## Tidslogg

|  |  |  |
| --- | --- | --- |
| **Datum** | **Timmar** | **Arbetsbeskrivning** |
| 2014-12-21 | 4 t 30 m |  |
| 2014-12-22 |  |  |
|  |  |  |

Beräknad tidsåtgång 925 minuter / 15.4 timmar.  
Verklig tidsåtgång 1260 minuter / 21 timmar.

# Uppgift 1- Planera

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steg | Plan.  tid | Beskrivning | Verkl.  tid | Problem på vägen |
| 1 | 15 m | **Planera uppgiften**  Förstå alla uppgifter och planera dem. | 15 m | Trodde att jag förstod allt direkt, men så var inte fallet. Förstod omfattningen bättre när jag kom till uppgifterna, speciellt planeringen. |
| 2 | 15 m | **Skapa reposition på github** | 10 m | Inga problem |
| 3 | 30 m | **Uppgift 1 – Planera (grovplanering)** | 4 t | Förstår inte att jag skrev 30 min här som planerad tid. Att planera alla iterationer var väldigt tidskrävande och krävde mycket tankearbete, fastän det var grovplanering. |

Grovplanering över iterationer

## Planering av Iteration 1

**Bakgrund:**Eftersom i det första läget handlar om att bygga upp någonting viktigt att visa för kunden så fokuserar denna iteration på just detta. Under iterationen ska ett webbgränssnitt byggas upp efter Användningsfallen:

* Användningsfall 1: Skapa användare
* Användningsfall 2: Skapa tävlingstillfälle
* Användningsfall 3: Återställa inloggningsuppgifter

Dessa användningsfall fokuserar på enstaka funktioner i systemet och därav i princip kräver en färdig bas att stå; denna bas kommer inte att implementeras under denna iteration. Denna iteration fokuserar specifikt på att endast utforma de nödvändiga webbgränssnittsdelar som krävs för var och en av användningsfallen. Dock så ska dessa ha ett gemensamt tilltalande formspråk och dela på css stilmallar och eventuella bilder. All funktionalitet från användningsfallet kommer dock inte att realiseras. Datat kommer inte att sparas någonstans, utan det kommer bara att leva kvar i minnet tills fönstret stängs.

**Funktionella krav (efter prioritet):**

1. Rätta till ouppklarade fel i klasserna Competition och Event som upptäcktes i enhetstesterna när objekten Event och Competition interagerade med varandra.
2. Webbgränssnitt ska utformas och kopplas till befintlig funktionalitet skriven i klasserna: Competition och Event i användningsfallet 2: Skapa tävlingstillfälle. Competition och Event klasserna måste och ska fungera ihop med webbgränssnittet. Det är detta som ska visas för kund i slutet av iterationen (likt SCRUM).

**Icke funktionella krav:**

* Det nya webbgränssnittet ska vara uppfattas som tilltalande och proffsigt av kund.
* Webbgränssnittet ska gå fort att arbeta i.

**Tidsåtgång under iteration:**

8 timmar

**Delmål för iterationen:**

* I slutet av iterationen ska ett färdigt webbgränssnitt presenteras för kund med funktionalitet ifrån användningsfall nr 2.
* Alla fel är åtgärdade som framkom under tidigare enhetstest.

## Planering av Iteration 2

**Bakgrund:**Nu när vi har ett visuellt gränssnitt som vi kan utgå ifrån sedan den tidigare iterationen så kan vi tillämpa den även på andra användningsfall. I den här iterationen så realiserar vi användningsfall 1; Skapa användare. På den tillämpar vi sedan webbgränssnittet. All funktionalitet i användningsfallet kommer dock inte att skapas. Användaren skapas bara i minnet tills fönstret stängs. Systemet skickar inte heller ut e-post eller sms sill den nya användaren.

**Funktionella krav (efter prioritet):**

1. Administratör kan skapa användare efter typerna: administratör, domare, medlem av klubb.
2. De nya skapade användarna ska ha uppgifter om:
   1. typ av användare
   2. Användarnamn och lösenord
   3. E-post och mobilnummer

**Icke funktionella krav:**

* Det nya webbgränssnittet ska vara uppfattas som tilltalande och proffsigt av kund.
* Webbgränssnittet ska gå fort att arbeta i.

**Tidsåtgång under iteration:**

8 timmar.

**Delmål för iterationen:**

* Användningsfall 1 ska kunna presenteras för kund i ett färdigt webbgränssnitt.
* Nya klasser gällande användarfallet ska vara skapade.
* De skapade klasserna och webbgränssnittet ska vara ihopkopplade och fungera.

## Planering av Iteration 3

**Bakgrund:**I denna iterationen förverkligas användningsfall nummer 3; där användaren kan återställa sina inloggningsuppgifter. Det visuella webbgränssnittet tillämpas som gjordes efter iteration 1 tillämpas också här.

**Funktionella krav (efter prioritet):**

1. En sida finns där användare kan välja att återställa sina lösenord.
2. En fungerande captcha finns där användare kan validera sin mänsklighet.
3. Användaren anger sin e-postadress eller sitt mobilnummer och blir talad för att ett meddelande skickats till dem med en återställningslänk. Denna skickas dock inte i praktiken i detta iterationssteg.

**Icke funktionella krav:**

* Det nya webbgränssnittet ska vara uppfattas som tilltalande och proffsigt av kund.
* Webbgränssnittet ska gå fort att arbeta i.

**Tidsåtgång under iteration:**

8 timmar.

**Delmål för iterationen:**

* Användningsfall 3 ska kunna presenteras för kund i ett färdigt webbgränssnitt.
* Nya klasser gällande användarfallet ska vara skapade.
* De skapade klasserna och webbgränssnittet ska vara ihopkopplade och fungera.

