikationer. Derudover varierer teststrategien også for de forskellige udviklingsmodeller.

2.4.3 Hovedtestplan

Hovedtestplanen beskriver alt testarbejde som skal udføres i et bestemt projekt, herunder de bestemte niveauer som skal indgå i testen, og relationerne mellem disse niveauer og mellem testniveauerne og de respektive udviklingsaktiviteter. Hovedtestplanen bør diskutere, hvordan testerne vil implementere teststrategien i projektet (dvs. tilgangen til test). Hovedtestplanen bør være konsistent med testpolitikken og -strategien. På specifikke områder, hvor den ikke er det, skal disse afvigelser og undtagelser forklares, herunder hvilken som helst effekt disse afvigelser kunne have. For eksempel hvis det er organisationens teststrategi at gennemføre en komplet runde af regressionstest på et uforandret system umiddelbart før frigivelse, men det nuværende projekt vil ikke gennemføre regressionstest, så bør testplanen forklare hvorfor det er planlagt således og hvad der vil blive gjort for at afbøde enhver risiko på grund af denne afvigelse fra den normale strategi. Testplanen bør også indeholde en forklaring af alle andre effekter som kan forventes på grund af denne afvigelse. For eksempel ved at springe over regressionstest kan det være nødvendigt at planlægge en vedligeholdelsesfrigivelse en måned efter projektets første frigivelse. Hovedtestplanen bør komplementere projektplanen eller driftsvejledningen på den måde, at den skal beskrive det testarbejde, der er en del af det større projekt eller driften.

Selv om det specifikke indhold og strukturen i hovedtestplanen varierer afhængigt af organisationen, dens dokumentationsstandarder og formaliteten i projektet, omfatter en hovedtestplan typisk følgende emner:

- Elementer, der skal testes og dem, der ikke skal testes.
- Kvalitetskarakteristika, der skal testes og dem, der ikke skal testes.
- Tidsplan og budget for testen (der bør være afstemt med projekt- eller driftsbudgettet).
- Testafviklingscyklusser og deres sammenhæng med softwarereleaseplanen.
- Relationer og leverancer mellem test og andre medarbejdere eller afdelinger.
- Definition af hvilke testelementer, der er omfattet og hvilke der ikke er omfattet af hvert af de beskrevne niveauer.
- Specifikke startkriterier, fortsættelseskriterier (afbrydelse/genoptagelse), og slutkriterier for hvert niveau og relationerne mellem niveauerne.
- Testprojektrisici.
- Overordnet styring af testindsatsen.
- Ansvar for at udføre hvert enkelt testniveau.
- Input og output fra hvert testniveau.

Ved mindre projekter eller driftsopgaver, hvor der kun er formaliseret ét testniveau, vil hovedtestplanen og testplanen for dette formaliserede niveau ofte være samlet i ét dokument. Hvis systemtest f.eks. er det eneste formaliserede niveau, mens uformel komponent- og integrationstest udføres af udviklere, og uformel accepttest udføres af kunderne som en del af en betatestproces, så kan systemtestplanen omfatte de elementer, der er omtalt i dette afsnit.

Desuden afhænger testen typisk af andre aktiviteter i projektet. Hvis disse aktiviteter ikke er tilstrækkeligt dokumenteret, specielt hvad angår deres indflydelse på og relationer til testen, kan emner med relation til disse aktiviteter være dækket i hovedtestplanen (eller i den pågældende niveautestplan). Hvis konfigurationsstyringsprocessen f.eks. ikke er dokumenteret, skal testplanen angive, hvordan testobjekter skal leveres til testgruppen.

2.4.4 Testniveauplan

Testniveauplaner beskriver de bestemte aktiviteter, der skal udføres inden for hvert testniveau eller i nogle tilfælde testtype. Testniveauplaner udvider hovedtestplanen hvor det er nødvendigt for det specifikke niveau eller den specifikke testtype, der dokumenteres. De giver detaljerede informationer om tidsplanlægning, opgaver og milepæle, der ikke nødvendigvis er dækket i hovedtestplanen. Hvis der desuden bruges forskellige standarder og skabeloner til specifikation af test på forskellige niveauer, vil disse detaljer være behandlet i testniveauplanen.

I mindre formelle projekter eller driftsopgaver er en enkelt testniveauplan ofte det eneste teststyringsdokument, der bliver skrevet. I sådanne situationer kan nogle af de informationselementer, der er nævnt tidligere i dette afsnit være dækket i dette testplansdokument.

For agile projekter kan sprint- eller iterationstestplaner træde i stedet for niveautestplaner.

2.4.5 Styring af projektrisici

En vigtig del af ordentlig planlægning omfatter håndtering af projektrisici. Man kan finde projektrisici ved at anvende samme processer som dem der er beskrevet i afsnit 2.3. Når de forskellige projektrisici er fundet, skal de håndteres af projektlederen. Det er ikke altid, at testgruppen har mulighed for at reducere sådanne risici. Nogle projektrisici bør håndteres af testmanageren, f.eks.:

- Parathed af testmiljø og værktøjer
- Rådighed over testpersonale samt deres kvalifikationer
- Mangel på standarder og regler for og teknikker til testarbejdet.

Metoderne til styring af projektrisici omfatter tidlig forberedelse af testdelprodukter, forudgående test af testudstyr, forudgående test af tidligere versioner af produktet, skrappere startkriterier for testen, krav til testbarhed, deltagelse i reviews af tidlige projektarbejdsprodukter, deltagelse i ændringsstyring og overvågning af projektfremdrift og -kvalitet.

Når en projektrisiko er fundet og analyseret, er der generelt fire muligheder for at styre den:

6. At mindske risikoen gennem forebyggende foranstaltninger som mindsker risikoens sandsynlighed og/eller effekt
7. At lægge planer for at mindske virkningerne, hvis risikoen bliver en realitet
8. At overlade risikoen til en anden part
9. At ignorere eller acceptere risikoen.

Valget afhænger af de fordele og muligheder samt de omkostninger og yderligere, potentielle risici, der hører sammen med de forskellige løsninger. Når der er udarbejdet en beredskabsplan for en

projektrisiko, skal man finde en ejer, og man skal beslutte hvornår og hvordan beredskabsplanen træder i kraft.

2.4.6 Andre arbejdsprodukter i test

Test indebærer udarbejdelsen af en række andre arbejdsprodukter som f.eks. fejlrapporter, testcase-specifikationer og testlogs. Testanalytikere og tekniske testanalytikere udarbejder de fleste af disse arbejdsprodukter. I pensum for Advanced Test Analyst diskuteres overvejelser i forbindelse med at udarbejde og dokumentere disse arbejdsprodukter. Testmanageren skal sikre at der er konsistens mellem disse arbejdsprodukter samt kvaliteten af dem ved hjælp af følgende aktiviteter:

- Etablering og overvågning af metrikker som viser kvaliteten af disse arbejdsprodukter, som f.eks. procentdelen af afviste fejlrapporter
- Samarbejde med testanalytikere og tekniske testanalytikere for at udvælge og tilpasse egnede skabeloner for disse arbejdsprodukter
- Samarbejde med testanalytikere og tekniske testanalytikere for etablere standarder for disse arbejdsprodukter, som f.eks. detaljeringsniveauet i test, logs og rapporter
- Anvendelse af egnede teknikker og relevante deltagere og interessenter i reviews af testarbejdsprodukter.

Omfanget, typen af og specifikke detaljer i testdokumentationen kan afhænge af forskellige overvejelser blandt andet den valgte softwareudviklingslivscyklus, gældende standarder og regulativer samt produktkvaliteten og projektrisici for systemet som udvikles.

Der findes forskellige kilder til skabeloner for testarbejdsprodukter som f.eks. IEEE 829 [IEEE829]. Det er vigtigt at testmanageren husker at dokumenterne i IEEE 829 er designet så de kan anvendes i enhver industri. Som sådan har skabelonerne et højt detaljeringsniveau, som i nogle tilfælde passer til og i andre tilfælde ikke passer til en bestemt organisation. Det er god praksis at tilpasse IEEE 829 dokumenterne, når der laves standard skabeloner, som skal anvendes i en bestemt organisation. Ved konsekvent at anvende skabeloner kan behovet for uddannelse reduceres og processer bliver mere ensartet på tværs af organisationen.

Som en del af test skal der laves rapporter om testresultater, som typisk udarbejdes af testmanageren og som beskrives senere i dette kapitel.

2.5 Testestimering

Estimering, som ledelsesaktivitet, er fremstillingen af et tilnærmet mål for de omkostninger og afslutningsdatoer, der er tilknyttet aktiviteterne i en bestemt driftsopgave eller et bestemt projekt. De bedste estimater:

- Repræsenterer den samlede viden fra erfarne praktikere og støttes af de involverede deltagere
- Giver specifikke, detaljerede oversigter over de omkostninger, ressourcer, opgaver og medarbejdere, der er involveret
- Præsenterer for hver estimeret aktivitet, de mest sandsynlige omkostninger, og den mest sandsynlige arbejdsindsats og varighed.

Estimering af software- og systemarbejde har længe været kendt for at være fyldt med problemer, både tekniske og politiske, selv om de bedste fremgangsmåder for estimering i forbindelse med projektstyring er veletablerede. Testestimering er anvendelsen af disse bedste fremgangsmåder på de testaktiviteter, der er knyttet til et projekt eller en driftsopgave.

Testestimering bør indeholde alle de aktiviteter, der er involveret i testprocessen som beskrevet i kapitel 1. De estimerede omkostninger, den estimerede arbejdsindsats og især varigheden af testafviklingen er ofte mest interessant for ledelsen, da testafviklingen typisk er på den kritiske vej i projektet. Men estimater for testafviklingen har en tendens til at være vanskelige at fremstille og til at være upålidelige, når den overordnede kvalitet af softwaren er lav eller ukendt. Desuden vil kendskab og erfaring med systemet højst sandsynligt påvirke kvaliteten af estimatet. En normal praksis er også at estimere antallet af nødvendige testcases, men det virker kun hvis man kan antage at der ikke vil være mange defekter i softwaren. Antagelser, der gøres under estimering, skal altid dokumenteres som en del af estimeringen.

Testestimeringen bør tage alle faktorer, der kan påvirke omkostninger, arbejdsindsats og varighed af testaktiviteterne, i betragtning. Disse faktorer omfatter (men er ikke begrænset til):

- Det krævede kvalitetsniveau for systemet
- Størrelsen af det system, der skal testes
- Historiske data fra tidligere testprojekter som kan udbygges med data fra industrien eller benchmark data fra andre organisationer
- Procesfaktorer, bl.a. teststrategi, udviklings- eller vedligeholdelseslivscyklus og procesmodenhed og nøjagtigheden af projektestimatet
- Materielle faktorer, herunder testautomatisering og -værktøjer, testmiljø, testdata, udviklingsmiljø(er), projektdokumentation (f.eks. krav, design osv.) og genbrugelige test-delprodukter
- Medarbejderfaktorer, herunder chefer og tekniske ledere, den overordnede ledelses forpligtelse og forventninger, evner, erfaring og holdninger i projektgruppen, stabiliteten i projektgruppen, projektgruppens relationer, støtte til test- og debugging-miljøer, domæneviden og adgang til uddannede kontraktansatte og konsulenter
- Kompleksiteten af processer, teknologi, organisation, antallet af testinteressenter, sammensætningen og den fysiske placering af undergrupper
- Betydeligt behov for opsætning, træning og orientering
- Kompleks timing af levering af komponenter, specielt ved integrationstest og testudvikling
- Skrøbelige testdata (f.eks. data som er tidsfølsomme).

Kvaliteten af den software, der er leveret til test er også en vigtig faktor, som testmanageren bør overveje når der estimeres. For eksempel hvis udviklerne anvender god praksis, som f.eks. automatiseret komponenttest og kontinuerlig integration så vil op til 50% af fejlene blive fjernet inden levering af koden til testgruppen (se [Jones11] for mere information om effektiviteten af disse metoder i at fjerne defekter). Nogle mener at agile metoder, f.eks. testdrevet udvikling, giver en højere kvalitet leveret til test.

Estimering kan enten foretages bottom-up eller top-down. De følgende teknikker kan bruges til testestimering enten hver for sig eller i kombination:

- Intuition, gæt og tidligere erfaring
- Nedbrydning af arbejdsopgaver (WBS, Work Breakdown Structure)
- Sessioner med teamestimering (f.eks. Wide Band Delphi)
- Virksomhedsstandarder og -normer
- Procentdele af den totale arbejdsindsats i projektet eller bemandingsniveauer (f.eks. forholdet mellem antallet af testere og antallet af udviklere)
- Organisatorisk historik og metrikker, bl.a. metrikbaserede modeller, der estimerer antallet af defekter, antallet af testcyklusser, antallet af testcases, den gennemsnitlige arbejdsindsats for hver test og antallet af involverede regressionscyklusser

- Branchegennemsnit og forudsigelsesmodeller, som f.eks. function-points, antal kodelinjer, estimeret arbejdsindsats for udvikling eller andre projektparametre (for eksempel se [Jones07]).

I de fleste tilfælde skal estimatet, når det er udarbejdet, sendes til ledelsen sammen med en begrundelse (se afsnit 2.7). Der følger ofte nogle forhandlinger, der giver en ny bearbejdning af estimatet. Ideelt set repræsenterer estimatet den bedst mulige balance mellem organisatoriske og projektmæssige mål inden for områderne kvalitet, tidsplanlægning, budget og funktioner.

Ethvert skøn bygger på de foreliggende oplysninger på det tidspunkt, det blev udarbejdet. Tidligt i et projekt er der kun adgang til begrænset information, og den kan ændres med tiden. Man bør opdatere sine estimater for at de kan afspejle nye og ændrede oplysninger.

2.6 Definition og brug af testmetrikker

En ledelseskliché siger, at det, der bliver målt, bliver gjort. Desuden siger den at hvad der ikke bliver målt ikke bliver gjort, fordi det der ikke bliver målt er nemt at ignorere. Det er derfor vigtigt, at et passende sæt af metrikker defineres for enhver bestræbelse, herunder test.

Testmetrikker kan klassificeres som tilhørende en eller flere af følgende kategorier:

- Projektmetrikker, der måler fremdrift mod vedtagende projektslutkriterier, som f.eks. procentdelen af testcases som er afviklet, bestået, og fejlet
- Produktmetrikker, der måler nogle af produktets attributter, som f.eks. om produktet er blevet testet eller tætheden af defekter
- Procesmetrikker, der måler test- eller udviklingsprocessens egnethed, som f.eks. procentdelen af defekter fundet ved at teste
- Personalemetrikker, der måler enkeltpersoners eller gruppers evner, f.eks. implementering af testcases inden for en givet tidsplan.

En metrik kan høre til to, tre eller endda fire kategorier. For eksempel kan et trenddiagram, der viser hvor mange defekter der findes dagligt, være relateret til et slutkriterie (ingen nye defekter fundet i en uge), kvaliteten af produktet (test kan ikke finde yderligere fejl i det), og egnetheden af testprocessen (test finder tidligt i testafviklingen perioden et stort antal defekter).

Personalemetrikker er særligt følsomme. Ledere forveksler sommetider metrikker, som primært er procesmetrikker for at være personalemetrikker, hvilket fører til katastrofale resultater, når folk bevidst handler på en måde for at påvirke målingerne, så resultatet i større grad bliver til deres fordel. Den rette motivation og vurdering af testpersonale diskuteres i kapitel 7 i dette pensum og i pensum for Expert Test Management [ISTQB ETL SYL].

På avanceret niveau fokuserer vi for det meste på brugen af metrikker til at måle fremdriften i testen, dvs. projektmetrikker. Nogle af projektmetrikkerne der anvendes til at måle fremdrift i testen er også relateret til produktet og processen. Yderligere information om ledelsesmæssig brug af produkt- og procesmetrikker findes i pensum for Expert Test Management. Yderligere oplysninger om brugen af procesmetrikker findes i pensum for Expert Improving the Test Process [ISTQB ITP SYL].

