

Ablauf

1. Einführung in PSE
Zeit für Fragen
2. Anmeldung und Zeitplan
Zeit für Fragen
3. Themenvorstellung
Zeit für Fragen

Eine Aufzeichnung der Veranstaltung findet sich später im ILIAS

Bei Fragen:

- ▶ Fragen im Chat stellen
- ▶ Fragen werden dann vorgelesen und beantwortet

Praxis der Softwareentwicklung – WS 2020/21

Prof. Dr. Gregor Snelting

LEHRSTUHL PROGRAMMIERPARADIGMEN



Fakultät für **Informatik**

1. Einführung in PSE
Zeit für Fragen
2. Anmeldung und Zeitplan
Zeit für Fragen
3. Themenvorstellung
Zeit für Fragen

Eine Aufzeichnung der Veranstaltung findet sich später im ILIAS

Bei Fragen:

- Fragen im Chat stellen
- Fragen werden dann vorgelesen und beantwortet

- Ziel: Entwicklung eines mittelgroßen Systems im Team mit objektorientierter Softwaretechnik

- Ziel: Entwicklung eines mittelgroßen Systems im Team mit objektorientierter Softwaretechnik
- Zielsystem: max 10kLOC
objektorientierter Entwurf (UML), Implementierung (Java/C++/C#, ...), Qualitätssicherung (z. B. JUnit)

- Ziel: Entwicklung eines mittelgroßen Systems im Team mit objektorientierter Softwaretechnik
- Zielsystem: max 10kLOC
objektorientierter Entwurf (UML), Implementierung (Java/C++/C#, ...), Qualitätssicherung (z. B. JUnit)
- Teilnehmer: 3. oder 4. Sem. BA Informatik
Voraussetzung: Grundbegriffe der Informatik, Programmieren, Softwaretechnik I, Lineare Algebra I
- **Empfehlung:** PSE erst, wenn alle Module aus 1./2. Semester bestanden sind
- PSE wird jedes Semester angeboten

- Umfang: 9 LP,
≈ 270 Arbeitsstunden / Teilnehmer,
≈ 2 Arbeitstage / Woche / Teilnehmer

- Pflichtveranstaltung im Rahmen der Soft Skills (2 LP) kann nur zusammen mit PSE belegt werden
- soll PSE auf 9 LP bringen; explizite Lernziele Teamfähigkeit, Sprach-/Kommunikationskompetenz, Projektplanung/-management

- Zeitplan: November 2020 – März 2021; 17 Wochen Praktikumsbetrieb nach Absprache
vorlesungsfreie Zeit muss zur Entzerrung genutzt werden, da sonst leicht Überlastung möglich
- 12 verschiedene Aufgabenstellungen von 11 Lehrstühlen
- objektorientiertes Phasenmodell verbindlich (vgl. Modulhandbuch)
- max. 30 Teams à 5-6 Studenten
Wünsche zu Teamzusammensetzung / Aufgabe werden nach Möglichkeit berücksichtigt
- Englische Projekte sind **freiwillig**

- Zeitplan: November 2020 – März 2021; 17 Wochen Praktikumsbetrieb nach Absprache
vorlesungsfreie Zeit muss zur Entzerrung genutzt werden, da sonst leicht Überlastung möglich
- 12 verschiedene Aufgabenstellungen von 11 Lehrstühlen
- objektorientiertes Phasenmodell verbindlich (vgl. Modulhandbuch)
- max. 30 Teams à 5-6 Studenten
Wünsche zu Teamzusammensetzung / Aufgabe werden nach Möglichkeit berücksichtigt
- Englische Projekte sind **freiwillig**
- Eventuell stehen nicht genügend Teilnehmerplätze zur Verfügung ⇒ Warteliste für nächstes Semester
- Bitte Webseite beachten:
<http://pp.ipd.kit.edu/lehre/WS202021/pse/>

Es gibt folgendes Zulassungsverfahren:

1. Teilnehmer melden sich im PSE-Verwaltungssystem an.
2. Gegenprüfung der formalen Voraussetzungen am IPD Snelting soweit möglich.
3. Rückmeldung an Betreuer, falls Gegenprüfung *nicht* erfolgreich.
4. Betreuer prüfen Notenspiegel der fraglichen Fälle.
5. Umverteilung der Teams in 2. PSE-Woche, falls Teams auseinander fallen.

- moderne Softwaretechnik ist wichtig für alle BA-Absolventen!
- vollständige Entwicklung eines größeren Systems
- Phasenmodell:
 1. Pflichtenheft
 2. Entwurf
 3. Implementierung
 4. Qualitätssicherung
 5. Abschlusspräsentation
- Phasenverantwortliche
- Teamarbeit (Teams à 5-6 Teilnehmer)
- durchgehend Objektorientierung
- Toolunterstützung, z. B. Eclipse, JUnit, ...

