

# Kvalitetsplan

Redaktör: Ruben Das

**Version 0.1**

## Status

Granskad	Andreas Runfalk	-
Godkänd	Andreas Runfalk	-

## PROJEKTIDENTITET

VT, 2015, Grupp 2  
Linköpings Tekniska Högskola, IDA

## Gruppdeltagare

Namn	Ansvar	Telefon	E-post
Adam Sestorp	Team leader	070 9987270	adase035@student.liu.se
Dennis Ljung	Dokumentansvarig	070 8568148	denlj069@student.liu.se
Alexander Yngve	Utvecklingsansvarig	076 2749762	aleyn573@student.liu.se
Martin Söderén	Analysansvarig	070 8163241	marso329@student.liu.se
Ruben Das	Kvalitetssamordnare	073 7355892	rubda680@student.liu.se
Sebastian Fast	Arkitekt	073 3885208	sebfa861@student.liu.se
Johan Isaksson	Testledare	070 2688785	johis024@student.liu.se

**Hemsida:** <http://pum-2.ida.liu.se/>

**Kund:** SAAB

**Kontaktperson hos kund:** Daniel Simon

**Kursansvarig:** Kristian Sandahl

**Handledare:** Andreas Runefalk

## Innehåll

<b>1 Syfte</b>	<b>1</b>
<b>2 Referenser</b>	<b>1</b>
2.1 Interna referenser . . . . .	1
2.2 Externa referenser . . . . .	1
<b>3 Hantering</b>	<b>1</b>
3.1 Organisation . . . . .	1
3.2 Uppgifter . . . . .	1
3.3 Roller och ansvar . . . . .	1
<b>4 Dokumentation</b>	<b>2</b>
4.1 Syfte . . . . .	2
4.2 Dokumentlayout och konsistens . . . . .	2
4.3 Minimikrav för dokumentation . . . . .	2
4.3.1 Kravspecifikation . . . . .	2
4.3.2 Projektplan . . . . .	2
4.3.3 Kvalitetsplan . . . . .	2
4.3.4 Utvärdering av förstudien . . . . .	2
4.3.5 Arkitekturdokument . . . . .	2
4.3.6 Testplan . . . . .	3
4.3.7 Teknisk dokumentation . . . . .	3
4.3.8 Användarhandledning . . . . .	3
4.4 Övrig dokumentation . . . . .	3
4.4.1 Mötesrapporter . . . . .	3
4.4.2 Tidsrapporter . . . . .	3
4.4.3 Statusrapporter . . . . .	3
<b>5 Standarder</b>	<b>3</b>
5.1 Syfte . . . . .	3
5.2 Innehåll . . . . .	3
5.2.1 Kod . . . . .	3
5.2.2 Testning . . . . .	4
<b>6 Granskningar</b>	<b>4</b>
6.1 Syfte . . . . .	4
6.2 Minimikrav . . . . .	4
6.2.1 Kravgranskning . . . . .	4
6.2.2 Arkitekturgranskning . . . . .	4
6.2.3 Slutgranskning . . . . .	4
6.2.4 Dokumentgranskning . . . . .	4
<b>7 Test</b>	<b>4</b>
<b>8 Dokumenthantering</b>	<b>4</b>
<b>9 Riskhantering</b>	<b>4</b>
<b>Bilaga A Kodstandard</b>	<b>5</b>

## Dokumenthistorik

Version	Datum	Utförda förändringar	Utförda av	Granskad
0.1	2015-02-09	Första utkast	Ruben Das	

## 1 Syfte

Detta dokument specificerar kvalitetsplanen som grupp 2 i kursen programvaruutvecklingsmetodik (TDDD77) kommer att följa. Den ger riktlinjer samt procedurer som bör följas av projektgruppen för att kunna leverera en produkt av tillräckligt hög kvalitet till kunden. Produkten består av en optimeringsalgoritm som kan lösa kvadratiska problem utvecklad i språket C. Detta dokument följer paragrafer som är specificerade i IEEE-STD-730-2002.

## 2 Referenser

### 2.1 Interna referenser

[Sestorp, 2015] Sestorp, Adam (2005), Projektplan, Institutionen för datavetenskap (IDA) vid Linköpings universitet, Linköping.

[Ljung, 2015] Ljung, Dennis (2005), Kravspecifikation, Institutionen för datavetenskap (IDA) vid Linköpings universitet, Linköping.

