

1DV404 – HT2014
Iterativ mjukvaruutveckling
Laboration 3 - Testning

av Da-Sing Trinh
@dt222cc

Planering

Uppgift	Beskrivning	Planerad tid HH:MM
1	Planering	00:30
2	Testplan	03:00
3	Design och implementation	06:00
4	Enhetstestning	04:00
5	Implementera testsviten och kör	04:00
6	Integrationstestning	03:00
7	Reflektion	01:00
	Sammanställa dokument	00:30

Summering

22:00

Tidslog

Datum	Uppgift	Beskrivning	Förbrukad tid HH:MM
11.12.2014	1	Planering	00:20
12.12.2014	2	Mycket förstudier, förvirrad över hur mycket jag ska ha i dokumentet. Saknar kanske material. Kommer att komplementtera dokumentet mera längre fram.	03:30
13.12.2014	3	För mycket tid lade ner här. Det var väldigt svårt och detta hade jag planerat in. Fick göra om en hel del, göra ändringar på användningsfallen. Det var svårt att implementera ett helt funktionellt kod. Jag har förstått tanken med detta men som jag nämde var det mycket svårt att arbeta med detta. Det känns som jag har missat en del.	07:00
13.12.2014	3	Jag gjorde om uppgiften. Jag bytte användningsfall till ett "enklare". Kodade i JavaScript (det strulade med visualstudio).	03:30
14.12.2014	3&4&7	Gjorde förändringar i klasserna. Kollade på QUnit. Försökt arbeta med testsvits/testfall/testfixtur, vet inte om jag har gjort rätt. Inte helt klar med uppgift 4. Har kanske arbetat med uppgift 5-6, jag vet inte. Gjort några korta reflektioner.	07:00
15.12.2014	4	Gjort förändringar i klasserna igen, lite mer testbar tror jag att det blev. Skrivit dokumentet.	06:00
15.12.2014	5	Jobbat med koden en bit. Försökt göra testerna.	05:25
17.12.2014	5	Jobbat med dokumentet, fylla i data. Paketera upp det lite fint. Kanske gjort någon justering i koden.	05:40
17.12.2014	6	Försökt arbeta med integrationstestning men vet inte precis hur jag ska göra det, inte så mycket tid för att pyssla med det.	02:00
17.12.2014	7	Funderingar och skrivit reflektionerna	01:35
Summering			31:30

Version	Datum	Testplan	Författare
1.1	2014-11-12		dt222cc

1. Introduktion

Dokumentet ska specificera vilka tester som ska utföras och hur de ska genomföras. Dokumentet kommer att fungera som en referens för att visa att de ställda kraven har uppfyllts.

Bakgrund

Testerna ska se till att kunden får en produkt som har den kvalitet och prestanda som förväntades att produkten ska ha. Testerna som ska genomföras är till för att säkerställa att de krav som har ställts på systemet ska hålla.

2. Översikt

Dokumentet kommer att definiera är:

- Omfattning, fokusområden och mål.
- Vilka tester som ska utföras och hur de ska genomföras.
- Ansvarsområden.
- Den teststrategi och typer av tester för den här versionen.
- Eventuella risker, frågor, antaganden och test beroenden

3. Testmål

Testplanen har som mål att definiera de alla områden på ett tydligt sätt. Alla krav ska verifieras och riskerna minimerade.

Det som testerna kommer att innefattas av är:

- Funktionella krav
- Prestanda krav
- Säkerhets krav
- Use-Case krav

4. Teststrategi

De aktuella Use-Casen samt de krav som visionen och den kompletterande/supplementerande specifikationen tar upp kommer att användas för att driva testerna.

Systemet ska testas genom att utsätta den för olika fall som den kan stöta på och måste kunna hanteras med. Testplanen är strukturerad så att alla test som ska utföras på en enhet ligger tillsammans.

Detta ska förenkla jobbet med att se vilka tester som ska utföras på varje enhet. Meningen med strukturen är att den lätt ska kunna matchas mot kravspecifikationen.

