

Laboration 3

Slänger in en kort passus här om att jag kan ha missförstått detta helt. Rapporten kommer att handla om DFA generellt, och inte specifikt för de testfall jag har skapat. Begär gärna en komplettering om detta är fel!

Discovery

Steget innebär att man försöker identifiera och upptäcka användarbehov, affärskrav och tekniska krav hos funktionen som utvecklas. Det ska vara ett samarbete mellan testare, utvecklare och produktägare, där man med hjälp diskussioner, användarintervjuer och annan feedback hjälps åt att identifiera möjliga scenarier eller use-cases och fastställa de förväntade resultaten. Gruppen som träffas bör vara 3-6 personer.

Steget bör hållas så tätt inpå utvecklingsstarten av en ny user-story som möjligt för att förhindra att detaljer "försvinner" eller glöms bort.

När steget är slutfört ska alla intressenter, både tekniska och icke-tekniska, ha en gemensam uppfattning om arbetet som ska utföras.

I mitt test där man försöker registrera sig för nyhetsbrevet utan att ha godkänt villkoren tänker jag att både testare, utvecklare och produktägare har samarbetat.

T.ex. kan produktägaren/affärssidan kan ha legala krav på sig att ett godkännande måste ha gjorts.

Formulation

Nu är det dags för teamet att använda insamlad information för att formulera och definiera testbara krav/beteenden i form av scenarier. Ett scenario definierar det förväntade beteende på ett tydligt och kortfattat sätt i Gherkin-syntax och enligt formatet Given-When-Then:

- Given - beskriver förutsättningarna
- When - beskriver händelserna
- Then - beskriver det förväntade resultatet

När formuleringssteget är klar ska det finnas ett gemensamt förstående hos alla inblandade parter om vad scenariot beskriver och vad som ska testas.

Om Cucumber används ska slutresultat av steget vara en eller flera feature-filer.

Automation

I detta steg implementerar teamet automatiserade tester för att verifiera att funktionerna fungerar som förväntat och uppfyller de definierade kraven. Detta innebär vanligtvis att man använder testramverk eller verktyg som stöder BDD, som t.ex. Cucumber. Val av verktyg och tekniker kan variera beroende på vilken

funktionalitet som man ska testa, och hur den ska testas.

UI-tester kan genomföras med t.ex. Selenium eller Cypress medan API-tester kan göras med t.ex. REST Assured.

I detta steg arbetar man oftast enligt TDD-principen, dvs. att man först skriver ett test som fallerar och sedan fyller på med precis så mycket kod som behövs för att testet ska bli godkänt.

Efter detta startar en loop av refaktoring av koden följt av en verifikation att testet fortfarande är godkänt.

För just Cucumber börjar man med att skapa Step Definitions utifrån feature-filerna och fyller sedan på med de faktiska implementationerna av testerna.

Oftast behövs också ett antal hjälpklasser skapas för samla gemensamma beteenden.

Automatiseringsprocessen säkerställer att systemets beteende testas och valideras noggrant, och hjälper till att identifiera och åtgärda eventuella avvikelser från det förväntade beteendet.