Final report

Albin Rössle, 920124

Dalsher Jalal Kertan, 980329

Anwarr Shiervani, 000601

Timmy Malm, 000120

Andreas Pettersson, 980417

Georges Kayembe, 940716

Customer value och scope

Scope och framgångskriterier

Del A

Kunden var nöjd med de grundläggande funktionerna som är presenterades i appen och tycker att dem är relativt värdeskapande då appen väl förhåller sig till det externa behovet. Det fanns en del funktioner som kunde tagits med i appen och det slutgiltiga resultatet med arbetet. Dessa funktioner är tillägg som eventuellt kan läggas till i framtiden. Kunden hade en idé för ett liknande projekt i framtiden. Idén var att utforma en app som inte enbart begränsas till filmer utan kan vidgas till att exempelvis svajpa för aktiviteter som en grupp utför tillsammans, exempelvis restauranger, liseberg, naturparker osv.

Produktägaren hade återkommande möten och intervjuer med den externa aktören. Till en början var intervjuerna på varierande dagar och var för det mesta korta möten under några minuter över telefonsamtal. Men i och med att gruppen arbetade med scrum bestämdes att möten skulle hållas på söndagar så att gruppen kunde redovisa vad som åstadkommits i sprinten men också för att gruppen kunde få feedback på färdigställda stories. En annan anledning var för att gruppen skulle få reda på vad som var det minimalt värdeskapande inkrementet för kunden inför kommande sprint och vad som bör prioriteras.

För det mesta hade kunden åsikter om designen, övergripande funktioner och helhetsintrycket av appen. Det fanns en del bakomliggande arbete som handlade om kodande vilket gjorde att gruppen ibland fick gå emot kunden och det påverkade det administrativa arbete med den externa aktören. Ett exempel på detta är att redan i andra sprinten presenterade kunden att han ville ha trailer på film och detta var inte möjligt då gruppen hade mycket annat bakomliggande arbete i koden. Men för det mesta kände kunden att vad som prioriterades inför sprinten skapade ett minimalt inkrementellt värde.

Gruppen hade vid tillfällen en bristfällig kommunikation vilket gjorde det svårare att samarbeta. Detta var en av framgångsfaktorerna som gruppen arbetade en del med senare i projektet.

Del B

Kunden tyckte att det fanns 3 andra egenskaper av appen som kunde ha förbättrats:

- “Trailer” på filmer
- “Prestationer och belöningar” för användning av appen
- “Kontoinställningar”

Trailer var för att visa en grafisk illustration av filmer i form av video. för denna funktion var det tänkt att vi skulle hämta data direkt från TMDB eller en tredjeparts app som exempelvis youtube. När det kommer till prestationer ville kunden ha belöningar i form av troféer för olika aktiviteter som utförs i appen, exempelvis om kunden svajpar 100st. filmer får kunden en grafisk trofé. Hela tanken var att göra appen mer social och interaktiv. Inställningar för profilen var för att kunna lägga till namn, ålder, status och övriga personliga uppgifter. Detta var i syfte till att göra appen personlig.

Det främsta framgångskriteriet är att gruppen har en bra kommunikation där man planerar och samarbetar tillsammans.

Del C

Alla dessa önskade funktioner låg relativt bra i prioritet. T.ex. prestationer och kontoinställningar fanns det inte användning för tills mot slutet. För dessa funktioner hade det kort och gott behövts mer tid.

Trailers hade kanske gått att få in sista sprinten men då vi inte var helt överens om den skulle prioriteras in eller inte hade vi haft nytta av att kunden var bättre insatt i våran process så att vi kan göra prioriteringar vi är mer enhälliga om.

Gruppen får en förbättrad kommunikation genom att ha mera möten och aktivt delta i chattdiskusioner på daily scrum. En förbättrad kommunikation kan även ske genom en kommentarer i kod.

User Stories

Del A

Vi har skrivit stories enligt principen: Som… vill jag… för att… där vi inkluderar estimates enligt fibonaccis talföljd och acceptanskriterier. Produktägaren har även gett varje story en prioritet. Vi har alla haft ansvar att skriva stories men med större fokus på att produktägaren och kunden gör det. Oftast förfinades stories och bedömdes gemensamt vid sprint planning.