Brug af metrikker giver testere mulighed for at rapportere resultater på en ensartet måde og muliggør sammenhængende sporing af udviklingen over tid. Testmanagere skal ofte fremlægge metrikker på forskellige møder, hvor interessenter på forskellige niveauer, lige fra teknisk personale til topledelsen, kan deltage. Fordi metrikker sommetider anvendes til at bestemme om et projekt er en succes eller ej, bør man være meget omhyggelig, når det bestemmes, hvad der skal måles, hvor ofte det skal

rapporteres, og hvilken metode der skal bruges til at præsentere oplysningerne. En testmanager skal især overveje:

- Definition af metrikker. Der bør defineres et begrænset sæt af brugbare metrikker. Metrikker bør defineres ud fra en eller flere specifikke målsætninger for projektet, processen og/eller produktet. Metrikker bør defineres med henblik på at give en balance, da en enkelt metrik kan give et misvisende indtryk af status eller tendenser. Når disse metrikker er blevet defineret, er det vigtigt at alle interessenter er enige om måden som de skal fortolkes for at undgå uenighed når målingerne diskuteres. Der er ofte en tendens til at definere for mange metrikker i stedet for at begrænse sig til de mest relevante
- Opfølgning på metrikker. Rapportering og sammenlægning af metrikker skal være så automatiseret som muligt for at reducere tidsforbruget til at foretage og behandle målinger. Variationer af målinger over tid for en specifik metrik kan afspejle andre oplysninger end den fortolkning, som blev aftalt i definitionsfasen af metrikker. Testmanageren bør være parat til omhyggeligt at analysere målingers eventuel divergens fra det forventede resultat og årsagerne til denne divergens
- Rapportering af metrikker. Målet er at give en umiddelbar forståelse af informationerne til ledelsesformål. Præsentationer kan enten vise et øjebliksbillede af en metrik på et bestemt tidspunkt eller vise udviklingen i en metrik over tid, så tendenser kan evalueres
- Gyldigheden af målinger. En testmanager skal også kontrollere de oplysninger, der bliver rapporteret. Målingerne for en metrisk afspejler muligvis ikke den rigtige status af et projekt eller kan vise en alt for positiv eller negativ tendens. Før data præsenteres skal testmanageren checke det både for nøjagtighed og for det budskab, som det sandsynligvis vil give.

Der er fem primære dimensioner, som testfremdriften overvåges i forhold til:

- Produktrisici (kvalitetsrisici)
- Defekter
- Test
- Dækning
- Tillid.

Produktrisici, defekter, test og dækning kan måles, og bliver det ofte, og rapporteres på bestemte måder under projektet eller driftsopgaven. Hvis disse måleresultater er relateret til de definerede slutkriterier angivet i testplanen, kan de give en objektiv referenceramme til at bedømme, hvornår testindsatsen er færdig. Tillid, som kan måles gennem undersøgelser eller ved at anvende dækningsmetrikker som en erstatning, rapporteres ofte subjektivt.

Metrikker i forhold til produktrisici:

- Procentdelen af risici som er fuldstændig dækket af bestået test
- Procentdelen af risici hvoraf nogle eller alle tilhørende test fejler
- Procentdelen af risici som ikke er fuldstændig testet
- Procentdelen af dækkede risici sorteret efter risikokategori
- Procentdelen af risici som er identificeret efter den indledende analyse af kvalitetsrisici.

Metrikker i forhold til defekter:

- Det samlede antal fundne defekter i forhold til det samlede antal løste defekter
- Gennemsnitstiden mellem afvigelser (meantime between failure) eller afvigelseshyppighed
- Opdeling af antallet eller procentdelen af defekter kategoriseret efter følgende:
    - bestemte testelementer eller –komponenter
    - underliggende årsager
    - kilder til defekter (f.eks. kravspecifikation, ny funktion, regression osv.)

    - testfrigivelser
    - fase, hvor defekten blev indført, registreret eller fjernet
    - prioritet/alvorsgrad
    - afviste eller duplerede defektrapporter.
- Tendenser for forløbet tid fra rapportering til løsning af en defekt
- Antallet af rettelser af defekter som medfører nye defekter (nogle gange kaldet datter-defekter (daughter bugs)).

Metrikker i forhold til til test:

- Totalt antal test, der er planlagt, specificeret (udviklet), kørt, bestået, fejlet, blokeret og sprunget over
- Status for regressions- og gentest, inklusiv tendenser og totaler for regressionstest og gentest afvigelser
- Antal planlagte testtimer pr. dag i forhold til det faktisk opnåede antal testtimer
- Testmiljøets tilgængelighed (procentdelen af planlagte testtimer hvor testmiljøet kan benyttes af testgruppen).

Metrikker i forhold til testdækning:

- Dækning af krav- og designelementer
- Risikodækning
- Miljø-/konfigurationsdækning
- Dækning af kode.

Det er vigtigt, at testmanageren forstår at fortolke og anvende metrikker relateret til dækning med henblik på at forstå og rapportere teststatus. For højere testniveauer som f.eks. systemtest, systemintegrationstest og accepttest udgør det primære testgrundlag normalt arbejdsprodukter såsom kravspecifikationer, designspecifikationer, usecases, user stories, produktrisici, understøttede miljøer og konfigurationer. Metrikker baseret på strukturel kodedækning anvendes mere til lavere testniveauer som f.eks. komponenttest (f.eks instruktions- og forgreningsdækning) og komponentintegrationstest (f.eks grænsefladedækning). Testmanageren kan bruge metrikker relateret til kodedækning til at måle, om testene rammer strukturen i systemet under test. Dog bør rapportering af testresultater for højere testniveauer ikke omfatte metrikker relateret til kodedækning. Desuden skal testmanageren forstå, at selvom 100% af de strukturelle dækningsmål nås i komponenttest og komponentintegrationstest, er der stadig defekter og kvalitetsrisici, som skal afdækkes i de højere testniveauer.

Metrikker kan også relateres til aktiviteterne i den fundamentale testproces (beskrevet i pensum for Foundation-niveauet og i dette pensum). Ved at gøre dette kan metrikker bruges til at overvåge selve testprocessen samt fremdriften mod projektmålene igennem hele testprocessen.

Metrikker til at overvåge testplanlægnings- og kontrolaktiviteter:

- Risiko, krav eller dækning af andre elementer i testgrundlaget
- Opdagelse af defekter
- Planlagte versus faktiske timer brugt på at udvikle test-delprodukter og afvikle testcases.

Metrikker til at overvåge testanalyseaktiviteter:

- Antallet af fundne testbetingelser
- Antallet af fundne defekter i løbet af testanalyse-fasen (f.eks. som en del af identificering af risici eller andre testbetingelser ud fra testgrundlaget).

Metrikker til at overvåge testdesignaktiviteter:

- Procentdelen af testbetingelser dækket af testcases

- Antallet af fundne defekter i løbet af testdesign-fasen (f.eks. ved at lave testcases ud fra testgrundlaget).

Metrikker til at overvåge testimplementeringsaktiviteter:
- Procentdelen af konfigurerede testmiljøer
- Procentdelen af oprettede datarækker
- Procentdelen af automatiserede testcases.

Metrikker til at overvåge testafviklingsaktiviteter:
- Procentdelen af planlagte testcases som er afviklet, bestået og fejlet
- Procentdelen af testbetingelser som er dækket af afviklede (og/eller bestået) testcases
- Planlagte versus faktiske defekter rapporteret/rettet
- Planlagt versus faktisk opnået dækning.

Metrikker til at overvåge testfremdrift og afsluttende aktiviteter indbefatter at de relateres til milepæle, start- og slutkriterier (defineret og godkendt under testplanlægningen):
- Antallet af planlagte og afviklede testbetingelser, testcases eller testspecifikationer fordelt på om de er bestået eller fejlet
- Total antal defekter ofte fordelt på alvorsgraden, prioritet, nuværende tilstand, undersystem påvirket eller anden klassifikation (se kapitel 4)
- Antallet af krævede, accepterede, implementerede og testede ændringer
- Planlagte versus faktiske omkostninger
- Planlagt versus faktisk varighed
- Planlagte versus faktiske datoer for testmilepæle
- Planlagte versus faktiske datoer for testrelaterede milepæle (f.eks. frysning af koden)
- Status for produktrisiko (kvalitetsrisiko) ofte fordelt på afbødet versus ikke afbødet, store risikoområder, nye risici opdaget efter testanalyse osv.
- Procentdelen af tabt testindsats, omkostning eller tid på grund af blokerende hændelser eller planlagte ændringer
- Gentest- og regressionsteststatus.

Metrikker til at overvåge testlukningsaktiviteter:
- Procentdele af testcases som under testafviklingen er afviklet, bestået, fejlet, blokeret og sprunget over
- Procentdel af testcases som er lagt på lageret for genbrugelige testcases
- Procentdel af testcases som er automatiseret eller planlagte versus faktiske automatiserede testcases
- Procentdel af testcases som indgår i regressionstest
- Procentdel af defektrapporter som er løst/ikke løst
- Procentdel af fundne og implementerede test-delarbejdsprodukter.

Desuden anvendes standard projektledelsesteknikker som f.eks. nedbrydning af arbejdsopgaver (WBS, Work Breakdown Structure) ofte til at overvåge testprocessen. I agile grupper som anvender burndown diagram udgør test en del af fremdriften på en user story. Hvis man anvender Lean ledelsesteknikker overvåges testfremdriften for hver user story ofte ved at flytte user story kortet gennem en række kolonner på en Kanban-tavle.

Baseret på et defineret sæt af metrikker kan måleresultater rapporteres verbalt i prosaform, numerisk i tabeller eller visuelt i grafer. Måleresultaterne kan bruges til en række formål, herunder:
- Analyse, for at afdække hvilke tendenser og årsager der kan findes i testresultaterne

- Rapportering, for at kommunikere testresultaterne til de interesserede projektdeltagere og interessenter
- Kontrol, for at ændre retningen på testen eller projektet som helhed og for at overvåge resultaterne af denne kursændring.

Hvilke måder, der er bedst til indsamling, analyse og rapportering af disse testmåleresultater, afhænger af de specifikke informationsbehov, mål og muligheder for dem, der skal bruge disse måleresultater. Desuden bør indholdet af testrapporter varieres afhængig af målgruppen.

I forhold til testkontrol er det vigtigt, at metrikkerne igennem hele testprocessen (når testplanlægning er færdig) giver testmanageren de nødvendige oplysninger til at kunne lede testindsatsen til en vellykket gennemførelse af testmissionen, -strategierne og målene. Derfor skal disse informationsbehov overvejes under planlægningen og overvågning skal omfatte indsamling af enhver nødvendig metrik relateret til arbejdsprodukter. Mængden af nødvendig information og indsatsen til at indsamle den afhænger af forskellige projektfaktorer, bl.a. størrelse, kompleksitet og risiko.

Testkontrol skal reagere på oplysninger, der genereres af testen samt til projektets eller indsatsens skiftende betingelser. Hvis dynamisk test for eksempel afslører klynger af defekter i områder, som man ikke regnede med indeholdt mange defekter, eller hvis perioden for testafvikling afkortes på grund af en forsinket start af testen skal risikoanalysen og planen revideres. Dette kan give en omprioriteret test og en omfordeling af den resterende testafviklingsindsats.

Hvis der i rapporten over testfremdrift opdages afvigelser fra testplanen skal der udføres testkontrol. Testkontrol sigter mod at rette projektet og/eller testen i en mere vellykket retning. Når testresultaterne bruges til at påvirke eller måle kontrolindsatsen på projektet, skal følgende forhold overvejes:

- Revision af kvalitetsrisikoanalysen, testprioriteter og/eller testplaner
- Tilføjelse af ressourcer eller forøgelse af projekt- og testindsatsen på anden måde
- Udsættelse af frigivelsesdatoen
- Svækkelse eller forstærkning af testslutkriterierne
- Ændring af projektets omfang (funktionelt eller ikke-funktionelt).

Implementering af sådanne muligheder kræver typisk konsensus mellem projekt- eller driftsopgaveinteressenterne og samtykke fra projekt- eller driftscheferne.

Måden, en testrapport er sat op på, afhænger i høj grad af målgruppen, f.eks. projektledelsen eller den forretningsmæssige ledelse. For en projektleder er det sandsynligvis interessant at have detaljerede oplysninger om defekter. For den forretningsmæssige ansvarlige kunne et vigtigt rapporteringspunkt være status for produktrisici.

2.7 Den forretningsmæssige værdi af test

Testmanageren skal optimere testen så den giver størst mulig forretningsmæssig værdi. Det giver ikke god forretningsmæssig værdi at teste alt, for en altomfattende test vil forsinke leverancen urimeligt og vil koste mere end den sparer. Det giver heller ikke god forretningsmæssig værdi at teste for lidt, for så vil leverancen indehold alt for mange defekter. Den optimale mængde test ligger mellem disse to yderpunkter. Testmanageren skal hjælpe interessenterne til at forstå værdien af test og hvad der er den optimale mængde af test.

Selv om de fleste organisationer betragter test som værdifuld på en eller anden måde, er det kun få chefer, inklusive testmanagere, der kan kvantificere, beskrive eller formulere denne værdi. Desuden fokuserer mange testmanagere og testere på de taktiske detaljer ved test (aspekter, der er specifikke

for opgaven eller testniveauet), mens de ignorerer de mere strategiske (højere niveau) spørgsmål i forbindelse med test, som andre projektdeltagere, især chefer, bekymrer sig om.

Test giver værdi til organisationen, projektet og/eller driftsopgaven på både kvantitative og kvalitative måder:

- Kvantitative værdier omfatter afsløring af defekter, der således forhindres eller rettes før frigivelse, afsløring af defekter, der således er kendt før frigivelsen (dokumenteret, måske med instruktioner om hvordan de kan undgås, men ikke rettet), reduktion af risiko gennem kørsel af test og levering af oplysninger om projekt-, proces- og produktstatus
- Kvalitative værdier omfatter et forbedret ry for kvalitet, gnidningsløse og mere forudsigelige frigivelser, øget tillid, beskyttelse mod retsligt ansvar og reduktion af risikoen for tab af hele opgaver eller endog liv.

Testmanagere bør have forståelse for, hvilke af disse værdier, der gælder for deres organisation, projekt og/eller driftsopgave, og kunne kommunikere om testen ved hjælp af disse værdier.

En velkendt metode til måling af den kvantitative værdi og effektiviteten af test kaldes kvalitetsomkostninger (eller nogle gange omkostningen ved dårlig kvalitet). Kvalitetsomkostninger omfatter, at projekt- eller driftsomkostninger klassificeres i fire kategorier relateret til udgiften til produktdefekter:

- Omkostningerne ved forebyggelse, f.eks. uddannelse af udviklere i at skrive kode der er lettere at vedligeholde eller er mere sikker
- Omkostningerne ved registrering, f.eks. skrive testcases, konfigurere testmiljøer og reviewe krav
- Omkostningerne ved interne afvigelser, f.eks. rettelse af fejl fundet under test eller review før frigivelse
- Omkostningerne ved eksterne afvigelser, f.eks. udgifter til support i forbindelse med levering af fejlbehæftet software til kunderne.

En del af testbudgettet er omkostning til registrering (dvs. penge som bliver brugt selvom testerne ikke finder defekter, f.eks. udgifter til at udforme test). Den resterende del er omkostningen ved interne afvigelser (dvs. de faktiske udgifter i forbindelse med de fundne defekter). De totale omkostninger ved registrering og interne afvigelser er typisk langt under omkostningerne ved eksterne afvigelser, og det gør, at test giver udmærket værdi for pengene. Ved at bestemme omkostningerne i disse fire kategorier, er det muligt for testmanagere at fremstille en overbevisende businesscase for test.

Mere information om den forretningsmæssige værdi af test, inklusiv kvalitetsudgiften, kan findes i [Black03].

2.8 Distribueret, outsourcet og insourcet test

I mange tilfælde udføres noget eller måske alt testarbejdet af personer, som befinder sig forskellige steder, som er ansat i forskellige virksomheder eller som er adskilt fra resten af projektgruppen. Hvis testarbejdet sker flere steder, kan det kaldes distribueret. Hvis testarbejdet udføres et eller flere steder af medarbejdere, der ikke er kollegaer med resten af projektgruppen, og som ikke er placeret sammen med projektgruppen, kan testarbejdet kaldes outsourcet. Hvis testarbejdet udføres af medarbejdere, der er placeret sammen med projektgruppen, men som ikke er kollegaer med dem, kan testarbejdet kaldes insourcet.

Fælles for alle disse typer testarbejde er behovet for klare kommunikationskanaler og veldefinerede forventninger til missioner, opgaver og leverancer. Projektgruppen skal basere sig mindre på uformelle kommunikationskanaler som korridorsnak og kollegaer, der bruger social tid sammen. Det er vigtigt at

have aftalt hvordan kommunikationen skal foregå, f.eks. eskalering af problemer, hvilken type information og hvilke kommunikationsmetoder. Alle parter skal have en klar forståelse af deres rolle og ansvarsområder samt hvad de andres roller og ansvarsområder er for at undgå misforståelser og urealistiske forventninger. Placering, tidszone, kulturelle og sproglige forskelle øger sandsynligheden for at kommunikationsproblemer eller problemer relateret til forventninger opstår.

