



Utveckling av ett riskhanteringssystem

Jan Leppänen

Examensarbete
Informations- och medieteknik
2016

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations- och medieteknik
Identifikationsnummer:	5540
Författare:	Jan Leppänen
Arbetets namn:	Utveckling av ett riskhanteringssystem
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	Falck Oy
<p>Sammandrag:</p> <p>Examensarbetet behandlar utvecklingen av ett riskhanteringssystem för Falck Oy. I arbetet genomgås det krav som ställs av arbetarskyddslagen och kvalitetshanteringsstandarden samt specifikationer som företaget har för systemet. Det tas upp goda principer för utveckling av system samt punkter som det lönar sig att tänka på under utvecklingen. Systemet utvecklas på basis av en iterativt utvecklingsmodell. Arbetet beskriver de olika utvecklingsfaser som fanns under projektet samt hur de krav som ställdes utfördes. En huvudpunkt i nya systemet är användarvänlighet. Det beskrivs hur användarvänligheten tas i beaktande under utvecklingen av systemet. Resultatet för arbetet är en genomgång av krav och specifikationer som företaget ställt för systemet samt med en användarenkät angående användarvänlighet där det nya och det gamla riskhanteringssystemet jämförs.</p>	
Nyckelord:	Riskhanteringssystem, Användarvänlighet, PHP, PDO, Bootstrap, MySQL
Sidantal:	
Språk:	Svenska
Datum för godkännande:	

DEGREE THESIS	
Arcada	
Degree Programme:	Information and media technology
Identification number:	5540
Author:	Jan Leppänen
Title:	Development of a risk management system
Supervisor (Arcada):	Johnny Biström
Commissioned by:	Falck Oy
<p>Abstract:</p> <p>This thesis is about the development of a risk management system for Falck Oy. The thesis includes requirements set by the Occupational Safety and Health Act, Quality management systems standard and specifications set by the company for the new system. The topics include good practices for system development and tips to use under the development. The system is developed by using an iterative development model, and the different phases and techniques used for the development is described in the thesis. A main point in the new system is usability, which is taken into account during the development process. The result of the completed product is measured by a review of the given requirements and a survey for the users, comparing the old and the new system.</p>	
Keywords:	Risk management system, Usability, PHP, PDO, Bootstrap, MySQL
Number of pages:	
Language:	Swedish
Date of acceptance:	

INNEHÅLL

1	Inledning.....	7
1.1	Krav	8
1.2	Specifikationer	8
1.2.1	<i>Anmälningsförfarande</i>	8
1.2.2	<i>Användarhantering</i>	9
1.2.3	<i>Användarvänlighet</i>	9
1.3	Avgränsning.....	10
2	Alternativa lösningar på problemet	10
2.1	Färdiga programvara	10
2.2	Självutvecklad programvara	11
2.3	Val av lösning	12
3	Utveckling	13
3.1	Goda principer för utveckling.....	13
3.2	Planering	15
3.3	Utvecklingsfas	16
3.3.1	<i>Avvikelseanmälan</i>	16
3.3.2	<i>Avvikelsehantering</i>	19
3.3.3	<i>Administration</i>	23
3.3.4	<i>Rapportering</i>	24
3.4	Användarvänlighet.....	25
3.4.1	<i>Anmälan</i>	26
3.4.2	<i>Hantering</i>	27
3.5	Testning.....	28
3.6	Införandet av systemet	29
4	Systembeskrivning.....	29
4.1	Gränssnitt	29
4.2	Datastruktur	32
5	Evaluering av det nya riskhanteringssystemet	34
5.1	Genomgång av krav	34
5.2	Användarenkät	35
6	Slutsatser och diskussion.....	41
	Källor	42

Figurer

Figur 1. Grunden för iterativ utvecklingsmodell	13
Figur 2. Feedback enkät.....	14
Figur 3. Avvikelseanmälan, modell version	17
Figur 4. Avvikelseanmälan, hårdkodad information	17
Figur 5. Administrativ för typer och ställen	18
Figur 6. Visandet av aktiva avvikelse typer	19
Figur 7. Avvikelsehantering, startsida	20
Figur 8. Avvikelsehantering, riskevaluering första iterationen	21
Figur 9. Förklaringsenkät för avvikelsen.....	22
Figur 10. Färdiga avvikelsehanterings sidan	23
Figur 11. Administrativ, användarvy.....	24
Figur 12. Avvikelse rapportering	25
Figur 13. Anmälningenkät, säkerhetsanmälan	26
Figur 14. Anmälningenkät, kvalitetsanmälan	27
Figur 15. E-post funktion om nya avvikelser	30
Figur 16. Användarrättighetskontroll	30
Figur 17. Kommentarsektion.....	31
Figur 18. Databasstruktur, avvikelse tabeller	33
Figur 19. Databasstruktur, användartabeller	34
Figur 20. Enkätfråga 8, resultat	40
Figur 21. Enkätfråga 9, resultat	40
Figur 22. Enkätfråga 10, resultat	40

Tabeller

Tabell 1. Enkätfråga 1, resultat	36
Tabell 2. Enkätfråga 2, resultat	36
Tabell 3. Enkätfråga 3, resultat	37

Tabell 4. Enkätfråga 4, resultat	38
Tabell 5. Enkätfråga 5, resultat	38
Tabell 6. Enkätfråga 6, resultat	39
Tabell 7. Enkätfråga 7, resultat	39

1 INLEDNING

Behovet för ett nytt riskhanteringssystem uppstod på bilbärgningsföretaget Falck Oy när man konkurrensatte personalens sjukvårdsförsäkring, eftersom försäkringsbolaget erbjöd ett riskhanteringssystem i samband med försäkringen. Riskhanteringssystemet som försäkringsbolaget erbjöd var ett webbaserat system, där personalen kan anmäla ifall det förekommer några riskavvikelser såsom att någon skadar sig under arbetstid.

Examensarbetet handlar om utvecklingsprocessen av ett riskhanteringssystem för Falck Oy. Företaget har som önskemål att också kunna använda systemet för hantering av kvalitetsavvikelser, enligt kvalitetshanteringsstandarden ISO 9001:2015. Systemet innehåller två delar, anmälning av risk- och kvalitetsavvikelser och en hanteringsdel där man hanterar de inlämnade anmälningarna. Informationen om avvikelserna samt deras hantering skall sparas i en databas som dokumentation, vilket arbetarskyddslagen 23.8.2002/738 kräver.

I arbetet tas det upp saker som, olika alternativ för att lösa problemet att anskaffa ett nytt riskhanteringssystem, iterativt sätt att utveckla, samt goda principer som det lönar sig att tänka på då man utvecklar ett system. Eftersom användarvänligheten är en viktig punkt, tas det upp vad som kan definieras som användarvänlighet och hur det tas i beaktande i utvecklingen, samt resultat om användarenkät angående det nya systemet.

Målsättningen är att företaget skulle ha ett riskhanteringssystem som fyller de krav och specifikationer som är satta. Systemet skall vara webbaserat på grund av företagets affärsverksamhet, då cirka hälften av personalen inte jobbar på kontoret, utan kör bärgnings- eller servicebil och inte har tillgång till en dator. Med ett webbaserat system möjliggör man tillgång till systemet både via en mobilapparat och en dator.

1.1 Krav

Arbetsgivaren är skyldig att tillräckligt systematiskt ta i beaktande arbetets och verksamhetens risker och olägenheter som beror på arbetet, arbetstider, arbetslokalen och arbetsmiljön i övrigt. Ifall olägenheterna inte kan undanröjas måste arbetsgivaren bedöma deras betydelse för arbetstagarnas säkerhet och hälsa. Såna sammanhang är till exempel olycksfall, yrkessjukdomar och arbetsrelaterade sjukdomar och risksituationer som förekommit. (Finlex. 2002, 10§)

Detta betyder att avvikelserna måste dokumenteras och värderas så att det är möjligt att identifiera de största riskfaktorerna inom arbetet.

Enligt standarden ISO 9001 måste man bestämma en process för hela organisationen för kvalitetshandling, samt definiera punkter som kommer att följas. Det skall också vara möjligt att följa, mäta och om möjligt, analysera händelserna. (Suomen Standardisoimisliitto SFS. 2015, 4.1)

1.2 Specifikationer

Medan Falck har haft i bruk det nuvarande elektroniska risk- och kvalitetshandlings systemet, samlade jag feedback från användarna om olägenheter som systemet har. Tillsammans med riskhanteringskoordinatorerna valde vi de mest väsentliga kraven som det nya systemet skulle uppfylla samt att genomföra vissa andra förbättringar.

1.2.1 Anmälningförfarande

Alla skall ha möjlighet att anmäla risk- och kvalitetsavvikelser utan att behöva registrera sig eller ha ett användarkonto i systemet. Ifall man vill göra en anmälan anonymt måste det vara möjligt, medan ifall man gör anmälan och meddelar e-posten, så får man en bekräftelse när fallet är behandlat.

Eftersom en stor del av personalen kör bärgningsbil eller servicebil, måste det gå enkelt att göra en anmälan med en mobilapparat.

När man gör en anmälan måste man kunna lägga med en bild. På mobilsidan vore det bra ifall man direkt kan fotografera med telefonen utan att separat behöva ta foton och sedan klistra in dem i anmälan. I nuvarande systemet borde det vara möjligt, men det fungerar inte.

Vid anmälningstillfället måste man kunna välja om det gäller en riskavvikelse eller kvalitetsavvikelse.

1.2.2 Användarhantering

Det skall finnas möjlighet att själv inom företaget hantera användarkonton som finns i systemet. I nuvarande systemet måste man beställa nya användare från programvarans underhållare, med hjälp av e-legitimation, och tar det upp till ca en vecka för användarkontot att aktiveras, vilket man vill undvika i det nya systemet.

För tillfället finns det inga användargrupper för att dirigera anmälningarna till den rätta personen som hanterar fallen, utan alla anmälningar visas till alla användaren. För att göra det bekvämare för användaren att hantera fall, borde var och en bara se anmälningar som hör till deras delområde. På grund av att det inte finns några användargrupper, får alla användare e-post meddelande då det kommer en ny anmälan. Vilket ledde till att efter en tid började användarna ignorera e-posten och på grund av det missa anmälningar som hörde till dem. I nya systemet vill man att bara de som hanterar det specifika delområdet som anmälan angår, får e-post meddelandet.

1.2.3 Användarvänlighet

A och o i det nya systemet skall vara användarvänlighet. Med att göra systemet användarvänligt undviker inte användaren att fylla i avvikelseanmälningar eller att hantera dem. Systemet skall vara klart och tydligt samt lättförstått, så att man inte behöver

fundera eller lära sig på nytt hur man använder systemet varje gång. Anmälningssidan skall innehålla fält bara för den information som krävs för den specifika avvikelsetypen, på hanteringssidan skall man behöva fylla i bara den relevanta informationen och åtgärder som fallet kräver.

1.3 Avgränsning

Systemet innehåller bara risk- och kvalitetsavvikelseanmälning- och hanteringsprocessen. Till det hör arbetarskyddslagen 23.8.2002/738 kapitel 2, Arbetsgivarens allmänna skyldigheter, samt villkor som omfattar risk- och kvalitetsavvikelseanmälning och -hantering för ISO 9001:2015 kvalitetshanteringsstandard.