1. Pflichtenheft

Phasenziel

detaillierte Festlegung der Leistungsmerkmale eines Systems

Grundprinzipien

- Präzision
- Vollständigkeit
- Konsistenz

Vorgehen

- Systemmodell (grobe Übersicht), Systemumgebung (Hard/Software)
- vollständige funktionale Anforderungen
- GUI-Entwürfe (manuell oder programmiert)
- ausführliche Testfallszenarien

verlangt wird

Abgabe des Pflichtenheftes nach 3 Wochen;
Erläuterung im ersten Kolloquium

objektorientiert (UML)

Phasenziel

- Festlegung der Klassenstruktur
- Schnittstellendefinition der Klassen
- Beziehungen zw. Klassen (Vererbung, Assoziationen)
- Klassendiagramm, ausgewählte Sequenzdiagramme, evtl. Zustandsdiagramm
- Einsatz von Design Patterns, MVC

objektorientiert (UML)

Phasenziel

- Festlegung der Klassenstruktur
- Schnittstellendefinition der Klassen
- Beziehungen zw. Klassen (Vererbung, Assoziationen)
- Klassendiagramm, ausgewählte Sequenzdiagramme, evtl. Zustandsdiagramm
- Einsatz von Design Patterns, MVC

Grundprinzipien

- Geheimnisprinzip
- schwache Kopplung
- hohe Kohäsion
- Lokalitätsprinzip
- Wiederverwendbarkeit von Klassen/Subsystemen

objektorientiert (UML)

Phasenziel

- Festlegung der Klassenstruktur
- Schnittstellendefinition der Klassen
- Beziehungen zw. Klassen (Vererbung, Assoziationen)
- Klassendiagramm, ausgewählte Sequenzdiagramme, evtl. Zustandsdiagramm
- Einsatz von Design Patterns, MVC

Grundprinzipien

- Geheimnisprinzip
- schwache Kopplung
- hohe Kohäsion
- Lokalitätsprinzip
- Wiederverwendbarkeit von Klassen/Subsystemen
- OO: Vererbung/dynamische Bindung statt Fallunterscheidung

Vorgehen

- Kombination von Top-Down und Bottom-Up Design
- Identifikation von Klassen, Vererbung, Assoziationen
- Festlegung der Schnittstellen aller Klassen
- informelle Beschreibung aller Klassen
- evtl. Einsatz von Entwurfsmetriken

Vorgehen

- Kombination von Top-Down und Bottom-Up Design
- Identifikation von Klassen, Vererbung, Assoziationen
- Festlegung der Schnittstellen aller Klassen
- informelle Beschreibung aller Klassen
- evtl. Einsatz von Entwurfsmetriken

verlangt wird

- Abgabe der UML-Diagramme nebst informeller Beschreibung nach 4 Wochen; Verteidigung im zweiten Kolloquium
- *Nachweis der Evolutionsfähigkeit* (z. B. Lokalitätsprinzip)

3. Implementierung

Phasenziel: Programmierung des Systems

Grundprinzipien

- Programmierung in einer objektorientierten Sprache
- Umsetzung der Architektur

3. Implementierung

Phasenziel: Programmierung des Systems

Grundprinzipien

- Programmierung in einer objektorientierten Sprache
- Umsetzung der Architektur

Vorgehen

- Implementierungsplan vorher
- Implementierung der Methoden
- funktionaler Komponententest mit JUnit, evtl. Überdeckungstests; verschränkt mit Implementierung
- Realisation der Szenarien aus Pflichtenheft

verlangt wird

- Implementierungsplan; Implementierung;
- Implementierungskolloquium

4. Qualitätssicherung

Phasenziel: Test des Systems

Grundprinzipien

- werkzeugunterstützte Qualitätssicherung

4. Qualitätssicherung

Phasenziel: Test des Systems

Grundprinzipien

- werkzeugunterstützte Qualitätssicherung

Vorgehen

- Integrationstest, Robustheitstest
- Prüfen der Szenarien aus Pflichtenheft

verlangt wird

- Testbericht; Systemabnahme (1 Woche vor Abschluss)

5. Abschlusspräsentation

- Abschlusspräsentation (Frühjahr 2021)
- Nach Möglichkeit mehrere Lehrstühle zusammen



image source: <https://www.flickr.com/photos/svenwerk/506579282/>

Allgemeine Hinweise und Tipps

20 Seiten Hinweise und Tipps (im Laufe der letzten Jahre entstanden) für Betreuer und Studierende als PDF gibt es auf der PSE-Webseite zum Download:

<http://pp.ipd.kit.edu/lehre/WS202021/pse/>

Ihre Beiträge dazu nehmen wir gerne hier entgegen:
<https://git.scc.kit.edu/IPDSnelting/pse-tipps/>

Ablauf

1. Einführung in PSE
Zeit für Fragen
2. Anmeldung und Zeitplan
Zeit für Fragen
3. Themenvorstellung
Zeit für Fragen

Eine Aufzeichnung der Veranstaltung findet sich später im ILIAS

Bei Fragen:

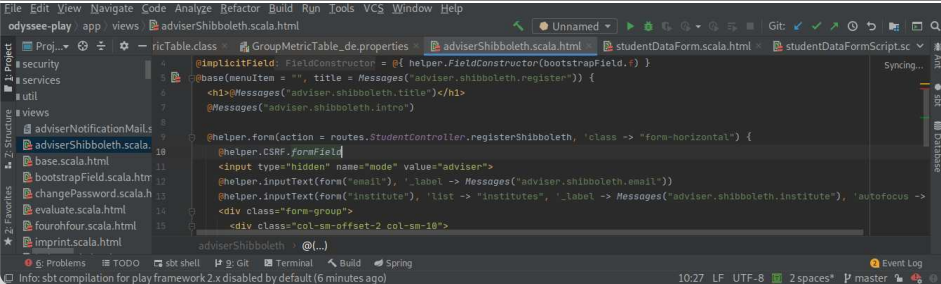
- ▶ Fragen im Chat stellen
- ▶ Fragen werden dann vorgelesen und beantwortet

Praxis der Softwareentwicklung – WS 2020/21

Auftaktveranstaltung, 5. November 2020

Prof. Dr. Gregor Snelting, Johannes Bechberger, Andreas Fried

LEHRSTUHL PROGRAMMIERPARADIGMEN



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
odyssee-play | app | views | adviserShibboleth.scala.html

Project
├── security
├── services
├── util
├── views
│   ├── adviserNotificationMail.s
│   ├── adviserShibboleth.scala
│   ├── base.scala.html
│   ├── bootstrapField.scala.htm
│   ├── changePassword.scala.h
│   ├── evaluate.scala.html
│   ├── fourohfour.scala.html
│   └── imprint.scala.html
└── Z: Structure