På tværs af alt testarbejdet er der også behovet for afstemning af metoder. Selvom uoverensstemmelser mellem metoder kan ske på ethvert projekt sker det oftere i situationer hvor arbejdet er distribueret og/eller udføres af eksterne folk. Hvis to testgrupper bruger forskellige metoder, eller testgruppen bruger en anden metode end udviklingsafdelingen eller projektledelsen, vil dette resultere i alvorlige problemer, specielt under testafviklingen. Hvis kunden for eksempel anvender en agil udvikling, mens testserviceudbyderen følger en pre-defineret testmetode som forudsætter en sekventiel livscyklus, så vil timing og måde som testelementer leveres til testserviceudbyderen være et hoveddiskussionspunkt.

Ved distribueret test skal opdelingen af testarbejdet på flere steder være klart beskrevet og ske efter grundige overvejelser. Uden denne vejledning foretager den mest kompetente gruppe måske ikke det testarbejde, de er kvalificeret til. Hvis de enkelte grupper ikke ved hvad de er ansvarlige for kan det ske at grupperne ikke laver det de skal. Forventningerne til hver enkelt gruppe skal kommunikeres klart og tydeligt. Uden omhyggelig styring vil testarbejdet som helhed lide under mangler (der forøger den resterende kvalitetsrisiko ved leveringen) og overlapninger (der reducerer effektiviteten).

Endelig er det ved alle den slags testarbejder af kritisk betydning, at hele projektgruppen udvikler og opretholder tillid til, at hver testgruppe udfører sin rolle ordentligt på trods af organisatoriske, kulturelle, sproglige og geografiske grænser. Mangel på tillid fører til ineffektivitet og forsinkelser i forbindelse med verifikationsaktiviteter, fordeling af skyld for problemer og organisatorisk politisering.

2.9 Anvendelse af industristandarder for softwaretest

Både i pensum for Foundation-niveauet og for de avancerede niveauer refereres der til en række standarder. Standarderne dækker livscyklusser for softwareudvikling, softwaretest, kvalitetskarakteristika for software, reviews og fejlhåndtering. Testmanagere skal være opmærksomme på standarder, virksomhedens politik for hvordan standarder skal anvendes og om standarder er nødvendige eller brugbare i situationen.

Standarder kan komme fra forskellige kilder:

- Internationale, eller med internationale mål
- Nationale, f.eks. nationale anvendelser af internationale standarder
- Domænespecifikke, f.eks. når internationale eller nationale standarder er tilpasset til bestemte domæner eller udviklet til specifikke domæner.

Internationale standard organisationer omfatter ISO og IEEE. ISO er den Internationale Standard Organisation, også kaldet IOS. Den består af medlemmer, som repræsenterer den nationale organisation i deres land som er mest relevant for det område som standardiseres. Denne internationale organisation har fremmet en lang række standarder som er nyttige for testere, som f.eks. ISO 9126 (som erstattes af ISO 25000), ISO 12207 [ISO12207] og ISO 15504 [ISO15504].

IEEE [www.ieee.org] er Institute of Electrical and Electronics Engineer, der er en faglig organisation med basis i USA. Der er nationale repræsentanter til rådighed i mere end 100 lande. Organisationen har foreslået et antal standarder, der er nyttige for softwaretestere, som f.eks. IEEE 829 [IEEE829] og IEEE 1028 [IEEE1028].

Mange lande har deres egne specifikke standarder. Nogle af disse standarder er nyttige for softwaretest. Et eksempel er den britiske standard BS 7925-2 [BS7925-2], der giver oplysninger om mange af de testteknikker, der er beskrevet i pensum for Advanced Test Analyst og for Advanced Technical Test Analyst.

Nogle standarder anvendes kun i et bestemt domæne, og nogle af disse standarder har betydning for softwaretest, softwarekvalitet og softwareudvikling. For eksempel inden for luftfart anvendes U.S. Federal Aviation Administration's standard DO-178B (og den europæiske pendant ED 12B) inden for luftfart til civile formål. Denne standard foreskriver kriterier for bestemte niveauer af strukturel dækning ud fra hvor kritisk software som skal testes er.

Et andet eksempel på en domænespecifik standard er U.S. Food and Drug Administration's Title 21 CFR Part 820 [FDA21] som anvendes for systemer i medicinalindustrien. Denne standard anbefaler visse strukturelle og funktionelle testteknikker. Standarden anbefaler desuden teststrategier og testprincipper som er i overensstemmelse med ISTQBs pensum.

I nogle tilfælde påvirkes test af standarder eller udbredte metoder, som ikke har test som primært fokusområde, men som i stor udstrækning påvirker den sammenhæng som test foregår i, nemlig softwareprocessen. De omfatter to nøgleprocesområder, verifikation og validering, som ofte ses som referencer til testniveauer (henholdsvis systemtest og accepttest). De påvirker også i forhold til teststrategi, hvor det ofte anses for at være nødvendigt at omfatte analytisk, krav-baseret test som en del af teststrategien.

Der findes tre andre vigtige eksempler som er PMIs PMBOK, PRINCE2®, og ITIL®. PMI og PRINCE2 er projektstyringsrammeværktøjer som ofte anvendes i henholdsvis Nordamerika og Europa. ITIL er et rammeværktøj der kan sikre at IT afdelingen leverer værdiskabende services til hele virksomheden. Både terminologien og aktiviteterne beskrevet i disse rammeværktøjer er signifikant forskellige fra ISTQBs pensum og begrebsliste. Når testmanageren arbejder i en virksomhed som anvender PMIs PMBOK, PRINCE2, og/eller ITIL skal testmanageren kunne forstå det rammeværktøj, måden det er implementeret på og dets terminologi så godt at han eller hun kan tilpasse sit arbejde til den kontekst.

Uanset hvilken standard eller metode som anvendes er det vigtigt at huske, at de er lavet af grupper af fagfolk. En standard er et produkt af den samlede erfaring og visdom i den gruppe, som laver standarden, men også af gruppens svagheder. Testmanagere bør være opmærksomme på standarder, som gælder for deres miljø og sammenhæng uanset om det er formelle standarder (internationale, nationale eller domænespecifikke) eller standarder og anbefalet praksis i den pågældende virksomhed.

Hvis det overvejes at anvende flere standarder er det vigtigt at holde sig for øje at nogle standarder ikke stemmer overens med andre og sommetider indeholder definitioner, som er direkte modstridende. Testmanageren bør bedømme om de forskellige standarder er brugbare i den sammenhæng, som testen foregår. Informationen i en standard kan være til nytte for et projekt eller den kan hindre projektet. Dog kan standarder fungere som en reference til afprøvet bedste praksis og give et udgangspunkt for at organisere testprocessen.

Nogle gange er det obligatorisk at følge standarder som påvirker testen. Det er op til testmanageren at være opmærksom på de standarder, der skal overholdes, og sikre, at der opretholdes en passende overholdelse.

3 Reviews – 180 min.

Nøgleord

Revision, uformelt review, inspektion, ledelsesreview, moderator, review, reviewplan, reviewer, teknisk review, walkthrough.

Læringsmål for reviews

3.2 Ledelsesreviews og revision

TM-3.2.1 (K2) Forstå de særlige kendetegn ved ledelsesreviews og revision.

3.3 Styring af reviews

TM-3.3.1 (K4) Analysér et projekt for at vælge den relevante reviewtype og for at fastlægge en plan for reviews der sikrer korrekt udførelse, opfølgning og placering af ansvaret.

TM-3.3.2 (K2) Forstå de faktorer og færdigheder og den nødvendige tid, der kræves for at deltage i reviews.

3.4 Metrikker til reviews

TM-3.4.1 (K3) Definér processer og produktmetrikker som skal anvendes i reviews.

3.5 Styring af formelle reviews

TM-3.5.1 (K2) Forklar ved hjælp af eksempler kendetegnene ved et formelt review.

3.1 Introduktion

Reviews blev introduceret i pensum for Foundation-niveauet som en statisk testaktivitet for produkter. Revision og ledelsesreview fokuserer mere på software processen end på software arbejdsprodukter.

Da reviews er en form for statisk test, er en testmanager ansvarlig for anvendelsen, specielt med hensyn til testprodukter. I den bredere sammenhæng af software projekter, bør dette ansvar dog være et spørgsmål om organisatorisk politik. I betragtning af den mulige udbredte anvendelse af formelle reviews på tværs af mange discipliner, både før og i software-projekter, kan den ansvarlige part være en testmanager, en kvalitetsansvarlig eller en uddannet reviewkoordinator. I dette pensum bliver den ansvarlige kaldt for reviewleder.

Reviewlederen skal sikre at der eksisterer et miljø som er befordrende for gennemførelsen af de succesfaktorer, som er defineret i pensum for Foundation-niveauet. Reviewlederen bør også sørge for en metode der kan måle den effektive værdi af reviews.

Fordi testere har en stærk forståelse af den operationelle adfærd og de krævede karakteristika af software systemet, er testinvolvering i reviews vigtig.

Deltagere i reviews må trænes i teknikken for at forstå deres roller i de forskellige typer af reviews. Alle reviewdeltagere må kunne indse fordelene ved et veludført review.

Når reviews gøres ordentligt, er de den største og mest omkostningseffektive bidragyder til den samlede leverede kvalitet. Det er således af afgørende betydning, at reviewledere er i stand til at gennemføre effektive reviews i deres projekter og demonstrere fordelene ved disse.

Mulige reviews indenfor et projekt:

- Kontraktreview ved projektets start og ved større milepæle
- Kravreview når kravene er klar til review - ideelt set både for funktionelle og non-funktionelle krav
- Designreview på højt niveau når det overordnede, arkitektoniske design kan reviewes
- Detaljerede designreview når det detaljerede design er klar til review
- Kodereviews af koden efterhånden som den bliver skrevet, evt. med komponenttest og dens resultater
- Testarbejdsprodukt-reviews, som dækker testplaner, testbetingelser, resultater af risikoanalyse, testcases, testdata, testmiljøer og testresultater
- Testindgangsreviews (test parathed) og testudgangsreviews for hvert test niveau, som henholdsvis checker test startkriterier før start af testafvikling og test slutkriterier før afslutning af testen
- Godkendelsesreviews, anvendes til at opnå kundens eller interessentens godkendelse af systemet.

Udover at anvende flere reviewtyper til et produkt, er det vigtigt for en reviewleder at huske, at mens reviews kan finde fejl i det statiske dokument, bør reviews suppleres med andre former for statisk test (f.eks. statisk analyse) og dynamisk test af koden. Ved at bruge en kombination af disse teknikker, forbedres testdækningen og vil finde flere fejl.

Forskellige teknikker har et forskelligt fokus. F.eks. kan et review eliminere et problem på kravniveau, før problemet bliver implementeret i koden. Statisk analyse kan hjælpe med at håndhæve kodestandarder og checke for problemer det kan være besværligt for teamet at finde ved en undersøgelse af arbejdsproduktet. Inspektioner kan ikke kun føre til opdagelse og fjernelse af

defekter, men kan også træne forfattere i, hvordan de undgår, at skabe defekter i deres arbejdsprodukter.

Pensum for Foundation-niveauet introducerede følgende typer af reviews:
- Uformelt review
- Walkthrough
- Teknisk review
- Inspektion.

Udover disse kan en testmanager også blive involveret i:
- Ledelsesreview
- Revision.

3.2 Ledelsesreviews og revision

Ledelsesreviews bruges til at overvåge fremdrift, vurdere status og træffe beslutninger om fremtidige tiltag. Disse reviews understøtter beslutninger om fremtiden for projektet, f.eks. at tilpasse omfanget af ressourcer, at gennemføre afhjælpende foranstaltninger eller ændre projektets omfang.

Følgende er de vigtigste kendetegn for ledelsesreviews:
- Udført af eller for den leder, der har det direkte ansvar for projektet eller systemet
- Udført af eller for en interessent eller beslutningstager, f.eks. en leder på højt niveau eller en direktør
- Kontrollere sammenhæng med og afvigelser fra planer
- Kontrollere tilstrækkeligheden af ledelsesprocedurer
- Vurdere projektets risici
- Evaluere effekten af handlinger og måder til at måle disse påvirkninger
- Producere lister med handlingspunkter, problemer der skal løses og beslutninger der er truffet.

Ledelsesreviews af processer som f.eks. projekttilbageblik (f.eks., erfaringer), er en integreret del af procesforbedringsaktiviteter.

Testmanagere bør deltage i ledelsesreviews på testfremdrift og kan tage initiativ til at starte sådanne reviews.

Revisioner udføres normalt for at demonstrere overensstemmelse med et defineret sæt af kriterier, sædvanligvis en gældende standard, lovgivningsmæssig begrænsning eller en kontraktlig forpligtelse. Som sådan har revisioner det formål at give en uafhængig evaluering af overholdelse af processer, bestemmelser, standarder osv.

Følgende er centrale kendetegn for revisioner:
- Ledet og styret af en ledende auditør
- Dokumentation for overholdelse indsamlet gennem interviews, udsagn og ved undersøgelse af dokumenter
- Dokumenterede resultater omfatter observationer, anbefalinger, korrigerende handlinger og en bestået/ikke bestået vurdering.

3.3 Styring af reviews

Reviews bør planlægges, således, at de finder sted på naturlige steder eller milepæle indenfor softwareprojektet. Typisk bør reviews afholdes efter krav og design definitioner med associerede

reviews, startende med forretningsmæssige mål og arbejde sig ned til det laveste niveau af designet. Ledelsesreviews bør finde sted ved store projektmilepæle, ofte som en del af en verifikationsaktivitet før, under og efter testafviklingen og andre væsentlige projektfaser. Reviewstrategien skal koordineres med testpolitikken og den overordnede teststrategi.

Før en overordnet reviewplan på projektniveau formuleres bør reviewlederen (f.eks. en testmanager) tage hensyn til:

- Hvad skal reviewes (produkt og proces)
- Hvem skal involveres i det specifikke review
- Hvilke relevante risikofaktorer skal dækkes.

Tidligt i projektplanlægningsfasen bør reviewlederen identificere det emne som skal reviewes og vælge den relevante reviewtype (uformelt review, walkthrough, teknisk review eller inspektion, eller et miks af to eller tre typer) og niveauet af formalitet. Dette er tidspunktet hvor yderligere træning i reviews kan anbefales. Derfra kan et budget (tid og ressourcer) for reviewprocessen tildeles. Vurderingen af budgettet bør omfatte en evaluering af risikoen og en beregning af investeringsafkastet (ROI – return of investment).

Investeringsafkastet til reviews er forskellen mellem omkostningerne ved at gennemføre reviews og omkostningerne ved at håndtere de samme defekter på et senere tidspunkt (eller miste dem helt), hvis reviewet ikke var blevet udført. Beregning af kvalitetsomkostninger kan som beskrevet i kapitel 2.7 bruges til at fastsætte dette beløb.

Beslutningen for det optimale tidspunkt at udføre reviews på, afhænger af:

- Tilgængelighed af de emner der skal reviewes i et tilstrækkeligt slutformat
- Tilgængelighed af de rette personer til reviewet
- Tidspunktet hvor den færdige version af emnet skal være tilgængeligt eller klar
- Den tid der kræves til reviewprocessen for det specifikke emne.

Reviewlederen bør fastlægge passende metrikker til at evaluere reviews under testplanlægningen. Hvis der er anvendt inspektioner, så bør korte inspektioner gennemføres på forfatterens anmodning, når dokumentfragmenter er afsluttet (f.eks. individuelle krav eller områder).

Formålet med reviewprocessen skal defineres under testplanlægningen. Dette omfatter udførelse af effektive og solide reviews, og opnå overensstemmende beslutninger om tilbagemeldinger på reviews.

Projektreviews bliver ofte holdt for det samlede system og kan også være nødvendige for sub-systemer eller de enkelte softwareelementer. Antallet af reviews, typen af reviews, organiseringen af reviews, og personerne som er involveret er alle afhængige af projektets størrelse og kompleksitet, og af produktrisici.

For at være effektive, skal deltagerne i reviews have et passende niveau af viden om teknik og procedure. Det giver effektive reviews, hvis deltagerne er grundige og har sans for detaljer, og det er også værd at lægge mærke til klarhed og korrekte prioriteringer i reviewkommentarerne. Der kan være et behov for uddannelse og metodisk viden for at reviewerne forstår deres rolle og ansvar i processen.

Reviewplanlægning bør forholde sig til risici ved tekniske faktorer, organisatoriske faktorer og menneskelige spørgsmål når de udføres. Adgang til reviewere med tilstrækkelig teknisk viden er kritisk for et vellykket review. Alle teams i et projekt bør involveres i planlægning af reviews, hvilket skulle sikre at hvert team er forpligtet til succes med reviewprocessen. Planlægning skal sikre at enhver organisation afsætter tilstrækkelig tid til at de krævede reviewere kan forberede og deltage i reviewet på relevante steder i projektets tidsplan. Tiden bør også planlægges for eventuel nødvendig teknisk

eller procesmæssig træning for reviewere. Backup reviewere skal identificeres i tilfælde af, at nøgle reviewere skulle blive utilgængelige på grund af ændringer i personale- eller forretningsplaner.

