

Organisatorisches zum Softwareprojekt

Anette Frank, Raphael Schumann

Computerlinguistik
Universität Heidelberg
Wintersemester 2023/24

Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
 - Erfolgreiche Teilnahme an ICL, PII, FF-SM
 - Bestehen der Orientierungsprüfung
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
 - Erfolgreiche Teilnahme an ICL, PII, FF-SM
 - Bestehen der Orientierungsprüfung
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
 - Erfolgreiche Teilnahme an ICL, PII, FF-SM
 - Bestehen der Orientierungsprüfung
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

Überblick

Herzlich Willkommen zur Begleitveranstaltung zum Softwareprojekt

- Voraussetzungen:
 - Erfolgreiche Teilnahme an ICL, PII, FF-SM
 - Bestehen der Orientierungsprüfung
- Dauer: 1 Semester
- 6 LP → ca. 12h/Woche + 4 LP ÜK

Inhalte

Im Softwareprojekt soll eine computerlinguistische
Aufgabenstellung

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

Ziele

Am Ende des Softwareprojekts können Sie...

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

Ziele

Am Ende des Softwareprojekts können Sie...

- 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 Ihre Projektergebnisse präsentieren und analysieren:

Ziele

Am Ende des Softwareprojekts können Sie...

- ① aus einem theoretischen Ansatz einen konkreten Projektplan erarbeiten;
- ② den Plan in Teamarbeit umsetzen:
 - die Ihnen zugewiesenen Module plangemäß implementieren;
 - Module so testen und dokumentieren, dass andere Teammitglieder sie verwenden können;
 - **gemeinsame** Instrumente entwickeln und einsetzen, die sicherstellen, dass Sie den Plan einhalten, und dass Sie Fragen und Probleme rechtzeitig klären.
- ③ Ihre Projektergebnisse präsentieren und analysieren:

Ziele

Am Ende des Softwareprojekts können Sie...

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

Ziele

Am Ende des Softwareprojekts können Sie...

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

Teamarbeit

“gemeinsame Instrumente entwickeln und einsetzen, die sicherstellen, dass Sie den Plan einhalten, und dass Sie Fragen und Probleme rechtzeitig klären.”

- regelmäßige Treffen - in der Gruppe und mit dem Betreuer
- gemeinsames “Logbuch” (Wiki oder gemeinsames Dokument)
- Kommunikation über Chat, Slack, ...
- gemeinsamer Terminkalender
- wöchentliche Statusreports aller Teilnehmer
- Codereviews
- Online oder Offline
- Einsetzen eines/r Projektmanagers/in

Teamarbeit

“gemeinsame Instrumente entwickeln und einsetzen, die sicherstellen, dass Sie den Plan einhalten, und dass Sie Fragen und Probleme rechtzeitig klären.”

- regelmäßige Treffen - in der Gruppe und mit dem Betreuer
- gemeinsames “Logbuch” (Wiki oder gemeinsames Dokument)
- Kommunikation über Chat, Slack, ...
- gemeinsamer Terminkalender
- wöchentliche Statusreports aller Teilnehmer
- Codereviews
- Online oder Offline
- Einsetzen eines/r Projektmanagers/in

Semesterplan

Datum	Inhalt
17.10. bis 22.10. (So!) 23.10.	Organisation, Projektvorstellung (Plenum) Email mit Projektwünschen an frank rschuman Projekteinteilung
24.10. 30.10. ...	Kickoff I Kickoff II Statusmeetings
14.11. ...	Besprechung Forschungsplan (Einreichung: 12.11.) Statusmeetings
28.11. ...	Spezifikationsvorträge (Plenum) Statusmeetings
06.02.	Abschlussvorträge, Demos (Plenum)
01.03. (Fr)	Projektabgabe

Projektanmeldung

Bis Sonntag, **22.10., 23:59 Uhr**, eMail an
{frank|rschuman}@cl.uni-heidelberg.de mit

- 1 Betreff: "SWP Projektanmeldung"
- 2 Mind. zwei Wunschprojekten mit Priorität
- 3 Falls Sie im Vorfeld eine Gruppe bilden wollen, die Namen der anderen Gruppenmitglieder.
- 4 Programmierkenntnisse in Python: (1: sehr gut, 2: gut-mittel, 3: schwierig, 4: gar nicht)
- 5 Kenntnisse und Erfahrung in Statistik, maschinellem Lernen, neuronalen Netzen
- 6 Fachrichtung und Fachsemester

Gruppeneinteilung

- 3-5 Mitglieder je Gruppe
- Vorgabe (Vorschlag) je nach Projekt
- Gruppen können gern im Vorfeld gebildet werden
- Bei Interessenskonflikten versuchen wir zu schlichten
- Jeder Teilnehmer bekommt ein Projekt

Forschungsplan: Abgabe 12.11.

- Erste, übergreifende Projektbeschreibung: Ziel, Methode, Evaluierung, Daten, Tools
- Beschreibt die Problemstellung (**Ziel**) und den Lösungsansatz (**Methode**) in eigenen Worten!
- Wie sollen die Ergebnisse **evaluiert** werden?
- Welche **Tools** und **Daten** sollen verwendet werden?
- *Abgabe schriftlich, ca. 2 Seiten*
- Besprechung 14.11.

Spezifikationsvortrag: 28.11.

- **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 (mit Parallelisierung!)
- **Konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: 28.11.

- **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 (mit Parallelisierung!)
- **Konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: 28.11.

- **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 (mit Parallelisierung!)
- **Konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: 28.11.

- **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 (mit Parallelisierung!)
- **Konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 20 Minuten*

Spezifikationsvortrag: 28.11.

- **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 (mit Parallelisierung!)
- **Konkrete Planung** von Programmarchitektur, Datenstrukturen, Schnittstellen
- *Vortrag im Plenum, maximal 20 Minuten*

Statusmeetings

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

Abschlussvortrag: 06.02

- Kurze Darstellung von Ziel und Methode
- Konkrete Umsetzung der Methode
- 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: 01.03. (**TBD**) – **keine Ausnahmen möglich!**

Voraussetzung für einen Schein:

- Teilnahme an allen gemeinsamen Veranstaltungen (Plenum) sowie allen Statusmeetings (max. 2x entschuldigt mit Attest)
- Forschungsplan
- Spezifikationsvortrag
- Abschlussvortrag (+ Demo, sofern sinnvoll)
- Dokumentation und Archivierung des Projekts
 - Dokumentation des Quellcodes
 - README mit Installationshinweisen
 - Schriftlicher Ergebnisbericht (je nach Projekt, ca. 4 Seiten), ggf. zusätzliche Anleitungen/Userguide
 - Lizenzierung (falls der Code veröffentlicht werden soll)

Leistungsnachweis

Abgabe: 01.03. (**TBD**) – **keine Ausnahmen möglich!**

Voraussetzung für einen Schein:

- Teilnahme an allen gemeinsamen Veranstaltungen (Plenum) sowie allen Statusmeetings (max. 2x entschuldigt mit Attest)
- Forschungsplan
- Spezifikationsvortrag
- Abschlussvortrag (+ Demo, sofern sinnvoll)
- Dokumentation und Archivierung des Projekts
 - Dokumentation des Quellcodes
 - README mit Installationshinweisen
 - Schriftlicher Ergebnisbericht (je nach Projekt, ca. 4 Seiten), ggf. zusätzliche Anleitungen/Userguide
 - Lizenzierung (falls der Code veröffentlicht werden soll)

Benotung

Benotungskriterien:

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

In Sonderfällen können Einzelnoten vergeben werden!

Vorstellung der Themen