4.1 Testens presentation

I detta dokument kommer alla tester att beskrivas med en tabellrad enligt nedan.

Test nr x	Testansvarig	Beskrivning av test	Deadline
-----------	--------------	---------------------	----------

4.2 Misslyckade tester

Vid ett misslyckat test ska resultaten analyseras för att ta reda på vad som gick fel vid testet, åtgärder ska vidtas och sen ska ett nytt test utföras. Felrapportering ska också utföras.

Visar det sig att ett test för att verifiera ett krav är omöjligt att utföra, ska ett gemensamt beslut mellan projektgrupp, beställare och kund tas, så att det kan ändras eller till och med strykas.

5. Testomgivning

[...]

6. Modultester

[...]

7. Integrationstester

[...]

8. Systemtester

[...]

9. Resurser

Roll	Kontakt uppgifter	Ansvar/uppgifter
System testare	[namn] [e-mail adress] [telefon/mobilnummer]	Utför tester, rapporterar/dokumenterar resultat.

10. Planering/schema

[...]

11. Referenser

<http://epf.eclipse.org/wikis/openup/>

<http://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/>

<http://www.computing.dcu.ie/~davids/courses/CA267/ieee829mtp.pdf>

<http://www.isy.liu.se/edu/projekt/tsrt71/2007/ikp/Testplan%20version%201.0.pdf>

<http://www.adelaide.edu.au>

Version	Datum	Klassbeskrivningar	Författare
1.1	2014-12-12	För UC7– Registrera gren (avregistrera gren)	dt222cc

Flöden – Registrera gren

UC

Initiering: Användaren väljer att registrera en ny gren.

Pre: Deltävlingen registrerade. Användaren (administratör) är inloggad i systemet.

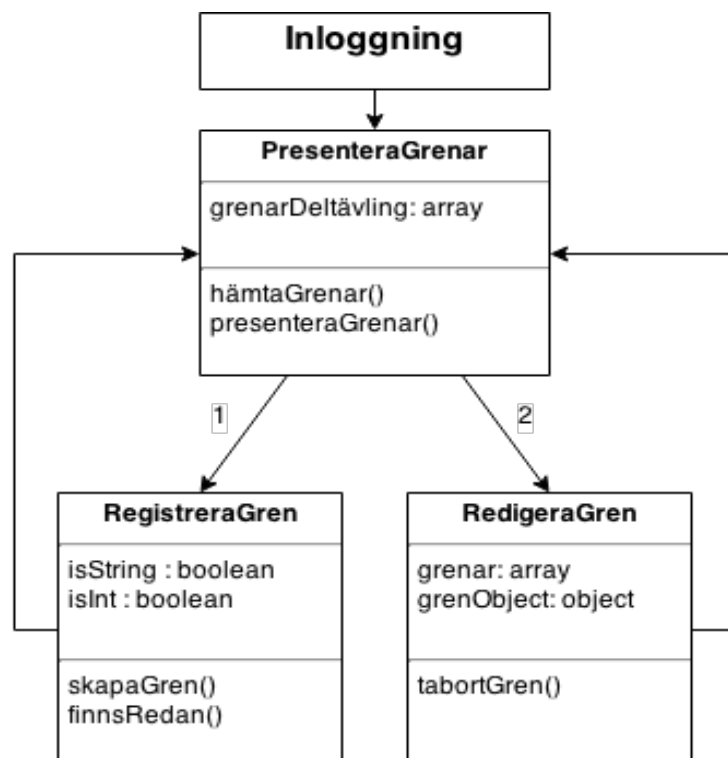
Primärt flöde

1. Aktören väljer i vilken deltävling som det ska läggas till en ny gren.
2. Systemet presenterar ett formulär med namn för grenen, och antal deltagare som det ska vara för grenen.
3. Den nya grenen läggs till i en lista med alla grenar för deltävlingen.

