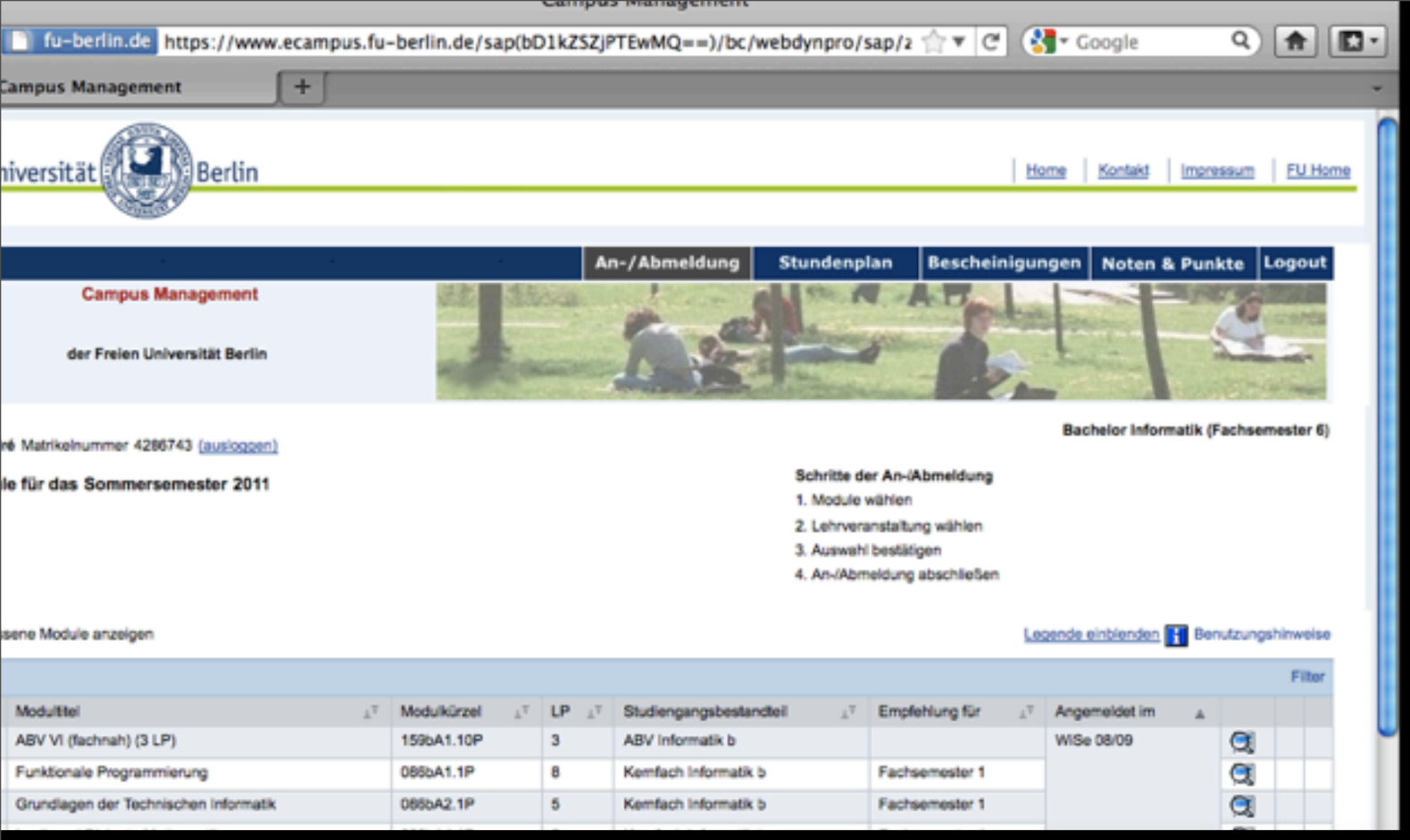
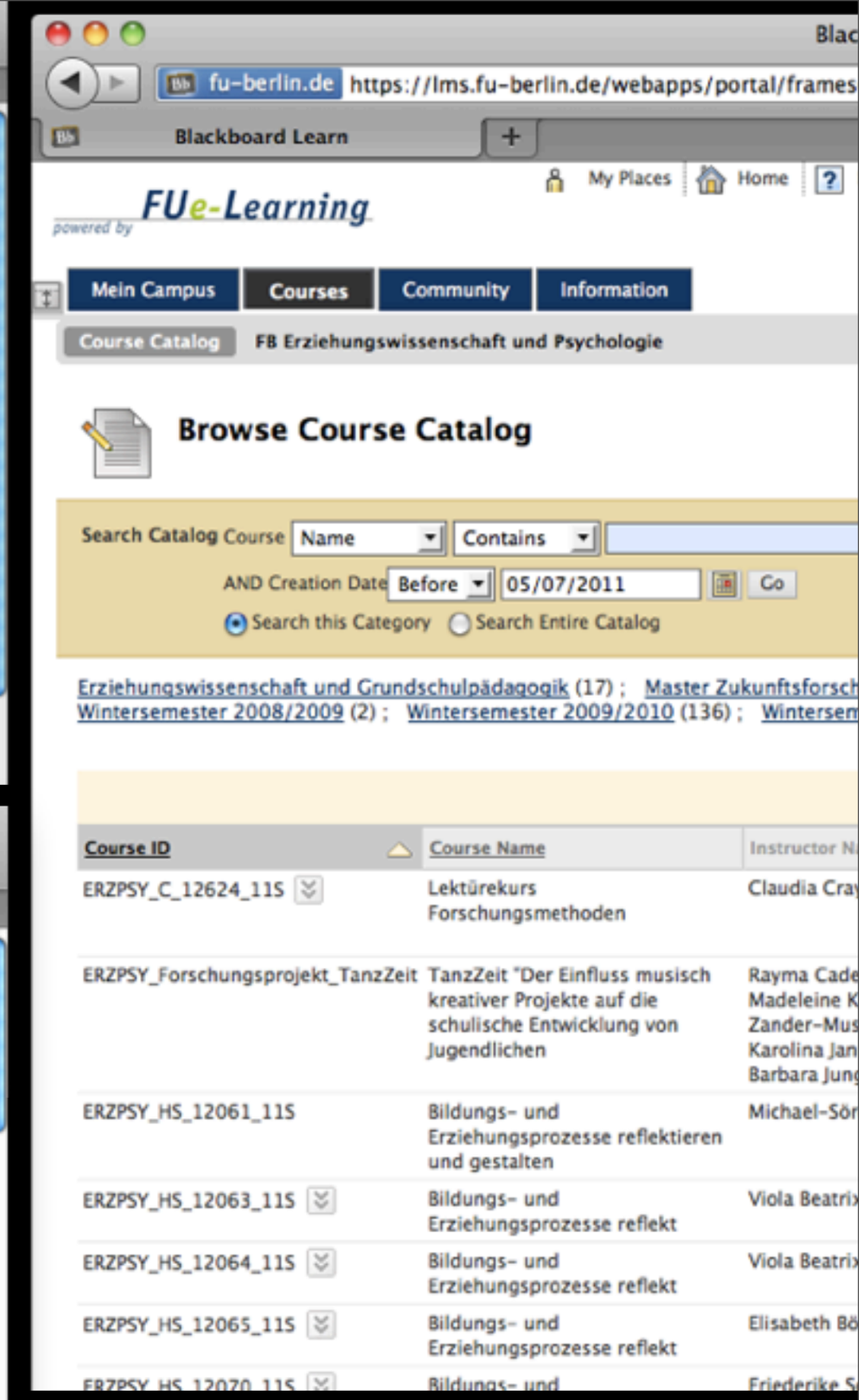
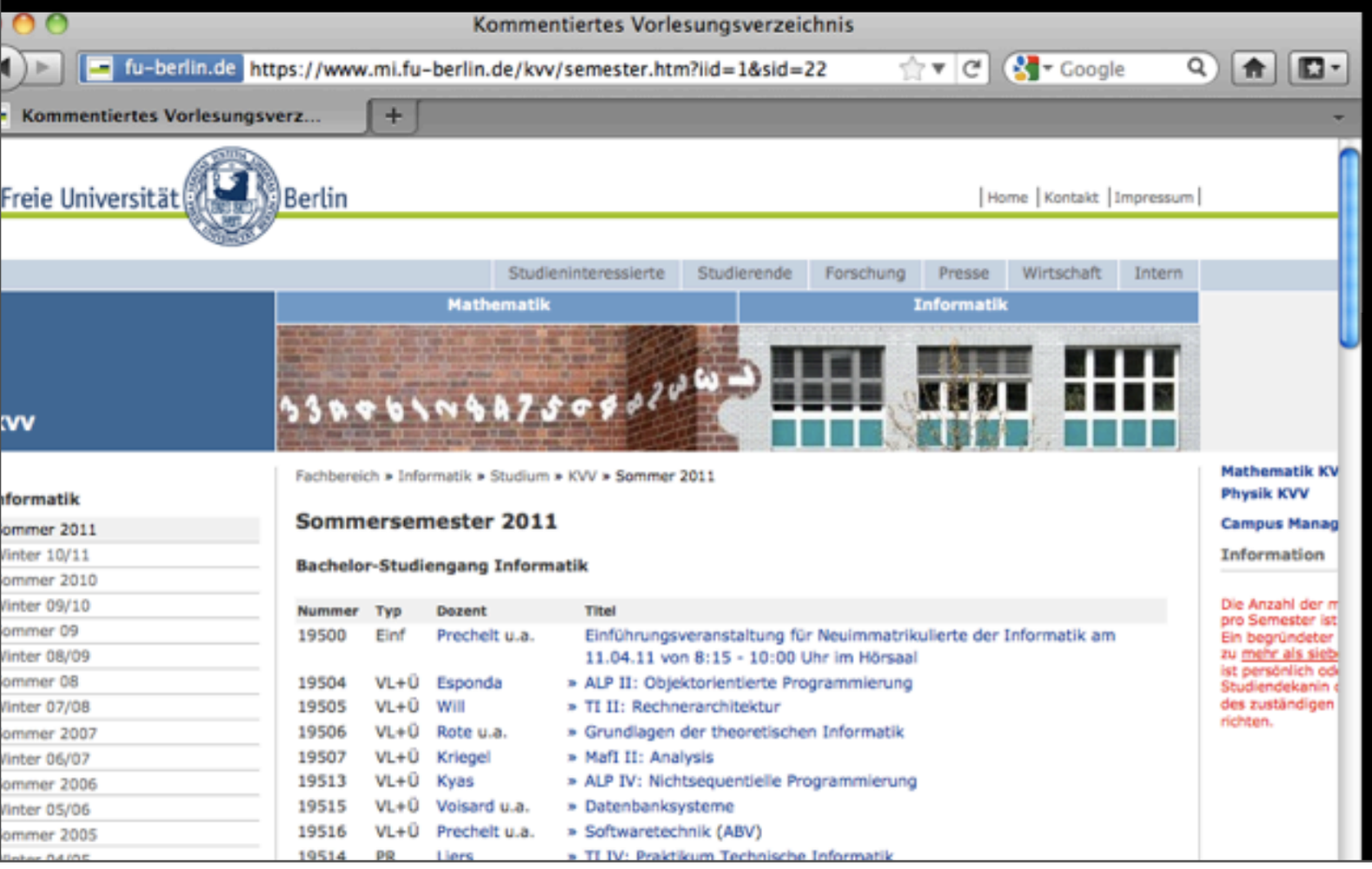
Freitag, 6. Mai 2011







