



Designa webbapplikation

Översikt

- Planering.
- Planering för design.
- Design av Models, Controllers och Views.

.eerec

Lektion 1: Planering

- Projektmetodiker.
- Vattenfallsmodellen.
- Agile Development Model.
- Test-Driven Development.
- Unified Modeling Language.
- Inhämta krav.
- Funktionella krav.
- Tekniska krav.
- User stories.
- Att skriva user stories.
- Att skriva user stories –med acceptanskriterier.

.eerec

Projektmetodiker

Utvecklingsmodell	Beskrivning
Vattenfallsmodellen	Aktiviteter för att bygga en application utförs i sekvens i distinkta faser med klara leveransbitar.
Iterative Development Model	Applikation byggs i delar som sammanfogas, genom att använda arbetsversioner, dessa är testade tills de blir den slutliga programkoden.
Prototype Model	Baseras på några få krav från organisationen. Feedback för prototype används för som grund för att utveckla den slutliga produkten.
Agile Development Model	Applikation byggs i snabba distinkta cyklar, dessa integrerar eventuella förändringar och eventuella krav som förändras under processen.
Extreme Programming	Börjar med lösa ett fåtal kritiska uppgifter. Utvecklare testar den förnekade lösningen och får synpunkter från projektägare. Därefter tas en lösning fram efter mer detaljerade krav.
Test-Driven Development	Ett testprojekt skapas, mot detta testprojekt testas förändringar, genom hela projektet.
Unified Modeling Language	UML-diagram används för planering och dokumentering genom hela projektet.

.earec

Vattenfallsmodellen

- Modellen bygger på ett logiskt och sekventiellt genomförande av projektet.
- Varje steg i processen skall avslutas innan nästa påbörjas.
- Nyckelord är kontroll, planering, processer och uppföljning.
- Till skillnad från agila projekt, anses ändringar som en risk för projektet.
- Leverans av slutprodukten sker i slutet av projektet, efter tester av både leverantören och kunden.

.earec

Lång tid, många problem - vattenfallsmodellen

- Normalt tar ett vattenfallsprojekt 12-20 månader för att gå från efterfrågad funktionalitet till leverans.
- Detta skapar många problem, t ex kan kravmängden växa, liksom att det är en högre sannolikhet att något förändras under tiden.
- Dokumentation i ett projekt av denna typen tenderar att bli omfattande, primärt för att det är det främsta sättet att kommunicera mellan varje steg i processen.

.earec

.....

.....

Fördelar -vattenfallsmodellen

- Fördelen med metodiken är att i de projekt där slutprodukten är känd, vet leverantören hur produkten skall konstrueras och inga förändringar sker under tiden.
- Då leverantören har erfarenhet hur lång tid varje steg i processen tar, går det att detaljplanera, optimera processer, utföra enligt specifikationen – helt enkelt ett snabbt och effektivt genomförande.

.eerec

Bäst för -vattenfallsmodellen

- Metodiken är bäst för leveranser av standardlösningar med en låg del konfiguration.
- IT-projekt är däremot oftast komplexa projekt, med en hög förändringsbenägenhet, unika lösningar där slutprodukten inledningsvis inte är känd, varkens för beställaren eller leverantören.
- IT-projekt är ofta ett lärande projekt, där produkten växer fram genom kommunikation och reflektion.

.eerec

Process -vattenfallsmodellen

- Projektet kan utformas med lite olika processteg och definitioner, men grunden bygger på att varje steg ska avslutas och granskas innan nästa påbörjas.
- Skapar en säker och trygg process där få misstag görs.
- För ett IT-projekt omfattar stegen i regel kravbearbetning, analys och design, utveckling, test, acceptanstester och slutlig leverans.
- Om någon ändring läggs till i projektet, behöver samtliga steg genomföras igen.

.eerec

Agile Development Model

- Samlingsnamn för ett antal metoder.
- De olika metoderna följer gemensamma värderingar, principer och synsätt, dessa finns beskrivna i ett manifest.
- Agile flyttar fokus på hur traditionella systemutvecklingsprojekt bedrivs till att fokusera på människor, kommunikation, samarbete med kunden och en fungerande produkt, framför en tung dokumentation.