adviserShibboleth @ @(...)
4 @implicitField: FieldConstructor = @{ helper.FieldConstructor(bootstrapField.f) }
5 @base(menuItem = "", title = Messages("adviser.shibboleth.register")) {
6     <h1>@Messages("adviser.shibboleth.title")</h1>
7     @Messages("adviser.shibboleth.intro")
8
9     @helper.form(action = routes.StudentController.registerShibboleth, 'class -> "form-horizontal") {
10         @helper.CSRF.formField
11         <input type="hidden" name="mode" value="adviser">
12         @helper.inputText(form("email"), '_label -> Messages("adviser.shibboleth.email"))
13         @helper.inputText(form("institute"), '_list -> "institutes", '_label -> Messages("adviser.shibboleth.institute"), 'autofocus ->
14         <div class="form-group">
15             <div class="col-sm-offset-2 col-sm-10">
```

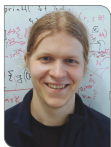
Info: sbt compilation for play framework 2.x disabled by default (6 minutes ago) 10:27 LF UTF-8 2 spaces* master

PSE-Homepage: <http://pp.ipd.kit.edu/lehre/WS202021/pse/>

E-Mail: pse-orga@lists.kit.edu

ILIAS: https://ilias.studium.kit.edu/goto.php?target=crs_1249351

Persönlich:



Johannes Bechberger



Andreas Fried



Prof. Dr.-Ing. Gregor Snelting

Allgemeine Fragen

- Keine Fragerunde im Anschluss an diese Präsentation
- ILIAS-Forum für allgemeine Fragen
- Personen-/Teambezogene Fragen an Betreuer und/oder `pse-orga@lists.kit.edu`



The screenshot shows the ILIAS interface for the course 'Praxis der Softwareentwicklung (PSE)'. At the top, there is a course icon, the title 'Praxis der Softwareentwicklung (PSE)', and an 'Aktionen' dropdown menu. Below the title is a description: 'Objektorientierte Entwicklung eines Softwaresystems im Team. Dieser Kurs dient zur Beantwortung allgemeiner Fragen die im Laufe des PSEs auftreten.' and a status indicator 'Status: Offline'. A navigation bar contains 'Inhalt', 'Info', and 'Lernfortschritt' tabs. The 'Inhalt' tab is active, showing a section titled 'Forum' with a description 'Forum für allgemeine Fragen' and 'Beiträge (Ungelesen): 0 (0)'. To the right, there is a 'Kalender' widget showing a calendar for April 2020, with the 30th highlighted. Below the calendar is an 'iCal' button.

Die Teams werden in **der nächsten Woche** eingeteilt

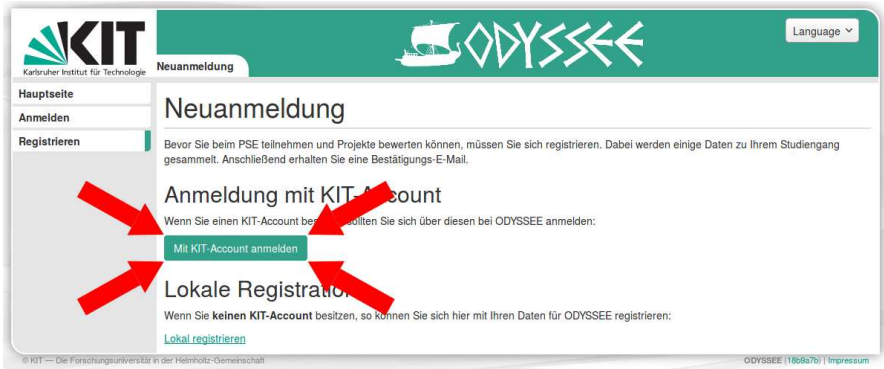
URL: <https://pse.informatik.kit.edu>

Eintragungen sind ab 17:30 Uhr **bis Sonntag** möglich

Anmeldung mit **KIT-Account** (u????)

Gruppen mit ≤ 5 Teilnehmern sind möglich

Ergebnis per E-Mail-Benachrichtigung und in ODYSSEE



The screenshot shows the 'Neuanmeldung' (New Registration) page of the ODYSSEE system. The page has a green header with the KIT logo and the ODYSSEE logo. A navigation menu on the left includes 'Hauptseite', 'Anmelden', and 'Registrieren'. The main content area is titled 'Neuanmeldung' and contains three sections: 'Neuanmeldung', 'Anmeldung mit KIT-Account', and 'Lokale Registrierung'. The 'Anmeldung mit KIT-Account' section has a green button labeled 'Mit KIT-Account anmelden' which is highlighted by four red arrows. The 'Lokale Registrierung' section has a blue link labeled 'Lokal registrieren'.

Neuanmeldung

Bevor Sie beim PSE teilnehmen und Projekte bewerten können, müssen Sie sich registrieren. Dabei werden einige Daten zu Ihrem Studiengang gesammelt. Anschließend erhalten Sie eine Bestätigungs-E-Mail.

Anmeldung mit KIT-Account

Wenn Sie einen KIT-Account besitzen, sollten Sie sich über diesen bei ODYSSEE anmelden:

[Mit KIT-Account anmelden](#)

Lokale Registrierung

Wenn Sie **keinen KIT-Account** besitzen, so können Sie sich hier mit Ihren Daten für ODYSSEE registrieren:

[Lokal registrieren](#)

© KIT — Die Forschungsuniversität in der Helmholtz-Gemeinschaft

ODYSSEE (18c9a7b) | Impressum

Prüfungsordnung

Erforderlich

