Web Applications 2013

Intro

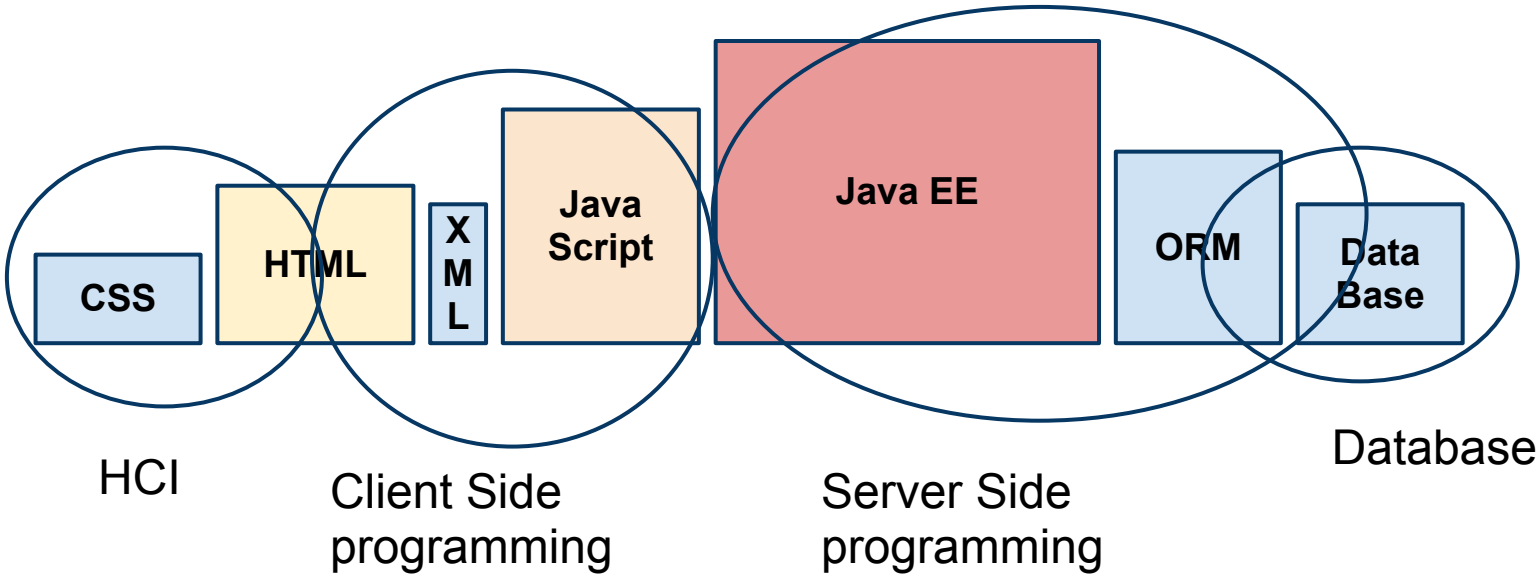
Designing a Web Application Course

Impossible to cover the area, have to choose ...

1. In-depth coverage of some part (will miss the big picture, will not be able to implement an application)? Or ...
2. ...overview of a major parts (will miss the in-depth knowledge, but will be able to implement at least basic application)?
3. .. other???

My choice: 2!

Course Profile



This is a programming course

This is a very applied course

What you will Gain

Area and technology overview

Increased programming, design and architecture skills (programming in the large)
- Many of you will end up in this area

Hands on experience with potentially very complicated/large software systems (touching many courses in one application)

Knowledge of standards and specifications, deeper understanding of API's,

What we will do

Practical experiments with a a number of concepts and quite a few techniques (during the workshops)

Workshops will present three application architecture/design approaches ...

- Request based approach
- Service based approach
- Component based approach

... and also some persistence issues (databases) and a glance at enterprise applications
... and finally a little "real time" web and upcoming technologies

Finding Facts

Very hard to find a single (or even multiple < 10) book (s)

- Recommended http://www.apress.com/9781430228899
- Also, ... area is moving too fast... books quickly deprecated

Recommended strategy

- Basic overview: Lectures, Wikipedia, Book
- More details: Book, Web articles (search)
- Final: Specifications and standards (more to come)

Many (carefully selected) links from course page!

Have to be careful on the Web, there are bad advices, contradicting advices, ...

The Holy Links

Problematic Aspects

Sadly even rather basic web applications consists of a great many (ugly) details

Easy for "newbies" to produce (very time consuming, strange) errors)

Must be pedantic: It's not just the code, is this allowed/not allowed?, is this handled automatically or not? config files, path's, directories, spelling, ...this is a "close to reality" course!

What's wrong here? "myHomepage.html "

Fixing the Problematic Aspects

… find the code snippet you need!

