Ian Mãnebo Frost

Erhvervsakademiet sjælland, roskilde | 4 semester eksamen 2019

Exploratory Testing

Ved brug af microsoft test manager

Indhold

[Indledning 2](#_Toc532600228)

[Motivation 2](#_Toc532600229)

[Problemstilling 2](#_Toc532600230)

[Metode 2](#_Toc532600231)

[Planlægning 3](#_Toc532600232)

[Hvad er Exploratory Testing? 3](#_Toc532600233)

[Microsoft Test Manager 5](#_Toc532600234)

[Microsoft Test Manager Struktur 5](#_Toc532600235)

[Exploratory Testing funktion 6](#_Toc532600236)

[Hvornår ville man vælge Exploratory Testing? 7](#_Toc532600237)

[Konklusion 8](#_Toc532600238)

[Refleksion 8](#_Toc532600239)

[Litteraturliste 9](#_Toc532600240)

[Hjemmesider: 9](#_Toc532600241)

[Videoer: 9](#_Toc532600242)

# Indledning

Mange IT-systemer bliver testet vha. automatiseret testværktøjer. Vi har igennem vores studieforløb stiftet bekendtskab med bl.a. Selenium som tester GUI, værktøjer som disse er yderst effektive da de er automatiseret hvilket resulterer i mindre udgifter og det er mindre tidskrævende. Men selvom automatiseret tests er populære, er de stadig ikke perfekte, de kan ikke skelne mellem de helt små detaljer, som kun det menneskelige har sans for. Jeg har valgt at sætte mit fokus på Exploratory Testing da det netop omhandler at teste systemet manuelt som bruger.

# Motivation

Da jeg arbejdede med mit 2. semester projekt lavede vi hele tiden ændringer i programmet for at få det til at virke, fra de mindste bugs til de største bugs.   
Vi fulgte Test Driven Development metoden, hvor vi havde lavet use cases og skulle derfra teste de enkelte metoder, hvilket viste sig som værende meget slavisk arbejde.  
Jeg stiftede bekendtskab med exploratory testing gennem min vejleder Mohammed, og efter lidt research på nettet da så min første youtube video, vidste jeg at det var et emne jeg gerne ville gøre mig i til eksamen.   
Exploratory testing adskiller sig fra hvad jeg har lært indtil videre under datamatiker uddannelsen. Som studerende ønsker jeg at få så bred en viden indenfor testing da jeg ønsker at gøre en karriere indenfor området når jeg er færdig uddannelsen.

# Problemstilling

* Hvad er Exploratory Testing?
* Hvordan udfører MTM Exploratory Testing og hvilke værktøjer kan man bruge?
* Hvornår ville man vælge Exploratory Testing?

# Metode

For at udarbejde den bedst mulige synopsis har jeg valgt at benytte mig af følgende metoder:

**Research** – Læsning og forståelse af relevant viden indenfor mit emne, dette vil primært foregå igennem nettet og videoer

**Udførelse –**  Udføre samt relatere viden overfor praksis. Jeg har valgt at benytte mig af programmet Microsoft Test Manager for at stifte bekendtskab med et værktøj som giver mig en bedre forståelse af Exploratory Testing. Jeg vil følge en guide på nettet for hvordan en virksomhed vil benytte sig af det, så jeg får den bedst mulige forståelse. Dette vil også være en del af min fremlæggelse til den mundtlige eksamen

**Synopsis skrivning –** Dokumentere min proces. Her bliver der lagt fokus på det mest relevante og vigtigste af hvad jeg har lavet research om. Dette gøres for at give læren og censoren den tilstrækkelig viden for at kunne forstå emnet samt forberede potentielle spørgsmål til eksamen.

# Planlægning

Jeg har til sinde i dette afsnit valgt at give et visuelt billede af hvordan jeg har planlagt mit forløb med dette projekt.   
Som ses på nedenstående billede har jeg skrevet sideløbende på min rapport mens jeg har lavet research og benyttet mig af Microsoft Test Manager, da jeg mente det var det bedste at gøre for at få den bedste dokumentering.



