

Inhaltsverzeichnis

1. Projektorganisation
 - 1.1. Projektplan
 - 1.2. Durchführung
 2. Protokoll: Kundengespräch 2020-05-06
 - 2.1. Anwesend
 - 2.2. Ziel des Meetings: Erklärung und Vorstellung des Prototypen / Feedback zu bisherigem Stand
 - 2.3. Das wurde dem Kunden erklärt:
 - 2.4. Fragen an den Kunden
 - 2.5. Fazit
 3. Überarbeitetes Use Case Model
 4. Use-Case Model BauphysikSE1
 - 4.1. Identifizierte Use Cases
 - 4.2. Use Case Diagramm
 - 4.3. Ausgearbeitete Use Cases
 - 4.4. Use Case: Programmstart
 - 4.5. Use Case: Reihenberechnung durchführen
 - 4.6. Use Case: Reihenberechnung durchführen
 - 4.7. Use Case: Plausibilität prüfen
 - 4.8. Use Case: Berechnungsdaten laden
 - 4.9. Use Case: Berechnungsdaten speichern
 - 4.10. Use Case: Daten drucken
 5. Anwenderdokumentation
 - 5.1. Benutzerhandbuch
 - 5.2. Flyer
 6. Entwicklerdokumentation
- Entwurfsdokumentation
1. Systemarchitektur
 2. Verwendete Frameworks/Module
 3. Struktur der gespeicherten Dateien
 4. Sequenzdiagramm
 5. Schnittstellen
 6. Testdokumentation
 7. Testkonzept Projekt Bauphysik
 - 7.1. Testobjekte
 - 7.2. Testmethoden
 - 7.3. Testplanung
 8. Test-Log
 - 8.1. Testscripte
 - 8.2. Testergebnisse
 9. Betriebsdokumentation

1. Projektorganisation

1.1. Projektplan

1.1.1. Einführung

Wir als Team haben, im Rahmen der Lehrveranstaltung SE 2, ein Problem mithilfe eines Softwareproduktes lösen wollen. Für dieses Vorhaben mussten entsprechende Methoden, Modelle und Vorgehensweisen gefunden und richtig eingesetzt werden.

Das Projekt wurde über 2 Semester, Wintersemester 19/20 und Sommersemester 20, bearbeitet.
Im Folgenden Dokument werden Planung, Durchführung und Ergebnisse nachvollziehbar präsentiert werden.

1.1.2. Aufgabenstellung und Ausgangssituation

Aufgabenstellung

Im Folgenden wird die Aufgabenstellung vom ersten Semester zitiert:

Hintergrund

Der Wärmedurchgangskoeffizient eines Bauteils U ist eine wichtige bauphysikalische Größe, welche sowohl für die Ermittlung des Energiebedarfs eines Gebäudes als auch für den Nachweis des Mindestwärmeschutzes und der Tauwasserfreiheit berechnet werden muss. Studierende des Bauingenieurwesens führen derartige Berechnungen im Rahmen der Lehrveranstaltung Bauphysik aus. Ein häufiger Anwendungsfall sind Wände aus mehreren parallelen Schichten (z. B. Innenputz/ Mauerwerk/Dämmmaterial/Außenputz). Um möglichst eine große Vielfalt an Aufgaben für die Studierenden kreieren und die Lösungen überprüfen zu können, wird eine Software benötigt. Diese Software ist zunächst nur für die Lehrenden gedacht, könnte aber zukünftig z. B. auch über OPAL für die Studierenden bereitgestellt werden.

Zielstellung

Entwickeln Sie für den beschriebenen Einsatz ein SW-System, mit dem 1. die erforderlichen Daten zur Berechnung von U eingegeben werden können: die Wärmeübergangswiderstände, die Anzahl der parallelen Schichten n , die Dicken und Wärmeleitfähigkeiten sowie alternativ die Wärmedurchlasswiderstände iR der Materialien (bei komplexen Konstruktionen und bei ruhenden Luftschichten), 2. die Berechnungen ausgeführt und 3. die Ergebnisse entsprechend der Gleichungen ausgegeben werden können.

Ausgangssituation

Im ersten Semester wurde das Projekt initialisiert, eine Anforderungsanalyse erarbeitet, auf eine technische Lösung geeinigt und das Projekt geplant.

