

Verksamhetsanalys, Planering och Utveckling av en webbtjänst

Tomas Kindstedt

| | |
|---|--|
| EXAMENSARBETE | |
| Arcada | |
| | |
| Utbildningsprogram: | Informations- och medieteknik |
| | |
| Identifikationsnummer: | 4170 |
| Författare: | Tomas Kindstedt |
| Arbetets namn: | Verksamhetsanalys, Planering och Utveckling av en webbtjänst |
| Handledare (Arcada): | Magnus Westerlund |
| | |
| Uppdragsgivare: | R5 Motion och Hälsa |
| | |
| <p>Sammandrag:</p> <p>Arbetet behandlar utvecklingsprocessen av programmet SECTMA, ett cirkelträningsprogram designat för människor som utövar kampsporter. Programmet är designat för alla konditionsnivåer från toppidrottare till de som endast vill förbättra sin kondition. Arbetet är tudelat, i den första delen går jag igenom teori för STOF och Agile för att skapa en grund och en förståelse för hur jag vill utveckla programmet. Den andra delen följer med utvecklingsprocessen stegvis.</p> <p>STOF erbjuder ett ramverk för att analysera en verksamhetsmodell, som ger oss en grund för projektet. De olika områden som STOF tar upp är: Service, Teknologi, Organisation och Finans. Varje delområde går igenom skilt och man ställer frågor och ger svar på dem för att man skall begrunda alla väsentliga frågor.</p> <p>Utvecklingsmodellen som används är Agile, ett sätt att utveckla mjukvarubaserade tjänster i cykler. Hur kravhantering fungerar i agila utvecklingsmodeller tas också upp. Själva utvecklingsprocessen beskrivs steg för steg, allt från planeringen till programmeringsfas, som förklaras i fyra olika iterationer. Tjänsten skall vara lätt att använda och skall kunna fungera på så många plattformar som möjligt. En enkel och klar design är också viktig.</p> <p>Till sist reflekterar jag över de resultat jag nått i utvecklingsprocessen. Jag sammanfattar mina tankar om Agile och STOF, samt reflekterar över hur de fungerade tillsammans för att utveckla produkten.</p> | |
| Nyckelord: | Agile, STOF, programutveckling, mobila program, webbapplikation, cirkelträning, webbtjänst |
| Sidantal: | 36 |
| Språk: | Svenska |
| Datum för godkännande: | 13.6.2013 |

| | |
|---|--|
| DEGREE THESIS | |
| Arcada | |
| Degree Programme: | Information- and Mediatechnology |
| Identification number: | 4170 |
| Author: | Tomas Kindstedt |
| Title: | Business Analysis, Planning and Development of a web service |
| Supervisor (Arcada): | Magnus Westerlund |
| Commissioned by: | R5 Athletics and Health |
| <p>Abstract:</p> <p>The work deals with the development of SECTMA, a circle training program designed for people who practice martial arts. The program is designed for all fitness levels from top athletes to those who just want to improve their fitness level. The work is divided into two parts; in the first part I go through theory for STOF and Agile to create a foundation and an understanding of how I want to develop the program. The second part follows the development stages.</p> <p>STOF offers a framework to analyze a development model; this gives us a basis for the project. The different areas STOF addresses are: Service, Technology, Organization and Finance. Each area is reviewed separately and I ask questions and give answers to the questions in order to go through all the issues.</p> <p>The development model used is called Agile, a way to develop services in cycles. How requirement management works in agile development models is also discussed. The development process is described step by step, from planning to each of the programming phases, which are explained in four different iterations. The service should be easy to use and be able to work on as many platforms as possible. A simple and clear design is also important.</p> <p>Finally, I reflect on the results I got from the development process. I summarize my thoughts on Agile and STOF and reflect on how they worked together to develop the product.</p> | |
| Keywords: | Agile, STOF, program development, mobile programs, web application, circle training, web service |
| Number of pages: | 36 |
| Language: | Swedish |
| Date of acceptance: | 13.6.2013 |

INNEHÅLL

| | | |
|----------|--|-----------|
| 1 | Inledning..... | 7 |
| 1.1 | Bakgrund..... | 7 |
| 1.2 | Mål..... | 8 |
| 1.3 | Avgränsning..... | 8 |
| 2 | STOF..... | 9 |
| 2.1 | Service..... | 10 |
| 2.1.1 | <i>Analys</i> | 11 |
| 2.2 | Teknologi..... | 12 |
| 2.2.1 | <i>Analys</i> | 13 |
| 2.3 | Organisation..... | 14 |
| 2.3.1 | <i>Analys</i> | 15 |
| 2.4 | Finans..... | 15 |
| 2.4.1 | <i>Analys</i> | 15 |
| 3 | AGILE..... | 16 |
| 3.1 | Kravhantering i Agile..... | 17 |
| 4 | Utveckling och helhetsdesign..... | 18 |
| 4.1 | Planering..... | 18 |
| 4.1.1 | <i>Krav</i> | 19 |
| 4.1.2 | <i>Design</i> | 19 |
| 4.1.3 | <i>Programutveckling</i> | 21 |
| 4.2 | Uppbyggnad..... | 21 |
| 4.2.1 | <i>Iteration 1</i> | 21 |
| 4.2.2 | <i>Iteration 2</i> | 24 |
| 4.2.3 | <i>Iteration 3</i> | 26 |
| 4.2.4 | <i>Iteration 4</i> | 27 |
| 5 | Presentation av prototyp..... | 29 |
| 6 | Slutord..... | 34 |
| | Källor..... | 35 |

Figurer och Tabeller

| | |
|---|----|
| Figur 1. STOF-modellen (Bouwman et. al., 2008, s. 36). | 9 |
| Figur 2. En iterativ utvecklingsmodell.(Wikipedia, 2013b). | 16 |
| Figur 3. Första versionen av designen för webbapplikationen. | 20 |
| Figur 4. R5s hemsida, som var inspirationen till designen. | 20 |
| Figur 5. En del av källkoden från första versionen av formulärsidan. | 23 |
| Figur 6. Bild på första versionen av filtersidan. | 24 |
| Figur 7. jQuery kod för hur slideToggle funktionen fungerar. | 25 |
| Figur 8. Första versionen av resultatsidan. | 25 |
| Figur 9. Tabellen som rörelserna sparas i. | 27 |
| Figur 10. Idiorm skickar en förfrågan till databasen och returnerar resultatet till sidan. | 28 |
| Figur 11. En bild på filtersidan från en Android tablett. | 29 |
| Figur 12. Hur resultatsidan ser ut när man öppnar den. | 30 |
| Figur 13. Exempel på hur man kan använda sidan för att se på närmare instruktioner för en rörelse. | 31 |
| Figur 14. Hur applikationen ser ut på en iOS mobiltelefon. | 32 |
| Figur 15. Hela resultatsidan sätt från en dator. | 33 |

FÖRORD

Tanken bakom arbetet var att undersöka STOFs möjligheter som ett ramverk för utveckling av mobila produkter. Det kändes som tillräckligt nytt och fräscht även om det redan varit med i några år och det var något sådant som jag var intresserad av att göra. Jag ville göra något som inte så många gjort tidigare.

Agile är den utvecklingsmetoden som jag själv prövat på några gånger och har haft bra erfarenheter med. Jag tyckte att det skulle vara bra att pröva en kombination av dessa två. R5 gav mig möjligheten att utveckla en tjänst åt dem med rätt så fria händer, detta gav mig chansen att pröva saker som jag kanske inte fått göra för andra företag.

Jag vill tacka min handledare Magnus Westerlund för hans åsikter och förslag för arbetet samt min fästmö Mikaela Oksanen som gav mig stöd när jag behövde det både för arbetet och i livet.

Tomas Kindstedt

Helsingfors

24.5.2013

1 INLEDNING

I ett samhälle där IT-tjänster av olika sorter dominerar det vardagliga livet måste vi också se till att vi inte blir för bekväma av oss och oförmögna att hålla uppe den kondition som behövs för att leva ett långt och hälsosamt liv. Även människor som har bra kondition behöver hjälp för att träna på rätt sätt. Alldeles för många tränar på fel sätt och fel muskler. Detta leder till att deras träning inte gör någon större nytta. Applikationen som företaget R5 Motion och Hälsa (hädanefter R5) vill utveckla handlar om att skapa ett cirkelträningsprogram så att man tränar på rätt sätt. Programmet skall vara till för både amatörer och professionella kampsportsidrottare. Applikationen är en webbtjänst som ger information om vad man skall träna under olika förhållanden och vid olika tidpunkter. Applikationen ger några rörelser man skall träna och instruktioner hur man skall utföra rörelserna.

