**Risikoanalyse**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Risici | Risiko | Sandsynlighed | Konsekvens | Risikotal | Revideret  sandsynlighed | Revideret  konsekvens |
| 1 | Mangel på underviser | Lav | 30% | 25 | 7,5% |  |  |
| 2 | Mangel på enkelt studerende | Høj | 95% | 10 | 9,5% | 75% | 10 |
| 3 | Mangel på flere studerende | Mellem | 35% | 65 | 22,75% |  |  |
| 4 | Problemer med udviklingsmiljø | Lav | 15% | 40 | 6% |  |  |
| 5 | Tab af data | Lav | 5% | 90 | 4,5% | 5% | 30 |
| 6 | Gruppekonflikt | Mellem | 35% | 30 | 10,5% |  |  |
| 7 | Mangel på motivation (enkelt) | Høj | 90% | 10 | 9% | 80% | 10 |
| 8 | Mangel på motivation (flere) | Mellem | 35% | 75 | 26,25% | 25% | 75 |
| 9 | Mangler og fejl i artefakter | Høj | 85% | 80 | 68% | 35% | 40 |
| 10 | Kundekrav / mangel på erfaring | Høj | 90% | 25 | 22,5% | 20% | 15 |
| 11 | Mangel på kommunikation | Høj | 75% | 70 | 52,5% | 20% | 20 |
| 12 | Disponering af tiden | Mellem | 60% | 50 | 30% |  |  |
| 13 | Uoverensstemmelse mellem udvikler og kunde | Lav | 10% | 80 | 8% | 10% | 20 |
| 14 | Enkelt person der er fagligt bagefter | Høj | 90% | 10 | 9% | 30% | 5 |
| 15 | Flere personer der er fagligt bagefter | Mellem | 60% | 80 | 48% | 30% | 40 |
| 16 | Problemer med eksterne systemer | Lav | 15% | 95 | 14,25% | 5% | 95 |
| 17 | Mangel på brugervenlighed | Lav | 15% | 85 | 12,75% | 5% | 85 |

*Risiko: lav/mellem/høj Sandsynlighed: 1-100% Konsekvens: 1-100 prioritet: sandsynlighed\*konsekvens*

## 

## 

## **Risikobeskrivelser**

**1. Mangel på underviser:** Underviser er forhindret i at møde op af forskellige grunde eller har undervisning / anden aktivitet andet sted.

**2 og 3. Mangel på en enkelt/flere studerende:** Studerende er forhindret i at møde op grundet transportproblemer, sygedage. Dovenskab og slendrian

**4. Problemer med udviklingsmiljø:** Problemer med deling af data, manglende programmer og andre problemer, i udviklingsmiljøet.

**5. Tab af data:** Tab af artefakter grundet nedbrud, korruption af filer og konflikter i git.

**6. Gruppekonflikt:** Uenighed omkring hvordan konkrete opgaver skal løses

**7 og 8. Mangel på motivation:** Tab af motivation grundet træthed, kedsomhed osv.

**9. Mangel på sammenhængen i artefakter:** Artefakter stemmer ikke overens. fx. UseCase-diagrammer og domain-model matcher ikke.

**10. Mangel på erfaring:** Studerende har ikke den nødvendige erfaring med projekter og har en manglende viden omkring UP og diverse artefakter.

**11. Mangel på kommunikation:** Fejl der opstår på grund af dårlig eller manglende kommunikation i blandt teamet eller mellem teamet og kunden.

**12. Disponering af tid:** Kan være på grund af andre risici, såsom fejl i artefakter, dårlig kommunikation, dårlig planlægning, overraskelser osv.

**13. Uoverensstemmelser mellem udvikler og kunde**: Vores løsning er ikke hvad kunden havde forestillet sig.

**14 og 15. fagligt bagefter**: Udvikler/udviklere har ikk det det faglige niveau til at løse en konkret opgave eller dele af projektet

**16. Problemer med eksterne systemer:** Teamet kan ikke dele deres implementation med hinanden. Databasen, GitHub, Visio eller lignende er nede.

