

# **Examensarbete**

## **Utveckling och implementering av webbapplikation vid Nautor**

Adrian Fellman

Examensarbete för ingenjörsexamen (YH)-examen  
Utbildningsprogrammet för produktionsekonomi  
Vasa 2016



## **EXAMENSARBETE**

Författare: Adrian Fellman

Utbildningsprogram och ort: Produktionsekonomi, Vasa

Handledare: Mikael Ehlers, Kjell Björklund

Titel: Utveckling och implementering av webbapplikation vid Nautor

---

Datum 22.03.2016

Sidantal 39

Bilagor

---

### **Abstrakt**

Detta examensarbete är gjort åt Oy Nautor Ab. Företaget har under en längre tid haft en önskan att ta i bruk ett fullständigt ERP-system. Men de anser att de inte är redo att ta det klivet i detta skede. Därför vill de ha en applikation som kan användas för att ta reda på vilka krav företaget kan tänkas ställa på ett ERP-system och för att förbereda företaget på användningen av ett centralt informationssystem.

Syftet med detta examens arbete och min uppgift har varit att för det första ta reda på vilka funktioner företaget är i behov av och vilken slags data de behöver ha tillgång till. Och för det andra att utveckla och implementera en applikation som uppfyller dessa krav.

Resultatet av detta arbete blev en skräddarsydd webbapplikation som företaget använder för insamling av data, övervakning, planering och rapportering.

---

Språk: Svenska

Nyckelord: Webbapplikation, ERP-system, utveckling, implementering

---

## **BACHELOR'S THESIS**

Author: Adrian Fellman

Degree Programme: Industrial Management

Supervisors: Mikael Ehres, Kjell Björklund

Title: Development and Implementation of Web Application at Nautor

---

Date 22.03.2016

Number of pages 39

Appendices

---

### **Summary**

This Bachelor's thesis was done on behalf of Oy Nautor Ab. The company has for some time been interested in implementing a full scale ERP-system, but they don't consider themselves ready to take the step at this time. Because of that they wanted an application that is easy to use and that could be used to find out what kind of functions the company might need from an ERP-system and to prepare the company to implement such a system.

The purpose of this Bachelor's thesis is firstly to find out what kind of functions the company might need and what kind of data is important. Secondly the purpose is to develop an application that meets these demands.

The result of this thesis was a customized web application that the company uses to gather data, supervise, plan and generate reports.

---

Language: Swedish

Key words: Web application, ERP system, implementation, development

---

## Innehållsförteckning

1 Inledning.....	1
1.2 Bakgrund.....	1
1.3 Syfte .....	2
1.4 Uppgift .....	2
1.5 Avgränsning .....	3
2 Presentation av företaget .....	4
2.1 Produkter .....	5
3 Teori .....	7
3.1 Enterprise Resource Planning System.....	7
3.2 Implementering av ett ERP-system .....	8
3.3 Engineering To Order.....	9
3.4 Data kvalitet .....	9
3.5 Komplexa processer och utveckling av mjukvara .....	10
4 Teknologier .....	12
4.1 Back-end.....	12
4.2 Front-end .....	13
4.3 Databas .....	14
5 Inledande arbete .....	15
5.1 Insamling av data.....	17
5.2 Manpower.....	18
5.3 Quality Log.....	21
5.4 Pool.....	22
5.5 POC .....	23
5.6 Utveckling av applikationen.....	23
5.7 Webbapplikation .....	23
6 Nautor – Planner.....	24
6.1 Manpower.....	25
6.2 Pool.....	28
6.3 Quality Log.....	29
6.4 Notes.....	30
6.5 Ibruktagande och skolning .....	31

7 Resultat.....	32
7.1 Diskussion.....	33
7.2 Förslag till fortsatt forskning.....	34
8 Slutord.....	35
Källförteckning.....	36

## Figurförteckning

<i>Figur 1 Swan 90S, Alberto Cocchi</i> .....	5
<i>Figur 2 Swan 80FD, Carlo Borlenghi</i> .....	6
<i>Figur 3 Excelsystem</i> .....	18
<i>Figur 4 Excelblad</i> .....	19
<i>Figur 5 Login</i> .....	25
<i>Figur 6 Manpower Log</i> .....	26
<i>Figur 7 Manpower Add New</i> .....	27
<i>Figur 8 Manpower visualisering</i> .....	27
<i>Figur 9 Pool</i> .....	28
<i>Figur 10 Quality Log</i> .....	29
<i>Figur 11 Avvikelser för ett projekt</i> .....	29
<i>Figur 12 Lägg till ny avvikelse</i> .....	30
<i>Figur 13 Profilsida med notes</i> .....	31

## 1 Inledning

Att implementera ett Enterprise Resource Planning (ERP) system kan ge ett företag eller en organisation stora fördelar genom att förbättra deras operativa effektivitet, men tyvärr är det inte helt riskfritt. Ett stort problem som många stöter på när de skall implementera ett ERP system är skillnaden mellan vad systemet kan erbjuda och vad företaget ställer för krav. Att designa ERP-system så att de passar för företag har alltid varit ett av de svåraste skedena under en implementering. Det är vanligt att företaget eller organisationen tvingas ändra hur de gör saker för att det skall passa ihop med det nya systemet. Om företaget inte anpassar sig riskerar det en riskfylld och dyr övergångsfas.

Att identifiera ett företags behov och krav på ett ERP-system är viktigt eftersom de två vanligaste orsakerna till ett potentiellt misslyckande under implementeringen är att företaget beställer fel funktioner och att systemets designer missuppfattar kraven som ställs. (Panayiotou & Gayialis & Evangelopoulos & Katimertzoglou, 2015, s. 629)

### 1.2 Bakgrund

Det här examensarbetet har gjorts åt Oy Nautor Ab som är beläget i Jakobstad. Företaget har under en längre tid haft en önskan att implementera ett Enterprise Resource Planning system för att bland annat underlätta planeringen och skapa en känsla av kontroll över företagets resurser. Hittills har företaget använt sig av flera olika datorprogram och en otalig mängd Excel-filer för att organisera företaget. Detta har lett till man tvingats pussla ihop data och ödsla tid på att söka efter relevant data i ett oorganiserat och okoordinerat filsystem.

Ledningen på företaget har visioner om vad som skulle kunna krävas av ett potentiellt ERP-system i framtiden och åsikter om vilka funktioner som skulle förenkla administreringen och planeringen. De vill gärna ha ett mycket visuellt system där användaren kan skapa sig en snabb överblick och sedan fokusera noggrannare på mera specifika uppgifter.

De är medvetna om att implementering av ett ERP-system är en invecklad, riskfylld och potentiellt dyr process, speciellt om den görs på fel sätt. De är också medvetna om ett av de större problemen när det gäller Nautor och ERP-system. Nautor är ett Engineering To Order

företag, vilket betyder att de genast från början ställer krav på ett ERP-system som väldigt få kan leverera utan skräddarsydda modifikationer.

### **1.3 Syfte**

Syftet med mitt examensarbete är att hjälpa företaget med att sänka tröskeln och göra implementeringen smärfriare genom att förbereda dem på användningen av ett ERP-system.

Detta görs genom att ta reda på vilka krav företaget har på ett ERP-system och vilken data de behöver ha tillgång till i det nya systemet. Första delen av arbetet har gått ut på att ta reda på vad företaget behöver. Och den andra delen har varit att utveckla och även implementera en applikation som uppfyller dessa krav.

### **1.4 Uppgift**

Min uppgift var att utveckla en applikation åt Oy Nautor Ab som kan hjälpa företaget att ta reda vilka typer av funktioner de vill ha i ett framtida ERP-system. De hade flera idéer som de redan funderat ut och uppmanade mig även att komma med egna förslag, eftersom de inte hade mycket erfarenhet inom området sedan tidigare.

Företaget ville att all information skulle finnas på ett och samma lättåtkomliga ställe. Eftersom de tidigare använt sig av Excel-filer i stor utsträckning ville de nu ha en mera organiserad uppbyggnad av informationen.

De hade sedan tidigare många olika datorprogram som skötte om en eller flera uppgifter. Till exempel hade de ett program för e-mail och bokning av mötesrum, ett annat program för kontroll av arbetssäkerhet, ett program för tidsuppföljning samt ett för kvalitetsuppföljning med mera. Ett önskemål var således att slippa detta i framtiden eftersom det försvårar planeringen när företaget skall jämföra data.

En viktig del av examensarbetet var att applikationen skulle kunna visa var alla anställda befinner sig ifall det skulle hända en olycka där någon av fabrikerna måste utrymmas, till



exempel om det blev en brand. Företaget ville då ha en funktion där de kan skriva ut en rapport över var alla befinner sig och som går att ge åt räddningsverket när de anländer.

Under mina studier på Yrkeshögskolan Novia och under den tid jag jobbat på Nautor och andra företag har jag insett vikten av att kunna planera och ha möjlighet att ta goda beslut. Något som är betydligt lättare om det finns tillgång till relevant och bra presenterad data.

## **1.5 Avgränsning**

Begränsningar måste sättas upp eftersom examensarbetet annars skulle ha kunnat bli hur stort som helst. Målet med projektet var inte att skapa en permanent lösning. Applikationen som gjordes behövde således inte utvecklas lika långt som en applikation som skall användas under många år.

En mycket stor del av Nautors verksamhet cirkulerar kring deras ritningar. Detta sker i ett enormt system där ingenjörerna använder sig av programvara som SolidWorks, Siemens NX och AutoCad. Flera av dessa program har egna mjukvaror som underlättar organiseringen av filer. Därför finns det inget behov för min applikation att i detta skede röra om i något så viktigt.

## 2 Presentation av företaget

Oy Nautor Ab är ett företag som är beläget i Jakobstad, Kållby och Kronoby. De bygger högklassiga lyxjakter och anses vara ledare inom segelbåtsproduktion.

