

1 Implementierungsplanung

Die Meilensteine lassen sich wie folgt zusammenfassen:

- Nach der ersten Woche soll ein GUI-Dummy fertig sein, der sich über eine JAR öffnen lässt. Ein GUI-Dummy ist eine GUI, die schon das fertige Layout jedoch noch keine Funktionalität besitzt. Außerdem sind intern alle Interfaces aller Pakete zu implementiert, sowie die kompletten Pakete HighLevel, DataTypes und StringResource. Die Grammatik für die formalen Eigenschaften wird erstellt.
- Am Ende der zweiten Woche soll die GUI-Funktionalität fertig sein. Gemeint ist damit, dass alle Buttons intern schon abgefangen werden. Außerdem läuft die Implementierung weiterer Muss- und Sollkriterien.
- In der dritten Woche werden die Funktionen zum Speicher und Laden implementiert sowie alle Muss- und Sollkriterien fertig gestellt. Das Produkt soll also nach dem Ende der dritten Woche schon einsatzfähig sein.
- In der vierten und letzten Woche werden eventuell die Kann- und Wunschkriterien umgesetzt. Außerdem werden die Dokumente zur Vorstellung der Phase, welche die Präsentation sowie das Änderungsdocument zu vermeintlichen Änderungen im Entwurf und oder Pflichtenheft darstellen, erstellt. Falls die eingeplante Zeit für Muss- und Kannkriterien nicht reichen sollte, dient diese Zeit auch als Puffer.

Die Packages (hier fett) sind mit allen ihren Meilensteinen (falls nicht anders angegeben) jeweils einem der Phasenverantwortlichen zugeordnet.

Die Meilensteinen sind Personen zugeordnet.

Die Zahl in der Klammer jedes Meilensteins erklärt, in der wievielten Woche der Meilenstein implementiert werden soll.

Die Klassen, die die Funktionalität Speichern und Laden ermöglichen sind im Paket SaverLoader. In der Aufteilung sind mit SSpeichern und Laden in den Paketen, die Teile der Software in SaverLoader gemeint. Sie sind unter anderen Paketen aufgelistet, da sie für das Speichern und Laden der Klassen innerhalb der Pakete zuständig sind.

Mit Restliche Implementierung im Paket Toolbox ist die Implementierung aller Klassen innerhalb dieses Paketes gemeint, die sonst nicht explizit genannt sind.

Dieser Zeitplan baut auf das Entwurfsdokument und das Klassendiagramm auf.

Der Begriff Basisimplementierung umfasst alle Klassen innerhalb eines Paketes, die für die Muss- und Sollkriterien implementiert werden müssen.

Was	Wer
CodeArea	Schnell
Interfaces (1)	Klein
Basisimplementierung (1-2)	Schnell und Klein
Kannkriterien (4)	Schnell und Klein
CElectionDescriptionEditor	Schnell
GUI-Dummy (1)	Klein
GUI-Funktionalität (2)	Klein
Interfaces (1)	Klein
AntlrHandler (2-3)	Klein
Fehlererkennung (2-3)	Klein
Speichern und Laden (3)	Klein
Sollkriterien (3)	Klein
Kannkriterien (4)	Klein
BooleanExpEditor	Schnell
Interfaces (1)	Schnell
GUI-Dummy (1)	Schnell
Fehlererkennungskommunikation (2)	Schnell
GUI-Funktionalität (2)	Schnell
Speichern und Laden (3)	Schnell
Sollkriterien (3)	Schnell
AntlrHandler (2-3)	Schnell
Kannkriterien (4)	Schnell
PropertyList	Hanselmann
Interfaces (1)	Hecht
GUI-Dummy (1)	Hecht
GUI-Funktionalität (2)	Hecht
Ergebnisdarstellung (2-3)	Hecht
Speichern und Laden (3)	Hecht
SaverLoader	Hanselmann
Interfaces (1)	Hecht
Options	Hanselmann
Interfaces (1)	Stapelbroek
Basisimplementierung (1-2)	Stapelbroek
Kannkriterien (4)	Wohnig