Klausurergebnisse

Übungsschein Grundbegriffe der Informatik	<input type="text" value="bestanden"/>
Klausur Grundbegriffe der Informatik	<input type="text" value="bestanden"/>
Abschlussaufgaben Programmieren	<input type="text" value="warte auf Ergebnis"/>
Klausur Lineare Algebra I	<input type="text" value="bestanden"/>
Übungsschein Softwaretechnik I	<input type="text" value="bestanden"/>
Klausur Softwaretechnik I	<input type="text" value="warte auf mündliche Nachprüfung"/>
Klausur Lineare Algebra II	<input type="text" value="warte auf Ergebnis"/>
Klausur Höhere Mathematik I / Analysis I	<input type="text" value="bestanden"/>
Klausur Höhere Mathematik II / Analysis II	<input type="text" value="bestanden"/>
Klausur Algorithmen I	<input type="text" value="nicht begonnen/nicht bestanden"/>

Meine Kontaktdaten
veröffentlichen



Wenn Sie diese Option aktivieren, stimmen Sie zu, dass Ihre Teammitglieder Ihren Namen und Ihre E-Mail-Adresse sehen können. Im Gegenzug werden Ihnen die Namen und E-Mail-Adressen Ihrer Teammitglieder angezeigt.

Mit der Registrierung bestätigen Sie, dass Sie kein falschen bzw. unwahren Angaben gemacht haben. Es ist nur ein Konto pro Student erlaubt. Mehrfachregistrierungen werden nicht geduldet. Ein Verstoß gegen diese Regel oder das Angeben von falschen Informationen kann nach Ermessen der PSE-Administration bestraft werden und zum Ausschluss aus dem PSE führen.

Prüfungsordnung

Erforderlich

Klausurergebnisse **Übungsschein Grundbegriffe der Informatik**

Klausur Grundbegriffe der Informatik

Abschlussaufgaben Programmieren

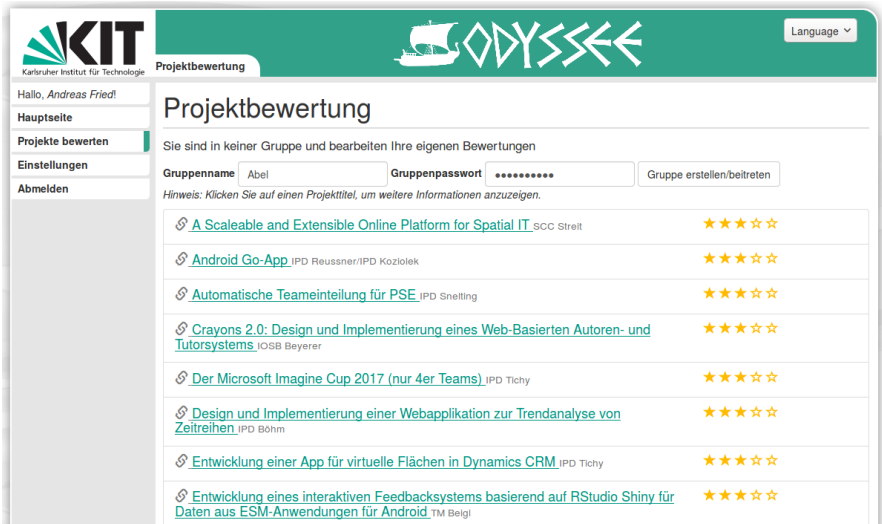
- Note ≤ 4.0 ausgehängt
- Einsicht genommen, Note ≤ 4.0 auf der Klausur
- Vorläufiges Ergebnis $\leq (4.0)$ im Campus-System

⇒ **bestanden**

Meine Kontaktdaten
veröffentlichen

wenn Sie diese Option aktivieren, bestätigen Sie zu, dass Ihre Teammitglieder Ihren Namen und Ihre E-Mail-Adresse sehen können. Im Gegenzug werden Ihnen die Namen und E-Mail-Adressen Ihrer Teammitglieder angezeigt.

Mit der Registrierung bestätigen Sie, dass Sie kein falschen bzw. unwahren Angaben gemacht haben. Es ist nur ein Konto pro Student erlaubt. Mehrfachregistrierungen werden nicht geduldet. Ein Verstoß gegen diese Regel oder das Angeben von falschen Informationen kann nach Ermessen der PSE-Administration bestraft werden und zum Ausschluss aus dem PSE führen.



The screenshot shows the 'Projektbewertung' (Project Evaluation) page of the ODYSSEE system. The header is green with the KIT logo on the left, the 'ODYSSEE' logo in the center, and a 'Language' dropdown on the right. A sidebar on the left contains navigation links: 'Hauptseite', 'Projekte bewerten' (highlighted), 'Einstellungen', and 'Abmelden'. The main content area has a greeting 'Hallo, Andreas Friedl!' and a title 'Projektbewertung'. Below the title, it states 'Sie sind in keiner Gruppe und bearbeiten Ihre eigenen Bewertungen'. There are input fields for 'Gruppenname' (containing 'Abel') and 'Gruppenpasswort' (masked with dots), and a 'Gruppe erstellen/beitreten' button. A hint says 'Hinweis: Klicken Sie auf einen Projekttitel, um weitere Informationen anzuzeigen.' Below this is a list of projects, each with a link icon, a title, a subtitle, and a 5-star rating.

Project Title	Subtitle	Rating
A Scaleable and Extensible Online Platform for Spatial IT	SCC Streit	★★★★★
Android Go-App	IPD Reussner/IPD Koziolek	★★★★★
Automatische Teameinteilung für PSE	IPD Snelting	★★★★★
Crayons 2.0: Design und Implementierung eines Web-Basierten Autoren- und Tutorsystems	IOSB Beyerer	★★★★★
Der Microsoft Imagine Cup 2017 (nur 4er Teams)	IPD Tichy	★★★★★
Design und Implementierung einer Webapplikation zur Trendanalyse von Zeitreihen	IPD Böhm	★★★★★
Entwicklung einer App für virtuelle Flächen in Dynamics CRM	IPD Tichy	★★★★★