# Hvad er Exploratory Testing?

Når man benytter sig af Exploratory Testing er fokusområdet lagt på selvlæring og frigørelse. Når man skal teste skal der tænkes ud af boksen og ikke følge testcases som er skrevet på forhånd med et formål. Når en tester følger Exploratory Testing benytter han sig af programmet, finder ud af hvilke potentielle fejl og bugs der kunne opstå, det er derfor vigtigt at man som tester benytter programmet som en klient ville gøre.

For at reflektere mit emne over noget af vores undervisning har jeg valgt at snakke om dens placering i den Agile Testing Quadrant. Den Agile Testing Quadrant er model der kategoriserer forskellige test metoder op i forskellige dele. Det er med til at skabe overblik og se hvilken værdi den pågældende metode har for teamet og produktet.

Hvis man ser på Exploratory Testings placering i forhold til den Agile Testing Quadrant, er den placeret i Q3. Q3 er området, hvor alt forretning logik starter, det er her man laver sine traditionelle user acceptance tests, dette vil altså sige at Exploratory Testing ikke blot udføres af én aktør, men man tilhører et team som snakker sammen på daglig basis, hvor man skal dokumentere hvad der bliver testet, da det er med til at skabe feedback og gøre produktet bedre. Eftersom det er testeren selv, som skal gennemgå systemet er det også oplagt at det bliver beskrevet som værende manuelt testing.

Som tester har man på forhånd lagt et område man vil gå ind i, et eksempel på dette kunne være følgende; et betalingsområde for en hjemmeside, hvor man skal tænke på mulige fejl, så vores tester har købt en vare på hjemmesiden, nu skal han indtaste sine personlige oplysninger og kommer ved en fejl til at indtaste et postnummer på 5 cifre eller et postnummer på 4 cifre men som ikke eksisterer i DK. Dette bliver godtaget alligevel, testeren notér fejlen og rapportere til teamet. Dette vil altså sige at man er ikke fastlåst på en specifik arbejdsopgave, men et område.

Exploratory Testing adskiller sig derfor meget fra scripted testing som jeg har arbejdet med under mit 2. semester projekt, her er der til formål kun at teste det som test cases beskriver.

Ifølge flere eksperter som benytter sig af Exploratory Testing følger proceduren fem stadier som er følgende: [[1]](#footnote-1)

1. **Bug kategorisering** - Først kategoriseres bugs i forhold til hvor vigtige de kan være for programmet
2. **Test Charter** - Herefter bliver der skabt en Test Charter, som skitserer områder man skal teste indenfor
3. **Time box** -Testerne år sammen i parvis og arbejder intensivt op til 90 minutter, hvor man kører alle scenarier igennem. Dette kaldes for timebox
4. **Result review** - Her tjekkes der for hvilke resultater man har fået, og hvilken læring man har fået ud af programmet
5. **Debriefing** - Her briefes der i teams, om de fejl man har fundet har stemt overens med de test charter man har skabt tidligere, samt om der er brug for yderligere tests.

Der er forskellige værktøjer som er tilgængelige for at benytte sig af Exploratory Testing. Jeg har givet to eksempler på værktøjer udover det jeg vil benytte mig af.

**Tosca Tricentis -** Tosca Tricentis benytter sig af samme funktioner som MTM, her kan der skabes test cases til automatiske test funktioner samt exploratory testing, her der også lagt fokus på at finde bugs og skabe konstant feedback for at få det bedste produkt Tosca Tricentis har til forskel for MTM at det er en abonnement man skal subscribe på, for at få adgang til deres produkter, hvor jeg har gennem mit studie adgang til alle microsofts produkter i Imagine. Tricentis beskæftiger sig ikke kun med exploratory testing, deres produkter strækker sig bredt indenfor programmering, de har værktøjer fra web API til mobile programmering[[2]](#footnote-2). Tricentis tilbyder ydermere også et træningsforløb med programmet, hvor man har til formål at blive specialist med Exploratory Testing, dette munder ud i en online eksamen hvor man kan få et bevis og certifikat, der kan vedhæftes på ens CV.