- **Many code samples on course page**
- **All possible to run**
- Have really tried to comment the code, many README-files
  (if some comment wrong or missing please let me know)
- Will save you a lot of time …
  (students tends to realize this too late .. .)

Also

- All workshop PMs are revised, now using code skeletons for kickstart
- All slides reviewed

Still there are things to grasp…

Skills Needed

Very good programmer!!!

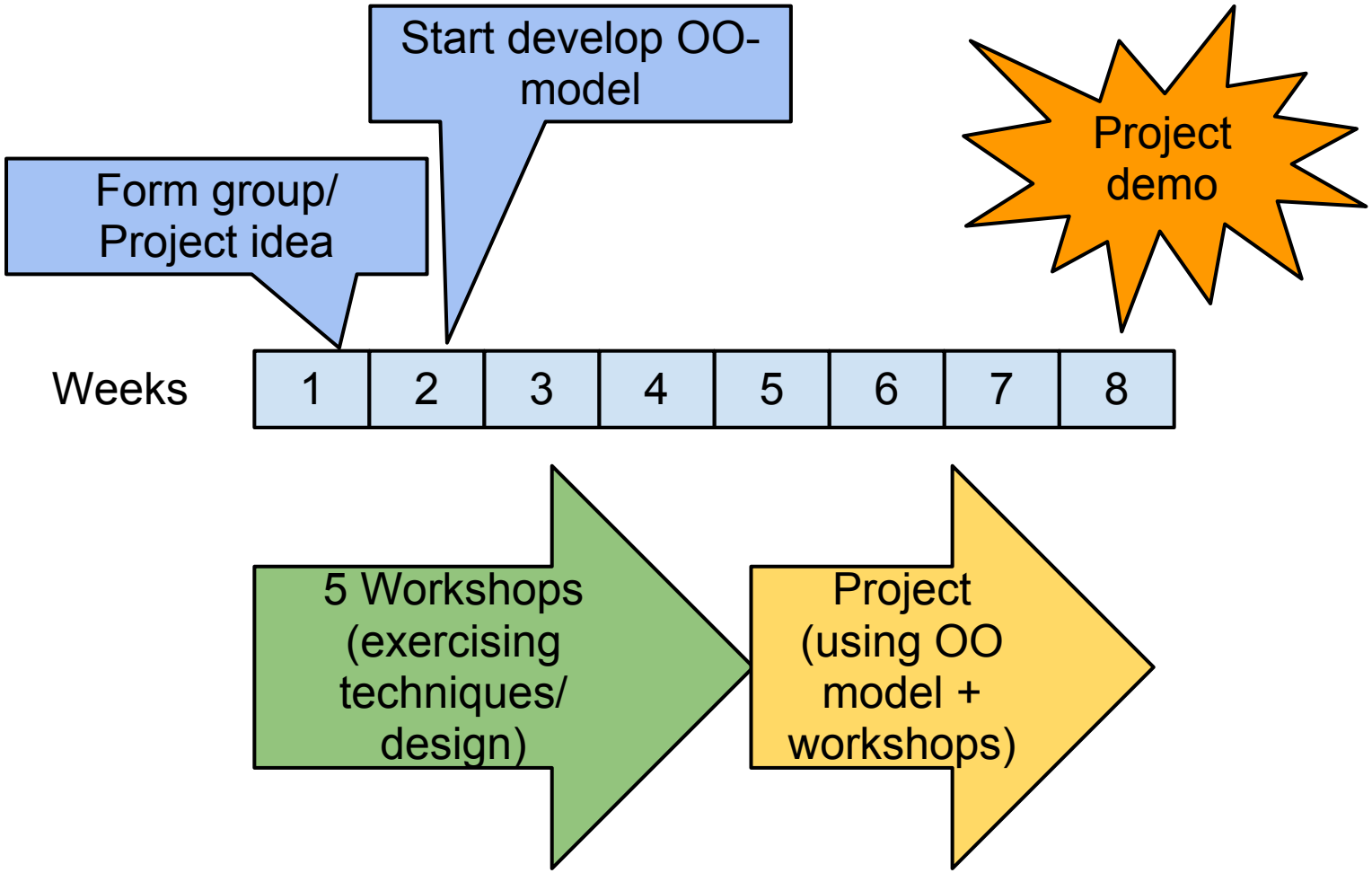
Good at solving problems in general

- It's not just programming, other problems will show up ...
- ... not always obvious where the problem comes from...
- ... can't guarantee bug-free environment (or code samples) , must be able to do "workarounds"
- ... must be able to carefully and thoroughly read error messages