Acceptanskriterierna kunde vara skrivna både ur ett användarperspektiv för att sätta krav på funktion/design men även ur ett utvecklarperspektiv för att strukturera och följa program-design. Kunden tyckte om arbetet med att skapa acceptanskriterier för stories då det gav honom en tydlig överblick för hur gruppen arbetade med scrum. Att kunden deltog i att skriva acceptanskriterier gjorde att vi fick goda förutsättningar att förstå kunden bättre. En del av acceptanskriterierna förblev oförändrade under hela projektet och ändrades inte, detta för att gruppen inte tog initiativet till att skriva nya stories eller utveckla befintliga stories.

Del B

Vi hade haft nytta för framtiden att tänka mer långsiktigt och mer genomgående utveckla stories som kan bli relevant senare så vi inte går miste om bra idéer bara för att de inte ligger i backloggen. Eventuellt kan man även i framtiden ha fler specifika acceptanskriterier för själva kodande och inte bara sådant som har med administrativt arbete att göra.

Del C

Förslagsvis kan man förbättra planeringen för framtiden genom att börja använda epics som man sen bryter ner till “enkla” user stories som utvecklingsteamet ska bygga i kommande sprint. Epics kan eventuellt hjälpa oss att få en övergripande bild på hur kunden vill ha det. Vidare kan man även ha avsatt tid för att skriva user stories tillsammans så man gör det kontinuerligt.

Ett sätt att förbättra acceptanskriterier är att skriva fler och mer specifika acceptanskriterier för att vidare parametrisera storyn och tydliggöra vad utvecklingsteamet ska bygga.

Acceptanstester

Del A

Vi började relativt sent i projektet med att genomföra acceptanstester då det var något vi till att börja med hade missat att få in i samarbetet med kunden. När vi började använda oss av dem fanns ingen konkret plan för hur de skulle genomföras eller vad vi skulle få ut av dem. Det gjorde att vi hade svårt att veta om kunden ansåg att acceptanskriterierna var uppfyllda eftersom testproceduren inte var tydligt specificerad.

Del B

Det hade varit önskvärt att börja med acceptanstester från första sprinten och få kundens ögon på huruvida acceptanskriterierna uppfylls eller inte. Utifrån det kan man sedan ändra på inkrementet beroende på feedback för att säkerställa att kunden är nöjd med hur sprinten utförts. Om detta görs på samma sätt varje gång är det lättare att få ut något av det.

Del C

Genom att från början av projektet vara medveten om vikten av kundens åsikt kring huruvida ett acceptanskriteria är uppfyllt eller inte, och ha en plan för hur acceptanstesterna ska genomföras med kunden kan man på ett tydligare sätt få ut något av testerna.

KPI:er och deras påverkan

Del A

De tre KPI som gruppen använde var välmående, upplevd kunskapsutveckling och spenderad tid. Det fanns ett fjärde KPI, men det presenterades i Jira som velocity/burndown. Spenderad tid beräknades i antal timmar varje person lagt ner på varje sprint under en vecka, detta inkluderar då även timmar för möte i gruppen, möte med handledare, spenderad tid för chatt diskussioner samt individuellt arbetade timmar. För upplevd kunskapsutveckling bedömde gruppen individuellt hur man tycks ha utvecklats på kunskapsnivå från skala 1-10. Välmående beskriver hur individer kände under dagar som man arbetade. En del kunde känna sig

produktiva medans andra stressade eller missnöjda. Velocity var inte särskilt användbar då det skedde stora förändringar i vårt arbete.

Del B

Gruppen kände att det var ganska svårt att mäta och uppskatta nyckeltalen men kände att det påverkade arbetet då det gav en överblick över processen och delvis hur man kan ändra på arbetssättet baserat på KPI:n. Spenderad tid kunde kvantifieras till x antal timmar och blev en relativt godtycklig samt konkret bedömning av arbetssättet. Gruppen kände att tiden påverkade arbetssättet på ett positivt sätt för att det blev enklare att basera det efter ett scrum arbete.

Del C

En KPI som man kan tänka sig skulle skapa mer värde för kunden är Customer Satisfaction. Vi har såklart varit i konstant kontakt med kunden, men det skulle varit smartare om vi bad kunden fylla i ett formulär efter varje Sprint när produkten testas. På det här sättet så blir det enklare för oss att se hur tillfredsställd kunden har känt sig, utan att gå in på detalj kring enskilda funktioner.