Entwicklung eines interaktiven Feedbacksystems basierend auf RStudio Shiny für Daten aus ESM-Anwendungen für Android	TM Belgi	★★★★★

OpenPowerViz: POWERLINK Real-time Ethernet Visualization IAR Wörn	★★★★★
Privacy-Crashcam-App für Android JOSB Beyerer	★★★★★
Robot Health Monitoring IAR Asfour	★★★★★
Serveless System für mobile Internet der Dinge Anwendungen TM Beigl	★★★★★
Software Kompatibilität auf allen Ebenen IPD Tichy	★★★★★
Stromverbrauchsanalyse an einem realen HPC-System SCC Streit	★★★★★
Studienplanung als Generierung von Workflows mit Compliance-Anforderungen: Planerstellung und Visualisierung IPD Böhm	★★★★★
Werkzeug zur Analyse formaler Eigenschaften von Wahlverfahren ITI Beckert	★★★★★
Werkzeug zur Spezifikation und Verifikation von Software für Produktionsanlagen ITI Beckert	★★★★★
„Internet der Dinge“ Web-Dashboard JOSB Beyerer	★★★★★

Speichern

Prüfungsanmeldung: über das KIT-Studierendenportal

<https://campus.studium.kit.edu/>

Anmeldezeitraum: 5. – 8. November (in Ausnahmen bis 11.12.)

Danach keine An- und Abmeldung mehr möglich.

⇒ Aussteiger bekommen 5.0 (Keine Ausnahmen!)

Schritte

1. TSE (Nr. 7500075) zu überfachlichen Qualifikationen hinzufügen
2. zu TSE (Nr. 7500075) anmelden
3. zu PSE (Nr. 7500076) anmelden

fehlende Noten?

Falls Noten noch nicht eingetragen sind oder Nachprüfungen anstehen:

1. Auf jeden Fall in ODYSSEE anmelden
2. Mail an `pse-orga@lists.kit.edu`
3. Notenauszug an Betreuer schicken
4. Campus-Anmeldung schnellstmöglich nachholen

Erstes Kolloquium zwischen 14.12. und 18.12.

⇒ Anmeldeschluss 11.12.

**Bis 8. November Campus-Anmeldung oder E-Mail
Sonst keine Einteilung!**

- Das erste Treffen mit den Betreuern findet in der übernächsten Woche statt.
- Der genaue Termin und die Art variieren von Gruppe zu Gruppe.
- Entweder: Informationen stehen in der **Projektbeschreibung/Webseite**
- Oder: Ein Betreuer kontaktiert Sie per **E-Mail**

Fragen Sie im Zweifelsfall bei den Betreuern für ihr Thema nach.

Terminübersicht (exemplarisch)

Phase	von – bis	Dauer
Auftaktveranstaltung	05.11.	
Anmeldung/Einteilung	05.11. – 11.11.	
Erstes Gruppentreffen	16.11. – 20.11.	
Pflichtenheft	23.11. – 11.12.	3 Wochen
Entwurf	14.12. – 22.01.	4 Wochen
Implementierung	25.01. – 19.02.	4 Wochen
z.B. Klausurpause	22.02. – 05.03.	
Qualitätssicherung	08.03. – 26.03.	3 Wochen
interne Abnahme	29.03. – 02.04.	
Abschlusspräsentation	05.04. – 09.04.	

Terminübersicht (exemplarisch)

Phase	von – bis	Dauer
Auftaktveranstaltung	05.11.	
Anmeldung/Einteilung	05.11. – 11.11.	
Erstes Gruppentreffen	16.11. – 20.11.	
Pflichtenheft	23.11. – 11.12.	3 Wochen
Entwurf	14.12. – 22.01.	4 Wochen
Implementierung	25.01. – 19.02.	4 Wochen
z.B. Klausurpause	22.02. – 05.03.	
Qualitätssicherung	08.03. – 26.03.	3 Wochen
interne Abnahme	29.03. – 02.04.	
Abschlusspräsentation	05.04. – 09.04.	

Terminübersicht (exemplarisch)

Phase	von – bis	Dauer
Auftaktveranstaltung	05.11.	
Anmeldung/Einteilung	05.11. – 11.11.	
Erstes Gruppentreffen	16.11. – 20.11.	
Pflichtenheft	23.11. – 11.12.	3 Wochen
Entwurf	14.12. – 22.01.	4 Wochen
Implementierung	25.01. – 19.02.	4 Wochen
z.B. Klausurpause	22.02. – 05.03.	
Qualitätssicherung	08.03. – 26.03.	3 Wochen
interne Abnahme	29.03. – 02.04.	
Abschlusspräsentation	05.04. – 09.04.	

In jeder Gruppe: Verbindliche wöchentliche Treffen mit den Betreuern!

Unverbindliche Tipps & Tricks gibt es auf der PSE-Homepage zum Download:

<http://pp.ipd.kit.edu/lehre/WS202021/pse/>

- Versionskontrolle?
- UML-Tool?
- Dokumente schreiben?
- ...

Aktualisierungen oder Ergänzungen? Pull-Request!

<https://git.scc.kit.edu/IPDSnelting/pse-tipps>

Weitere Fragen: ILIAS-Forum

Ablauf

1. Einführung in PSE
Zeit für Fragen
2. Anmeldung und Zeitplan
Zeit für Fragen
3. Themenvorstellung
Zeit für Fragen