Företaget grundades 1966 av Pekka Koskenkylä i Jakobstad, Finland. I början kunde de fokusera på serieproduktion av högklassiga segelbåtar. Viktigt för företaget och det som gav dem ett gott namn inom industrin var att de var kompromisslösa när det gällde design och kvalitet. I nuläget har de en årlig omsättning på ungefär 50 miljoner euro.

Den första båten som tillverkades av Nautor var 36-fot lång och med namnet Tarantella. Nu har företaget sålt över 2000 jakter som sträcker sig från 36-fot till lyxjakter på 131-fot. Mer än 95 procent av Nautors båtar seglar fortfarande på haven.

Nautor har nu ungefär 250 anställda på tre olika platser i Finland. I Kållby har företaget en fabrik där de tillverkar skroven och utför laminering av stora och små delar till båtarna. I Kållby utförs också monteringen av mindre jakter, som till exempel 53-fotare. Företaget har också möjlighet att testa båtarnas sjöduglighet inomhus under de kalla månaderna.

I Kronoby tillverkar företaget inredning till alla båtar vid sitt snickeri.

I Jakobstad finns Nautor BTC, Boat Technology Center som byggdes år 2002. Det är Nautors största fabrik och den inkluderar även huvudkontoret. Vid BTC utförs monteringen av båtarna och även sjösättningen och testning vid Nautors egna marina.

Sedan 1998 har företaget drivits under Italiensk ledning då Leonardo Ferragamo investerade i Oy Nautor Ab. Företaget blev då en del i Nautors Holding som även inkluderar Marina Management och det engelska skeppsvarvet Camper & Nicholsons. (Nautor's Swan Press Kit, u.å.)

## 2.1 Produkter

Nautor designar och bygger lyxjakter av världsklass som både erbjuder hög prestanda och bekvämlighet. Tillsammans med den Argentinske arkitekten German Frers och hans design team har Nautor ungefär 20 designers som utvecklar båtarna. Företaget utmärker sig i branschen tack vare att de klarar av hela produktionsprocessen själv och anses världsledande inom båtbyggarteknologi. Även om företaget tidigare kunde bygga sina båtar i serier har de nu tvingats specialisera båtarna allt mera efter kundernas behov. Detta gör att varje båt som lämnar Nautors skeppsvarv är unik.

Till Nautors produktsortiment hör segelbåtar av bland annat modellerna Swan 60, Swan 66, Swan 80, Swan 95, Swan 105 och Swan 115. Alla modeller har varianter som riktar in sig olika mycket på bekvämlighet eller prestanda. De olika varianterna kallas FD (Flushed Deck), S (Semi Raised Deck) och RS (Raised Deck).

Eftersom en stor del av företagets båtar används i tävlingssammanhang fokuserar de väldigt starkt på att göra båtarna snabba. Detta görs genom att hålla nere vikten och optimering av skrov och segel.



*Figur 1 Swan 90S, Alberto Cocchi*



*Figur 2 Swan 80FD, Carlo Borlenghi*

## 3 Teori

Det här kapitlet kommer att handla om den teori jag tagit till mig och valt att använda för att kunna slutföra mitt examensarbete. Jag kommer att behandla saker som är relevanta så att läsaren kan förstå varför jag tagit vissa beslut under arbetsprocessen. En stor del av examensarbetet har varit att försöka lista ut vilken typ av data som är relevant för företaget och jag kommer i detta kapitel att försöka förklara hur prioriteringarna gjorts. Det kommer också att handla en hel del om den teori som ingått i själva utvecklingen av applikationen.

### 3.1 Enterprise Resource Planning System

Ett Enterprise Resource Planning system eller kort ERP-system är ett datasystem som integrerar en organisations viktigaste processer i ett enda nätverk. Dessa processer kan inkludera kundrelationer, försäljning, inköp, produktionsplanering med mera. Målet med ett ERP-system är att göra organisationen eller företaget effektivare.

Ett ERP-system erbjuder många verktyg för att effektivera driften av ett företag. Till exempel automatiseras många processer och relevant data görs åtkomlig. Systemet gör det möjligt att definiera viktiga processer i företaget och att övervaka dem så att de sköts ordentligt. Systemet gör det också möjligt att skydda känslig data så att endast personer med passande rättigheter har åtgång till informationen. Det går också att få hjälp med att planera och förutspå hur framtiden potentiellt kan se ut när det gäller till exempel försäljning eller lagerhållning. När det gäller kundrelationer är det också vanligt att ERP-system erbjuder verktyg för att hantera dem. Kort sagt så är huvuduppgiften för ett ERP-system att översätta data till information som kan användas som bas när användaren skall ta beslut. (Syspro, u.å.)

Under de senaste decennierna har datoriseringen växt så kraftigt att det nu är ett måste för företag att kunna använda informationssystem. Systemen bidrar med att förenkla informationsflödet i organisationer och det kräver att personal på alla nivåer klarar av att arbeta med systemen. (Gripe & Rodello, 2011, s. 441-458)

Från början utvecklades ERP-system för att organisationer skulle kunna klara av de funktionella krav som ställdes på dem. Traditionellt inkluderar ERP-system funktioner för att hantera kundrelationer, varuhantering, lagerhantering, personalresurser och

organisationens ekonomi. Organisationer som implementerade ERP-system hoppades att med dess hjälp bli mera flexibla och effektivare. Företag som SAP och Oracle erbjuder standardiserade ERP-system som företag kan köpa och anpassa till sin verksamhet så gott det går. Ett funktionerande ERP-system höjer ett företags effektivitet genom att stärka de processer som det utför. (Mathrani & Mathrani & Viehland, 2013, s. 363-386)

### **3.2 Implementering av ett ERP-system**

Ett av de svåraste besluten en projektledare kan ställas inför är ifall ett ERP-system skall införskaffas och i sådana fall hurudant system som skall skaffas. Görs rätt beslut kan projektledaren bli en hjälte och görs fel beslut kan de ekonomiska resultaten bli katastrofala. I en ideal situation skall företaget klara av att implementera systemet inom utsatta budgeter. Både ekonomiska och tidsmässiga. När det gäller implementering av ERP-system finns det många parametrar att ta i beaktande.

En bra början kan vara att hyra in en konsult som hjälper med implementeringen. Faktum är att man inte vet vad man inte vet så att anlita en mera erfaren person kan visa sig vara mycket lönsamt. En konsult kan erbjuda guidning genom hela implementeringsprocessen. Många konsulter påstår att det går att spara pengar genom att anlita dem eftersom de kan erbjuda en programvara till lägre kostnad. Även om detta inte är sanningen kan det visa sig lönsamt i längden att ha använt sig av en konsult.

Det går också att förlita sig på en person inom företaget som kan arbeta med implementeringsprocessen. Ett problem med det kan vara att personen sannolikt inte har kunskaper som motsvarar de som en konsult har. Det är viktigt att lägga upp realistiska mål när ett ERP-system skall implementeras. Implementeringen är utan tvekan en svår process så företagets mål måste ha det i beaktande. (Hanawalt, 2015)

När små till medelstora företag skall implementera ERP-system kan resurserna vara mycket begränsade. Stora företag har mycket bättre kapacitet att hantera de problem som dyker upp under implementeringen. Detta gör implementeringsfasen betydligt mera riskfylld för mindre företag. I en ideal situation skall ett företag klara av implementeringen inom utsatt budget och tid. Detta är en utmaning eftersom det är svårt att planera hela implementeringsfasen innan processen påbörjats. (Ying, Colin, Mahmood, 2014, s. 358-360)

### 3.3 Engineering To Order

Engineer To Order innebär att kunden är väldigt involverad i tillverkningsprocessen. Till skillnad från till exempel Make To Order finns det inte en färdig produkt som går att sälja. Kunden har möjlighet att själv påverka designen för att de skall få en produkt som de är nöjda med. Denna företagsmodell är vanlig inom flygindustrin och energiindustrin. Det som mest skiljer åt Make To Order från andra modeller är att designfasen ofta överlappar tillverkningsfasen och försvårar hela tillverkningsprocessen. Detta leder ofta till mycket längre ledtider när produktionen skall ha till exempel råmaterial och ritningar. Ett företag som använder denna modell måste vara förberedda på att klara av dessa utmaningar. (Epstein, 2012)

Företag som använder sig av Engineer To Order är ofta små eller mellanstora och har ett starkt tekniskt kunnande. Deras klienter är ofta större företag som söker en leverantör som kan uppnå de krav som de ställer på produkten.

De utmaningar ett Engineer To Order företag ställs inför är flera. Till exempel kan det vara svårt att förutspå fel och brister innan produkten tillverkats eftersom det sannolikt inte gjorts något liknande tidigare. Det kan också vara svårt att hålla deadlines eftersom oförutsägbara problem dyker upp. Detta kan i sin tur leda till att företaget får mindre tid än väntat att ägna till testning av produkten. (Cutler, 2008)

### 3.4 Data kvalitet

Det kan vara en mycket svår uppgift att sköta om ett stort informationssystem. Att inte underhålla systemet tillräckligt bra kan leda till att det slutar att göra företaget mera effektivt och lönsamt. För att ett informationssystem skall fortsätta bidra med positiva effekter krävs det en viss investering från företaget. Denna investering kan göras genom att se till så att informationen i systemet är av hög kvalitet. Även om många företag upplever problem med kvaliteten på sin information är det få som gör något åt det.