Social Contract and Effort

Social contract

Del A

Vårt sociala kontrakt har inte använts speciellt mycket. Vi skrev ett socialt kontrakt under en tidig fas av projektet, men vi uppdaterade det aldrig ordentligt och gruppen upplevde att vi kunde jobba bra med projektet utan att använda kontraktet. Istället för kontraktet använde vi separata styrdokument som används på ett lite annorlunda sätt till det sociala kontraktet. Dessa styrdokument bestämde hur vi skulle jobba och kunde hjälpa folk att komma igång med vissa delar av arbetet. Till exempel så skrevs ett dokument som specificerade programstrukturen för projektet för att alla skulle veta hur programmet var strukturerat och så att alla kunde hålla sig till denna struktur.

Del B

Det bästa hade varit om vi hade haft ett socialt kontrakt som kunde hjälpa gruppen att jobba optimalt. Ett socialt kontrakt som uppmuntrar att alla är aktiva inom projektet och som hjälper gruppen fördela arbetet beroende på medlemmarnas styrkor och kunskaper. Ett optimalt sociala kontrakt skulle också innefattat rollen som våra styrdokument tog.

Del C

När vi märkte att vi inte använde det sociala kontraktet så skulle vi ha uppdaterat det för att bättre stämma in på situationen inom vår grupp. Vi hade ett socialt kontrakt som främst uppmuntrade till att alla kommer till mötena, hur vi hade mötena och att medlemmarna kontribuerade till inlämningarna; men detta var inte ett problem inom vår grupp. Istället för detta skulle vi kunna uppdatera det för att se till att arbetet delades upp jämnare och som på något sätt uppmuntrar till att detta görs väl. Hade vi ändrat på kontraktet för mycket skulle det enkelt kunna uppdateras för att bättre passa vår situation eftersom det hade varit mer relevant till projektet, men genom att inte uppdatera det alls så var det aldrig speciellt relevant till gruppens situation. Uppdatering av det sociala kontraktet skulle antingen kunna ske utöver styrdokumenten, eller tillsammans med styrdokumenten.

Spenderad tid

Del A

Vi har dokumenterat en uppskattning för hur mycket tid de olika gruppmedlemmarna spenderade på projektet under varje vecka. Vi använde detta mått för att se till att alla jobbade relativt lika och att timmarna låg runt 20 per vecka. Vi pratade om att det kunde användas för att jämna ut arbetet ifall någon skulle jobbat extra lite eller extra mycket en vecka genom att denne jobbar mer eller mindre under kommande vecka, men den situationen uppstod inte i gruppen.

Del B

Optimalt hade gruppen använt tidsmåttet för att fördela arbetet så jämnt som möjligt, och för att se om någon jobbar mycket mer eller mycket mindre än resten av gruppen och att följaktligen justera arbetsfördelningen inom gruppen. Vi skulle även kunna justera hur mycket arbete vi kunde göra under en vecka beroende på hur lång tid våra olika user stories tog i relation till de 20 timmar som gruppen skulle snitta.

Del C

Genom att alla mäter sin spenderade tid mer noggrant och genom diskussion på mötena om det hade vi kunnat utnyttja tidsmåttet bättre än vi gjort för att fördela arbetet mer jämt och även för att justera den totala mängden arbete per vecka. Detta skulle framför allt åstadkomma jämnare fördelat arbete.

Design decisions and product structure

Designval

Del A

Vi bestämde oss att använda dessa följande designverktyg:

- Model View Controller (MVC): Med anledningen att skapa en lämplig systemarkitektur och underlätta eventuella kommande förändringar i koden.
- Git & Github: För att hantera och lagra samt möjliggöra att alla gruppmedlemmar kan arbeta parallellt med projektet.
- Jira: För att underlätta ärendehantering och smidigare samarbete kring Scrum
- The Movie Data Base (TMDB): För att samla in vårt databas material.
- Firebase: För att möjliggöra datalagring samt realtidskommunikation mellan enheter.
- Flutter & Dart: Ramverk respektive programmeringsspråk som vi använde oss för att bygga vår applikation.

Kunskapsnivån när det gäller dessa verktyg varierar naturligtvis från individ till individ inom gruppen. Vissa av oss hade exempelvis tidigare erfarenhet med Jira medans att andra

upptäckte begreppet för första gången. Detta ledde givetvis till att vi, i tidigt stadium, tänkte dela oss i tre ansvarsområde, det vill säga: grafik, databashantering och backend. Den strategin visade sig vara olönsam eftersom vi snabbt märkte att det kunde resultera till att individerna inom gruppen blir insnöade i ett specifikt område och tappar koll för projektet i helheten.

