

# Projektplan

3 september 2015

Version 1.0

---

**Gruppmedlemmar:**

Reshad Ahmadi

Maryam Bayat

**Handledare:**

Kenneth Nilsson

**Examinator:**

Björn Åstrand

# Innehåll

<b>1</b>	<b>Introduktion</b>	<b>3</b>
1.1	Projektmodell . . . . .	3
1.2	Utvecklad produkt . . . . .	3
1.3	Syfte och Mål . . . . .	4
1.3.1	Företagsmål . . . . .	4
1.3.2	Produktmål . . . . .	4
1.3.3	Projektmål . . . . .	4
1.4	Begränsningar . . . . .	4
<b>2</b>	<b>Metod</b>	<b>4</b>
<b>3</b>	<b>Projektorganisation</b>	<b>4</b>
3.1	Utvecklingsorganisation . . . . .	4
3.2	Intressenter . . . . .	4
<b>4</b>	<b>Krav på hårdvara och programvara</b>	<b>5</b>
<b>5</b>	<b>Nedbrytning av arbete</b>	<b>5</b>
5.1	Aktiviteter . . . . .	5
5.2	Leverabler . . . . .	6
<b>6</b>	<b>Uppgifter</b>	<b>6</b>
6.1	Milstolpar . . . . .	6
6.2	Tidplanering . . . . .	6
<b>7</b>	<b>Uppföljning, rapportering och kvalitetssäkring</b>	<b>6</b>
<b>8</b>	<b>Risikanalys</b>	<b>7</b>
<b>9</b>	<b>Referenser</b>	<b>8</b>

# 1 Introduktion

Att laga Ravioli hemma hade varit jobbigt och tidskrävande. Det tar väldigt lång tid att fylla på en ravioli deg(utkavlade degen) med fyllningar och resultat inte blir likadana. Det finns olika typer av Ravioli maskiner på marknaden redan nu. En typ av ravioli maskin som visas på figur1, underlättar processen, men det mesta görs manuellt och resultat inte blir likadana (mer förklaring).

Den andra typen av maskinen är väldigt stor med väldigt högt pris, som medför att de inte kan användas av hushåll, se figur 3.

Det finns olika typer av Ravioli maskiner på marknaden redan nu. Entyp ravioli maskin som visas på figuren ?? är den enklaste modellen som underlättar lite processen, men det mesta görs manuellt. ) Detta projekt ämnat till att skapa en Ravioli maskin. Projektet är en egen ide som skall utföras av två studenter vid Högskolan i halmstad.

Iden bakom projektet baseras på behov av en Ravioli maskin hemma. Det finns färdiga industriella maskiner i marknaden. Dessa maskiner är väldigt stora och tunga samt att de är dyra. Tanken är att man skapar en liten och billigt Ravioli maskin.

## 1.1 Projektmodell

Projektet kommer att följa den modell som presenteras i kap. 3, [1].

## 1.2 Utvecklad produkt

Detta projekt är ämnat till att skapa och utveckla mjukvaran till ett cykelgarage. Den slutliga produkten ska tillgodose behovet av välorganiserade cykelgarage med minimala stöldrisker.

Hela systemet kommer att bestå av en operatör, där systemdatorn med programvaran och en streckkodutskrivare finns, som är ansluten till själva garaget. Med hjälp av operatören skall kunder kunna registreras/avregistrera. Varje cykel ska kunna identifieras med en streckkod tillhandahållen av skrivaren. Garaget ska ha en ingång med streckkodläsare, panel för PIN-kod och datorstyrt lås samt en utgång med streckkodläsare. Registrerade ägare ska kunna ha ett begränsat antal cyklar i garaget.

Ett normalt användningsscenario kan se ut enligt följande:

1. En cykelägare med cykel kommer till ingången.
2. Ägaren låser upp dörren genom att läsa av streckkoden på cykeln.
3. Ägaren ställer cykeln i garaget, låser den och går sen ut genom utgången.

Samt:

1. En cykelägare utan cykel kommer till ingången.
2. Ägaren låser upp dörren genom att slå in sin PIN-kod.
3. Ägaren hämtar sin cykel.
4. Ägaren läser av streckkoden på cykeln vid utgången och går.

## 1.3 Syfte och Mål

### 1.3.1 Företagsmål

För att uppmuntra användandet av cyklar som transportmedel vill företaget underlätta cykelanvändningen genom att erbjuda säkra, organiserade cykelgarage som inte blir överfulla. Målet är att cyklister hellre ska välja företagets organiserade garage framför öppna, publika cykelställen med större trängsel och ökad risk för stöld och vandalism.

### 1.3.2 Produktmål

Målet med produkten är att tillhandahålla säkra, organiserade cykelgarage där kunden inte behöver oroa sig över att cykeln står på öppen eller allmän plats. Garaget ska kräva minimal övervakning och underhåll men kunna identifiera de människor som kommer och går vilka cyklar de har med sig. Systemet ska även ha mycket hög driftssäkerhet för att undvika att kunder inte kommer åt sina cyklar.