Under selve udførelsen af formelle reviews, skal reviewlederen sikre at:

- Tilstrækkelige metrikker for reviews er givet af deltagerne for at tillade evaluering af reviewets effektivitet
- Checklister er oprettet, og vedligeholdt for at forbedre fremtidige reviews
- Defekt alvorsgrad og prioritetsevaluering er defineret til brug i fejlhåndtering af emner fundet under review (se kapitel 4).

Efter hvert review skal reviewlederen:

- Samle reviewmålinger og sørge for at konstaterede problemer og emner er løst i tilstrækkelig grad til at kunne opfylde den specifikke målsætning med reviewet
- Bruge reviewmetrikker til at bestemme investeringsafkastet (ROI) for reviews
- Give feedback til de relevante interessenter
- Give feedback til reviewdeltagerne.

For at vurdere effektiviteten af reviews, kan testmanagere sammenligne faktiske resultater fundet i efterfølgende test (dvs. efter reviewene) med resultater fra reviewrapporter. I det tilfælde hvor et arbejdsprodukt er reviewet, godkendt baseret på reviewet, men senere fundet fejlfyldt, skal reviewlederen overveje, hvorledes reviewprocessen har kunnet lade defekterne passere. Sandsynlige årsager omfatter problemer med reviewprocessen (f.eks. dårlige start-/slutkriterier) forkert sammensætning af reviewteamet, utilstrækkelige reviewværktøjer (checklister, m.v.), utilstrækkelig træning og erfaring i reviews, for lidt forberedelse og for kort tid til selve reviewet.

Et mønster af undslupne defekter (specielt store defekter), der gentages på tværs af flere projekter viser, at der er betydelige problemer med afviklingen af reviews. I en sådan situation er der behov for at reviewlederen gennemgår processen og træffer passende foranstaltninger. Af forskellige grunde er det også muligt at reviews kan miste deres effektivitet over tid. En sådan effekt vil blive afsløret i et projekttilbageblik ved reduceret effektivitet i defektafsløring i reviewene. Igen må reviewlederen her undersøge og rette årsagerne. Under alle omstændigheder bør reviewmetrikker ikke bruges til at straffe eller belønne de enkelte reviewere eller forfattere men bør fokusere på selve reviewprocessen.

3.4 Metrikker til reviews

Reviewledere (der som nævnt ovenfor kan være testmanagere) må sikre at metrikker er tilgængelige for at:

- Evaluere kvaliteten af det reviewede emne
- Evaluere omkostningerne ved at foretage reviewet
- Evaluere den nedadgående kurve for defekter ved at have foretaget reviewet.

Reviewledere kan bruge metrikkerne til at bestemme investeringsafkastet og effektiviteten af reviews. Disse metrikker kan bruges til rapportering og procesforbedringsaktiviteter.

For hvert reviewet produkt kan følgende metrikker måles og rapporteres til produktevaluering:

- Arbejdsproduktets størrelse (sider, linjer af kode osv.)
- Forberedelsestid (inden reviewet)
- Tiden brugt til at gennemføre reviewet
- Tid brugt til at rette defekter
- Varighed af reviewprocessen
- Antallet af defekter fundet og deres alvorsgrad

- Identifikation af defektklynger i arbejdsproduktet (dvs. områder der har en højere defekttæthed)
- Type af review (uformelt review, walkthrough, teknisk review eller inspektion)
- Gennemsnitlig defekt tæthed (f.eks. fejl pr. side eller pr. tusinde linjer kode)
- Estimerede resterende defekter (eller resterende defekt tæthed).

For hvert review kan følgende metrikker måles og rapporteres for procesevaluering:

- Defektafsløringseffektivitet (under hensyntagen til defekter senere i livscyklussen)
- Forbedring af reviewprocesindsatsen og -timing
- Den procentvise dækning af planlagte arbejdsprodukter
- Typer af fundne defekter og deres alvorsgrad
- Deltagerundersøgelser om effektiviteten af reviewprocessen
- Omkostninger af kvalitetsmålinger for reviewdefekter versus dynamiske test og produktionsfejl
- Forbindelse af revieweffektivitet (review type versus defekt påvisnings effektivitet)
- Antal reviewere
- Defekter fundet pr. forbrugt arbejdstime
- Anslået sparet arbejdstid i projektet
- Gennemsnitlig defektindsats (dvs., samlet antal afsløringer og fiks tid divideret med antallet af defekter).

Desuden er de metrikker der er nævnt for produktevaluering ovenfor, også nyttige i procesevaluering.

3.5 Styring af formelle reviews

Pensum for Foundation-niveauet beskriver de forskellig faser i et formelt review: planlægning, kick-off, individuel forberedelse, reviewmøde, omarbejde og opfølgning. Reviewlederen er nødt til at sikre at alle trin i reviewprocessen bliver fulgt for at kunne gennemføre formelle reviews korrekt.

For formelle reviews gælder:

- Definerede start- og slutkriterier
- Checklister som skal anvendes af reviewere
- Leverancer som f.eks. rapporter, evalueringsskemaer og sammendrag af reviews
- Metrikker til måling af et reviews effektivitet og fremdrift.

Før starten af et formelt review må reviewlederen sikre sig at forudsætningerne er opfyldt som defineret i procedurer og startkriterier.

Hvis forudsætningerne for et formelt review ikke er opfyldt, kan reviewlederen foreslå reviewmyndigheden en af følgende beslutninger:

- Omdefinering af reviewet med reviderede mål
- Korrigerende handlinger før reviewet kan fortsætte
- Udskydelse af reviewet.

Som en del af at kontrollere et formelt review, er disse reviews overvåget i forbindelse med det samlede (højere niveau) program og er forbundet med projektets kvalitetssikringsaktiviteter. Kontrol af formelle reviews omfatter feedback information ved brug af produkt og proces metrikker.

4 Fejlhåndtering – 150 min.

Nøgleord

Uregelmæssighed, defekt (fejl), fejlprioriteringsudvalg, afvigelse, falsk-negativ resultat, falsk-positiv resultat, faseinddæmning, prioritet, underliggende årsag, alvorsgrad.

Læringsmål for fejlhåndtering

4.2 Fejllivscyklus og softwareudviklingslivscyklus

TM-4.2.1 (K3) Udarbejd en fejlhåndteringsproces for en testorganisation, der omfatter arbejdsgangen for fejlregistrering. Processen skal kunne overvåge og kontrollere et projekts fejl gennem hele testens livscyklus.

TM-4.2.2 (K2) Forklar hvilken proces og hvilke deltagere der er nødvendige for at opnå en effektiv fejlhåndtering.

4.3 Fejlrapportinformation

TM-4.3.1 (K3) Beskriv den data- og klassifikationsinformation der skal indsamles i løbet af fejlhåndteringsprocessen.

4.4 Vurdering af procesevne ud fra fejlrapportinformation

TM-4.4.1 (K2) Forklar hvordan statistik fra fejlrapporter kan bruges til at vurdere testprocesserne og softwareudviklingsprocesserne.

4.1 Introduktion

En organisations fejlhåndteringsproces og det værktøj der anvendes til at styre dette arbejde, er af kritisk betydning, ikke kun for testgruppen men for alle teams som er involveret i softwareudvikling. De informationer, der indsamles af en effektiv fejlhåndteringsproces, gør det muligt for testmanager og andre projektinteressenter at få indsigt i projektets tilstand gennem hele udviklingens livscyklus og ved at indsamle og analysere data over tid, er det muligt at lokalisere potentielle forbedringsområder i test- og udviklingsprocessen.

Udover at forstå den overordnede fejllivscyklus og hvordan den bruges til at overvåge, styre og kontrollere både test- og softwareudviklingsprocesser, må en testmanager også være bekendt med hvilke data der er kritiske at indsamle, og må være fortaler for korrekt brug af både processen og det valgte fejlhåndteringsværktøj.

4.2 Fejllivscyklus og softwareudviklingslivscyklus

Som beskrevet i pensum for Foundation-niveauet bliver fejl introduceret når en person begår en fejltagelse under udviklingen af et arbejdsprodukt. Dette arbejdsprodukt kan være en kravspecifikation, en user story, et teknisk dokument, en testcase, kode eller ethvert andet arbejdsprodukt, som er produceret i løbet af softwarens udvikling eller vedligeholdelse.

Fejl kan opstå på ethvert tidspunkt i softwareudviklingens livscyklus og i ethvert software-relateret arbejdsprodukt. Derfor skal der i alle faser af softwarens udviklingscyklus være aktiviteter, som kan opdage og fjerne potentielle defekter. For eksempel kan statiske testteknikker (dvs. reviews og statisk analyse) bruges på designspecifikationer, kravspecifikationer og kode før disse arbejdsprodukter leveres til efterfølgende aktiviteter. Jo tidligere fejl bliver opdaget og fjernet, desto billigere er det at sikre systemets kvalitet. Det er billigst at finde og rette fejl i den fase hvor de opstår (dvs. når softwareprocessen opnår perfekt faseinddæmning). Yderligere finder statisk test som beskrevet i pensum for Foundation-niveauet, fejlene direkte i stedet for at finde afvigelser. Når der ikke skal bruges debugging-aktiviteter til at lokalisere fejlen, vil det blive billigere at fjerne den.

Under dynamisk test som komponenttest, integrationstest og systemtest afsløres en fejl når den frembringer en afvigelse, som resulterer i en forskel mellem testens aktuelle og forventede resultat (dvs. en uregelmæssighed). I nogle tilfælde opdager testeren ikke uregelmæssigheden, og der fås et falsk-negativt resultat. Hvis testeren opdager uregelmæssigheden, kræver situationen yderligere undersøgelser. Denne undersøgelse begynder med udfyldelse af en fejlrapport.

I Test Driven Development (TDD) kan en automatiseret komponenttest bruges som en form for eksekverbar designspecifikation. Testen kan afvikles straks, efterhånden som koden udvikles. Indtil en komponent er færdig, vil nogle eller alle tests fejle. Derfor udgør en afvigelse i en sådan test ikke en fejl og typisk registreres den ikke.

4.2.1 Fejlarbejdsgang og tilstande

De fleste testorganisationer bruger et værktøj til at administrere fejlrapporter gennem fejlens livscyklus. En fejlrapport udvikler sig typisk gennem en arbejdsgang og bevæger sig gennem en række tilstande på sin vej gennem fejlens livscyklus. I de fleste af disse tilstande ejer en enkelt deltager rapporten og er ansvarlig for at udføre en opgave før rapporten flytter til den næste tilstand (og overføres til den næste ansvarlige part). I sluttilstandene har fejlrapporten ingen ejer, fordi ingen handlinger er påkrævet. Sådanne sluttilstande kan være:

- Lukket: den underliggende fejl er rettet og verificeret gennem en bekræftelsestest
- Annulleret: fejlrapporten er ugyldig

- Kan ikke reproduceres: uregelmæssigheden kan ikke længere observeres
- Udskudt: uregelmæssigheden relaterer sig til en reel fejl, men vil ikke blive rettet i projektets levetid.

For fejl, der er opdaget af testere under testen, er der tre tilstande hvor fejlen tilhører testgruppen:

- Den oprindelige tilstand
  - I denne tilstand, indsamler en eller flere testere de nødvendige oplysninger til dén, som er ansvarlig for at rette fejlen. Se i afsnit 4.3 hvad der skal tages med i en fejlrapport
  - Denne tilstand kan også kaldes "åben" eller "ny".
- Den returnerede tilstand
  - I denne tilstand, har modtageren af rapporten afvist den eller ønsker yderligere oplysninger. Denne tilstand kan vise en mangel i den indledende informationsindsamling eller i selve testen og testmanageren bør overvåge antallet af returnerede rapporter. Testeren skal sørge for supplerende oplysninger eller må bekræfte at rapporten skal afvises
  - Denne tilstand kan også kaldes "afvist" eller "præcisering".
- Bekræftelsestesttilstand
  - I denne tilstand, vil testeren afvikle en bekræftelsestest (ofte følge trinene for at reproducere afvigelsen fra selve fejlrapporten) for at afgøre, om rettelsen har løst problemet. Hvis bekræftelsestesten viser at fejlen er rettet, skal testeren lukke rapporten. Hvis bekræftelsestesten viser at fejlen ikke er rettet, skal testeren genåbne rapporten og overføre den til den tidligere ejer, som derefter kan udføre det nødvendige arbejde for at rette fejlen
  - Denne tilstand kan også kaldes "rettet" eller "verifikation".

4.2.2 Styring af ugyldige fejlrapporter og dubletter

I nogle tilfælde er en uregelmæssighed ikke et symptom på en fejl, men skyldes et problem med testmiljøet, testdata, andre elementer af test-delprodukter eller testerens egen misforståelse. Hvis testeren opretter en fejlrapport som efterfølgende viser sig ikke at relatere sig til en fejl i det testede produkt, er det et falsk-positivt resultat. Sådanne rapporter bliver typisk annulleret eller lukket som ugyldige fejlrapporter. I nogle tilfælde kan en fejl vise forskellige symptomer, som testeren opfatter som uafhængige. Så vil testeren typisk oprette to fejlrapporter. Hvis to eller flere fejlrapporter viser sig at skyldes samme underliggende årsag, bliver en af rapporterne typisk bevaret mens de øvrige lukkes som dubletter.

Ugyldige fejlrapporter og dubletter repræsenterer et vist niveau af ineffektivitet, men det er uundgåeligt at testere registrerer en vis mængde, og det må testmanageren acceptere. Hvis testmanageren forsøger at fjerne alle ugyldige fejlrapporter og dubletter, vil antallet af falsk-negative resultater typisk stige fordi testerne er blevet afskrækket fra at lave fejlrapporter. Det reducerer testorganisationens evne til at finde fejl, hvilket i det fleste tilfælde er relateret til et af hovedmålene med testorganisationen.

4.2.3 Tværfaglig fejlhåndtering

Selvom testorganisationen og testmanageren typisk ejer den overordnede fejlhåndteringsproces og fejlstyringsværktøjet, er et tværfagligt team generelt ansvarlig for, at forvalte de rapporterede fejl og mangler i et givet projekt. Udover testmanageren deltager typisk repræsentanter fra udvikling, projektledelse, produktledelse (kunde/forretning) og andre interessenter i fejlstyringsudvalget.

Når uregelmæssigheder opdages og indføres i fejlstyringsværktøjet, skal fejlstyringsudvalget mødes for at afgøre, om hver enkelt fejlrapport repræsenterer en gyldig fejl og om den skal rettes eller udskydes. Denne beslutning kræver at fejlstyringsudvalget overvejer fordele, risici og omkostninger ved at rette eller ikke rette fejlen. Hvis fejlen skal rettes, skal teamet prioritere rettelsen i forhold til andre projektopgaver. Udvalget bør konsultere testmanageren og testgruppen for relevante oplysninger så som hvor vigtig en fejl er i forhold til andre fejl.

Et fejlstyringsværktøj bør ikke bruges som en erstatning for god kommunikation ej heller skal fejlstyringsudvalgets møder bruges som en erstatning for effektiv brug af et godt fejlstyringsværktøj. Kommunikation, passende værktøjssupport, en veldefineret fejllivscyklus, og et engageret fejlstyringsudvalg er alle nødvendige for en effektiv fejlhåndtering.

4.3 Fejlrapportinformation

Når en fejl opdages (som en del af statisk test) eller en afvigelse observeres (som en del af dynamisk test), bør data indsamles af de involverede personer og indgå i fejlrapporten. Disse oplysninger bør være tilstrækkelige til tre formål:

- Håndtering af rapporten gennem hele fejlens livscyklus
- Vurdering af projektets status, specielt for produktkvalitet og testfremdrift
- Vurdering af procesevne (som omtalt i kapitel 4.4 nedenfor).

De data, der er nødvendige for fejlstyringsudvalget og projektets status kan variere afhængigt af, hvornår fejlen er opdaget i livscyklussen, typisk med færre nødvendige oplysninger tidligere (f.eks. krav, reviews og komponenttest). Dog bør kernen af indsamlede oplysninger være konsekvent i hele livscyklussen og ideelt på tværs af alle projekter for at muliggøre en meningsfuld sammenligning af fejldata i hele projektets levetid og på tværs af alle projekter.

Indsamling af data om fejl kan hjælpe ved styring og kontrol af testen og ved evaluering af slutkriterierne. Data om fejl bør f.eks. kunne vise fejltæthed, trend for fundne og løste defekter, gennemsnitlig tid fra opdagelse til rettelse, og bør kunne vise intensitet af fejl (f.eks. MTBF-analyse).