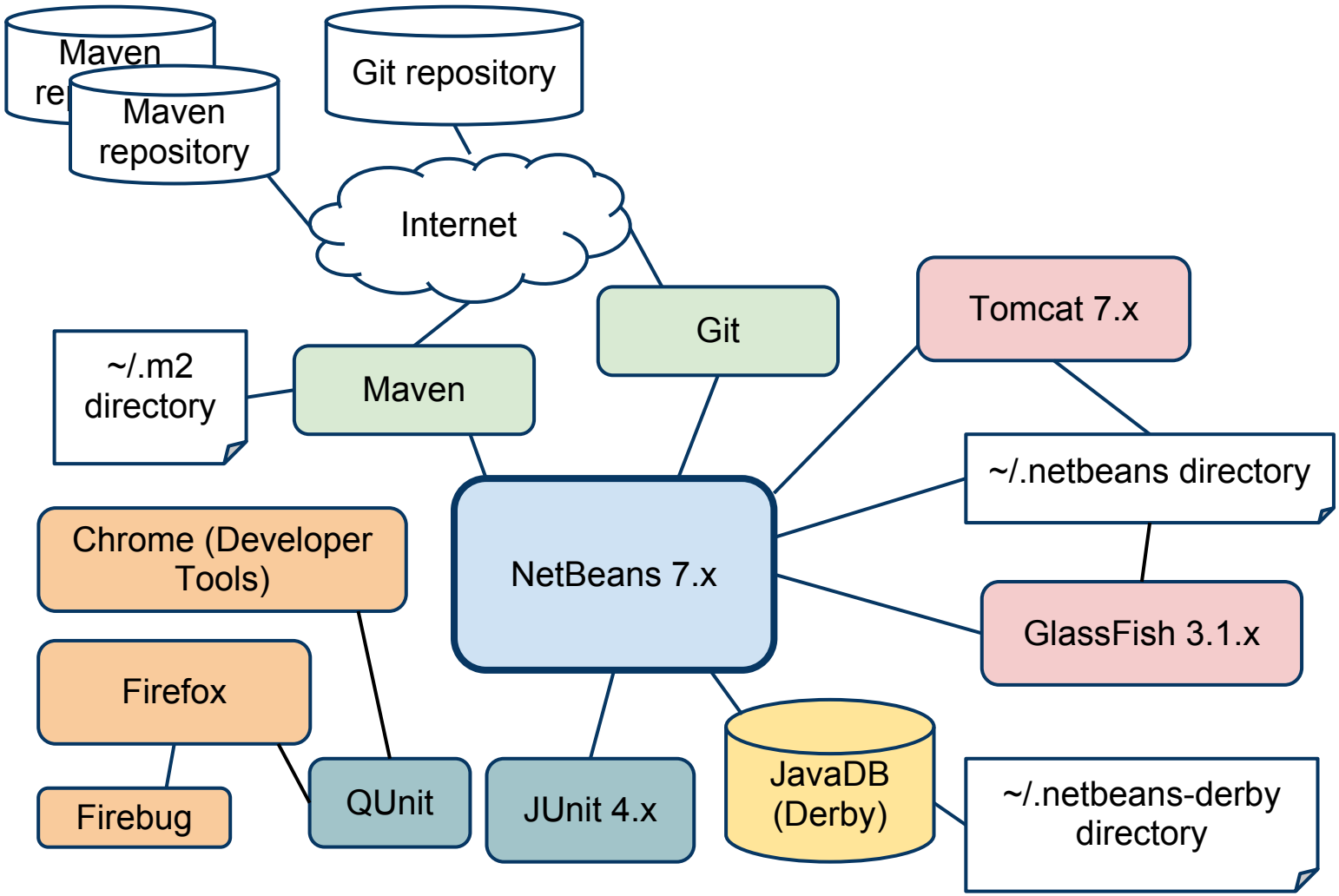
Good at speed-browsing text (documentation), searching information (or be willing to exercise it)

Accept the fact that where surfing on top of "tons" of software no time to understand in depth (programming by bulldozer)

Planning


Start develop OO-model
Form group/ Project idea
Project demo
Weeks
1
2
3
4
5
6
7
8
5 Workshops (exercising techniques/ design)
Project (using OO model + workshops)

Development Environment



Environment Disclaimer

This environment is installed at STUDAT but...

...**prefer a local installation (i.e. your laptop)**

Everything is platform independent and open source. Download, install, ...

NetBeans STUDAT: the version to run!
/chalmers/groups/ws-devel/netbeans-7.3/bin/netbeans (not in any menu)

NetBeans 7.x

Assume most of you have seen before...

- Central development/coding tool
- Create projects, and files, using wizards (File > New ...)
- Manages environment, Maven, JUnit, Tomcat, GlassFish, ...