Min uppgift är att samarbeta med R5 och använda deras expertis för att utveckla en tjänst som hjälper människor på alla konditionsnivåer. Tjänsten kommer att heta SECTMA som står för Speed Endurance Circuit Training for Martial Arts. Tanken med slutarbetet är att skriva om hur man kan utveckla en hälsofrämjande tjänst genom att använda STOF, ett ramverk för en mobilservice verksamhetsmodell, som är utvecklat av Bouwman et. al. (2008). STOF baserar sig på fyra domäner: Service, Teknologi, Organisation och Finans. Med dessa fyra element kan man göra en helhetsanalys av tjänsten före utvecklingen inleds. Tanken är att göra detta medan jag använder en agil utvecklingsmodell där tjänstutvecklingen sker i cykler.

1.1 Bakgrund

Orsaken till arbetet handlar både om eget intresse och att jag blev kontaktad för att göra detta arbete i samarbete med några andra studeranden från Arcada vilka startade ett företag och behövde någon med IT-kunskap för att kunna förverkliga sin idé. Jag har alltid gillat samarbete emellan högskolans olika linjer så detta intresserade mig mycket. För mig var det viktigt att hitta ett projekt som intresserade mig och passade mig personligen. R5 erbjöd mig en möjlighet att göra något jag finner intressant samtidigt som

jag har en möjlighet att lära mig implementera andra personers visioner och idéer till verklighet - i det här fallet i form av en webbapplikation.

Jag är medveten om att övervikt kan bli ett problem i vårt samhälle och kommer antagligen att bli ett större problem desto bekvämare samhället blir. Idén med applikationen är att kombinera bekvämlighet med en vilja att träna rätt, man kan vara bekväm samtidigt som man tränar upp sin kondition.

1.2 Mål

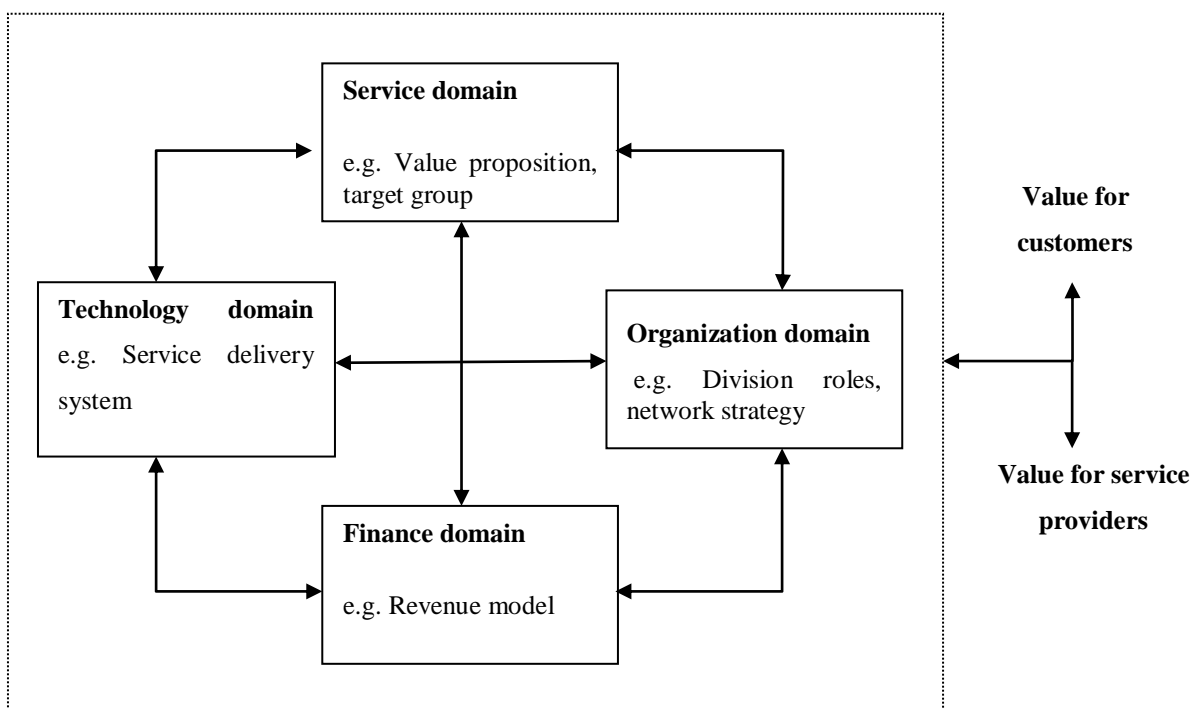
Examensarbetets mål är tudelat. I den första delen undersöker jag en verksamhetsmodell och en utvecklings modell för att utveckla applikationer för webb- och mobilplattformar. I den andra delen designar och utvecklar jag en applikation i samarbete med R5 där jag försöker använda dessa metoder i utvecklingsprocessen. Målet är alltså att skapa en webbtjänst i form av en applikation som hjälper dem som tränar olika kampsporter att träna rätt med hjälp av ett cirkelträningsprogram. Det skall också fungera på en lägre nivå för dem som endast tränar för att öka konditionen.

1.3 Avgränsning

Arbetet använder sig av två olika teorier. Tjänsteutvecklingsmodellen STOF av Bouwman et. al. (2008) som ännu är relativt ny inom branschen samt utvecklingsprocessen Agile, en benämning på en grupp utvecklingsprinciper med vissa karaktärsdrag. Agile har redan bevisats vara fungerande då flera stora företag tagit i bruk den eller en variant av denna. (IBM, 2013) I detta arbete används namnet Agile som en förkortning för agila utvecklingsprinciper. Litteraturstudierna avgränsas till dessa två metoder. Arbetet kommer att vara en helhetsanalys för tjänsten, för att se om man kan utveckla en fungerande tjänst med dessa metoder. Jag kommer också att rikta in mig på användarupplevelsen och inte koncentrera mig på den administrativa sidan ännu som t.ex. ett administrationsgränssnitt för att lägga till mera data.

2 STOF

Bouwman et. al. (2008) föreslog och evaluerade ett ramverk för att utforma en affärsmodell för informations- och kommunikationsteknik (IKT) tjänster. STOF ramverket består av fyra element: Service, Technology, Organization och Finance. Figuren nedanför visar hur de fyra elementen är relaterade till varandra inom ramverket.



Figur 1. STOF-modellen (Bouwman et. al., 2008, s. 36).

När man designar en ny, teknologibaserad, tjänst måste man se på de teknologiska aspekterna för tjänsten, men också på både de praktiska och de teoretiska problemen som existerar för att göra tjänsten lyckad både från en användares och från en tjänsteleverantörs synvinkel (Bouwman et. al., 2008). Målet är alltså att genom användningen av STOF få ihop en fungerande tjänst som skulle kunna vara konkurrenskraftig och användarvänlig. Avsikten med STOF är således att skapa en helhetsbild som gör det lättare att utveckla tjänsten genom att gå igenom alla skeden och beakta tjänstutvecklingen ur alla synvinklar som STOF tar upp. Ifall man följer dessa punkter så borde slutresultatet vara en lyckad tjänst.

