

## Teoridel

Maximillian Lundberg .NET24

### G-FRÅGOR

#### 1. Vad innebär projektledning och projektstyrning rent allmänt?

Projektledning och projektstyrning handlar om hur man organiserar arbetet i gruppen. Det kan involvera möten, att assigna uppgifter som ska göras, hålla code reviews eller bara allmänt diskutera vad som behöver göras inom projektet,

Inom just programmering är det vanligt att ge någon en ledande roll som till exempel en scrum master som hanterar utdelningen av tasks och ser till att deadlines hålls.

Det kan också handla om konflikthantering och hur gruppen hanterar motgångar som missade deadlines.

Det handlar också om hur de olika delarna av gruppen kommunicerar med varandra, till exempel när en grupp är klar med sina uppdrag kan en annan grupp börja på nästa steg.

#### 2. Vad handlar gruppdynamik om och varför är det viktigt att förstå och arbeta med?

Gruppdynamik handlar om hur de olika medlemmarna eller "grupperna" inom gruppen arbetar och kommunicerar med varandra.

Speciellt inom programmering finns det väldigt många inriktningar att specialisera sig till, det är därför viktigt ha tydlig kommunikation mellan de olika arbetsgrupperna så att man undviker missförstånd eller konflikter.

Det kan betyda att man inte bör använda inriktning specifika termer när man pratar med andra medlemmar eller gruppen, utan prata i mer allmänna termer så andra förstår en.

Det kan också vara bra att hålla koll på vilka som fungerar och jobbar bra tillsammans eller ge någon en roll specifikt för att kommunicera mellan medlemmarna, som tex Devops.

En dålig gruppdynamik är när folk inte kommunicerar med varandra, inte vågar ställa frågor eller missförstår varandra hela tiden, vilket kan leda till konflikter.

Med en bra gruppdynamik vet alla vad de ska göra och är inte rädda att lyfta fram frågor som de har, vilket leder till ett effektivare grupparbete.

#### 3. Beskriv översiktligt vad agila metoder är

Agila metoder är en arbetsmetodik som används inom programmering där man fokuserar på att dela upp arbetet i mindre delar och planera utvecklingen kring det.

Det är stort fokus på att kunna jobba i grupp och samarbeta med andra genom checkins och möten.

Genom att dela upp arbetet i mindre delar och utveckla funktioner stegvis är det enklare för kunden att se och begära förändringar och för utvecklarna att presentera nya funktioner till kunden och få feedback direkt

Genom att ha en så kallad Continuous Integration/Continuous Delivery (CI/CD) minskar man "ner tiden" för produkten och kan snabbt fixa fel och buggar samt uppdatera produkten utan att behöva stänga ner programmet.

#### **4. Beskriv följande agila metoder översiktligt.**

##### **1. scrum**

Fokuserar på att planera arbetet kring "sprints", där en backlog är allt som behöver göras. Efter sprinten har man sprint reviews och retrospektiv. arbetet leds av en scrummaster

##### **2. Kanban**

En agil metod som fokuserar på att visualisera workflows och med cards/post-it lappar. Arbetet delas ut via carden och sätts i To do-Doing-Done kategori, ingen scrum master eller "ledare" utan man väljer uppgifter baserat på cardsen och jobbar på den så länge det behövs.

##### **3. Extreme Programming**

delar upp workflows i "iterationen", fokus på att hålla koden ren och läsbar. använder test driven-development där du skriver kod som säger vad koden ska göra innan man börjar utveckla. Uppmuntrar "parprogrammering" där man skriver koden samtidigt som en annan reviderar den "live", alltså samtidigt som man skriver.

### **Reflektion kring projektet (CC)**

#### **1. Hur planerade ni arbetet? Förklara översiktligt hur ni gjorde och hur du tycker att det fungerade.**

Vi planerade inte alls, nada zitt. tvärtom det var en väldigt fientlig atmosfär för att ens assigna dom enklaste av sprinten. De få verktygen vi tänkte använda fungerade inte såsom gira, men det visste alla och sket i, även fast det togs upp på mötena. Det var bara en till av många bullshit ursäkt vi kunde ge Johan under vecko mötet till varför inget jobb hade gjorts.