Alternativ flöde

1. Grenen finns redan
2. Användaren måste mata in ett heltal för antal deltagare.
3. Användaren väljer att avbryta

Designelement



Version 1.1	Datum 2014-12-15	Enhetstestning	Författar dt222cc
-----------------------	----------------------------	-----------------------	-----------------------------

Jag har valt att testa klasserna RegistreraGren och RedigeraGren, med mer focus på den första.

Klass: RegistreraGren

Denna klass tar emot en array (de registrerade grenar), namnet på nya grenen och dess storlek (antal deltagare). Om grenen redan finns så läggs grenen "inte" till utan visar ett meddelande att grenen redan finns. Om grenen inte finns så läggs grenen i arrayen som innehåller alla grenar.

Testsvits, testfixturer och testfall

Jag valde att ha med dessa tester för de testar om en ny gren registreras eller inte, om grenen redan fanns i systemet eller om man skickar fält i fel format.

Test	Testfall	Testdata: Grenar	Indata: Namn	Indata: Storlek	Förväntad resultat
1	Lägga till gren	Bom, 10 Hopp, 12	Barr	6	Bom, 10 Hopp, 12 Barr, 6
2	Lägga till gren som redan finns	Bom, 10 Hopp, 12	Bom	5	Bom, 10 Hopp, 12
3	Lägga till gren med fel format	Bom, 10 Hopp, 12	5	Simmning	Bom, 10 Hopp, 12
4	Lägga till gren med tomma fält	Bom, 10 Hopp, 12	""	""	Bom, 10 Hopp, 12

Klass: RedigeraGren

Användaren klickar på en knapp som tillhör grenen. En popup visas med fält som man skriver in det nya namnet och storlek.

Testsvits, testfixturer och testfall

Jag valde att ha med dessa tester för de testar om grenen förändrades eller inte, omman skickar in fält i fel format.

Test	Testfall	Testdata: Grenar	Testdata: Gren	Indata: Namn	Indata: Storlek	Förväntad resultat
1	Redigera första grenen	Bom, 10 Hopp, 12	Bom	Fristående	8	Fristående, 10 Hopp, 12
2	Redigera andra grenen	Bom, 10 Hopp, 12	Hopp	Ringar	8	Bom, 10 Ringar, 8
3	Fel format	Bom, 10 Hopp, 12	Bom	7	Hopp	Bom, 10 Hopp, 12
4	Tomma fält	Bom, 10 Hopp, 12	Bom	""	""	Bom, 10 Hopp, 12

Version 1.1	Datum 2014-12-17	Implementatera testsviten och kör	Författare dt222cc
----------------	---------------------	-----------------------------------	-----------------------

Jag använder tabellen från tidigare men modifierad med "verklig resultat". Jag hade en hel del problem med att implementera testkoden. Det blir inte som jag hade förväntat mig. Jag testade köra med Qunit för JavaScript men jag är inte så insatt/bra med det. Behöver öva mer. En del av testerna är inte implementerad, jag hade inte tillräcklig tid med det. Det var ett svårt och jobbigt moment detta.

Analys och förbättringar

Resultaten presenteras nu som [object Object],[object Object],[object Object].

Det skulle vara bättre om de byts ut mot de fält som de har = name: "Bom", size: "10".

Jag ville lägga till fler testfaller som kontrollerar att det man matar in är i rätt format. Har inte implementerat det, så jag kan inte precis testa det. Mer testfaller som testar att klassen innehåller vissa objekt som t.ex att en object som sparar undan de värden man skickade med "finns".