.carec

Syfte och struktur -Agile Development Model

- Inom Agile finns det ett antal olika metodiker, dessa beskriver hur projektet kan arbeta agilt.
- Vanligaste metodiken i dag är Scrum, men även Extrem Programmering, DSDM och Kanban används.
- Metodikerna har mycket gemensamt och används oftast tillsammans för att skapa de bästa möjliga förutsättningarna för projektet.

eng:Agile = smidig, vög, lättroilig

.carec

Centrala principer -Agile Development Model

- Centrala principer inom Agile:
 - Transparans.
 - Iterativ och inkrementiellt.
 - Prioritera och fokusera.
 - Kontinuerlig förbättring.
 - Ett team med rätt förutsättningar.
 - Kommunikation och samarbete.
 - Enkla verktyg.
 - Hög kvalitet.

<http://agilemanifesto.org/principles.html>

.carec

Metodiker -Agile Development Model

- Lean, Scrum, Kanban, Extrem programmering (XP) och Dynamic System Development Method (DSDM).

.eeec

Test-Driven Development

- Test-Driven Development =TDD, är en viktig del inom agil utveckling.
- Introducerades 2002 av Kent Beck, accepterad och rekommenderad teknik vid programmering, dock med förbehåll.
- TDD förespråkar att efter initial planering systemdesing, kravställning etc, börjar skriva testfall, före implementering.

.eeec

Kod för testing skrivs först -Test-Driven Development

- Kod för testning skrivs innan kod för applikationen.
- Kod som skrivs för test, läggs i separata filer.
- När programkod börjar utvecklas, testas utveckling mot testkod.
- Om något går fel, får vi oftast tips om vart det har gått fel.

```
test@ubuntu:~/test1$ ls
test.php
test@ubuntu:~/test1$ phpunit --color Test.php
PHPUnit 3.5.0 by Sebastian Bergmann and contributors.

F
Time: 07.44, Memory: 4.00M

There was 1 failure:
1) setUpMethod: setUpMethodFailed
Failed asserting that 1 matches expected 0.

FAILURES!
Tests: 1, Assertions: 1, Failures: 1
test@ubuntu:~/test1$
```

.eeec

Regelverk -Test-Driven Development

- I TDD används unit-tests, dessa används för att undersöka en mindre isolerad del i systemet.
- Regelverk:
 - du får inte skriva produktionskod, om det inte finns ett fallerande test som garanterar detta.
 - i ett unit-test, får det inte vara mer programkod än nödvändigt, för att få den att falla.
 - du får inte skriva mer produktionskod, än vad som är nödvändigt för att få ett test som har fallerat skall gå igenom.

.eerec

Unified Modeling Language

- Unified Modeling Language (UML) är en industristandard för att visa design för applikation som är baserat på objektorienterad teknologi.
- Är ingen utvecklingsmodell, utan används oftast för planering och dokumentation.
- Ett antal diagram skapas:
 - Diagram som beskriver beteende.
 - Diagram som beskriver interagering.
 - Diagram som beskriver struktur.

.eerec

Inhämta krav

- När du tilldelas ett projekt, behöver du skapa dig en vision av lösningen.
- Vision är oftast vag och kräver en djupare undersökningen, innan du kan lägga till detaljer och försäkra dig om att din lösningen uppfyller de krav som ställs från uppdragsgivaren.
- Insamling av information görs oftast genom intervjuer av olika grupper.
- Krav från uppdragsgivaren delas in i två typer:
 - Funktionella krav.
 - Tekniska krav.

.eerec

Funktionella krav

- Beskriver hur applikationen skall fungera, kallas oftast för beteende krav.
- Inkluderar:
 - Krav för användaregränssnitt.
 - Hur användare kan arbeta med applikationen.
 - Krav från organisationen.

.carec

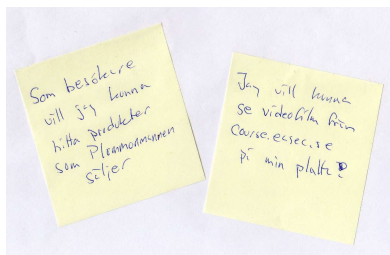
Tekniska krav

- Beskriver tekniska funktioner för applikationen, t ex tillgänglighet, säkerhet och prestanda.
- Kallas även för icke funktionella krav.