Freitag, 6. Mai 2011

(5-A-Regel)
zunächst Titelfolie direkt sondern sofort rein ins Geschehen
-> Kontext herstellen mit was wir momentan zu tun an unserer Uni
-> Wir: ”hey, dass können wir doch auch/besser ?“

Das ist unsere Motivationsfolie:
Warum haben wir das überhaupt gemacht ?

Scetris

Score Contest 2011 - Project MyCourses

David Bialik, Julian Fleischer, Hagen Mahnke, Konrad Reiche, André Zoufahl

Freitag, 6. Mai 2011

Frage: Textüberschrift oder Bildüberschrift von ‚Scetris‘ ?

Greeting, fellow followers!

* My name is <...>
  - me and konrad (point to) are going to show you the results of our p.
* Team Scetris:
- David Bialik
- Hagen Mahnke (not here)
- Konrad Reiche
- André Zoufahl

* first software project

MyCourses
A Course Scheduling System

- web-based
- resource management
- collaborative & computer-aided course scheduling

Andre: quick overview was mycourses war, was es tun sollte

Freitag, 6. Mai 2011

This is a screen shot of our application.

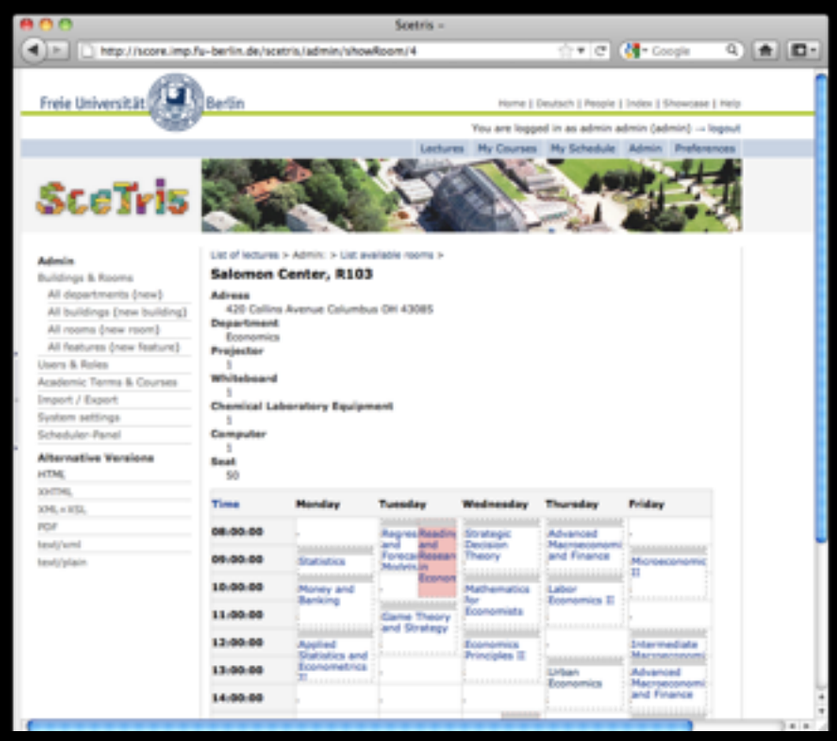
* web-based
-> provides means for (courses, students, bla)

(A) initial requirements
* strong focus on usability
* collaborative *and* automatic/semi-autom. course scheduling

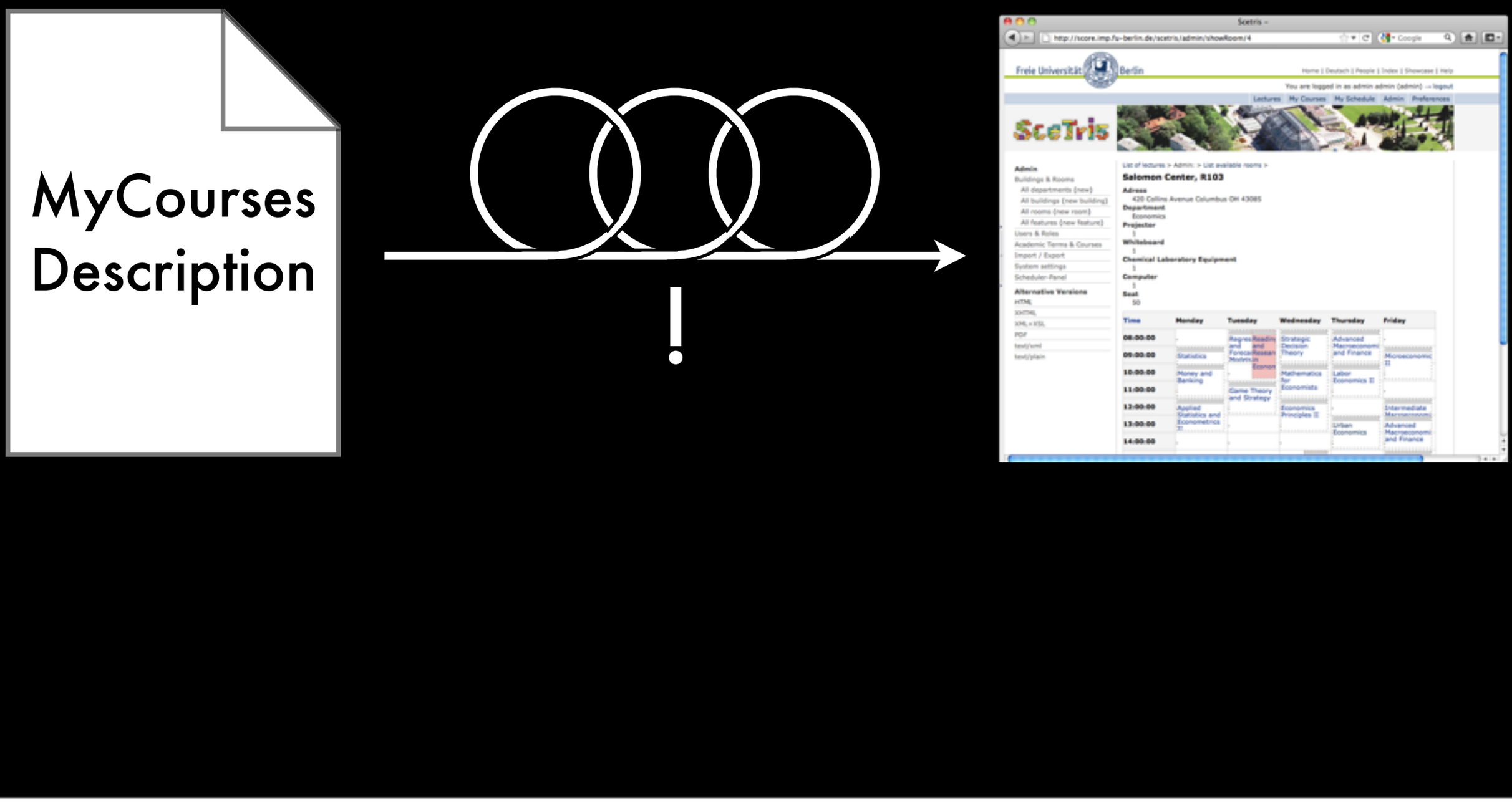
(B) first software project
* confronted with many issues:
  - configuration-, build-, source code management
  - methodology
  - technical issues
    + course scheduling? -> genetic algorithm
    + database access (-> active record)
    + tool-use
    + how to design a web-application (-> 3-tier)
  - new technologies:
    + web-technologies (xml, html, css)

MyCourses Description

?



Andre: wie kamen wir also von diese spec zu unserem fertigen/derzeitigen Produkt ?