Fejldata, der skal indsamles kan omfatte:

- Navnet på den person som opdagede fejlen
- Rollen på den person, som har fundet fejlen (f.eks. slutbruger, forretningsanalytiker, udvikler, teknisk support person)
- Testtype, der er udført (f.eks. brugervenlighedstest, performancetest, regressionstest)
- Et resumé af problemet
- En detaljeret beskrivelse af problemet
- Trin til at reproducere afvigelsen (for en fejl), sammen med det aktuelle og det forventede resultat (uregelmæssigheden fremhæves), inkl. skærmdumps, databasedumps, og eventuelle logs hvis det er relevant
- Hvilken fase i livscyklus fejlen er introduceret, opdaget, og fjernet inkl. testniveauet, hvis det er relevant
- Det arbejdsprodukt, hvor fejlen blev introduceret
- Alvorsgraden af effekten på systemet og/eller produktets interessenter (sædvanligvis bestemt af systemets tekniske opførsel)
- Prioritering af at få rettet fejlen (sædvanligvis bestemt af afvigelsens effekt på forretningen)
- Det delsystem eller den komponent hvor fejlen findes (til analyse af fejlklynger)
- Den projektaktivitet, der foregik da problemet blev opdaget
- Identifikationsmetode som afslørede problemet (f.eks., review, statisk analyse, dynamisk test, brug i produktion)

- Fejltype (svarer sædvanligvis til fejltaksonomien, hvis en sådan anvendes)
- Kvalitetskarakteristikker som påvirkes af fejlen
- Det testmiljø hvor fejlen blev opdaget i (for dynamisk test)
- Projektet og produktet hvor problemet eksisterer
- Den nuværende ejer, f.eks. den person der er tildelt arbejdet med problemet, forudsat at rapporten ikke er i en sluttilstand
- Rapportens aktuelle tilstand (normalt styret af fejlstyringsværktøjet som en del af livscyklussen)
- De specifikke arbejdsprodukter (f.eks. testelementer og deres releasenumre) hvor problemet blev observeret, sammen med det specifikke arbejdsprodukt hvor problemet i sidste ende blev løst
- Effekten på projekt- og produktinteressenternes interesser
- Konklusioner, anbefalinger og godkendelser af de foranstaltninger, der er taget/ikke taget for at løse problemet
- Risici, omkostninger, muligheder og fordele ved rettelse/ikke rettelse af fejlen
- Datoerne for, hvornår de forskellige overgange i fejllivscyklus forekom, ejerne af rapporten ved hver overgang, og de handlinger der er truffet af projektteamets medlemmer for at isolere, reparere og verificere fejlrettelsen
- En beskrivelse af, hvordan fejlen i sidste ende blev rettet, og anbefalinger til test af rettelsen (hvis fejlen blev løst ved en ændring i softwaren)
- Andre referencer, som den test der afslørede fejlen og risikoen, kravet eller andre elementer fra testgrundlaget, som relaterer til fejlen (for dynamisk test).

Forskellige standarder og dokumenter: ISO 9126 [ISO9126] (bliver erstattet af ISO 25000), IEEE 829 [IEEE829], IEEE 1044 [IEEE1044] og Ortogonal Defekt Klassificering eksisterer for at hjælpe testmanageren med at afgøre, hvilke oplysninger der skal indsamles til fejlrapportering.

Uanset hvilke specifikke informationer, der er nødvendige i fejlrapporten, er det vigtigt, at testerne indtaster oplysninger, der er korrekte, kortfattede, præcise, objektive, relevante og rettidige. Selv om manuel indgriben og ansigt-til-ansigt kommunikation kan overkomme problemer i data i en enkelt fejlrapport kan problemer i fejlrapportdata give uovervindelige forhindringer i forhold til retvisende vurdering af projektets status, testfremdrift og procesevner.

4.4 Vurdering af procesevne med fejlrapporteringsinformationer

Som omtalt i kapitel 2, kan fejlrapporter være nyttige ved overvågning af projektets status og rapportering. Selv om de procesmæssige konsekvenser af metrikker primært behandles i pensum for Expert Test Management [ISTQB ETM SYL], bør en testmanager på avanceret niveau være opmærksom på, hvad fejlrapporter betyder for vurderingen af test- og softwareudviklingsprocesser.

Ud over at give mulighed for overvågning af testfremdrift som beskrevet i kapitel 2 og afsnit 4.3 skal fejlinformationer også understøtte procesforbedringsinitiativer, som for eksempel:

- Brug af information om introduktions-, opdagelses- og fjernelsesfasen til at vurdere faseinddæmning og foreslå måder hvorpå effektiviteten i fejlfindingen i de enkelte faser kan forbedres
- Brug informationer om introduktionsfasen til en Pareto-analyse af de faser hvor det største antal fejl er introduceret. Det kan målrette forbedringer, der reducerer fejlantallet
- Brug information om fejlens rodårsag til at fastslå de underliggende årsager til introduktion af fejlen. Det kan muliggøre procesforbedringer, der reducerer det samlede antal defekter
- Brug information om introduktions-, opdagelses- og fjernelsesfasen til at udføre en kvalitetsomkostningsanalyse med det mål at minimere omkostningerne forbundet med fejl

- Udfør en analyse af fejlklynger ved hjælp af fejloplysninger om den enkelte komponent til bedre at forstå de tekniske risici (for risikobaseret test) og for at forbedre processen for problematiske komponenter.

Brugen af metrikker til at vurdere hvor effektiv testprocessen er, bliver diskuteret i pensum for Expert Test Management [ISTQB ETM SYL].

I nogle tilfælde kan et team fravælge at spore fejl, der er fundet i dele af eller hele softwareudviklingens livscyklus. Det sker ofte i effektivitetens navn og for at reducere omkostningerne. I virkeligheden gør det procesevnerne for test og softwareudvikling mindre synlige og på grund af manglende pålidelige data bliver det vanskeligt at gennemføre forbedringer som foreslået ovenfor.

5 Forbedring af testprocessen – 135 min.

Nøgleord
Capability Maturity Model Integration (CMMI), Critical Testing Processes (CTP), Systematic Test and Evaluation Process (STEP), Test Maturity Model integration (TMMi), TPI Next

Læringsmål for forbedring af testprocessen

5.2 Processen for testforbedring
TM-5.2.1 (K2) Forklar ved hjælp af eksempler hvorfor det er vigtigt at forbedre testprocessen.

5.3 Forbedring af testprocessen
TM-5.3.1 (K3) Definér en plan for forbedring af testprocessen som følger IDEAL modellen.

5.4 Forbedring af testprocessen med TMMi
TM-5.4.1 (K2) Opsummér baggrund, omfang og formål for TMMi-modellen til forbedring af testprocessen.

5.5 Forbedring af testprocessen med TPI Next
TM-5.5.1 (K2) Opsummér baggrund, omfang og formål for TPI Next-modellen til forbedring af testprocessen.

5.6 Forbedring af testprocessen med CTP
TM-5.6.1 (K2) Opsummér baggrund, omfang og formål for CTP-modellen til forbedring af testprocessen.

5.7 Forbedring af testprocessen med STEP
TM-5.7.1 (K2) Opsummér baggrund, omfang og formål for STEP-modellen til forbedring af testprocessen.

5.1 Introduktion

Når en testproces først er etableret, skal den undergå kontinuerlig forbedring. i dette kapitel dækkes først generiske forbedringsemner, derefter følger en introduktion til nogle specifikke modeller, der kan bruges til forbedring af testprocessen. Testmanagere bør antage at de vil være de drivende kræfter bag ændringer og forbedringer af testprocessen og skal derfor kende til teknikkerne diskuteret i dette kapitel, og som er accepteret inden for branchen. Yderligere information omkring testforbedringsprocessen diskuteres i pensum for Expert Improving the Test Process.

5.2 Processen for testforbedring

På samme måde som test bruges til at forbedre software, vælges og bruges softwarekvalitetsprocesser til at forbedre softwareudviklingsprocessen (og de resulterende softwareprodukter). Der kan også ske procesforbedringer af testprocesserne. Der er forskellige måder og metoder tilgængelige til forbedring af test af software og af systemer, der indeholder software. Disse metoder sigter mod at forbedre processen og derved de leverede produkter ved hjælp af vejledninger og forbedringsområder.

Test tegner sig ofte for en betydelig del af de totale projektomkostninger. Men der er kun begrænset opmærksomhed på testprocessen i de forskellige modeller til forbedring af softwareprocessen, f.eks. CMMI (se flere detaljer nedenfor).

Testforbedringsmodeller, f.eks. Test Maturity Model integration (TMMi®), Systematic Test and Evaluation Process (STEP), Critical Testing Processes (CTP) og TPI Next® blev udviklet for at dække den manglende dækning af test i de fleste software procesforbedringsmodeller. Hvis de anvendes rigtigt kan man ved hjælp af disse modellers metrikker i en vis udstrækning sammenligne på tværs af organisationer.

Når modellerne gennemgås i dette afsnit, skal det ikke ses som en anbefaling til at bruge netop dem, men som et forsøg på at give et repræsentativt overblik over, hvordan modellerne fungerer, og hvad de omfatter.

5.2.1 Introduktion til procesforbedring

Procesforbedringer er både relevante for softwareudviklingsprocessen og for testprocessen. At lære af sine egne fejltagelser, gør det muligt at forbedre den proces, organisationer bruger til at udvikle og teste software. Deming-forbedringscyklussen: Plan, Do, Check, Act (planlæg, udfør, kontroller, reager), er blevet brugt i mange årtier, og den er stadig relevant, når testere skal forbedre den proces, der bruges i dag.

En forudsætning for procesforbedring er troen på, at kvaliteten i et system i høj grad påvirkes af kvaliteten af den proces, der bruges til at udvikle softwaren. Forbedret kvalitet i softwarebranchen reducerer behovet for ressourcer til at vedligeholde softwaren og giver derfor mere tid til udarbejdelse af flere og bedre løsninger i fremtiden. Procesmodeller giver et udgangspunkt for forbedringer gennem måling af organisationens procesmodenhed. Modellen udgør også en ramme for forbedring af organisationens processer på basis af resultatet af en vurdering.

Ved at vurdere procesmodenheden kan man blive motiveret til at forbedre processerne. En senere procesvurdering vil så kunne vise effekten.

5.2.2 Typer af procesforbedring

Det er en udbredt metode at anvende modeller til vurdering hvilket sikrer en ensartet tilgang til forbedring af testprocesser ved hjælp af praksis, som er afprøvet og som folk har tillid til.

Modeller til procesforbedring inddeles i to typer:

1. Procesreferencemodellen anvendes til at vurdere en organisations evne i forhold til modellen og organisationen i forhold til modelrammen, til at lave en køreplan for at forbedre processen og måle organisationens modenhed som en del af vurderingen
2. Indholdsreferencemodellen giver en forretningsmæssig vurdering af organisationens muligheder for at forbedre og i nogle tilfælde mulighed for at sammenligne med gennemsnittet for branchen ved hjælp af objektive målinger. Ud fra denne vurdering kan der laves en plan for at forbedre testprocessen.

Forbedring af testprocessen kan også opnås uden at anvende en model ved for eksempel at anvende en analytisk tilgang og evalueringsmøder (retrospektive møder).

5.3 Forbedring af testprocessen

It-branchen kan arbejde med modeller til forbedring af testprocessen for at nå til et højere niveau af modenhed og professionalisme. Branchestandardmodeller hjælper med til at udvikle tværorganisatoriske metrikker og mål, der kan bruges til sammenligning. Ud fra behovet for procesforbedring i testbranchen er der opstået flere bud på anbefalede processer. Disse omfatter STEP, TMMi, TPI Next og CTP. De trinvise modeller, som TMMi og CMMI, giver standarder til sammenligning på tværs af forskellige virksomheder og organisationer. De kontinuerte modeller, som CTP, STEP og TPI Next, giver en organisation mulighed for at løse de problemer, der har højest prioritet, i en mere fri implementeringsorden. Hver enkelt af disse gennemgås yderligere i dette afsnit.

Alle disse modeller giver organisationer mulighed for at bestemme, hvor de befinder sig med hensyn til deres aktuelle testprocesser. Når der først er udført en vurdering, giver TMMi og TPI Next en foreskreven køreplan for forbedring af testprocessen. Derimod giver STEP og CTP organisationerne metoder til at bestemme, hvor de får det største afkast af investeringen i procesforbedringer, og overlader til organisationerne at vælge den passende køreplan.

Når der er opnået enighed om, at testprocesserne skal gennemgås og forbedres, kan implementeringstrinene til procesforbedring for denne aktivitet defineres ud fra IDEAL[SM] modellen [IDEAL96]:

- Initiering af forbedringsprocessen
- Diagnosticering af den nuværende situation
- Etablering af plan for forbedring af testprocessen
- Handling for at implementere forbedringen
- Læring baseret på forbedringsprocessen.

Initiering af forbedringsprocessen
Før procesforbedringsaktiviteten starter, skal formålene, målene, omfanget og dækningen af procesforbedringerne aftales med interessenterne. Valget af procesmodel sker også i denne aktivitet. Modellen kan enten vælges blandt dem, der er publiceret (som f.eks. CTP, STEP, TMMi og TPI Next), eller den kan udvikles internt i virksomheden. Derudover skal succeskriterierne defineres, og der skal implementeres en metode, som de kan måles efter under forbedringsaktiviteten.

Diagnosticering af den nuværende situation

Den aftalte vurderingsmetode anvendes, og der udarbejdes en vurderingsrapport, som indeholder en vurdering af de nuværende fremgangsmåder inden for test samt en liste over mulige procesforbedringer.

Etablering af plan for forbedring af testprocessen
Listen over mulige procesforbedringer sættes i prioriteringsrækkefølge. Rækkefølgen kan være baseret på investeringsafkast, risici, overensstemmelse med organisatorisk strategi og/eller målbare kvantitative eller kvalitative fordele. Efter at have foretaget prioriteringen udarbejdes en plan for at implementere forbedringerne.

Aktiviteter for at implementere forbedringen
Planen for implementering af forbedringer udføres. Dette kan omfatte eventuelt nødvendig uddannelse eller oplæring, pilotanvendelse af processerne og endelig fuld implementering af disse.

Læring baseret på forbedringsprogrammet
Når procesforbedringerne er gennemført fuldt ud, er det vigtigt at følge op på hvilke fordele (af dem der blev aftalt før forbedringen samt uforudsete forbedringerne) der er opnået. Det er også vigtigt at følge op på, hvilke af succeskriterierne for procesforbedringsaktiviteten der er opfyldt.

Afhængigt af den anvendte procesmodel er det på dette trin i processen, at overvågningen af det næste modenhedsniveau starter, og der tages en beslutning om enten at starte forbedringsprocessen igen eller standse aktiviteten på dette punkt.

5.4 Forbedring af testprocessen med TMMi

Testing Maturity Model integration (TMMi) er opbygget af fem niveauer og er ment som et supplement til CMMI. Hvert af niveauerne indeholder definerede procesområder, der skal være 85% opfyldt ved at opnå specifikke og generiske mål, før organisationen kan gå videre til næste niveau.

TMMi modenhedsniveauerne er:

- Niveau 1: Initiel
  Startniveauet repræsenterer en tilstand, hvor der ikke er nogen formelt dokumenteret eller struktureret proces. Test udarbejdes ad hoc efter kodning, og test betragtes som det samme som debugning. Formålet med test forstås som at bevise, at softwaren fungerer
- Niveau 2: Styret
  Det andet niveau kan nås ved at testprocesser klart adskilles fra debugging. Det kan opnås ved at angive testpolitikker og -mål, introducere aktiviteterne i den grundlæggende testproces (f.eks. testplanlægning) og ved at implementere basale testteknikker og -metoder
- Niveau 3: Definet
  Det tredje niveau nås, når en testproces er integreret i softwareudviklingslivscyklussen, og den er dokumenteret i formelle standarder, procedurer og metoder. Der foretages reviews og der bør være en klar softwaretestfunktion, der kan kontrolleres og overvåges
- Niveau 4: Målt
  Niveau fire er nået, når testprocessen på organisationsniveau kan måles og styres på en effektiv måde, som kommer specifikke projekter til gode
- Niveau 5: Optimeret
  Det sidste niveau repræsenterer en tilstand i testprocessens modenhed, hvor data fra testprocessen kan bruges som en hjælp til at forhindre defekter, og hvor fokus er på optimering af den etablerede proces.

For mere information om TMMi, se [vanVeenendaal11] og [www.tmmi.org].

5.5 Forbedring af testprocessen med TPI Next

TPI Next model definerer 16 nøgleområder, som hver dækker et bestemt aspekt af testprocessen, som f.eks. teststrategi, metrikker, testværktøjer og testmiljø.

Modellen definerer fire modenhedsniveauer:

- Initiel
- Kontrolleret
- Effiektiv
- Optimeret.

Der er defineret bestemte kontrolpunkter til at vurdere hvert nøgleområde for hvert modenhedsniveau. Resultatet opsummeres og præsenteres visuelt ved hjælp af en modenhedesmatrice, som dækker alle nøgleområder. Ud fra testgruppens behov og kapacitet defineres formålene med forbedringerne og hvordan de opnås.

