

Rapport 1DT908 Mjukvaruutvecklingsprojekt

Reflektions rapport för projekt i mjukvaruutveckling



Författare: Emil Ulvagården
Termin: VT2024
Kurskod: 1DT908

Redogör för de faktorer som gör mjukvarudesign och utveckling komplext

Mjukvarudesign och utveckling är ett komplext då flertalet faktorer behöver uppfyllas för en lyckad produkt. Först och främst behöver produkten uppfylla de krav som ställs av kunden. Dessa krav kan förändras under arbetes gång vilket kan kräva stora förändringar eller så är kraven ofullständiga och utvecklaren behöver tolka vad kunden vill.

Den andra faktorn är att produkten ska vara användarvänlig för personer som inte använt produkten tidigare. Det kan vara svårt för utvecklaren under utveckling samt testning känner till hur produkten fungerar och tar därför vissa designplaceringar för givet.

Den tredje anledningen är att produkten kan vara tekniskt avancerad. Att en produkt är tekniskt avancerad medför att utvecklaren behöver ha kunskaper inom flertalet områden som till exempel databaser, nätverk och säkerhet.

Den fjärde faktor är att produkten kan behöva underhållas eller uppdateras. Att underhålla en produkt kräver förståelse över produkten och feedback från kunder för att korrigera buggar. Att uppdatera produkten kräver också en förståelse över produkten där nya krav från kund kan behöva implementeras.

En sista faktor är att utvecklaren behöver kvalitetssäkra produkten som levereras inom den angivna budgeten och tidsplanen. Detta kan vara svårt då problem kan uppstå under produktionen vilket gör att utvecklaren behöver arbeta extra för att få färdigt projektet i tid samtidigt som produktens kvalitet hålls.

Beskriv grundläggande begrepp inom mjukvaruutveckling, dess olika faser, roller, intressenter och metoder

Grundläggande begrepp är mjukvaruutveckling, kravhantering, design, implementering, distribution och underhåll. Mjukvaruutveckling är processen av att skapa programvara. Kravhantering är att få kundens krav till användbara punkter för utvecklarna. Design är planen för hur programvaran ska fungera och dess utseende. Implementering är skrivandet och testande av produkten. Distribution är att göra produkten tillgänglig för kunden och underhåll är att göra förbättringar samt fixa buggar.

Det första som ska göras för ett projekt är att fastställa kravhanteringen för att med hjälp av kraven skapa en design för produkten. När designen är skapad kan implementeringen påbörjas. När kraven är uppfyllda och implementeringen är avslutad kan produkten distribueras och sen börja underhållas.

De roller som används är produktägare, projektledare, utvecklare och designer. Produktägaren är den person som är ansvarig för kravhanteringen. Projektledaren är den som ansvarar och leder utvecklarna. Utvecklarna är dem som implementerar kraven som produktägaren har framtagit. Designers är dem som tar fram en design för hur produkten ska se ut.

Intressenterna är kunden, användaren, produktägaren och utvecklarna. Kunden är en organisation eller person som beställt produkten och användaren är de som kommer interagera med produkten. Produktägaren är den som är ansvarig för kraven och utvecklingsprojektet. Utvecklarna är personerna som utvecklat produkten.

De metoder som kan användas vid mjukvaruutveckling är vattenfall, agil, scrum och funktionsdriven utveckling [1]. Vattenfall är metoden där en plan fastställs innan projektet påbörjas och sen utförs planen helt innan produkten ses över. Agil

utveckling använder korta cykler för att anpassa arbetet eller förändra krav. Scrum är en metod liknande agil metod där regelbundna möten och korta cykler används för att organisera utvecklarna. Funktionsdriven utveckling fokuserar på att utveckla funktionaliteter på regelbundna intervall.

Beskriv principer för agila metoder för mjukvaruutveckling, deras syfte, och hur de skiljer sig från andra metoder.

Agila metoder används för sin flexibilitet, samarbete och kontinuerlig förbättring av produkter. Syftet med agila metoder är att få utvecklarna att samarbeta med varandra till skillnad från mer strikta metoder där kommunikation mellan utvecklare sällan utnyttjas. Agila metoder främjar samverkan med kunder för att säkerställa att produkten uppfyller kraven och kan förbättras vid behov. Traditionella metoder samarbetar sällan med kunder på samma nivå vilket gör det svårt att anpassa produkten för föränderliga krav. Agila metoder följer inte en strikt plan som de traditionella metoderna gör utan de agila metoderna är flexibla utifrån kraven som kunden ställt på produkten. De agila metoderna fokuserar på att kontinuerlig leverans av mjukvara medan de traditionella metoderna levererar mjukvaran först när den är färdig.