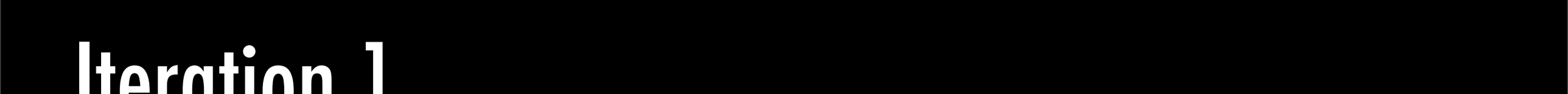


Freitag, 6. Mai 2011

losely based on spiralmodel -> work in iterations

Gedanken Andre: durch die iterationen nun wirklich schnell durch, so dass wir evtl. zum schluss noch zeit haben, wirklich eine conclusion haben/ zeit für ein, zwei noch wichtige dinge haben, die mehr emergent waren (z.b. hours devoted to project, essen bei den treffen, you know ?)

Iteration 1



Freitag, 6. Mai 2011

Andre: mein vorschlag für eine zeitleiste, die nicht zu sehr ablenken soll,
im weiteren sind auch die überschriften nicht mehr auf selber breite, sondern wandern auch mit der zeit (siehe unten)

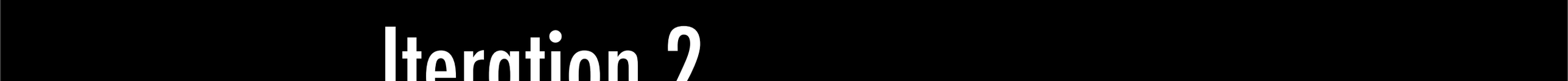
requirements elicitation
hagen hats gemacht > es wird mehr arbeit als gedacht
überblick verschaffen über scheduling, c-ci-ce-cei, programs

alle waren auf generischer algo-> eines tages:
uns fehlt eine manuelle schnittstelle

BSP: sequenzdiagramm von hagen mit man-machine solution

June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

Iteration 2

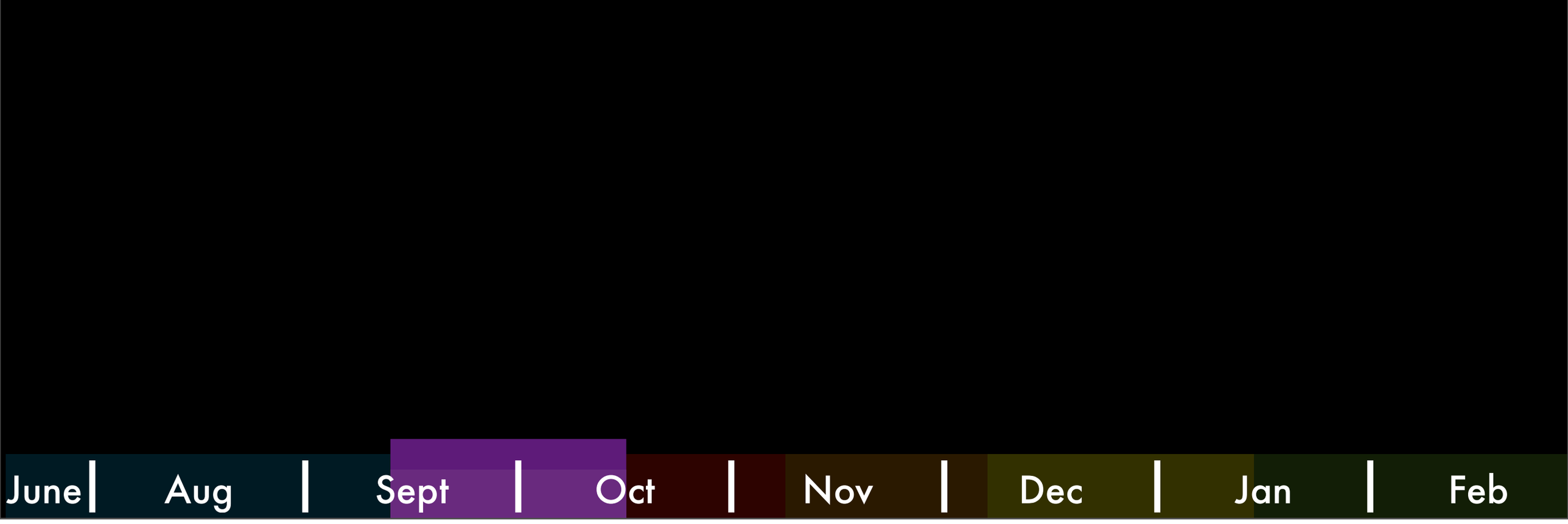


Freitag, 6. Mai 2011

mit java und php angefangen aus einem dokument
codegenerator angefangen
2 von 24 einzelpersoneniterationen
(weniger als 10% -> gerechtfertigt)
BSP: holy-grail relations.xml -> SQL + JAVA + PHP

entschluss für java -> ein ökosystem
erstes hackfest: working in seperate groups
scheduler implementiert + backend, wenig frontend

ant buildingtool angefangen
BSP:



initial design phase
* created relational model
* also started work on an object oriented model
* recognized soon, that disparity could break us the neck

SOLUTION
=> we declared entities using a custom XML-based format

Relational model

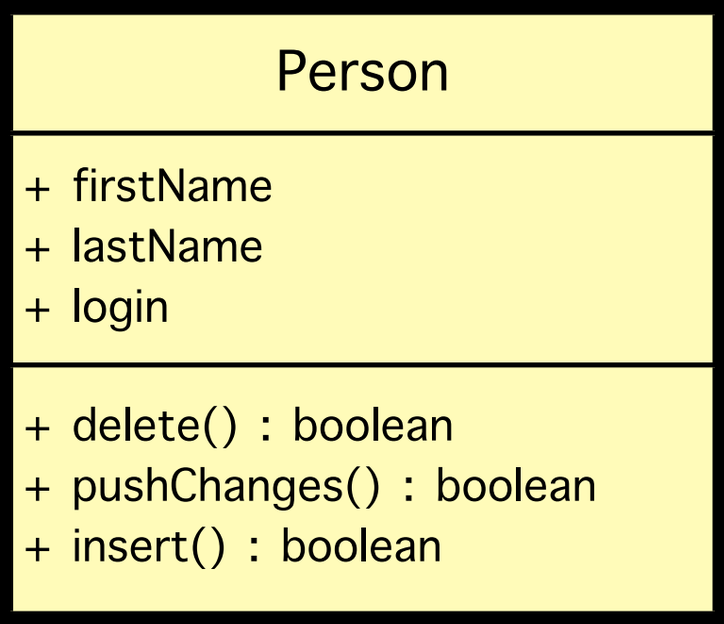
first_name	last_name	login

June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

initial design phase
* created relational model
* also started work on an object oriented model
* recognized soon, that disparity could break us the neck

SOLUTION
=> we declared entities using a custom XML-based format

Object oriented model



Relational model

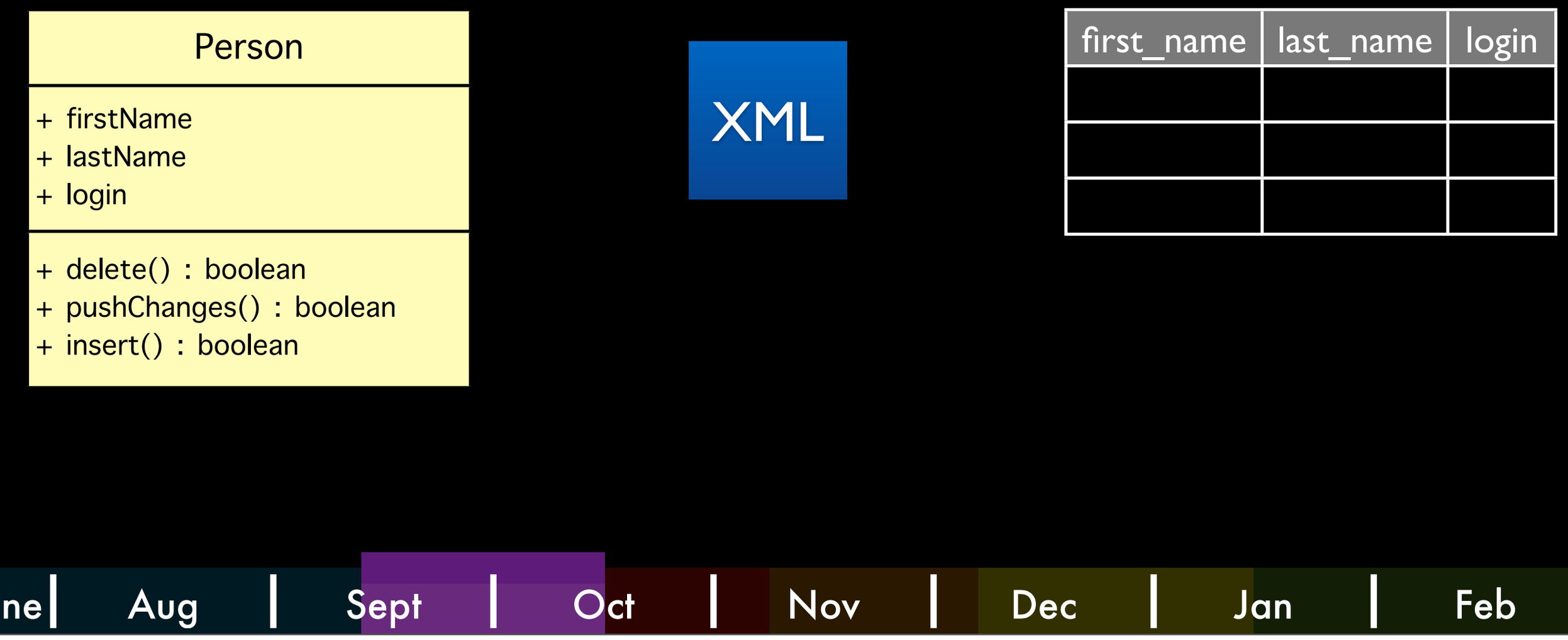
first_name	last_name	login

June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

Freitag, 6. Mai 2011

initial design phase
* created relational model
* also started work on an object oriented model
* recognized soon, that disparity could break us the neck