TPI Next fungerer uafhængig af hvilken som helst model til forbedring af softwareprocessen på grund af dens generelle tilgang. Den dækker testtekniske aspekter og hjælper ledelsen med at træffe beslutninger [deVries09].

For mere information om TPI Next, se [www.tpinext.com].

5.6 Forbedring af testprocessen med CTP

Den basale forudsætning for Critical Testing Process vurderingsmodellen er at visse testprocesser er kritiske. Disse kritiske processer understøtter succesfulde testgrupper, hvis de udføres korrekt. Hvis det ikke sker, vil det være svært for selv dygtige testere og testmanagere at få succes med opgaven. Modellen identificerer tolv kritiske testprocesser. CTP er hovedsagelig en indholdsreferencemodel.

CTP modellen er en kontekstafhængig tilgang, der tillader, at modellen skræddersys, bl.a. ved:

- Identifikation af specifikke udfordringer
- Erkendelse af attributter for gode processer
- Valg af rækkefølgen og vigtigheden af implementering af procesforbedringer.

CTP modellen kan tilpasses inden for sammenhængen af alle livscyklusmodeller for softwareudvikling.

Udover interviews af deltagere anvender CTP modellen metrikker til at sammenligne organisationer i forhold til gennemsnittet i branchen og bedste praksis.

For mere information om CTP, se [Black03].

5.7 Forbedring af testprocessen med STEP

STEP står for "Systematic Test and Evaluation Process". Denne metode kræver ikke, at forbedringer sker i en bestemt rækkefølge. Det samme gælder for CTP, men ikke for TMMi og TPI Next.

STEP er primært en indholdsreferencemodel baseret på den ide, at test er en livscyklusaktivitet, der starter under formuleringen af krav og fortsætter indtil systemet trækkes tilbage. STEP-metoden lægger vægt på "test, derefter kodning" gennem anvendelse af en kravbaseret teststrategi, der skal sikre, at en tidlig oprettelse af testcases validerer kravspecifikationerne før design og kodning.

Metodens grundlæggende principper indebærer:

- En kravbaseret teststrategi
- Testindsatsen starter i begyndelsen af livscyklussen
- Testene anvendes som krav og brugsmodeller
- Design af test-delprodukter styrer design af softwaren
- Defekter findes tidligt eller undgås helt
- Defekter analyseres systematisk
- Testere og udviklere samarbejder.

I nogle tilfælde blandes STEP vurderingsmodellen med TPI Next modenhedsmodellen.

For mere information on STEP, se [Craig02].

6 Testværktøjer og automatisering – 135 min

Nøgleord

Open-source-værktøj, brugerdefineret værktøj

Læringsmål for testværktøjer og automatisering

6.2 Valg af testværktøj

TM-6.2.1 (K2) Beskriv de ledelsesmæssige opgaver ved valg af et open-source værktøj.
TM-6.2.2 (K2) Beskriv de ledelsesmæssige opgaver ved valg af et brugerdefineret værktøj.
TM-6.2.3 (K4) Vurdér en givet situation med planlægning af valg af værktøj, herunder risici, omkostninger og fordele ved det valgte værktøj.

6.3 Livscyklus for testværktøj

TM-6.3.1 (K2) Forklar de forskellige faser i et værktøjs livscyklus.

6.4 Registrering af data med testværktøj

TM-6.4.1 (K2) Beskriv hvordan registrering og vurdering af metrikker kan forbedres ved hjælp af værktøjer.

6.1 Introduktion

Dette afsnit udvider i forhold til pensum for Foundation-niveauet ved at dække en række generelle begreber som en testmanager skal overveje i relation til værktøjer og automatisering.

6.2 Valg af testværktøj

Der er en række forskellige spørgsmål som en testmanager skal overveje når der vælges testværktøjer.

Det mest almindelige valg, historisk set, er at købe et værktøj fra en kommerciel leverandør. I nogle tilfælde kan det være det eneste valg. Men der er andre muligheder, bl.a. open source-værktøjer og tilpassede værktøjer, som også er realistiske alternativer.

Uanset hvilken type af værktøj, skal en testmanager undersøge de samlede ejeromkostninger gennem værktøjets forventede levetid ved at udføre en cost-benefit analyse. Dette emne er dækket nedenfor i afsnittet om investeringsafkast (ROI).

6.2.1 Open source-værktøjer

Der findes open source-værktøjer til næsten alle dele af testprocessen, f.eks. håndtering af testcases, defekter og automatiseret test. For open source-værktøjer er det vigtigt at være opmærksom på omkostningerne til (manglende) support. Mens selve værktøjet generelt er billigt, så findes der måske ingen formel support for brugerne af værktøjet. Mange open source-værktøjer har dog aktive brugere, der formidler uformel support til brugerne.

Mange open source-værktøjer blev oprindeligt udviklet for at løse et specifikt problem eller besvare et enkelt spørgsmål. Derfor kan værktøjet måske ikke udføre alle de funktioner som et lignende indkøbt værktøj. Derfor bør en omhyggelig analyse af de faktiske behov for testgruppen udføres før man vælger et open source-værktøj.

En fordel ved at bruge open source-værktøjer er, at de normalt kan ændres eller udvides af deres brugere. Hvis testgruppen har disse kernekompetencer, kan værktøjet modificeres til at arbejde sammen med andre værktøjer eller ændres, så det passer til teamets behov. Flere værktøjer kan kombineres for at løse problemer som èt færdigkøbt værktøj ikke kan løse. Naturligvis, jo flere værktøjer man anvender og jo flere modifikationer der laves, jo mere kompleksitet tilføres. Testmanageren må sørge for at testgruppen ikke begynder at bruger open source-værktøjer blot for at bruge dem, som med andre værktøjer, så skal indsatsen altid været rettet mod at udlede et positivt investeringsafkast (ROI).

En testmanager skal forstå licensordningen for det valgte værktøj. Mange open source-værktøjer kommer med en variation af GNU General Public License som specificerer at distribution af software altid skal være under de samme vilkår, som da det blev modtaget. Skulle testgruppen foretage ændringer i værktøjet for bedre at støtte deres test, skal alle disse ændringer stilles til rådighed for alle eksterne brugere af værktøjet under værktøjslicensen. Testmanageren skal checke de juridiske konsekvenser af, at omfordele softwaren til deres organisation.

Organisationer som udvikler sikkerhedskritisk eller forretningskritisk software eller er genstand for overholdelse af lovgivningen kan have problemer med at anvende open source-værktøjer. Mens mange open source-værktøjer er af høj kvalitet, er nøjagtigheden af et givet open source-værktøj sandsynligvis ikke certificeret. Indkøbte værktøjer er ofte certificerede for deres nøjagtighed og

egnethed til en bestemt opgave (f.eks., DO-178B). Mens et open source-værktøj kan være lige så godt, kan certificeringen være gruppens ansvar, og det kan skabe ekstra overhead.

6.2.2 Brugerdefinerede værktøjer

Testgruppen kan opleve at de har et specifikt behov, for hvilket der ikke findes nogen leverandør af værktøj eller open source-værktøj til rådighed. Det kan skyldes en proprietær hardwareplatform, et tilpasset miljø eller en proces der er blevet ændret på en usædvanlig måde. I sådanne tilfælde, hvis kernekompetencen eksisterer i teamet, kan testmanageren overveje at udvikle et brugerdefineret værktøj.

Fordelene ved at udvikle et brugerdefineret værktøj er, at værktøjet kan opfylde teamets behov præcist og kan operere effektivt i den kontekst som teamet kræver. Værktøjet kan udvikles til at interagere med andre værktøjer i brug, og at generere data i den eksakte form som er nødvendig for teamet. Derudover kan værktøjet måske finde anvendelse i organisationen andre steder end i det umiddelbare projekt. Men, før planlægning af frigivelse af værktøjet for andre projekter, er det vigtigt at reviewe formål, hensigt, fordele og mulige ulemper først.

En testmanager som overvejer at udvikle brugerdefinerede værktøjer skal også overveje mulige negative konsekvenser. Tilpassede værktøjer er ofte afhængige af den person som har udviklet værktøjet. Derfor skal brugerdefinerede værktøjer være tilstrækkeligt dokumenteret, så de kan vedligeholdes af andre. Hvis dette ikke sker, kan værktøjet blive "forældreløst" og ikke blive brugt, når værktøjets skaber forlader projektet. Med tiden kan det ske, at brugerdefinerede værktøjer bliver brugt til noget andet end hvad der oprindelig var hensigten. Det kan give problemer med værktøjets kvalitet og kan f.eks. give falske positive defektrapporter eller danne forkerte data. Et brugerdefineret værktøj er et softwareprodukt og er underlagt krav til udvikling som ethvert andet softwareprodukt. Tilpassede værktøjer skal designes og testes for at de kan fungere som forventet.

6.2.3 Investeringsafkast - Return of Investment (ROI)

Det påhviler en testmanager at sikre at alle værktøjer der er introduceret til testgruppen tilfører værdi til teamets arbejde og kan vise et positivt investeringsafkast for organisationen. For at et værktøj skaber reelle og varige fordele, må der udføres en cost-benefit-analyse før køb eller udvikling af et værktøj. I denne analyse af investeringsafkastet bør man overveje både engangs- og gentagne udgifter. Nogle kan måles i penge, andre er ressourcer, tid og risici der kan reducere værdien af værktøjet.

Engangs-omkostninger omfatter:

- Definition af krav til værktøjet for at opfylde målsætninger og mål
- Evaluering og valg af det korrekte værktøj og værktøjsleverandør
- Indkøb, tilpasning eller udvikling af værktøjet
- Udføre grunduddannelsen i værktøjet
- Integration af værktøjet med andre værktøjer
- Fremskaffelse af den hardware/software som er nødvendig for at understøtte værktøjet.

Gentagne omkostninger omfatter:

- Ejerskab
    - licenser og supportgebyrer
    - vedligeholdelsesomkostninger for selve værktøjet
    - vedligeholdelse af artefakter skabt ved hjælp af værktøjet
    - udgifter til uddannelse og mentorordning.
- Portering af værktøjet til forskellige miljøer

- Tilpasning af værktøjet til fremtidige behov
- Forbedring af kvalitet og processer for optimal udnyttelse af de udvalgte værktøjer.

En testmanager skal også overveje de mulige omkostninger ved ethvert værktøj. Tiden der er brugt på at erhverve, administrere, træne og hjælpe med at bruge værktøjet kunne være blevet brugt på faktiske testopgaver, og derfor kan der være behov for flere testressourcer up front inden værktøjet tages i brug.

Der er mange risici ved at bruge værktøjer. Ikke alle værktøjer leverer faktiske fordele som kan opveje disse risici. Risici for værktøjer blev drøftet i pensum for Foundation-niveauet. Testmanageren bør også overveje følgende risici, når der overvejes investeringsafkast (ROI):

- Organisationens umodenhed (den er ikke klar til at bruge værktøjet)
- Artefakter skabt af værktøjet, kan være vanskelige at vedligeholde og kræver mange revisioner, når softwaren under test ændres
- Reduktion af testanalytikeres involvering i testopgaver kan reducere værdien af test (f.eks. kan effektiviteten af defektafsløringen blive reduceret når det kun er automatiserede scripts der bliver kørt.

Endelig skal en testmanager se på de fordele der kan høstes af brugen af værktøjet. Fordele ved indførelse af værktøjer og deres brug kan omfatte:

- Reduktion af gentagende arbejde
- Reduktion af testcyklustid (f.eks. ved brug af automatiseret regressionstest)
- Reduktion af omkostninger ved testafvikling
- Forøgelse af vise typer af test (f.eks. regressionstest)
- Reduktion af menneskelige fejl i forskellige faser af test. Eksempler:
    - testdata kan blive mere valide ved brug af værktøjer til datagenerering
    - sammenligninger af testresultater kan blive mere korrekte ved brug af sammenligningsværktøjer
    - indtastning af testdata kan blive mere korrekte ved at anvende et scripting værktøj.
- Reduktion af den krævede indsats for at få adgang til information omkring test, for eksempel:
    - værktøjsgenererede rapporter og metrikker
    - genbrug af testaktiver som f.eks. testcases, test scripts og testdata.
- Forøgelse af test som ikke var mulig uden værktøjer (f.eks. performancetest, load test)
- Forbedring af status for testere som skaber automatisering og testgruppen som helhed ved at demonstrere forståelsen og brugen af avancerede værktøjer.

I almindelighed er det sjældent at en testgruppe kun bruger et enkelt værktøj. Det samlede investeringsafkast (ROI) som testgruppen vil opnå er normalt en funktion af alle de værktøjer, der anvendes. Værktøjer er nødt til at dele information og arbejde kooperativt sammen. Det er tilrådeligt med en langsigtet og omfattende testværktøjsstrategi.

6.2.4 Udvælgelsesproces

Testværktøjer er en langsigtet investering, som sandsynligvis strækker sig over mange iterationer i et enkelt projekt og/eller gældende for mange projekter. En testmanager skal overveje et værktøj fra flere forskellige synsvinkler.

- For forretningen er et positivt investeringsafkast (ROI) krævet. For at opnå høj værdi i deres investeringer, skal organisationen sikre, at de værktøjer der skal fungere sammen – som kan omfatter både testværktøjer og ikke testværktøjer – arbejder sammen. I nogle tilfælde skal processer og tilslutningsmuligheder for værktøjsanvendelse forbedres, for at tilvejebringe en sådan interaktion, og dette kan tage noget tid at opnå

- For projektet skal værktøjet være effektivt, f.eks. ved at undgå tastefejl når man indtaster data. Værktøjet kan være lang tid om at give et positivt Investeringsafkast (ROI). I mange tilfælde kan investeringsafkastet (ROI) først komme i en anden release eller under vedligeholdelse, snarere end under det oprindelige projekt, hvor automatiseringen startede. Testmanageren må overveje livscyklus for den samlede applikation
- For brugerne skal værktøjet hjælpe til at udføre opgaverne på en effektiv måde. Man må kunne lære at bruge værktøjet hurtigt og med minimalt stress. Desuden har brugerne behov for uddannelse og mentoring.

For at sikre at alle synspunkter er taget i betragtning, er det vigtigt at skabe en køreplan for introduktionen af et testværktøj.

Udvælgelsesprocessen for et testværktøj blev allerede diskuteret i pensum for Foundation-niveauet som følger:

- Vurdere den organisatoriske modenhed
- Identificèr krav til værktøjet
- Vurdèr værktøjer
- Vurdèr sælger eller service support (open source-værktøjer, brugerdefinerede værktøjer)
- Identificèr interne krav til coaching og mentoring i brugen af værktøjerne
- Vurdèr uddannelsesbehov og overvej den aktuelle testgruppes testautomatiseringsevner
- Beregn cost-benefit (som omtalt i afsnit 6.2.3 ROI).

Testmanageren må overveje disse egenskaber for hver type af værktøj:

Analyse:

- Vil dette værktøj kunne “forstå” det input der gives?
- Er værktøjet egnet til formålet?

Design:

- Vil dette værktøj hjælpe med at designe testprodukter baseret på eksisterende information (f.eks. testdesignværktøjer der skaber testcases ud fra kravene)?
- Kan designet genereres automatisk?
- Kan den aktuelle testware kode blive genereret eller delvist genereret i et vedligeholdelsesvenligt og brugbart format?
- Kan de nødvendige testdata genereres automatisk (f.eks. data genereret fra kodeanalyse)?

Data og testudvælgelse:

- Hvordan udvælger værktøjet de data der skal bruges (f.eks. Hvilke testcases skal udføres med Hvilke sæt af data)?
- Kan værktøjet acceptere udvælgelseskriterier som indtastes enten manuelt eller automatisk?
- Kan værktøjet afgøre hvordan man forvanske data produktionsdata baseret på udvalgte input?
- Kan værktøjet afgøre hvilke test der er nødvendige, baseret på dækningskriterier? (kan værktøjet ved et givet sæt af krav f.eks. krydse sporbarhed for at afgøre hvilke testcases der er nødvendige til testudførelsen?)

Udførelse:

- Vil værktøjet køre automatisk eller vil manuel indgriben være nødvendig?
- Hvordan stopper og genstarter værktøjet?

- Skal værktøjet kunne "lytte" til relevante hændelser (skal et teststyringsværktøj f.eks. kunne opdatere status for en testcase hvis nogen rapporterer en defekt mod testcasen, og defekten i øvrigt er lukket?)

Evaluering:
- Hvordan vil værktøjet afgøre om det har modtaget et korrekt resultat (f.eks. vil værktøjet bruge et testorakel for at afgøre korrektheden af svaret)?
- Hvilke typer af fejl kan værktøjet afsløre?
- Yder værktøjet tilstrækkelig logning og rapportering?