I takt med att teknologin blivit en allt större del av företags verksamhet har det blivit allt viktigare att kontrollera information som flödar i företaget. För att ett företag skall vara marknadskraftigt, lönsamt och effektivt krävs det information av hög kvalitet. (Xu & Horn Nord & Brown & Daryl Nord, 2002)

I ett informationssystem förs data fram och tillbaka hela tiden. Systemet kan bli uppgraderat och ny mjukvara kan läggas till medan gammal tas bort. Databaser kan bytas ut och förändras. Detta leder ofta till att informationssystemet blir bättre medan själva informationen blir sämre. Data av hög kvalitet i kombination med ett bra informationssystem är en stor tillgång för ett företag. Medan data av låg kvalitet i kombination med ett bra system kan vara en stor börda. Informationsteknologi kan ses som en förstärkare, med bra data blir allting bättre och med dålig data blir allting sämre. Företag kan tjäna eller förlora stora summor pengar beroende på om de baserar sina beslut på bra eller dålig information. (Maydanchik, 2007)

För att veta ifall information är av hög kvalitet måste det finnas ett sätt att mäta dess kvalitet. För att kunna mäta data måste användaren av informationen ha en uppfattning om vad det innebär att data har hög kvalitet. För hög kvalitet krävs att informationen är fullständig, relevant, pålitlig, och konsistent. Det är även viktigt att informationen kommer i rätt tid och att den är användbar. Det är bra om informationen kan uppnå alla dessa punkter eftersom den inte är av hög kvalitet om den bara uppnår några av dem. (Bobrowski & Marré & Yankelevich, 1999)

### **3.5 Komplexa processer och utveckling av mjukvara**

Processen med att utveckla mjukvara är komplex och inkluderar de flesta fall många individer på alla nivåer av utvecklingskedet, både inom organisationen och ofta även utanför. Detta leder till att det är svårare och svårare att hantera system desto större de blir. De största systemen kan till och med anses vara omöjliga att kontrollera och kräva många erfarna experter. Detta beror ofta på att det är svårt att precisera kraven på systemen på grund av deras storlek. (Abu Rub & Issa, 2012, s. 122-137.)

Det finns många modeller som går att följa när mjukvara skall utvecklas. Några populära är The Waterfall model, Rapid Application Development model och The Prototyping Model.



The Waterfall model är den klassiska modellen där utvecklaren stegvis jobbar fram enligt en plan och vartefter uppfyller de mål som satts för varje steg i processen. Nackdelen med denna process är att det kan vara svårt att göra ändringar ett skede i processen är klart.

Rapid Application Development modellen följer konceptet att bra mjukvara kan produceras snabbt genom workshops och fokusgrupper där utvecklaren samlar krav på systemet, skapar prototyper och gör många tester. Utvecklaren är noggrann med att hålla skapade scheman och deadlines.

The Prototyping model går ut på att skapa en prototyp i form av en tidig gissning på hur slutprodukten kommer att se ut. Sedan byggs det vidare på den och utförs tester tills en prototyp har uppnåtts som fyller alla krav. Utgående från den så skapas sedan det slutgiltiga systemet. (Rouse, 2010)

För det här arbetet har The Prototyping Model valts som lämplig metod. Detta på grund av projektets natur. När det inleddes var företaget ganska osäkert på vad de ville ha. Med The Prototyping Model finns möjlighet att utveckla projektet under dess gång vart efter man märker nya mål man vill uppnå.

De flesta projekt som går ut på att utveckla mjukvara misslyckas på ett eller annat sätt. Upp till 80 procent av alla försök lever inte upp till de mål som satts för dem. Detta beror ofta på att de inte håller budgeten eller saknar funktioner som det kommits överens om på förhand. För att utvecklingen av ny mjukvara skall lyckas finns det flera saker som är bra att tänka på. Det är bra att bestämma sig på förhand för vilken utvecklingsmodell som skall användas, vilka funktioner programvaran skall innehålla hur designen skall se ut och att testa programmet innan användning. Det är även bra att planera ibruktagning och överföring av data från tidigare system. (Perks, 2006)

## 4 Teknologier

För att kunna utveckla en webbapplikation måste man använda sig av flera olika teknologier. I detta kapitel kommer de teknologier som använts för att utveckla applikationen att presenteras.

Valet av använda teknologier och redskap för detta examensarbete grundar sig på flera olika saker. Det är förstås viktigt att utvecklaren av applikationen kan använda de verktyg som behövs, och förstår sig på de teknologier som är involverade i utvecklingsprocessen.

### 4.1 Back-end

Det är viktigt att teknologierna är pålitliga och har bra dokumentation. Som exempel kan vi ta programmeringsspråket PHP som använts för att skriva applikationen. Det är ett gammalt och extremt vanligt språk på webben med bra dokumentation och stöd från organisationen som utvecklar det. PHP är rekursiv akronym för PHP: Hypertext Preprocessor. Det är ett programmeringsspråk som används på webbservrar för att skapa dynamiskt innehåll. Att det används på webbservrar betyder bland annat att användare av hemsidor eller webbapplikationer aldrig ser PHP-filer utan endast får den data som filerna levererar.

För mitt examensarbete är PHP mycket viktigt eftersom det kan kallas för hela webbapplikationens motor. Till exempel så hanterar PHP filerna det data som användaren skickar till servern och sparar den i databasen. Som alternativ fanns programmeringsspråket Python, ett språk som jag från tidigare kunde bra. Men det fanns bättre stöd för PHP både från företagets sida och på internet. (PHP, u.å.)

Som server användes Apache som är en fri webbserver. Servern är utvecklad av Apache Software Foundation och är världens mest använda webbserver. Apache släpptes redan 1995 så det är en gammal server som ständigt uppgraderats under åren. En stor anledning till att Apache blivit så populär är att det är gratis programvara med öppen källkod. (Apache, u.å.)

## 4.2 Front-end

På klientens sida används HTML, som står för HyperText Markup Language, och är den struktur som allting på internet följer. Första versionen skapades av Tim Berners-Lee redan år 1991. HTML har alltså använts ända sedan internets uppkomst. I praktiken används HTML för att strukturera dokument genom att definiera till exempel rubriker och stycken.

HTML5 är den nyaste versionen av HTML och med den kom många nya funktioner. Detta har lett till att webbapplikationer har blivit mycket populärare än tidigare. HTML5 inkluderar funktioner för att hantera video, ljud och förstås text och bild. Sedan 2014 har HTML5 varit den rekommenderade versionen att använda av World Wide Web Consortium. (W3, u.å.)

CSS som står för Cascading Style Sheet och används tillsammans med HTML för att ändra på presentationsstilen på dokument är också en viktig del av applikationen. CSS kan användas för att ändra på textens typsnitt eller storlek och för att ändra bakgrundsfärgen på dokument och mycket mera. Det är en gammal teknik som skapades redan 1994 av Håkon Wium. (W3, u.å.)

För att göra applikationen så användarvänlig som möjligt användes JavaScript. Det är ett High-level programmeringsspråk som används mestadels på webben. Javascript exekveras eller med andra ord körs på klientens egna dator i deras webbläsare ifall den klarar av Javascript, vilket alla moderna webbläsare gör.

Man använder Javascript genom att bädda in det i dokument skapade med HTML och kan köras när användare laddar webbsidor eller använder funktioner på dem. Vanliga användningsområden för Javascript är att till exempel växla annonsbilder med en viss frekvens eller kontrollera data som användare fyller i på sidor innan den sparas. (Mozilla, u.å.)

För att underlätta utvecklingen med JavaScript användes JQuery, som är ett extremt populärt JavaScript-bibliotek. Det används för att förenkla processen med att använda JavaScript. Eftersom JQuery bygger på JavaScript kan det göra allt som JavaScript kan, även med JQuery. Biblioteket är det mest populära och används på mer än 30 procent av de största sidorna på internet. (JQuery, u.å.)

Applikationen måste se bra ut och fungera på olika stora skärmar därför användes en teknologi som heter Bootstrap. Bootstrap är ett HTML, JavaScript och CSS ramverk som användes för att snabba upp utvecklingen av webbapplikationen. Ramverket används på

miljontals sidor på internet. Den största fördelen med att använda Bootstrap är att det förenklar processen med att göra applikationer anpassningsbara för olika stora skärmar, exempelvis vanliga datorskrmar och telefonskrmar. (Bootstrap, u.å.)

### 4.3 Databas

Som databashanterare för webbapplikationen blev MySQL valt. MySQL är fri programvara och är en av de populäraste databashanterarna i världen. Det används oftast på Linux men finns även på många andra vanliga operativsystem, som till exempel Windows. (MySQL, u.å.)

MySQL använder sig av frågespråket SQL, som de flesta databasadministratörer är vana att använda. Så var även fallet vid Nautor, vilket bidrog till valet att använda MySQL. Programmets huvudsakliga utvecklare är två finlandssvenskar vid namn David Axmark och Michael Widenius. Programmet har sedan 2008 ägts av Sun Microsystems som sedan föll i Oracles ägo. (Oracle, u.å.)

För att administrera MySQL-databasen har ett verktyg vid namn phpMyAdmin blivit använt. Programmet är skrivet i PHP vilket ansågs vara en fördel eftersom själva applikationen även var skriven i PHP.

Med phpMyAdmin kan användaren enkelt och snabbt sköta om sin databas via sin webbrowser. Till exempel kan phpMyAdmin skapa nya databaser eller lägga till och radera data. phpMyAdmin klarar av att importera och exportera data till många olika format, bland annat CSV, SQL och XML. (PhpMyAdmin, u.å.)

## 5 Inledande arbete

Målet med arbetet var relativt klart när arbetet påbörjades, men dessvärre var vägen dit inte lika klar. Kortfattat var uppgiften att ta fram en applikation som skulle hjälpa företaget att ta reda på vad de hade för krav på ett potentiellt ERP-system. Detta skulle uppnås genom att skapa applikationen så att den var mycket anpassningsbar för att göra det möjligt att testa olika förslag och idéer som kan dyka upp.