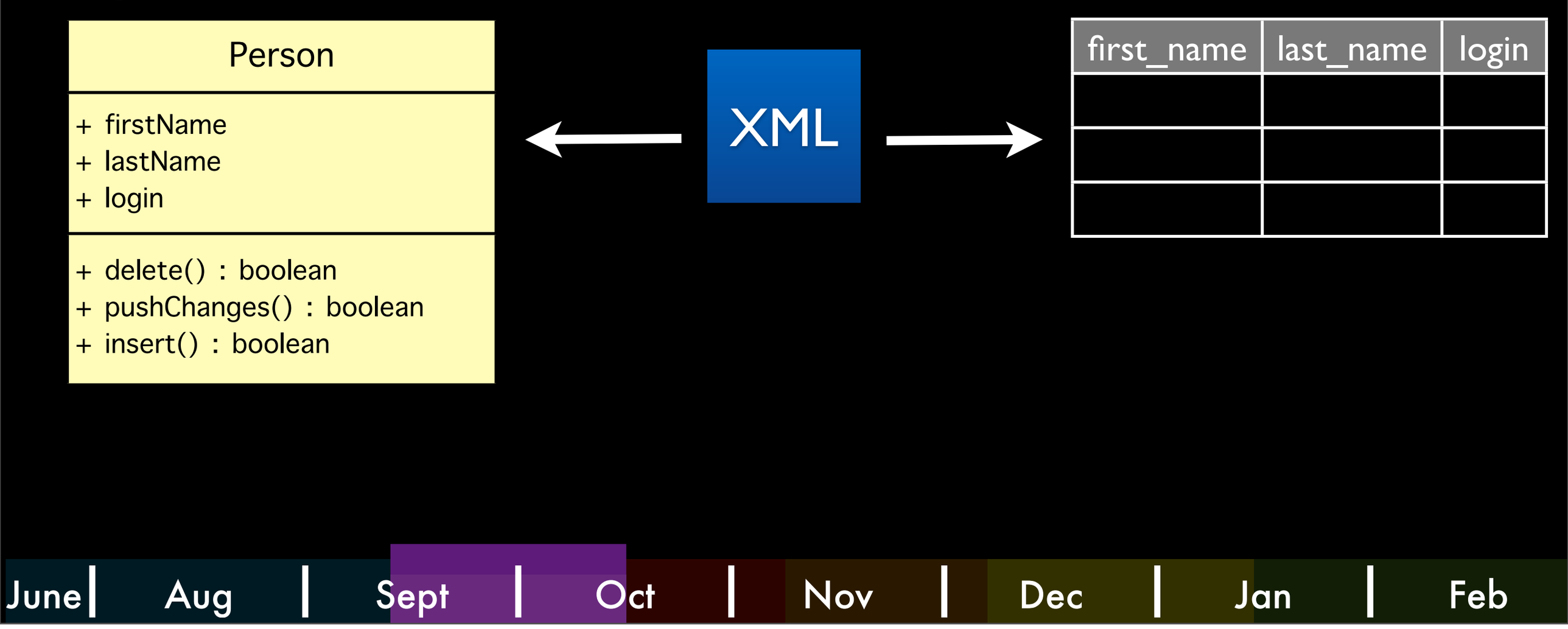
SOLUTION
=> we declared entities using a custom XML-based format



Freitag, 6. Mai 2011

initial design phase
* created relational model
* also started work on an object oriented model
* recognized soon, that disparity could break us the neck

SOLUTION
=> we declared entities using a custom XML-based format



Freitag, 6. Mai 2011

initial design phase
* created relational model
* also started work on an object oriented model
* recognized soon, that disparity could break us the neck

SOLUTION
=> we declared entities using a custom XML-based format

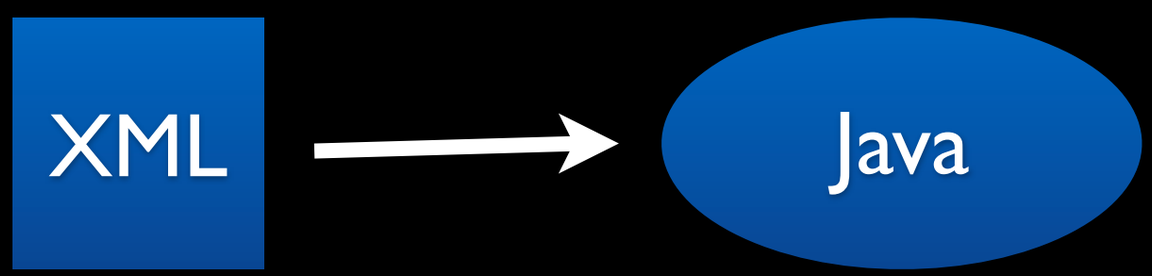
since all information was available in a well-defined format

built a toolbox which provided the following
XML -> Java (according to active record pattern ~explain~)

...the tool we built...
automatically set up the database

AND hypertext documentation
extremely necessary
* 42 entities and m-n relationships
* multitude of references between these

Java Code -> GUI View
landed late
business logic resulted by applying aspects
and including custom controller code (form commit)



June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

Freitag, 6. Mai 2011

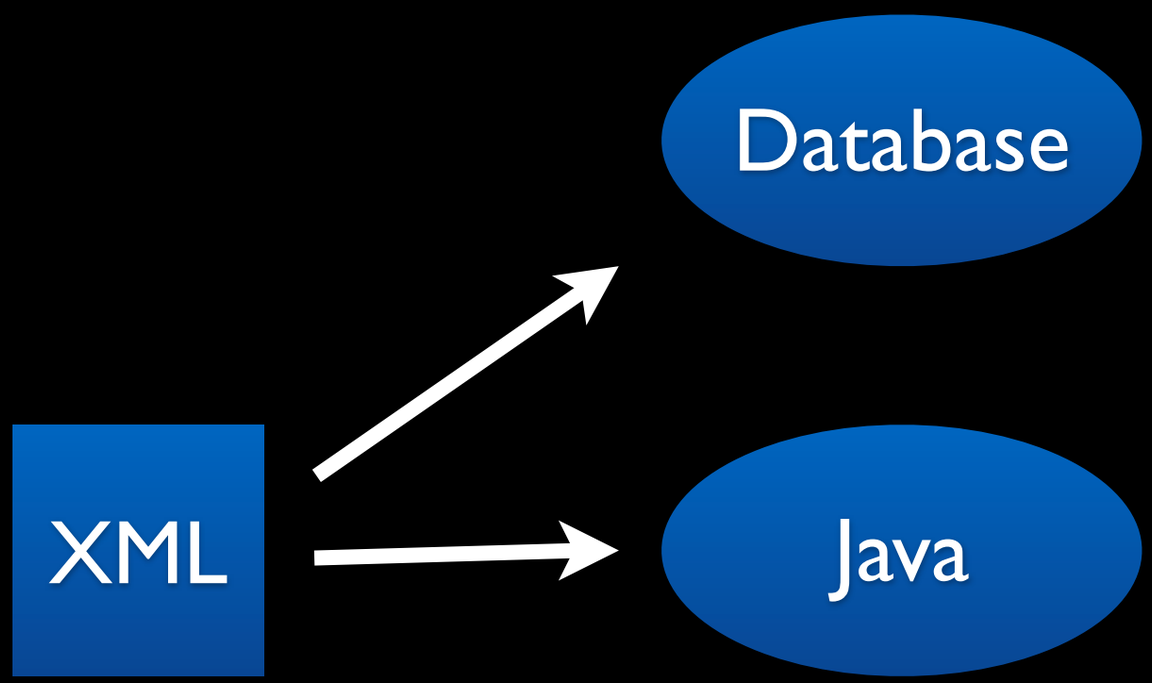
since all information was available in a well-defined format

built a toolbox which provided the following
XML -> Java (according to active record pattern ~explain~)

...the tool we built...
automatically set up the database

AND hypertext documentation
extremely necessary
* 42 entities and m-n relationships
* multitude of references between these

Java Code -> GUI View
landed late
business logic resulted by applying aspects
and including custom controller code (form commit)