2.1 Service

Serviceområdets viktigaste aspekt är att analysera tilläggsvärdet åt kunden eller användaren. Service design är en beskrivning av de värden som service leverantören har som mål att ge kunden eller användaren. Komponenter som när, var och vem som använder tjänsten samt priset på tjänsten skall också analyseras. Mobila tjänster borde stöda olika typer av användarbehov: spontana, tidskritiska arrangemang, effektivitet, underhållning och krav för att hållas mobilt. (Eriksson et. al. 2013)

Det finns många saker att tänka på ur en service synvinkel då man designar enligt STOFs normer. Målgruppen för tjänsten måste först definieras. Det kan vara en nisch tjänst som endast intresserar ett fåtal personer eller så kan det vara en mycket allmän tjänst som kan intressera alla, men kan då samtidigt inte vara lika specialiserad för den enskilda människan. (Bouwman, 2006)

Hur kan man då skapa tilläggsvärde för användarna? Många saker kan göra tjänsten värdefull för olika personer, det gäller att hitta en målgrupp för att veta hur man kan skapa tilläggsvärde för dem. En orsak varför det lönar sig att ha en målgrupp är för att man då bättre vet vad ens användare vill ha. Ifall en målgrupp saknas så kan tjänsten ofta bli utspridd över många områden och då lider eventuellt tjänstekvalitén. Det lönar sig att jämföra de tekniska möjligheterna med användarnas behov och önskemål. (Bouwman, 2006)

Hur får vi användaren att lita på tjänsten? Det uppnås bäst genom att hitta en balans mellan säkerhet samt lätthet i användandet av tjänsten. Ibland upplevs säkerhetsprocedurer som jobbiga och de skrämmer bort användare som inte är vana vid dem. Nya funktioner kan läggas till, men om användaren måste ge för mycket information om sig själv och upplever sin integritet kränkt, bör man kanske inte lägga till den nya funktionen. (Bouwman, 2006)

Vad får användaren att återvända? Första gången är allt nytt, efter en tid har man lärt sig tjänsten, vid det skedet måste man ha något som får användaren att stanna trots att personen i fråga kanske inte skulle behöva det. Hur får man användaren att bli fast i tjänsten utan att störa deras upplevelse? Man kan t.ex. hitta på något sätt att få användaren att

investera tid i tjänsten och på så sätt vilja använda den mera, eftersom de redan investerat i den (Bouwman, 2006).

2.1.1 Analys

Arbetets syfte var redan från början att fylla ett behov för en applikation av denna typ. Ett lätt sätt att hålla reda på vad och hur man skall träna samt hur många gånger man upprepar en rörelse och hur länge man tränar denna rörelse. Applikationens mål är alltså att förbättra användarens träningserfarenhet. Enligt Pine och Gilmore (1999) är funktioner som förbättrar erfarenheten, en drivkraft för en lyckad service design. En annan sak som är viktig är sättet man kan sprida ut applikationen till dem som tränar och vill använda den. Ett sätt är att sätta upp en plansch i träningslokalen med en QR-kod där användaren genast kan gå till rätt sida på internet för att lätt komma åt tjänsten. Detta ökar chansen till en större användarbas. Förutom detta bör det finnas andra sätt att komma åt applikationen så att så många användare som möjligt har en chans att hitta den ifall de är intresserade. Eftersom det är mest sannolikt att de som tränar använder applikationen, så skulle det vara viktigt att koncentrera sig på lokala ställen som ett gym eller internet-sidor som är relaterade till ämnet för att marknadsföra sig.

Det var viktigt för oss att skilja mellan de olika kampsporterna eftersom någon som utövar kickboxning och någon som utövar MMA har väldigt olika krav för vilka muskler de skall träna och vilka rörelser är bättre att utöva ju närmare ens match man kommer. Det var också viktigt att kunna välja sin konditionsklass baserad på ett antal rörelser, som R5s expertis kommit fram till. Applikationen fungerar på det sättet på alla nivåer där människor tränar även om dess ursprungliga idé var att hjälpa kampsportare.

En prototyp av applikationen gjordes och testades internt för att sedan utveckla tjänsten vidare enligt de agila-utvecklingsprinciperna. Det bestämdes att inte genast lägga in alla rörelser och funktioner för enkelhetens skull och för att testen av konceptet fungerar. I varje utvecklingscykel efter detta läggs mera och mera av tjänstens funktioner med, tills man är nöjd med slutresultatet.

Tjänsten designades och planerades med tanke på en smarttelefon och att alla som använder tjänsten har en mobil internetanslutning för möjligheten att t.ex. kunna se på instruktionsvideon ifall personen behöver det för att utföra rörelsen som föreslås. Tanken

var också att tjänsten skulle fungera på så många olika plattformar som möjligt så därför bestämdes det att tjänsten skulle vara webbaserad. Frågan om kostnader för tjänsten togs också upp och då var tanken att tjänsten skulle vara ett sätt att ge mervärde för övriga tjänster som R5 redan erbjuder och via dem skulle användningen av tjänsten vara gratis.

2.2 Teknologi

Teknologidomänens krav baserar sig på service domänen. Teknologidesign är en beskrivning av den grundläggande organisationen av ett tekniskt system (t.ex. teknisk arkitektur, applikation och enhet) som krävs för att leverera den tjänst som framkommer från servicedomänen. (Eriksson et. al., 2013)

Teknologidomänen har precis som servicedomänen en del frågor som bör besvaras. Hur arrangerar man ett säkert sätt att använda tjänsten och en säker kommunikation? Det gäller att hitta rätt balans mellan hur lätt det är att använda tjänsten mot hur lätt det går att missbruka tjänsten. (Bouwman, 2006)

Hur kommer man fram till den kvalitet man vill ha på tjänsten? Här gäller det att balansera kvalitet mot kostnader. Allt går att göra perfekt, men vid vilket skede är kostnaderna för stora för att det inte skall löna sig. (Bouwman, 2006)

Integration till gamla system ifall så krävs. Hur kan man integrera de nya systemen? Vad krävs för att man skulle kunna göra detta? Vid något skede måste man jämföra kostnader mot flexibilitet. Att integrera tjänster är oftast en bra lösning, men om kostnaderna blir för höga lönar det sig inte alltid. Olika plattformar och system tolkar saker på olika sätt. Om tjänsten skall fungera i alla system måste man ofta göra om den för olika apparater. Det är ofta lätt att utveckla tjänster genom att använda programmeringsspråk som stöder många olika plattformar och på det sättet slippa göra om samma produkt. (Bouwman, 2006)

Hurudan tillgänglighet för en användare bör det vara? En sak som debatteras på en sådan punkt är öppna mot stängda system. Öppna system används då alla kan komma åt

tjänsten antingen gratis eller mot betalning. Stängda system används mest för interna bruk inom företag. En annan sak som är relevant att ta upp är debatten om system som alltid kräver en internetanslutning. En del tjänster behöver internet, medan andra inte gör det. Möjligheten att göra en kompromisslösning där man kan ladda ner en del eller all information för produkten så att den kan fungera på någon nivå. (Bouwman, 2006)

Hur skall man hantera användarprofiler? En möjlighet är att ha ett öppet program för alla användare, då behöver man inga användarprofiler alls. Om man trots det vill ha profiler så hur skall de hanteras? De kan skapas automatiskt eller så kan man ha en person som skapar varje användarprofil skilt för sig för att förbättra säkerheten, följa upp användaren eller begränsa vem som kan använda tjänsten. (Bouwman, 2006)

2.2.1 Analys

Som tidigare nämnts är ett av de största målen med projektet att så många som möjligt lätt skall få tillgång till applikationen och att primär fokus ligger i att ha en webbapplikation som man har tillgång till från en smarttelefon. Detta kommer att kräva en webbserver samt en databas för att spara data om alla tillgängliga rörelser. Instruktionsfilerna kommer att sparas på youtube för att lätt kunna ge tillgång till dem på många olika sätt och på samma gång spara på kostnader för lagring.

De tekniska verktygen som används är följande:

- HTML: HyperText Markup Language är det språk man använder för att bygga upp websidor.
- jQuery: Ett JavaScript bibliotek som används för att köra scripts på webbsidan för att förbättra funktionaliteten och göra sidan mera avancerad än vad man kan göra med HTML. I denna tjänst används jQuery främst för att skapa animationer som gör sidan snyggare och samtidigt ger det ett bättre stöd för de flesta webbläsarna än vad CSS kan erbjuda.
- CSS: Cascading Style Sheet används för att skilja åt designen av sidan och innehållet av sidan, som normalt skrivs i HTML.

- MySQL: Ett databassystem som låter oss spara den information vi behöver i databaser för att sedan lätt kunna plocka ut den information som vi behöver till applikationen.
- PHP: Används för att ta ut information från databasen och använda den i applikationen.
- Idiorm: Ett lager för PHP som gör det lättare att få ut data från databasen. Det är ett objektorienterat sätt att få ut data med minimala konfigurationsbehov genom att konvertera databasmodeller till klasser, tabellrader till objekt samt celler till objekt-attribut. Idiorm används för att underlätta användningen av PHP i många fall för mindre hemsidor. (Idiorm documentation, 2013)
- Youtube: Googles kända videoservice. Används i detta fall för att lätt lagra och sprida instruktionsvideon.