När projektet inleddes fanns det några krav som man från Nautors sida visste att man ville ha med i den slutgiltiga applikationen. Följande punkter är saker som man ville ha med från början:

- Man skulle kunna granska var produktionsarbetare befann sig för dagen.
- Man skulle kunna printa ut en rapport över var alla befann sig.
- Det skulle finnas någon typ av visualisering över var alla befann sig.
- Man ville ha ett nytt övervaknings- och rapporteringssystem för kvalitetsuppföljning.

Efter hand dök det upp flera önskemål och idéer för applikationen men de ovannämnda fanns med från början.

Den första funktionen, där man önskade sig ett sätt att se var alla befann sig i produktionen kom från att man ville få en bättre uppsikt över hur man hade placerat personalresurserna. För tillfället så var det bara arbetsledarna som hade uppdaterad information om var produktionspersonalen befann sig i realtid.

Den andra funktionen där man ville kunna printa ut en rapport över var personalen befann sig var ämnad att fungera som en sorts säkerhetsfunktion ifall till exempel en eldsvåda skulle bryta ut. Det fanns redan ett sätt att få en dylik rapport men det krävdes många mellansteg och det data man fick fram var rätt så otydligt och intetsägande för någon som inte jobbar på företaget, till exempel en person från räddningsverket.

Behovet av en visualisering kom från produktionschefens sida. Han hade i tidigare arbeten jobbat med system som hade bra visualiseringsfunktioner och såg gärna att Nautor skulle ha något motsvarande.

Företaget ville också ha ett system för kvalitetsövervakning av deras projekt. Nautor hade redan ett system som byggde på ganska avancerade Excelfiler. Excelfilerna hade utvecklats av ett annat företag åt Nautor. Filerna var lätta att använda och tydliga att fylla i, men det fanns flera problem med dem. Flera personer kunde inte jobba på samma fil samtidigt. Detta ställde till med stora problem när företaget närmade sig kontrolleringstillfället där kvalitetsövervakningen skulle gås igenom. Detta ledde till misstag när man slog ihop flera filer till en enda. Ett annat problem med Excel-filer är att desto fler som jobbar på dem, desto sannolikare är de att bli korrupta, vilket betyder att det inte längre går att öppna filerna. Därtill hade antalet projekt lett till att det blivit svårt att hålla koll på alla olika Excelfiler. Det kunde även vara mycket svårt att jämföra data från de olika projekten eftersom all data fanns i olika filer.

Under arbetets gång dök det som tidigare nämnt upp flera nya förslag på funktioner som man ville ha i applikationen.

- Arbetsledarna hade ett förslag på en typ av pool-funktion.
- Kvalitetschefen ville ha ett sätt att automatiskt skicka en notifiering till ansvarig när ett kvalitetsproblem fylldes in i applikationen.
- Produktionschefen ville ha ett sätt att hålla koll på tidsåtgången för alla projekt och deras skeden.

När det fanns en lista över saker som skulle finnas med började projektet att klarna lite men det var ännu mycket som var oklart. Till exempel var det ännu inte beslutat vilken teknologi applikationen skulle tillverkas med.

Ett förslag i ett tidigt skede var att göra applikationen helt och hållet i Excel och VBA, det vill säga Visual Basic for Applications. Andra alternativ till detta var en skrivbordsapplikation eller en webbapplikation.

Efter att ha testat att tillverka applikationen i Excel blev det snabbt klart att programmet hade allt för många restriktioner. Det blev problem när flera användare skulle ha tillgång till

samma fil på samma gång. Även automatisering av till exempel sparandet av data blev mycket invecklat. Ett stort problem var också att formlerna som styrde applikationen blev väldigt långa och tvingades snurra genom flera loopar för att bearbeta all data. Detta ledde till att detta tillvägagångssätt blev skrotat relativt snabbt. Som kommer att beskrivas användes ändå Excel för att samla in data som grund till den slutgiltiga applikationen.

De andra alternativen till typer av applikationer var skrivbordsapplikation och webbapplikation. Från IT-avdelnings sida ville man helst inte ha en skrivbordsapplikation av den orsaken att den i sådana fall skulle behöva installeras på datorerna för alla som skulle använda den.

Alternativet som återstod var en webbapplikation. Detta var både uppdragsgivaren och jag själv nöjda med, eftersom jag hade kunskaper inom webbt teknologier och det skulle kräva minsta möjliga administration från företagets sida när applikationen skulle tas i bruk och underhållas.

En webbapplikation kan köras och uppdateras på en enda server. Detta gör att användare av applikationen inte behöver installera något nytt program eller sköta uppdateringar själva. Vilken relativt modern webbläsare som helst skulle klara av att köra applikationen.

## **5.1 Insamling av data**

För att man skulle ha någon data att hantera i applikationen behövde man få den någonstans ifrån. Helst skulle denna process vara så automatiserad som möjligt för att spara tid. På ett par planeringsmöten behandlades exakt vilken data man ville ha och vem på företaget som för tillfället hade tillgång till den.

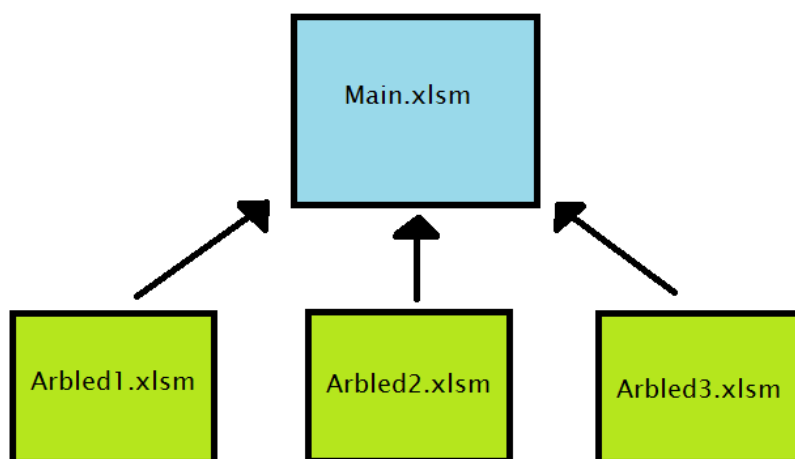
Alla data som skulle föras in i applikationen måste föras in manuellt. Med andra ord så användes inga censorer på maskiner eller dylikt, utan informationen skulle baseras på vad användare förde in för hand. På grund av detta var det extra viktigt att göra applikationen så användarvänlig och praktisk som möjligt för att användare skulle vilja använda den. Om man kan förvänta sig att all data förts in i applikationen i tid och korrekt kan man lita på informationen.

## 5.2 Manpower

Den första delen av applikationen som skapades skulle komma att bli kallad Manpower och dess uppgift var att hålla koll på hur man hade allokerat personalresurserna i produktionen vid Nautors alla tre fabriker.

Information om var personalen befann sig för dagen fanns hos varje arbetsledare eftersom det var deras uppgift att dela ut uppgifter för dagen. Så ett sätt att få den informationen digitaliserad och samlad på ett enda ställe var första uppgiften.

Ett prototyp system skapades i Excel, detta på grund av att alla arbetsledare var vana att jobba med det programmet och det går snabbt att tillverka enkla prototyper i Excel.



*Figur 3 Excelsystem*

Ett anspråkslöst system byggdes upp med hjälp av Excel och några moduler skrivna i VBA. Varje arbetsledare som hade ett eget projekt de ansvarade över eller en avdelning som de höll koll på, tilldelades en Excel-fil som skulle fyllas i varje morgon.



Project	Number	Name	Supervisor	Ext/Int	Status	Task
Projekt 1	56897	Stina	Knut	Ext	Work	Montering
Projekt 2	56897	Erik	Knut	Int	Work	Montering
Projekt 3	56897	Karl	Knut	Ext	Sick	Montering
Projekt 4	56897	Otto	Knut	Ext	Work	Montering

Figur 4 Excelblad

I Excelfilen fanns en karta eller motsvarande visualisering över deras avdelning med numrerade sektioner. Under visualiseringen fanns en lista med alla anställda och ett antal kolumner med data som skulle fyllas i.

För varje anställd som en arbetsledare hade ansvar för så fanns sju fält att fylla i.

- Projekt
- Nummer
- Namn
- Arbetsledare
- Extern eller intern
- Status
- Uppgift

Under projekt så fyllde man i vilket projekt den anställda arbetade med. Detta ändrade ganska sällan så arbetsledaren behövde inte ändra det ofta. Under nummer och namn så fylldes den anställdes id-nummer och namn i. I arbetsledarkolumnen fyllde arbetsledaren i

sitt eget namn. Under extern och intern fylldes det i ifall den anställda var en intern eller extern anställd. Det vill säga om det var inhyrd arbetskraft eller inte. I statuskolumnen fyllde man i vad de anställda hade för status för dagen. Till exempel om den var i arbete, sjuk eller ledig. Under uppgift så fylldes dagens uppgift i. Detta var tänkt som ett sätt att veta till exempel var i en båt en anställd jobbade.

Några av dessa kolumner togs bort i webbapplikationen eftersom de inte ansågs tillräckligt viktiga. Till exempel så försvann extern/intern och uppgift samt nummer.

För att göra ifyllandet så enkelt som möjligt användes Excels dataverifieringsfunktion för att begränsa vad användarna kunde fylla i de olika fälten. När en användare klickade på ett fält de ville fylla i så kom det upp en lista med giltiga alternativ. Tanken var att arbetsledaren endast skulle behöva göra ändringar i sin fil ifall det skedde en ändring på deras avdelning, till exempel om någon blev sjuk eller flyttade till ett annat projekt under dagen. På detta vis minimerade man tidsåtgången när det gällde insamling av data.