Fastän det i ISO 9001 standarden krävs ett dokumentbehandlingssystem för processer inom företaget, kommer det inte att vara med i detta arbete på grund av att det finns ett separat system som redan är i bruk för detta ändamål.

2 ALTERNATIVA LÖSNINGAR PÅ PROBLEMET

2.1 Färdiga programvara

Efter den preliminära genomgången med riskhanteringskoordinatören om önskemål som systemet skulle innehålla, var det klart att alternativet att behålla det nuvarande systemet inte var rimligt.

En färdig programvara som skulle fylla största delen av kraven, var Granite Partner:s system, Granite för riskhantering. I systemet är det möjligt att själv definiera riskpunkterna som finns inom företaget. Det finns också en mobilversion som var ett av de krav som var satta.

Under utredningen av lämpliga programvaror som skulle fylla de behov som fanns, upptäcktes det att urvalet av lämpliga riskhanteringssystem var litet. Eftersom största delen av de leverantörer som erbjuder ett omfattande system som innehåller alla de krav och specifikationer som var uppsatta, var systemen riktade mer för stora företag och kostnaderna enligt det. Ett lämpligt alternativ skulle ha varit Logic Manager men prisklassen på den var utanför budgeten som var satt.

2.2 Självutvecklad programvara

Som alternativ för att utveckla systemet själv, valde jag tre stycken olika system. Vaadin, MEAN Stack och en kombination av HTML, JavaScript, CSS, Bootstrap och PHP. Alternativ för databasteknik var MySQL och MongoDB.

- Vaadin (Vaadin 2015)
Vaadin är ett gratis Java ramverk, menat för webbaserade applikationer. Med hjälp av Vaadin kan man skapa webbaserade applikationer bara med att skriva kod i Java. Ramverket bygger upp HTML och CSS delen automatiskt. Positivt med Vaadin är att applikationerna körs på server-sidan, och inte på klient-sidan. Vilket gör applikationen säkrare, eftersom applikationens all logik utförs på servern istället för klient-sidan (Vaadin 2016). Ett annat positivt tillägg som jag personligen tycker om är Vaadin Sampler, vilket är ett grafiskt hjälpmedelssystem som innehåller exempel kod hur man implementerar de olika delar som finns att använda.
- MEAN Stack (MongoDB, Express.js, Angular.js, Node.js)
Med termen MEAN stack hänvisar man till en samling av JavaScript baserade teknologier för utveckling av webbapplikationer. Applikationer som utvecklas med MEAN stack är så kallade single-page applikationer, vilket betyder att applikationen är avsedd att användas på en sida. Det vill sägas att all HTML, JavaScript och CSS kod samt all information från servern blir laddad på första

sidan, och innehållet på sidan ändras dynamiskt, utan att behöva ladda upp en ny sida (mean.io 2016).

- HTML, JavaScript, CSS, Bootstrap, PHP

Kombinationen av dessa programmeringsspråk tillsammans med Bootstrap ramverket, är ett mer “klassiskt” sätt att skapa webbsidor, medan de två tidigare nämnda är mer moderna alternativ. Bootstrap är ett front-end ramverk, som innehåller gränssnittkomponenter. Med hjälp av Bootstrap går det lätt att skapa dynamiska och responsiva webbapplikationer (Bootstrap 2016).

2.3 Val av lösning

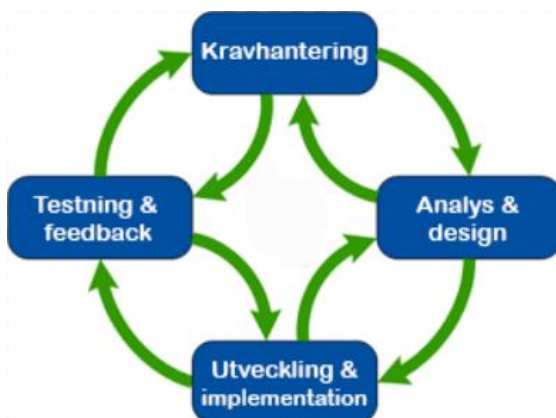
Efter att alternativen för det nya systemet var kartlagda gjorde vi beslutet att utveckla systemet själv på grund av att få systemet att vara sådant som möter de behov som företaget har samt på grund av kostnadsbesparingar.

Med tanke på de krav och funktioner som systemet skulle innehålla, skulle MEAN stack inte vara ett lämpligt val på grund av sättet det är menat att användas. Det enda negativa med Vaadin är att ramverket är relativt nytt, och det på grund av det finns det inte så mycket hjälp på t.ex. sidan stackoverflow (Stackoverflow 2016), då man stöter på problem. På basis av det samt mina tidigare kunskaper samt erfarenhet och kunnande inom företaget, valde jag att använda PHP, JavaScript, HTML, CSS och Bootstrap. Som startunderlag för Bootstrap valde jag att använda SB Admin 2 (Startbootstrap 2014).

Med den kombinationen jag valt att programmera systemet är det möjligt att använda både MySQL eller MongoDB, beslöt jag att använda MySQL som databasteknologi. Eftersom MySQL databaser är mer optimerade för att hantera sökningar med specifika kriterier, blir rapporteringen som krävs av systemet lättare (Hall 2014). När MongoDB:s fördelar istället är att skalbarhet och hantering stora mängder av data, vilket inte är nödvändigt för detta system eftersom mängderna av data inte kommer att vara utanför MySQL:s förmåga. (Kaplan 2014)

3 UTVECKLING

Systemet utvecklades med en iterativ utvecklingsmodell, vilket betyder att man delar upp utvecklingen av ett större system i mindre bitar. I den iterativa utvecklingsmodellen arbetar man med ett delmoment från början till slut, d.v.s. planerar, programmerar och testar. Det gör man för alla delmoment tills systemet är färdigt utvecklat (Rouse 2011). Då man jobbar med en iterativ modell går det lättare att göra förändringar medan utvecklingen pågår. Med att använda ett så kallat test-drivet utvecklingssätt, minskar man antalet buggar som förekommer i koden (Verisoneone 2015).



Figur 1. Grunden för iterativ utvecklingsmodell

3.1 Goda principer för utveckling

Förrän man börjar utveckla ett system är det bra att veta hur man skall börja och hur man får slutresultatet att vara bra. Kampf har samlat några punkter hur man kan definiera vad bra kod är (Kampf 2008). Enligt Kampf är det en balans mellan fler saker som exempel begriplighet, läsbarhet, mängd av fel i koden samt modifierbarhet.

Guzel påpekar goda principer som det lönar sig att följa för att skriva läsbar kod. Kommentera i koden vad funktionen gör, men man skall undvika onödig kommentering för självklarheter. Indragningen på koden och namngivningen av variabler skall vara konsekvent. När man använder SQL specialord, lönar det sig att skriva med versaler, för att göra det lättare att skilja ut dem från koden (se Figur 6 för exempel på kod) (Guzel 2011).

Det räcker inte bara att skriva kod som går att läsa, utan koden måste också vara buggfri. I Gurocks artikel om praktiska tips för att bygga buggfri kod, nämner han viktiga punkter såsom att ha både dedicerade och andra användare för att testa programmet för att upptäcka problem. Användaren skall också ha möjlighet att rapportera ifall det uppstår fel samt kunna ge feedback. För detta skapades det en enkät som användaren kunde fylla i (se Figur 2). Med hjälp av en bra utvecklingsmiljö som färdigt innehåller en debugger som upptäcker fel redan i programmeringsfasen minskar på mängden buggar som uppstår (Gurocks 2007). En bra utvecklingsmiljö för PHP programmering är PHPStorm IDE från JetBrains, som var vald som bästa PHP utvecklingsmiljö i Sitepoints undersökning (Sitepoint 2014).

Palautteen ilmoittaminen

Palautteen tiedot

Palautteen tyyppi:

Bugi
 Kehitysidea

Otsikko:

Kuvaus:

Mikäli kyseessä oli bugi, kuvaile mitä teit kun virhe ilmaantui

Mikäli kyseessä on bugi, ole ystävällinen ja ota kuvankaappaus virheestä ja liitä se ilmoitukseen

Lisää kuva:

No file chosen

Figur 2. Feedback enkät

För ett bra system räcker inte bara läsbar och buggfri kod, utan databassidan måste också vara väl gjord. Microsoft (Microsoft 2010) definierar en bra databasdesign med några principer. Den första är att redundant information är dåligt på grund av att det för utrymme och ökar risken på fel och inkonsekvenser. Andra principen är korrektheten och fullständigheten av informationen. Ifall databasen innehåller felaktig information, kommer alla rapporter som är gjorda på basis av informationen som databasen innehåller vara felaktigt. För att undvika dessa problem lönar det sig att dela informationen i databasen i ämne-baserade tabeller, för att minska redundant information. Tabellerna skall innehålla information för att kunna knyta relaterade tabeller ihop, som till exempel med foreign key:s i MySQL.

Istället för att använda normal PHP för att göra databasförfrågningar, lönar det sig att använda PHP Data Objects eller PDO. Med hjälp av PDO undviker man SQL injektioner (Wurzer 2012), vilket betyder att någon användare själv kan skriva SQL kommandon för att antingen få ut data som inte är avsett att vara tillgängligt eller skriva in felaktig data. Istället för att skriva hårdkodade värden i SQL kommandon efter '=' tecken, som exempel "SELECT * FROM users WHERE type='admin'" använder man i PDO variabler för värden, på det sättet kan inte någon annan lägga in sina egna kommandon (se Figur 6 för exempel på PDO).

3.2 Planering

Det första man skall göra innan man börjar programmera ett system är att samla den information som man behöver, såsom hurudan information som kommer att sparas i databasen eller vilken typ av användare det kommer att finnas (Perks 2006). Med att samla information om vad som behövs kan man slå fast helheten om vad som programmet kommer innebära. Detaljen kan man finslipa medan man jobbar vidare. Då är det lättare att komma igång med utvecklingen och att välja rätt metoder som man jobbar med. När man jobbar med en iterativ modell behöver inte planeringen vara fullständigt klar innan man börjar utveckla de enskilda delmoment (Forsyth 2013).

3.3 Utvecklingsfas

För att minimera skadan ifall det nya systemet inte skulle bli klart i tid, beslöts det att börja utvecklingsprocessen från det mest väsentliga, d.v.s. avvikelseanmälningen. Med att ha anmälningsprocessen klar fastän resten av systemet inte skulle vara det, kan man göra anmälningar och inga avvikelser skulle gå förlorade.

Därefter skulle hanteringssidan utvecklas, så att det var möjligt att följa med anmälningar och hantera dem. Vyn skulle innehålla information om hur man undviker situationen i framtiden samt riskklassen.

I sista fasen skulle det komma tilläggssegenskaper såsom möjlighet att mata in kommentarer på fall, fält för estimerade kostnader och verkliga kostnader, färgkodning riskklasserna, möjlighet att skicka en enkät till användaren ifall man behöver mer information om t.ex. vad som hänt.