June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

Freitag, 6. Mai 2011

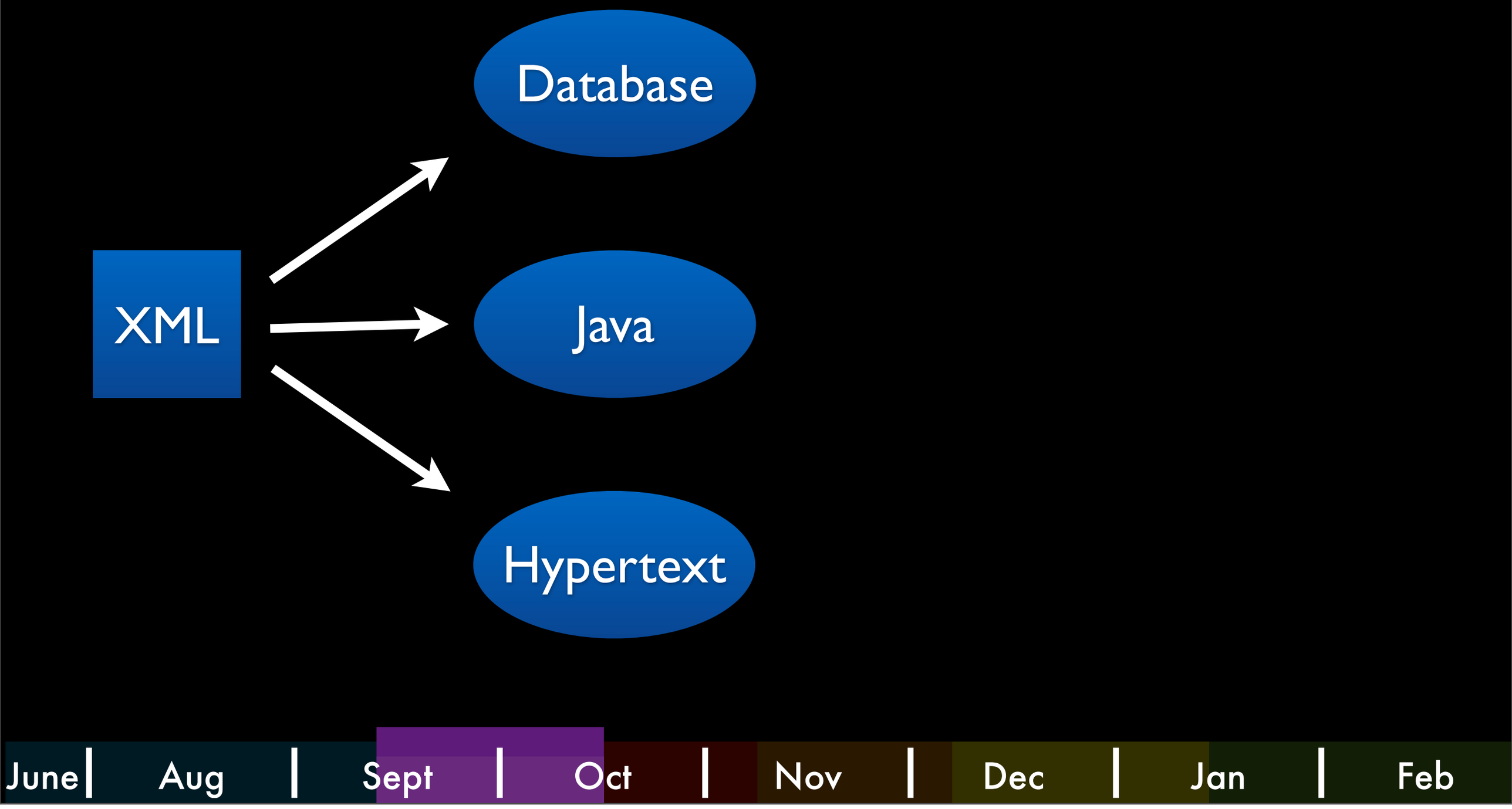
since all information was available in a well-defined format

built a toolbox which provided the following
XML -> Java (according to active record pattern ~explain~)

...the tool we built...
automatically set up the database

AND hypertext documentation
extremely necessary
* 42 entities and m-n relationships
* multitude of references between these

Java Code -> GUI View
landed late
business logic resulted by applying aspects
and including custom controller code (form commit)



Freitag, 6. Mai 2011

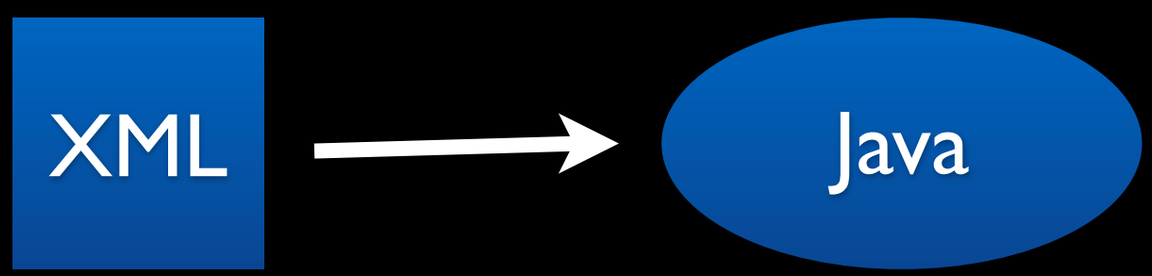
since all information was available in a well-defined format

built a toolbox which provided the following
XML -> Java (according to active record pattern ~explain~)

...the tool we built...
automatically set up the database

AND hypertext documentation
extremely necessary
* 42 entities and m-n relationships
* multitude of references between these

Java Code -> GUI View
landed late
business logic resulted by applying aspects
and including custom controller code (form commit)



June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

Freitag, 6. Mai 2011

since all information was available in a well-defined format

built a toolbox which provided the following
XML -> Java (according to active record pattern ~explain~)

...the tool we built...
automatically set up the database

AND hypertext documentation
extremely necessary
* 42 entities and m-n relationships
* multitude of references between these

Java Code -> GUI View
landed late
business logic resulted by applying aspects
and including custom controller code (form commit)

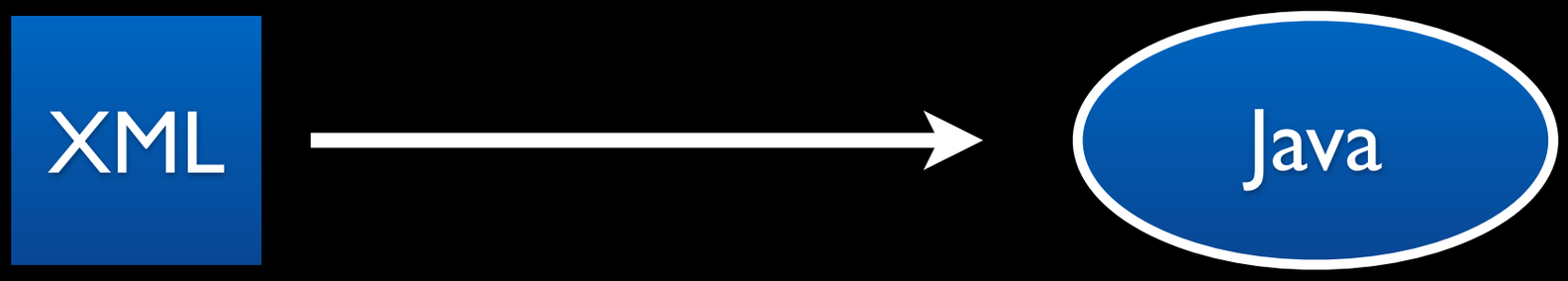

```
<entity name="Person">
    <attribute name="first_name"
        type="string" />

    <attribute name="last_name"
        type="string" />

    <attribute name="login_name"
        match="[a-z][a-z0-9]+"
        type="string"
        unique="true"
        min="3" max="40" />

    <attribute name="home_department"
        ref="Department"
        nullable="true" />
</entity>
```

June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

Freitag, 6. Mai 2011

Andre: ja, layout code <-> zeitleiste ist broken, ich arbeite dran

This is a very simplifed piece of our XML declaration




```
<entity name="Person">
    <attribute name="first_name"
        type="string" />

    <attribute name="last_name"
        type="string" />

    <attribute name="login_name"
        match="[a-z][a-z0-9]+"
        type="string"
        unique="true"
        min="3" max="40" />

    <attribute name="home_depart
        ref="Department"
        nullable="true" />
</entity>
```

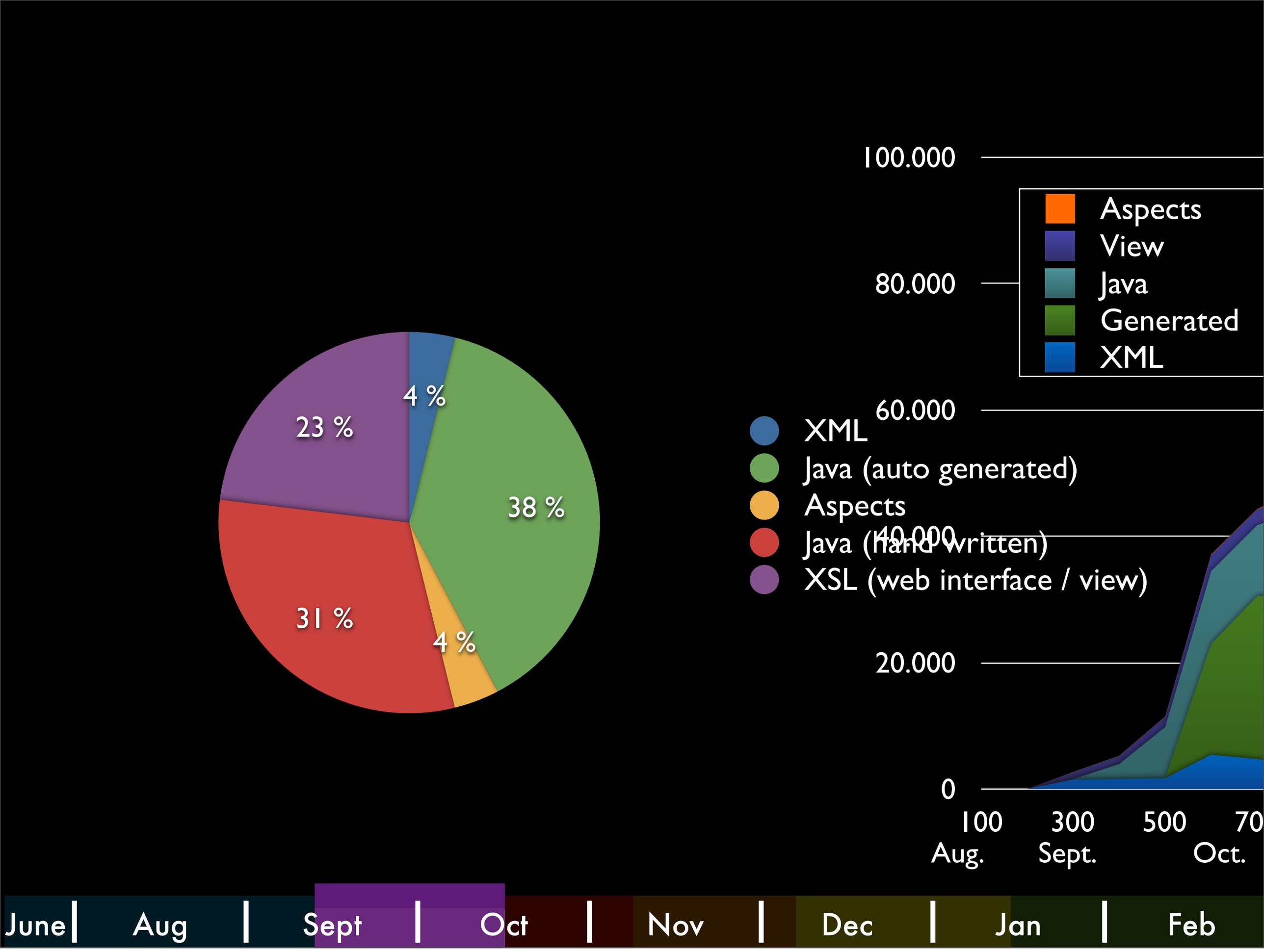
```
@Entity(name = "Person")
public class Person {
    public class Form {
        @Required
        public String firstName;

        @Required
        public String lastName;

        @Required @Min(3) @Max(40)
        @Match("[a-z][a-z0-9]+")
        public String loginName;

        public Department homeDepartment;
    }
}
```