Testdata: RegistreraGren

Test	Testansvarig	Testfall	Förväntad resultat	Verklig resultat	Godkänt/ Undergänt
1	dt222cc	Lägga till gren	Bom, 10 Hopp, 12 Barr, 6	Bom, 10 Hopp, 12 Barr, 6	Godkänt
2	dt222cc	Lägga till gren som redan finns	Bom, 10 Hopp, 12	Bom, 10 Hopp, 12	Godkänt
3	dt222cc	Lägga till gren med fel format	Bom, 10 Hopp, 12	**	**
4	dt222cc	Lägga till gren med tomma fält	Bom, 10 Hopp, 12	**	**

Testdata: RedigeraGren

Test	Testansvarig	Testfall	Förväntad resultat	Verklig resultat	Godkänt/ Undergänt
1	dt222cc	Redigera första grenen	Fristående, 10 Hopp, 12	Fristående, 10 Hopp, 12	Godkänt
2	dt222cc	Redigera andra grenen	Bom, 10 Ringar, 8	Bom, 10 Ringar, 8	Godkänt
3	dt222cc	Fel format	Bom, 10 Hopp, 12	**	**
4	dt222cc	Tomma fält	Bom, 10 Hopp, 12	**	**

Kod – Registrera gren

```
"use strict";

// Klassen RegistreraGren
function RegistreraGren(array, name, size) {
  var grenObject = {name: name, size: size};
  return finnsRedan(grenObject, array);
}

// RegistreraGren.prototype.finnsRedan = function(object, array) { // Utan QUnit
function finnsRedan(object, array) { // För QUnit
  var count = 0;

  for (var i in array) {
    if (object.name === array[i].name) {
      console.log("Grenen " + object.name + " finns redan!");
      break;
    }
    else if (object.name === undefined) {
      console.log("Grenen har inga värden. Gren registreras inte.");
      break;
    }
    else {
      count++;
    }
  }

  // Grenen finns inte, registrera gren till array
  if (count === array.length) {
    array.push(object);
  }

  renderaGrenar(array); // this.renderaGrenar(array);
  return array;
}

// RegistreraGren.prototype.renderaGrenar = function(array) {
function renderaGrenar(array) {
  // Tänkt att alla objekt i arrayen ska renderas på html dokumentet. (kanske som separat klass)
}
```

Mer kod – Redigera gren

```
"use strict";

// Klassen RedigeraGren
function RedigeraGren(array, name, newName, newSize) {

    for (var i in array) {
        if (array[i].name == name) {
            array[i].name = newName;
            array[i].size = newSize;
            break;
        }
    }
    uppdateraGrenar(name, newName, newSize);
    return array;
}

// RedigeraGren.prototype.uppdateraGrenar = function(id, name, size) { // Utan QUnit
function uppdateraGrenar(id, name, size) { // För QUnit

    var grenar = document.getElementById("grenar"), li = document.getElementById(id),
    dt = document.getElementById(id + "P"), dd = document.getElementById(id + "Antal"),
    img = document.createElement("img"), img2 = document.createElement("img"),
    a = document.createElement("a"), a2 = document.createElement("a"); // Behövde komprimera detta för
dokumentet

    li.id = name;
    dt.id = name + "P";
    dd.id = name + "Antal";
    dt.innerHTML = name;
    dd.innerHTML = "Antal deltagare: " + size;
    img.src = "pics/delete.png";
    img2.src = "pics/clock.png";
    a.setAttribute("href", "#");
    a2.setAttribute("href", "#");

    a.onclick = function() {
        grenar.removeChild(li);
        // Saknar en metod som ska ta bort objektet från arrayen.
        // Ska anropa en klass/function som renderar om grenarna.
    };

    a2.onclick = function() {
        // Steget före redigera gren.
        // alt1: är en popup där man skriver de nya värden.
        // alt2: ingen bild, utan input längre ner på sidan där man skriver vilken gren som ska ändras och det
        nya namnet/antal.
    };

    a.appendChild(img), a2.appendChild(img2);
    dt.appendChild(a), dt.appendChild(a2);
}
```

