Metodevalg

* Match projektets mål med vilkår og betingelser
  + Team størrelse
    - Teamet består af 4 personer
  + Teamorganisering
    - Vi er co-located, samlet i samme lokale
  + Konsekvens af fejl
    - ingen kritiske konsekvenser - mest kritiske, bruger data bliver synligt
    - mindre kritisk bruger mister indtastet data
    - brugere kan miste lysten til at bruge systemet hvis de opdager det indeholder fejl
  + Kunder
    - Ingen reelle kunder.
    - Mulige kunder:
      * Fitnessvirksomheder
      * Skoler
      * Enkeltpersoner der matcher persona, og som download via appstore
  + Krav
    - Licens til forskellige app-stores
    - Sikkerhed i forhold til brugerprofiler og brugerdata
  + Arkitekttur/teknologi
    - En webklient i ASP.NET, samt en serverside backend hvor applikationen er deployed
  + Refactoring
    - “Kunsten af skære det unødvendige væk”
    - Ingen avatar, som kan pyntes og udbygges.
    - Ingen shop, hvor man kan købe items, boost, avatarpynt mv.
  + Primært mål
    - At oprette en applikation, som inspirerer og motiverer til motion i dagligdagen.
    - At konceptet kan udviddes, så målgrupppen udviddes til at omhandle organisationer
* Plot værdier ind i Boehms projektkort
  + Personnel:
    - 40% Level1 / 15% level 2 og 3 [Alle udviklere er Level 1A]
  + Dynamism:
    - 5% per måned
  + Culture:
    - 25% - Order
  + Size:
    - 4
  + Criticality:
    - Comfort
  + Tilføj evt. yderligere akser
* Overvej risici
* Vælg metode eller metodekombination der giver færrest risici
* Lav planen

Risikoanalyse.

|  |  |
| --- | --- |
| **Risici** | **Effekt** |
| E-Tech. Usikkerhed med Teknologien | 3 |
| E-Coord. Interessanter der skal koordineres | 0 |
| E-Cmplx. Systemets kompleksitet | 2 |
| A-Scale. Størrelse og kritiskhed | 1 |
| A-YAGNI. Simple design eller YAGNI | 2 |
| A-Churn. Udskiftning af udviklere | 1 |
| A-Skill. Ikke nok uddannede folk til at arbejde agilt | 1 |
| P-Change. Mange ændringer | 1 |
| P-Speed. Behov for hurtige resultater | 2 |
| P-Emerge. Uklare krav | 2 |
| P-Skill. Mangel på specialister | 2 |

**Metodevalg**

Ud fra risikoanalysen fremgår at en plandreven tilgang vil medføre størst risici. Derfor vil det være optimalt at vælge en plandreven tilgang. Det ses dog, at der er nogle risici forbundet med en agil tilgang. Derfor har vi lavet mitigation på de 5 største risici der er projektet.

**Risikostyring**

Risikoliste

1. Webservice
2. Entity Framework
3. Databasesikkerhed
   1. Hashing af databaser
   2. Personligdatasikkerhed
4. Brugervenlighed
5. Test-First Development
6. ASP.NET som teknologi
7. UI design af websiden
   1. HTML/CSS
   2. Bootstrap
8. Android kompabilitet
9. Metodekendskab (agile, plandreven alt efter hvad vi vælger)
   1. Arbejdsfordeling
   2. Ansvarsfordeling
10. Code ownership

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Beskrivelse | Sandsynlighed | Effekt | Ranking |
| 1 | Problemer med Kodning og opsætning af Webservice | 9 | 10 | 90 |
| 2 | der kan opstå problemer med at bruge Entity frameworket, pga. manglende erfaring | 5 | 9 | 45 |
| 3 | manglende erfaring med at bruge sikkerhed i forbindelse med brugerdata | 9 | 3 | 27 |
| 4 | der skal tænkes brugervenlighed ind i alle dele af projektet | 3 | 8 | 24 |
| 5 | Der kan opstå problemer med at kode test-first da det er nyt for udviklere | 5 | 4 | 20 |
| 6 | ASP.NET er en model der ikke er kendt af udviklergruppen | 7 | 9 | 63 |
| 7 | Som backendudviklere er det ikke vores stærke side at kode en poleret UI. | 10 | 9 | 90 |
| 8 | udviklerne har ingen erfaring med at konsumere en webservice fra en Android cient | 9 | 8 | 72 |
| 9 | Afhængig af hvilken arbejdsmetode der vælges kan metodekendskab få indflydelse på hvor nemt/svært det bliver at arbejde efter metoden | 5 | 4 | 20 |
| 10 | hvis en udvikler bliver syg eller lign. har de andre udviklere så den nødvendige viden der skal til for at arbejde videre | 1 | 8 | 8 |

**Mitigation**

|  |  |  |
| --- | --- | --- |
| Risiko Beskrivelse | Ranking | Mitigation |
| 1. Problemer med Kodning og opsætning af Webservice | 90 | Kode simpel  hello world prototype af webservice |
| 7. Som backendudviklere er det ikke vores stærke side at kode en poleret UI. | 90 | hyre en fantastisk god grafisk designer, som vil lave det lækkert for ingen penge |
| 8. udviklerne har ingen erfaring med at konsumere en webservice fra en Android cient | 72 | kode simpel hello world android client som konsumerer webservicen |
| 6. ASP.NET er en model der ikke er kendt af udviklergruppen | 63 | kode simpel hello world prototype  i ASP.NET |
| 5. der kan opstå problemer med at bruge Entity frameworket, pga. manglende erfaring | 45 | kode simpel  prototype med oprettelse og opdatering af database med code-First |