När anmälnings- och hanteringsdelmomenten är klara, utvecklas det en grafisk administrativ del för att hantera användare, användarrättigheter, klienter samt avvikelsetyper och organisationsställen.

3.3.1 Avvikelseanmälan

Utvecklingen av anmälningsenkäten började med att skapa en modellversion, utan några back-end förbindelser. Med att skapa en modellsida får man användarna engagerade från början till att vara med i utvecklingsprocessen. Då får man feedback under utvecklingen om problem eller ifall det fattas någon information som behövs samlas in (se Figur 3).

Havainnon ilmoittaminen

Havainnon tiedot

Havainnon tyyppi:

- Turvallisuushavainto
- Laatuhavainto
- Ympäristöhavainto

Paikka:

Vahinkopäivä: Kellonaika:

Otsikko:

Kuvaus:

Lisää kuva:

No file chosen

Figur 3. Avvikelseanmälan, modell version

I den första versionen var all information hårdkodad vilket betyder att man måste ändra i själva koden ifall man ville lägga till eller ta bort något som t.ex. typen av anmälan (se Figur 4).

```
<div class="form-group">
  <label>Havainnon tyyppi: </label>

  <div class="radio">
    <label>
      <input type="radio" name="optionsRadios" id="optionsTurvallisuus"
        value="turvallisuus" checked>Turvallisuushavainto
    </label>
  </div>
  <div class="radio">
    <label>
      <input type="radio" name="optionsRadios" id="optionsLaatu"
        value="laatu">Laatuhavainto
    </label>
  </div>
  <div class="radio">
    <label>
      <input type="radio" name="optionsRadios" id="optionsYmparisto"
        value="ymparisto">Ympäristöhavainto
    </label>
  </div>
</div>
```

Figur 4. Avvikelseanmälan, hårdkodad information

Att engagera användaren från början visade sig vara ett bra beslut på grund av att redan under den första iterationen av anmälningsenkäten kom det feedback om ändringar som

skulle göras, som exempel att istället för ”skadedatum” skulle det vara ”händelsedatum” för att alla anmälningar inte är skador.

I den andra iterationen skapades grunden till databasen för informationen som finns på enkäten samt annan information som behövs senare i behandlingsprocessen såsom vad anmälans status är, när anmälan är gjord och när den är stängd.

I den tredje och sista iterationen för anmälningsdelmomenten kom med den administrativa delen, en tabell i databasen för de olika typerna av avvikelser och olika organisationsställen. Med hjälp av den administrativa delen går det att lägga till nya värden och välja vilka är aktiva och visas i anmälningsenkäten (se Figur 5) då minskar mängden som behövs programmeras och koden blir lättläst (se Figur 6).

Vaihnkotyytit ja paikat		Lisää
Nimi	Tyyppi	
Hinaus ja Tiepalvelut	Paikka	
Järjestelmät	Paikka	
Kelatie	Paikka	
Laatuhavainto	Havainnon tyyppi	
Medical	Paikka	
Palvelukeskus	Paikka	
Sisäisetpalvelut	Paikka	
Tietoturvahavainto	Havainnon tyyppi	
Turvallisuushavainto	Havainnon tyyppi	
Vakuutuskäsittely	Paikka	
Verkosto	Paikka	
Ympäristöhavainto	Havainnon tyyppi	

Figur 5. Administrativ för typer och ställen

```

//list all active incident locations
$stmt = $db->prepare("SELECT * FROM incidentTypes WHERE inc_Spec=:location
                    AND inc_Active=:activeStatus ORDER BY inc_Spec ASC");
$stmt->bindParam(':location',$location);
$stmt->bindParam(':activeStatus',$classActive);
$stmt->execute();
while ($row = $stmt->fetch()) {
    echo '<option value="' . $row['inc_Attribute'] . '">' . $row['inc_DisplayName'] . '</option>';
}

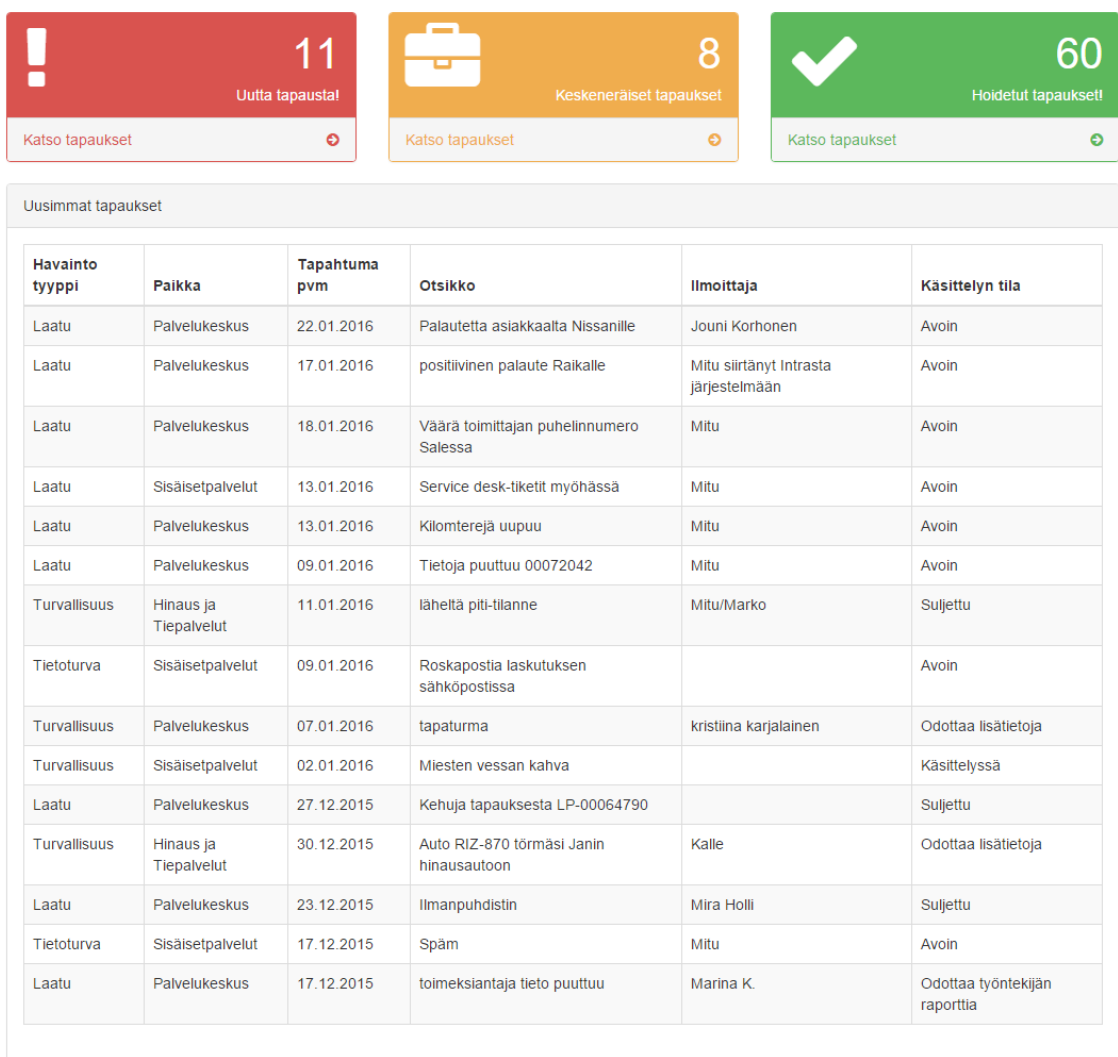
```

Figur 6. Visandet av aktiva avvikelsetyper

3.3.2 Avvikelsehantering

När databasen var klar för anmälningsdelen, kunde man börja utveckla hanteringsmodulen. Då man börjar hantera anmälningar, är startsidan den som man öppnar först. På startsidan skall man kunna se all den essentiella informationen om nya och obehandlade fall. Startsidan innehåller tre rutor och en lista på de 15 nyaste anmälningarna (se Figur 7). De tre rutor är färgkodade. Röd för nya, obehandlade fall. Gul för fall som personen som behandlar fallen har öppnat men saknar någon information eller det finns något uppdrag man måste göra förrän man kan markera fallet som behandlat. Grön för behandlade fall.

Med att trycka på någon av de tre rutorna listas det ut bara fall som behandlar den kategorin. För att hantera eller se noggrannare information om fall får man upp den med att trycka på fallet i listan.

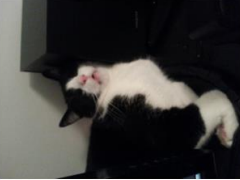


Figur 7. Avvikelsehantering, startsida

I avvikelsehanteringsvyn skulle det finnas all den information som finns i anmälan och en sektion för att avgöra riskallvarlighet, där hanteraren fyller i orsaken för händelsen, hur man undviker att saken upprepas, vad som gjorts för saken skull, samt en tabell för att välja riskklassen (se Figur 8 för första iterationen av avvikelsehanteringsvyn). För att följa goda databaspraxis, skapades det en egen tabell för riskevalueringen, med en relation till tabellen som innehåller information om själva risken.

Tapaus: Malmi - Laatuhavainto 9.6.2015

Havainnon tyyppi: Laatuhavainto Tarkennus: -Valitse- Ilmoittajan nimi: Sähköposti:

Paikka: Toimisto Tapauksen kuva: 

Vahinkopäivä: 9.6.2015 Kellonaika: 21:30

Otsikko: Kissa nukkuu Tapausnumero: KI-554

Kuvaus: Kissa nukkuu oimittaisessa asennossa

Riskin vakavuus:

Syy tapahtumalle: Tottelemattomuus

Ehkäisymenetelmät:

Toimenpiteet:

Riskiluokitus:

●	●	●	●
●	●	●	●
●	●	●	●
●	●	●	●

Figur 8. Avvikelsehantering, riskevaluering första iterationen

Efter den första iterationen kom det upp att den normala riskklassificeringstabellen inte fungerade i vårt fall. En normal riskklassificeringstabell baserar sig på hur sannolikt och hur allvarlig fallet är, men eftersom systemet också hanterar kvalitetsavvikelser, finns det situationer där sannolikheten är liten men allvarligheten är intolererbar.