.carec

User stories

- Beskriver vad användare vill göra, används för identifiera funktionella krav för din applikation.



.carec

Att skriva user stories

- User stories skrivs enligt:
 - Som <roll> vill jag <mål/önskan/händelse>.
 - Som mobilanvändare vill jag se en responsiv version av webbplatsen.
 - Som <roll> vill jag <mål/önskan/händelse> för att <syfte>.
 - Som nyfiken besökare vill jag hitta en intressant utställning för att lära mig mer.
 - Som mobilanvändare vill jag få en responsiv version av webbplatsen för att den skall ladda snabbare och vara anpassad då man kan använda den i mobilen.

.eeec

Att skriva user stories – med acceptanskriterier

- User stories skrivs enligt:
 - Som besökare vill jag kunna se de senaste twittringarna från företaget direkt på startsidan. Denna funktion är viktig eftersom den gör att jag kan få all kommunikation från företaget samlad på en plats och inte missar något.
- Acceptanskriterier:
 - Webbplatsens startside visar de 3 senaste tweetsen.
 - Tweetsen visas inom 15 minuter från att de twittrades.
 - Om en tweet raderas ska den inte visas på webbplatsen.
 - Länkar i tweetsen ska fungera.
- Förslag till lösning:
 - Använd modulen Twitter_db.
 - Sätt cron job att köras var 15:e minut.
- Kommentarer(er)
 - Lägg även till en knapp "Följ oss på Twitter" i anslutning till tweetsen.

.eeec

Lektion 2: Planering för design

- Planering för design av databas.
- UML Domain Model.
- Logical Data Model (LDM).
- Struktur för fysisk databas.
- Kommunikation med DBA.
- Design av databas i Agile utveckling.
- Planera för distribuerade applikationer.
- Planera för hantering av state.
- Planera för olika språk.

.eeec

Planering för design av databas

- När du har en bra uppfattning om de funktionella kraven och de tekniska kraven för den föreslagna webbapplikationen, så är det dags att planera själva implementeringen.
- Även om inte alla webbapplikationer använder databas, så gör en majoritet av dessa detta.
- Design påbörjas på högre nivå med att ta fram diagram för UML Domain Model och Logical Data Model (LDM).



UML Domain Model

- UML Domain Model eller konceptuell modell, visar konceptet för de objekt som webbapplikationen skall hantera.
- Inkluderar inga detaljer om egenskaper för de olika delkomponenterna, men visar på relationer mellan dessa.
- UML Domain Model används för att diskutera med uppdragsgivare.



Logical Data Model (LDM)

- Logical Data Model (LDM) är som UML Domain Model, men med alla detaljer inkluderat.
- Exempelvis:
 - Datatyper.
 - Egenskaper.



Struktur för fysisk databas

- I din projektplan, bör följande finnas med:
 - Tabeller, är fundamentet för lagring av information. Här skall du definierar kolumner, datatyper, primärnyckel och eventuella relationer.
 - Views, det sättet som information presenteras från databasen, t ex att information från två tabeller sammanförs med join.
 - Stored procedures, är en sekvens av operationer i databasen som har definieras. Några kan vara komplexa och utföra avancerade operationer.
 - Säkerhet, du måste definierar hur webbapplikationen skall autentiseras mot databasserver och hur tillgång skall ske till varje ingående tabell.

.eeec

Kommunikation med DBA

- I större organisationer, hanteras SQL av Database Administrators.
- Viktigt att ha bra dialog med dessa, DBA definierar eventuella krav som finns för att kunna nyttja databas.
- Om du bygger en webbapplikation, så här DBA:er ansvariga för att skapa databas på rätt server eller servrar om kluster används.
- De är även ansvariga för att tilldela rättigheter.

.eeec

Design av databas i Agile utveckling

- I Agile utveckling är karaktärsdragen för metoden, väldigt liten planering och dokumentering.
- Kunskap om eventuella krav förändras under arbetets gång.
- UML Domain Model kommer att skapas en gång, detta kommer att ske i initialskedet.
- Logical Data Model kommer att skrivas först när de behövs, när du har börjat att skriva programkod.
- Under utvecklingen sker diskussion med DBA:er.

.eeec

Databas design i Agile utveckling (forts.)

