

## Bachelor of Science (BSc) in Informatik Modul Software-Projekt 3 (PM3)

# Einführung und Kick-off

WIN: Kurt Bleisch (bles), Daniel Liebhart (lieh),  
Johanna Decurtins (zubj)  
ZH: Reto Ferri (feit), Gerhard Wyder (wydg)  
Johanna Decurtins (zubj)

Ausgabe: HS24

# Um was geht es?

- Sie entwickeln in Teams eine selbstgewählte Projektidee von der Vision bis zu einer lauffähigen, grösseren Applikation.
- Dabei werden die vermittelten Analyse- und Entwurfs-Kompetenzen aus dem Modul SWEN1 angewendet (Wissenstransfer).
- Weiter werden Realisierungs- und Projektmanagement-Kompetenzen beübt sowie die Sozial- und Selbstkompetenzen (Non-Technical Skills abgekürzt NonTechS) gestärkt.



# Agenda

---

1. Einführung
2. Erklärung Projektergebnisse
3. Input Schreiben 1
4. Anwesenheitskontrolle
5. Teamauslösung
6. Erster Auftrag

# Fachliche Lernziele PM3

- Sie sind in der Lage,
  - eine grössere, objektorientierte Softwareapplikation von der Idee bis zur lauffähigen Applikation (hier Prototyp genannt) mit einem vorgegebenen iterativ-inkrementellen Softwareentwicklungsprozess zu realisieren.
  - die im Softwareentwicklungsprozess definierten Entwicklungs- und Projektmanagement-Artefakte selbständig zu erstellen.
  - das Projekt im Team iterativ und risikogetrieben zu planen und abzuwickeln.
  - die zu erstellenden Entwicklungs-Artefakte mit einer standardisierten Notation (wie UML) zu modellieren, darzustellen und zu kommunizieren.
  - konsequent Analyse-, Architektur- und Design Patterns sowie angemessenen Frameworks und Tools (für Versionierung, Build etc.) einzusetzen.
  - ein objektorientiertes Design inkrementell umzusetzen, zu testen und laufend zu verbessern (Refactoring).
  - wartbare und erweiterbare Software zu erstellen (Clean Architecture, Clean Code).

# Sprachliche/NoTechS Lernziele PM3

---

- Sie sind in der Lage,
  - eine technische Recherche (Konkurrenzanalyse, Technologie) durchzuführen und standardkonform festzuhalten.
  - technische Inhalte adressatengerecht mündlich und schriftlich zu kommunizieren
  - Ihre Kompetenzen optimal in ein Team einzubringen.
  - Ihre Rolle im Team ausreichend zu reflektieren und sie angemessen zu entwickeln

# Kommunikative Inhalte Projektmodule

## Software-Projekt 1-4

### PM1

Sitzungsleitung

Feedback auf Meetings

Beschlussprotokolle

Rollenreflexion

### PM2

Präsentation  
Projektidee für Investoren

Projektidee entwickeln und  
formulieren

Reflexion

### PM3

Präsentation  
Projekt für Anwender oder  
potentielle Käufer

Technisches Dokument mit  
Abstract schreiben

Reflexion

### PM4

Workshops:

Präsentationstraining mit  
Videoaufnahmen

Technische Dokumente  
schreiben

Vermarktung der Projekte  
(Videoclips, etc.)

Wissenschaftliche vs.  
technische Dokumente

# Organisation

---

- 4 Lektionen Praktikum pro Woche
- Theoretische Grundlage:
  - Modul SWEN1 ([Link](#))
  - Projektmanagement für PM3
- Kurs auf Moodle für PM3:
  - Termine, Unterlagen und Abgaben ([Link](#))
- Keine Semesterendprüfung (SEP)

# Ablauf

- Sie entwickeln in 5er (+/-1) Teams selbständig eine Anwendung
  - Von der Idee bis zur Realisation
- Teams
  - Studierende werden eingeteilt bzw. ausgelost
  - Team bestimmt Projektleiter (PL und Stv.) selber
- Iterationen/Meilensteine
  - Fixe zweiwöchige Iterationen
  - 3 Meilensteine mit Abgabe der Artefakte (s. Auftrag Projektergebnisse)
  - 2 Präsentationen der Meilensteinresultate im Plenum (siehe Auftrag Meilenstein-Präsentationen)
- Iterations-Reviews mit dem Fachdozierenden
  - Fixer Zeitpunkt (**aktuelle Termine im Moodle-Kurs** unter «Termine <Klasse>»)
  - Fortschritt und Planung der nächsten Iterationen (s. Auftrag Iterations-Review)
  - Feedback zu den Abgaben
- Rolle Dozierende: Stakeholder / Coaches