2.3 Organisation

Organisationsdomänen handlar om att välja rätt partners, fördela roller som behövs samt att utveckla sätt att styra och organisera aktiviteter. I det här fallet är organisationsdelen av projektet rätt så enkel eftersom det är ett litet företag och endast en person jobbar heltid på detta projekt. (de Reuver et. al., 2010)

För ett större företag skulle andra saker gälla. Vilka partners bör man välja? Vilken partner är den som bidrar med de mest kritiska elementen för att tjänsten skall fungera? Det gäller att gå igenom gamla partners och nya potentiella partners, eller om man inte hittar det man söker efter från en partner, får man göra det själv. (Bouwman, 2006)

Hur öppet är projektet och hur långt in får andra partners komma? En balans mellan öppenhet och antalet användare måste hittas. Om tjänsten är gratis är det en mycket lägre tröskel att börja använda tjänsten, men då är det också svårare att förtjäna pengar. Om projektet är öppet måste ändå någon leda det för att inte skapa kaos, oftast är det en av partnererna som håller i tråden medan de andra stöder denna. Vid något skede måste man dock överväga vad som gynnar de enskilda partnererna mest och vad som befrämjar hela nätverket mest. (Bouwman, 2006)

Hur komplext är nätverket? Både på en teknologisk och på en teknisk nivå måste man komma underfund med hur man kan minska komplikationer och effektivera de kritiska resurserna och möjligheterna. (Bouwman, 2006)

2.3.1 Analys

Vi behöver en projektledare, en utvecklare samt en person med kunskap om de olika sätten att träna. Projektledaren och utvecklaren kommer att vara samma person eftersom projektet är så litet. Personen med kunskapen är från R5 och kommer samtidigt att representera kunden och deras perspektiv. Efter att tjänstutvecklingen är klar så kommer R5 att ta över med sin organisation.

2.4 Finans

Finansdomänens uppgift är att definiera hur mycket resurser företaget kan bidra med för att utveckla tjänsten och är därför ofta en absolut gräns för hur länge och hur många som kan vara med och utveckla tjänsten. Den finansiella designen innebär en analys av hur projektet avser att skapa en vinst för tjänsten och företaget i allmänhet. Det måste också funderas över hur risken, investeringen och intäkterna fördelas mellan de inblandade parterna. Det hela handlar om att skapa en intäktslogik för företaget och tjänsten i relation till uppskattade kostnader. (Eriksson et. al., 2013)

2.4.1 Analys

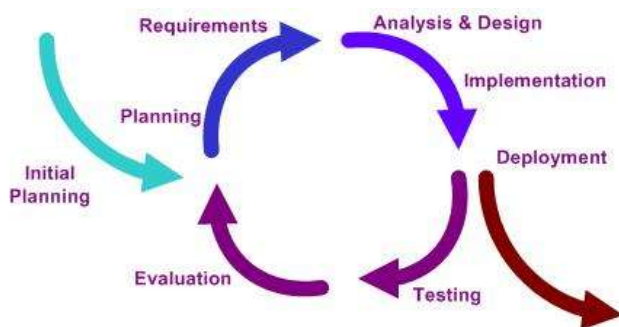
Det har funnits några tankar kring hur man skulle kunna göra projektet finansiellt genomförbart. Eftersom det är ett nytt företag som inte har mycket pengar eller stor budget är första prioriteringen att hålla projektets kostnader låga. Detta uppnås genom att använda en studerande som gör ett slutarbete om saken. Företaget hyr redan en webbserver så kostnaderna för driften kommer inte att stiga även om man lägger till en dylik applikation.

Några olika möjligheter diskuterades för att kunna förtjäna pengar på projektet. Första alternativet är att inte alls ta betalt för tjänsten utan istället göra den tillgänglig som en extra tjänst till dem som redan använder sig av R5s tjänster. Det andra alternativet är att

öppet marknadsföra tjänsten och genom användning av reklam förtjäna på tjänsten. Det tredje alternativet är att marknadsföra tjänsten för att visa att konceptet fungerar och sedan sälja den till ett större företag.

3 AGILE

Agile är en grupp av utvecklingsprinciper för mjukvara som baserar sig på iterativ och inkrementell utveckling (se figur 2). Det underlättar anpassningsbar planering, evolutionär utveckling och leverans, en s.k. timeboxed metod där tiden i projektet delas upp i lådor av en viss tidsperiod (t.ex. ett dygn eller en vecka), samt ett snabbt och flexibelt svar till ändringar. (Agilemanifesto, 2001)



Figur 2. En iterativ utvecklingsmodell.(Wikipedia, 2013b).

Inkrementell mjukvaruutveckling började användas redan i mitten av 50-talet, men det var först under mitten av 90-talet som tidiga versioner av Agile började framkomma. Det hela började som en reaktion till tunga utvecklingsprocesser som t.ex. Vattenfallmetoden där utvecklingen går stegvis och när alla steg genomgåtts är tjänsten klar. Detta gör det mycket svårare och ofta dyrare att gå tillbaka för att göra ändringar. Förespråkare för Agile påstår att detta är ett steg tillbaka mot den processen som ursprungligen användes för att utveckla mjukvara. Tidiga adaptationer av Agile är bl.a. Scrum och Rational Unified Process (RUP) (Wikipedia, 2013a). År 2001 träffades 17 mjukvaruutvecklare för att diskutera lätta utvecklingsmetoder (Agilemanifesto, 2001). Detta ledde till det s.k. agila manifestet som beskriver de tillvägagångssätt som nuförtiden kallas agil mjukvaruutveckling. Samtidigt bildade en del av utvecklarna den agila alliansen, en organisation som främjar mjukvaruutveckling enligt manifestets principer. (Agilemanifesto, 2001)

Det finns många olika typer av agila utvecklingsprinciper. De flesta stödjer utveckling, samarbete, samspel samt en adaptiv process genom hela projektet. Agile bryter ner uppgifter i små tidsperioder på t.ex. en vecka där man planerar, designar, kodar och testat allt under en tidscykel. Ofta har man också en vag helhetsbild av slutprodukten, men eftersom element ändrar hela tiden bör man inte fästa sig allt för mycket vid den. Varje iteration ger inte nödvändigtvis en ny tjänst eller ens en uppdatering av en tjänst, men det är ändå ett steg framåt i utvecklingen då målet i slutet av varje iteration är att ge ut en nyare version av applikationen med minimal mängd fel. (Beck, K. 1999)

Varje grupps storlek i en agil utvecklingsmiljö är rätt liten (under 10 personer), detta för att göra kommunikation och samarbete så bra som möjligt. Kommunikationen sker oftast via diskussioner med varandra istället för att skicka papper med instruktioner. Det är också meningen att man ordnar korta dagliga möten inom gruppen så att varje medlem i gruppen kan berätta vad de gjort föregående dag och vad de kommer att göra till nästa. Dessa möten kallas ofta för ”stand-up” möten eftersom de inte bör ta mera än 15 minuter och om man står upp under dem så håller man ofta denna tidsram. Oftast finns det även en person som representerar kunden som tjänsten utvecklas för, så att man får ett annat perspektiv på tjänsten än utvecklarens. (Planbox, 2013)

De flesta företag som använder en agil princip för att utveckla mjukvara har sin egen version av hur det fungerar. Alla har olika behov och Agile är lätt att skräddarsy för varje företag så att de får ut så mycket som möjligt av processen. Agile passar bra för både små och stora företag, den anpassas enligt vilken typ av projekt det är och hur stort projektet är. (Wikipedia, 2013a) Jag beskriver min användning av Agile i ett senare kapitel då jag behandlar utvecklingen av tjänsten i praktiken.