[Fast, 2015] Fast, Sebastian (2005), Arkitektur, Institutionen för datavetenskap (IDA) vid Linköpings universitet, Linköping.

[Isaksson, 2015] Isaksson, Johan (2015), Testplan, Institutionen för datavetenskap (IDA) vid Linköpings universitet, Linköping.

### 2.2 Externa referenser

IEEE-STD-730-2002 SQAP group 1 SE 2003-2004

Svenska skrivregler (2000), Svenska språknämnden, 2 uppl, Liber AB, Stockholm. ISBN47-04974-X

Projektmodellen Lips (2011), Tomas Svensson, Christian Krysanter och Studentlitteratur. Upp-  
laga 1:1, Studentlitteratur AB, Stockholm. ISBN-978-91-44-07525-9

## 3 Hantering

### 3.1 Organisation

Alla projektmedlemmar har sin egen roll, vilket medför att de har ett eget ansvarsområde att ta hand om. Projektgruppens kvalitetssamordnare ska ansvara för att projektmedlemmarna ska följa riktlinjerna som tas upp i detta dokument.

### 3.2 Uppgifter

Kvalitetssamordnaren ansvar för att:

- Se till att alla dokument är grammatiskt korrekta.
- Se till att all kod skrivs efter standarden som nämns i dokumentet.
- Se till att all kod ska vara väldokumenterad.

### 3.3 Roller och ansvar

Kvalitetssamordnaren ska ansvara för att ha koll på vilka komponenter som behöver kvalitetsgranskas. Tillsammans med projektgruppen kommer arbetet med att granska komponenter fördelas jämnt, då arbetsbördan ska vara likvärdig för alla.

## 4 Dokumentation

### 4.1 Syfte

Denna del kommer att beskriva de nödvändiga dokumenten för projektet. Syftet med dokumentationen är att möjliggöra förståelse för projektet.

### 4.2 Dokumentlayout och konsistens

Dokumentet skall vara skriva på svenska. De ska även vara skrivna i latex, men slutprodukten ska vara i pdf-format. Varje dokument ska följa en specifik mall, vare sig LIPS-mallar eller standard IEEE-mallar.

### 4.3 Minimikrav för dokumentation

- Kravspecifikation
- Projektplan
- Kvalitetsplan
- Utvärdering av förstudien
- Arkitekturdokument
- Testplan
- Teknisk dokumentation
- Användarhandledning

#### 4.3.1 Kravspecifikation

Kravspecifikationen består främst av de krav som ställs på produkten som ska tas fram. Även produktens syfte samt mål. Dokumentet heter "Kravspecifikation", [Ljung, 2015].

#### 4.3.2 Projektplan

Projektplanen består främst av aktiviteter, milstolpar och organisationsplan. Dokumentet heter "Projektplan", [Sestorp, 2015].

#### 4.3.3 Kvalitetsplan

Kvalitetsplan är detta dokument och ska ge övergripande riktlinjer för de kvalitetskriterier som skall följas av projektmedlemmarna. [Das, 2015].

#### 4.3.4 Utvärdering av förstudien

Detta dokument innehåller en utvärdering av förstudien. Detta dokument är ej påbörjad, då förstudien inte är klar.

#### 4.3.5 Arkitekturdokument

Detta dokument ska innehålla en övergripande design över hur produkten ska se ut i form av figurer och pseudokod. Arkitekturdokumentet uppdateras löpande just nu. Dokumentet heter "Arkitektur", [Fast, 2015].

#### 4.3.6 Testplan

Testplanen innehåller testning som ska genomföras under projektets iterationer. Dokumentet heter "Testplan", [Isaksson, 2015].

#### 4.3.7 Teknisk dokumentation

Detta dokument kommer innehålla en övergripande dokumentation över hur systemet är uppbyggt och hur det funkar. Detta dokument är ej påbörjat.

#### 4.3.8 Användarhandledning

Användarhandledningens innehåll kommer bestå av hur produkten används. Detta dokument är ej påbörjat.

### 4.4 Övrig dokumentation

#### 4.4.1 Mötesrapporter

Mötesrapporter ska innehålla datumet när mötet hölls, punkter som diskuterades och även vad som kom fram av diskussioner i små drag.