Reflektera över val av mjukvaruutvecklingsmetoder för aktuellt projekt.

För aktuellt projekt valdes scrum då de krav som ställts av kund var fastställda och kunden ville ha kontinuerlig överblick av programmets utveckling. I början valdes en sprint tid på 1 vecka då vi kontinuerligt fick respons av kund om vad som var bra och vad som var mindre bra. Senare under projektet gång byttes sprint tiden till 2 veckor och den kontinuerliga responsen fortsatte. I starten av projektet är det viktigt att komma i gång snabbt och börja få feedback av kunden om vad som är bra och vad som är mindre bra och när projektet väl har kommit igång och kundens feedback är välkänd flyter projektet på bättre och sprint tiden kunde förlängas. Av dessa anledningar ansågs scrum som en lämplig mjukvaruutvecklingsmetod.

Jämför och resonera kring vilken metodik som är mest lämplig för projektet.

Om mjukvaran som ska utvecklas endast ska fokusera på funktionalitet och ej design kan funktionsdriven utveckling vara ett bra alternativ om den kombineras med en agil metod för att samarbeta med kunden så att de producerade funktionerna är korrekta utifrån kraven. Det kan även vara en bra idé att köra på någon agil metod om funktionalitet inte är det enda kravet och design också spelar roll. Detta för att säkerställa att kunden är nöjd med programmets utseende samt de funktionaliteter som finns. I de fallen där kunden är bestämd på exakt vad som de vill ha kan vattenfallsmetoden vara ett alternativ. Det finns dock risker med vattenfallsmetoden då kunden kan ändra sig under utvecklingen och när produkten väl är färdig stämmer den inte längre med kundens krav. Den mest lämpliga metod skulle vara en agil metod och med möjlighet för att kombinera in funktionsdriven utveckling.

Reflektera över hur effektiva grupper skapas och vad som krävs av individ och övriga gruppmedlemmar, hur hanteras problem och konflikter.

Effektiva grupper skapas genom god kommunikation och bra samarbete. Med god kommunikation mellan gruppmedlemmar kan problemområden för enskilda individer diskuteras med de andra gruppmedlemmarna och om det behövs kan utvecklarna lägga mer fokus på problemområdet. God kommunikation behöver inte vara endast bland utvecklarna utan det kan även innefatta kommunikation mellan utvecklingsgrupper som sitter i samma projekt men har olika områden och behöver vissa grunddelar för att själva kunna fortsätta. Kommunikationen kan även vara mellan produktägare och kund så att när kraven ändras kan utvecklarna få reda på det tidigt i stället för när produkten ska distribueras. För att utvecklarna ska få reda på krav förändringar krävs god kommunikation mellan utvecklingsgrupperna och produktägaren. Om en god kommunikation kan hållas i grupperna blir samarbetet enklare att hålla och utvecklarna vet om var och hur det går för sina gruppmedlemmar så att de kan hjälpa till om det skulle behövas. Av individerna krävs det att alla gör sitt jobb och delar med sig till de andra gruppmedlemmarna om hur arbetet går, var dem är och arbetar samt vilket krav de arbetar på. De övriga gruppmedlemmarna måste visa förståelse för att arbetet kan ta längre tid hos vissa än hos andra samt de måste vara öppna för att hjälpa sina gruppmedlemmar om det skulle behövas. De övriga gruppmedlemmarna måste även själva vara öppna för att kommunicera med varandra om sina egna möjliga problem samt hur arbetet går i övrigt. Om problem uppstår för en individ finns möjligheten att resterande gruppmedlemmar kan komma med lösningsförslag eller gå in och hjälpa till med problemet. Om konflikter uppstår i olika lösningsalternativ kan de olika gruppmedlemmarna diskutera och väga de olika lösningsalternativen mot varandra för att komma fram till en lösning.

Del 2

Redogör vad yrkesrollen mjukvaruutvecklare kan innebära.