June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

this is a tiny piece of the resulting java-code

* which illustrates the programmatically accessible forms

* annotations are our own work, too

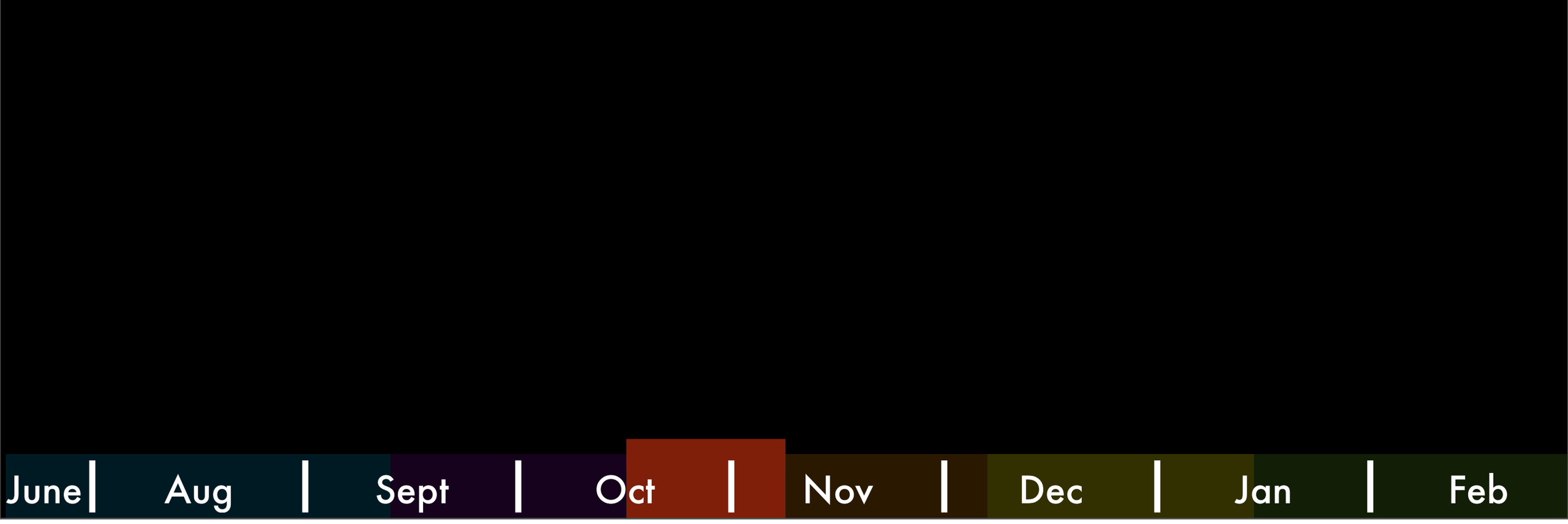


Freitag, 6. Mai 2011

Andre: inkonsitenz in den grafiken: hier hat xml und aspects 4% in dem verlaufsdiagram hat xml > aspects, ebenso kommt das mit view nicht hin
-> welche davon nehmen ?

GRAFIK ÜBER CODE-ANTEILE
(stats)