### 1.3.3 Projektmål

Målet med detta projekt är att utveckla en effektiv och lätt Ravioli maskin . Det skall vara mycket billigare jämfört med priserna på marknaden.

## 1.4 Begränsningar

Eftersom existerande verktyg används är den enda stora begränsningen den tid det tar att genomföra projektet. Tiden är låst till en deadline som inte kan flyttas, och personalresurser är begränsade. Följaktligen är kvaliteten den enda variabel som kan ändras om projektet löper risk att inte bli klar på utsatt tid.

## 2 Metod

## 3 Projektorganisation

### 3.1 Utvecklingsorganisation

Projektmedlemmarna är följande:

- Reshad Ahmadi (ansvarig för elktro samt mek delen)
- Maryam Bayat (ansvarig för data delen)

### 3.2 Intressenter

Det finns ett antal intressenter i projektet. Kursledningen agerar som den tänkta kunden som är en tågstation i södra Sverige, kap. 3 i [1].

- Kursledningen i kursen ETSA01 på LTH – tillhandahåller simulator av hårdvaran (kursansvarig Martin Höst)

- Projekthandledare – mottagare av leverabler såsom dokument (Björn Regnell)

Lärarna utför acceptansprovning på den levererade produkten, för att se till att produkten uppfyller:

- kraven i kravspecifikationen,
- kvaliteten på kravspecifikationen är tillräckligt bra,
- kvaliteten på testplanen är tillräckligt bra
- samt projektmodellen har följts.

## 4 Krav på hårdvara och programvara

Existerande verktyg och programvara kommer att användas. Öppna format och fri programvara ska användas i största möjliga mån före stängda och kommersiella system på grund av portabilitet och projektets låga budget.

Verktyg	Syfte	Anmärkning
Subversion (SVN)	Versionshantering	-
Eclipse	Utvecklingsmiljö för Java	SVN-plugin finns.
Latex	Dokumenthantering	God tillgänglighet på program.
Java	Kompilator och VM	-
Assembla	Projektverktyg online	Hosting för versionshantering. Använder delvis fri mjukvara. <a href="http://www.assembla.com">www.assembla.com</a> .

För testning finns simulator för hårdvara implementerad i Java.

## 5 Nedbrytning av arbete

### 5.1 Aktiviteter

**Projektplanering** En planering utav arbetet som skall definiera uppgifter samt milstolpar i projektet.

**Kravspecificering** Krav diskuteras, eliciteras och omformuleras.

**Högnivå-design** Design av systemet för en översiktlig struktur, som resulterar i lågnivå-design senare.

**Testplanering** En testplan beskriver hur programmet kommer fungera i olika testscenarion samt förhållanden.

**Implementering** Utgående från designen sker implementeringen i programkod. Olika delar av systemet implementeras parallellt.

**Kvalitetsförsäkring** När programmet är färdigskrivet måste programmet testas för att försäkra kunden att programmet uppfyller kraven ställda på kravspecifikationen.

**Rapportering** Rapporteringen sker internt inom gruppen för att försäkra gruppmedlemmarna om respektive person är färdiga med sina uppgifter eller ej.

## 5.2 Leverabler

**Projektplan** Övergripande planen för hur projektet ska genomföras. Detta dokument. Dokumentansvarig: Johan Lundström.

**Kravspecifikation** Kundens lista på specifika krav, vilket produkten begärs kunna lösa. Dokumentansvarig: Mazdak Farzone.

**Design** Programdesign utifrån kravspecifikationen.

**Programkod** Systemimplementation i Java som formar mjukvaran.

**Testplan** Dokumentet som beskriver interna samt externa tester som utförs på programmet. Dokumentansvarig: Björn Lennernäs.

**Testprotokoll** Ifyllda protokoll över de tester man utfört på programvaran.

## 6 Uppgifter

Deluppgift	Beskrivning	Tid	Krav
T1	Projektplan	8 dagar , 31/3-7/4	
T2	Kravspecifikationen	14 dagar , 7/4-21/4	T1
T3	Design	7 dagar, 21/4-28/4	T1
T4	Testplan	7 dagar, 21/4-28/4	T2
T5	Kodning	14 dagar, 28/4-12/5	T2
T6	Testning	14 dagar, 4/5-18/5	T1, T5

### 6.1 Milstolpar

- Specifikationer ( $M1$ )
- Design ( $M2$ )
- Kodning ( $M3$ )
- Testning ( $M4$ )

### 6.2 Tidplanering

## 7 Uppföljning, rapportering och kvalitetssäkring

Gruppen ser till att mötas regelbundet. På mötena granskas arbete utfört av gruppmedlemmar, speciellt inför varje extern granskning av gruppens handledare så granskar gruppen de dokument eller program som skall lämnas in till handledaren och avgör om de behöver kompletteras eller ej. Handledaren godkänner kvaliteten på arbetet eller ber om komplettering. På mötena utses även

ansvariga för olika delar av projektet, samt vilka som skall arbeta på vilket dokument eller program. Ansvaret ligger sedan hos den ansvariga för dokumentet/programmet att se till att de arbetare personen har tillgång till utför ett effektivt arbete, och att arbetet är slutfört inför nästa deadline.

