# Test strategi (TJ)

*Udarbejdelse af afsnittet om test strategi har vi brugt Software Engineering, A Practioner’s Approach 7. edition, kapitel 17 og 18.*

Test er et vigtigt element i ethvert software projekt, da et program ikke er færdigt før det er fejlfrit, brugervenlige og klar til brug. Et program er fejlfrit indtil næste fejl er fundet! Betyder det så at man aldrig er færdig med at teste? Ikke nødvendigvis. Når programmet udgives er det brugerne der overtager test af softwaren, men indtil da, så er det software teamet der tester. Det behøver ikke nødvendigvis at være to forskellige personer der programmerer og tester softwaren. ”Mindre” tests som unit- og integrationstest kan sagtens udføres af den person der har programmeret modulerne.

Følgende er en definition af relevante testmetoder samt en teststrategi for salgsmodulet.

### Unit test

Unit testning er typisk den første test der vil forekomme. I unit tests tester man de mindste dele af systemet, de enkelte delelementer, som blandt andet kan være simple CRUD kommandoer. Unit testning koncentrerer sig om testning af logikken og datastrømmen indenfor en komponent.

Når man unit tester ser man på at datastrømme ind i modulet er korrekte og at de ikke ændrer sig til uventede værdier eller typer. Samtidig testes om det er muligt, at nå hvert eneste scenarie i if-else statements og om loops holder sig indenfor grænserne. Loops, arrays og lignende mekanismer fejler oftest ved grænserne af det definerede spektrum. Hvad sker der når arrayet kommer til det n’te element og loopet passere tællerens værdi?   
Til sidst tester man ekstensivt de fejlscenarier en kunde kan finde på at kreere, selvom det er svært at forestille sig hvad de kan finde på. Selv hvis fejl opstår, og det gør de højst sandsynligt på et tidspunkt, så er det nødvendigt at designe moduler således, at fejlmeddelelserne bliver forståelige for brugerne og modulet viderestiller til næste vindue. Fejlmeddelelser skal ikke sige ”Du burde ikke kunne få denne fejl”, da det 1. ikke hjælper brugeren videre og 2. fortæller brugeren at noget er alvorligt galt.

### Integrationstest

Integrationstest udføres for at finde ud af om komponenterne man har unit testet kan interagere med hinanden. Integrationstests ser på hvilke værdier, og hvis programmeringssproget er typestærkt, hvilke typer der bliver returneret eller sendt videre. Stemmer værdierne og typerne ikke overens med de metoder der skal modtage får man problemer. Samtidig kan data blive tabt på tværs af interfaces, eksempelvis mellem salgsmodulet og serveren, eller databasen.

### Strategi

Testleder: Tobias

I forbindelse med kodningen af salgsmodulet ønsker vi at unitteste umiddelbart imens eller efter at kodningen af de enkelte delelementer er færdige. Unittesten udføres af personen, der har programmeret delelementet, fordi han har størst kendskab til dets kald, typer og værdier.

*Test miljø*: Testen skal udføres på seneste version af Visual Studio i et Windows 8.1 miljø, da det højst sandsynligt vil være det kunden kommer til at køre systemet på.

*Test grupper*: Integrationstesten af delelementerne i salgsmodulet vil blive udført som par test af teammedlemmerne. Her vil det blive relevant at teste forbindelse mellem klasser, webshop og databasen.

*Udførelse:* Både unit- og integrationstesten skal udføres manuelt.

*Test tidsplan*:Vi forventer at selve testningen vil foregå i samme sprint som kodningen tager sted og som projektet skrider frem vil der også indgå integrationstest mellem de enkelte delelementer løbende.

*Test resumé*: Det forventes at der tages noter og væsentlige fejl, ændringer og bemærkninger skrives i et dokument sideløbende med test. Dokumentet skal afleveres sammen med salgsmodulet.