Lite testkod – alt1: i window.onload

```
"use strict";

// Testkod
window.onload = function(){

  // testdata
  var indata = [{name: "Bom", size: "10"}, {name: "Hopp", size: "12"}];
  var result1 = [{name: "Bom", size: "10"}, {name: "Hopp", size: "12"}, {name: "Barr", size: "6"}];

  // Testfall 1: Lägg till ny gren.
  var test1 = new RegistreraGren(indata, "Barr", "6");
  if (test1 === result1) {
    console.log("test 1, ok");
  } else {
    console.log("test 1, fail");
  }

  // Testfall 2: Gren finns redan
  var test2 = new RegistreraGren(indata, "Bom", "10");
  if (test2 === indata) {
    console.log("test 2, ok");
  } else {
    console.log("test 2, fail");
  }

  // Testfall 3: Redigera grenar
  var test3 = new RedigeraGren(indata, "Bom", "Fristående", "8");
  var test3b = new RedigeraGren(indata, "Hopp", "Ringar", "8");
};
```

Mer testkod – alt2: QUnit

```
test( "Datatyper", function() {
  ok( typeof RegistreraGren == "function" , "Klassen RegistreraGren() implementerad." );
  ok( typeof RedigeraGren == "function" , "Klassen RedigeraGren() implementerad" );
});

test( "Lägg till ny gren", function() {
  var grenar = [{name: "Bom", size: "10"}, {name: "Hopp", size: "12"}]; // Testdata
  ok(RegistreraGren(grenar, "Barr", "6"), ([{name: "Bom", size: "10"}, {name: "Hopp", size: "12"}, {name: "Barr", size: "6"}]), "Grenen registrerades.");
});

test( "Grenen finns redan", function() {
  var grenar = [{name: "Bom", size: "10"}, {name: "Hopp", size: "12"}]; // Testdata
  ok(RegistreraGren(grenar, "Bom", "10"), ([{name: "Bom", size: "10"}, {name: "Hopp", size: "12"}]), "Grenen finns redan.");
});

test( "Redigera gren", function() {
  var grenar = [{name: "Bom", size: "10"}, {name: "Hopp", size: "12"}]; // Testdata
  ok(RedigeraGren(grenar, "Bom", "Fristående", "8"), ([{name: "Fristående", size: "8"}, {name: "Hopp", size: "12"}]), "Array ok");
});
```

Version	Datum	Uppgift 7 – Reflektion	Författare
1.1	2014-07-12		dt222cc

Planerad tidsåtgång	Förbrukad tidsåtgång	Skillnad
22 h	31 h 30 h	9 h 30 min

Jag tyckte att denna labb var väldigt svårt, mycket nytt och svåra ämnen. Det blir svårt att planera något som man inte har gjort tidigare och att hitta information. Jag tyckte att jag planerade in för mycket tid i början men nu när man ha jobbat igenom uppgifterna insåg jag att jag behövde ännu mer tid. Uppgifterna krävde mycket förberedelser, det var svårt att veta om jag har gjort rätt eller inte.

Jag hade det svårt att bestämma mig om vilken användningsfall jag skulle använda. Det fanns svåra och mindre svåra men inte precis enkla. Jag fick göra om vissa uppgifter eftersom jag tyckte att jag gjorde dem på ett fel/dåligt sätt. Pågrund av det gick det åt mycket tid. Det är förmodligen att många andra har nästan samma intryck på att kursen är väldigt svårt att hänga med på.

Behöver mer "demo" som man kan titta på, så att man vet hur man ska göra. Liksom xUnit? junit? Hur fungerar dem, man får ju hitta all material själv, nästan, och det kan ta väldigt mycket tid för att hitta bra information/referenser. Liksom mer vägledning är skönt att ha om man ska ta sig igenom uppgifterna på ett bra och effektivt sätt.

Eftesom det är så mycket nytt så kan det vara svårt att hänga med även om det finns en röd tråd. Ja det kan vara så att ni vill att vi ska försöka förstå och ta hand om detta själva, men... svårt.. Det här med testningen, framförallt integrationstestningen, de är väldigt svårt att utföra utan att någon har visat hur det fungerar och går till, "live" eller med inspelat demo. Våra system kommer inte vara så avancerade så integrationstestningen är inte så smidig att göra. Om jag inte hade uppfattat det rätt då.