Yrkesrollen som mjukvaruutvecklare kan innebära flertalet olika arbetsuppgifter. Som mjukvaruutvecklare kan man sitta med programmering för att skapa program och applikationer men man kan lika väl sitta och designa systemarkitekturen eller designa applikationer. En uppgift som mjukvaruutvecklare kan vara att felsöka applikationer och identifiera varför felet uppstår. Detta för att kvalitetssäkra en produkt och optimera produktens prestanda samt stabilitet. En mjukvaruutvecklare kan sitta med dokumentation av kod eller användarinstruktioner. En mjukvaruutvecklare kan även ha som arbetsuppgift att undersöka sammanfogningar mellan olika hårdvarukomponenter och olika system. En mjukvaruutvecklare kan ha som arbetsuppgift att sammanfoga versioner med andra gruppers arbeten för att bygga ihop en applikation. Mjukvaruutvecklare kan även ha som uppgift att spåra olika kodversioner och kolla var olika förändringar görs ifall tidigare versioner av ett program fungerat bättre än den nya versionen för att kunna återgå till en gammal version.

Beskriv grundläggande arbetsuppgifter som en mjukvaruutvecklare kan få göra, försök också få med en progression över tid.

Om mjukvaruutvecklaren börjar med att programmera applikationer kan de få börja med att få implementera nya funktioner eller lösningar för att sedan gå in mot att reparera buggar och underhålla applikationer. Om en mjukvaruutvecklare börjar med dokumentation görs det i samband med programmeringen då de får börja kommentera kod för att andra lätt ska kunna läsa in funktionaliteten av koden. De kan senare bli involverade i teknisk dokumentation. Om en mjukvaruutvecklare börjar arbeta med design kan de få börja med att implementera andras designer och med tiden kan de få börja göra mer beslut gällande slutgiltig design eller själv få skapa en egen design. Om en utvecklare har som arbetsuppgift att undersöka kopplingar mellan komponenter eller applikationer kan man få börja med att läsa sig till fler kunskaper kring området. Senare i karriären kan man få hantera mentor eller föreläsare för att lära andra i samma område.

Återberätta kortfattat vad du har lärt dig om yrkesrollen via gästföreläsningarna och motivera varför du tycker att det var intressant.

Det som jag lärt mig om yrkesrollen från gästföreläsningarna har varit hur pass brett som arbetsuppgifterna kan vara även om man arbetar på samma företag och samma projekt så kan ens arbetsuppgifter vara långt ifrån varandra. Kan vara någon som arbetar med endast koppling till en databas medan någon inte använder databasen. En annan sak som jag lärt mig alla uppgifter som görs inte alltid görs för att leverera en produkt till en kund utan vissa uppgifter görs som testning inför framtida projekt. En annan sak som jag lärde mig från gästföreläsningen var hur man som mjukvaruutvecklare kan arbeta för att hjälpa andra företag med dem som problem som de företaget har. Detta var intressant då man fick en inblick i hur en mjukvaruutvecklare kan arbeta på en dag med uppgifter som de har för att utveckla en bra produkt.

Vad i det du gör i det nuvarande projektet kan ses som del av rollen mjukvaruutvecklare?

Den del som jag har haft hand om kan ses som en del av rollen mjukvaruutvecklare då jag behövt undersöka hur kopplingen till en databas ska göras och hur vi på ett effektivt sätt ska plocka ut nödvändig information ur databasen. Det arbetsuppgifter som jag haft för att lösa detta har varit att programmera både kopplingen till databasen och vad för information som plockas ur databasen samt av vilken funktion. Uppgiften kräver även en hel del testning för att säkerställa att allt fungerar som det ska och om det inte gör det behöver koden förändras.

Motivera varför just du kommer vara lämplig i rollen som mjukvaruutvecklare.

Jag är lämplig under rollen som mjukvaruutvecklare då jag undersöker lösningar till de krav som ställs av kund men även gruppmedlemmar om de skulle få något problem eller vilja ha en specifik funktionalitet. Jag håller enkelt diskussioner med gruppmedlemmar

om hur arbetet går och när problem uppstår går det enkelt att diskutera om de skulle behöva extra hjälp eller om det tror att de kan lösa problemen själva. Jag arbetar gärna med uppgifter som kräver komplexa lösningar men även enklare uppgifter då oavsett svårighet och komplexitet antar jag gladeligen utmaningar. Jag tycker även om att undersöka om mina lösningar ger rätt resultat och om de inte gör det är det kul att försöka ta reda på vad som är fel. När felet väl har lokaliserats är det intressant att leta efter potentiella lösningar eller om det är så att man behöver börja om på nytt.

Referenser

[1] B. Appelbaum. "Top 6 Software Development Methodologies." Planview.com.
<https://blog.planview.com/se/top-6-software-development-methodologies/#:~:text=Top%20Software%20Development%20Methodologies%201%201.%20Vattenfall,Extreme%20Programming%20%28XP%29%20...%206%206.%20Lean%20> (accessed 2024-03-25)