

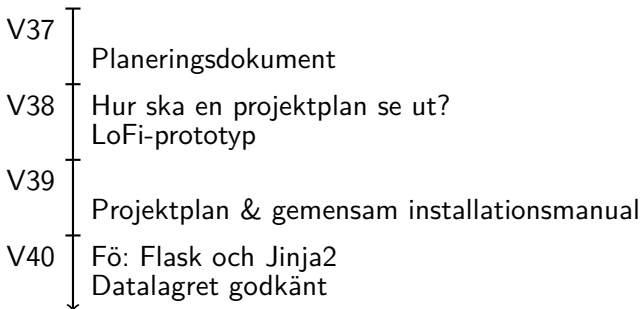
TDP003

Föreläsning 2

Filip Strömbäck

- 1 Kursinformation
- 2 Projektplan
- 3 Frågor
- 4 Genomgång av projekt
- 5 Vad är ett API?
- 6 Kom ihåg

Vad händer härnäst?



Kom ihåg

- Webreg: skriv upp er
- Skapa repository i GitLab

- Dagbok: mjukvara
 - Emacs
 - Rednotebook
 - GNOME Tomboy
 - ScribeFire Next 4.0

Leverabler

- Förberedelse: undersök existerande system och fundera på hur ni vill ha det
 - Leverabler: skisser på systemet (LoFi-prototyp), projektplan
- Konstruera: design och implementation av systemet
 - Två delsystem ska byggas: Presentation och data
 - Leverabler: fungerande system och dokumentation
- Överlämna: färdigställande och inlämning av projektet
- Uppföljning: reflektion, testning av varandras system, utvärdering av hur projektet gått

LoFi-prototyp

- Systemskiss
- Pappersprototyp
- Rita, klippa, klistra
- Rita upp på papper hur ert system ska se ut
- Torrkör med varandra och se om ni har missat något

Individuell reflektion

- Egen programmerardagbok
 - Vanligt textdokument – snyggt och prydligt med namn och datum etc.
 - Blogg – valfri blogg med bra stöd för uppslagning via datum
- En skriftlig reflektion görs baserat på dagbok och Code Complete
 - Läs Code Complete, Part I, Part II och Part IV (gärna Part V, Part VI och Part VII också)
- Välj ut teman i Code Complete som du känner ni kommit i kontakt med i projektet eller som intresserar dig och arbeta med dem extra noga

Programmerardagbok

Varje dag som arbete görs i projektet ska det införas i en personlig programmerardagbok. Innehållet ska åtminstone kortfattat tala om vad du arbetat med, vilka problem/intressanta saker man stött på, hur detaljer i samarbetet fungerat i paret, om du kört fast: på vad?

Veckovis sammanfattar du hur du själv tycker du arbetat – är du nöjd med din insats? Vad har du gjort över förväntan? Vad har du gjort under förväntan?

Exempel

2012-09-10

Jobbat med refactoring av datalagret och namngivning av interna variabler enligt Code Complete kapitel 11. Koden blev mer läsbar med det tog lite onödigt lång tid. Borde ha gjort rätt från början. Hittade även en bugg i vår sökningsalgoritm. Måste fixa den snarast. Buggen är att det inte går att söka på åäö, behöver felsöka lite mer.

- 1 Kursinformation
- 2 **Projektplan**
- 3 Frågor
- 4 Genomgång av projekt
- 5 Vad är ett API?
- 6 Kom ihåg

Vitsen med en projektplan

- Beskriver *hur* resultatet ska uppnås
 - Hur ska ni samarbeta
 - Vilka verktyg ska användas
 - etc.

Vitsen med en projektplan

- Beskriver *hur* resultatet ska uppnås
 - Hur ska ni samarbeta
 - Vilka verktyg ska användas
 - etc.
- Överenskommelse mellan er och kunden

Vitsen med en projektplan

- Beskriver *hur* resultatet ska uppnås
 - Hur ska ni samarbeta
 - Vilka verktyg ska användas
 - etc.
- Överenskommelse mellan er och kunden
 - ...samt inom gruppen

Projektplan

En projektplan ska (åtminstone):

- Vara nedskriven i förväg
- Beskriva:
 - *Vad* som ska uträttas
 - *När* det ska uträttas
 - *Hur* det ska uträttas
 - Men *ingen* design
- Vara välskriven, strukturerad, kortfattad och begriplig
- Ta hänsyn till tänkbara olyckshändelser
- Utvecklas i takt med projektet (ett levande dokument)

En projektplan kan innehålla

- Översikt
Intro till arbetet, kunden, gruppens organisation
- **Fasplan/Tidsplan** - obligatorisk
Vilka utvecklingsfaser, vilka produkter, vilka datum?
- Organisationsplan
Vilka team, vems ansvar?
- Testplan
Vem? Hur? Verktyg?
- **Hantering av artefakter**
Var och hur ska källkod och dokument lagras?
- Dokumentationsplan
Vilka, när, till vem? Vem godkänner?