Gruppen använder sig av projektverktyget Assembla som är speciellt utvecklat för programvaruutveckling. Webb sidan ger bland annat tillgång till ett antal viktiga verktyg för konfigurationshantering och intern rapportering. Gruppens dokument och program är kan laddas upp på sidan, där samtliga medlemmar i gruppens tillåts ändra dokumenten och programmen. Ändringarna kommenteras av den som utfört ändringen och både det som ändrats på och den nya versionen av dokument eller programmet sparas. På så sätt upprätthålls en meningsfull dialog för ändringar i dokumenten/programmen samt att alla versioner av dokumentet kan återskapas.

På webbsidan upprätthåller gruppen även uppgifter om vem som skall göra vad och när det skall vara klart, samt andra viktiga tidpunkter och uppgifter som var och när nästa möte skall hållas och hur medlemmarna skall förbereda sig inför detta möte. Webb sidan möjliggör även en kommunikation mellan olika medlemmar i gruppen, där de olika medlemmarna kan rapportera till exempelvis en ansvarig för ett visst dokument/program vad som gjorts, av vem och vad som återstår att göra.

## 8 Riskanalys

**Projektdeltagare tillgänglighet:** En eller flera nyckelpersoner i gruppen blir sjuka eller har hög frånvaro.

*Sannolikhet:* Medel

*Effekt:* Hög

*Strategi:* Genom kontinuerlig kommunikation och kunskapsspridning inom gruppen kan detta undvikas.

*Riskindikator:* Risken kan upptäckas vid bortfall av gruppmedlemmar på möten och att deluppgifter inom projektet inte blir genomförda i tid.

**Kravförändringar:** Förändring eller missförstånd av krav i kravspecifikationen för projektet kan innebära problem längre fram vid tester av mjukvaran.

*Sannolikhet:* Låg

*Effekt:* Hög

*Strategi:* Kravspecifikationen bör utarbetas grundligt för att minimera att risken skall ske.

*Riskindikator:* Vid systemtest och övriga tester kan risken upptäckas.

**Organisation:** Beslutsfattandet från projektledarens sida är bristfällig, vilket kan medföra osäkerhet i arbetet för alla projektdeltagare.

*Sannolikhet:* Låg

*Effekt:* Medel

*Strategi:* Genom kontinuerlig kommunikation inom gruppen och rapportering till projektledaren kan effekterna minimeras.

*Riskindikator:* Risken kan upptäckas vid dålig organisation av möten och om deluppgifter inom projektet börjar bli osammanhängande på grund av bristande kommunikation mellan projektdeltagare.

**Tidsuppskattning:** Dålig uppskattning av tidsupptagandet för delmoment inom projektet kan försena hela projektet.

*Sannolikhet:* Medel

*Effekt:* Hög

*Strategi:* Fortlöpande kommunikation och rapportering av delmoment inom projektet mellan projektledaren och projektdeltagare skall minimera eventuella tidsförseningar.

*Riskindikator:* Risken kan upptäckas vid försening av deluppgifter.

**Problem med mjukvara:** Vid tester av mjukvara kan rapportering av problem uppkomma.

*Sannolikhet:* Medel

*Effekt:* Medel

*Strategi:* Ständiga tester av mjukvara skall genomföras för problemfri försäkringen.

*Riskindikator:* Vid diverse tester av mjukvara kan risken påträffas.

**Tekniska kommunikationsproblem:** Att vår kommunikationscentral över internet kraschar.

*Sannolikhet:* Låg

*Effekt:* Hög

*Strategi:* Om detta händer så kommer kommunikationen ändå fortsätta över internet fast utanför vår kommunikationscentral.

*Riskindikator:* Risken kan upptäckas vid störningar med kommunikationscentralen, att hemsidan inte fungerar helt korrekt.

**Problem med mjukvara:** Medlemmar saknar eller får problem med mjukvaran/projektverktyget.

*Sannolikhet:* Medel

*Effekt:* Låg

*Strategi:* En övergripande utbildning av gruppmedlemmar minskar sannolikheten att missförstånd och förseningar uppstår.

*Riskindikator:* Risken upptäcks exempelvis när medlemmarna engagerar sig för lite eller när en ny version av en mjukvara släpps.

## 9 Referenser

- [1] Software Engineering Research Group, Department of Computer Science, Lund University. *Project and Exercises in the Software Engineering Process*





Figur 1: Exempel på ett meddelande

