# Summary Report for MyCourses by Scetris

## *Outline*

- Who we are

  Our skills (general), names and a photo

- Intended and actual process

  What we wanted to do and how, followed by deviations

- Problem Statement

  The requirements that were given and how we supplemented them

- Our design and our decisions

  - explain the design decisions where the design is mentioned

  - Subcategories include: Technology, MVC, Meta, ...

- Implementation

  How well we fared with our design, where we refactored

- The button

  The result of our work, i.e. the product and what it can/can't do plus where improvement is useful and feasible

- Timeline

  How the plan was, what happened when, possibly a graphical presentation

- What we learned

  What is our overall impression of the process, what would we change next time, what was great

- Conclusion

  Final thoughts on process and product

## *Who we are*

We are a team of five undergraduate students of the Free University of Berlin. Even though we are all in the same year our knowledge regarding the creation of software and the used technologies varied widely. We were however all inexperienced in project management and none of us had been part of a software development process before. This said we will introduce ourselves with names and a picture so you can get an idea.

….

## *Intended and actual process*

As mentioned before our team lacked experience in software engineering and we chose to follow a mostly agile software process as we were certain that we would learn a lot along the way and our first ideas and predictions were hardly precise. Choosing an agile approach we hoped to adapt easily to the things we would learn and thus rapidly improve our process and our design. While a strict agile approach was infeasible with all of us spending most of our time on courses at university we used our own ideas roughly based on agile software engineering. Our approach consisted of heavy usage of technology to ease communications and organization, regular meetings with the whole team, working alone and often in small teams without fixed times. All of these will be explained in more detail now.

We decided to use a trac system providing us with tickets and a wiki to coordinate the assignment of tasks and gather information about progress and possible problems. To ensure consistency in the code and gain backups we used subversion. For general communications a mailing-list was set up and heavily used.

We held regular meeting with the whole team to discuss what has been achieved, what could be improved and how the time until the next meeting will be spend. These meetings addressed mostly organizational issues but design and implementation were also discussed. From the end of May until the beginning of October meetings were held on a two-weeks basis. With the beginning of our third iteration in October we picked up the frequency and met weekly. This is mostly due to the fact that productivity increased a great deal around September and with the beginning of our third iteration increased coordination of efforts was necessary because the parts of our systems were ready for integration with each other.

Much of the work was done alone due to the fact that we could hardly find suitable → Termine/Zeiten <--- and the experience that productivity slipped with more people present. Still a significant amount of development was done in small teams. This was reasonable as we tried to avoid having single experts, because of the risks this poses. In particular we used pair-programming to implement the scheduler. This was done for two reasons, firstly to ensure a high quality of the code and secondly to have two people equally familiar with it.

## *Problem Statement*

Score Project description. Point out the areas that were unclear to us and how we gathered information to clarify these areas.

## *Our Design and Decisions*

Short introduction stating that the different categories of design will be described and a rationale for each will be given.

### Technologies

Why we used the technologies we did and why we did not use some others

- Programming Language
  - AspectJ as we were all familiar with Java and wanted to have aspects, which is especially useful to supplement auto-generated code
- DBMS
  - Postgres as we wanted it to be open source and had already worked with Postgres
- ORM
  - No existing solution but our own as we felt that existing solutions would take a similar amount of time to get used to.
- Server
  - Tomcat because of????

Tomcat ist kein fester Bestandteil unserer "Toolchain", wir benutzen ihn nur einfach als Tool so wie wir Eclipse benutzen. Wir unterstützen aktiv auch Jetty (aber erst seit diesem WE), und man kann es jetzt bereits ohne Probleme z.B. in Glassfish einbinden. Wenn wir ein Web-Archive deployen ist der Servlet-Container auch gar nicht von Interesse – er ist auch gar nicht Teil unseres Software-Bundles.

### MVC

Why we decided to use a MVC design and how it is implemented. What the advantages are.

Im Gegensatz zu Tomcat ist es wichtig, das verwendete Web-Framework zu erwähnen. Es ist bei Webanwendungen durchaus üblich eins zu benutzen (Rails, Lift, Grails, Struts, ...), aber wir haben auch hier unser eigenes geschrieben: weave. Nahezu alle folgen dem MVC-Pattern, was bei uns von Interesse ist, ist die Austauschbarkeit dank XSL-T.

### Meta

Why a framework for common tasks has been created and how it affected our development process.

### Scheduler-Algorithm

Why we have chosen a genetic algorithm for our scheduler and how we tested and improved performance.

## *Implementation*

Not sure what goes here, if anything at all

## *The button*

Description of our product. Features and the concept behind the UI. Possible improvements and add-ons.

Wir haben ein Konzept hinter unserer UI *g* ??

## *Timeline*

An outline of what happened when and where we kept the plan and where not. Possibly a graphical presentation. Must emphasize the fact that the plan evolved on the go.

## *What we learned*

What each of us has learned technology wise and what we have learned about projects.

Wir haben ja von Anfang an nicht Wasserfall gemacht. Allerdings hat sich unser Modell dahingehend verändert, dass die Iterationen kürzer geworden sind und es kleinere Untereinheiten gibt, die besonders im Uni-Betrieb jetzt nahezu wöchentlich, ineinander übergehend, beinahe fließend, statt finden. Auch sehr interessant, was Prechelt genau hier zu sagen wird.

### *Conclusion*

A short summary repeating the process, and the product.