Dock uppstod det en sak som alla hade gemensamt, det vill säga att ingen hade sedan tidigare jobbat med Flutter & dart. Vi valde Flutter och Dart främst för att ramverket möjliggör att utveckla en applikation dedikerad till olika plattformar (ex: Android och Iphone). Men vi blev tvungna att lägga hela veckan 4 för att alla gruppmedlemmar skall bekanta sig med programmering i Flutter och Dart, vilket bromsade till viss grad arbetsprocessen samt ledde till att vi inte hade något konkret att visa till kunden i början av vecka 5.

Del B

Ett alternativ hade varit att tidigt välja ett programmeringsspråk som de flesta i gruppen var bekanta med. Sedan hade vi kunnat tillsammans med kunden undersöka vad ett sånt val hade inneburit. Det är värt att notera att trots att vi använder Flutter och Dart så kan slutresultatet endast köras på en Android enhet. Så med andra ord hade det räckt att vi utvecklade exempelvis en android applikation med Java som programmeringsspråk i och med att alla hade åtminstone lite erfarenhet i Java.

Del C

Valet av Flutter och Dart som ramverk respektive programmeringsspråk har inte påverkat projektet på ett negativt sätt. Det har tvärtom varit gynnsamt ur en programmerares synvinkel i och med att vi har lärt oss ett nytt programmeringsspråk.
Men fortsättningsvis får vi ha i åtanke att det kan bli tidsmässigt svårt att kasta sig in i en ny och okända utvecklingsmiljö däremot kan det vara effektivt att lösa ett uppdrag med de verktygen som man redan är bekant med.

Teknisk Dokumentation

Del A

I början av projektet resonerade vi kring olika typ av dokumentation som skulle användas som underlag för att underlätta arbetsprocessen. Vi skapade en lathund om hur vi ska komma igång med Dart Doc (Darts standard för dokumentera kod). Vi skapade en lathund om hur vi ska komma igång med Git och Github samt hur vi kan testa koden.
Vi diskuterade om att utföra en bättre teknisk dokumentation genom att skapa UML-diagram för att få ytterligare förståelse på hur appen är uppbyggd.

De delarna som inte har funkat så bra är kod dokumentation samt UML-diagram. När det gäller kod dokumentation så har vi helt enkelt slarvat med det. Principen var att dokumentera sin kod vid själva utvecklingen i programkod och ingen av oss kan påstå sig ha följt det ordentligt. Ju mer koden växte desto svårare blev det att förstå samt skriva tester.

Idéen om UML-diagram kom sist efter att vi upptäckte svårigheterna att förstå varandras kod samt skriva tester. Vi hann inte att utföra det på grund av tidsbrist och valde att prioritera andra funktionalitet som skapar mervärde för kunden.

Del B

Det hade varit effektivt om vi initialt började med att framställa ett Requirements and Analysis Document (RAD) samt System Design Document (SDD) som utgjorde underlag för systemdesign.
Detta med andra ord hade inneburit att vi designade hela applikationen genom att först ta fram en domain model inklusive alla eventuella klasser och attribut samt deras egenskaper.

Del C

En lämplig strategi till nästa gång kan förslagsvis vara att vi tillsammans designar från starten den abstrakta kodarkitekturen. Detta kommer att favorisera en bättre kodförståelse inom gruppen och det kommer naturligtvis minimera problemet som uppstod när vi skulle skriva tester för andras kod.

Underhålla dokumentation

Del A

Vi har använt mycket av vår dokumentation antingen för att styra vårt arbetssätt eller för att dokumentera kod m.m. ur ett hypotetiskt perspektiv där vi hade fortsatt med projektet mycket längre. Vi har varit slarviga med att uppdatera vissa styrdokument utan bara haft konsensus att vi har förändrat något. Dessa förändringar gjordes främst vid sprint retrospective med samtliga närvarande.

Del B

I längden hade vi behövt vara bättre på att uppdatera vår dokumentation, både styrdokument och kommentarer/tester. Annars hade vi suttit i en sits där alla har olika uppfattningar om hur vi jobbar och man kan märka ganska fort att man har mindre nytta av likväl styrdokument som övrig dokumentering när dokumentationen inte överensstämmer med verkligheten.

