LEHR- UND FORSCHUNGSEINHEIT
PROGRAMMIERUNG &
SOFTWARETECHNIK

# Softwaretechnik 2015/2016

PST Lehrstuhl

Prof. Dr. Matthias Hölzl

Joschka Rinke

# AGENDA

**Übung 1:**
**15.10.2015**

- **Organisatorisches**
- **Ausblick**
- **Wiederholung**

# Organisatorisches

- **Anmeldung zur Vorlesung über UniWorX**
- **Donnerstags zwei Übungstermine (10-12 Uhr & 12-14 Uhr)**
- **Vorlesungsseite (regelmäßig) besuchen**
- **Fragen bitte im die-informatiker Forum oder in der facebook-Gruppe (Softwaretechnik LMU WS 15/16) stellen**

# Organisatorisches

- **Klausur- & Nachholklausurtermin stehen noch nicht fest**
- **Bereits bestehende Klausurtermine anderer Veranstaltungen per E-Mail an mich (Betreff: „Klausurtermine“)**
- **Termin wird auf VL-Seite bekannt gegeben**
- **Anmeldung zur KL erfolgt (später) über UniWorX**
- **Sprechstunde: Mo 14-16 Uhr**

# Organisatorisches

**Übungsblätter:**

- **Heute kein Übungsblatt**
- **Nicht bewertet, nicht korrigiert**
- **Lösung wird in Übung erarbeitet**
- **Erscheinen (wenn möglich) freitags auf VL-Seite,**
  **Ausnahmen werden auf VL-Seite bekanntgegeben**

# Wiederholung & Ausblick

**Heute:**

- **Kurze Wiederholung zu Vorlesung**
  - **→ klassische (traditionelle) & agile Ansätze**
- **Kurzer Ausblick**
  - **→ Anforderungsanalyse (Use Cases)**
  - **→ Architektur & Design**
  - **→ Design-Patterns**
  - **→ Implementierung**
  - **→ Testen**

# Wiederholung - Entwicklungsprozesse

**Softwaretechnik ist die**
**„systematische Vorgehensweise für die Entwicklung, den Betrieb und die Stilllegung von Software“**
**Modellierung der Prozesse:**

- **Klassische (traditionelle) Ansätze für Entwicklungsprozesse**
  **„industrielle Herangehensweise“**
- **Agile Ansätze für Entwicklungsprozesse**
  **„kreativer Prozess“**

# Wiederholung - Entwicklungsprozesse

- **Klassische Ansätze: (Adaption aus der Bauindustrie)**
  **meist klar definierte Phasen mit eindeutigen Ergebnissen (Deliverables)**
  **z.B. Wasserfallmodell, iterativer Prozess, Spiralmodell**
- **Agile Ansätze: (basierend auf agilem Manifest)**
  **stärken der menschlichen Aspekte:**
  - **→ Individuen & Interaktion > Prozesse & Werkzeuge**
  - **→ funktionierende SW > umfassende Dokumentation**
  - **→ Zusammenarbeit mit Kunden > Vertragsverhandlungen**
  - **→ Reagieren auf Veränderung > Befolgen eines Plans**

  **z.B. Scrum, XP**

# Wiederholung - Entwicklungsprozesse

- **Klassische Ansätze:**
  - **+ Fortschritt überwachbar**
  - **+ eher unkompliziert**
  - **– starr und unflexibel**
  - **– späte Änderungen teuer**
  - **– viele Vorhersagen notwendig**
- **Agile Ansätze:**
  - **+ sehr flexibel**
  - **+ späte Änderungen unproblematischer**
  - **+ weniger Planung & Dokumentation**
  - **– sehr abhängig von Zusammensetzung des Teams**

# Ausblick – Anforderungsanalyse, Architektur, Design

- → **Use Cases erstellen**
- → **Unified Modeling Language (UML):**
  - **Standardisierte Modellierungssprache**
  - **Weit verbreitet**
  - **Tools zum Zeichnen → Visualisierung**
    **→ Dokumentation**
  - **Struktur Diagramme:**
    **Klassendiagramm, Objektdiagramm, …**
  - **Verhaltens Diagramme:**
    **Sequenzdiagramm, Aktivitätsdiagramm, …**

# Ausblick – Design Patterns, Implementierung, Testen

- **Design Patterns (Entwurfsmuster):**
  **Bewährte Lösungsansätze für wiederkehrende Entwurfsprobleme in der Softwareentwicklung**
  **z.B. Observer, Visitor, Singleton, …**
- **Implementierung:**
  **Umsetzung des Entwurfs**
- **Testen**
  **Unit Test**
  **Integration Test**