3.1 Kravhantering i Agile

Kravhantering i en Agile process går ut på att effektivera arbetet så mycket som möjligt. Man bör minska sin orderstock så att det inte släpar efter mer än två iterationer. Hur vet man hur många krav som hinner utföras inom en iteration? Det gäller att gå tillbaka till de senaste iterationerna för att se hur många krav som utförts då och på det sättet kan

man beräkna hur många som kan utföras under nästa iteration. Om man håller sig till detta borde inte den ogjorda arbetsmängden bli alltför stor. Om orderstocken blir för stor så slösas det tid på att hantera den stora mängden information över vad som ännu är ogjort. (IBM, 2008)

Kraven beskrivs effektivt, men med tillräckliga detaljer. Ifall kravet inte är tillräckligt detaljrikt beskrivet så har utvecklarna, testarna och andra svårt att förstå exakt hur man vill att det skall fungera. Problemet med att använda s.k. user stories är att de ofta inte är tillräckligt detaljrika och därför kan use case metoden vara en bättre lösning för Agile. (IBM, 2008)

Varje partner har olika intressen. Genom att skilja på de berörda parterna i olika grupper, beroende på hur stort intresse de har i mjukvaran, kan man sedan dokumentera de olika affärsmålen så att de gör de berörda parterna nöjda. Man kan också uppdatera parterna genom att visa hur mål uppfylls i iterationerna och reflektera över målen i slutet på varje iteration. På det sättet kan man hålla deras intresse för tjänsten. (IBM, 2008)

4 UTVECKLING OCH HELHETSDESIGN

Kapitlet går igenom hela utvecklingsprocessen från början till slut. Varje iteration beskrivs skilt för sig och varje skede av utvecklingen är uppdelat i sin skilda del. Jag valde att presentera på detta sätt eftersom jag anser att det ger en så klar bild som möjligt av utvecklingsprocessen.

4.1 Planering

Planeringsskedet består av tre steg: Krav, design och uppbyggnad. Först funderar man vad som behövs för att programmet skall vara användbart och vilket extra material man vill ha med. I designskedet funderar man hur applikationen skulle kunna se ut samt hur man kan integrera utseendet med tidigare tjänster från företaget. I utvecklingsdelen analyserar man hur man skall utveckla applikationen och vad som behövs för detta.

4.1.1 Krav

Det största kravet för applikationen var att det skulle fungera på många olika plattformar, men huvudplattformen skulle vara en mobil webbläsare. Detta ledde till att applikationen utvecklades i webbformat för maximal spridning. Detta betyder också att tjänsten skall vara skalbar till alla typer av apparater.

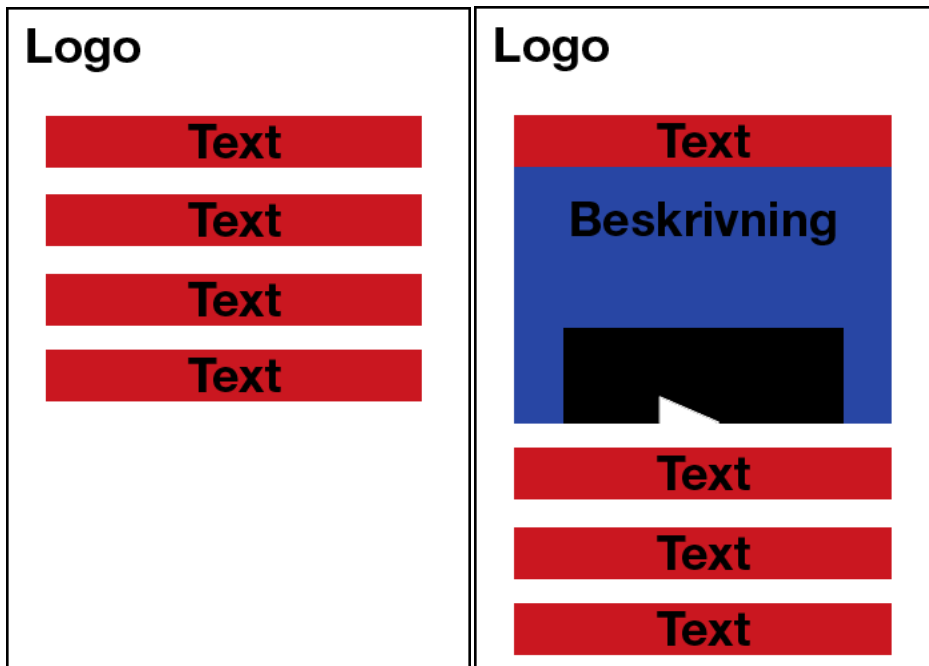
Produktions- och driftkostnaderna skulle också vara låga eftersom det är ett nystartat företag som inte har så mycket pengar. Färgvärlden skall motsvara det som R5 redan tidigare valt dvs. Rött, blått och vitt. Fonten skulle vara klar och lätt att läsa.

Applikationen bör också kunna fungera för människor med varierande konditionsnivå så att både toppidrottare och personer med dålig kondition skulle kunna använda den. Den skall också upplevas vara enkel och klar så att alla förstår vad de skall göra utan destomera handledning.

I början fanns det inte så många krav, vilket gjorde att min undersökning av SOTF stod för en stor del av grunden för arbetet så att man hade en klar bild av var man stod och vad allt man ännu måste tänka på. Jag försökte att inte låsa in mig med mina lösningar så att det lätt skulle gå att ändra ifall det krävdes. Användarvänligheten var den största prioriteringen. Det finns så många olika plattformar att det gäller att hitta något som fungerar överallt och för så många som möjligt.

4.1.2 Design

Designen har några viktiga komponenter. Den största faktorn är att det ser snyggt ut på en mobiltelefon eller tablett och det andra är att färgkombinationerna som används passar ihop med det färgschemat som R5 redan slagit fast för sig själva.



Figur 3. Första versionen av designen för webbapplikationen.



Figur 4. R5s hemsida, som var inspirationen till designen.

4.1.3 Programutveckling

Programmet utvecklas genom användning av den agila utvecklingsmetoden och eftersom det bara är en utvecklare så behövs inga möten med gruppen. Däremot träffar jag beställaren med jämna mellanrum i utvecklingsprocessen för att se att visionen och verkligheten fungerar tillsammans. Den första iterationen kommer att bestå av en grundläggande version där man bestämmer ramar för arbetet utan en större tanke på design eller funktionalitet. Efter det kommer varje iteration att vara ca tre dygn, tid för feedback samt kommunikation mellan parterna om det behövs. Rent tekniskt kommer vi att behöva en webbserver och en databas för att kunna genomföra detta.

Eftersom jag fått rätt så fria händer använder jag STOFs principer som grund för mitt arbete. På detta sätt försäkras jag mig om att det jag gör inte är helt i fel riktning från vad som behövs för användarna. Detta märks t.ex. genom att jag försöker göra produkten så enkel att använda som möjligt för att locka användaren att komma tillbaka och pröva tjänsten en gång till. Jag fick också en mycket bättre helhetsbild av tjänsten då jag gick igenom punkter från STOF med beställaren för att veta hur de vill utveckla produkten.

4.2 Uppbyggnad

Detta kapitel beskriver vad som hänt i varje skede av utvecklingen. I slutet på varje iteration är det meningen att det som ändrats eller lagts till skall fungera utan några större problem. Jag valde att presentera arbetet i iterationer eftersom jag ansåg att det var ett klart sätt att visa hur arbetet framskridigt och samtidigt visar det hur jag tänker i varje skede av utvecklingen.

4.2.1 Iteration 1

Den första versionen hade som mål att skapa ramarna för designen samt att man får en idé hur programmet skall fungera samt om det kan fungera på detta sätt. Det hela hölls så enkelt och klart som möjligt. Jag började genom att skapa ett enkelt html formulär där det fanns olika radioknappar med olika värden. För varje grupp av radioknappar gavs ett skilt namn, så man kan skilja på dem senare för noggrannare indelning. I början

var min tanke att skapa funktionerna för formuläret med ren JavaScript, men jag gav snart upp med den idén, eftersom källkoden var enklare och kortare att skriva med JavaScriptbiblioteket jQuery.

Eftersom tjänsten främst skall användas på mobiltelefoner så valde jag att rada upp de olika grupperna av radioknappar under varandra (se figur 6). För tillfället finns det två olika sorters radioknappar – numeriska knappar och litterära knappar. De litterära knapparna innehåller information om vilken kampsport du utövar medan de numeriska knapparna poängsätter din kondition genom konditionsrelaterade frågor.