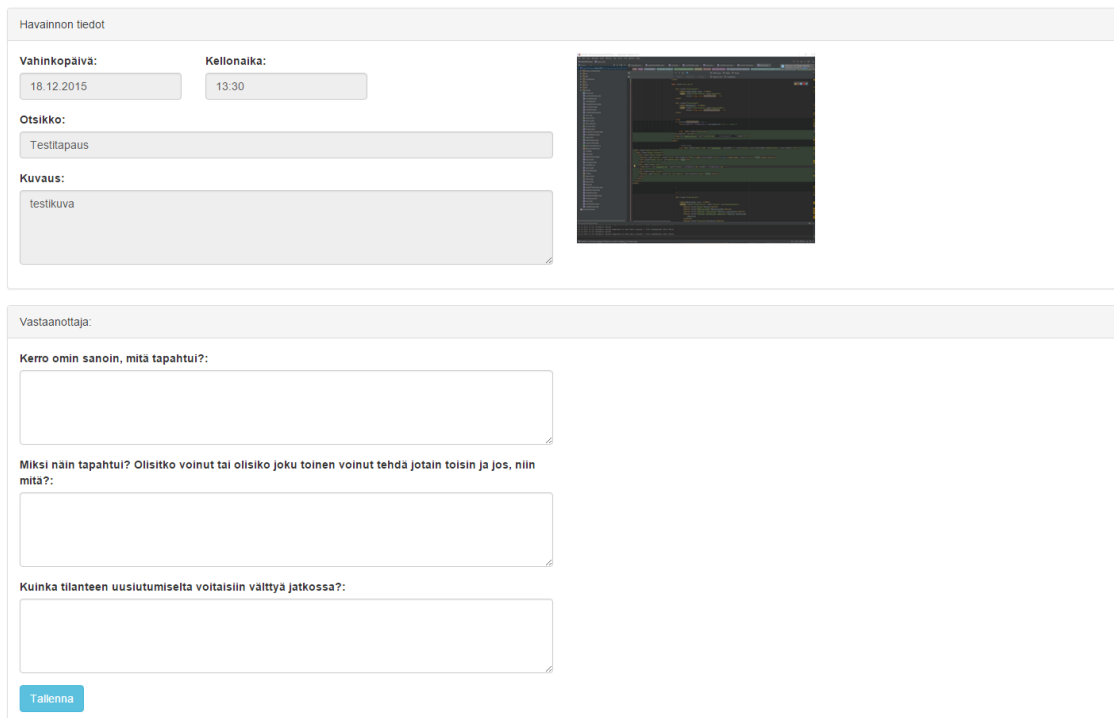
I den andra iterationen delades riskevalueringen i två delar, sannolikhet och allvarlighet, vilket tillåter alla variationer av riskkvalificering. Som nytt i iterationen kom med möjligheten för hanterare att kommentera fall och fält för kostnader som uppstått på grund av avvikelser. Eftersom evalueringssättet ändrades till en rullgardinsmeny, var man tvungen att lägga till färgkodningen i CSS för att hantera rullgardinsmenyfälten samt kod för att hantera färgen då man byter värde. Under testningen av iterationen förekom det inga problem eller ändringar som skulle göras.

Tredje iterationen hämtade med sig möjligheten att skicka e-post till användaren ifall det fattades någon information eller förklaring på vad som hänt enligt dem. Om riskklassen är evaluerad som måttlig eller högre, är det ikryssat automatiskt att skicka e-post till

användaren för en förfrågan över utredning över vad som hänt. Ifall hanteraren väljer att inte skicka, måste man ge en förklaring varför inte.

E-posten som användaren får innehåller text att man måste fylla i en enkät, där man förklarar vad som hänt, samt en länk till enkäten. Enkäten innehåller datum, klockslag, rubrik, beskrivning av avvikelser, bild ifall det finns, samt enkäten som man skall fylla i där man förklarar vad som hänt, varför hände det och hur skulle man kunna undvika det i framtiden. All information om vem som gjort anmälan och riskevalueringen syns inte för användaren (se Figur 9).

Tapaus: Sisäisetpalvelut - Laatu 18.12.2015



Havainnon tiedot

Vahinkopäivä: 18.12.2015 Kellonaika: 13.30

Otsikko: Testitapaus

Kuvaus: testikuva

Vastaanottaja:

Kerro omin sanoin, mitä tapahtui?:

Miksi näin tapahtui? Olisiko voinut tai olisiko joku toinen voinut tehdä jotain toisin ja jos, niin mitä?:

Kuinka tilanteen uusiutumisetä voitaisiin välttää jatkossa?:

Tallenna

Figur 9. Förklaringsenkät för avvikelser

I den fjärde och sista iterationen för hanteringsdelen, ändrades informationen som fylls i under riskevalueringssktionen till vad som hände innan fallet, åtgärder och ett nytt fält för ansvarig person. Ifall man vill att någon specifikt är ansvarig för att göra åtgärder kan man lägga personen som ansvarig, och då får personen e-post om saken. Fältet för orsaken

flyttades till sektionen om avvikelser, och skapades en administrativ del för orsaker som är specifika för avvikelsetypen.

The image shows two sections of a web application interface. The top section, titled 'Havainnon tiedot:', contains a form for reporting an incident. It includes fields for incident type (radio buttons for Turvallisuushavainto, Laatuhavainto, Ympäristöhavainto, Tietoturvahavainto), cause (Syy tapahtumalle: Järjestelmä), reporter name (Ilmoittajan nimi), email (Sähköposti), quality incident type (Laatuhavainnon tyyppi: Reklamaatio), department (Palkka: Sisäiset palvelut), date (Tapahtumapäivä: 18.12.2015) and time (Kellonaika: 13:30), title (Otsikko: Testitapaus), case number (Tapausnumero), description (Kuvaus: testikuva ei tule näkyviin järjestelmässä), and handling status (Käsittelyn tila: Avoin). A 'Tallenna' button is at the bottom. A screenshot of a code editor is visible in the background. The bottom section, titled 'Riskin arviointi:', contains a form for risk assessment. It includes fields for incident details (Tapahtumaa edeltävät hetket:), actions (Toimenpiteet: Korjattu koodinpätkä), responsible person (Vastuuhenkilö:), probability (Todennäköisyys riskille: Epätodennäköinen), risk level (Riskiluokka: Merkityksetön riski), cost estimate (Kustannusarvio: 0), and actual cost (Toteutunut kustannus: 0). A checkbox for 'Läheta selvityspyyntö:' is also present. A 'Tallenna' button is at the bottom.

Figur 10. Färdiga avvikelshanterings sidan

3.3.3 Administration

Innan den grafiska administrationsdelen programmerades skapades all ny information som nya användare eller användarrättigheter direkt in i databasen via phpMyAdmin. För att effektivt kunna upprätthålla informationen behövdes det ett grafisk användargränssnitt inne i systemet. På grund av att en grafisk administrativdel för avvikelsetyper och olika organisationsställen redan var gjord, användes den som grunden till att skapa de delar som fattades. Enlight Gurock lönar det sig att använda beprövad kod för att minska mängden av buggar (Gurock 2007), och den administrativa delen var testad i en tidigare fas i utvecklingen.

En punkt som skulle tas i beaktande var användarskapande var på databassidan är att, man skall aldrig spara användarens lösenord som normal, läsbar text, utan det skall alltid sparas krypterat så att fastän någon skulle få tag på databasen, får de inte lösenordet direkt, utan bara det krypterade lösenordet. Som lösning användes det PHP:s eget hash sha256 funktion, med hjälp av det får man lösenordet sparad i databasen i ett oläsbar format. Till goda principer för lösenordshantering hör också att inte låta någon annan ändra på lösenordet, utan ifall man glömmer det, kan administratören skicka en återställnings e-post till användaren som byter ut själv lösenordet.

Käyttäjät Poista käyttäjä

Etunimi:
Jan

Sukunimi:
Leppänen

Käyttäjätunnus:
leppja

Sähköposti:
jan.leppanen@falck.fi

Lähetä uusi salasana Tallenna

Käyttäjällä oikeudet:

- Ympäristö
- Turvallisuus
- Käyttäjät
- Käyttäjryhmät
- Laatu
- Raportti
- Sietämätön
- Palvelukeskus
- Sietämätön

Poista

Lisäoikeudet:

Lisää

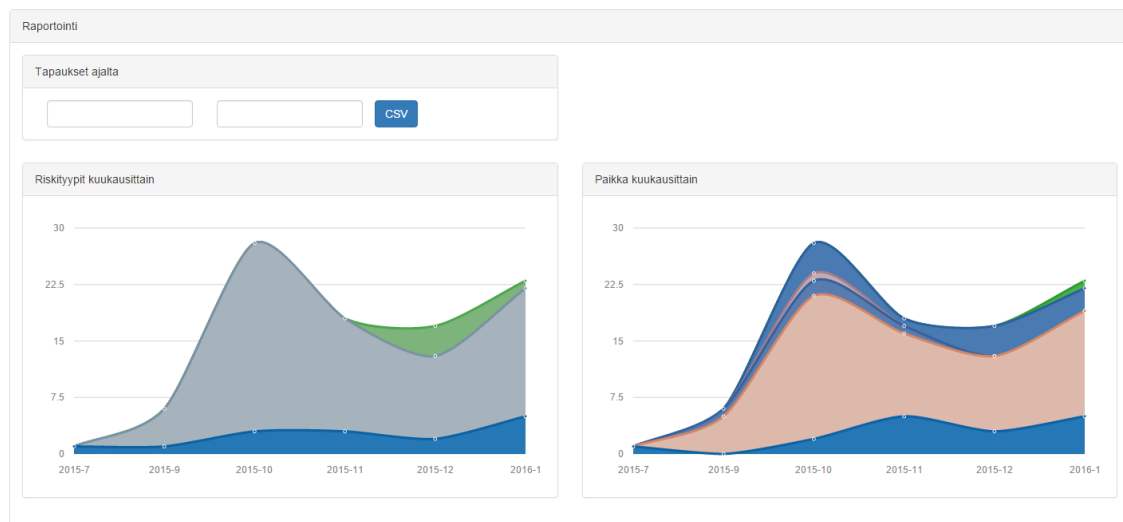
Figur 11. Administrativ, användarvy

3.3.4 Rapportering

Under utvecklingen var det osäkert hurdan information som behövs rapporteras. På grund av det blev rapporteringsdelen enkel med en visuell graf för mängden avvikelser per risktyp samt en graf för mängden avvikelser per organisationsställe (se Figur 12). Det är

också möjligt att ta ut en CSV fil, som innehåller alla avvikelser och informationen om dem.