# Meilensteine und Iterationen

- **M1: Projektskizze, 1 Iteration (2 Wochen)**
  - Evaluation: Stakeholder-Agreement über Vision, Projektziele und Umfang, vereinbarter erster Phasen-/Iterationsplan, erste Risiken identifiziert
  - Artefakte: Projektskizze (Vision/Idee, Konkurrenzanalyse, erste Anforderungen, Wirtschaftlichkeit, Grobplanung, Risiken)
- **M2: Lösungsarchitektur, 3 Iterationen (6 Wochen)**
  - Evaluation: Vision, Ziele und Anforderungen stabil (80%), verifizierte (mit Quellcode) und dokumentierte Softwarearchitektur, Vereinbarung der Features für den Prototyp (M3)
  - Artefakte: Quellcode der Lösungsarchitektur (inkl. Code-Dokumentation), technischer Bericht I (Zusammenfassung von Use-Case-Modell, zusätzlichen Anforderungen, Domänenmodell, dokumentierter Softwarearchitektur und Projektmanagement)
- **M3: Prototyp, 2 Iterationen (4 Wochen)**
  - Evaluation: Feature-complete und stabil, lauffähig, Usability durchdacht und vorzeigbar
  - Artefakte: Quellcode der Applikation (inkl. Code-Dokumentation), technischer Bericht II (Zusammenfassung aller Entwicklungs- und Projektmanagement-Artefakte)

# Szenario oder Praxisbezug

- Sie nehmen an einem Innovations-Wettbewerb teil.
- In zufälligen Teams sollen Sie neue Anwendungsideen in Form eines Prototyps entwickeln.
  - Funktionalität ist zentral. Die zentralen, wichtigen und auch anspruchsvollen Punkte Ihrer neuen Anwendungsidee sind realisiert.
  - Das UI entspricht von der Bedienung her in etwa einer richtigen Anwendung, muss aber graphisch nicht besonders ausgearbeitet werden.
- Es geht nicht darum, eine marktfähige Anwendung, auch nicht als Vorstufe, zu realisieren.
  - Es ist normal, dass Sie noch nicht die ganze Anwendung programmiert haben.
  - Es ist normal, dass nicht-funktionale Anforderungen nicht realisiert sind.
- Nach 12 Wochen (am Schluss des Semesters) präsentieren Sie Ihren Prototyp, und erst dann wird entschieden, ob daraus ein richtiges Projekt entsteht oder nicht.

# Didaktische Vorgaben in PM3 (1/7)

---

- PM3 ist die Anwendung von SWEN1
  - Bei der Projektidee und der Durchführung haben diese didaktischen Ziele Priorität
- Wichtig: Anspruchsvolle Funktionalität
  - Bietet genug Stoff für objektorientierte Analyse und objektorientiertes Design
  - Nur so können sinnvoll Design-Patterns eingesetzt werden
- Wichtig: Usability
  - Der einzige Aspekt, der für die Beurteilung des UI herangezogen wird

# Didaktische Vorgaben in PM3 (2/7)

- Neutral: Datenerfassung und –Verwaltung (CRUD)
  - Für die fertige Anwendung natürlich notwendig
  - , aber um Funktionalitäten auszuprobieren nicht wirklich wichtig
  - , aber langweilig in der Analyse und im Design
  - Darf ohne Probleme «gefaked» werden
- Neutral: Visualisierung von Daten
  - Primär eine UI-Angelegenheit
  - Im Wesentlichen der Einsatz von Graphik-Bibliotheken

# Didaktische Vorgaben in PM3 (3/7)

- Neutral: Einsatz von Bibliotheken mit enger Breite der Funktionalität
  - Zum Beispiel Matrix-Berechnungs-Bibliothek
  - Bibliothek darf Architektur nicht vorgeben
  - Kann durchaus reizvolle Resultate bringen
  - Einarbeitungsaufwand nicht unterschätzen
  - KI Bibliotheken gehören auch dazu.
- Unwichtig: Graphisches Aussehen des UI
  - Ein schönes UI wirkt positiv in der Schlusspräsentation
  - Nicht notenrelevant, kein «Schönheitswettbewerb»

# Didaktische Vorgaben in PM3 (4/7)

- Unwichtig: Persistenz
  - Sinnvolle Arbeitsdaten dürfen am Anfang per Programmcode initialisiert werden.
  - Es hat keinerlei Auswirkungen auf die Notengebung, wenn geänderte Daten nach Beendigung der Anwendung verloren gehen
  - Allenfalls können beim Schliessen der Anwendung die aktuellen Daten in ein File (z.B. JSON Format) gespeichert werden.
- Negativ: Einsatz eines Frameworks
  - Ein Framework gibt die Architektur vor. Daher kann dieses didaktische Ziel dann gar nicht erreicht werden.
  - Die Einarbeitung in ein neues Framework ist mit viel Aufwand verbunden, der dann für die eigentlichen didaktischen Ziele fehlt.