I JavaScriptet (figur 5) valde jag att använda en switchfunktion för att skilja på de litterära och numeriska knapparna. Jag övervägde också möjligheten att använda en if/else sats istället för switch, men eftersom programmet blir mera komplicerat senare, ville jag spara möjligheten att lägga till flera switchar för en noggrannare indelning.

Slutligen lade jag till ett alert för testbruk så att jag snabbt kunde se om sidan fungerade som den skulle.

```

//kör funktionen när man trycker på knappen med id "getValue"
$('#getValue').click(function(){
    //level är värde för konditionsklassen som kommer senare
    var level = 0;
    //martial är värde för vilken sport du tränar för
    var martial = '';
    //För varje input radioknapp som finns
    $('input[type="radio"]').each(function(){
        //och är vald
        if($(this).is(':checked')){
            //knappar har olika namnvärden
            switch($(this).attr('name')){
                /*om värdet är martialart sparas värdet för den
                valda knappen till variabeln martial*/
                case 'martialart':
                    martial = $(this).val();
                    break;
                /*I andra fall sparas det numeriska värdet för
                knappen för senare bruk genom addition*/
                default:
                    level = level + parseInt($(this).val());
                    break;
            }
        }
    });
    //Testklass för sidans funktioner
    var level;
    if (level > 21) {
        var levelString = "Advanced";
    } else if (level < 22 && level > 17) {
        var levelString = "Above average";
    } else if (level < 18 && level > 13) {
        var levelString = "Average";
    } else if (level < 14) {
        var levelString = "Poor";
    }
    alert(martial+" "+level+" "+levelString);
});

```

Figur 5. En del av källkoden från första versionen av formulärsidan.

SECTMA R5

Select Martial Art:

- Boxing
- MMA
- Jujitsu
- Kickboxing
- All

Cooper test:

- over 3200 meters
- 3200-3000 meters
- 2999-2700 meters
- 2699-2400 meters
- under 2400 meters

Figur 6. Bild på första versionen av filtersidan.

4.2.2 Iteration 2

I planeringen av resultatsidan spelade mobilvänligheten en stor roll. Användargränssnittet måste planeras så, att det lätt kunde anpassas till ett litet utrymme, men så att den också fungerade på större ytor (t.ex. datorer och surfplattor). Man måste också tänka på att då man använder tjänsten med mobiltelefoner eller surfplattor använder man ofta fingrar istället för en mus. Detta betyder att man vill undvika alltför små ytor så att det inte blir svårt för användaren att använda tjänsten.

Då resultatsidan planerades funderade man på att användaren möjligtvis känner till en del av rörelserna. Då vore det onödigt att tvinga användaren gå igenom en massa information för att hitta det som är viktigt att veta om de rörelser som han inte känner till.

jQuery har en funktion som heter slideToggle som öppnar och stänger rutor då man trycker på ett visst område. Med CSS skapade man stora klara ytor med namnen på de olika rörelserna (hädanefter namnruta). Tilläggsinformationen placerades i rutor under namnen och efter det gömdes utrymmet från användaren. Med jQuery skapades en funktion som lyssnar efter knapptryckningar på namnrutorna.

Lyssnarens uppgift är att vänta på ett knapptryck från användaren. Då användaren trycker på en namnruta aktiveras slideToggle, vilket öppnar en ruta med tilläggsinformation smidigt genom att använda en glidanimation. Det som är bra med slideToggle är att den också stänger det valda området då man trycker på knappen en gång till. Detta ger användaren mycket mera utrymme och minimerar behovet att scrola på sidan.

```
$('.moves .title').click(function() {  
    .....  
    $(this).parent().find('.descriptions').slideToggle();  
});  
$('.moves img').click(function() {  
    .....  
    $(this).parent().find('.descriptions').slideToggle();  
});
```

Figur 7. jQuery kod för hur slideToggle funktionen fungerar.



Figur 8. Första versionen av resultatsidan.

4.2.3 Iteration 3

I iteration 1 användes jQuery för att köra en for-loop genom de olika punkterna i formuläret och poängsätta svaren enligt de förutsatta värdena. Detta ändrades i iteration 3 då man valde mellan AJAX och PHP för att välja hur man ville ta ut informationen.

Valet blev PHP eftersom informationen finns i databaser och sidan behöver endast visa upp informationen. Detta underlättade också inmatning av data till html eftersom PHP funktioner kan sättas rakt in i HTML källkod och man behöver inte bifoga informationen från JavaScript till HTML. Man hade heller inte ett behov att ladda sidan utan återladdning som det först var planerat. Tvärtom, återladdningen underlättade sidbytet.

Databasen formades av två huvudsakliga grupper av rörelser – sportspecifika och allmänna rörelser. Dessa två tabeller innehåller informationen för rörelserna som applikationen använder. Den mest väsentliga informationen av strukturen kan ses i figur 9.

För att göra koden tydlig och klar valde jag att använda Idiorm, ett PHP bibliotek för sökfunktioner mellan databas och PHP. Till skillnad från andra ORM system (Object/Relational Mapping), behöver man med Idiorm inte skapa modeller som man mappar till PHP tabeller. (Idiorm documentation, 2013)

För en fungerande Idiorm krävs en idiorm.php fil samt en konfigurationsfil (Idiorm GitHub, 2013). I konfigurationsfilen definierar man den information som behövs för databasen såsom databasnamn och användarnamn. Man definierar också valbara inställningar såsom i vilken form man vill ha resultatet.

| Field | Type | Default | Extra |
|----------------------|-------------|-------------|----------------|
| <u>id</u> | int(11) | <i>None</i> | auto_increment |
| name | varchar(45) | <i>None</i> | |
| video_code | varchar(45) | <i>None</i> | |
| sport_type | varchar(45) | <i>NULL</i> | |
| movement_type | varchar(45) | <i>None</i> | |
| persons | varchar(45) | <i>None</i> | |
| level | int(2) | <i>None</i> | |
| bodypart | varchar(45) | <i>None</i> | |

Figur 9. Tabellen som rörelserna sparas i.

4.2.4 Iteration 4

Den sista iterationen i det här arbetet handlade om att städa upp det som gjorts i de tidigare iterationerna för att få en presenterbar produkt. Det gällde också att bygga upp sökmetoderna och byta ut testdata till verklig data.

Designen för resultatsidan snyggades upp med CSS genom att bl.a. runda hörnen på rutor. Sidan med frågor gjordes mera användarvänlig för mobila apparater med hjälp av HTML-taggen labels. Små färgjusteringar gjordes också så att R5s logo passade bättre in på sidan. Resultatsidan använder mest PHP. HTML används främst till småsaker som att sätta in viewport samt andra funktioner som hittas i head taggen i filter.php filen. Själva innehållet är gjort med PHP och CSS.

Sidan gör en callback funktion till en PHP fil som i sin tur hämtar data från databasen med hjälp av Idiorm. Resultaten av callbackfunktionen blir en lista med objekt. Antalet objekt är lika många som antalet rörelser som skall visas. Listan från callbackfunktionen går igenom på resultatsidan och för varje objekt skrivs det ut en ruta med information om rörelsen.

Funktionerna som finns i en skild PHP fil använder alla Idiorm. Resultaten som man visar på resultatsidan varierar enligt de variabler man skickar med callbackfunktionen till Idiorm. Detta beror på att Idiorm placerar in variabeln i en förfrågan som liknar en

SQL sats. Beroende på informationen man vill ha från databasen fastställer man parametrar som:

- select – vilken kolumn från tabellen man vill ha.
- where – vad det skall likna.
- where_gte – var variabeln är lika stor eller större än.
- where_lte – var variabeln är lika liten eller mindre än.

Parametrarna som användes i detta program är främst olika kombinationer av select och where parametrar.

