

Softwareprojekt

Katja Markert und Tsz Kin Lam
mit Material von früheren Semestern

Computerlinguistik
Universität Heidelberg
Sommersemester 2022

Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
 - Prog I und II (Äquivalente für Informatiker)
 - ECL (nicht für Informatiker)
 - Statistische Methoden (für alle)
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

Inhalt

Selbständige, eigenverantwortliche Durchführung einer NLP/ML Aufgabe im Team

- Planung
- Implementierung
- Testen
- Software Packaging und Dokumentation
- Projektpräsentation

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einer abstrakten Idee einen Projektplan erarbeiten
- ② den Plan in konkrete Arbeitspakete herunterbrechen
- ③ die Arbeitspakete aufteilen, zeitlich abstimmen, zuweisen und als Team ausführen
- ④ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ➊ aus einer abstrakten Idee einen Projektplan erarbeiten
- ➋ den Plan in konkrete Arbeitspakete herunterbrechen
- ➌ die Arbeitspakete aufteilen, zeitlich abstimmen, zuweisen und als Team ausführen
- ➍ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einer abstrakten Idee einen Projektplan erarbeiten
- ② den Plan in konkrete Arbeitspakete herunterbrechen
- ③ die Arbeitspakete aufteilen, zeitlich abstimmen, zuweisen und als Team ausführen
- ④ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einer abstrakten Idee einen Projektplan erarbeiten
- ② den Plan in konkrete Arbeitspakete herunterbrechen
- ③ die Arbeitspakete aufteilen, zeitlich abstimmen, zuweisen und als Team ausführen
- ④ eure Projektergebnisse präsentieren und analysieren

Ziele

Am Ende des Softwareprojekts könnt ihr...

- ① aus einer abstrakten Idee einen Projektplan erarbeiten
- ② den Plan in konkrete Arbeitspakete herunterbrechen
- ③ die Arbeitspakete aufteilen, zeitlich abstimmen, zuweisen und als Team ausführen
- ④ eure Projektergebnisse präsentieren und analysieren

Von der Idee zum Projekt

- Gib die Aufgabe in eigenen Worten wieder
- Modularisiere die Aufgabe, definiere Abhängigkeiten zwischen Modulen
- Verteile die Aufgaben innerhalb des Teams
- Prioritisiere die Unteraufgaben
- Entwickle einen Zeitplan für die Projektlaufzeit (Meilensteine, Parallelisierung von Aufgaben, Deadlines)
- Definiere Programmarchitektur, Datenstrukturen und Schnittstellen.

Vom Plan zu konkreten Arbeitspaketen

- Entwerfe gemeinsame Bibliotheken und Routinen
- Implementiere die zugewiesenen Komponenten nach Zeitplan
- Dokumentiere und teste Module, damit andere Teammitglieder sie verwenden können
- Entscheidet über **gemeinsame** Instrumente zur Kommunikation und Fehlerverfolgung (Kalender, Wiki, Trello, Google Docs/Sheets, Slack, ...)

Ergebnispräsentation

- Definiert und beschreibt Projektziele klar
- Beschreibt Ergebnisse und Methoden
- Demonstration der Software
- Beschreibt Hauptschwierigkeiten und Probleme
- Beschreibt mögliche Verbesserungen und zukünftige Arbeit

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
- Dokumentation der Vereinbarungen in gemeinsamem “Logbuch”, z.B. Wiki, gemeinsamer Terminkalender
- Wöchentliche Statusreports aller Teilnehmer
- Gemeinsame Codereviews
- Regelmäßige Restrukturierung des Codes (refactoring), um die Lesbarkeit und Wartbarkeit zu verbessern
- 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
- Dokumentation der Vereinbarungen in gemeinsamem “Logbuch”, z.B. Wiki, gemeinsamer Terminkalender
- Wöchentliche Statusreports aller Teilnehmer
- Gemeinsame Codereviews
- Regelmäßige Restrukturierung des Codes (refactoring), um die Lesbarkeit und Wartbarkeit zu verbessern
- Einsetzen eines/r Projektmanagers/in

Semesterplan

<https://www.cl.uni-heidelberg.de/courses/ss22/softwareprojekt/material/>

26.4.2022	Organisation, Projektvorstellung
30.4.2022, Sa	Email mit Projektwünschen
3.5.2022	Projekteinteilung
10.5.2022	Kickoff-Meeting I
17.5.2022	Kickoff-Meeting II
23.5.2022, Mo	Forschungsplan Einreichung
24.5.2022	status meeting
31.5.2022	status meeting
7.6.2022	Spezifikationsvorträge (Plenum, SR24)
14.6.2022	status meeting
...	...
19.7.2022	status meeting
26.7.2022	Abschlussvorträge, Demos (Plenum, SR24)
5.8.2022, Fr	Abgabe

Wichtig: Sprache

Dieses Jahr bitte alles auf Englisch!

Projektanmeldung

Bitte schicken Sie bis Samstag, **30.4., 12 Uhr Mittags**, eine englische Mail an `{markert|lam}@cl.uni-heidelberg.de` mit

- 1 Betreff: "SWP Sign Up"
- 2 Mind. zwei Wunschprojekten mit Priorität
- 3 Falls Sie schon im Vorfeld eine Gruppe bilden wollen, die Namen der anderen Gruppenmitglieder.
- 4 Programmierkenntnisse: Sprachen (1 - 5, 1 gar nicht, 5 Experte)
- 5 Kenntnissen und Erfahrung in Statistik, maschinellem Lernen, neuronalen Netzen
- 6 Fachrichtung und Fachsemester

Gruppeneinteilung: bis zum 3.5.