# Didaktische Vorgaben in PM3 (5/7)

- **Negativ: Relationale Datenbank**
  - Grundsätzlich sind relationale Datenbanken eine gute Sache und bieten mit den ACID-Eigenschaften die Basis vieler kommerzieller Geschäftsanwendungen.
  - Leider steht ihr Einsatz den didaktischen Zielen von PM3 im Wege.
  - Der Einsatz einer relationalen Datenbank hat fast zwangsläufig eine Modellierung gemäss relationalem Datenmodell zu Folge, was einen gewissen Widerspruch zu OO-Analyse darstellt.
  - Erfordert eigentlich den Einsatz eines Frameworks. Siehe vorheriger Punkt.

# Didaktische Vorgaben in PM3 (6/7)

- Zwiespältig: Client-Server
  - Kann für gewisse Spiele (z.B. Poker) «notwendig» sein, da ein Spieler ja nicht die Karten seiner Gegner sehen darf, was bei einer Desktop-App ja kaum zu vermeiden wäre
    - Es gibt das Konzept vom «Hot-Swap», wo jeder Spieler hintereinander seine Eingaben machen kann, nach dem er einen Dialog weggeklickt hat. Kommt aus der grauen Vorzeit der Computerspiele und macht nur bedingt Spass, wäre aber eine Möglichkeit
    - Man «Faked» mehrere Clients, in dem auf demselben Desktop mehrere UI offen sind. Dies wäre für die Schlussdemonstration auch die einfachste Variante.
  - Kann die Trennung von UI und Domänenlogik «erzwingen»
  - Kann selber programmiert werden, oft wird aber ein Framework eingesetzt (siehe dort)



# Didaktische Vorgaben in PM3 (7/7)

- UI-Technologie
  - Erste Wahl ist JavaFX, da dies in früheren Semestern bereits angewendet wurde.
  - Browser Technologien sind auch denkbar, führen fast zwangsläufig zu Client-Server Architekturen.
    - Möglich, wenn die Client-Server Problematik gelöst werden kann.
    - Gerade bei Studierenden mit grosser Praxiserfahrung ist aber oft sehr viel Knowhow vorhanden.
    - Ohne Support durch Dozierenden.

# Projektidee

- Grundsätzlich frei
  - Spiele, wie Geschäftsanwendungen
  - Desktop und Android-Anwendungen
- Sie muss die didaktischen Ziele unterstützen
- Sie muss genug Stoff für das ganze Semester liefern
  - Funktionalität ist notenrelevant
  - 1 Credit entspricht 25–30 Std Studierenden-Aufwand  
→ Pro Person 120 Stunden für PM3 bis SW13!
  - Die Fachdozierenden werden mit den Teams zu gegebener Zeit eine Vereinbarung über den umzusetzenden Teil der Anforderungen treffen und die Regeln festlegen, wie Änderungen daran noch vorgenommen werden können.

# Randbedingungen

- Programmiersprache
  - Primär Java, da dies alle beherrschen
  - allenfalls C# oder C++ (nach Absprache mit Fachdozenten)
- Anwendung des Theoriestoffs von SWEN1
- Selber definierte Software Architektur
  - Mindestens geschichtete Architektur, besser wäre Clean-Architecture
- OO-Analyse und OO-Design für Domänenlogik
- UI muss die gesamte Funktionalität der Domänenlogik bedienbar machen.
- Keine Frameworks (z.B. Spring), keine relationale Datenbank.
- Einsatz von Bibliotheken, die UI-Technologie und eine allfällige Client-Server Anwendung muss mit Fachdozierendem abgesprochen werden.

# Benotung

- Modulnote = Teamnote + Individualbewertung + Teambeitrag
- Teamnote
  - Projektskizze (M1) (20%)
    - Fachliche Aspekte (10%)
    - Sprachliche Aspekte (10%)
  - Lösungsarchitektur (M2) (30%)  
Technischer Bericht I
  - Prototyp (M3) (30%)
  - Technischer Bericht II (20%)
    - Fachliche Aspekte (10%)
    - Sprachliche Aspekte (10%)

# Individualbewertung (fakultativ)

---

- Projektleiter bewertet Teamarbeit jedes seiner Teammitglieder
- Teammitglieder bewerten Projektmanagement des Projektleiters
- In 0.5 Schritten, zwischen -1 und +1
- Summe im Team muss 0 ergeben
- **Anträge** schriftlich mit Begründung 1 Woche vor Schlussbesprechung

**Dozierende entscheiden abschliessend über deren Anwendung!**

# Team-Beitrag (fakultativ)

---

- Dozierende bewerten Teamarbeit jedes Teammitglieds
- In 0.5 Schritten, zwischen -1 und +1
- Werte  $\neq 0$  sind die Ausnahme!

