

# Softwareprojekt

Katja Markert und Julius Steen  
mit Material von früheren Semestern

2024

# Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- 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 und zuweisen (und 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 und zuweisen (und ausführen)
- ➍ 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 und zuweisen (und 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 und zuweisen  
(und 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 und zuweisen (und 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

## 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

# Projektanmeldung

Anmeldung per Mail an `markert@c1.uni-heidelberg.de` **und(!)** `steen@c1.uni-heidelberg.de` bis zum **21.04., 23:59 Uhr**, mit:

- 1 Eurem Namen, Studienfach und Fachsemester
- 2 (mindestens) zwei Wunschprojekten mit Priorität
- 3 Falls Ihr schon im Vorfeld eine Gruppe bilden wollt, die Namen der anderen Gruppenmitglieder
- 4 Programmiersprachen mit Kompetenzeinschätzung (1: wenig Erfahrung; .... 5: Experte)

## Gruppeneinteilung: vor dem 23.04.

- 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

## Die ersten beiden Meetings: 23.04./30.04.

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



## Forschungsplan:

Abgabe Mo 6.5. 12 Uhr; Besprechung 7.5.

- 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*

## Spezifikationsvortrag: Di 14.05., 13.15 Uhr (Plenum)

- **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 25 Minuten*

## Spezifikationsvortrag: Di 14.05., 13.15 Uhr (Plenum)

- **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 25 Minuten*

## Spezifikationsvortrag: Di 14.05., 13.15 Uhr (Plenum)

- **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 25 Minuten*

## Spezifikationsvortrag: Di 14.05., 13.15 Uhr (Plenum)

- **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 25 Minuten*

## Spezifikationsvortrag: Di 14.05., 13.15 Uhr (Plenum)

- **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 25 Minuten*

## Statusmeetings

- Individuelle Termine für jede Gruppe mit Betreuer (ca. 45 Minuten)
- Bei Bedarf wöchentlich zur Seminarzeit (Di, 13-17 Uhr).
- Keine Meetings außerhalb der Seminarzeit - *come prepared!*
- Zusätzlich: Treffen der Teams ohne Betreuer (ca. 2 mal wöchentlich)

## Abschlussvortrag: 16.07., 13.15 Uhr (Plenum)

- 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: 26.07.

- Harte Deadline: keine Ausnahme möglich
- Abgabe per email
- Projekt kommt mit Readme und muss funktionsfähig sein
- Gerne: GitLab und institutsinterne Freigabe

## Leistungsnachweis

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

# 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-...](mailto:resources@cl.uni-...)
- Software Styleguide: <https://wiki.cl.uni-heidelberg.de/bin/view/Main/Resources/SoftwareStyleguide>
- Verwenden Sie ein Versionskontrollsystem (git!)