Descriptive essay - Søren Larsen

The 4 agile testing quadrants

I den agile verden, sørger man for kun at bruge værktøjer og metoder der gavner udviklingen og projektet. Dette gøre sig selvfølgelig også gældende inden for agile-testing. I modellen "The agile testing quadrants", findes fire kvadranter, der hver for sig beskriver forskellige grunde til at teste. Modellen deler de fire kvadranter ind i en matrice, hvor de bliver skilt ad, efter forskellige formål. Den ene akse deler modellen mellem tests der støtter udviklerne, og test der giver kritik/feedback af produktet. Den anden akse deler tests designet til en mere forretningsorienteret del, op med tests designet til en teknologiorienteret del.

Første kvadrant

Her finder vi Unit og Component tests. Unit tests sørger for at teste de helt små dele af systemet i dybden. Component tests sørger for at teste samspillet af flere forskellige større dele af programmet. Disse typer tests er afgørende for kunne udføre Test-first princippet, som er en vigtig del af den agile udvikling. Denne type udvikling hjælper teamet med at finde fejl tidligt i forløbet, og samtidig giver det dem en stærk test-suite at støtte sig op ad, hvis produktet eventuelt ændre sig.

Anden kvadrant

Her arbejdes der stadig med tests der støtter udviklerne. Dog er der tale om test/aktiviteter der er mere rettet mod kunder og selve forretningsdelen af projektet. Der er stadig tale om tests, men de er stykket sammen så de kan forklares til forretningseksperter, og de er skrevet ud fra kundernes eksempler. Formålet er at lave tests der viser systemets egentlige funktion og opførsel på et højere plan.

Tredje kvadrant

Her er vi havnet i den anden side af matricen. Her laves flere aktiviteter der giver kritik af det egentlige produkt. Specielt i et forretningsmæssigt henseende. I denne kvadrant bliver der udført manuelle tests af programmets egentlige funktioner. Der udføres scenario tests, hvor man forsøger at opstille en brugers færden og brug af produktet, for at teste om det lever op til forventninger/krav.

Fjerde kvadrant

Sidste kvadrant sørger for at teste kvaliteten af produktets tekniske(Non-funktionelle) krav. Her tænkes på ting som performance, sikkerhed, robusthed. Her gøres ofte brug af værktøjer til at udsætte produktet for stress, som man ikke nødvendigvis kan gøre fra programmatiske tests. Resultatet af arbejdet i fjerde kvadrant, kan bruges til at udvikle tests/arbejde i første og anden kvadrant. Altså finder vi en specifik fejl i fjerde kvadrant, kan vi skrive tests til den unit/component det vedrører i første kvadrant.

System Testing

System testing udføres for at teste produktets krav. Her er der tale om de funktionelle samt non-funktionelle krav. Altså, kan systemet det vi forventer at det kan. Og gør systemet det på en måde vi kan acceptere. Der er tale om black-box-testing. Det vil sige at en tester ikke behøver dybere forståelse af produktet. System testing udføres som regel efter Unit Testing og Integration Testing.

Exploratory Testing

Denne form for testing skiller sig lidt ud fra andre dele af testing. Selve begrebet er ikke som sådan en test-type som fx Unit eller Component test. Det er mere en tankegang/mindset. Det er en måde hvor man udforsker sit system mens man tester. Dog udforsker man indenfor nogle rammer der bliver fastsat inden starten. Der er altså ikke tale om ad-hoc testing eller bug bash, hvor man bare prøver alt hvad der falder en ind. Man vælger nogle områder, og støder man ind i problemer noteres disse ned. Specielt hvis man er ved at komme udenfor de fastsatte rammer, noteres der ned hvor der bør laves

mere udforskende testing. Det gælder altså om at følge sin fornemmelse mens man navigerer rundt i systemet.