Jonas Kjær Simonsen

Descriptive Essay

De 4 Agile Testing Quadrants

Quadrant 1

Den første quadrant repræsenterer ved at være test-driven development som værende en stor udviklings proces i agile udvikling. Her skriver testene, før selve program koden skrives. Til at udvikle disse tests bruger man unit og component tests, hvor unit test handler om at man tester små dele af projektet, mens component test handler om at teste hvordan dele af et større system opfører sig. Disse tests er begge noget man normalt automatisere.

Quadrant 2

Den anden quadrant repræsentere ved at være business orienteret. Her udføres test som går ud på at vise og validerer hvad det færdige produkt skal kunne. Test vil som udgangspunkt blive lavet ud fra kundernes beskrivelser og eksempler af produktet. I denne del er der meget fokus på prototyper, funktionelle tests, eksempler mm. Disse tests er tildelsnoget man automatisere så kunden kan give hurtig feedback til udviklingsholdet.

Quadrant 3

Den tredje quadrant repræsenteres ved at være business orienterede. Her udføres test som bl.a. user acceptence test, usability test og exploratory test. Det særlige ved disse tests er at det udelukkende handler om at komme med kritik af produktet. Dette gøres bl.a. ved at emulere en rigtig bruger for at se hvordan han/hun bruger produktet. Disse tests er man nød til at kører manuelt, idet man emulere en rigtig bruger, og man må derfor bruge ens sanser og opfattelse til at vurdere om produktet er godt nok.

Quadrant 4

Den fjerde quadrant er repræsenteret ved at være teknisk orienterede. Præcis ligesom i den tredje quadrant handler det også her om at kritiserer produktet, dog bare de tekniske ting ved produktet. Det kan være f.eks. performance og sikkerhed. Disse tests kræver som ofte eksterne værktøjer.

System testing

System testing handler om at teste produktets opførsel, og handler bl.a. om at validerer om produktet opfylder de krav som det var givet til produktet. System testing bruges til at undersøge både funktionelle og ikke-funktionelle krav. Når de funktionelle krav skal testes bruges som ofte black-box testing, som betyder at man intet kender til hvordan koden fungere og dens opbygning, det eneste man kender er hvad input er og hvad output skal være, det er udelukkende ud fra denne form for informationer man laver de forskellige tests. Til disse former for tests kan man f.eks. bruge decision tables til at illustrere hvad input muligheder og output resultater skal være.

Exploratory testing

Exploratory test er en tilgang hvor testene, ikke er med til at planlægge de forskellige tests, men kun koder og udfører testene. Ved selve planlægningen, planlægges en x periode, hvori der skal testes. Dette kan f.eks. være 2-3 timer, hvor testerne kun skal side og teste. Under selve planlægningen, planlægges der også hvad der skal testes samt hvornår i perioden det skal testes. Ved udførelse af testene, er det meningen at testerene skal udføre 3 ting: test design, test execution og learning. Det handler om at testeren selv skal kontrollere execution og design og ud fra udførelsen af sine test, kan opnå ny viden om systemet, der kan hjælpe ham med at lave bedre tests.