Eftersom en arbetsledare var ansvarig över samma personer under en längre tid och ett projekt på Nautor pågår under en relativt lång tid så behövde de oftast bara ändra vilken sektion de anställda jobbade under och vad de hade för status.

Med hjälp av Excels olika låsfunktioner var filen gjord så att man inte skulle kunna ändra något annat än det som var tillåtet. Detta var ett sätt att undvika att användare skulle konstra med filerna och till exempel bryta länkar i systemet.

Hela systemet var uppbyggt så att det styrdes via en huvudfil. I huvudfilen kunde man få en snabb överblick av alla data som samlats in och så kunde man ändra de listor som begränsade vad användarna, i detta fall arbetsledarna kunde fylla i som alternativ i sina respektive filer.

Huvudfilen sparade också dagligen det data som samlats in så man skulle få en slags historik från dag till dag. Data kunde användas i en graf som gick att ställa in så man såg hur resurserna allokerats för varje dag.

Arbetsledarna var rätt så nöjda med detta system, eftersom det gav dem en viss struktur i planeringen och ett sätt för dem att snabbt se hur andra arbetsledare allokerat sina personalresurser.

Men nackdelarna med att använda ett system uppbyggt i Excel var flera och som tidigare nämnts användes inte Excel i den slutgiltiga applikationen överhuvudtaget. Till exempel var det mycket svårare att få till en automatiserad applikation än vad det skulle vara att göra den med hjälp av någon annan teknologi och så hade man på företaget erfarenhet av att Excel-

filer som öppnas många gånger om dagen av olika personer har en tendens att bli korrupta. Och ifall de blev korrupta måste man gå via IT-avdelningen för att återställa dem från serverns backupsystem. Sådana saker behöver man inte oroa sig för om man använder en teknologi lämpad för många användare.

### 5.3 Quality Log

Nästa del som blev planerad till applikationen var ett nytt system för införing och uppföljning av kvalitetsproblem. På Nautor är kvaliteten av högsta vikt så ett fungerande system var verkligen viktigt. Att samla in data till denna sektion var enklare eftersom det inte var lika många personer involverade i införandet av data.

Från tidigare hade företaget ett sorts Excel baserat system som de köpt av ett företag som utvecklar mjukvara. Detta system var mycket enkelt uppbyggt på olika templates som användes varje gång man startade ett nytt projekt.

Detta system fungerade relativt bra och man var van att använda det på företaget. Men som tidigare påpekats finns det olika problem som dyker upp när man förlitar sig på enbart Excel och dessa problem dök även upp här.

För det första så upplevde man på företaget att man drunknar i Excel-filer och att det är svårt att veta vilken som skall fyllas i när man hittar ett kvalitetsproblem. Ett annat problem var att det blev problem när flera personer behövde jobba med samma fil samtidigt, till exempel just före ett möte där man skulle gå igenom kvalitetsövervakningen.

Men det som var positivt för mitt uppdrag var att all data fanns tillgänglig i Excel-filerna och kunde importeras till den nya applikationen utan desto mera bekymmer. Den nya applikationen blev uppbyggd så att den använde nästan exakt samma data som man använt sig av tidigare. Detta på grund av att det tidigare systemet fungerade bra förutom att det var uppbyggt på utspridda Excel-filer. Om man gjorde den nya applikationen på ett liknande sätt skulle övergången också bli betydligt smidigare än om man tänkte ut ett helt nytt sätt att hålla koll på kvalitetsövervakningen.

En hel del data fördes in för varje problem som upptäcktes, detta för att man skulle kunna följa upp och föra statistik över var problem uppstår. Data som fördes in var följande:

- Vem som var ansvarig för att reda upp problemet
- När problemet blivit upptäckt
- Under vilken fas man upptäckt problemet
- Vilken kod problemet hade
- Förslagen åtgärd
- En beskrivning av problemet
- Fysisk placering av problemet

Den nya applikationen behövde åtminstone klara av att hantera denna data och även den data som kom på förslag att läggas till i den.

Med en riktig webbapplikation skulle det vara lätt att automatisera saker som vilken tid och datum som felet blivit infört i systemet och av vem det blivit infört.

#### **5.4 Pool**

En funktion som dök upp under ett senare skede av utvecklingen var den så kallade Pool-sektionen. Detta var ett önskemål från arbetsledarna. De ville ha ett sätt att lägga anställda i en arbetspool ifall de hade extra kapacitet. Arbetsmängden på Nautor är mycket varierande från dag till dag och vecka till vecka, så detta var något man hade frågat efter en längre tid.

För att samla in data till denna funktion krävdes inte desto mera arbete. Införande av data skulle vara upp till arbetsledarna själva vart efter de använde funktionen i webbapplikationen.

Den data som behövde föras in under utvecklingen i systemet var en lista på alla anställda som gick att lägga i arbetspoolen. Och under användningen skulle man kunna fylla i ett namn på vem som läggs i arbetspoolen, ett startdatum, ett slutdatum och en kort kommentar. Sedan kan en arbetsledare gå in och se vilka som är tillgängliga och som de kan plocka till ett arbete.

## 5.5 POC

Som tidigare nämnts hade produktionschefen även ett önskemål om att få en funktion där man kunde följa upp tidsåtgången för alla projekt och alla moment under produktionsfasen. Detta genom att räkna ut POC, det vill säga Percentage Of Completion.

Planeringen av denna funktion inleddes men blev avbruten på grund av olika saker, till exempel så krävde andra saker att man fokuserade på dem och inte hoppade från skede till skede i projektet. POC funktionen visade sig också snabbt vara ett betydligt större projekt än man trott från början. Detta resulterade i att utvecklingen av POC funktionen inte rymdes med i detta examensarbete, och det kommer således inte heller att gås mera in på hur funktionen sedan blev.

## 5.6 Utveckling av applikationen

När man visste vad man ville ha för typ av data i den slutgiltiga applikationen och hur man kunde samla in den var det dags att börja planera och utveckla själva applikationen.

Som nämns i teorikapitlet av detta arbete valdes ett tillvägagångssätt som kallas The Prototyping Model. Detta på grund av att man inte riktigt visste från början vad det är man vill ha och hur man skall uppnå det. Med detta tillvägagångssätt har man möjlighet att ändra sig vartefter nya idéer dyker upp.

En nackdel med detta sätt att projektet snabbt kan bli lite av ett lappverk av olika idéer och att man kan tvingas göra om hela sektioner av applikationen för att få ordning på saker och ting.

## 5.7 Webbapplikation

Det var nu klart att applikationen skulle göras i form av en webbapplikation. Det betyder att man behöver ett serversystem att ha den på. Till detta valdes Apache eftersom det är det helt klart vanligaste systemet på webben i dagens läge och har så varit under en väldigt lång tid.

Vilket språk applikationen skulle skrivas i behövde också klargöras. Efter lite planering med IT-avdelningen på Nautor föll valet på PHP, ett mycket pålitligt språk som jag själv hade tillräckligt kunnande inom för att kunna programmera applikationen med det.

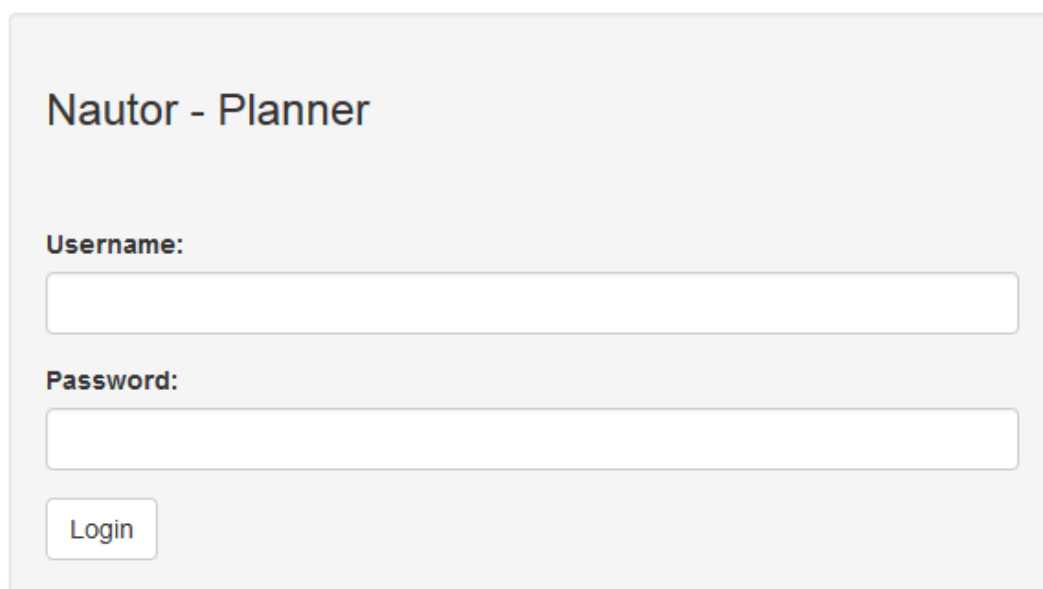
För att applikationen skulle bli interaktiv och användarvänlig så användes JavaScript i samband med JQuery. Och för att skynda på designprocessen användes ramverket Bootstrap som beskrivs i teorikapitlet.

Under utvecklingen av den slutliga webbapplikationen användes det Excelsystem som tidigare beskrivits för att samla in data. Detta visade sig vara mycket bra eftersom det dök upp önskemål om funktioner och det dagliga rapportandet blev en vanesak för användarna.

I detta skede blev även databasen skapad. Som nämnts i teorikapitlet användes MySQL som administrerades med hjälp av PhpMyAdmin.

## 6 Nautor – Planner

Webbapplikationen fick namnet Nautor – Planner. Varje användare av applikationen har en unik profil som de loggar in på när de skall arbeta. Den fungerar bra i alla moderna webbläsare även om vissa knappar och liknande kan se lite olika ut.



