

# CoMa PHP1

Abschlusspräsentation

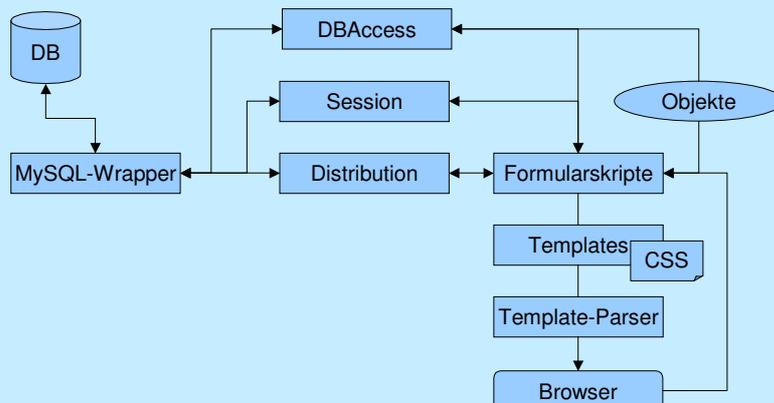
## Übersicht

- (1) Struktur von CoMa **PHP1**
  - Architektur
  - Zuständigkeiten
  - Features & Highlights
  - Fehlende Funktionalitäten
  - Eingeschränkte Funktionalitäten
- (2) Entwicklungsprozess
  - Entwicklungsphasen
  - Kooperation/Koordination
- (3) Demonstration
- (4) Abschließende Bemerkungen
  - Planung vs. Endprodukt
  - Fazit
  - Feedback

# (I) Struktur von CoMa PHP1

- Architektur
- Zuständigkeiten
- Features & Highlights
- Fehlende Funktionalitäten
- Eingeschränkte Funktionalitäten

## Architektur



# Architektur

- Was hat sich bewährt?
  - Grundsätzlich die durchgängig versuchte Modularisierung
    - Kapselung aller Datenbankzugriffe (SQL)
    - Sonstige Programmlogik in Formularskripten ohne HTML-Code
    - Kapselung des Layouts durch Templates/CSS ohne PHP-Code
  - Die Fehlerbehandlung (obwohl verbesserungswürdig)

# Architektur

- Was hat sich nicht bewährt?
  - DBAccess
    - Zu umfangreich, zu unübersichtlich und teilweise redundant
    - Aufspaltung in kleinere Einheiten wäre besser gewesen
  - Die Objekte
    - In dieser Form ungeeignet, weil zu umständlich zu handhaben
    - Häufig nur wenige Informationen nötig („mit Kanonen auf Spatzen“)
  - Untertemplates in dieser Form
    - Zu viele nötig (teilweise für eine HTML-Seite drei Verschachtelungsebenen, häufig zwei)
    - Schleifenkonstrukt hätte Abhilfe schaffen können

# Zuständigkeiten

⇒ Bei Bedarf am Ende

# Features & Highlights

⇒ Demonstration

## Fehlende Funktionalitäten

Was fehlt an CoMa, wodurch sein Zweck unerfüllt bleibt?

Keine derartigen fehlenden Funktionalitäten bekannt, d.h. CoMa ist funktionsfähig.

## Eingeschränkte Funktionalitäten

Welche Funktionalitäten gibt es, durch deren Fehlen zwar nicht die Grundfunktionalität von CoMa beeinträchtigt wird, die aber komfortabel/wünschenswert gewesen wären?

⇒ Demonstration

## (II) Entwicklungsprozess

- Entwicklungsphasen
- Kooperation/Koordination

## Entwicklungsphasen

- Planung, Strukturierung, Konzeption (Mitte Nov. – Anfang Dez.)
  - Grundlage: Spezifikation von Jan/Sandro/Tom
  - Einzelne Änderungen/Anpassungen/Erweiterungen daran
  - Strukturierung der anfallenden Aufgaben
  - Festlegung der Architektur
  - Festlegung der Zuständigkeiten
    - Einbezug von zeitlichen Abhängigkeiten („A muss fertig sein, bevor mit B begonnen werden kann“), so dass niemand unbeschäftigt war
- Erste Implementierungsphase (Anfang Dez. – Anfang Jan.)
  - Nach Aufgabeneinteilung
  - DBAccess-Fortschritt massiv durch SQL-Diskussion beeinträchtigt
  - Verzögerungen durch starke Abhängigkeit von DBAccess

## Entwicklungsphasen

- Reorganisation (Anfang Jan.)
  - Allgemeiner Austausch über den Stand der Dinge
  - Korrekturen an der Spezifikation
  - Daraus resultierend Restrukturierung von Aufgaben und Zuteilungen
- Zweite Implementierungsphase (seit Anfang Jan.)
  - DBAccess bis zuletzt im „Fluss“
  - Größerer Rückstand auch bzgl. Formularskripte und Templates
  - Unterteilung in Milestones „M1“, „M2“, „M3“, „Presentation“
  - Rückstand trotz „Todesmarsch“ auf ein bis zwei Wochen geschätzt
  - Reduzierung des Umfangs ermöglichte Realisierung noch
  - Aber keine explizite Testphase mehr (nur parallel zur Entwicklung)