Was	Wer
PropertyChecker	Hanselmann
Interfaces (1)	Hanselmann
CBMC-Ansteuerung (2-3)	Stapelbroek
Codegenerierung (2-3)	Hanselmann und Stapelbroek
Eventuelle Multithreadingverbesserung (4)	Stapelbroek
Parametereditor	Hanselmann
Interfaces (1)	Wohnig
GUI-Dummy (1)	Wohnig
GUI-Funktionalität (2)	Wohnig
Speichern und Laden (3)	Wohnig
DataTypes	Hanselmann
Implementierung (1)	Stapelbroek und Hanselmann
Toolbox	Schnell
Restliche Implementierung (1)	Stapelbroek
BooleanExpressionToASTConverter (2-3) (Erstellung des AST)	Schnell und Klein
UserActionKlasse (1-2)	Wohnig
StringResource	Hanselmann
Interface (1)	Hanselmann
Anlegen und verwalten der Textdatei Deutsch (1)	Hanselmann
Anlegen Textdatei Englisch (4)	Hanselmann
HighLevel	Hanselmann
Komplette Implementierung (1)	Wohnig
Sonstiges	Hanselmann
Erstellung JAR (1)	Wohnig
Präsentationsplanung (4)	Schnell und Hanselmann
Änderungsdokument erstellen (1-4)	Wohnig
BooleanExp Antlr-Grammatik erstellen (1)	Schnell und Klein (Schnell betreut diesen Meilenstein)

Hier die Formatierung nach Personen sortiert:

Schnell:

- BooleanExpEditor Interfaces (1)
- BooleanExpEditor GUI-Dummy (1)
- CodeArea Basisimplementierung [mit Klein] (1-2)
- BooleanExpEditor GUI-Funktionalität (2)
- BooleanExpEditor Fehlererkennungskommunikation (2)
- Toolbox BooleanExpressionToASTConverter [mit Klein] (2-3)
- BooleanExpEditor Speichern und Laden (3)
- BooleanExpEditor Sollkriterien (3)
- BooleanExpEditor Kannkriterien (4)
- CodeArea Kannkriterien [mit Klein] (4)
- Sonstiges Präsentationsplanung [mit Hanselmann] (4)

Klein:

- CodeArea Interfaces (1)
- CElectionDescriptionEditor GUI-Dummy (1)
- CElectionDescriptionEditor Interfaces (1)
- CodeArea Basisimplementierung [mit Schnell] (1-2)
- CElectionDescriptionEditor GUI Funktionalität (2)
- CElectionDescriptionEditor AntlrHandler (2-3)
- CElectionDescriptionEditor Fehlererkennung (2)
- Toolbox BooleanExpressionToASTConverter [mit Schnell] (2-3)
- CElectionDescriptionEditorSollkriterien (3)
- CElectionDescriptionEditor Speichern und Laden (3)
- CodeArea Kannkriterien [mit Schnell] (4)

- CElectionDescriptionEditor Kannkriterien (4)

Hecht:

- SaverLoader Interface (1)
- PropertyList Interfaces (1)
- PropertyList GUI-Dummy (1)
- PropertyList GUI-Funktionalität (2)
- PropertyList Ergebnisdarstellung (2-3)
- PropertyList Speichern und Laden (3)
- StringResource Anlegen Textdatei Englisch (4)

Stapelbroek:

- Datatypes Implementierung [mit Hanselmann] (1)
- Options Interface (1)
- ToolBox Restliche Implementierung (1)
- Options Basisimplementierung (1-2)
- PropertyChecker CBMC-Ansteuerung (2-3)
- PropertyChecker Codegenerierung [mit Hanselmann] (2-3)
- PropertyChecker Eventuelle Multithreadverbesserung (4)