Raportointi:



Figur 12. Avvikelserapportering

3.4 Användarvänlighet

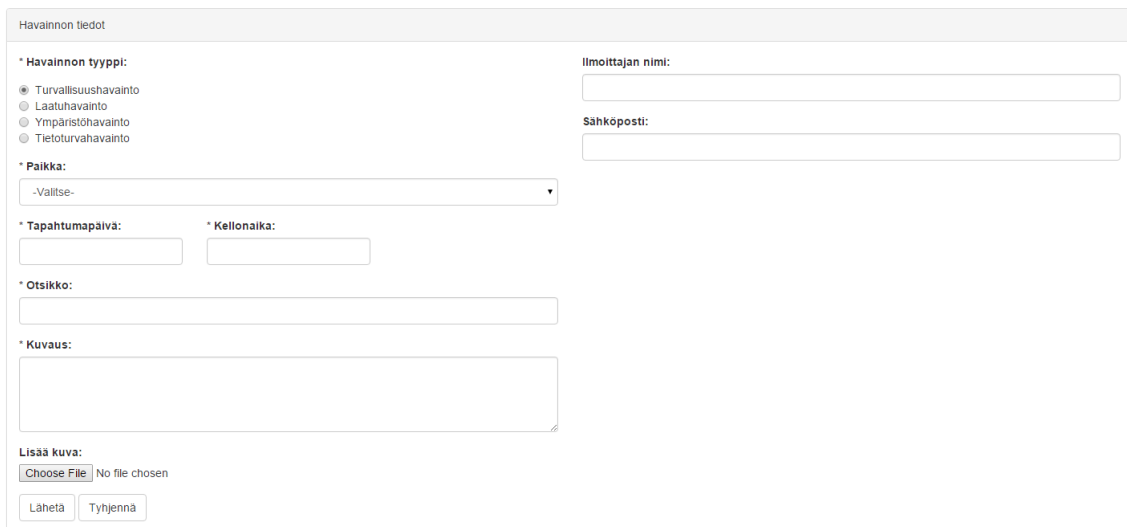
Enligt de användare som använde det gamla systemet, var användarvänligheten en av de största orsakerna till att man undvek att använda det. Vad är då användarvänligt? Nielsen påpekar att användarvänlighet kan definieras i fem punkter (Nielsen 2012)

- Hur lätt det är att lära sig använda systemet
- Hur effektivt är det att jobba med systemet
- Hur lätt minns man hur systemet fungerade efter en tid när man inte använt den
- Hur många fel gör man och hur allvarliga är de
- Hur trevlig är designen på systemet

Nya systemet utvecklades med tanke på dessa punkter. Genom att använda bootstrap får man sidan att vara responsiv och se modern ut, vilket gör att det går smidigt och är trevligt att jobba med systemet på en mobil apparat.

3.4.1 Anmälan

Det första steget med ett fungerande riskhanteringssystem är att användarna anmäler när det händer avvikelser, vilket betyder att anmälningsenkäten skall vara lätt att använda och det skall vara lätt att minnas hur det fungerar. Då skall det inte finnas några överloppspunkter som man måste fylla i när man anmäler saker. Som exempel, när man väljer säkerhetsanmälan finns det bara det väsentliga och den information man vill följa eller rapportera (se Figur 13).



The screenshot shows a web form titled "Havainnon tiedot" (Observation details). It contains several input fields and a list of radio buttons. The form is organized into two columns. The left column contains: a radio button group for "Havainnon tyyppi" (Type of observation) with options "Turvallisuushavainto" (selected), "Laatuhavainto", "Ympäristöhavainto", and "Tietoturvahavainto"; a "Paikka:" (Location) dropdown menu; "Tapahtumapäivä:" (Date) and "Kellonaika:" (Time) input fields; an "Otsikko:" (Title) text field; and a "Kuvaus:" (Description) text area. The right column contains: an "Ilmoittajan nimi:" (Reporter name) text field; and a "Sähköposti:" (Email) text field. At the bottom, there is a "Lisää kuva:" (Add image) section with a "Choose File" button and "No file chosen" text, and two buttons: "Lähetä" (Send) and "Tyhjennä" (Clear).

Figur 13. Anmälningsenkät, säkerhetsanmälan

Medan när fallet handlar om en kvalitetsavvikelse, finns det fler punkter som är gömda då när det inte behövs fyllas (se Figur 14).

Figur 14. Anmälningsenkät, kvalitetsanmälan

Med att visa bara det väsentliga och gömma resten, fyller man inte in fel eller onödig information. Då blir upplevelsen och effektiviteten med systemet bättre och man har mindre att komma ihåg till nästa gång när man fyller i enkäten.

3.4.2 Hantering

Hanteringsprocessen börjar med att användaren som hör till en viss användargrupp, som sköter till exempel kvalitetsavvikelser, får meddelande per e-post att det har kommit en ny kvalitetsavvikelseanmälan. I det tidigare systemet fick alla användare meddelande, oberoende av om anmälan tillhörde deras ansvarsområde. Med att undvika att skicka onödig e-post till användaren, minskar man jobb som hanteraren måste göra, då man inte måste öppna alla fall och se om det hör till sitt eget ansvarsområde eller inte.

Såsom i anmälningsdelen, är det speciellt viktigt att bara relevant information skulle behöva fyllas i, och alla andra fält skulle gömmas. Liknande funktion implementerades till avvikelsehanteringsdelen som i Figur 13 och Figur 14, för att visa bara den informationen som skall fyllas i.

Ett av det gamla systemets problem var att ifall man ville nämna någon person som ansvarig för att sköta fallet, måste man skapa en uppgift för åtgärder som måste göras, men ifall personen som nämnt ansvarig var fel, måste den personen först ”slutföra” åtgärdsuppgiften som var tillsatt, efter det måste man skapa en ny åtgärdsuppgift och nämna en ny ansvarsperson. I det nya systemet kan man ändra på ansvarspersonen utan att behöva skapa skilda uppgifter, och bara lämna en kommentar om varför man ändrat på ansvarspersonen.

3.5 Testning

Testningen gjordes för det mesta under utvecklingsfasen, eftersom utvecklingen gjordes med en iterativ utvecklingsmodell. Vilket betyder att efter varje ny iteration testades all ny kod och nya egenskaper. Under utvecklingen användes det tre stycken personer som testade, riskhanteringskoordinatören som var dedikerad testare som testade alla versioner och den ansvariga personen för administreringen för det färdiga systemet. De två andra testarna var mer ansvariga över att testa funktionaliteter som till exempel förklaringsenkäten som skickas då man behöver mera information om vad som hänt.

När utvecklingen var klar, gjordes det testning av alla användare som är ansvariga över något organisationsområde, och har hanteringsrättigheter i systemet. Då testades det att begränsningen för användarna fungerade rätt och rätta personer fick meddelande om nya avvikelsetestningar.

Under den sista testningsfasen när utvecklingen var klar, upptäcktes det problem med funktionaliteten att nämna någon person som ansvarig över ett specifikt fall, då personen inte har användarkonto och kunde inte komma åt fallet. Det ändrades till att istället för att få en länk per e-post direkt till fallet, får man en länk till en inloggningssida som bara kräver e-posten som är satt som ansvarsperson, och då kommer man åt fallet.

3.6 Införandet av systemet

Istället för att använda den databas som använts under utvecklings fasen, skapades det en tom kopia av databasen, som skulle fungera som verkligdatabas för systemet. På det sättet säkerställs det att ingen felaktig information kommer med i den riktiga databasen. Med att ha en tom databas gör man också den sista testningen för administrativa delen att skapa avvikelsetyper, organisationsställen, användare m.m.

Riskhanteringssystemet togs i bruk den 29.10, två månader innan deadlinen. En vecka innan det ordnades det utbildning för de personer som använder hanteringsdelen. De som gjorde den sista testnings fasen. Den 2.10 hölls det utbildning för resten av personalen om anmälningsdelen och med samma visades principerna för hanteringsdelen.

4 SYSTEMBESKRIVNING

4.1 Gränssnitt

Processen börjar med avvikelseenkäten (se Figur 13), där man fyller i typen av avvikelse, som exempel säkerhetsavvikelse, inom vilken organisationsställe har det hänt, datum och klocktid. Sedan fyller man en beskrivning vad som har hänt och har möjlighet att bifoga en bild, med en mobilapparat kan man fotografera, då blir fotot bifogat automatiskt. Anmälaren har också möjlighet att lämna in anmälan anonymt eller fylla i namnet och e-post adressen. Ifall man har fyllt i e-post adressen, får man bekräftelse då fallet har blivit behandlat.

När anmälan skickas, skrivs den in i databasen med status ”öppen”, efter det skickas det e-post till alla användare som har rättigheter till den typen av avvikelser (se Figur 15), e-posten innehåller en länk till det nya fallet.

```

$stmtement = $db->prepare("SELECT DISTINCT users.usr_Email FROM users left join userRights on users.usr_id=userRights.usr_id
WHERE userGroup=:incidentLocation OR userGroup=:incidentType;");
$stmtement->bindParam(':incidentLocation', $incidentLocation);
$stmtement->bindParam(':incidentType', $incidentType);
$stmtement->execute();
$sto="";
$header="Uusi tapaus riskihallinnassa";
while($row=$stmtement->fetch())
{
    $sto=$sto.$row['email'].", ";
}
    sendMail($sto, $message, $header);

```

Figur 15. E-post funktion om nya avvikelser

När hanteraren får meddelande om nytt fall, kan hen antingen öppna fallet via länken i e-posten, eller via hanteringsstartssidan (se Figur 7). För att kunna hantera fall måste man logga in i systemet, vid inloggningen skapas en PHP session. Sessionen innehåller användarinformation såsom användarrättigheter. Då man öppnar ett fall, granskas det att användaren har rätta rättigheter för att hantera fallet (se Figur 16), ifall användaren inte har rättigheter till fallet, flyttas hen till hanteringsstartssidan, om användaren inte är inloggad flyttas hen till inloggningssidan.

```

if(isset($_SESSION['username'])) {
    if (!validateGroup("Riski") && !validateGroup("Paiikka")) {
        if (!validateAttribute($caseType) && !validateAttribute($caseLocationUnEdited) && !validateIncidentHandler($caseResponsible)) {
            moveToPage("etusivu.php");
        }
    }
} else {
    moveToPage("login.php");
}

```

Figur 16. Användarrättighetskontroll

Den första sektionen på hanteringssidan är information om fallet (se Figur 10), som innehåller informationen som anmälaren har fyllt i, samt fallets status och en översiktlig orsak till händelsen, som företaget har identifierat som sannolik orsak för avvikelsetypen. Med en hjälp av en översiktlig orsak är det lättare att identifiera problem som exempelvis brist på utbildning eller brist på verktyg.

Andra sektionen är evaluering av fallet (se Figur 10). Sektionen innehåller information om vad som hände innan fallet, åtgärder som skall göras, personen som är ansvarig för att utföra åtgärderna, sannolikheten för risken, riskklass, kostnadsestimering, verkliga kostnader och en kryssruta för att skicka e-post till användaren om tilläggsinformation,

ifall kryssrutan är ikryssad syns det ett fält för användarens e-postadress. Kostnadsestimering och verkliga kostnader fälten används för att beräkna kostnader som uppstått på grund av fallet. Då kostnaderna överstiger 500€ eller riskklassen är måttlig eller högre, blir kryssrutan för att skicka meddelande till användaren automatiskt ikryssad. Ifall man i det skedet inte vill skicka meddelande till användaren, måste man fylla i en förklaring varför man inte skickar meddelandet.

Företaget har definierat tre sannolikhetsklasser samt fem riskklasser, sannolikhetsklasserna är, osannolikt (grön), möjlig (gul) och sannolik (röd), riskklasserna är meningslös (grön), dräglig (gul), måttlig (orange), betydelsefull (röd) och odräglig (mörkröd). Med att kombinera de två fälten får man statistik om vilken sorts avvikelser det händer. Då fallet värderas odrägligt, skickar systemet information om fallet till alla inom företagets styrelse, med en liknande funktion som i Figur 15.

Då evalueringen är gjord, men åtgärdsuppgifterna pågår eller man väntar på någon information (annan en från användaren) ändrar man statusen på fallet till ”behandling pågår”. Ifall man skickar meddelande till användaren, ändras status automatiskt till ”väntar på användarens svar”, då användaren har svarat på enkäten, ändras statusen till ”behandling pågår” automatiskt. Då evalueringen är klar och åtgärderna är gjorda, och ingen mer information väntas i fallet, ändrar man fallets status till behandlad.