#### 4.4.2 Tidsrapporter

Tidsrapporter ska fyllas i senast varje söndag tills projektavslut. Varje projektmedlem ska även skriva in antalet timmar de har jobbat med en viss uppgift. Denna rapport ska skickas till handledaren.

#### 4.4.3 Statusrapporter

Statusrapporter ska innehålla projektgruppens status om hur arbetet går, vilka problem som har uppstått och vad man har lyckats med. Denna rapport ska skickas till handledaren och även till kunden om kunden begär det.

## 5 Standarder

### 5.1 Syfte

Denna del av dokumentet är till för att klargöra vilka standarder som bör följas för att uppnå en god kvalitet av kod.

### 5.2 Innehåll

- Skrivning av kod
- Testning av kod

#### 5.2.1 Kod

Kod skall vara skriven i språket C99 och följa standarden som tas upp i bilaga A. Även kommentering av kod ska följa standarden som tas upp i bilagan.

Koden ska kompileras med strängen som syns nedan. Filerna kommer kompileras med gcc 4.8. Flaggor som har valts är för att få en väldigt sträng "rättning" av koden som skrivs.

- gcc -Wall -pedantic -std=c99 -o namn namn.c

### 5.2.2 Testning

Testning ska ske efter riktlinjerna som finns i "Testplan", [Isaksson, 2015].

## 6 Granskningar

### 6.1 Syfte

Denna del av dokumentet avser de granskningar som ska göras på de komponenter som tillslut kommer ge oss en slutprodukt.

### 6.2 Minimikrav

#### 6.2.1 Kravgranskning

Kravgranskning skall göras för att säkerhetsställa att kraven i "Kravspecifikation", [Ljung, 2015] är tillräckliga samt realiserbara.

#### 6.2.2 Artitekturgranskning

Artitekturgranskning skall göras för att undersöka om den uppfyller det som har sagts i "Kravspecifikation", [Ljung, 2015]. Även att artituren håller hög kvalité samt är realiserbar inom tidsramen.

#### 6.2.3 Slutgranskning

Slutgranskning innefattar att slutprodukten granskas så att den uppfyller det som har specificerats i kravspecifikationen samt artitekturdesignen.

#### 6.2.4 Dokumentgranskning

Alla dokument som produceras av projektgruppen ska granskas för att undersöka om de håller riktlinjerna som har nämnts i detta dokument. D.v.s undersöka om de håller en god nivå.

## 7 Test

Alla projektmedlemmar ska ansvara för testning av komponenter. Detaljer om hur dessa tester ska gå tillväga finns i "Testplan", [Isaksson, 2015].

## 8 Dokumenthantering

Alla dokument som skickas till diverse personer, såsom handledare och kund sparas på vår Git. Medan dokument såsom tids-, mötes- och statusrapporter sparas i Google Drive.

## 9 Riskhantering

Riskhantering kommer göras gemensamt av projektmedlemmarna för att bedöma om vilka åtgärder som bör tas.



## Bilaga A Kodstandard

- Indentering sker med två blanksteg.
- Funktionsnamn och variabelnamn skrivs med små bokstäver med understreck som separator mellan ord.
- Pekare skrivs med asterisken direkt efter datatypen.
- Filer inkluderas där de behövs, antingen i c- eller h-filen.
- Den öppnande måsvingeparentesen skrivs på samma rad som funktionsnamnet och returtypen. Den stängande parentesen skrivs ensam på raden efter den avslutande satsen i funktionen. If-satser och liknande skrivs på samma sätt.
- Typedef sker separat från struct-deklarationer.
- Kommentarer skrivs som i exemplet nedan.
- Koden skall vara skriven på engelska.

```
/*  
  Author: Alexander Yngve  
  Date: 2015-02-11  
  Description: Short description of why this file is needed.  
*/  
  
#include <stdio.h>  
  
/* Short explanation of why this type is needed. */  
struct struct_t {  
    int* pointer;  
    int number;  
};  
  
typedef struct struct_t struct_t;  
  
/* Short explanation of why this function is needed. */  
int main() {  
    printf("Hej\n");  
  
    struct struct_t data_1;  
    data_1.number = 4;  
    printf("%i\n", data_1.number);  
  
    struct_t data_2 = {NULL, 5};  
    printf("%i\n", data_2.number);  
  
    return 0;  
}
```