- I Agile utveckling kommer design av databas vanligtvis att förändras.
- Utvecklare bör kunna förändra databas, utan att behöva konsultera DBA:er.
- För detta syfte, bör du ha SQL Databas på utvecklingsdatorn.
- I några fall arbetar du med existerande databas, arbeta med en kopia av databas på en isolerad utvecklingsmaskin.

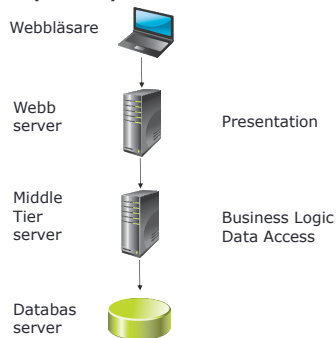


Planera för distribuerade applikationer

- För mindre applikationer med liten trafik, väljs oftast en värd för alla komponenterna.
- Om applikationen växer, är det vanligast att använda distribuerad konfiguration.
- Här används en skiktbaserad arkitektur, som bestå av:
 - Presentation layer.
 - Business logical layer.
 - Data access layer.
 - Database layer.



Planera för distribuerade applikationer (forts.)



Planera för hantering av state

- Lagring på klientsidan:
 - Cookies.
 - Query strings, är en del av URL efter ?, används oftast för att skicka över värden eller annan information till server. Alla webbläsare har stöd för detta, några dock med begränsningar. Ingen känslig information skall skickas med funktionen.
- Lagring på serversidan:
 - TempData.
 - Application State.
 - Session State.
 - Profile Properties.
 - Tabell i databas.

.carec

Planera för olika språk

- Internet är ett internationellt nätverk.
- Planera för vilket eller vilka språk som skall användas.
- Det finns koder som definierar detta, delas upp i två delar:
 - Språk, exempelvis en för engelska.
 - Region, specificerar regionala skillnader, exempelvis hur datum skriv.
 - en-US eller en-UK, datumformat mm/dd/åå respektive dd/mm/åå.
- Olika tekniker, resursfil eller separata views.

.carec

Lektion 3: Design av Models, Controllers och Views

- Design av Models.
- Entity Framework.
- Design av controllers.
- Design av views.
- Skapa Wireframes.

.carec

Design av Models

- En viktig del i designprocess för MVC är design av Model.
- Varje modelklass representerar objekt som din applikation hanterar.
- Först måste struktur och design för model göras, innan planering av controller och views startas.
- Utgå från user stories, för att ta reda på vilka klasser som skall finnas och vad dessa klasser hanterar.

.eerc

Design av Models (forts.)

- Domain Model och Logical Data Model, kan användas för planering av struktur.
- Ta reda på vilka relationer som finns.

.eerc

Exempel



.eerc

Entity Framework

- Entity Framework är ett Object Relational Mapping (ORM) ramverk.
- När detta ramverk används i din MVC-applikation kommer länkar att skapas för de klasser du använder.
- Hur skall databas skapas:
 - Database-First.
 - Model-First.
 - Code-First.

.eeec

Design av controllers

- Controllers implementerar logik för indata.
- De tar emot indata i form av HTTP requests och väljer både model och view för att besvara denna begäran.
- I MVC finns det vanligtvis bara en controller per model.
- Varje controller kan hantera flera händelser.

.eeec

Design av views

- Gränssnitt för användare är en vital komponent för alla system, detta på grund av att det är den komponent som presenterar din applikation för slutanvändare.
- Det finns en till flera relation mellan controller och views.
- Exempelvis:
 - När användare vill se information om en produkt, hanterar view visningen av denna.
 - När användare vill se flera produkter, kommer en annan view att visa dessa produkter.

.eeec

Design av views (forts.)

- Några saker skall visas på samtliga sidor, exempelvis logotyp och navigering – använd template view.
- Någon del dyker upp på många sidor – använd partial view.



Skapa Wireframes

- Wireframes används för att skissa viktiga delar av användaregränssnittet.
- Använd inte detta för alla views.
- Wireframes används för att skissa för t ex beställare funktionen för de olika delarna i användaregränssnittet.
- Innehåller ingen grafik, färger eller logotyper.



Övning Design av MVC-applikation



Repetitionsfrågor

.c@ce