Del C

Vad vi hade kunnat göra för att komma närmre kontinuerlig dokumentation vid fortsatt projekt är att ta dokumentation i beaktning på ett annat sätt när vi skriver stories samt göra en tydligare grej av att uppdatera styrdokument i samband med förslagsvis sprint retrospective.

Kod Standard

Del A

Vi hade inte direkt någon standard på hur vi skulle skriva kod. Vi fick tipset någon gång under handledningstid från handledare men det blev tyvärr inte något av det. Vi delade däremot det ansvaret till alla i gruppen, det vill säga att ansvaret att skriva kod på ett tydligt sätt så att andra kan förstå.

Del B

Ett alternativ hade varit att bestämma en kodstandard som skulle följas under utvecklingen. Vi hade förslagsvis kunnat införa ett gemensamt sätt att deklarera/döpa variablerna eller klasser.

Del C

Vi kan i framtiden skapar en dokumentation där vi tydligt bestämmer hur vi tänker skriva kod och vilka regler som gäller. Detta kan underlätta arbetsprocessen och göra koden tydlig och lättläst.

Applications of Scrum

Scrum-roller

Del A

Vi har använt följande roller:

- Scrum Master (1)

- Produktägare (1)

- Kod-testare (2)

- Kod-granskare (2)

Dessa roller delades ut under läsvecka 4 och vi hade så småningom försökt att sätta oss in i respektive roll. Scrum Master och produktägare är mer eller mindre självklara roller, men det blev lite mer komplicerat när vi skulle även skapa roller inom utvecklingsteamet (vilket alla, tekniskt sätt, ingår i ändå). Det som blev svårt var att man förblev på standby som kod-testare/kod-granskare. Man fick helt enkelt vänta tills personen var färdig med sin user story för att man ska kunna granska/testa deras kod. Istället för detta så bestämde vi oss eventuellt för att göra om Kod-rollerna till process-ansvar så att de t.ex. inte hade i uppgift att testa kod utan ansvar för hur vi testar kod.

Del B

Som examinatorn redan hade nämnt i en av diskussionssidan för kursen: Vi måste undvika att dela upp rollerna på det här sättet, då det ger upphov till ett horisontellt arbetssätt. Man ska inte skapa en icke-lösning till ett icke-problem. Och det var exakt det som hände, så

examinatorn har helt och hållet rätt när det kommer till denna punkt. Man ska helt enkelt ha ett vanligt utvecklingsteam utan någon sådan uppdelning.

Del C

Vi börjar med att avskaffa de två rollerna: kod-testare och kod-granskare. Nu så kan man flytta sina user stories till olika statusar, som t.ex. ”Testning” och ”Granskning”, när man är känner sig färdig med sin story. I verkligheten så kanske blir det att man istället får sin kod granskad av en annan person när man skapar en PR (”Pull request”).

Agila tillämpningar

Del A

Centralt för vår Scrum.process är vår scrum-board i Jira. Jira har hjälpt oss med att arbeta på ett metodiskt sätt och det har fungerat som ett bra stöd för när man vill bibehålla en bra struktur på sina user stories. På Jira så kan man bland annat se statistik för föregående Sprints, vilket gör det enklare för oss att kolla tillbaka på hur bra/dåligt det hade gått.

I Discord så har vi en enskild kanal för Daily Scrum, som man använder för att informera andra om hur det har gått med sina stories och hur man tänker fortsätta jobba.

Del B

Jira var ett jättebra verktyg, men det var också en del av Jiras funktioner som var förvirrande och en del funktioner som man önskade fanns. Till exempel så kunde man initialt inte tilldela en user story till två olika personer. Det var också svårt med att se föregående Sprints, och de user stories som ingick; såsom de ingick. Det skulle varit bättre om vi hade ett verktyg som var mer användarvänligt då Jira kanske funkar bättre om man lägger mycket tid på att förfina verktyget.

Daily Scrums i Discord var ännu en bra grej, men det har däremot funnits vissa stunder då man kanske inte var helt nöjd med användandet av Daily Scrum. Det är i dessa stunder som man kanske blir lite osäker om förloppet av andras stories som har blivit tilldelade till de, och att man önskar mer aktivitet.