The image shows a login form for 'Nautor - Planner'. The form is contained within a light gray rounded rectangle. At the top left of the form, the text 'Nautor - Planner' is displayed in a dark gray font. Below this, there are two input fields. The first is labeled 'Username:' and the second is labeled 'Password:'. Both labels are in a bold, dark gray font. Each label is followed by a white rectangular input field with a thin gray border. Below the password field, there is a button labeled 'Login' in a dark gray font, enclosed in a white rounded rectangle with a thin gray border.

*Figur 5 Login*

### 6.1 Manpower

Den första sektionen som blev klar för bruk var Manpower-funktionen. Dess uppgift var att göra det enkelt för arbetsledarna att rapportera och hålla koll på hur de allokerat personalresurserna i produktionen vid Nautors tre fabriker.

Name	Foreman	Boat	Department	Status	Delete
KARIN	Rolf	Projekt 2	Avdelning 1	Vacation	Delete
KARL	Knut	Projekt 2	Avdelning 3	Sick	Delete
PELLE	Knut	Projekt 1	Avdelning 1	Work	Delete

*Figur 6 Manpower Log*

Applikationen använder sig av AJAX-teknologi för att användaren skall kunna skicka data till servern utan att hela sidan laddas om. På så vis blev användningen av Manpower mycket smidig. Användaren upplever att allting görs i realtid vilket även är sanningen.

Högst upp på sidan finns en sök funktion. Om användaren skriver in en bokstavskombination, till exempel ett namn, en arbetsledare eller namnet på ett projekt så filtrerar applikationen bort alla alternativ som inte innehåller kombinationen. På så vis kan användaren snabbt hitta till exempel de anställda som han har ansvar för eller alla anställda under ett visst projekt.

För varje anställd finns det fyra olika rutor som man kan fylla i. I den första rutan så kan man ändra vilken arbetsledare som har ansvar för den anställda. Detta är viktigt eftersom anställda ofta byter uppgifter och avdelningar på Nautor. I den andra rutan så kan man fylla i vilken båt den anställde jobbar på. Och i den tredje rutan så går det att ange under vilken avdelning personen i fråga jobbar. I den fjärde och sista rutan så anger man personens status för dagen. I den rutan finns alternativ som på jobb, sjuk och pappaledig.

I rutorna är det begränsat vad man kan fylla i, precis som det var i det tidigare Excelsystemet. Begränsningarna funkar så enkelt att PHP läser in godkända alternativ från några textfiler på servern.

Det finns även en knapp för att ta bort en anställd som slutar. Knappen blev diskuterad en del innan den togs med i programmet eftersom det inte är bra om man råkar ta bort någon i misstag. Men arbetsledarna var noga med att de vill ha en knapp för att ta bort anställda så den togs med i den slutgiltiga applikationen.



NAUTOR'S SWAN Profile Manpower Pool Quality

Visualization Log Add New

Add new employee to database

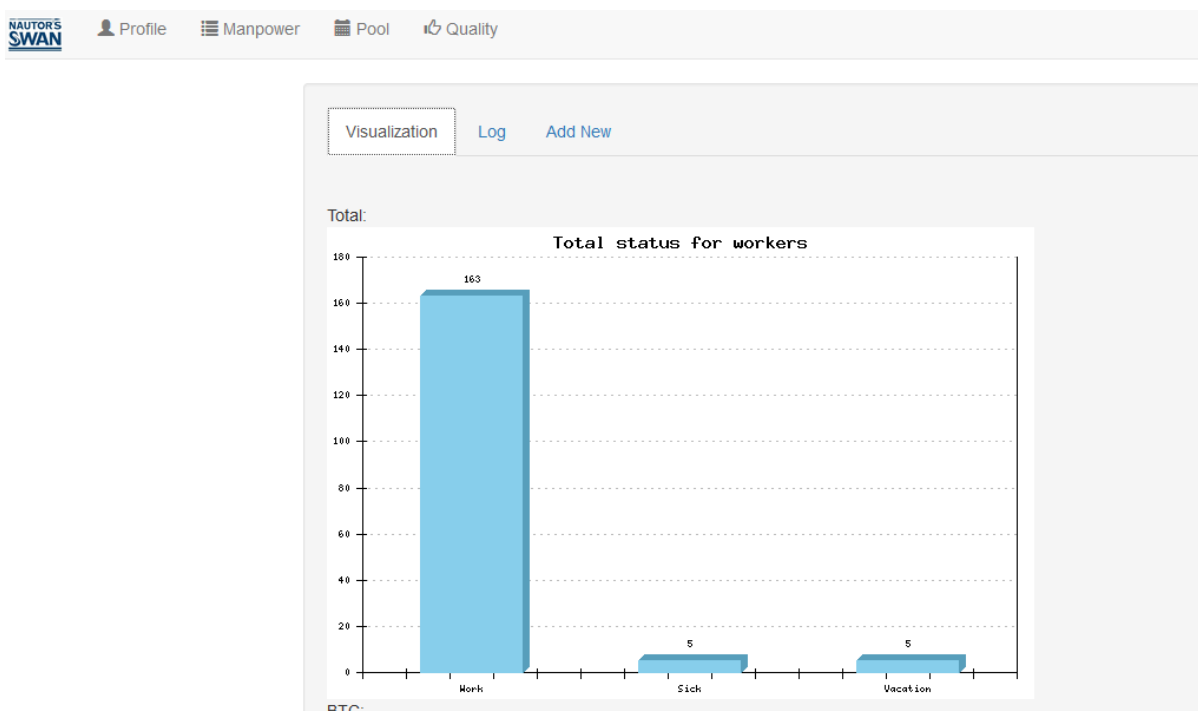
Full Name:

Save

*Figur 7 Manpower Add New*

Under Manpower finns också en flik där användare kan lägga till nya anställda i databasen. Tanken har genom projektet varit att ge användarna så mycket frihet som möjligt för att undvika onödiga mellanhänder. Till exempel som att tvinga arbetsledarna att skicka en begäran för att få med en ny anställd i systemet.

När man öppnar Manpower i applikationen möts man först av en mycket enkel visualisering i form av olika grafer. Det finns en graf som visar hur statusen för anställda ser ut totalt för de olika fabrikena och sedan en skild graf för varje fabrik.



*Figur 8 Manpower visualisering*

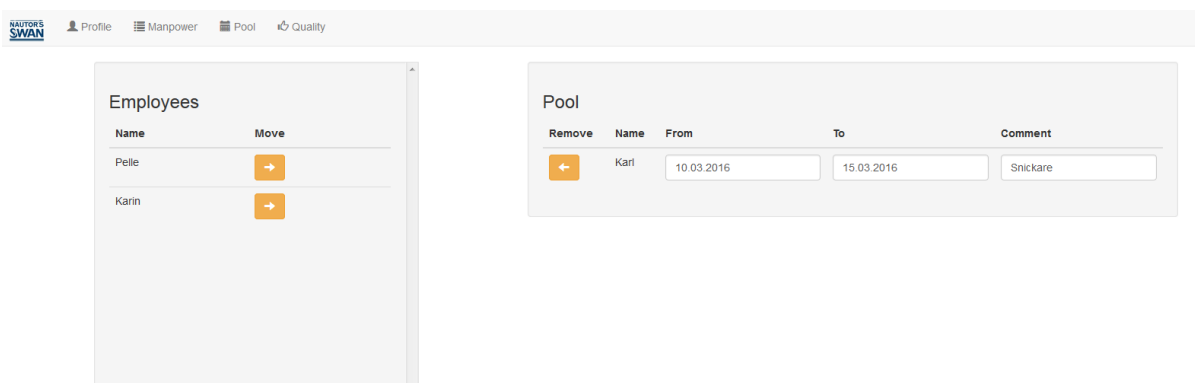
Graferna är uppdelade så att de visar varje projekt enskilt och varje avdelning under det projektet. Graferna genereras av PHP och visar alltid situationen i realtid. För användaren visas de i bildformat så de är lätta att ladda ner till datorn för till exempel användning i rapporter.

Manpower inkluderar även en knapp som skapar en förenklad vy av hela produktionen och en lista över var de anställda befinner sig. Denna rapport går snabbt att printa ut och är mycket tydlig. Detta var ju ett av kraven från Nautor, för att de snabbt skulle kunna printa ut en rapport ifall det blir till exempel en eldsvåda.

Rapportering av var anställda befinner sig är kanske inte så noggrann som krävs för att man skall hitta dem inne i fabriken ifall det sker till exempel en eldsvåda och fastigheten fylld med rök, men för dagligt bruk i arbetet räcker den bra till. Detta beror på att man ville börja med ett lite enklare system så att webbapplikationens ibrukttagande inte skall bli för krångligt.

## 6.2 Pool

Nästa del av applikationen som blev klar att tas i bruk var den så kallade Pool delen. Denna del blev tillverkad på önskemål från arbetsledarna.