Tredje sektionen är tilläggsinformation som användaren har fyllt i, ifall man har skickat meddelande om det. Sektionens fält är samma som i Figur 9, då man inte skickar begäran om mer information, blir denna sektion tom.

Fjärde sektionen är för kommentarer som hanterare kan lämna för fallet (se Figur 17).

Kommentit		
Kasittelija	Päivämäärä	Kommentti
Mira Matikainen	07.10.2015 08:46	Uusi televisio paikallaan
Kimmo Ukkonen	02.10.2015 09:55	Uusi telkkari näyttää tulleen paikalle. Odotetaan silti Jonilta lopullista kustannusta.

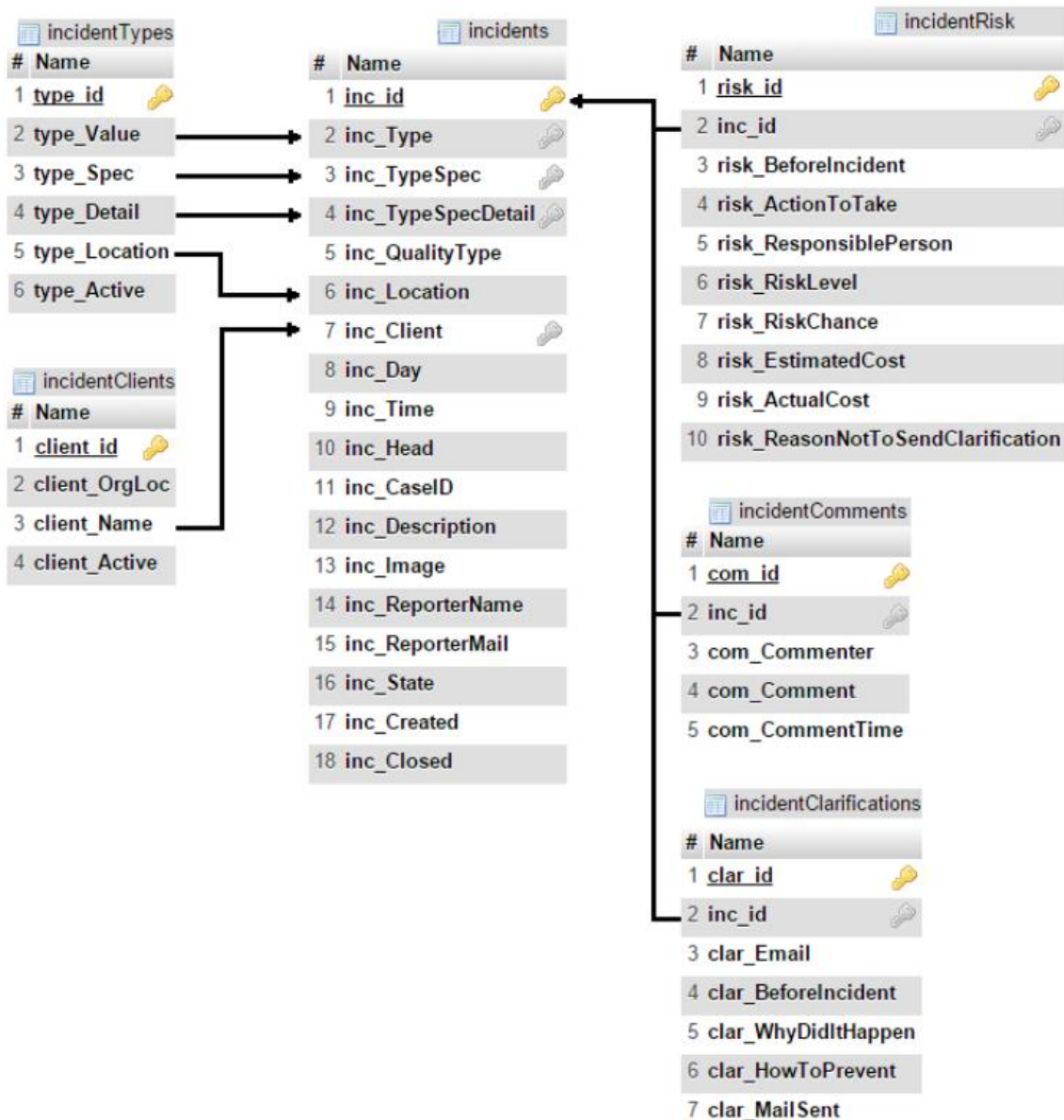
Figur 17. Kommentarsektion

4.2 Datastruktur

Avvikelserna är delade i fyra tabeller och administrativa sidan för avvikelse typer och klienter är delade i två tabeller i databasen. Då har varje delmoment en egen tabell, enligt de goda databashanteringsmetoderna (Microsoft 2010) (se Figur 18).

De administrativa tabellerna är `incidentTypes` och `incidentClients`. `IncidentTypes` innehåller information om avvikelse typer så som riskavvikelse eller kvalitetsavvikelse, organisationsställen inom företaget så som call center, interna tjänster eller franchise nätverket. `IncidentClients` innehåller information om klienter som finns och till vilket organisationsställe de hör till.

Avvikelse tabellerna är `incidents`, `incidentRisk`, `incidentComments` och `incidentClarification`, tabellerna motsvarar de olika segment som finns i avvikelsehanteringsvyn. `Incidents` tabellen innehåller all den information som anmälaren fyller i samt fält som ifylls i hanteringskedet, information om fallets status, när fallet har skapats och när fallet är hanterat och blivit stängt. `IncidentRisk` tabellen innehåller information allvarlighet och kostnader som uppstått på grund av avvikelsen. `IncidentComments` innehåller kommentarer som hanterare kan lämna till fall. `IncidentClarifications` innehåller information som lämnas in ifall man har begärt användaren fylla information om fallet.



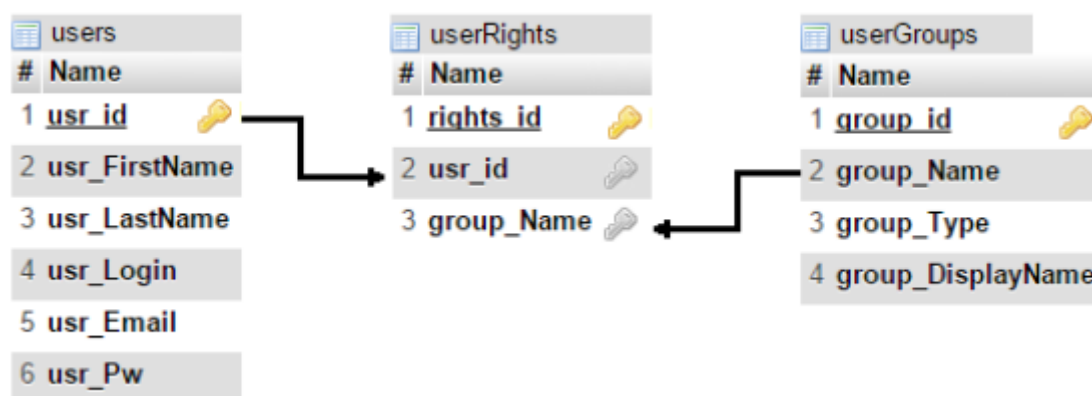
Figur 18. Databasstruktur, avvikelsetabeller

Användarinformationen är delad i tre tabeller i databasen, users, userGroups och userRights (se Figur 19). Users tabellen innehåller information om själva användaren såsom namn, epost och lösenord i krypterad form.

UserGroups tabellen innehåller de olika användargrupper som finns i systemet, då man skapar en ny avvikelsetyp eller ett nytt organisationsställe, skapas det automatiskt en ny användargrupp i tabellen. Det är också möjligt att skapa användargrupper via en administrativ vy i systemet, där man kan skapa administrativa användargrupper, som

exempel då man skapar en administrativ vy för att upprätthålla klienter, lägger man rättighetskrav till sidan och sedan skapar man en användargrupp till den.

Sista tabellen, userRights innehåller alla användarens rättigheter. Denna tabell används vid inloggning, då man skapar användarrättighetsvariabeln.



Figur 19. Databasstruktur, användartabeller

5 EVALUERING AV DET NYA RISKHANTERINGSSYSTEMET

5.1 Genomgång av krav

Enligt arbetarskyddslagen måste arbetsgivaren bedöma riskerna som finns på arbetsplatsen samt dokumentera olägenheter och göra åtgärder för att undvika fallet i framtiden. Krav för kvalitetshanteringsstandardens del var att vara möjligt att kunna följa med avvikelser samt analysera händelserna. Avvikelseanmälan tillsammans med den administrativa delen tillåter arbetsgivaren att beakta faror som finns på arbetsplatsen, i hanteringsdelen följer man med att man gör de åtgärder för att undvika olyckor i framtiden av samma orsak. Då när all information är dokumenterad i elektronisk format i databasen, fyller den arbetarskyddslagens krav, och med den rapporteringsdelen gör man analyseringen och rapporteringen lätt med tanke på kvalitetsstandarderna.

En av de största kraven var att användargränssnittet skulle vara användarvänligt och bekvämt att använda. Utvecklingen av systemet baserade sig långt på den punkten, som exempel var det fokus på att användaren inte skulle behöva fundera på vad som måste

fills i, utan beroende på avvikelseypen och organisationsställen visades bara de fält som faktiskt skulle fyllas. Redan under testningsfasen fick användargränssnittet mycket positiv feedback av användarna.

Avvikelseanmälningen skulle vara öppen för alla utan att behöva registrera sig, och möjlighet att kunna lämna in anonymt. Enkäten måste också vara mobilvänlig på grund av att en stor del av personalen inte jobbar på kontoret. Anmälaren måste också kunna välja vilken avvikelseyp anmälan är. Systemet är en webbaserad möjliggör det att alla har tillgång att göra anmälningar. Eftersom det i utvecklingen Bootstrap som ramverk är sidan mobilvänlig, och gör att till exempel att bifoga foton med i anmälan är möjligt med att fotografera direkt med telefonens kamera, och bilden läggs automatiskt med i anmälan. Med den administrativa delen som tillåter administratören att lägga till avvikelseyper i databasen, får man effektivitet med att visa de aktiva typerna i anmälningsdelen.

Vid hantering av användarkonton var det önskat att det skulle vara möjligt att upprätthålla dem inom företaget, utan att behöva beställa nya användare. I tidigare systemet fanns det inte möjlighet att skapa användare och det fanns inga användarrättigheter, utan alla användare hade rättigheter till alla anmälningar. Det ville man undvika i nya systemet. I databasen skapades det två tabeller för att hantera information om användare, en för själva användare som innehåller information som namn, e-post och lösenord i krypterad form, en annan tabell för själva användarrättigheterna. Användargrupperna skapas automatiskt enligt de avvikelseyper och organisationsställen som det finns, efter att man skapat dem kan man lägga till rättigheter till användaren. När man öppnar anmälan, granskar systemet rättigheterna som användaren har, ifall användaren inte har rätta rättigheter flyttas användaren till startsidan.