**Azure Test and Feedback extension –** Azure har udgivet et udvidelsesprogram til Azure DevOps, hvilket vil sige at man skal benytte sig af Azure DevOps for at kunne gøre brug af deres udvidelsesprogram. Det smarte ved dette er, du ikke skal have to programmer åbne for at se hvilke bugs man har fundet. Når man laver Exploratory Testing i dette program, optager den din skærm og alt hvad du klikker på, du kan endda indsætte screenshots alt efter behov, når dette er færdigt bliver det direkte sendt over til Azure DevOps. Men eftersom jeg ikke har kendskab til Azure DevOps har jeg ikke valgt at benytte mig af deres produkter.

Jeg vil tage udgangspunkt i Microsoft Test Manager, eftersom vi igennem Datamatiker studiet har stiftet bekendtskaber med andre Microsoft produkter, herunder Visual Studio og Visio, derfor følte jeg det lå meget nærliggende. En anden faktor som gjorde sig gældende for mit valg, var informationen til rådig på nettet om hvordan det skulle sættes op og hvor dybdegående informationen var.   
Jeg har derfor i løbet af min synopsis valgt at give en kort introduktion til hvad MTM er og hvordan en session[[3]](#footnote-3) udføres.

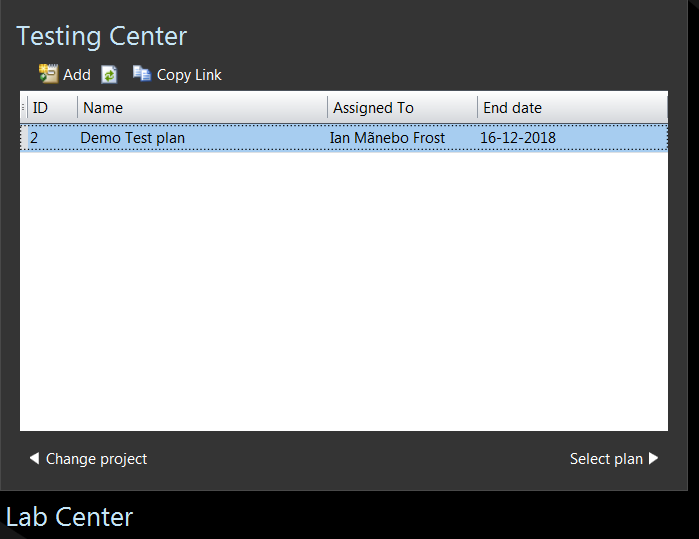
# Microsoft Test Manager

Microsoft udbyder en masse programmerings produkter og da testing indgår i softwareudvikling har de udviklet deres eget program, Microsoft Test Manager, til firmaer som ønsker at teste i et agilt miljø.

Microsoft Test Manager har primær til årsag at dokumentere den manuelle testers proces igennem systemet, ved hjælp af dets mange funktioner gør det arbejdet nemmere for testerens udviklingsteam at forstå hvor langt man er, samt hvilke opgaver der er fundet frem til.