**Delmål i iterationen:**Realisera Användningsfall 1.

**Krav ifrån användningsfallet:**

Funktionella krav:

Icke funktionella krav

1. Skriv tester för Användningsfall 1.
2. Realisera Användningsfall 1, se till att testerna går igenom felfritt.

## Iteration 3 – Delmål 3

1. Skriv tester för Användningsfall 3
2. Realisera Användningsfall 3, se till att testerna går igenom felfritt.

|  |  |  |  |
| --- | --- | --- | --- |
| Uppgift 1 | Uppgift 2 | Uppgift 3 | Uppgift 4 |
| Inception | Elaboration | Construction | Transition |
|  |  |  |  |

Iteration 1

## Kravanalys:

**Analys av krav från projektplanering**

* **Funktionellt krav, prioritet 1:**  
  *”Rätta till ouppklarade fel i klasserna Competition och Event som upptäcktes i enhetstesterna när objekten Event och Competition interagerade med varandra. ”*

Kravet är rimligt och är grunden för hela iterationen och krävs för att kunden ska kunna få en fungerande demo. Prioritet 1.

* **Funktionellt krav, prioritet 2:***”Webbgränssnitt ska utformas och kopplas till befintlig funktionalitet skriven i klasserna: Competition och Event i användningsfallet 2: Skapa tävlingstillfälle. Competition och Event klasserna måste och ska fungera ihop med webbgränssnittet.”*

Kravet är rimligt och måste fungera.

* **Icke funktionella krav:**

*”Det nya webbgränssnittet ska vara uppfattas som tilltalande och proffsigt av kund.”*

*”Webbgränssnittet ska gå fort att arbeta i.”*

**Förtydligade krav efter prioritet**

1. Klasserna Competition och Event ska gå igenom enhetstesterna.
2. Ett nytt webbgränssnitt ska designas som uppfattas som professionellt av kund.
3. Webbgränssnittet kopplas ihop med Competition- och Eventklasserna och fungera felfritt
4. En körbar demo ska genereras som kan visas upp för kund.
5. Webbgränssnittet ska gå fort att arbeta i.

## Klassdiagram

## Detaljerad planering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Steg** | **Beskrivning** | **Planerad tidsåtgång** | **Verklig tidsåtgång** | **Problem på vägen** |
| 1 | Analysera kraven | 15 m | 15 m | Gick bra utan problem. |
| 2 | Rätta till ouppklarade fel i klasserna Competition och Event som upptäcktes i enhetstesterna. | 1 t | 52 m | Stötte på problem som var större än väntat. Många loopar i domarjämförelse, vilka löste sig. |
| 3 | Designa ett tilltalande webbgränsnitt. | 2 t | 1 t 45 m | Flöt ihop med ihopkopplandet av klasserna. Designprocessen var kontinuerlig |
| 4 | Koppla ihop webbgränsnittet med funktionaliteten i Competition- och Eventklasserna. | 2 t | 1t 30m (så länge) | Svårt, webbgränsnittet har svårt att tillämpa klasserna på sättet som de är skrivna. |
| 5 | Testning av webbgränssnittet. Fungerar den som utlovat? Åtgärda eventuella fel. | 2 t |  |  |
| 6 | Skriv reflektion | 30 m |  |  |

Rapport:

Omskrivning av klassen, lägga till funktionalitet för HTML händelsekoppling och rendering av html

Webbgränssnitt ska utformas och kopplas till befintlig funktionalitet skriven i klasserna: Competition och Event i användningsfallet 2: Skapa tävlingstillfälle. Competition och Event klasserna måste och ska fungera ihop med webbgränssnittet. Det är detta som ska visas för kund i slutet av iterationen (likt SCRUM).

## Mål för iterationen:

Fixa fel som upptäcktes i testerna vid labb 3, mer specifikt:

* Det ska inte gå att lägga till en deltävling i tävlingen där deltävlingens starttid är mindre än tävlingens starttid.
* Det ska inte gå att lägga till en deltävling i tävlingen där deltävlingens sluttid är större än tävlinggens sluttid
* Det ska inte gå att lägga till en deltävling i tävlingen där deltävlingen saknar viktiga egenskaper/värden.
* Det ska inte gå att lägga till deltävlingen i tävlingen om den redan är inlagd i tävlingen.
* Det ska inte gå att lägga till deltävlingen i tävlingen om domaren som är anmäld till deltävlingen redan finns inlaggd i en annan deltävling under samma period.

# Uppgift 2 - Iteration 2

## Detaljerad planering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Steg** | **Beskrivning** | **Planerad tidsåtgång** | **Verklig tidsåtgång** | **Problem på vägen** |
| 1 | Skapa klasser för | 2 t |  |  |
| 2 | Tillämpa webbgränssnittet från tidigare iteration. | 2 t |  |  |
| 3 | Se funktionella krav 2. | 2 t |  |  |
| 4 | Testning av webbgränssnittet. Fungerar den som utlovat? Åtgärda eventuella fel. | 2 t |  |  |

# Uppgift 2 - Iteration 3

## Detaljerad planering

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Steg** | **Beskrivning** | **Planerad tidsåtgång** | **Verklig tidsåtgång** | **Problem på vägen** |
| 1 | Skapa klasser för | 2 t |  |  |
| 2 | Tillämpa webbgränssnittet från tidigare iteration. | 2 t |  |  |
| 3 | Se funktionella krav 2. | 2 t |  |  |
| 4 | Testning av webbgränssnittet. Fungerar den som utlovat? Åtgärda eventuella fel. | 2 t |  |  |

Captcha: https://developers.google.com/recaptcha/docs/display