# Wichtige Termine und Abgaben

- Siehe detailliertes Wochenprogramm auf Moodle mit den Abgabezeitpunkten
- Abgabe/Präsentation Projektskizze (M1)
- Abgabe/Präsentation Lösungsarchitektur (M2)
- Abgabe/Präsentation Prototyp (M3)
- Abgabe-Modalitäten:
  - auf Moodle im entsprechenden Team-Ordner
  - Quellcode als eine ZIP-Datei oder über GitHub
  - Alle anderen Artefakte als PDF-Dokumente
  - Namenskonvention: *PM3-HS24-<Klasse>\_ZH(WIN)-<Dokument>-Team<Nr.>*  
(z.B. *PM3-HS24-IT23a\_WIN-Projektskizze-Team1*)

- Jedes Team arbeitet mit einem eigenen GitHub Repository:
  - Quellcode des Projektes auf GitHub der ZHAW: <https://github.engineering.zhaw.ch/>
  - Fachdozent berechtigen
  - Namenskonvention für das Repository:  
*PM3-HS24-<Klasse>\_ZH(WIN)-Team<Nr.>* (z.B. *PM3-HS24-IT23a\_WIN-Team1*)



# NoTechS/Sprachliche Lernziele

---

- Teammitarbeit
- Projektmanagement
- Mündliche und schriftliche Kommunikation
  
- Rolle Sprachdozierende
  - Bewertung der sprachlichen Aspekte der Projektskizze
  - Bewertung der sprachlichen Aspekte des technischen Berichts II
  - Feedback zu Präsentation

# NoTechS in PM3

---

- Effektives Arbeiten im Team
- Reflexion und Weiterentwicklung der eigenen Rolle im Team
  - Projektleitung
  - Teammitglied
- Adressatengerechte schriftliche Kommunikation
  - Projektskizze, technischer Bericht
  - Vorbereitung auf PA/BA

# Agenda

---

1. Einführung
- 2. Erklärung Projektergebnisse**
3. Input Schreiben 1
4. Anwesenheitskontrolle
5. Teamauslösung
6. Erster Auftrag

# Vorgaben und Guidelines

---

- Auftrag Projektskizze (M1)
  - Auftrag Lösungsarchitektur (M2)
  - Auftrag Prototyp (M3)
  - Auftrag Präsentationen (M1, M2, M3)
  - Auftrag Iterations-Review
- 
- Alle Vorgaben und weitere unterstützende Unterlagen sind auf [Moodle](#) abgelegt.

# Agenda

---

1. Einführung
2. Erklärung Projektergebnisse
3. **Input Schreiben 1**
4. Anwesenheitskontrolle
5. Teamauslösung
6. Erster Auftrag

# Agenda

---

1. Einführung
2. Erklärung Projektergebnisse
3. Input Schreiben 1
4. **Anwesenheitskontrolle**
5. Teamauslösung
6. Erster Auftrag

# Agenda

---

1. Einführung
2. Erklärung Projektergebnisse
3. Input Schreiben 1
4. Anwesenheitskontrolle
- 5. Teamauslösung**
6. Erster Auftrag

# Warum werden die Teams ausgelost?

- Ist das normale Szenario in der Praxis – Softwareentwicklungsteams werden für ein Projekt in der Regel neu zusammengestellt!
- Nur durch eine zufällige Auslosung können auch die NoTechS angemessen gestärkt werden (aus der «Komfort-Zone» herausgehen).
- Es kann auch Spass machen, wieder einmal mit anderen Kommilitonen zusammenzuarbeiten😊



# Agenda

---

1. Einführung
2. Erklärung Projektergebnisse
3. Input Schreiben 1
4. Anwesenheitskontrolle
5. Teamauslösung
6. **Erster Auftrag**

# Erster Auftrag an die Teams

---

- Brainstorming Projektidee
- Wahl PL
- Bis nächste Woche muss Projektidee festgelegt sein!
- *Bitte rapportieren Sie als Team kurz den Stand der Arbeiten dem Fachdozierenden bevor Sie gehen.*