## Microsoft Test Manager Struktur

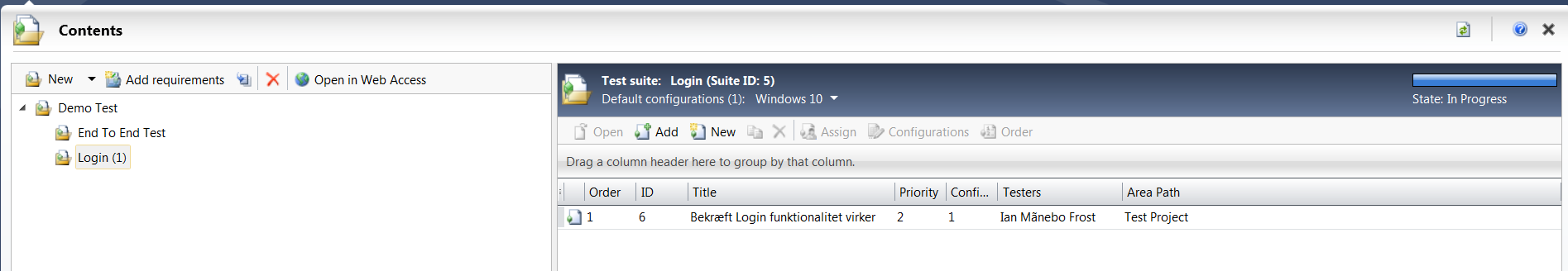
Microsoft Test Manager er forbundet til ens Visual Studio profil, der besidder offentliggjorte projekter. Første skridt er at vælge det specifikke projekt man ønsker at skabe sin tests til, som ses på ovenstående billede har jeg skabt et helt nyt projekt kaldet ”Test Project”.



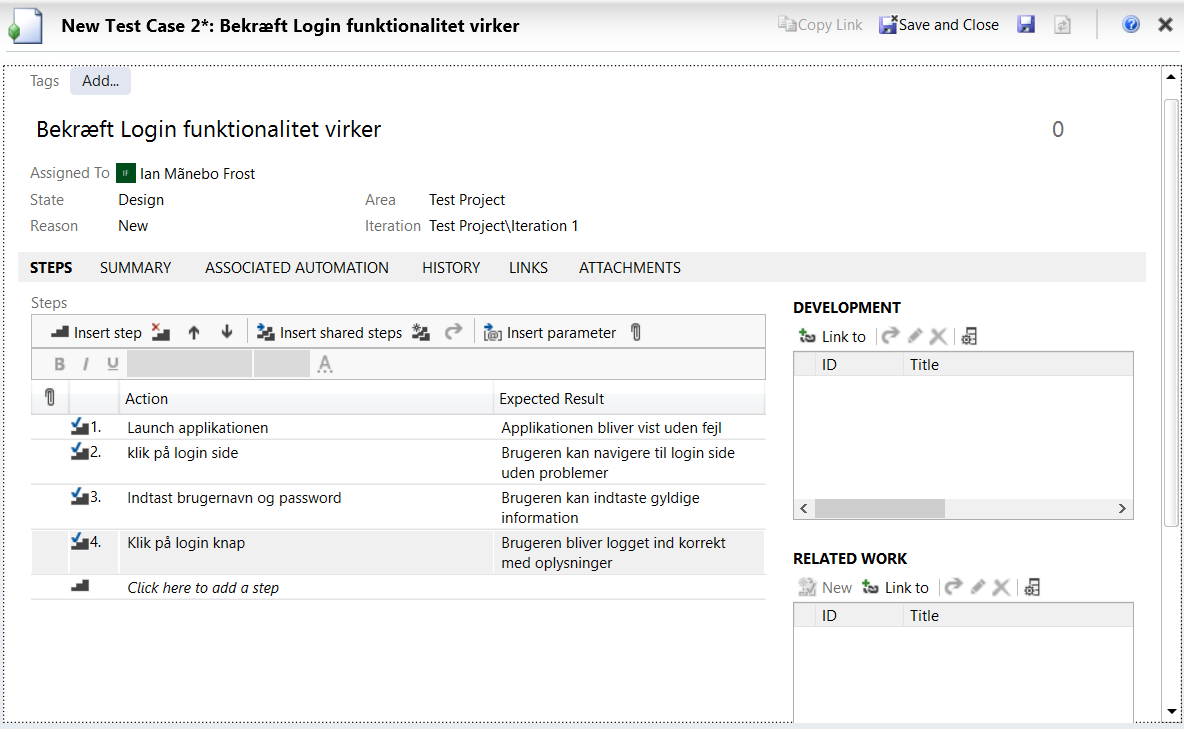
Efterfølgende skal der skabes en test plan. En test plan bliver delt op i tre kategorier: Et navn, hvilket område det kan dække og hvilken iteration den tilhører. Jeg har på nedenstående billede skabt et projekt der opfylder disse tre kriterier.