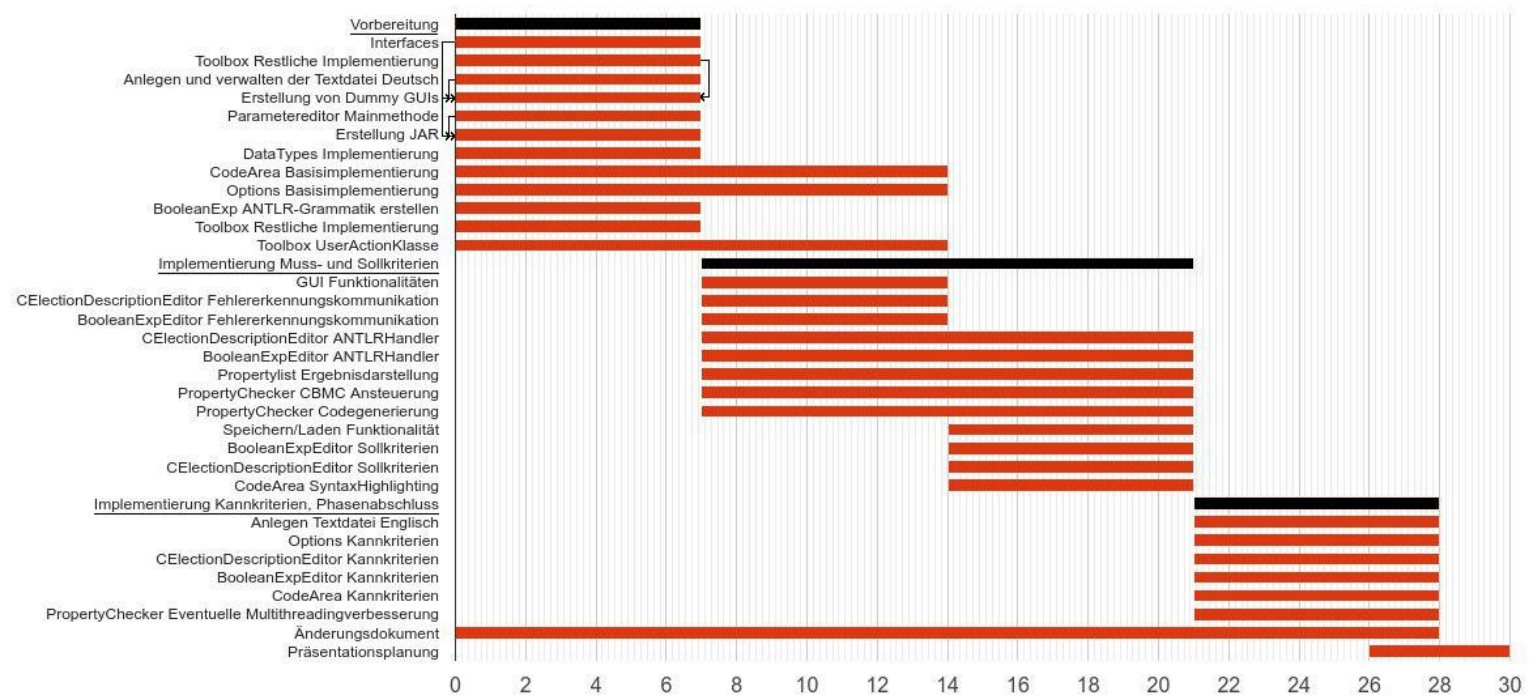
Hanselmann:

- Datatypes Implementierung [mit Stapelbroek] (1)
- PropertyChecker Interfaces (1)
- StringResource Interface (1)
- StringResource Anlegen und verwalten der Textdatei Deutsch (1)
- PropertyChecker Codegenerierung [mit Stapelbroek] (2-3)
- Sonstiges Präsentationsplanung [mit Schnell] (4)

Wohnig:

- Parametereditor Interfaces (1)
- Parametereditor GUI-Dummy (1)
- HighLevel Komplette Implementierung (1)
- Sonstiges Erstellung JAR (1)
- Toolbox UserActionKlasse (1-2)
- Parametereditor GUI-Funktionalität (2)
- Parametereditor Speichern und Laden (3)
- Sonstiges Änderungsdocument erstellen (1-4)
- Options Kannkriterien (4)

In dem folgenden Diagramm sieht man die Zeitplanung:
Die Pfeile beschreiben die Abhängigkeiten.



Projektzeit Tag 0 bis Tag 30