

Organisatorisches

Julia Kreutzer, Katja Markert, Stefan Riezler

Computerlinguistik
Universität Heidelberg
Sommersemester 2019

Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:

Programmierprüfung

Teilnahme am Ressourcenvorkurs

Teilnahme am Blockkurs joeynmt (15.04.-18.04.) für Projekte
bei Riezler/Kreutzer

- Dauer: 1 Semester

- 6 LP → ca. 12h/Woche + 4 LP ÜK



Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:

- Programmierprüfung

- Teilnahme am Ressourcenvorkurs

- Teilnahme am Blockkurs joeynmt (15.04.-18.04.) für Projekte bei Riezler/Kreutzer

- Dauer: 1 Semester

- 6 LP → ca. 12h/Woche + 4 LP ÜK

Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:

Programmierprüfung

Teilnahme am Ressourcenvorkurs

Teilnahme am Blockkurs joeynmt (15.04.-18.04.) für Projekte
bei Riezler/Kreutzer

- Dauer: 1 Semester

- 6 LP → ca. 12h/Woche + 4 LP ÜK



Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:

Programmierprüfung

Teilnahme am Ressourcenvorkurs

Teilnahme am Blockkurs joeynmt (15.04.-18.04.) für Projekte
bei Riezler/Kreutzer

- Dauer: 1 Semester

- 6 LP → ca. 12h/Woche + 4 LP ÜK

Heute

① Inhalte & Ziele

② Semesterplan

③ Benotung

④ Themen



Inhalte

Im Softwareprojekt soll eine computerlinguistische Aufgabenstellung

- ... weitgehend eigenverantwortlich und in Teamarbeit
- ... geplant,
- ... programmiert,
- ... getestet,
- ... dokumentiert und
- ... präsentiert werden.

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ➊ aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ➋ den Plan in Teamarbeit umsetzen.
- ➌ eure Projektergebnisse präsentieren und analysieren



Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
 - den Ansatz in eigenen Worten wiedergeben.
 - zu implementierende Module festlegen und Abhängigkeiten zwischen den Modulen definieren.
 - Aus dem Modulplan einen Zeitplan und eine Aufgabenverteilung entwickeln.
 - Programmarchitektur, Datenstrukturen und Schnittstellen festlegen.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ➊ aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ➋ den Plan in Teamarbeit umsetzen.
- ➌ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
 - die euch zugewiesenen Module plangemäß implementieren,
 - eure Module so testen und dokumentieren, dass andere Teammitglieder sie verwenden können.
 - **gemeinsame** Instrumente entwickeln und einsetzen, die sicherstellen, dass ihr den Plan einhaltet, und dass ihr Fragen und Probleme rechtzeitig klärt.
- ③ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten.
- ② den Plan in Teamarbeit umsetzen.
- ③ eure Projektergebnisse präsentieren und analysieren
 - Ergebnisse und Methoden so präsentieren und demonstrieren, dass andere Teams verstehen, was ihr gemacht habt.
 - Schwächen eurer Implementierung und Probleme beim Projektablauf identifizieren und Verbesserungsvorschläge/Gründe angeben



Teamarbeit

“gemeinsame Instrumente entwickeln und einsetzen, die sicherstellen, dass ihr den Plan einhaltet, und dass ihr Fragen und Probleme rechtzeitig klärt.”

- regelmäßige Treffen - in der Gruppe und mit dem Betreuer
- gemeinsames “Logbuch”, z.B. Wiki oder gemeinsames Dokument
- gemeinsamer Terminkalender
- wöchentliche Statusreports aller Teilnehmer
- Codereviews
- Skype o.ä.
- Einsetzen eines/r Projektmanagers/in

Teamarbeit

“gemeinsame Instrumente entwickeln und einsetzen, die sicherstellen, dass ihr den Plan einhaltet, und dass ihr Fragen und Probleme rechtzeitig klärt.”

- regelmäßige Treffen - in der Gruppe und mit dem Betreuer
- gemeinsames “Logbuch”, z.B. Wiki oder gemeinsames Dokument
- gemeinsamer Terminkalender
- wöchentliche Statusreports aller Teilnehmer
- Codereviews
- Skype o.ä.
- Einsetzen eines/r Projektmanagers/in



Semesterplan

`www.cl.uni-heidelberg.de/courses/ss19/
softwareprojekt/material/`



Projektanmeldung

Bitte schickt bis Donnerstag, **18.4., 19 Uhr**, eine Mail an `{kreutzer|markert|riezler}@cl.uni-heidelberg.de` mit

- 1 Betreff: "SWP Projektanmeldung"
- 2 Zwei Wunschprojekten mit Priorität
- 3 Falls ihr schon im Vorfeld eine Gruppe bilden wollt, die Namen der anderen Gruppenmitglieder.