Eine Aufzeichnung der Veranstaltung findet sich später im ILIAS

Bei Fragen:

- ▶ Fragen im Chat stellen
- ▶ Fragen werden dann vorgelesen und beantwortet

Lehrstuhl	Thema	Teams
Fraunhofer IOSB	Toolkit zur Bewertung interpretierbarer KI-Methoden (XAI) auf Klassifikatoren für Zeitreihendaten	1
IAR Asfour	Developer and Visualization Tools for Robot Bus Systems	1
IES / Fraunhofer IOSB	Video Reality 4D	1
IES / Fraunhofer IOSB	Re-Build 4D	1
IES / Fraunhofer IOSB	Intelligent Video Sampler 3D	1
IPD Snelting	Visualisierung von Typinferenz	2
IVD Dachsbacher	Computergrafik in der Spieleentwicklung	1

Fraunhofer IOSB	Verteilte und transitive Spracherkennung mit Open-Source Frameworks	1
Fraunhofer IOSB	Modulare Web-Visualisierung für Industrie 4.0 Anlagen	1
Fraunhofer IOSB	Generator für Verwaltungsschalen-Modelle aus diversen Ausgangsmodellen	1
TM Abeck	Healthcare Asset Management	1
Fraunhofer IOSB	FR ² A: Ein App-Framework für den Kulturgüterschutz	1
IPD Koziolk	Skalierbarer Git-Client	1
SCC	Building a computer game 'from scratch'.	1
SCC Streit	WebApp for the monitoring of stratospheric ozone	1

IPD Reussner, IFV Vortisch	Code Your Mobility: Entwicklung einer Web- plattform zur Konfiguration von Verkehrs- nachfragesimulationen	1
SCC	OIDC/OAuth2-WebDAV CSI for Kubernetes	1
ITEC Tahoori	Virtueller Konferenzsimulator mit Teleprä- senz	1
ITEC	Graphical editor for "Liberty" files	1
TM	An Interactive Data Collection and Labeling Web-Tool with Meta-Learning	2
ITI Beckert, ITI Sinz	Entwicklung eines digitalen Klassenbuchs für den Einsatz an Schulen	1
ITEC	Broadcasting Webservice for Smart TVs	2
IPD Reussner, IPD Koziolk	Write Your Own Android App	3

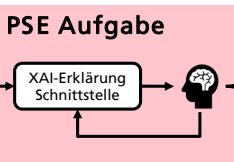
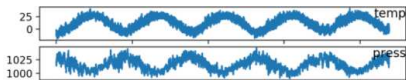
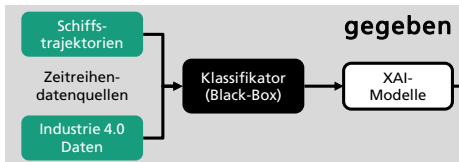
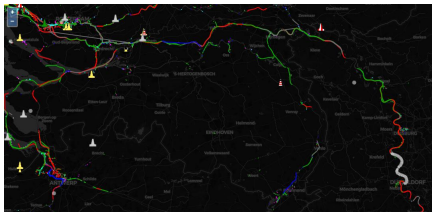
ITI Beckert, ITI Sinz	Erstellung und Training von Neuronalen Netzen durch visuelle Interaktion für den Schulunterricht	1
TM	Entwicklung eines 3D-Chatbots zur Emotionsregulation in Stresssituationen.	2

Fraunhofer IOSB

Toolkit zur Bewertung interpretierbarer KI-Methoden (XAI) auf Klassifikatoren für Zeitreihendaten

Teams: 1

Toolkit zur Bewertung interpretierbarer KI-Methoden (XAI) auf Klassifikatoren für Zeitreihendaten



Erklärungen für die internen Prozesse des Systems



Entwickler

Lokale und globale Erklärungen zur Modellvorhersage

Stärkung des Vertrauens der Benutzer in das System



Anwender

Toolkit zur Bewertung interpretierbarer KI-Methoden (XAI) auf Klassifikatoren für Zeitreihendaten

Ziel

- Entwicklung einer Applikation zur Verwaltung, Anwendung, Visualisierung und Bewertung von XAI-Erklärungen für Klassifikatoren von Zeitreihen

Aufgaben

- Entwicklung eines Systems zur Ausführung von XAI-Modellen auf Zeitreihendaten und zur intuitiven Darstellung der Ergebnisse
- Entwicklung einer Schnittstelle zur Auswertung von XAI-Erklärungen
- Verwendung von modell-agnostischen Verfahren zur Wiederverwendbarkeit auf neuen Modellen
- Web-basierte Benutzeroberfläche + Back-End-Entwicklung

IAR Asfour

Developer and Visualization Tools for Robot Bus Systems

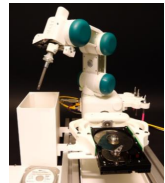
Teams: 1

Developer and Visualization Tools for Robot Bus Systems

- **Robots like ARMAR-6** use **bus systems** to read sensor values and control motors
- **Problem:** Sporadic bus failures which are hard to debug
Various potential reasons: hardware failure, software bugs, link losses, ...
- **Goal:** Develop a tool to monitor, log and visualize the status of the bus system in a robot
 - Display current status measurements (e.g. number of incoming/dropped messages)
 - Display bus topology (structure and status of connections)
 - Report useful error states and messages
 - Visualize the status and errors
 - Should be **easy to use for new robots**