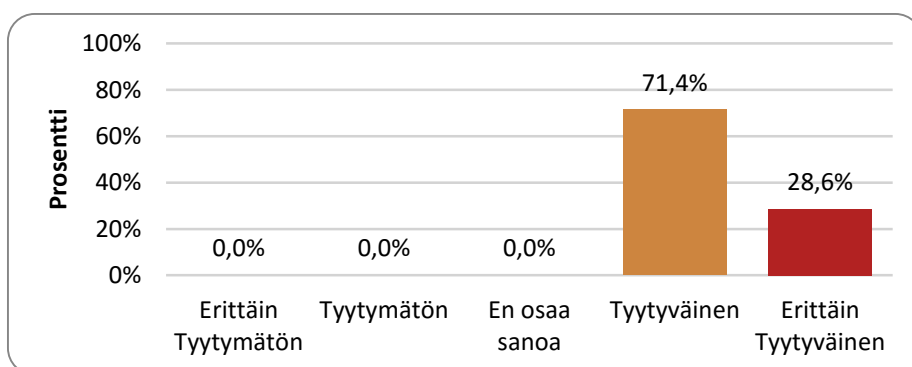
5.2 Användarenkät

Enkäten är gjord av företaget för att värdera det nya systemets framgång enligt de förväntningar som företaget ställt. I enkäten deltar det sju personer som har använt det gamla och det nya systemet som hanterare. I enkäten finns det tio frågor, fyra stycken

angående användbarhet, tre stycken angående utseendet och tre öppna frågor om systemet. Enkäten är skapat med Questbacks, Digium Enterprise enkätssystem.

Första frågan är om hur skulle man evaluera användbarheten för det nya jämfört med det gamla systemet, skalan är från 1-5, där 1 är mycket missnöjd och 5 är nöjd.

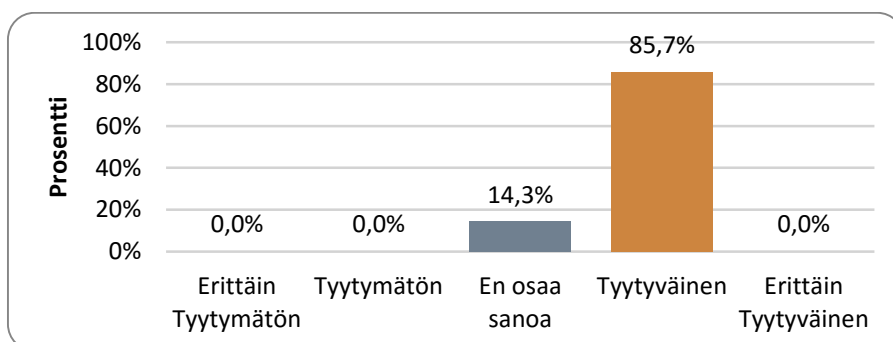
Resultatet är att ca 71% är nöjda och ca 28% är väldigt nöjda med ett svarsmedeltal på 4,29.



Tabell 1. Enkätfråga 1, resultat

Medeltal	Standardavvikelse	Median
4,29	0,45	4,00

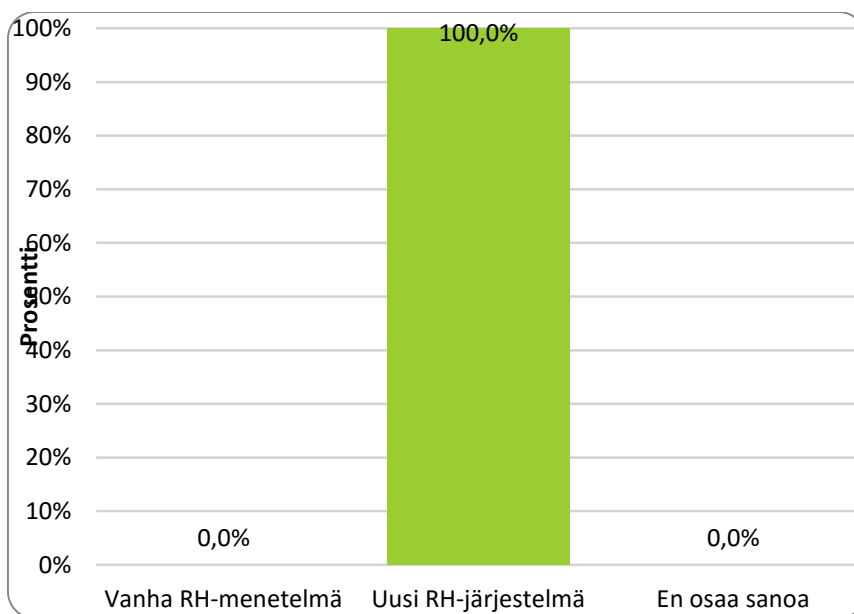
Andra frågan är ”Har tröskeln för att göra avvikelsetjänsten minskat, med implementeringen av det nya systemet”, med en skala 1-5.



Tabell 2. Enkätfråga 2, resultat

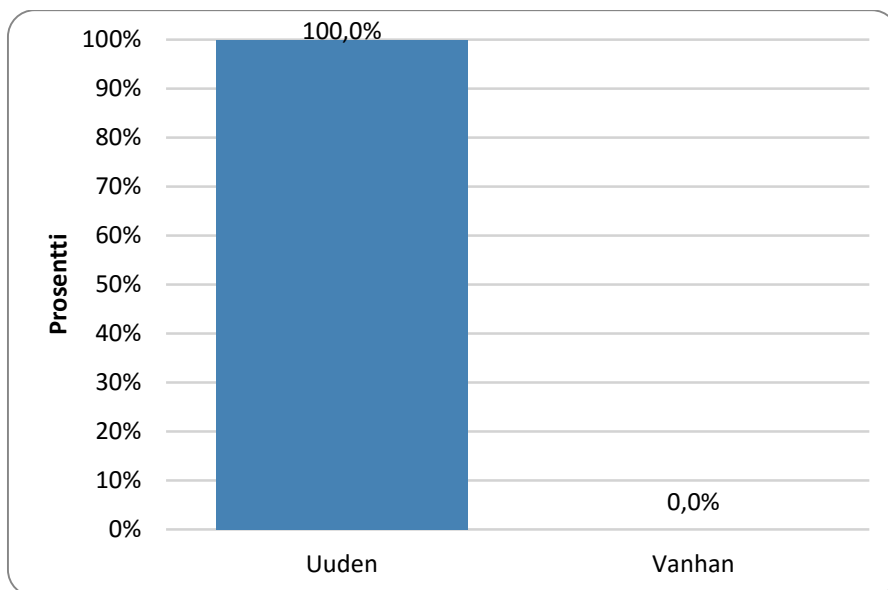
Medeltal	Standardavvikelse	Median
3,86	0,35	4,00

Tredje frågan är ”Hur skulle du estimerar användbarheten mellan systemen, vilkendera av dem är snabbare att använda” med tre alternativ, gamla, nya eller kan inte säga. 100% av användaren tycker användbarheten med nya systemet är bättre.



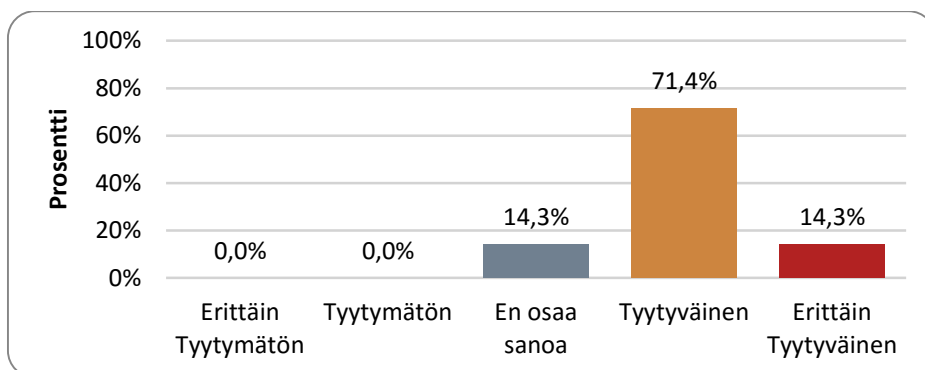
Tabell 3. Enkätfråga 3, resultat

Fjärde frågan ”Hurdan feedback har ni fått från era egna underordnade, har de tyckt mera om det nya eller gamla systemet?” med två alternativ, gamla och nya. Resultatet är att 100% av användarna har tyckt om det nya systemet mer än det gamla.



Tabell 4. Enkätfråga 4, resultat

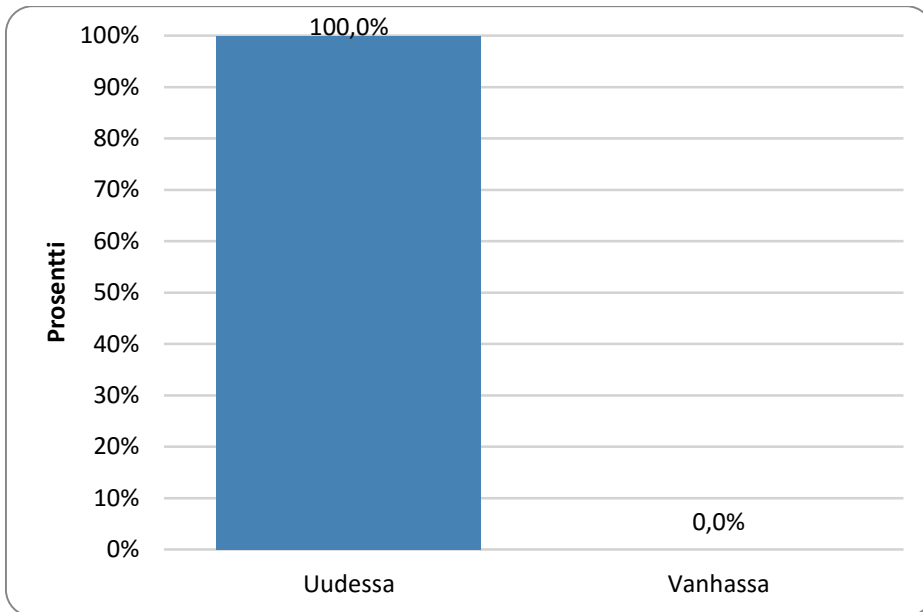
Femte frågan handlar om utseendet, ”Hur nöjd är du med systemets utseende”, med en skala 1-5, där 1 är väldigt missnöjd, 3 kan inte säga och 5 väldigt nöjd. Inga användare har varit missnöjda med utseendet, och största delen varit nöjda eller mycket nöjda.