Test plan er stedet man specificer de områder ens test cases skal dækkes. Det er her test teamet kan se hvilke der skal laves. En test plan indeholder suites, som er det overordnede område hvori de forskellige test cases ligger, her er der også beskrevet prioritering af test case, samt hvor mange ændringer der er blevet lavet på test casen. Ved hjælp af status proces baren i højre hjørne, kan man se om det er en aktiv suite (blå farve) eller en færdig suite (rød farve).

Her er det nemt og overskueligt for testeren at se indenfor hvilket område personen skal teste. Testeren kan enten tilføje bugs indenfor visse områder, eller blot udføre en af test cases.



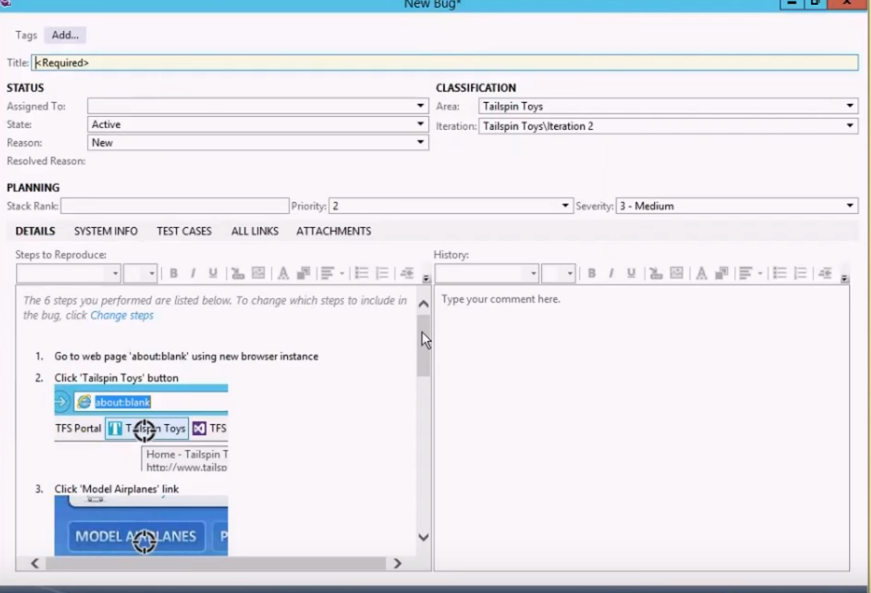
En test case i MTM består af en beskrivelse og et forventet resultat som ses på nedenstående billede



## Exploratory Testing funktion

Som tidligere beskrevet i synopsen, er frihed essentielt for at kunne benytte sig af Exploratory Testing. Frihed er en nødvendighed når man som tester skal anvende systemet som en bruger ville gøre.   
Microsoft Test Manager indeholder en funktion specifikt henvendt til Exploratory Testing. Når man benytter sig af den funktionalitet, forsvinder ens store klodsede Test Manager program og ind i stedet optræder en lille applikation, hvor man dokumenterer ens forløb med funktionen.   
 Exploratory mode optager i baggrunden hvilke funktioner man benytter sig af, af den pågældende applikation, herefter kan der oprettes en bug, og ved hjælp af optagelserne opretter MTM selv steps ud fra hvilke handlinger man har gjort sig brug af.   
 Udover at kunne optage kan der også tages screenshots, for at gøre det visuelt nemmere for testeren at dokumentere fejlen.   
Alt dette skal munde ud i en test case, der er autogeneret ud fra de optagelser fra exploratory funktionen, og som er veldokumenteret for udvikleren, så han konkret ved hvilke fejl der skal kigges på og gøres ved.