Vi hade dagliga standups fastän inga sprints fanns eller var assignade och all koden skrevs i sista minuten

UML diagram you ask? nope inget sådant eller ens program flödesdiagram. Vi diskuterade det men en kille deletade tiketen för det, han skrev noll rader kåd i slutandan men kom på tävlingen för att vara med vissa vad "vi" gjort

## **2. Lyckades ni leverera delresultat regelbundet? Varför eller varför inte?**

Hahahahaha, gud nej inte ens i närheten, vi fick inloggningen att fungera that's about it. Några annonser kunde visas men inte som det var tänkt.

Anledningen var för att folk inte ville eller kunde kåda och för att skolan lät oss komma undan med det. Seriöst 80% av gruppen var antingen vibe coders eller rakt av fuskade genom att kopiera andras kod.

Ingen ville erkänna att dom inte kunde kåda, eller att dom använde AI (eller fuskade). En person laddade upp en zip fil i koden, flera skrev ingen kod alls, en person försökte hoppa in 8 veckor efter projektet startade.

Helt ärligt, det var en enda stor shit show från början till start. Skolan borde ha gått in tidigt och sagt till dom som inte ville jobba och saboterade att skärpa sig, om inte för deras skull så för oss som faktiskt ville jobba på projektet.

## **3. Hur kommunicerade ni inom gruppen och med PO/beställare? Fungerade det bra eller inte?**

Vi hade som tidigare sagt dagliga standups men inga sprint tickets (nej jag skämtar inte, 100% seriös här)

Men det spelade ingen roll för dom flesta kom inte ens på det. Vi hade ett kontrakt skrivet som var bra på papper men betydde jack-shit i verkligheten då alla ignorerade det.

det pågick trakasserier och mobbning istället men för det mesta jobbade alla separat från varandra utan någon riktig kommunikation alls (om dom ens jobbade)

## **VG-fråga**

**Vad ser du för styrkor och brister i er metodik som helhet? Skulle någon annan metod eller enskilda verktyg kunnat hjälpa er bättre?**

Jag siktar inte på VG men kommer svara på den här frågan ändå bara bara för att beskriva hur otroligt efterbliven, toxic och korkad våran grupp var.

Vi hade inga styrkor och vår metod var “se hur långt vi kan skjuta på att arbete innan skolan callar våran bluff” (spoiler alert, det fungerade hela projektet “and then some”).

Jag tror inte att några verktyg skulle ha hjälpt oss prestera bättre för det största problemet var egentligen att de flesta i gruppen inte kunde programmera alls, that's about it.

Jag kommer vara lite kritisk gentemot skolan här och säga att det **ENDA** som kunde ha hjälpt oss var om skolan hade gått in och frågat “Vad in i helvete håller ni på med!?! ni har inte ens skrivit en rad kod eller planerat någonting alls och det har gått 10 veckor!!!!”, fast i ett tidigare skede.

Tycker inte det är okej att en grupp kan “rösta” att skita i allt arbete, det påverkar faktiskt oss som vill lära oss något från projektet och är verkligen inte okej, men så blev det ändå.

Ta gärna med det till nästa år, om gruppen inte börjar jobba på projektet och skriva kod inom 2 veckor, kalla in dom och kräv att dom startar, även om koden är dålig eller basic.

Och innan jag får höra “man lär sig från dåliga upplevelser” eller något liknande så vill jag säga **B-U-L-L S-H-I-T**. Man lär sig dåligt från dåliga exempel och bra från bra exempel. Om det stämde att man lär sig bra från dåliga upplevelser så borde skolan designa sina lektioner så dåligt som möjligt, anlita dom sämsta lärarna och låta folk utan kod kunskap organisera kursupplägget.