Iteration 3

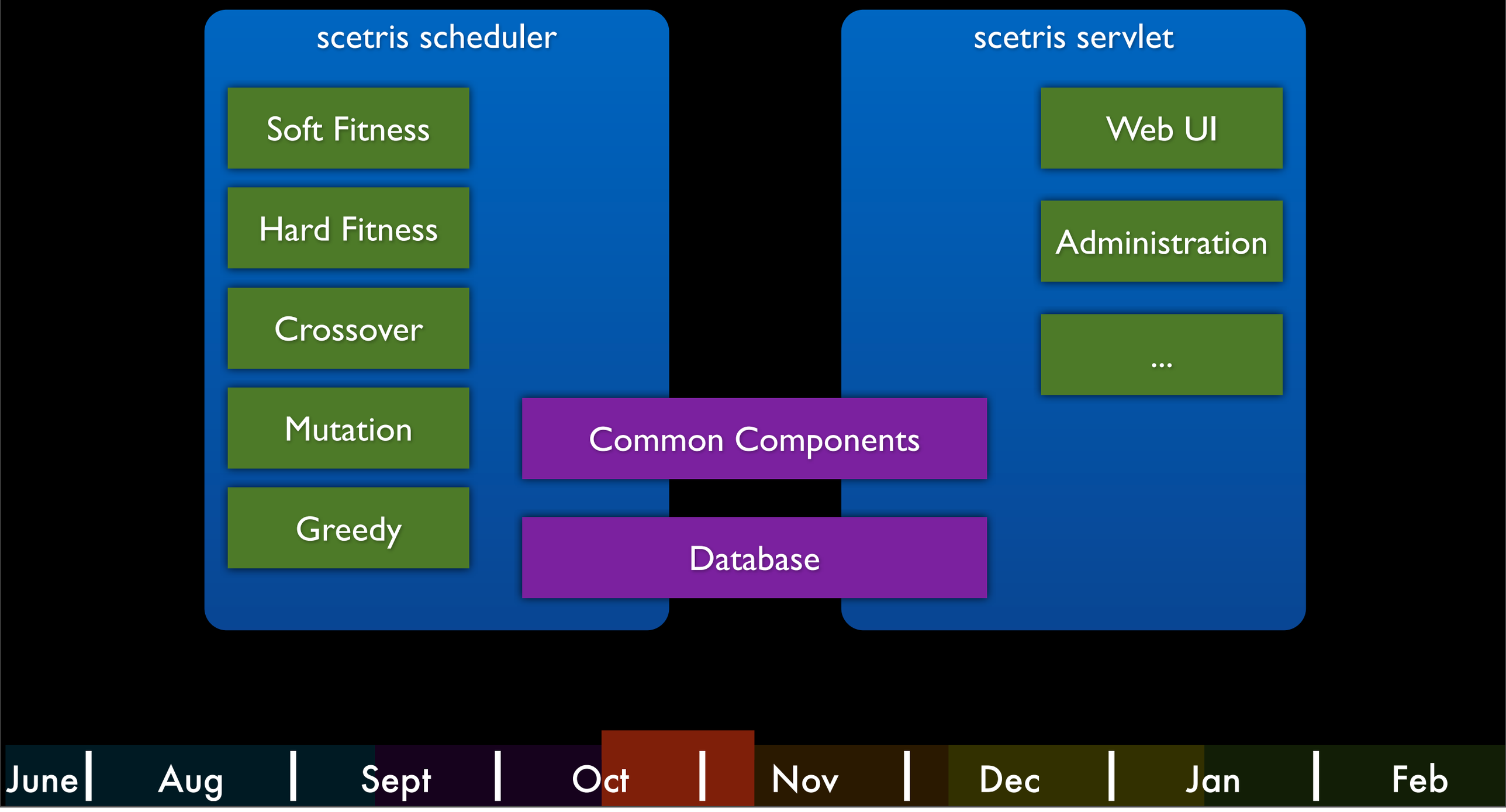


Freitag, 6. Mai 2011

scheduler + webframework integriert
lief als erste iteration wie geplant

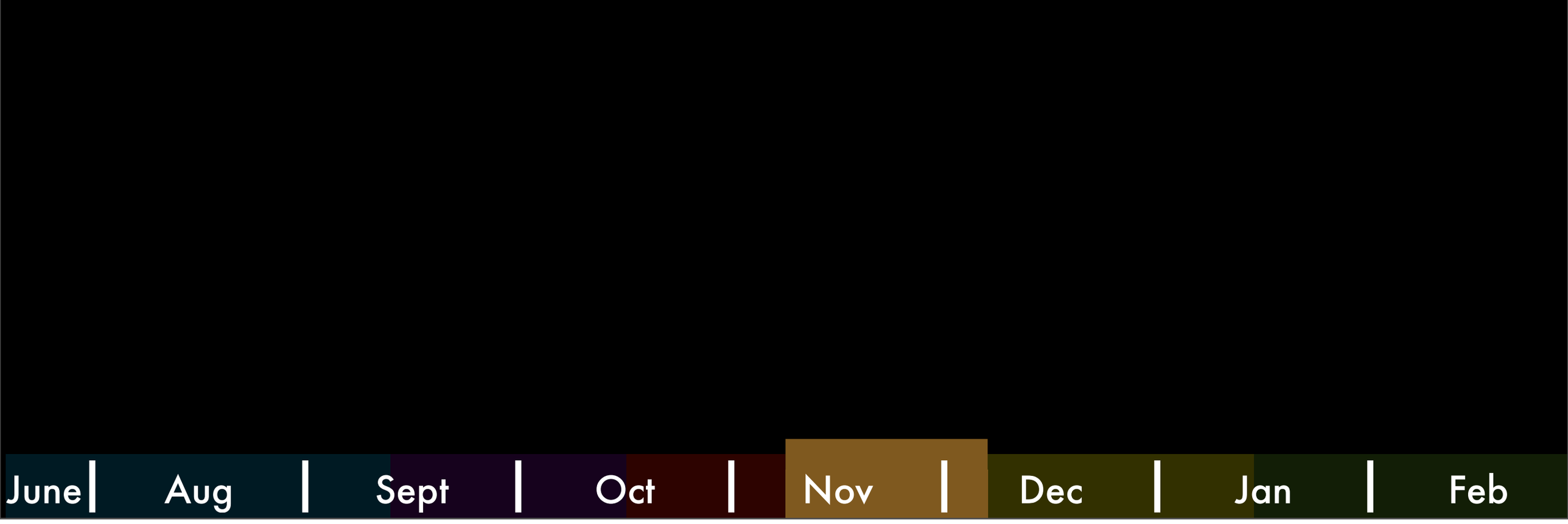
uni ging wieder los, dafür aber weekly meetings

BSP: integrationsgrafik vom scheduler <> ORM/webshit



Andre: sind die punkte im scetris-servlet ok, oder gibt es andere/bessere ?

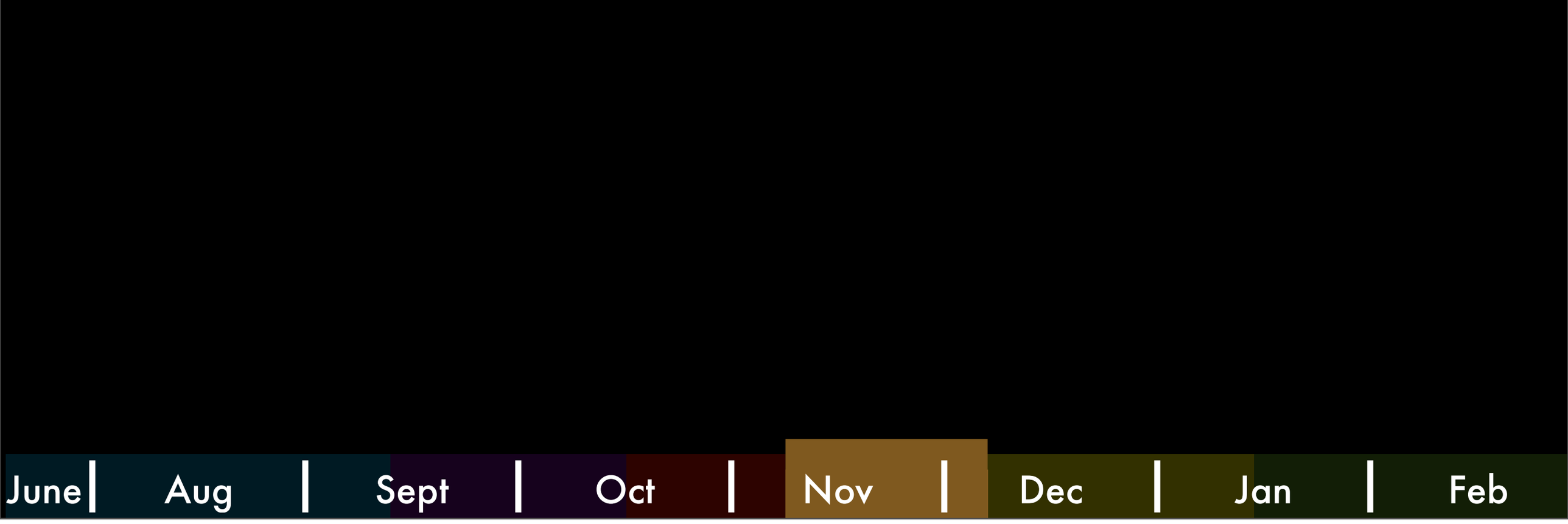
Iteration 4



usecases verfeinert
benchmarking

collaborativer scheduler
testdata generator
pdfexport
FU layout fürs web
viel QA // V&V

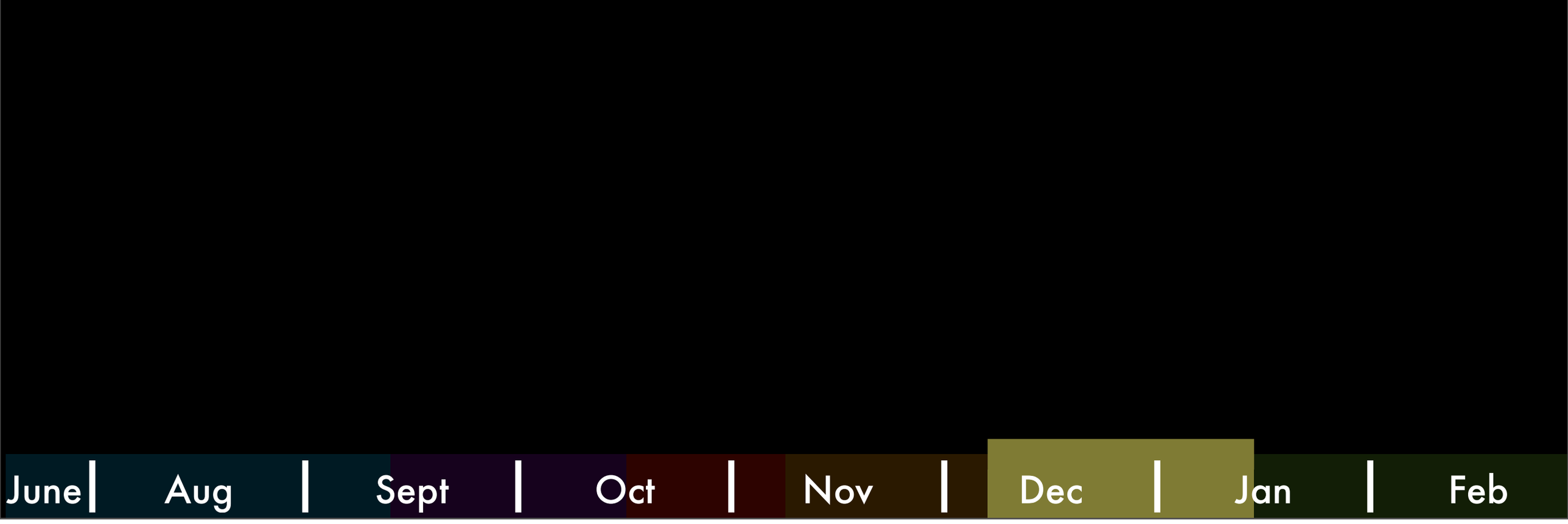
BSP: bild vom JVM-profiler ohne/mit querycache +
layoutchange von alt -> FU-stil



Freitag, 6. Mai 2011

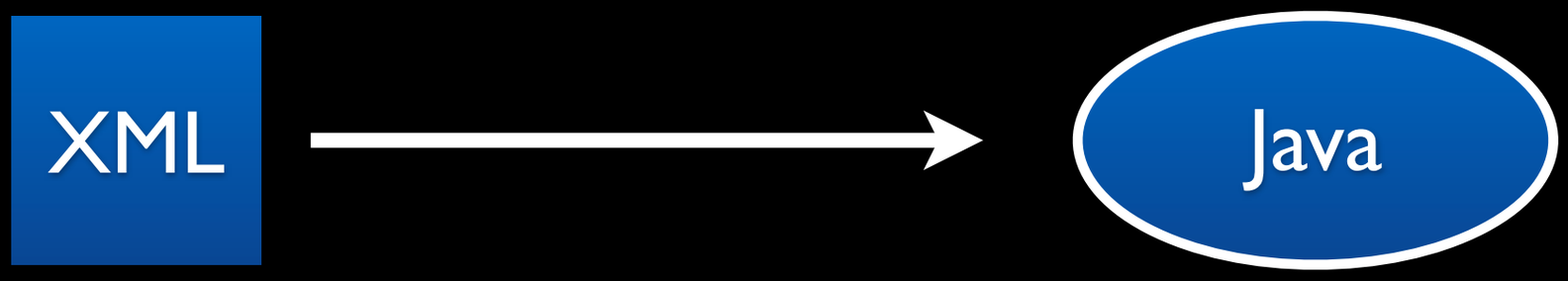
Andre: Konrad will mir Samstag(7.05) Bilder dazu reichen