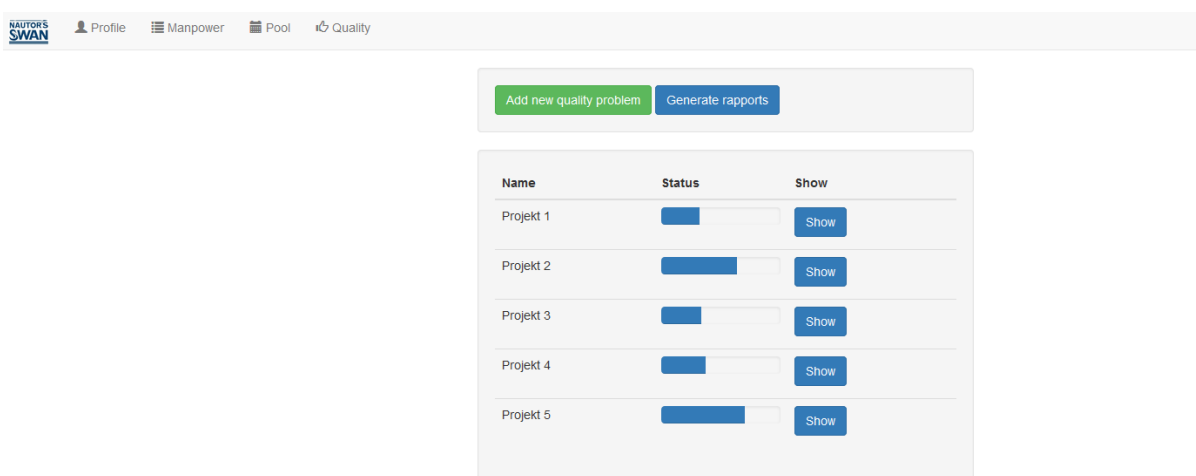
*Figur 9 Pool*

Systemet bakom funktionen är mycket enkelt. En arbetsledare kan klicka på en anställd i vänstra stapeln för att flytta den till arbetspoolen. Arbetsledaren kan då också ange ett startdatum och ett slutdatum för hur länge den anställde finns i poolen. Det går även att lägga

till en kommentar för att andra skall veta till exempel vad den anställde jobbar med sedan tidigare.

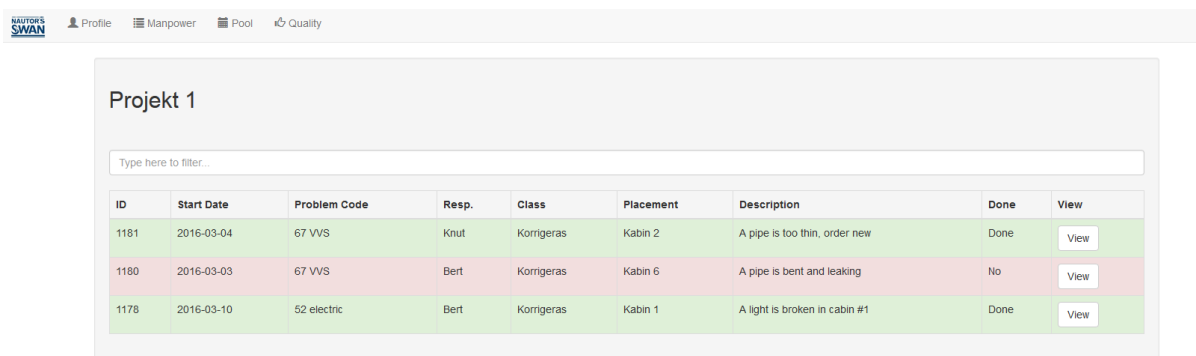
### 6.3 Quality Log

En stor del av tiden under utvecklingen lades på funktionen för kvalitetsövervakning. Utvecklingen skedde i nära samarbete med kvalitetschefen för att ett fungerande system skulle uppnås.



Figur 10 Quality Log

Det första som dyker upp när man öppnar Quality Log är en lista över alla projekt och en enkel visualisering av i vilket skede kvalitetsövervakningen för ett specifikt projekt är. Det går i denna vy att klicka sig vidare för att granska en specifik process.



Figur 11 Avvikelser för ett projekt

När man klickat sig vidare får man upp en lista över alla kvalitetsproblem som hittats på projektet. I de olika kolumnerna får man en snabb bild över vad kvalitetsproblemet handlar om. Här kan man använda en filterfunktion liknande den i Manpower för att hitta ett specifikt problem.

Vill man granska ett specifikt problem närmare kan man öppna det med Show-knappen.

Under Quality Log kan man även föra in nya kvalitetsproblem som upptäcks i systemet. Kvalitetsproblemet sparas i databasen och finns från och med det i systemet

The screenshot shows a web interface for adding a new quality problem. The form is titled 'Add new quality problem:' and contains the following fields:

- Project: A dropdown menu.
- Phase: A dropdown menu.
- Responsibility: A dropdown menu.
- Placement: A dropdown menu.
- Item #: A text input field with a small icon on the right.
- Problem code: A dropdown menu.
- Proposed action: A dropdown menu.
- Description: A large text area.

At the bottom of the form, there are three buttons: 'Save Quality Problem' (green), 'Empty' (orange), and 'Save and make another' (blue).

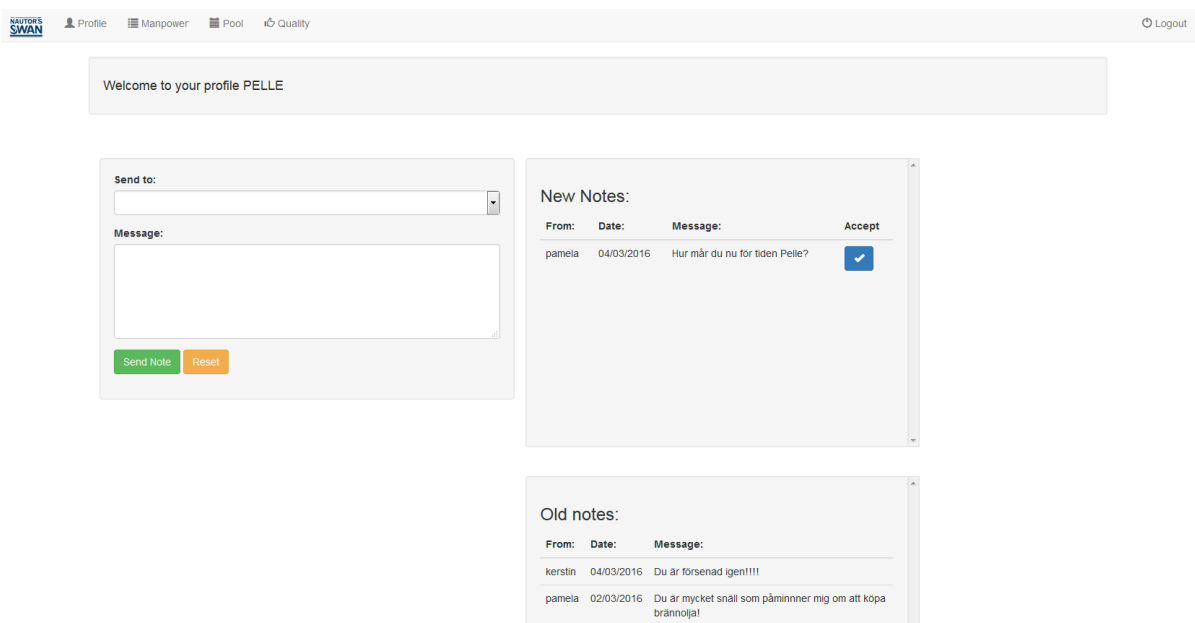
*Figur 12 Lägg till ny avvikelse*

Vilken data som skall sparas för varje kvalitetsproblem har bestämts från det tidigare systemet som fanns vid Nautor. En del har automatiserats som till exempel vilket datum och vilken tid problemet har förts in i systemet. När detta arbete blev skrivet var Quality Log fortfarande under utveckling. Det fungerar och används men förbättringar dyker upp vart efter användare kommer på dem.

## 6.4 Notes

Ett sorts meddelandesystem skapades även i Nautor – Planner. Tanken bakom detta system var att man skulle kunna skicka ett automatiskt meddelande till ansvarig person ifall ett kvalitetsproblem blir sparad i systemet. Men meddelandesystemet gick snabbt att utveckla

till en mera fullskalig funktion som i framtiden kan användas för att skicka vilka meddelanden som helst eller kan bli integrerad i andra funktioner i webbapplikationen. Under profil tabben kan varje användare läsa meddelanden som de har fått i applikationen. Till exempel om ett nytt kvalitetsproblem har dykt upp som de har ansvar för att reda upp.



Figur 13 Profilsida med notes

## 6.5 Ibrukttagande och skolning

När Nautor – Planner var redo att tas i bruk skrevs en kort och tydlig användarguide som skickad ut till alla användare via mejl. Eftersom det var en webbapplikation behövde de bara följa en länk och logga in. Under de följande dagarna höll jag skolningar för att kunna vara säker på att användarna visste hur applikationen skulle användas. Skolningarna tog cirka en kvart per användare och innehöll genomgång av alla de viktigaste sakerna.

Under ibrukttagandet av webbapplikationen visade det sig åter att det var ett bra beslut att använda ett enklare Excel-system under utvecklingen. Eftersom användarna redan var vana att rapportera dagligen och webbapplikationen fungerade betydligt smidigare och snabbare så blev det en klar förbättring.

## 7 Resultat

Resultatet av detta arbete blev en fullständig webbapplikation som tagits i bruk vid Nautor. Målet var att ta reda på vilka saker som skulle kunna önskas av ett ERP-system i framtiden och att förbereda företaget på en fullskalig flytt till ett ERP-system.

Att ta reda på vilka saker företaget skulle kunna önska från ett ERP-system uppfylldes så långt det var möjligt inom ramen för detta examensarbete. Ett ERP-system är förstås mycket större än Nautor – Planner och sträcker sig över flera avdelningar inom ett företag. Till exempel rör inte webbapplikation alls vid ekonomi eller lagerhållning men inom de avdelningar som hunnit bli inkluderade i webbapplikationen i detta skede börjar en bild att träda fram av vad man skulle kunna behöva i ERP-system.

För att förbereda företaget på att ta i bruk ett ERP-system och vänja personal vid tanken har detta examensarbete fungerat mycket bra. Under projektets gång har det mycket klart framkommit hur mycket arbete det krävs för att planera saker som datainsamling och processbeskrivningar. Det har också blivit tydligt hur man kan förenkla processer genom att inkludera dem och koppla dem samman i ett större system för att kunna jämföra data snabbt och enkelt.