ARMAR-6



KIT Gripper

Developer and Visualization Tools for Robot Bus Systems

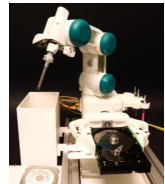
Goal: Develop a tool to monitor, log and visualize the status of the bus system in a robot

Tasks & Requirements:

- Monitoring bus system interface
 - Work with bus system standard **EtherCAT**
 - Use library **SOEM** [1] in **C/C++**
- User application in GUI framework **Qt 5** (C++)
 - Display status, messages, ...
 - Interactive data visualization (graphs, plots, topology, ...)
- Basic knowledge of C/C++ recommended



ARMAR-6



KIT Gripper

[1] Simple Open EtherCAT Master, <https://openethercatsociety.github.io/>

IES / Fraunhofer IOSB

Video Reality 4D

Teams: 1

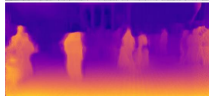
Video Reality 4D

- **CACTUS-3D** ist ein Softwarewerkzeug zur Generierung und Co-Registrierung von 3D Modellen aus Bilddaten
 - 3D Umgebungsmodelle, Tiefenkarten mittels neuronaler Netze, etc.
 - Bereits verfügbar aus früherem PSE (WS'15) + 5 Jahre Weiterentwicklung



- **Aufgabenstellung**

- **Mehrere zusammengesetzte 2D Videos sollen wieder erlebbar werden**
- Mit **Video Reality 4D** soll ein Werkzeug entstehen, welches 3D Rekonstruktionen von Umgebungsmodellen, monokularen Tiefenkarten sowie 3D Modelle von Personen über die Zeit animiert.
- Videos sollen über eine Zeitleise synchronisiert werden
 - wie bei Filmschnitt-Tools
- Interaktive Wiedergabe in 4D mit Zeitslider
- Echtzeit VR Anbindung 4D Szene mittels OpenVR



Team 1: Video Reality 4D

- Ziele
 - Effizientes Laden von 3D Modellen, Tiefenkarten, etc.
 - Effizientes Abspielen der 3D Modelle über die Zeit => 4D Animation
 - Möglichkeit zur manuellen Synchronisierung von verschiedenen zeitgleichen Videos
 - Gerenderte Szene als Video speichern
 - Manuelle Festlegung eines virtuellen Kamerapfad durch Szene mittels Keyframes
 - VR Modus
- Implementierung in C++
- Bibliotheken: Qt, OpenSceneGraph, OpenVR, QML



Beispiel: Final Cut Pro



Thomas Pollok, M. Sc. thomas.pollok@iosb.fraunhofer.de
Max Herrmann, M. Sc. max.herrmann@iosb.fraunhofer.de
Typ: Teilpräsenz, aber größtenteils Online über MS Teams

IES / Fraunhofer IOSB

Re-Build 4D

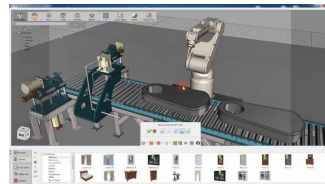
Teams: 1

Team 2: Re-Build 4D

- Ziele
 - Erzeugung eines interaktiven 3D Szenengraphen
 - Hinzufügen und entfernen von Objekten
 - Primitive Objekte oder fertige 3D Modelle
 - Beliebiges Platzieren, Rotieren und Skalieren
 - 3D Modellierung nach Grundriss Plan
 - Texturierung von Objekten und Platzierung von Lichtquellen
 - Animation von Objekten mittels Keyframes
 - Z.B. Roboterarm
- Implementierung in C++
- Bibliotheken: Qt, OpenSceneGraph, QML



Grundriss



Beispiel: SimLab



Thomas Pollok, M. Sc. thomas.pollok@iosb.fraunhofer.de
Max Herrmann, M. Sc. max.herrmann@iosb.fraunhofer.de
Typ: Teilpräsenz, aber größtenteils Online über MS Teams

IES / Fraunhofer IOSB

Intelligent Video Sampler 3D

Teams: 1

Team 3: Intelligent Video Sampler 3D

■ Ziele

- Werkzeug soll schnelles Zerlegen eines Videos in relevante Einzelbilder um eine Structure from Motion basierte 3D Rekonstruktion zu ermöglichen
 - Linear z.B. jedes n-te Frame, Visual-SLAM Basiert, GPS Metadaten-Basiert (Bei Drohne)
- Verwendung von semantischen KI Verfahren zum Erkennen von Vordergrund und Hintergrund
 - Dynamische Szenen sind schlechter als statische
- Automatisches erkennen von verschwommenen Bildern
- Ändern der Zielauflösung z.B. 4K => FullHD

■ Implementierung in C++

Bibliotheken: Qt, OpenSceneGraph, QML



Thomas Pollok, M. Sc. thomas.pollok@iosb.fraunhofer.de

Max Herrmann, M. Sc. max.herrmann@iosb.fraunhofer.de

Typ: Teilpräsenz, aber größtenteils Online über MS Teams



Nicht jedes Bild liefert neue, relevante Informationen um eine Szene zu rekonstruieren