På nedenstående billede ses exploratory funktionen i efter den har optaget en session og ens steps er blevet autogeneraliseret



## Hvornår ville man vælge Exploratory Testing?

Jeg har i sinde at undlade at svare på dette spørgsmål i min synopsis, da jeg ønsker at fremlægge min teori om, hvornår man benytter sig af exploratory testing samt hvilke fordele og ulemper det har, til eksamen.

## Konklusion

Jeg har igennem dette forløb lært at exploratory testing handler om at tænke ud af boksen og tænke på de fejl en automatiseret test ikke kan gennemskue. Exploratory Testing bygger på at teste indenfor et givet område men ikke en specifik case, hvilket er meget essentielt at man har friheden indenfor et område. For at kunne udføre Exploratory Testing er der udgivet forskellige udviklingsprogrammer, jeg gjorde mig bekendt med Microsoft Test Manager da vi har arbejdet med deres produkter ud igennem uddannelsen. I MTM er der en funktion som er specifik for exploraytory testing, den opfylder de 5 faser; bug kategorisering, test charter, time box, result reviewing og debriefing. Ved at gennemføre disse 5 faser har man udført exploratory testing, men det er op til teamet selv at afgøre hvornår man føler man har testet produktet nok igennem.

## Refleksion

Dette afsnit har til formål at dokumentere de tanker jeg har skabt mig over hvad jeg kunne have gjort anderledes, samt om jeg har fået det udbytte som jeg havde forventet jeg ville få, under mit forløb med denne opgave.   
**Planlægning:** Jeg startede forløbet op med at planlægge et skema som skulle diktere hvordan jeg skulle udnytte min tid bedst muligt. Jeg har efterfølgende fundet ud af at det ikke er en metode jeg vil følge i samme stil til fremtidige projekter, da jeg følte mig fastlagt og det hele skulle gå en smule hurtigere med dokumentering og forståelse for at kunne overholde tidsplanen. Dette fandt jeg hurtigt ud af ikke var ideelt og har derfor ændret i min tidsplan for at få den til at passe til mine behov.   
Min ide med at dokumentere løbende mens jeg gjorde mig bekendt med MTM har været en positiv ting, da jeg føler at det har hjulpet på min dokumentering.

**Automatisering:** Havde min synopsis skulle indeholde noget yderligere ville jeg gerne gå i dybden med hvordan man vha. MTM kan lave automatiserede tests og hvordan de virker, da jeg føler at det er relevant når jeg allerede omtaler MTM.

**Spørgeskemaer:** Jeg kunne havde henvendt mig eksperter og firmaer for at høre hvilke testmetoder de benytter sig af, og høre hvilke erfaringer de har gjort sig med exploratory testing.

# Litteraturliste

## Hjemmesider:

<https://www.guru99.com/exploratory-testing.html>

<https://www.softwaretestinghelp.com/tools/top-17-exploratory-testing-tools/>

<https://lisacrispin.com/2011/11/08/using-the-agile-testing-quadrants/>

<https://www.tricentis.com/resource-assets/exploratory-testing/>

<https://marketplace.atlassian.com/apps/1216562/exploratory-testing-for-jira?hosting=server&tab=overview>

<https://www.techarcis.com/exploratory-testing-should-be-performed-when-why/>

## Videoer:

<https://www.youtube.com/watch?v=_G7-gAaoGv0>

<https://www.youtube.com/watch?v=T5gV3C_AQ74>

<https://www.youtube.com/watch?v=O_FOQJsQx4k>

1. <https://www.techarcis.com/exploratory-testing-should-be-performed-when-why/> [↑](#footnote-ref-1)
2. <https://www.tricentis.com/software-testing-tools/> [↑](#footnote-ref-2)
3. En test case udførelse bliver betegnet som en session [↑](#footnote-ref-3)