En projektplan kan innehålla

- Utbildningsplan
Intern, extern. Vem, när, resurser?
- Plan för rapportering och granskningar
Vad, till vem, när?
- **Installationsplan**
Vilken procedur krävs för att komma igång?
- Plan för kvalitetssäkring
Standarder som ska användas?
- **Varuplan**
Vad ska levereras, när? Delleveranser?
- Resursplan
Persontid, datortid. Summering av milstolpar!

Milstolpar

Milstolpe

Projektplan inlämnad

Projektplan godkänd

LoFi-prototyp klar

load och get_project_count fungerar

Datalagret klart

Datalagret godkänt av assistent

Vi kan visa en enkel sida med Flask och Jinja2

Projektsidan fungerar

Datum

2017-09-10

2017-09-14

2017-09-15

2017-09-16

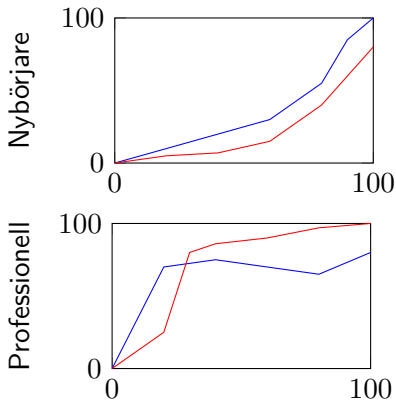
2017-09-20

2017-09-22

2017-10-01

2017-10-05

Tidsplanering



- Gör en tidsplan
 - Ligger ni efter, öka takten!
- Arbeta i början
- Dela upp problemet
- Sikta på professionell kurva

Arbetsinsats Leverabler

- 1 Kursinformation
- 2 Projektplan
- 3 Frågor**
- 4 Genomgång av projekt
- 5 Vad är ett API?
- 6 Kom ihåg

- 1 Kursinformation
- 2 Projektplan
- 3 Frågor
- 4 **Genomgång av projekt**
- 5 Vad är ett API?
- 6 Kom ihåg

Portfolioprojekt

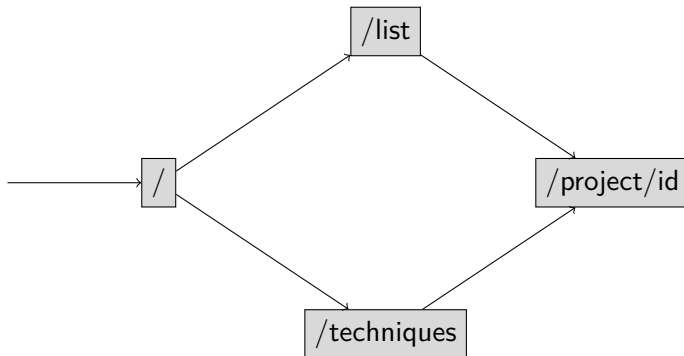
- Ett webbaserat system för att visa upp gjorda projekt
 - Tänk: Sourceforge/Github utan verktyg
- Listar de projekt ni har gjort och egna projektsidor med information om projektet
- Ett verktyg åt er själva
- Ni är själva beställare av systemet
- Ni och andra som ni visar era projekt är användare
- Kursledningen är projektledare som dragit upp riktlinjer för arbetsordning (som ni har att acceptera som givna krav)
- Inom givna ramar har ni fria händer

Idé

- Vi har en samling av projekt som vi vill visa
- Varje projekt har ett namn, en beskrivning, och en samling tekniker
- Vi vill kunna hitta alla projekt genom bläddra, söka
- Vi vill också kunna hitta alla projekt som involverar en viss teknik


Systemskiss

Fyra websidor med URL:er:




Startsidan

URL: `http://foo/`



Om mig:
~~~~~  
~~~~~

Senaste:
~~~~~






[Alla projekt](#)      [Alla tekniker](#)



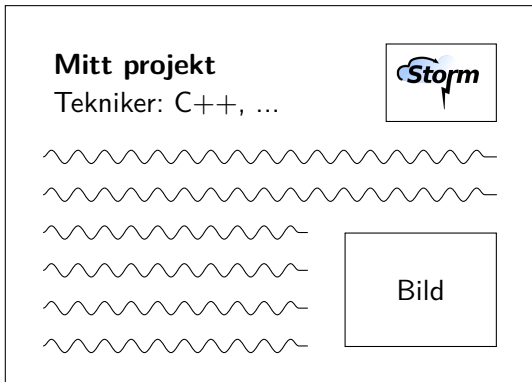
# Listsidan

**URL:** `http://foo/list`

| Sök: |                                                                                    |
|------|------------------------------------------------------------------------------------|
| A    |  |
| B    |  |
| C    |  |

# Projektsidan

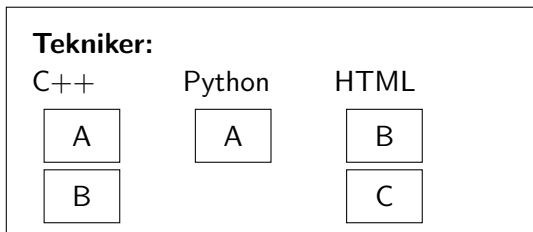
**URL:** `http://foo/project/project-id`



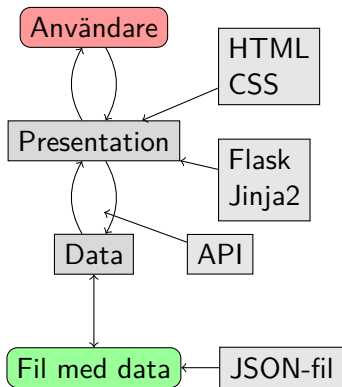
## Tekniksidan

**URL:** `http://foo/techniques`

- Lista över era projekt, baserat på använda tekniker
- Helt OK att lägga till en extra sida (exempelvis `http://foo/techniques/html`) som visar alla projekt som innehåller en viss teknik



# Arkitektur



- Två delsystem med olika ansvar
  - Presentation: användarhändelser
  - Data: datahantering
- Standardiserat källkodsgränssnitt (API) till delsystemet data
- Vi tillhandahåller test och testdata

- 1 Kursinformation
- 2 Projektplan
- 3 Frågor
- 4 Genomgång av projekt
- 5 **Vad är ett API?**
- 6 Kom ihåg

# API (Application Programming Interface)

- Strikt överenskommelse mellan moduler
- Att använda ett API är ett viktigt sätt att hantera komplexitet och isolera olika delar för att få dem utbytbara.
- Datamodulen har ett API bestående av 6 välspecifierade funktioner som presentationsmodulen får använda för att komma åt data.
- Så länge API:t (specifikationen av funktionerna) inte ändras kan vilken presentationsmodul användas på vilken datamodul som helst.

## “API” i en bil

- En bil har ett “API” bestående av gas, broms, koppling, växelspak och ratt
- Så länge API:t inte ändras kan vi i en given bil stoppa in vilken legitimerad förare som helst, och den kan hantera bilen
- Så länge API:t inte ändras kan en given legitimerad förare köra vilken bilmodell som helst

## Dyrt att ändra API

- När ett API väl är satt och börjat användas är det dyrt att ändra
- Bygger vi en bil med annat API (automatväxel, joystick) så måste alla förare uppdateras innan de kan använda bilen
- Utbildar vi alla förare för automatväxel måste alla manuellt växlade bilar uppdateras för att kunna köras
- En ännu mer grundläggande överenskommelse i trafiksystemet är att alla vägar, bilar och förare i Sverige skall använda högertrafik. Varför byter inte Storbritannien till högertrafik?



## Windows-API

- Microsoft Windows har ett API som alla windowsprogram använder för att visa sina fönster
- Så länge en ny version av Windows behåller samma API kommer gamla program att fortsätta fungera (bakåtkompatibilitet).
- Om en ny version av Windows byter API måste alla gamla program skrivas om!

## API till datalagret

- `load(filename)`  
Läser en JSON-fil, returnerar ett databasobjekt.
- `get_project_count(db)`  
Hämtar antalet projekt i databasobjektet.
- `get_project(db, id)`  
Hämtar projektet *id* från databasobjektet.
- `get_techniques(db)`  
Hämtar en sorterad lista med alla tekniker.
- `get_technique_stats(db)`  
Hämtar statistik över de tekniker som används.

## Sök i datalagret

```
search( db,  
        sort_by='start_date',  
        sort_order='desc',  
        techniques=None,  
        search=None,  
        search_fields=None)
```

- 1 Kursinformation
- 2 Projektplan
- 3 Frågor
- 4 Genomgång av projekt
- 5 Vad är ett API?
- 6 **Kom ihåg**

# Kom ihåg

- Anmäl er i Webreg!
- Börja med planeringsdokument/projektplan
- Skissa på er LoFi-prototyp

Filip Strömbäck

[www.liu.se](http://www.liu.se)