6.3 Livscyklus for testværktøj

Der er fire forskellige stadier i et værktøjs nyttefulde livscyklus som en testmanager skal styre, som følger:

1. Erhvervelse. Værktøjet skal erhverves som beskrevet ovenfor (se afsnit 6.2 Værktøjsudvælgelse). Efter en beslutning om at anvende værktøjet er taget, bør testmanageren tildele nogen (ofte en testanalytiker eller en teknisk testanalytiker) administreringen af værktøjet. Denne person skal træffe beslutninger om hvordan og hvornår værktøjet skal bruges, hvor skabte artefakter skal gemmes, navngivning m.v. At tage disse beslutninger up front snarere end at tillade dem at ske på en ad hoc måde kan gøre en signifikant forskel i det eventuelle investeringsafkast (ROI) af værktøjet. Der vil sikkert være brug for at levere uddannelse til brugerne af værktøjet
2. Support og vedligeholdelse. Der vil være krav om løbende support og vedligeholdelse af værktøjet. Ansvaret for vedligeholdelse af værktøjet påhviler administratoren af værktøjet eller det kan tilknyttes en dedikeret værktøjsgruppe. Hvis værktøjet skal arbejde sammen med andre værktøjer, skal dataudvekslingen og processer for samarbejde overvejes. Det er nødvendigt at overveje beslutninger om backup og gendannelse af værktøjet og dets outputs
3. Evolution - overvej konvertering. Med tiden kan ændringer i miljø, forretningsbehov eller leverandørens forhold kræve store ændringer for værktøjet og brugen af det. For eksempel kan en leverandør kræve at et værktøj skal opdateres. Det kan give problemer med andre værktøjer. På samme måde kan en ændring af miljøet af forretningsmæssige årsager give problemer. Jo mere komplekst driftsmiljøet er, jo mere kan ændringer påvirke anvendelsen. Afhængigt af hvilken rolle værktøjet spiller for testen må en testmanager have behov for at sikre, at organisationen kan håndtere den løbende vedligeholdelse
4. Pensionering. Den tid vil komme hvor værktøjet har overlevet sin anvendelighed. På dette tidspunkt vil værktøjet være nødt til at blive pensioneret yndefuldt. Funktionaliteten leveret af værktøjet skal udskiftes og data vil skulle bevares og arkiveres. Dette kan forekomme fordi værktøjet er ved slutningen af sin livscyklus eller simpelthen fordi det har nået et punkt hvor fordele og muligheder for omstilling til et nyt værktøj overstiger omkostningerne og risici.

Gennem værktøjets livscyklus er det testmanagerens ansvar at sikre en velfungerende fortsættelse af den nytte, som værktøjet har skabt for testgruppen.

6.4 Registrering af data med testværktøj

En testmanager kan designe og samle metrikker fra testværktøjer. Der findes forskellige værktøjer som kan registrere data i realtid og dermed reducerer arbejdet med at indsamle data. Testmanageren kan derefter bruge disse data til at styre den samlede testindsats.

Der findes forskellige værktøjer, der kan indsamle forskellige typer af data:
- Teststyringsværktøjer kan levere en række forskellige metrikker. Man kan registrere den aktuelle dækning af krav i testcases og automatiserede scripts, og man kan få et

øjebliksbillede af anvendte og planlagte testcases og deres aktuelle status (godkendt, fejlet, afvist, blokeret, i kø)
- Defektstyringsværktøjer kan levere et væld af informationer om defekter: aktuel status, alvorsgrad, prioritet, område osv. Andre oplysende data som de faser hvor defekterne er introduceret og fundet, hvor stor en del der bliver fundet, osv., hjælper testmanageren med procesforbedringen
- Statiske analyseværktøjer kan hjælpe med at opdage og rapportere problemer vedrørende vedligeholdelsesegnethed
- Performanceværktøjer kan levere værdifulde oplysninger om skalèrbarhed af systemet
- Dækningsværktøjer kan vise testmanageren, hvor meget af et system som er blevet berørt af en test.

Disse krav skal være korrekt implementeret som en del af værktøjskonfigurationen for at sikre, at de informationer, som spores af værktøjerne, kan rapporteres på måder som vil være forståelig for interessenterne.

7 Personlige kvalifikationer og sammensætning af testgruppen – 210 min.

Nøgleord
Uafhængighed i test

Læringsmål for personlige kvalifikationer og sammensætning af testgruppen

7.2 Individuelle kvalifikationer

TM-7.2.1 (K4) Analysér gruppemedlemmernes stærke og svage sider med hensyn til softwaresystemer, domæne- og forretningsviden, systemudvikling, softwaretest og samarbejdsevner.

TM-7.2.2 (K4) Analysér en given vurdering af kvalifikationerne i en gruppe og fastlæg en uddannelsesplan for udvikling af disse kvalifikationer.

7.3 Dynamik i testgruppen

TM-7.3.1 (K2) Diskuter de nødvendige "hårde" og "bløde" kvalifikationer som kræves for at lede en testgruppe i en given situation.

7.4 Testgruppen i den større organisation

TM-7.4.1 (K2) Forklar de forskellige muligheder for uafhængig test.

7.5 Motivation af testerne

TM-7.5.1 (K2) Giv eksempler på forskellige faktorer der motiverer og demotiverer testere.

7.6 Kommunikation

TM-7.6.1 (K2) Forklar de faktorer som påvirker kommunikationen internt i testgruppen og mellem testgruppen og dens interessenter.

7.1 Introduktion

Gode testmanagere rekrutterer, ansætter og vedligeholder grupper med en god blanding af kvalifikationer. Krav til kvalifikationer kan ændre sig over tid, så udover i første omgang at ansætte de rigtige folk er det også vigtigt at sørge for passende uddannelse/træning og muligheder for personlig udvikling, for at bevare testgruppen og for til stadighed at levere topresultater. Udover gruppens kvalifikationer skal testmanageren også vedligeholde et sæt af kvalifikationer som sætter folkene i stand til at arbejde effektivt under stort pres, og i et miljø med højt tempo.

Dette kapitel ser på hvordan man vurderer færdigheder, hvordan man udfylder hullerne for at skabe en synergiskabende gruppe der hænger godt sammen internt og er velfungerende i en organisation, og på hvordan en sådan gruppe motiveres og hvordan man kommunikerer effektivt.

7.2 Individuelle kvalifikationer

En medarbejders evne til at teste software kan opnås gennem erfaring eller ved uddannelse og træning. Hvert af følgende punkter kan bidrage til testerens viden:

- Anvendelse af softwaresystemer
- Kendskab til domænet eller virksomheden
- Deltagelse i aktiviteter i forskellige faser af softwareudviklingsprocessen, herunder analyse, udvikling og teknisk support
- Deltagelse i softwaretestaktiviteter.

Slutbrugerne af softwaresystemer har en god forståelse af hvordan systemet fungerer, hvor afvigelser vil have den største effekt, og hvordan systemet skal reagere i forskellige situationer. Brugere med domæneekspertise, ved, hvilke områder der er af størst betydning for virksomheden, og hvordan disse områder påvirker virksomhedens evne til at opfylde sine krav. Denne viden kan bruges til at hjælpe med til at prioritere testaktiviteterne, oprette realistiske testdata og testcases og verificere eller udarbejde usecases.

Kendskab til softwareudviklingsprocessen (behovsanalyse, arkitektur, design og kodning) giver indsigt i, hvordan fejltagelser fører til at defekter introduceres, hvor defekter kan findes, og hvordan man i første omgang kan forhindre, at de introduceres. Erfaring med teknisk støtte giver viden om brugeroplevelsen, forventninger og krav til brugervenlighed. Erfaring med softwareudvikling er vigtig for brugen af testværktøjer, der kræver ekspertise inden for programmering og design, samt for at deltage i statisk kodeanalyse, kodereview, komponenttest og teknisk integrationstest.
Specifikke softwaretestkvalifikationer omfatter de egenskaber, der er beskrevet i pensum for Foundation-niveauet, Advanced Test Analyst og Advanced Technical Test Analyst, f.eks. evnen til at analysere en specifikation, deltage i risikoanalyse, designe testcases og vise omhu ved afvikling af test og registrering af resultaterne.

Det er specielt vigtigt, at testmanagere har viden om, færdigheder og erfaring i projektledelse, da teststyring omfatter mange projektledelsesaktiviteter, f.eks. udarbejdelse af en plan, opfølgning på fremdrift og rapportering til interessenter. Hvis der ikke er en projektleder, kan testmanageren også påtage sig rollen som projektleder, især i de senere faser i et projekt. Hvad det kræver af kvalifikationer ligger ud over hvad der er beskrevet i både pensum for Foundation-niveauet og i dette pensum.

For en tester er det ikke tilstrækkeligt kun at være teknisk kompetent, for sociale færdigheder er også vigtige i testrollen. Som tester må man kunne give og modtage konstruktiv kritik, påvirke andre og forhandle. Desuden må en tester også kunne organisere sine opgaver, tænke i detaljer og beherske skriftlig og mundtlig kommunikation.

Den ideelle testgruppe består af en blanding af kvalifikationer og har forskellige niveuaer af erfaringer samt at medlemmerne af gruppen skal have en vilje og evne til at undervise og lære af deres kollegaer. I nogle situationer er visse færdigheder vigtigere eller mere respekteret end andre. I tekniske situationer hvor det er nødvendigt med f.eks. API test og programmeringsfærdigheder, kan tekniske færdigheder være højere værdsat end domæneviden. I en situation med black-box test kan domæneviden være det mest værdsatte. Det er vigtigt at huske på at situationer og projekter ændrer sig.

Når en testmanager laver et regneark til vurdering af færdigheder skal man liste alle færdigheder, som er vigtige for at varetage opgaven samt passende for stillingen. Når disse færdigheder er listet, kan hver person i gruppen vurderes ud fra et scoringssystem (f.eks. en vurdering fra 1 til 5, hvor 5 er det højeste, forventede færdighedsniveau for området). Man kan vurdere de enkelte testere og deres stærke og svage sider. Ud fra det kan man lægge planer for undervisning i grupper eller individuelt. En testmanager kan fastsætte personlige mål for de enkelte testere og bør definere hvordan man vurderer den enkeltes færdigheder.

Testere bør ansættes på langt sigt, ikke for at arbejde på et enkelt projekt. En testmanager kan skabe et miljø med vedvarende læring. Derved bliver gruppens medlemmer motiveret til at forbedre deres kvalifikationer og viden, så de er klar til den næste udfordring.

Det er sjældent at en testmanager har mulighed for at ansætte de perfekte testere til gruppen. Og selv hvis gruppens medlemmer passer perfekt til det nuværende projekt er de ikke nødvendigvis det perfekte miks for det næste projekt. Testmanageren bør ansætte folk som er intelligente, nysgerrige, fleksible, villige til at arbejde, i stand til at arbejde effektivt som en del af en gruppe, og villige og i stand til at lære. Selvom gruppen af perfekte enkeltpersoner sandsynligvis ikke er til rådighed er det stadig muligt at bygge en stærk gruppe, ved at afbalancere de enkelte medlemmers styrker og svagheder.

Ved hjælp af regnearket til vurdering af kompetencer kan testmanageren se hvor gruppen er stærk og hvor den er svag. Denne information danner grundlaget for en plan for udvikling af færdigheder og træning. Testmanageren skal beslutte hvordan han vil forholde sig til de områder, som er de svageste og har den største virkning på gruppens effektivitet og egnethed. En fremgangsmåde er træning, f.eks. at sende folk på et eksternt kursus, have interne træningssessioner, udvikle tilpasset træning, anvende e-læringskursus. En anden tilgang er selvstudium, f.eks. bøger, webinards, internetressourcer. Endnu en fremgangsmåde er krydstræning, f.eks. sætte en person som gerne vil lære en færdighed til at arbejde sammen med en som allerede har den færdighed, på en opgave som kræver den færdighed, have lokale eksperter til at give en kort introduktion til deres område, osv. (Mentoring er en lignende metode hvor en person som er ny i en rolle bliver sat sammen med en erfaren person, som har haft den rolle tidligere, og den erfarne person fungerer som en løbende ressource der giver råd og støtte.) Testmanageren skal både forholde sig til gruppens styrker og svagheder, når han planlægger træning og udvikling af kompetencer. For mere information om sådanne uddannelsesplaner for testgrupper se [McKay07].

7.3 Dynamik i testgruppen

For at kunne sammensætte den bedst mulige gruppe er udvælgelse af medarbejdere en af de vigtigste ledelsesopgaver i organisationen. Der er mange ting, der skal overvejes i forbindelse med de individuelle færdigheder, som jobbet kræver. Når der udvælges enkeltpersoner til at være med i en gruppe skal man overveje testgruppens indre dynamik. Komplementerer denne person de kvalifikationer og personlighedstyper, der allerede findes i testgruppen? Det er vigtigt at overveje fordelene ved at have forskellige personlighedstyper så vel som en blanding af forskellige tekniske

færdigheder i testgruppen. En stærk testgruppe er i stand til at håndtere flere projekter med varierende kompleksitet samtidig med, at den kan håndtere de sociale og menneskelige kontakter til andre projektgruppemedlemmer.

Test er ofte et arbejde under megen pres. Softwareudviklingsplaner bliver ofte afkortet, selv urealistisk afkortet. Interessenter har høje forventninger til testgruppen, nogle gange urimeligt høje. Testmanageren skal ansætte folk som kan håndtere situationer med højt pres, som kan afvise frustrationer og som kan koncentrere sig om arbejdet, selv når tidsplanerne virker umulige. Det er op til testmanageren at håndtere problemer relateret til tidsplaner og forventninger, men testmanageren skal også forstå at testgruppen også føler dette pres. Når testmanageren skaffer nye folk til gruppen, er det vigtigt at overveje arbejdsmiljøet og at matche personlighedstypen med miljøet. I et miljø hvor der er mere arbejde end tiden tillader, bør testmanageren søge efter folk, som afslutter deres arbejde og selv spørger hvad de kan gå i gang med, som det næste.

Den enkelte arbejder mere omhyggeligt hvis han føler sig værdsat og oplever at hans indsats er nødvendig. Alle medarbejdere må forstå at de hver især er vigtige medlemmer af gruppen og at den fælles indsats er afgørende for gruppens succes. Når denne dynamik opstår, vil medlemmerne træne hinanden gensidigt, gruppen vil selv fordele sine arbejdsopgaver, og testmanageren får tid til at håndtere eksterne problemer.

Nye gruppemedlemmer skal hurtigt optages i gruppen og få den tilstrækkelige vejledning og støtte. Hver person bør tildeles en defineret rolle i gruppen. Rollen kan baseres på en individuel vurderingsproces. Målet er at give hver enkeltperson en succesoplevelse, samtidig med at han bidrager til gruppens overordnede succes. Dette sker i det store og hele ved at matche personlighedstyper med gruppens roller og bygge på den enkeltes medfødte færdigheder, samtidig med at man forøger deres samlede sæt af færdigheder.

Når man skal beslutte hvem man vil ansætte eller tilknytte testgruppen, kan det være en hjælp at have en objektiv vurdering af færdigheder. Man kan vurdere ved hjælp af interviews, test af kandidater, gennemgang af arbejdseksempler og ved at checke referencer. Man bør vurdere følgende færdigheder:

Tekniske færdigheder ("hårde" færdigheder) kan afprøves ved at:

- Udlede testcases fra et kravdokument
- Reviewe teknisk dokumentation (krav, kode osv.)
- Skrive reviewkommentarer på en klar, forståelig og objektiv måde
- Anvende diverse testteknikker korrekt i forskellige scenarier (f.eks. anvende beslutningstabeller til at teste et sæt forretningsregler)
- Vurdere en afvigelse og præcist dokumentere den
- Demonstrere en forståelse af klassifikation af defektinformation
- Demonstrere en forståelse af den underliggende årsag til defekter
- Anvende et værktøj til test af et givet API, inklusiv positive og negative test
- Anvende SQL til at finde og opdatere information i en database, for at teste et givet scenarie
- Designe stilladser til testautomatisering som vil eksekvere flere test og indsamle testresultater
- Afvikle automatiserede test og fejlfinde afvigelser
- Skrive testplaner/specifikationer
- Skrive en testopsummeringsrapport som inkluderer en vurdering af testresultaterne.

Menneskelige færdigheder ("bløde" færdigheder) kan afprøves ved at:

- Præsentere information vedrørende et forsinket testprojekt
- Fremlægge en fejlrapport for en udvikler som ikke mener at der er tale om en fejl

- Træne en kollega
- Præsentere et problem for ledelsen om en ineffektiv proces
- Reviewe en kollegas testcase og præsentere kommentarerne for kollegaen
- Interviewe en kollega
- Give en udvikler et kompliment.