Del C

Man kan tänka sig att använda sig av en Scrum board som är mer användarvänlig. Man kan ju söka sig fram på Google och se om det finns något som gynnar utvecklingsteamet mer. Trello är ännu ett välkänt verktyg för agila projekt.

Tvinga folk att vara aktiva i Daily Scrum kan man inte göra, däremot så kan man eventuellt under Sprint Review berätta för personen vilken konsekvens som denna inaktivitet har haft för Sprinten i dess helhet. I någon Sprint så kanske har det skett att man har behövt övergå till ett horisontellt arbetssätt, istället för ett vertikalt, och man behöver vänta på att någon annan ska vara färdig med sin user story innan man fortsätter. Detta är alltså något som man måste påpeka under de möten man har med sin grupp.

Sprint Review

Del A

Vi bestämde oss för att ha en separat Sprint Review på fredagar innan vi planerade nästa Sprint. Att ha ett sådant tillfälle har gett oss och kunden (en familjemedlem till en i gruppen) mycket nytta. Under det här tillfället så har vi kunnat reflektera över vad som vi har kvar, alltså vår backlog, jämfört med den feedback som vi fick från kunden. Kundens feedback har inte påverkat vår Definition of Done, men däremot så har varsin kund deras egna prioriteringar och preferenser. Detta kan ha inneburit att vi, vid enstaka tillfällen, undermedvetet modellerat våra acceptanskriterier efter vad vi tror kunden vill ha.

Del B

Sprint Reviews borde ha haft större fokus på kundens behov. Det fanns ett tillfälle då vi valde att prioritera user stories över andra user stories som kunden hellre skulle vilja ha med enligt feedback från föregående Sprinten.

Del C

Man ska inte jobba mot kunden, utan man ska jobba med kunden. Och för att detta ska ske så måste vi se till att vi prioriterar kundens önskemål (och de user stories som man får därav)

över allt annat. Det spelar ingen roll vad vi tror att kunden vill ha, när vi faktiskt vet vad kunden vill ha enligt den feedback som vi har fått från honom. Sedan så finns det visserligen tillfällen i verkligheten där kunden ändrar sig hela tiden, men det är ett problem som man löser där och då beroende på situationen.

Verktyg för inlärning

Del A

Vi har satt upp ett dokument för en Git-lathund. Denna lathund kan man använda sig av om man behöver en ”quick start” till Git och hur Git fungerar. Ett dokument för testning har även skapats, däri så kan man leta efter information om hur koden bör testas. Till slut så har vi även dokumenterat strukturen av produkten men också hur man skall skapa kommentarer för sin kod.

Kunskapen som vi har fått från artiklar om Scrum har hjälpt oss med att använda Jira, vår Scrum board, på ett bättre sätt.

Vi har också kunnat kommunicera över Discord när vi stötte på eventuella problem. Discord har varit till stor hjälp när man har kört fast. Man kan helt enkelt skriva på Discord, sedan så kommer någon till slut och försöker hjälpa dig.

Del B

Dokumentation som är mer strukturerat och lite mindre svårläst.

Del C

För att uppnå struktur i dokumentationen så måste man sätta sig och jobba med den en del. Man behöver väga mellan fördelarna och nackdelarna med att spendera en längre tid på att förbättra och förfina dokumentationen: Kommer man att hinna med sina user stories om man ägnar mycket tid på dokumentationen?

Kurslitteratur

Del A

Vi har tagit del av alla föreläsningar, workshops och kurslitteratur. Innebörden av de olika koncepterna som har gåtts igenom och dess nytta anser vi oss ha förstått; Åtminstone efter intern diskussion eller utforskning. Vi har också varit inne på diskussionssidan för att se om andra har stött på likadana problem och vad kursansvarige/examinatorn har att säga.

Del B

Vi kunde ha varit mer delaktiga på diskussionssidan. Vi har själva sett hur väl det har fungerat för andra att ställa frågor, så vi skulle lika väl också kunna göra det med tanke på att den kursansvarige/examinatorn ändå svarar aktivt på frågor.

Del C

Varje gång som vi stöter på ett större problem så kan vi direkt göra ett inlägg på diskussionssidan. Även om vår tilldelade TA kommer eventuellt att kunna hjälpa oss med våra problem, så finns det kanske andra som har likadan problem som oss. Eller så kommer någon med en bättre lösning än TAn, vem vet?