**17. Mangel på brugervenlighed:** Programmet er ikke brugervenligt nok, efter kundens behov. (For svært at bruge eller for uoverskueligt)

## **Imødegåelsesstrategier**

**2. Mangel på enkelt studerende (Høj sandsynlighed):**

Mitigation: At uddele ansvar til alle medlemmer og holde hinanden motiveret

Monitoring: Føre fravær

Management: Tage kontakt til dem med højt fravær

**5. Tab af data (Høj konsekvens):**   
Mitigation: Vi forsøger at bruge github til at gemme alle artefakter og alt koden

Monitoring: hvis en mangler info til udarbejde af artifakt, tages kontakt til projektlederen

Management: Forsøg at genskabe tabt data, ellers udarbejd dem igen (gerne dem der lavede det i første omgang)

**7. Mangel på motivation ved en enkelt studerende (Høj sandsynlighed):**

Mitigation: Ved at opmuntre og hjælpe hinanden hvis der opstår problemer.

Monitoring: Disponering af tid i sammenligning med faktisk tid brugt på udførelse + kvalitet af artefakt

Management: giver opsang og forklar hvorfor det er vigtigt

**8. Mangel på motivation ved flere studerende (Høj konsekvens, høj risikotal):**Mitigation: Ved at opmuntre og hjælpe hinanden hvis der opstår problemer.

Monitoring: Disponering af tid i sammenligning med faktisk tid brugt på udførelse + kvalitet af artefakt

Management: tage et fællesmøde om at vi skal tage os lidt sammen

**9. Mangler og fejl i artefakter (Høj sandsynlighed, konsekvens og risikotal):**

Mitigation: Vi skal være opmærksomme på at benytte korrekt UML notation og bruge givet struktur på de forskellige artefakter.

Monitoring: Der kigges fælles review punkter igennem for check af fejl og mangler

Management: retter eventuelle fejl og mangler

**10. Kundekrav / Mangel på erfaring(Høj sandsynlighed):**Mitigation: Vær sikker på der er overensstemmelse mellem kundens forventninger og datamatikerens forståelse

Monitoring: Hvis man bliver i tvivl om hvad det er der præcist skal laves i givet artefakt

Management: hvis nødvendigt, tag kontakt til kunden og undersøg løsning på problemer

**11. Mangel på kommunikation (Høj Sandsynlighed og Konsekvens):**

Mitigation: Imødegås ved at have en klar projektplan og en klar fordeling af roller og opgaver. Ved at have en projektleder der har det fulde overblik over projektet og holder folk i gang.

Monitoring: Hvis folk laver det samme eller er i tvivl om hvilke opgaver de skal udføre.

Management: Tage et møde og få styr på opgaverne og hvad folk laver. Blive enige om nogle klare linjer.

**13. Uoverensstemmelser mellem udvikler og kunde (Høj Konsekvens)**:

Mitigation: Imødegås ved at afholde adskillige møder med kunden for at få afstemt forventninger ved at stille relevante spørgsmål

Monitoring: Hvis kunden er uenig i vores løsning

Management: Kunden har altid ret. Vi vil straks tilpasse vores løsning til kundens ønske

**14. Enkelte personer der er fagligt bagefter (Høj Sandsynlighed)**

Mitigation: Personen skal læse op på den viden man ikke har og sørge for at være forberedt til næste dag. Og sætte folk til at arbejde med de opgaver de har bedre styr på og viden om. Man kan også sætte dem sammen med folk der har viden, så begge får noget ud af det.

Monitoring: Hvis en person ikke har den nødvendige viden til at løse en opgave.

Management: Sætte personen til at læse op og få folk med viden til at løse opgaven.

**15. Flere personer der er fagligt bagefter (Høj konsekvens)**

Mitigation: Man skal læse op på den viden man ikke har og sørge for at være forberedt til næste dag. Og sætte folk til at arbejde med de opgaver de har bedre styr på og viden om. Man kan også sætte dem sammen med folk der har viden, så begge får noget ud af det.

Monitoring: Hvis flere personer ikke kan løse en opgave grundet manglende viden.

Management: Sætte folk til at læse op og få folk med viden til at løse opgaven.

**16. Problemer med eksterne systemer (Høj Konsekvens)**

Mitigation: Man kan ikke forhindre nedbrud af eksterne systemer.

Monitoring: Man kan ikke overvåge det som sådan, men vi vil straks være klar over det hvis problemet opstår

Management: Man vil benytte andre redskaber til at dele vores implementation og artefakter og få givet eksternt system til at fungere igen hvis muligt

**17. Mangel på brugervenlighed (Høj Konsekvens)**

Mitigation: Ved hjælp af nøje forberedte møder med kunden hvor man sørger for at stille relevante spørgsmål for at få den feedback man har brug for. Løbende brugertests.

Monitoring: Ved at analysere brugertests kan man se om man har et problem med brugervenligheden i vores system

Management: Tilføre flere ressourcer for at forsøge at løse problemet, da brugervenlighed er vigtigt.