Under visas hur man skickar en förfrågan till databasen och returnerar resultatet till sidan:

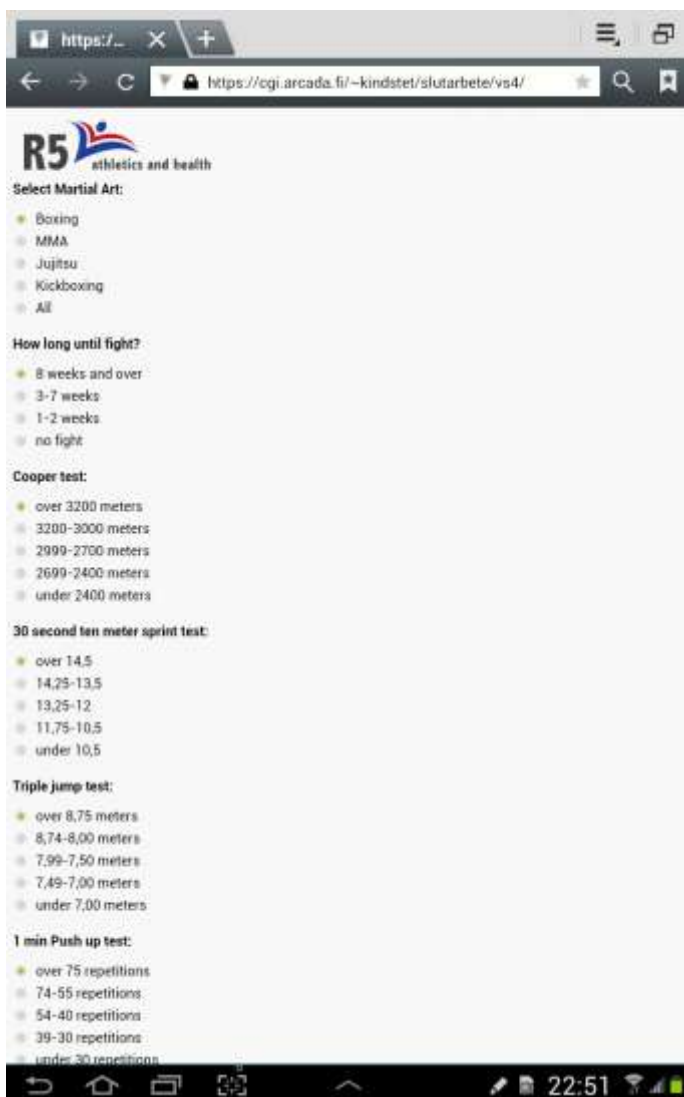
```
function get_moveName($moveName) {  
  
    $move = ORM::for_table('R5_movements')  
        ->select('name')  
        ->where('name', $moveName)  
        ->find_one();  
  
    echo($move->name);  
}
```

Figur 10. Idiorm skickar en förfrågan till databasen och returnerar resultatet till sidan.

5 PRESENTATION AV PROTOTYP

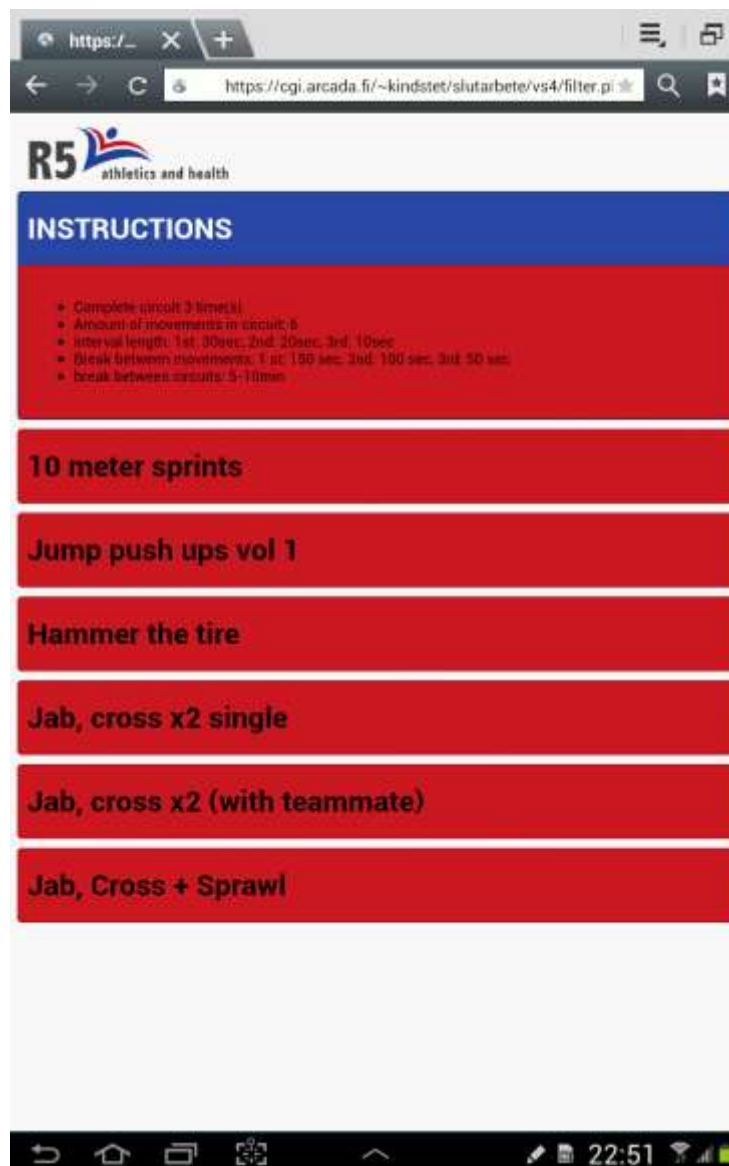
Tjänstens namn SECTMA som står för: Speed Endurance Circuit Training for Martial Arts. Den är i första hand designad för mobila plattformar, men fungerar också på andra plattformar.

Den första bilden visar hur filtersidan ser ut på en tablett. Här väljer man från de olika alternativen för att få ut optimalt resultat. Du fastställer din konditionsklass genom att svara på frågorna och ger dessutom den information om vilken kampsport du utövar samt hur länge det är före du har en match. De två senare är dock inte lika användbara för dem som endast använder tjänsten för att förbättra sin kondition. Huvudpoängen med tjänsten var alltid att förbättra träningen för dem som redan utövade en kampsport.



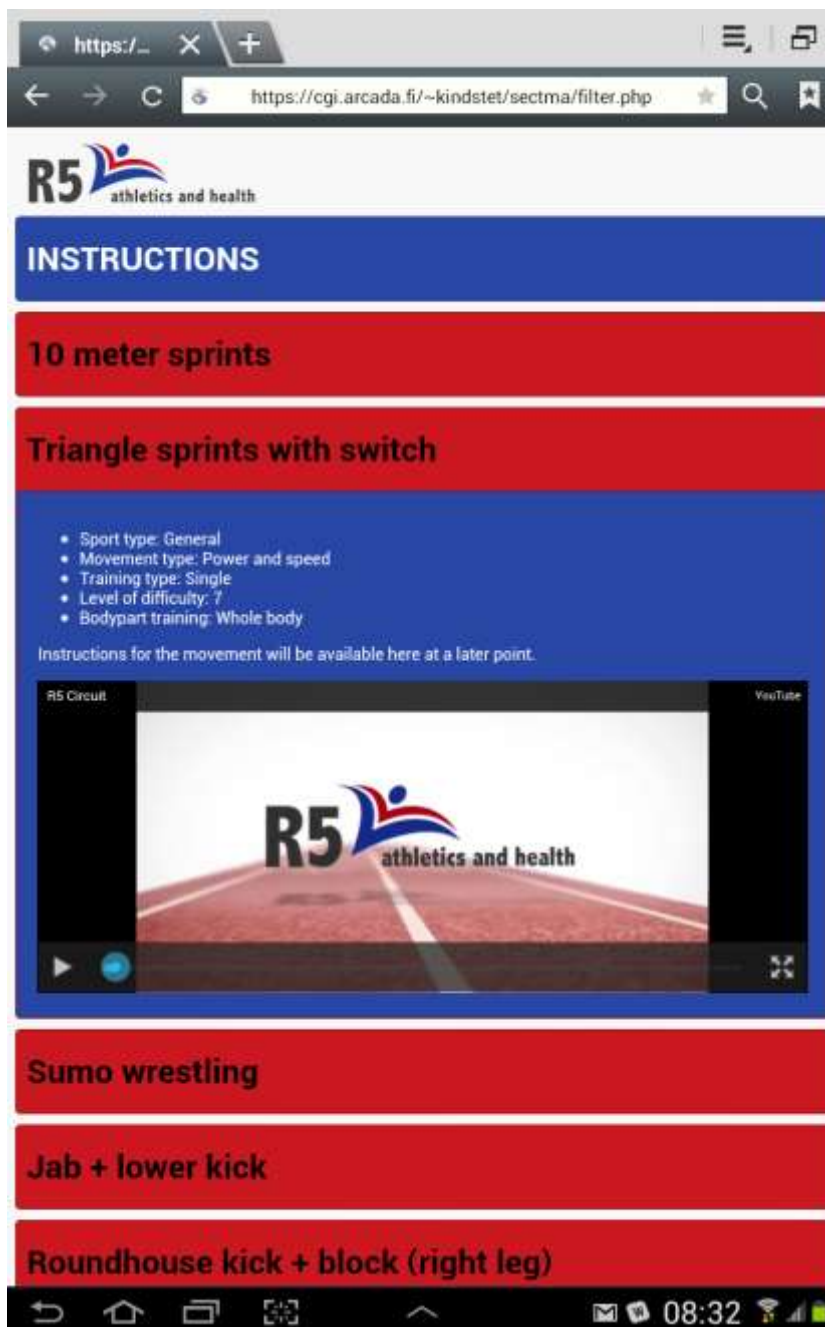
Figur 11. En bild på filtersidan från en Android tablett.

