

## De 4 agile test kvadranter

De fire test kvadranter er delt op mellem 2 akser, hvor den ene akse angiver om der fokuseres på teknologien eller firmaet, og den anden akse angiver, hvorvidt de udførte tests støtter teamet og dets udviklingsprocess eller kritiserer produktet og dets funktionalitet.

Navneangivelsen af kvadranterne (Q1, Q2, osv..) skal ikke forstås som en iterativ fremgangsmåde igennem de fire faser, men blot en måde hvorpå det er muligt at identificere kvadranterne.

### **Q1 - Teknologi fokuserede tests som støtter teamet**

Et eksempel i denne kvadrant er Test-Driven-Development fra XP metodologien, hvor der fokuseres på at få skrevet tests, som koden umiddelbart fejler, for derefter at optimere koden, så den består testen. Der fokuseres på den interne kode igennem unit testing og component testing, hvor forskellige elementer af programmet bliver testet imod forventede outputs.

### **Q2 - Business fokuserede tests som støtter teamet**

I en udviklingsprocess er det vigtigt at alle parter kommunikere med en fælles forståelse og enighed. For at sikre en fælles forståelse af domænet kan eksempler og simulationer bruges, da de er med til at sikre en enighed om, hvordan de forskellige elementer af projektet kommunikerer. Ser man på egentlige tests i denne fase, så bruges blackbox teknikken med funktionelle tests og man ser også på story tests som dækker over de angivne kriterier for hvornår en user story er gennemført.

### **Q3 - Business fokuserede tests som kritiserer produktet**

For bedst at kritisere produktet bruges der nogle forskellige metoder, som alle indebærer en bruger der interagerer med den funktionalitet man ønsker testet. Metoderne er exploratory testing, usability testing og user acceptance testing. De bruges bl.a. for at sikre at udviklerne ikke overser fejl i implementeringerne, og dermed for at sikre at funktionaliteten opfylder kravene.

### **Q4 - Teknologi fokuserede tests som kritiserer produktet**

Den sidste kvadrant dækker over teknikker som tester de non-funktionelle krav, ved eksempelvis at udføre performance og load testing. Der testes altså på områder som stabilitet, hastighed, sikkerhed, osv. Resultaterne giver udviklerne et svar på om produktet opfylder kravene og hvor det kan optimeres.

### **System testing**

En testmetode der udføres på et komplet system fra top til bund (ex. web-ui til database). Den udføres for at sikre at produktet overholder kravene og at de forskellige komponenter arbejder korrekt sammen. System testing udføres på et system, der indeholder alle de komponenter som har gennemgået integrations testing, og er det sidste skridt før acceptance testing.

### **Exploratory testing**

En hands-on metode hvor en bruger tester produktet gennem fysisk interaktion, typisk med stor frihed til brugerens handlinger og fremgangsmåde. Metoden bruges til at danne et indtryk af om produktet rent faktisk virker og testen er derfor også afhængig af brugerens evne til at "finde på" test cases imens de interagerer med de forskellige funktioner.