Selv om dette ikke er en komplet liste og selv om de ønskede færdigheder vil variere mellem miljøer og organisationer, er det en liste af færdigheder som normalt behøves. Ved at udarbejde effektive interviewspørgsmål og afprøve færdigheder, kan testmanageren vurdere kandidatens færdigheder og fastslå stærke og svage områder. Når først et godt sæt af spørgsmål er lavet kan det også anvendes på alle eksisterende ansatte, for at fastslå områder der bør forbedres og trænes.

Udover færdigheder som hver enkelt i gruppen behøver, skal testmanageren også have evne for kommunikation og sans for diplomati. Testmanageren skal også kunne beherske kontroversielle situationer, vide hvilke kanaler der skal anvendes for den nødvendige kommunikation, og kunne opbygge og vedligeholde relationer inden for organisationen.

7.4 Testgruppen i den større organisation

Organisationer kan indpasse test i den organisatoriske struktur på mange måder. Selv om kvalitet i softwareudviklingens livscyklus er alles ansvar, kan en uafhængig testgruppe i høj grad bidrage til et kvalitetsprodukt. I praksis varierer testfunktionens uafhængighed meget. Det kan ses af de følgende lister, der er ordnet fra mindste til største uafhængighed:

- Ingen uafhængige testere
  - I dette tilfælde er der ingen uafhængighed. Koden testes af den udvikler som skrev koden
  - Hvis udvikleren får tid til at foretage testen, vil han/hun afgøre, om koden fungerer efter hensigten, hvilket muligvis og muligvis ikke opfylder de faktiske krav
  - Udvikleren kan hurtigt rette alle fundne defekter.

- Test foretages af en en anden udvikler end den der skrev koden
  - Der er lille uafhængighed mellem udvikleren og testeren
  - En udvikler, der tester en anden udviklers kode, kan være tilbageholdende med at rapportere defekter
  - En udvikler er normalt indstillet på testcases med et positivt resultat.

- Test foretages af en tester (eller en testgruppe), der er en del af udviklingsgruppen
  - Testeren (eller testgruppen) kan enten rapportere til projektledelsen eller til lederen af udviklingsgruppen
  - Testeren er mest indstillet på at kontrollere, at kravene er overholdt
  - Da testeren er medlem af udviklingsgruppen, kan testeren have udviklingsansvar ud over testansvar
  - Testerens leder kan være mere fokuseret på at overholde tidsplaner end at opnå kvalitetsmål
  - Test kan have lavere status end udvikling i gruppen
  - Testeren har måske ikke autoritet til at påvirke kvalitetsmålene eller om disse mål opnås.

- Test foretages af testspecialister fra forretningsorganisationen, brugergruppen eller andre tekniske organisationer, der ikke beskæftiger sig med udvikling
  - Information om testresultater rapporteres objektivt til interessenter

  - Denne gruppes primære fokus er softwarekvaliteten
  - Udvikling af færdigheder og uddannelse er fokuseret på test
  - Test ses som en karriere
  - Testgruppen har er en særskilt ledelse som fokuserer på kvalitet.

- Eksterne testspecialister udfører test af specifikke testtyper
  - Ekspertise anvendes på specifikke områder hvor generel test højst sandsynlig ikke er tilstrækkelig
  - Testen kan dreje sig om brugervenlighed, datasikkerhed, performance eller andre områder, hvor det er nødvendigt at være specialiseret
  - Kvalitet bør være i fokus for disse enkeltpersoner men deres syn er begrænset til deres ekspertiseområde. Et produkt som har en god performance opfylder nødvendigvis ikke alle funktionelle krav, men det opdages ikke altid af performancespecialisten.

- Test foretages af en organisation, der er ekstern i forhold til virksomheden
  - Med denne model opnås maksimal uafhængighed mellem testeren og udvikleren
  - Testeren kan muligvis savne tilstrækkelig viden til at kunne teste på en kompetent måde
  - Der er behov for klare krav og en veldefineret kommunikationsstruktur
  - Kvaliteten af den eksterne organisation skal vurderes med jævne mellemrum.

Denne liste er baseret på individers typiske fokus men kan være forkert i den enkelte organisation. En stilling og en jobtitel bestemmer ikke nødvendigvis en persons fokus. Udviklingsledere kan være kvalitetsfokuserede og kan derfor være gode testmanagere. Uafhængige testmanagere kan rapportere til en række af ledere fokuseret på tidsplaner og kan derfor være mere fokuseret på tidsplaner end kvalitet. For at bestemme den bedste placering af testgruppen i en organisation, skal testmanageren forstå organisationens mål.

Der er varierende grader af uafhængighed mellem udviklings- og testgrupperne. Det er vigtigt at forstå, at der kan blive tale om et kompromis, hvor mere uafhængighed vil give mere isolation og mindre vidensoverførsel. Et lavere uafhængighedsniveau kan give forøget viden, men det kan også medføre mål, der er i konflikt med hinanden. Uafhængighedsniveauet bestemmes også af den softwareudviklingsmodel, der anvendes, f.eks. er testerne oftest en del af udviklingsgruppen ved agil udvikling.

Alle de ovenfor nævnte muligheder kan være til stede i en organisation. Der testes måske både i udviklingsorganisationen og i en uafhængig testgruppe, og der sker eventuelt en endelig certificering i en ekstern organisation. Det er vigtigt at forstå ansvarsfordelingen og forventningerne for hver testfase og at sætte disse krav, så kvaliteten af det færdige produkt maksimeres, samtidig med at arbejdet holdes inden for tidsplanen og budgetbegrænsninger.

7.5 Motivation af testerne

En person, der arbejder med test, kan motiveres på mange måder. Disse omfatter bl.a.:

- Anerkendelse for udført arbejde
- Ros fra ledelsen
- Respekt i projektgruppen og blandt kollegaer
- Øget ansvar og uafhængighed
- Tilstrækkelig honorering for det foretagne arbejde (bl.a. aflønning, øget ansvar, anerkendelse).

Der findes påvirkninger fra projektet, der kan gøre det vanskeligt at anvende disse motiveringsværktøjer. En tester kan f.eks. arbejde hårdt på et projekt, der har en umulig deadline. Testeren kan gøre alt for at styre gruppens fokus på kvalitet, arbejde ekstra timer og yde en ekstra arbejdsindsats, men alligevel kan den overordnede ledelse beslutte at produktet skal leveres før kvaliteten er tilfredsstillende. Resultatet vil så være et dårligt produkt på trods af, at testeren har ydet sit bedste. Det kan virke demotiverende. Det er derfor vigtigt at anerkende testernes indsats, uanset om slutproduktet er en succes eller ej.

Testgruppen skal sikre, at den registrerer de rigtige metrikker for at vise det arbejde der er gjort for at gennemføre testen, afbøde risici og registrere resultaterne nøjagtigt. Det er vigtigt at disse data skal indsamles og vises. I modsat fald er det let for testgruppen at blive demotiveret, fordi den ikke modtager anerkendelse for et veludført arbejde.

Anerkendelse bestemmes ikke kun af de uhåndgribelige begreber respekt og godkendelse, det er også synligt i forfremmelsesmuligheder, øget ansvar og øget værdi. Hvis testgruppen ikke bliver respekteret, er disse muligheder måske ikke til stede.

Anerkendelse og respekt opnås, når det er klart, at testeren bidrager til projektets stigende værdi. I det enkelte projekt opnås dette bedst ved at involvere testeren helt fra begyndelsen af projektet og ved at bevare involveringen igennem livscyklussen. Som tiden går, vil testerne vinde de andre projektdeltageres respekt for deres positive bidrag til projektet. Dette bidrag bør også kvantificeres som omkostningerne ved kvalitetsreduktioner og risikoreduktion.

Testmanageren spiller en vigtig rolle i at motivere de enkelte medlemmer i testgruppen samt at optræde som en fortaler for testgruppen i en større organisationssammenhæng. Testmanageren bør anerkende de enkelte testeres indsats og skal også give en fair og ærlig tilbagemelding på fejltagelser. En fair og konsekvent testmanager vil motivere gruppens medlemmer ved sit eksempel.

7.6 Kommunikation

Testgruppe-kommunikation sker primært i forbindelse med:

- Dokumentation af testprodukter: teststrategi, testplan, testcases, testopsummeringsrapporter, hændelsesrapporter osv.
- Feedback på reviewede dokumenter: krav, funktionelle specifikationer, usecases, komponenttestdokumentation osv.
- Informationsindsamling og -spredning: samspil med udviklere, andre testgruppemedlemmer, ledelsen osv.

Kommunikationen mellem testere og andre interessenter skal være professionel, objektiv og effektiv for at kunne opbygge og vedligeholde respekten for testgruppen. Der kræves diplomati og objektivitet i forbindelse med feedback, specielt konstruktiv feedback, på andres arbejdsprodukter. Kommunikationen må være fokuseret på at nå testmålene og på at forbedre kvaliteten i både produkter og de processer som anvendes for at lave softwaresystemerne.

Testmanagere kommunikerer med et bredt publikum inklusiv brugere, medlemmer af projektgruppen, ledelsen, eksterne testgrupper og kunder. Kommunikationen skal være effektiv i forhold til målgruppen. For eksempel kan en rapport lavet til udviklingsgruppen, som viser trend og tendenser i antallet og alvoren af fundne defekter, være for detaljeret til at passe til et informationsmøde med topledelsen. I al kommunikation, men især præsentationer, er det vigtigt at forstå beskeden som afsendes, måder beskeden vil kunne blive modtaget, og de forklaringer som er nødvendige for at skabe de bedste rammer for at beskeden bliver accepteret. Da testmanageren ofte præsenterer information om projektstatus, er det vigtigt at denne information har det rette detaljeringsniveau (f.eks.

vil ledere som regel gerne se trends i defekter fremfor enkelte defekter) og at informationen bliver præsenteret på en klar og letforståelig måde (f.eks. simple diagrammer og grafer i farver). Egnet kommunikation gør det lettere at fastholde tilhørernes opmærksomhed, samtidig med at den rigtige besked leveres. Testmanageren bør se hver eneste præsentation som en mulighed for at fremme kvalitet og kvalitetsprocesser.

En testmanager kommunikerer ikke kun med folk uden for afdelingen (udadgående kommunikation). En vigtig del af en testmanagers arbejde er, at kommunikere effektivt inden for testgruppen (indadgående kommunikation) ved at videregive nyheder, instruktioner, ændringer i prioriteter samt anden standard information, som viderebringes i en normal testproces. En testmanager kan også kommunikere til specifikke personer både op (opadgående kommunikation) og ned (nedadgående kommunikation) i ledelseshierarkiet i organisationen. De samme regler gælder uanset kommunikationens retning. Kommunikationen skal være tilpasset tilhørerne, beskeden skal sendes effektivt og forståelse af beskeden skal bekræftes.

Testmanagere skal kunne håndtere forskellige måder at kommunikere på. Megen information kommunikeres via e-mail, mundligt, på formelle eller uformelle møder, i formelle eller uformelle rapporter og i testledelsesværktøjer, f.eks. defekthåndteringsværktøjer. Al kommunikation skal være professionel og objektiv. Skriftlig kommunikation er ofte tilgængelig lang tid efter selve projektet er afsluttet. Det er vigtig for testmanageren at lave professionel kvalitetsdokumentation som repræsenterer en organisation som promoverer kvalitet.

8 Referencer

8.1 Standarder

- [BS7925-2] BS 7925-2, Software Component Testing Standard.
  Kapitel 2
- [FDA21] FDA Title 21 CFR Part 820, Food and Drug Administration, USA
  Kapitel 2
- [IEEE829] IEEE Standard for Software and System Test Documentation
  Kapitel 2 og 4
- [IEEE1028] IEEE Standard for Software Reviews and Audits
  Kapitel 2
- [IEEE1044] IEEE Standard Classification for Software Anomalies
  Kapitel 4
- [ISO9126] ISO/IEC 9126-1:2001, Software Engineering - Software Product Quality,
  Kapitel 2 og 4
- [ISO12207] ISO 12207, Systems and Software Engineering, Software Life Cycle Processes
  Kapitel 2
- [ISO15504] ISO/IEC 15504, Information Technology - Process Assessment
  Kapitel 2
- [ISO25000] ISO/IEC 25000:2005, Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE)
  Kapitel 2 og 4
- [RTCA DO-178B/ED-12B]: Software Considerations in Airborne Systems and Equipment Certification, RTCA/EUROCAE ED12B.1992.
  Kapitel 2

8.2 ISTQB-dokumenter

- [ISTQB_AL_OVIEW] ISTQB Advanced Level Overview, Version 1.0
- [ISTQB_ALTM_SYL] ISTQB Advanced Level Test Manager Syllabus, Version 1.0
- [ISTQB_ALTTA_SYL] ISTQB Advanced Level Technical Test Analyst Syllabus, Version 1.0
- [ISTQB ETM SYL] ISTQB Expert Test Management Syllabus, Version 1.0
- [ISTQB_FL_SYL] ISTQB Foundation Level Syllabus, Version 2011
- [ISTQB_GLOSSARY] Standard glossary of terms used in Software Testing, Version 2.2, 2012
- [ISTQB ITP SYL] ISTQB Expert Improving the Test Process Syllabus, Version 1.0

8.3 Varemærker

De følgende registrerede varemærker og servicemærker bruges i dette dokument:

CMMI®, IDEAL℠, ISTQB®, ITIL®, PRINCE2®, TMMi®, TPI Next®

- CMMI er registreret i US Patent and varemærke kontor ved Carnegie Mellon University.
- IDEAL er et servicemærke af Software Engineering Institute (SEI), Carnegie Mellon University
- ISTQB er et registreret varemærke af den Internationale Software Testing Qualifications Board
- ITIL er et registreret varemærke, og et registreret EU mærke af Office of Government Commerce, og er registreret i U. S. Patent and varemærke kontor
- PRINCE2 er et registreret varemærke af Kabinet kontoret
- TMMi er et registreret varemærke af TMMi Foundation
- TPI Next er et registreret varemærke af Sogeti Nederland B.V.

8.4 Bøger

[Black03]: Rex Black, “Critical Testing Processes,” Addison-Wesley, 2003, ISBN 0-201-74868-1

[Black09]: Rex Black, “Managing the Testing Process, third edition,” John Wiley & Sons, 2009, ISBN 0-471-22398-0

[Black11]: Rex Black, Erik van Veenendaal, Dorothy Graham, “Foundations of Software Testing,” Thomson Learning, 2011, ISBN 978-1-84480-355-2

[Craig02]: Rick Craig, Stefan Jaskiel, “Systematic Software Testing,” Artech House, 2002, ISBN 1-580-53508-9

[Crispin09]: Lisa Crispin, Janet Gregory, “Agile Testing: A Practical Guide for Testers and Agile Teams”, Addison-Wesley, 2009, ISBN 0-321-53446-8

[de Vries09]: Gerrit de Vries, et al., “TPI Next”, UTN Publishers, 2009, ISBN 90-72194-97-7

[Goucher09]: Adam Goucher, Tim Riley (editors), “Beautiful Testing,” O'Reilly, 2009, ISBN 978-0596159818

[IDEAL96]: Bob McFeeley, “IDEAL: A User's Guide for Software Process Improvement,” Software Engineering Institute (SEI), 1996, CMU/SEI-96-HB-001

[Jones07]: Capers Jones, “Estimating Software Costs, second edition,” McGraw-Hill, 2007, ISBN 978-0071483001

[Jones11]: Capers Jones and Olivier Bonsignour, “Economics of Software Quality,” Pearson, 2011, ISBN 978-0132582209

[McKay07]: Judy McKay, “Managing the Test People,” Rocky Nook, 2007, ISBN 978-1933952123

[Musa04]: John Musa, “Software Reliability Engineering, second edition,” Author House, 2004, ISBN 978-1418493882

[Stamatis03]: D.H. Stamatis, “Failure Mode and Effect Analysis,” ASQC Quality Press, 2003, ISBN 0-873-89300

[vanVeenendaal11] Erik van Veenendaal, “The Little TMMi,” UTN Publishers, 2011, ISBN 9-490-986038

[vanVeenendaal12] Erik van Veenendaal, “Practical Risk-based Testing,” UTN Publishers, 2012, ISBN 978-9490986070

[Whittaker09]: James Whittaker, “Exploratory Testing,” Addison-Wesley, 2009, ISBN 978-0321636416

[Wiegers03]: Karl Wiegers, “Software Requirements 2,” Microsoft Press, 2003, ISBN 978-0735618794

8.5 Andre referencer

De følgende referencer peger på tilgængelige oplysninger på internettet. Disse referencer blev kontrolleret på tidspunktet for offentliggørelse af dette pensum for Advanced Test Manager.

http://www.istqb.org
http://www.sei.cmu.edu/cmmi/
http://www.tmmi.org/
http://www.tpinext.com/