Tabell 5. Enkätfråga 5, resultat

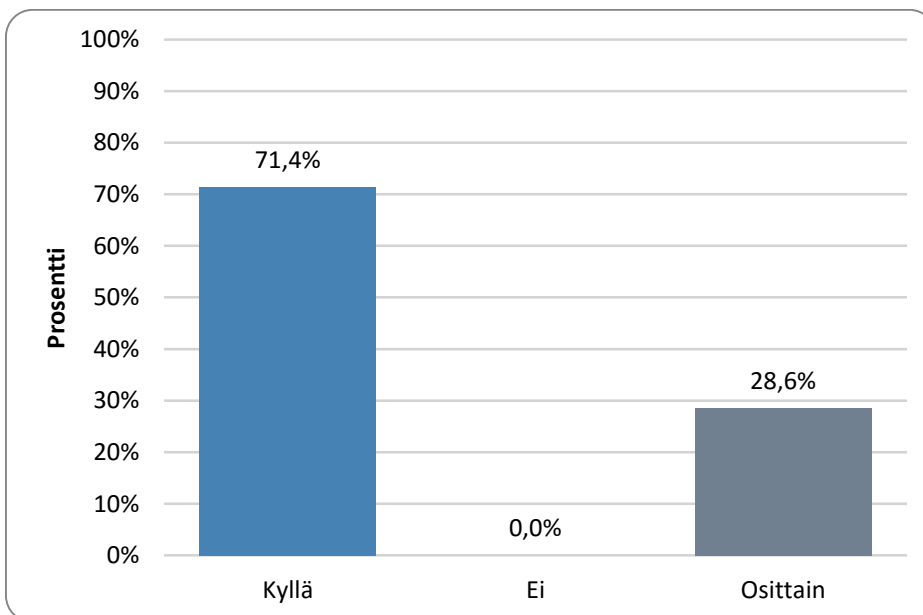
Medeltal	Standardavvikelse	Median
4,00	0,53	4,00

Sjätte frågan, ”När man jämför gamla och nya systemets fält, i vilkendera frågas det mer väsentliga uppgifter”. Frågan har två alternativ, nya systemet eller det gamla. 100% av användarna tycker mer om det nya systemet än det gamla, vilket också var målet för systemet.



Tabell 6. Enkätfråga 6, resultat

Sjunde frågan, ”Är hanteringssidan informativ och är utseendet trevligt”, med tre alternativ, ja, nej och delvis. Ca 71% av användarna tycker ja och ca 29% tycker delvis.



Tabell 7. Enkätfråga 7, resultat

De tre sista frågorna var öppna frågor. Fråga åtta ”Har du upptäckt brister i systemet”. Som svar kom det önskemål om att få ett meddelande när man har skickat e-post till användaren om mer information, samt e-post meddelande då när användaren har svarat på enkäten.

Kun tapaus lähetetään työntekijälle (ja odotamme työntekijältä vastausta), kaipaisin kahta asiaa:

- 1) ”Tapaus lähetetty ongelmanratkaisija@falck.fi-henkilölle”-tyyppistä infoa.
- 2) Ilmoitusta kun edellään mainittu henkilö on käynyt täyttämässä omalta osaltaan RH-menetelmän.

Olisi myös mielenkiintoista tietää, saisiko RH-järjestelmällä poistettua excel-tarpeen esimerkiksi tappioita koskien.

Eiköpä ne oo kehitysideoissa jo. ;)

En varsinaisia puutteita. Raportointi voisi olla vielä monipuolisempaa, mutta toisaalta e kehittyy koko ajan.

Selvityspyynnön lähettämässä henkilölle olisi hyvä tulla yksinkertainen viesti: ”selvityspyntö lähetetty”, jotta välttyään sähköpostien spämmäämiseltä.

Olisi myös toivottavaa saada sähköposti kun työntekijä on käynyt täyttämässä järjestelmään selvityspyynnön.

Figur 20. Enkätfråga 8, resultat

Fråga nio, ”Uppskattar du systemet vara nyttig”. Svaren var väldigt positiva och det påpekas att det hjälper med rapportering till klienter.

Järjestelmä on hyödyllinen. On tärkeää saada tämä data kootusti talteen ja koen järjestelmän helpokäyttöiseksi.

Kyllä, kun sen saa vielä kunnolla käyttöön ja raportoinnin yms muut sinne.

Ehdottomasti kyllä. Tätä kautta on hyvä parantaa toimintaa ja järjestelmä on myös loitava tuki tulevalle laatujärjestelmälle.

Järjestelmä on hyödyllinen, koska kaikki poikkeamat löytyvät samasta paikasta, ja se helpottaa myös osaltaan raportointia toimeksiantajille.

Figur 21. Enkätfråga 9, resultat

Fråga tio, ”Allmän feedback om systemet”. Feedbacken var bra och utvecklingen av systemet fungerade bra.

Hyvä järjestelmä ja potentiaalia tehhä paaaaaljon vielä parempi,

Järjestelmän kehitys on nopeaa ja siitä kehittyi koko ajan monipuolisempi työväline. Tämä on ollut hyvä investointi.

Miellyttävä ulkoasu ja helposti modattavissa sitä mukaan kun toteutettavia kehitysideoita tulee. Hyvä Janne! :)

Figur 22. Enkätfråga 10, resultat

6 SLUTSATSER OCH DISKUSSION

Målsättningen med arbetet var att skapa ett riskhanteringssystem för Falck förrän årsskiftet 2015 – 2016. Systemet börjades utveckla i juli, och systemet togs i bruk i slutet av oktober. Utvecklingen gick smidigt och företaget är nöjd med slutresultatet, som man kan se av enkäten. Alla krav samt önskemål som företaget hade uppfylldes. Det har kommit upp fler vidareutvecklingsidéer, som ett litet exempel, att styrelsen skulle ha möjlighet att se alla avvikelser, men inte få e-post meddelande av alla fall, utan bara de som de är ansvariga över. Vidareutvecklingen för tilläggssegenskaper har fortsatt sedan systemet infördes.

Själva projektet var väldigt intressant och jag lärde mig mycket om systemutveckling. Det jag mest lade märke till var hur bra och trevligt det är att jobba med en iterativ utvecklingsmodell. Ifall det uppstod några problem med koden, kunde man korrigera felet snabbt och kunde fortsätta utvecklingen med en fungerande kod. Med att jobba iterativt kunde man göra ändringar på designen och innehållet ifall det man planerat inte fungerade så bra man tänkt. Personligen är jag nöjd med resultatet och sättet utvecklingen utfördes, valet av programmeringsspråket och ramverket fungerade bra för detta ändamål.

KÄLLOR

Bootstrap. 2016. Tillgänglig: www.getbootstrap.com

Hämtad 14.2.2016

DB-Engines. 2016, *DB-Engines Ranking*. Tillgänglig: <http://db-engines.com/en/ranking>

Hämtad 14.2.2016

Finlex. 2002, *Arbetarskyddslag*. Tillgänglig:

<http://www.finlex.fi/sv/laki/ajantasa/2002/20020738> Hämtad 10.1.2016

Forsyth, Aaron. 2013, *Waterfall vs Agile Models in Software Development*. Tillgänglig:

<http://www.myglobalit.com/blog/waterfall-vs-agile-models-software-development>

Hämtad 27.1.2016

Gurock, Dennis. 2007, *12 Practical Tips for Building Bug-Free Software*. Tillgänglig

<https://blog.gurock.com/12-practical-tips-for-building-bug-free-software> Hämtad

27.1.2016

Guzel, Burak. 2011, *Top 15+ Best Practices for Writing Super Readable Code*.

Tillgänglig <http://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118>

Hämtad 30.1.2016

Hall, Steve. 2014, *MySQL vs Mongo DB*. Tillgänglig:

<https://www.scriptrock.com/articles/mysql-vs-mongodb> Hämtad 24.1.2016

Kampf, Eran. 2008, *How Do You Define "Good Code"?*. Tillgänglig:

<http://www.developerzen.com/2008/06/26/how-do-you-define-quotgood-codequot>

Hämtad 30.1.2016

Kaplan, Moshe. 2014, *When to Use MongoDB Rather than MySQL (or Other RDBMS): The Billing Example*. Tillgänglig: <https://dzone.com/articles/when-use-mongodb-rather-mysql>. Hämtad 24.1.2016

MEAN.io. 2016. Tillgänglig: www.mean.io
Hämtad 14.2.2016

Microsoft. 2010, *Database design basics*. Tillgänglig: <https://support.office.com/en-us/article/Database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5> Hämtad 1.2.2016

Nielsen, Jakob. 2012, *Usability 101: Introduction to Usability* Tillgänglig: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
Hämtad 24.1.2016

Perks, Mike. 2006, *Best practices for software development projects*. Tillgänglig: http://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html Hämtad 27.1.2016

Rouse, Margaret. 2011, *Iterative development definition*. Tillgänglig: <http://searchsoftwarequality.techtarget.com/definition/iterative-development>
Hämtad 27.1.2016

Sitepoint. 2014, *Best PHP IDE in 2014*. Tillgänglig: <http://www.sitepoint.com/best-php-ide-2014-survey-results/> Hämtad 5.2.2016

Stackoverflow.com. 2016. Tillgänglig: <http://stackoverflow.com>
Hämtad 14.2.2016

Startbootstrap.com. 2014, *SB Admin 2*. Tillgänglig: <http://startbootstrap.com/template-overviews/sb-admin-2/> Hämtad 5.8.2015

Suomen Standardisoimisliitto SFS, 2015
Quality management systems – Requirements (ISO 9001:2015)

Vaadin. 2016. Tillgänglig www.vaadin.com

Hämtad 14.2.2016

Verisonone. 2015, *Test-First Programming*. Tillgänglig: <https://www.verisonone.com/agile-101/agile-software-programming-best-practices/test-first-programming/> Hämtad 27.1.2016

Wurzer, Erik. 2012, *Why you Should be using PHP's PDO for Database Access*.

Tillgänglig: <http://code.tutsplus.com/tutorials/why-you-should-be-using-phps-pdo-for-database-access--net-12059> Hämtad 1.2.2016

BILAGOR

Käytettävyys:

1. Kuinka arvioisit uuden RH-menetelmän käytettävyyttä suhteessa edelliseen järjestelmään? (asteikolla 1-5, 5=paljon helpompi käyttää, 1=paljon vaikeampi käyttää)
2. Onko kynnys ilmoitusten tekemiseen ja RH-menetelmän käyttöön alentunut uuden järjestelmän myötä? (1-5)
3. Miten arvioisit järjestelmän käytön sujuvuutta suhteessa edelliseen järjestelmään, kumpaa on nopeampi käyttää? (2 vaihtoehtoa)
4. Minkälaista palautetta olet saanut omilta alaisiltasi - ovatko he pitäneet enemmän uuden vaiko vanhan järjestelmän käyttämisestä? (2 vaihtoehtoa)

Ulkoasu:

5. Miten tyytyväinen olet järjestelmän ulkoasuun? (asteikko 1-5)
6. Kun vertaat uuden RH-menelämän kenttiä suhteessa edelliseen järjestelmään, kummassa järjestelmässä kysytään tarkoituksenmukaisempia tietoja?
7. Onko tapausten käsittelysivu informatiivinen ja mielyttävän näköinen?

Yleinen: (avoimet vastaukset)

8. Oletko havainnut järjestelmässä puutteita?
9. Arvioitko järjestelmän olevan hyödyllinen?
10. Yleistä avointa palautetta järjestelmästä?