## Kooperation/Koordination

- Sämtliche gruppenrelevante Entscheidungen wurden diskutiert und gemeinsam gelöst, teils durch Mehrheitsbeschluss
- Keine Position „Kordinator“, aber Einteilung der Mitglieder in Verantwortungsbereiche
- Koordination und Abstimmung meist über wöchentliche Meetings, je nach Bedarf und Phase auch häufiger
- Kommunikation über E-Mail, Forum, Bugzilla, ICQ, ...
- Einbindung des Testers aus verschiedenen Gründen schwierig (⇒ Vortrag Oliver)

## (III) Demonstration

## (IV) Abschl. Bemerkungen

- Planung vs. Endprodukt
- Fazit
- Feedback

## Planung vs. Endprodukt

- Spezifikation „Spec1“ (Jan/Sandro/Tom) als Basis
- Datenmodell infolge globaler Anforderungen geändert
- Letztlich einige Einschränkungen (fehlender Komfort), vgl. Demonstration, aber CoMa voll funktionsfähig
- Keine Administratorrolle (über Chair gelöst)
- Zeitlicher Verzug führte zu „Todesmarsch“, wodurch das Testen und Bugfixen deutlich zu kurz kam ⇒ Fehler zu erwarten

## Fazit

- Was hat sich bewährt?
  - Planungs-/Konzeptionsphasen zu Beginn und Anfang Januar
  - Kooperation und Zusammenarbeit wurde von allen als positiv und produktiv empfunden
  - Homogene Gruppe, sachliche Diskussionen, demokratische Einigungen
  - Gruppenzusammensetzung insgesamt günstig
- Worüber gehen die Meinungen auseinander?
  - „Bürokratisierung“ hinsichtlich Coding Conventions u.Ä.
  - Einsatz der diversen Kommunikationsmedien
  - Koordinator-Position

## Fazit

- Was hat sich nicht bewährt?
  - Zu viele verschiedene, teils globale, teils gruppeninterne Spezifikationen begonnen
  - Diese außerdem häufig unvollständig und teilweise veraltet
  - Dadurch Überblick verloren, Irritationen, fehlender Informationsfluss
  - Insgesamt zu kurzfristige Planung und falsche Einschätzung des Zeitpunkts der Fertigstellung z.B. der Klasse DBAccess und der Formularskripte

## Fazit

- Was hat sich nicht bewährt?
  - Daraus resultierende Folgen u.a.:
    - DBAccess bis zuletzt „im Fluss“
    - Formularskripte zum Ende von allen im „Todesmarsch“ zu bearbeiten
    - Unterschiedliche Kenntnisstände
    - Know-how-Transfer nötig, aber nicht immer sinnvoll oder möglich
    - Immer wieder tauchten mangels weitsichtiger Planung neue Aufgaben auf, für die es keine Zuordnungen gab und die dann zugewiesen wurden (Mehrbelastung)

## Fazit

- Was hat sich nicht bewährt?
  - Obwohl bekannt war, dass Aufwände anfangs unterschätzt werden, waren wir zu ehrgeizig
    - Zwar dennoch ein funktionsfähiges, fertiges Produkt
    - Aber Preis dafür „Todesmarsch“, vermutlich Fehleranfälligkeit
  - Die Zusammenarbeit mit einem externen Tester in dieser Form
  - Ausdrücklich nicht die Schuld von Oliver

## Feedback

- Spezifikationsphase („Spec1“, „Spec2“, globale Diskussionen)
  - Gut, dass diese Zeit ausschließlich für Spezifikation reserviert war
  - Aber: Durch anschließende Verschmelzung der beiden Spezifikationen unter Berücksichtigung weiterer Wünsche beide unbrauchbar (DB-Schema!) ⇒ Frage nach eigentlichem Sinn
  - Außerdem: Wer sich nicht für Spezifikation gemeldet hat, hätte kein Vetorecht diesbezüglich haben dürfen
  - Vor allem SQL-Diskussion unnötig
    - Eigentlich schon durch Spezifikationen DB-Schema vorgegeben, jedoch dann unbrauchbar
    - Rolle der „SQL-Repräsentanten“ abermals sinnlos

# Feedback

- Entwicklungsphase (als Gruppe PHP1)
  - Von allen Teilnehmern positiv bewertet
  - Aufteilung in kleinere Gruppen gut, Gruppengröße okay
  - Evtl. über Einführung einer Koordinator-Position nachdenken
  - Praktikum insgesamt stark von der Implementierung getrieben ⇒ Nachdenken über evtl. komplexere Aufgabenstellung (mehr Spezifikation/Planung/feste Schnittstellen), aber dafür nur ein Teil davon zu entwickeln
  - Integration des Testers in größerem Maße kaum möglich
    - Durch Olivers Kooperationsbereitschaft, den Gesamterfolg nicht zu gefährden, dennoch das Beste daraus gemacht

# Ende

Vielen Dank für die Aufmerksamkeit!