- 3-5 Mitglieder je Gruppe
- Gruppen können gern im Vorfeld gebildet werden
- Bei Interessenkonflikten versuchen wir zu schlichten
- Projektzuweisung, Literaturzuweisung
- Jeder Teilnehmer bekommt ein Projekt
- Einige Projekte können auch an zwei Gruppen vergeben werden, die unterschiedliche Ansätze zum gleichen Problem verwenden

Die ersten beiden Kickoff-Meetings: 10.5./17.5.

- Literaturdiskussion
- Unklarheiten benennen, Fragen formulieren, Ressourcen identifizieren
- Plan erarbeiten: genaue Aufgabenstellung, Teilaufgaben, Zeitplan, Verantwortlichkeiten

Forschungsplan:

Abgabe Mo 23.5. 12 Uhr; Besprechung 24.5.

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

Status Meetings: 24.5/31.5.

- Status der Softwarespezifikation

Spezifikationsvortrag: Di 7.6., 13.15 Uhr (Plenum, SR24)

- **Inhaltliche Spezifikation:**
 - * Problemstellung, Lösungsansatz, Evaluation (*siehe Forschungsplan*)
 - * Auswahl benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - * Definition von Modulen/Aufgaben
 - * Aufgabenverteilung an Teammitglieder
 - * Zeitplan (Parallelisierung!)
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag mit Folien im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: Di 7.6., 13.15 Uhr (Plenum, SR24)

- **Inhaltliche Spezifikation:**
 - Problemstellung, Lösungsansatz, Evaluation (*siehe Forschungsplan*)
 - Auswahl benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - * Definition von Modulen/Aufgaben
 - * Aufgabenverteilung an Teammitglieder
 - * Zeitplan (Parallelisierung!)
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag mit Folien im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: Di 7.6., 13.15 Uhr (Plenum, SR24)

- **Inhaltliche Spezifikation:**
 - Problemstellung, Lösungsansatz, Evaluation (*siehe Forschungsplan*)
 - Auswahl benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - Definition von Modulen/Aufgaben
 - Aufgabenverteilung an Teammitglieder
 - Zeitplan (Parallelisierung!)
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag mit Folien im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: Di 7.6., 13.15 Uhr (Plenum, SR24)

- **Inhaltliche Spezifikation:**
 - Problemstellung, Lösungsansatz, Evaluation (*siehe Forschungsplan*)
 - Auswahl benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - Definition von Modulen/Aufgaben
 - Aufgabenverteilung an Teammitglieder
 - Zeitplan (Parallelisierung!)
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag mit Folien im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: Di 7.6., 13.15 Uhr (Plenum, SR24)

- **Inhaltliche Spezifikation:**
 - Problemstellung, Lösungsansatz, Evaluation (*siehe Forschungsplan*)
 - Auswahl benötigter Ressourcen und Verfahren/Algorithmen
- **Modularisierung und Aufgabenverteilung:**
 - Definition von Modulen/Aufgaben
 - Aufgabenverteilung an Teammitglieder
 - Zeitplan (Parallelisierung!)
- **konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag mit Folien im Plenum, maximal 20 Minuten*

Statusmeetings

- Individuelle Termine für jede Gruppe mit Betreuer (ca. 45 Minuten)
- Bei Bedarf wöchentlich zur Seminarzeit (Di, 13-15 Uhr).
- Keine Meetings außerhalb der Seminarzeit - *come prepared!*
- Zusätzlich: Treffen der Teams ohne Betreuer (ca. 2 mal wöchentlich)

Abschlussvortrag: 26.7., 13.15 Uhr (Plenum, SR24)

- Wie wurde der Lösungsansatz umgesetzt?
- Präsentation: Ziele, Methoden und (Schwerpunkt:) Evaluationsergebnisse
- Demo
- *Lessons Learned:*
 - Welche anfänglichen Hypothesen haben sich als falsch erwiesen?
 - Wodurch habt Ihr sie ersetzt?
 - Was hätte man besser machen können? Identifikation von Schwächen eurer Implementierung und Problemen beim Projektablauf
 - *Vortrag mit Folien, maximal 25 Minuten + 10 Minuten Fragen*

Deadline für das gepackte Projekt: 5.8.2022

- Harte Deadline: keine Ausnahme möglich
- Abgabe per email
- Projekt kommt mit README, Installationshinweisen und muss funktionsfähig sein
- Gerne: GitLab/GitHub und institutsinterne Freigabe
- Lizenzierung, falls Projekt veröffentlicht werden soll
- Dokumentation des Quellcodes
- Schriftlicher Ergebnisbericht (je nach Projekt, 4-5 Seiten)

Leistungsnachweis

- Teilnahme an allen gemeinsamen Veranstaltungen (Plenum, SR24)
- Forschungsplan
- Spezifikationsvortrag
- Abschlussvortrag + Demo
- Dokumentation und Archivierung des Projekts
 - Dokumentation des Quelltexts
 - README
 - Abschlussbericht

Benotung

Benotungskriterien:

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

In Sonderfällen können Einzelnoten vergeben werden!

Ressourcen@CL

- Nutzen Sie die vorhandenen Ressourcen:
 - <https://wiki.cl.uni-heidelberg.de/bin/view/Main/Resources/WebHome>
 - Ansprechpartner: Letitia Parcalabescu, resources@cl.uni-heidelberg.de
- Software Styleguide: <https://wiki.cl.uni-heidelberg.de/bin/view/Main/Resources/SoftwareStyleguide>
- Bitte ignorieren Sie die Deadlines am Ende des Software Style Guides. Der 5.8.2022 ist Ihre Deadline!
- Verwenden Sie ein Versionskontrollsystem (git!)