# Opgaver til test og kvalitetssikring

Herunder finder du en række praktiske opgaver/øvelser, som kan benyttes til test og kvalitetssikring.

## Øvelse 1 – Review – 1 time

Hvis man udvikler efter fasemodellen, produceres der i hver fase dokumentation (eksempelvis en kravspecifikation eller en designbeskrivelse), som kan afsluttes med et review. Man kan også benytte reviewet som feedback på en hel opgave, ved at lade eleverne reviewe hinandens materialer. Slutteligt kan underviseren vælge at udlevere et dokument, som eleverne skal finde fejl i.

Det er vigtigt at formalismen holdes.

Rollerne fordeles. Der skal være:

* En reviewleder
* En forfatter
* En referent (kan være forfatter)
* Reviewere

Forberedelse: Mødet indkaldes og dokumentation udleveres. Alle læser det igennem inden mødet.

Udførelse: Dokumentet gennemgås afsnit for afsnit. Problemer noteres ned. Løsninger nå godt diskuteres, men kun hvis der er tid. (styres af reviewlederen)

Hvis det er et godkendelsesreview tages der stilling til, om dokumentet kan godkendes.

Efter reviewet: Der rettes op på problemerne i dokumentet

## Øvelse 2 – Debugging

Som eksempel på debugging kan vi kigge på ”debugging tools” i App Lab, samt gøre brug af koden fra kasseapp’en. Debug vinduet er nederst i App Lab. Du kan sætte hastigheden for afviklingen ned på slideren, og trykke ”Break” for at stoppe afviklingen. Du kan også indsætte et ”break point” i koden ved at trykke på den linje, du vil stoppe ved (giver en blå markering). Du kan manuelt køre programmet videre ved at trykke på step knapperne, eller du kan udskrive værdien af en variabel i konsolvinduet. (Se eksempel her Link til video: Kasseapp i applab – 20:26 Test og debugging). Du kan selv prøve at finde fejl i din app (se øvelse 3), og derefter gennemføre debugging.

## Øvelse 3 – Accepttest og problemrapportering

Da accepttesten tager sit udgangspunkt i en specifikation, er testen mest relevant, når man arbejder efter en fasemodel (vandfaldsmodel). Man kan dog i alle projekter vælge at specificere og udføre en accepttest.

Som øvelse kan i gennemføre en accepttest på Kasseapp’en. Nedenstående eksempel indeholder kun 5 step, og dækker ikke hele funktionaliteten. I skal nu tilføje yderligere 5 step, så testen bliver komplet. I den sammenhæng skal i tænke på følgende:

* Alt funktionalitet skal dækkes
* Test for fejlbetjening
* Test med forskellige værdier – også ugyldige (store tal, mange decimaler, grænseværdier)
* Tænk destruktivt (hvordan kan jeg få det til at fejle?)

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Input** | **Forventet resultat** | **Resultat af testen Dato: 1/12-19** |
| 1 | Start Kasseapp | Kasseapp vises. Det skal tydeligt fremgå at der ikke er indtastet data | OK |
| 2 | Scan vare (Indtast 12.75 og tryk ”Tilføj vare”) | Man skal nu se at sum er 12.75 | OK |
| 3 | Indtast ugyldigt input (space) og tryk ”Tilføj vare” | Input ignoreres | Fejl #1 |
| 4 | Indtast negativt tal (-456.78) | ? |  |
| 5 | Tryk ”Betal med kort” | Man skal kunne se summen, og at der skal betales med kort | OK |
| 6 |  |  |  |

Når i har defineret testcasen skal i udføre testen. I den sammenhæng udfylder i feltet til højre (Resultatet af testen). Skriv OK, hvis det fungerer som forventet, og Fejl (+ fejlnummer) hvis der er fejl. For hver fejl i finder, skal der udfyldes en problemrapport (se nedenstående eksempel – den der finder fejlen, udfylder den øverste halvdel af rapporten)

Problem rapport nr.:

Problem vedrørende:

Problem beskrivelse:

Årsag og løsning:

Kategori:

Problem rettet:

1

Indtastning af ugyldige input

Når man ikke indtaster noget, eller hvis man indtaster et ugyldigt input (bogstav eller tegn) og trykker ”Tilføj vare” går app’en i baglås.

Kun tal kan adderes til sum. Der skal derfor testes for, om input er et gyldigt tal, inden det lægges til sum.

A

1/12-19 af LSK

Problem fundet:

1/12-19 af LSK

I skal nu prøve at behandle minimum et af de problemer, i har fundet. Gennem debugging finder i årsag og løsning. Derefter retter i problemet, og udfylder nederste halvdel af problemrapporten.

## Øvelse 4 - Brugertest

I et agilt udviklingsforløb kan vi benytte brugertesten til at evaluere en prototype. Vi kan også gennemføre brugertest på færdige produkter. Til dette er der udviklet et helt forløb ((Link: EMU Brugertest forløbsbeskrivelse.docx).

Det ville være nærliggende at gennemføre en brugertest på kasseapp’en, men da denne er meget simpel og stadig en del abstrakt, giver det ikke så meget mening.