Iteration 5



form-validation
summary report, danach drop
greedy

BSP: bsp von it.2 codegeneration aufnehmen -> form.class -> GUIform




```
<entity name="Person">
    <attribute name="first_name"
        type="string" />

    <attribute name="last_name"
        type="string" />

    <attribute name="login_name"
        match="[a-z][a-z0-9]+"
        type="string"
        unique="true"
        min="3" max="40" />

    <attribute name="home_depart
        ref="Department"
        nullable="true" />
</entity>
```

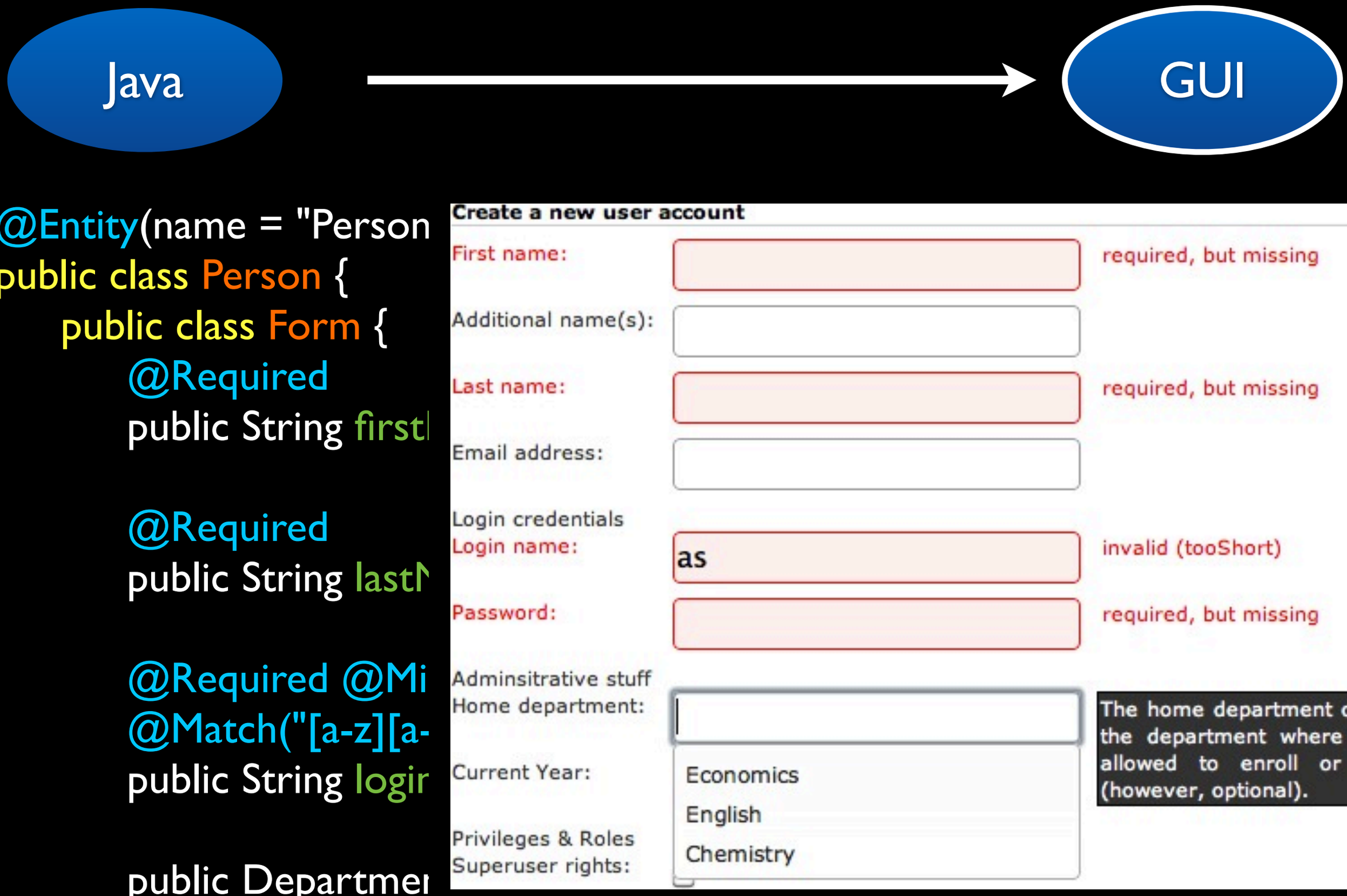
```
@Entity(name = "Person")
public class Person {
    public class Form {
        @Required
        public String firstName;

        @Required
        public String lastName;

        @Required @Min(3) @Max(40)
        @Match("[a-z][a-z0-9]+")
        public String loginName;

        public Department homeDepartment;
    }
}
```

June | Aug | Sept | Oct | Nov | Dec | Jan | Feb

this is a tiny piece of the resulting java-code

* which illustrates the programmatically accessible forms

* annotations are our own work, too



* gui also generated
* input validation automatically according to rules from xml-file

In our demo we will show you how to
-> easily customize and set up
-> the whole application
-> by changing a single file

Iteration 6

Freitag, 6. Mai 2011

afterwork,
pimping am gesamten projekt -> wegen weiterer abgabe
prepare final deliverable
dokumentation

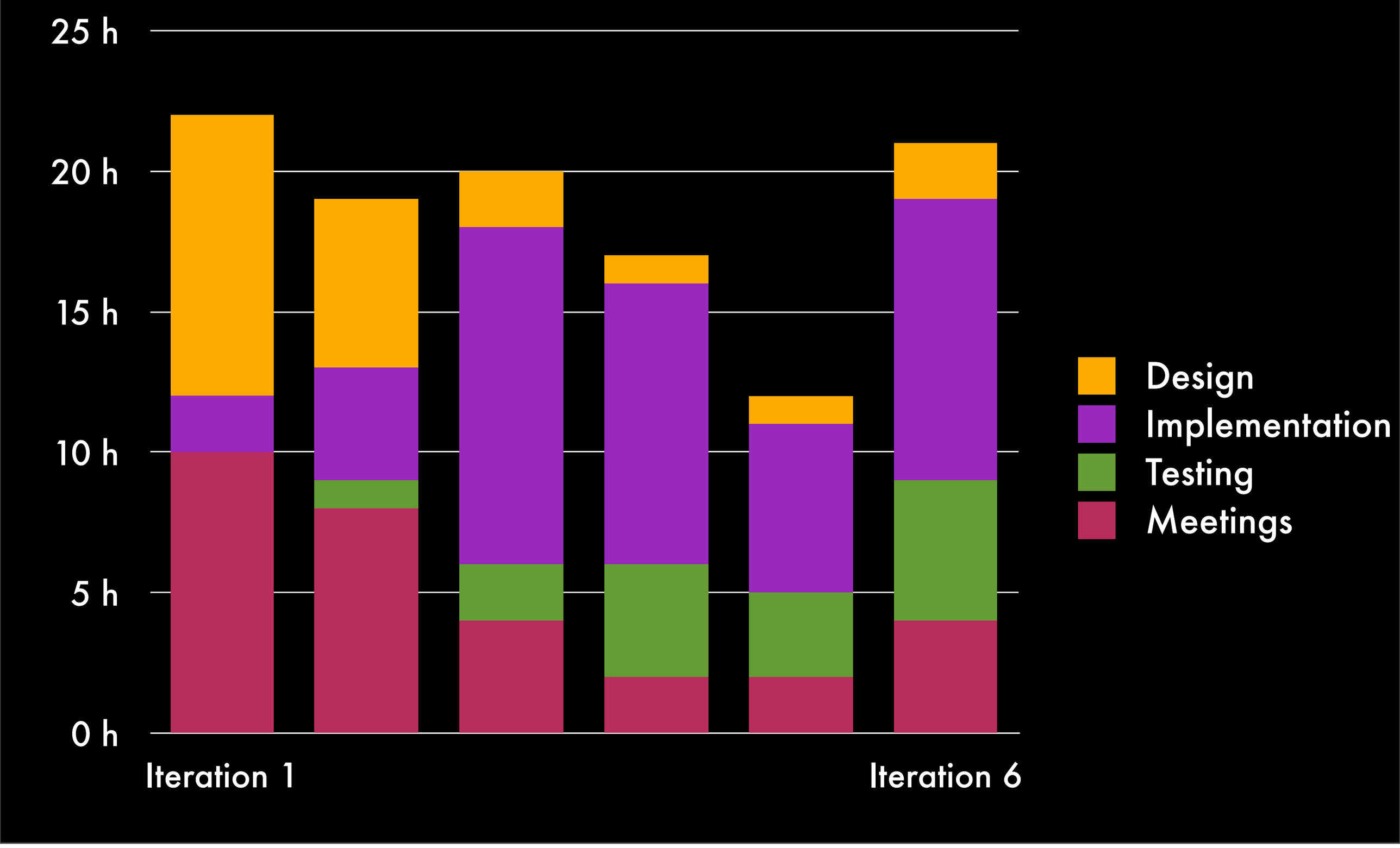
BSP: <x?>

Conclusion

- there is nothing like "complete spec"
- weekly meetings hard ?!
- fallbacks can hurt process

Andre: hier müssen wir das paket abschließen und können noch paar fakten einbauen (die ich oben auch schon mal erwähnt habe, z.b. eben hours devoted)

Hours devoted to the project / per person, per week



Freitag, 6. Mai 2011

we spent most time
in the first iteration

amount of time devoted to
implementation and testing grew

in our fifth iteration all of us were
quite exhausted, exams were approaching
-> motivation problem
=> project was late

we added another iteration
PROBLEMS:
* university, little time, especially for meetings
* regular meetings on a weekly basis
* “micro iterations” each week
* parallel testing and development, redesign as needed without breaking use-cases

...ended up with a lot more agile process

Thank You !

Any questions ?