Any tedious task can normally be done faster

- Search, Find, Replace (Edit menu)
- Refactoring (right click), rename, move
- Insert code (generate), constructors, setters/getters, ...

Configuration in ~/ .netbeans directory (delete and get a fresh installation)!

NetBeans Configuration

Menu Tools

Menu Item Options

- Many settings, default browser, tab Java even more...

Menu Item Plugins

- NetBeans is plugin based, i.e. possible to enhance
- We use plugins for Java SE, Java Web and EE, HTML5 and more... (will be there by default)

NOTE: NetBeans will start out rather clean, adding features automagically on demand...

Maven 3

"Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information."

Maven uses **archetypes** (blueprints) to create different types of projects (applications). We always uses/creates Maven-projects

A Project Object Model (POM) provides all the configuration for the project (in **pom.xml**). There should always be a pom.xml-file!

Maven has default ordered build/run lifecycle

- Process-resources, compile, process-test-resources , test-compile, test, package, install, deploy, run)
- Customizable, but we use default

Maven Project Directory Layout

Maven (default) assumes a fixed development directory layout (NetBeans will create, use Files tab to inspect)

```
myproj  (project directory)
      |-- src
      |    |--- main
      |    |     |--- java       // Application (Java) code
      |    |     |--- resources // Config,...
      |    |     |--- webapp    // HTML, CSS,...(only for webapps)
      |    |           |-- META-INF // Config for Java Web app
      |    |           |-- WEB-INF  // Config for Java Web app
      |    |--- test              // Test code
      |         |---java          // JUnit tests
      |         |---resources     // Test config
      |
      |-- target   // Compiled packed code (jar, war,...)
      |    |       // "the delivery" (final product)
      |-- pom.xml  // The POM
```

Maven Dependency Handling

Applications possible use a lot of libraries (API's)

- Dependencies between libraries (transitive dependencies)
- Complex to handle (jar-hell!)

Maven will handle transitive dependencies

- Will automatically download missing libraries from Web (central Maven repository or other) to local ~/.m2 directory
- Dependencies section in pom.xml will list all (direct) dependencies

Our projects will also be installed in ~/.m2 (will handle dependencies between our projects)

- Possible need to clean up m2 (if strange errors)

Maven Dependency Scopes

```
<!-- In pom.xml -->
<dependency>
...
   <scope> nnn </scope>
</dependency>

Where nnn = ...
```

- compile: This dependency is needed for compilation of the main source
- runtime: This dependency is needed for running the final artifact. It is not needed for compiling the main source or compiling or running the tests (jar added as library).
- provided: This dependency is needed for compiling and/or running the artifact but is not necessary to include in the package, because it is provided by the runtime environment (no added jars)
- test: This dependency is needed for compiling and running tests.
- ... and more
- Default: compile

Maven Problems

Can't just throw in all possible dependencies (and/or scopes) to be sure!

Will cause conflicts!

Running Maven

To build, install (= **deploy**) applications NetBeans will run Maven in background.

- Application run in place

Possible to run from command line, example

```
$ mvn test // Run phases (goals) related to testing
```

Git

"Git is a distributed revision control and source code management system" (initially by Linus Torvalds)

Assume most of you have some knowledge

- Mandatory to use Git for project
- If you haven't used Git, find some tutorial on the Web
- NetBeans can handle Git (a Git plugin) but easier to use command line (...I think)

Distributed Computing

All applications in course will be distributed

- Will have a client side (front end) and a server side (backend)
- Will do client and server programming

The Servers

Apache Tomcat, a Web server and Servlet container (more to come..)

GlassFish, a full Application server (more to come...)

Java DB (aka Derby), relational database management system (more to come...)

All managed from inside NetBeans

NOTE: There are many other servers possible to use, if having severe problems possible test another

The Clients

We'll only use browsers as clients (besides some tools)

Recommended browsers

- **Chrome** in particular the "Developer Tools" (built in). Chrome not in STUDAT :-(
- **Firefox** with Firebug add on
- Possible later versions of IE (not tested)

Test Driven Development

Of course we try to adhere to best practices

JUnit 4.x, unit testing for Java code (assume you have knowledge)

QUnit, unit testing of JavaScript, similar to JUnit

Note: JUnit is running in the Java SE environment (non-server environment). Have to handle (more later...)

Handling the Problems

- Use tools to test, verify, monitor, many in NetBeans (HttpMonitor, XML Verification, ...) ... also tools in browsers...
- Scale down, comment out ...
- Clean and build (possible delete **target** directory by hand)
- Check imports, class name correct but wrong package!
- Undeploy application from Server, restart Server
- Verify all configuration files
- Clean .m2 (rare)
- Check dependencies (conflicting...?)
- Clear browser cache
- Spelling!!!
- Check running Java processes with command jps -l (which holds the socket?)