Forschungsplan: Abgabe 14.5.

- Erste, übergreifende Projektbeschreibung: Ziel, Methode, Evaluierung, Daten, Tools
- Beschreibt die Problemstellung (**Ziel**) und den Lösungsansatz (**Methode**) in eigenen Worten!
- Wie plant ihr Ergebnisse zu **evaluieren**?
- Welche **Tools** und **Daten** sollen verwendet werden?
- *Abgabe schriftlich, ca. 2 Seiten*



Spezifikationsvortrag: 21.5.

- **Inhaltliche Spezifikation:**

- Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
- Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen

- **Modularisierung und Aufgabenverteilung:**

- Definition von Modulen/Aufgaben und Abhängigkeiten
- Zuordnung von Namen zu Aufgaben
- Zeitplan

- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen

- *Vortrag im Plenum, maximal 25 Minuten*



Spezifikationsvortrag: 21.5.

- **Inhaltliche Spezifikation:**

- Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
- Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen

- **Modularisierung und Aufgabenverteilung:**

- Definition von Modulen/Aufgaben und Abhängigkeiten
- Zuordnung von Namen zu Aufgaben
- Zeitplan

- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen

- *Vortrag im Plenum, maximal 25 Minuten*



Spezifikationsvortrag: 21.5.

- **Inhaltliche Spezifikation:**
 - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
 - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - Definition von Modulen/Aufgaben und Abhängigkeiten
 - Zuordnung von Namen zu Aufgaben
 - Zeitplan
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*



Spezifikationsvortrag: 21.5.

- **Inhaltliche Spezifikation:**
 - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
 - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - Definition von Modulen/Aufgaben und Abhängigkeiten
 - Zuordnung von Namen zu Aufgaben
 - Zeitplan
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*



Spezifikationsvortrag: 21.5.

- **Inhaltliche Spezifikation:**
 - Problemstellung, Lösungsansatz, Evaluation (→ *Forschungsplan*)
 - Auswahl/Sichtung benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - Definition von Modulen/Aufgaben und Abhängigkeiten
 - Zuordnung von Namen zu Aufgaben
 - Zeitplan
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 25 Minuten*



Statusmeetings

- Individuelle Termine für jede Gruppe mit Betreuer.
- Bei Bedarf wöchentlich zur Seminarzeit
- Keine Meetings außerhalb der Seminarzeit - *come prepared!*



Abschlussvortrag: 23.7.

- Wie wurde der Lösungsansatz umgesetzt?
- Präsentation der Evaluationsergebnisse
- Demo
- *Lessons Learned*: Identifikation von Schwächen eurer Implementierung und Problemen beim Projektablauf → Verbesserungsvorschläge/ Gründe angeben.
- *Vortrag im Plenum, maximal 25 Minuten*



Leistungsnachweis

Abgabe: 26.7. - **keine Ausnahme möglich!**

Voraussetzung für einen Schein:

- Teilnahme an allen gemeinsamen Veranstaltungen (Plenum)
- Forschungsplan
- Spezifikationsvortrag
- Abschlussvortrag + Demo, sofern sinnvoll
- Dokumentation und Archivierung des Projekts sowie schriftlicher Ergebnisbericht
 - Dokumentation des Quelltexts
 - Falls Projekt mit JoeyNMT: Pull Request für die Änderungen an JoeyNMT
 - README
 - Schriftlicher Ergebnisbericht (ungefähr 4 Seiten]

Leistungsnachweis

Abgabe: 26.7. - **keine Ausnahme möglich!**

Voraussetzung für einen Schein:

- Teilnahme an allen gemeinsamen Veranstaltungen (Plenum)
- Forschungsplan
- Spezifikationsvortrag
- Abschlussvortrag + Demo, sofern sinnvoll
- Dokumentation und Archivierung des Projekts sowie schriftlicher Ergebnisbericht
 - Dokumentation des Quelltexts
 - Falls Projekt mit JoeyNMT: Pull Request für die Änderungen an JoeyNMT
 - README
 - Schriftlicher Ergebnisbericht (ungefähr 4 Seiten]



Benotung

Benotungskriterien:

- ① Sorgfältige Herangehensweise bei Planung, Implementierung, Dokumentation
→ Evaluation durch Betreuer
- ② Präsentation des fertigen Projekts und Lessons Learned
→ Evaluation u.a. durch andere Projektgruppen
- ③ Zusammenarbeit im Team
→ Evaluation u.a. durch Teammitglieder

In Sonderfällen können Einzelnoten vergeben werden!



Vorstellung der Themen