Det som inte rymdes med var en funktion för att hålla koll på tidsåtgången för de olika projekten. Detta visade sig vara ett mycket stort projekt i sig och kommer att kräva en ordentlig genomgång av diverse processer innan man kan börja bygga upp ett system kring det. Nautors nuvarande system fungerar bra men kräver många mellanhänder och flera olika dataprogram innan informationen är presenterbar. Därmed finns rum för förbättringar.

Från användarna kom det mycket respons vilket skapar möjligheter för potentiell vidareutveckling. Arbetsledarna var nöjda med webbapplikationen och glada att deras röst blivit hörda när de önskat Pool applikationen. Uppdragsgivaren var nöjd på grund av att webbapplikationen fungerade mycket bra och smidigt i jämförelse med till exempel ett stort Excel-system.

Detta projekt krävde inga större investeringar. Företaget hade från tidigare servrar som gick att använda och all teknologi som användes hade väldigt fria licenser. De besparingar som görs tack vare detta projekt kan lättast mätas i tid. Den största tidsboven som elimineras är sökandet genom otaliga Excelfiler på företagets nätverk. Tidigare kunde det vara så svårt att

hitta rätt Excelfil att man måste gå och fråga någon som kan veta mera än en själv. För arbetsledarna sparas tid på grund av att de inte längre behöver gå och fråga var anställda finns eller om någon annan har extra kapacitet i produktionen. Med webbapplikationen finns all den informationen färdigt tillgänglig varenda morgon i realtid.

Som exempel på inbesparing kan Pool funktionen användas. Om man tänker sig att det varje vecka finns fem personer i produktionen som har för lite att göra en av dagarna, kan man visa att de har ledig kapacitet och flytta dem till en avdelning som har för lite, och på så vis spara in fyrtio arbetstimmar per vecka.

Med det nya kvalitetsövervakningssystemet sparas man speciellt mycket tid eftersom det tidigare systemet inkluderade en exceptionell mängd Excelfiler. För Nautor är kvalitet mycket viktigt så ett fungerande system som förbättrar kontroll av kvalitet sparas även pengar i form av lägre produktionskostnader, mindre garantikostnader och nöjda kunder.

Med kvalitetsövervakningssystemet kan man göra mycket stora besparingar. Alla avvikelser som åtgärdas innan jakten levereras blir betydligt billigare att åtgärda i produktionshallen än om man skall flyga ett par montörer till södern för att åtgärda felen under garantitiden.

Med de olika visualiseringarna får man snabbt en överblick av läget och kan ta beslut utan att behöva lägga ner mycket tid på att få fram information.

## **7.1 Diskussion**

Detta examensarbete har varit en lång process. Arbetet inleddes under senhösten och fortskred i jämn takt under vintern. Arbetet krävde att jag satte mig in i komplexa system och processer. Detta var en utmaning men samtidigt väldigt givande.

Arbetet har krävt samarbete med personer på flera olika avdelningar inom företaget och på olika nivåer inom organisationen. Detta har varit mycket intressant för det har visat hur olika perspektiv personer kan ha. Till exempel kan en process innehålla många skeden och personer som jobbar i de olika skedena kan ha varierande information.

En stor utmaning var att göra så många som möjligt nöjda. En ideal applikation skulle uppfylla allas behov, men en sådan applikation skulle bli för komplex för att någon skall

kunna orka använda den i praktiken. Det gällde att hitta en balans som resulterade i en så bra och användarvänlig webbapplikation som möjligt.

Det har varit intressant att se hur mycket man kan åstadkomma utan att behöva göra desto mera investeringar inom detta projekt. Om man kan tänka sig att arbeta med program som kanske inte är de tekniskt bästa som finns och klarar sig utan betala för konsulter och stöd kan man slippa väldigt billigt undan.

Jag upplevde att de jag jobbade med var mycket intresserade, säkert delvis på grund av att det handlade om att utveckla arbetsprocesser de var delaktiga i.

## **7.2 Förslag till fortsatt forskning**

Jag tror det här projektet har varit mycket nyttigt för företaget på det sätt att det visat vilka förbättringar man kan göra med hjälp av bra informationssystem. Det vore helt klart lönt att på allvar börja ta reda på vilka möjligheter företaget har att gå över till ett ERP-system. Det är ingen tvekan om att det skulle bli en lång process, men detta examensarbete har visat att viljan finns inom företaget att ta klivet.

Med tanke på hur smidigt det har varit att använda sig av en webbapplikation kan jag inte låta bli att tro att skrivbordsapplikationer börjar bli föråldrad teknologi.

Det skulle vara intressant att utveckla Nautor – Planner så långt att man kan involvera ännu flera avdelningar, som till exempel ekonomi och lagerhållning. Med mera data skulle man kunna skapa bättre rapporter och hitta samband mellan olika processer.



## 8 Slutord

Att arbeta med detta projekt har verkligen väckt ett starkt intresse inom ämnet och det har varit roligt att jobba med det. Även om det stundvis kändes väldigt utmanande och kraven ibland var höga har personalen vid Nautor varit mycket hjälpsam och välkomnande. Under examensarbetets gång har jag upplevt att jag gjort något nyttigt för företaget, vilket varit en stark motivationsfaktor. Jag vill tacka alla som varit involverade vid Nautor och min handledare vid Yrkeshögskolan Novia.

## Källförteckning

Apache [Online]

<http://www.apache.org/> [hämtat:21.03.2016]

Bootstrap [Online]

<http://getbootstrap.com/> [hämtat:21.03.2016]

CSS [Online]

<https://www.w3.org/Style/CSS/> [hämtat: 21.03.2016]

Cutler, T.R., 2008. Engineer-to-Order Process Defines Challenges of Quality [Online]

<http://www.qualitydigest.com/inside/quality-insider-article/engineer-order-process-defines-challenges-quality.html#> [hämtat: 22.3.2016]

Epstein, J., 2012. What is Engineer to Order? [Online]

<http://technews.tmcnet.com/channels/engineer-to-order/articles/308910-what-engineer-order.htm> [hämtat: 21.03.2016]

Faisal A. Abu Rub, Ayman A. Issa. 2012. A business process modeling-based approach to investigate complex processes: Software development case study. [Online]

<http://www.emeraldinsight.com/doi/full/10.1108/BPMJ-06-2014-0051> [hämtat: 21.03.2016]

Fernando Gustavo Dos Santos Gripe, Ildeberto Aparecido Rodello. 2011. A Theoretical Analysis of Key Points When Choosing Open Source ERP Systems. [Online]

<http://www.jistem.fea.usp.br/index.php/jistem/article/view/10.4301%252FS1807-17752011000200010> [hämtat: 08.02.2016].

Gripe, F. G. D. S., Rodello, I. A., 2011. A theoretical analysis of key points when choosing open source erp systems. [Online]

<http://www.scielo.br/pdf/jistm/v8n2/v8n2a10.pdf> [hämtat: 21.03.2016]

Hanawalt, D., 2015. 4 Valuable Lessons For Selecting And Implementing AnERP System [Online]

<http://search.proquest.com/docview/1714334981> [hämtat: 02.03.2016].

Hongjiang, X., Horn Nord, J., Brown, N., Nord, G.D. 2002. [Online]  
Data quality issues in implementing an ERP.

<http://search.proquest.com/docview/234924773> [hämtat: 21.03.2016]

HTML [Online]

<https://www.w3.org/html/> [hämtat:21.03.2016]

JavaScript [Online]

<https://developer.mozilla.org/en-US/docs/Web/JavaScript> [hämtat:21.03.2016]

JQuery [Online]

<http://learn.jquery.com/about-jquery/> [hämtat:21.03.2016]

Mathrani, S., Mathrani, A., Viehland, D., 2013. Using enterprise systems to realize digital business strategies [Online]

<http://www.emeraldinsight.com/doi/full/10.1108/JEIM-01-2012-0003>

[hämtat: 09.02.2016]

Maydanchik, A., 2007. Data Quality Assessment. USA: Technincs Publications New Jersey

MySQL [Online]

<http://www.oracle.com/us/products/mysql/overview/index.html> [hämtat:21.03.2016]

Nautor's Swan Press Kit [Online]

<http://us6.campaign-archive1.com/?u=859cfe5aaf7b87ca98830b017&id=2049c5fa20>  
[hämtat: 06.02.2016]

Nikolaos, A., Panayiotou Sotiris, P., Gayialis Nikolaos, P., Evangelopoulos Petros, K., Katimertzoglou, 2015. A business process modeling-enabled requirements engineering framework for ERP implementation. [Online]

<http://www.emeraldinsight.com/doi/full/10.1108/BPMJ-06-2014-0051>  
[hämtat: 07.01.2016].

Perks, M., 2006. Best Practices of software development projects. [Online]

[http://www.ibm.com/developerworks/websphere/library/techarticles/0306\\_perks/perks2.html](http://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html) [hämtat: 22.3.216]

PHP [Online]

<https://www.php.net/> [hämtat:21.03.2016]

PhpMyAdmin [Online]

<https://www.phpmyadmin.net/> [hämtat:21.03.2016]

Rouse, M., 2010. Learn IT: Software development. [Online]

<http://whatis.techtarget.com/reference/Learn-IT-Software-development>  
[hämtat: 21.03.2016]

Syspro [Online]

<https://www.syspro.com/product/what-is-erp> [hämtat: 21.02.2016]

Ying, X., Colin, J., Mahmood, A. 2014. An integrated decision support system for ERP implementation in small and medium sized enterprises. [Online]

<http://www.emeraldinsight.com/doi/full/10.1108/JEIM-10-2012-0077>

[hämtat: 21.03.2016]