Här ser man hur resultatsidan ser ut efter att man valt sina filter och gått vidare till nästa sida. Överst finns ett instruktionsblock som öppnar sig automatiskt. Detta block ger instruktioner om hela träningscykeln, som hur länge du skall utöva varje rörelse, hur många gånger du skall gå igenom cyklerna osv. Om du använt programmet tidigare och redan vet dessa instruktioner kan du förminska instruktionerna. Varje rörelse kan också förstöras eller förminskas om man behöver information om dem. Denna bild är tagen från en Android tablett.



Figur 12. Hur resultatsidan ser ut när man öppnar den.

Den här bilden visar hur man har förminskat instruktionerna och endast ser på videoinstruktionerna för en av rörelserna. Bilden är tagen från en Android tablett.



Figur 13. Exempel på hur man kan använda sidan för att se på närmare instruktioner för en rörelse.

Nästa figur visar hur applikationen ser ut på en mobiltelefon. I ena bilden är alla block stängda och i andra bilden är ett block öppet för att visa hur rutan för en rörelse kan se ut.



Figur 14. Hur applikationen ser ut på en iOS mobiltelefon.

Den sista bilden visar hur hela sidan ser ut på en dator. Två rörelser och instruktionerna är öppna för att demonstrera hur en typisk användningssession skulle kunna se ut. Varje rörelse har sina egna instruktioner som berättar bl.a. hur tung rörelsen är och vilken typ av rörelse det är. Det finns också en video som demonstrerar hur rörelsen skall utföras ifall detta inte är klart från de andra instruktionerna.

INSTRUCTIONS

- Complete circuit 3 times
- Amount of movements at each 4
- Interval length: 1st 20sec, 2nd 25sec, 3rd 30sec
- Break between movements: 1st 150 sec, 2nd 100 sec, 3rd 80 sec
- Break between circuits: 5-10min

10 meter sprints

- Sport type: General
- Movement type: Power and speed
- Training type: Single
- Level of difficulty: 7
- Bodypart training: Whole body

Instructions for the movement will be available here at a later point.



Triangle sprints with switch

Sumo wrestling

- Sport type: General
- Movement type: Power and speed
- Training type: Single
- Level of difficulty: 7
- Bodypart training: Whole body

Instructions for the movement will be available here at a later point.



Jab + lower kick

Roundhouse kick + block (right leg)

Walking roundhouse kicks

Figur 15. Hela resultatsidan sätt från en dator.

6 SLUTORD

Efter att ha studerat STOF och Agile har jag kommit fram till att de fungerar bra ihop och båda är alternativ som kan fungera tillsammans. I teorin tror jag att en kombination av dessa två skulle passa bäst för ett projekt som inte är alltför stort, detta baserar jag på mina egna erfarenheter. STOFs principer ger en bra grund för planering av arbetet och Agile fungerar utmärkt som flexibla utvecklingsprinciper. Personligen hade jag lite svårt att uppfylla alla punkter av dessa två metoder eftersom jag bara är en utvecklare för ett rätt litet projekt.

STOF fungerade mycket bra för att skapa en grund och riktlinjer för arbetet. Man kunde alltid titta tillbaka på de punkter jag tog upp för att se vart man var på väg och vad man ville lägga till. I sin helhet tyckte jag att STOF fungerade mycket bra till denna typ av projekt, men tycker samtidigt att den skulle fungera bättre för ett lite större projekt där man faktiskt skulle behöva kommunikation mellan projektmedlemmar. STOF gav mig också en mycket klarare bild av hur man skall utveckla tjänster och vad allt man måste tänka på.

Agile har tidigare visats fungera bra och gjorde det även här. Sättet att utveckla i iterationer fungerar mycket bra, men en del andra saker fungerade inte lika bra eftersom projektet var så litet. Jag kan se det fungera mycket bra i både medelstora och stora projekt. Jag gillade också flexibiliteten som det ger när man kan skräddarsy processen så den fungerar väl för varje företag.

Jag anser att de fungerar väl tillsammans. Om projektet varit lite större skulle det fungerat bättre, men med lite improvisation fungerade det väl nu också. STOF gav en mycket bra grund och Agile såg till att man inte tappade den röda tråden.

KÄLLOR

Agilemanifesto. 2001. Manifesto for Agile Software Development.

Tillgänglig: <http://Agilemanifesto.org/>

Hämtad: 7.5.2013

Beck, K. 1999. Embracing change with extreme programming.

Tillgänglig: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=796139>

Hämtad: 11.6.2013

Bouwman. 2006. STOF model and method : Design and research.

Tillgänglig: <http://www.cse.tkk.fi/fi/opinnot/T-109.4300/2009/luennot-files/stof.pdf>

Hämtad: 5.5.2013

Bouwman, H. Vos, H. och T. Haaker. 2008. Mobile Service Innovation and Business Models. Springer-Verlag, Berlin Heidelberg.

de Reuver, M, Stein, S och Hampe, F. 2010. Towards a Service Platform and Business Model for Mobile Participation.

Tillgänglig:

http://www.ict.tbm.tudelft.nl/fileadmin/Faculteit/TBM/Over_de_Faculteit/Afdelingen/Afdling_Infrastructure_Systems_and_Services/Sectie_Informatie_en_Communicatie_Technologie/medewerkers/mark_de_reuver/Publications/doc/De_Reuver_et_al_2010_-_Towards_a_Service_Platform_and_Business_Model_for_Mobile_Participation_ICMB.pdf

Hämtad: 7.5.2013

Eriksson et. al. 2013. Development of a local outdoor mobile tour guide in HTML5 – a business model approach.

Förhandstryck.

Pine, J. och J. Gilmore. 1999. The Experience Economy. Harvard Business School Press, Boston, MA.

Planbox. 2013. Agile Project Management Meetings

Tillgänglig:<http://support.planbox.com/knowledgebase/articles/206602-agile-project-management-meetings>

Hämtad: 11.6.2013

IBM. 2008. Handling Requirements Effectively on Agile Projects.

Tillgänglig: <http://www.ibm.com/developerworks/rational/library/edge/08/oct08/rivera1>

Hämtad: 14.5.2013

IBM. 2013. Be lean. Be agile. Work together.

Tillgänglig: <http://www-01.ibm.com/software/rational/agile/>

Hämtad: 28.5.2013

Idiorm documentation. 2013. Philosophy.

Tillgänglig: <http://idiorm.readthedocs.org/en/latest/philosophy.html>

Hämtad: 20.5.2013

Idiorm GitHub. 2013. idiorm.php.

Tillgänglig: <https://github.com/j4mie/idiorm/blob/master/idiorm.php>

Hämtad: 20.5.2013

Wikipedia. 2013a. Agile software development.

Tillgänglig: http://en.wikipedia.org/wiki/Agile_software_development

Hämtad: 7.5.2013

Wikipedia. 2013b. Iterativ utvecklingsmodell.

Tillgänglig: http://en.wikipedia.org/wiki/File:Iterative_development_model_V2.jpg

Hämtad: 7.5.2013