Die Anforderungsanalyse verlief sehr gut und die daraus gewonnenen Dokumente waren bereits in einem verfeinerten Zustand, wodurch hat das gesamte Team einen guten Eindruck des zu lösenden Problems hatte.

Am Ende des ersten Semesters verließ ein Teammitglied, der Analyst, das Team. Aufgrund der Erfolgreichen Anfangs- und Ausarbeitungsphase haben wir uns entschieden die Rollenverteilung nicht zu ändern.

Ziel des zweiten Semesters war die Konstruktions- und Übergangsphase erfolgreich abzuschließen. Mithilfe der in der Vorlesungen kennengelernten Methoden strebten wir eine koordinierte und kooperative Konstruktion einer Softwarelösung an, die unsere Tests besteht und gleichzeitig möglichst alle Anforderungen der Kundin erfüllt.

Dabei sollte, wie bereits schon in Software Engineering 1, mithilfe der Vorgehensweise OpenUP ein strukturierter Arbeitsablauf gewährleistet werden.

Aufgrund der COVID-19 Pandemie wurden jegliche Präsenztermine wie Meetings oder Kundengespräche vorerst komplett abgesagt und durch digitale Alternativen ersetzt.

1.1.3. Rahmenbedingungen

Das Team Die Tätigkeiten sind nach Rollen und Themengebiet aufgeteilt. Eine Person in Ihrer Rolle trägt Verantwortung in diesem Themenbereich und trifft hier nach eigenem Ermessen Entscheidungen. Alle anstehenden und abgeschlossenen Tätigkeiten werden flexibel in GitHub Issues festgehalten und soweit einer Iteration zugeordnet, bearbeitet.

Die Rollenverteilung sieht wie folgt aus:

Name, Vorname	Primäre Rolle	Sekundäre Rolle
Klassowski, Denis	Project Manager	Analyst
Däbler, Michael	Analyst	Technical Writer

Name, Vorname Grambole, Lukas	Primäre Rolle Analyst	Sekundäre Rolle ---
Griß, Christian	Architect	Developer
Baburkin, Yewgenji	Developer	Project Manager
Ullmann, Max	Developer	Tester
Lehmann, Christian	Tester	Architect
Fritzsche, Felix	Deployment Eng	Developer

Bearbeitungszeitraum

Der Bearbeitungszeitraum des Problems beträgt in etwa 7 Monate. Aufgrund der agilen Entwicklung ist die Übergabe der Softwarelösung mit dem 31.07.2020 festgeschrieben.

Ressourcen

Alle Teammitglieder des zweiten Semesters sind Studenten in Informatischen Studiengängen und Teilnehmer im Modul Software Engineering 2. Damit ist ein Verständnis von Informatischen Systemen gegeben und die Vorgehensweise OpenUP bekannt. Desweiteren besitzt jedes Mitglied einen Rechner mit Mikrofon und kann somit an digitalen Meetings teilnehmen.

Für Modellierung kann, im Rahmen der Lehrveranstaltung, das Programm Paradigm™ genutzt werden.

Als Versionsverwaltung der Software wird das Online Tool GitHub verwendet.

Kommunikation

Die Kommunikation innerhalb des Teams verläuft über 3 Kanäle.

1. **GitHub** Issues
2. **Discord** (Onlinedienst für Instant Messaging, Chat und Sprachkonferenz)
3. **WhatsApp** (Instant Messenger)

Die Kanäle haben dabei eine klare Zuordnung von Aufgabenfeldern.

GitHub Issues dient als Work-Item-List und sofortiges Feedback zu Pushes oder Lösungen.

Discord wird für Meetings jeglicher Art genutzt. (Iteration Meeting, Kleingruppen Meetings Kundengespräche, Pair Programming).

WhatsApp Vereinbarung von Meetings und Instant Feedback bei Problemen

Für die Kommunikation mit dem Kunden haben wir uns darauf geeinigt das Meeting in Discord abzuhalten. Dafür wurde das Programm per Telefonie im Vorfeld eingerichtet und die Nutzung erläutert.

Dokumentation

Die Dokumentation wurde in größtenteils in GitHub geführt, teilweise jedoch auf Google Docs ausgelagert. Dadurch konnten einige Dokumente einfacher gepflegt und auf dem aktuellen Stand gehalten werden.

Bei der Dokumentation haben wir uns an das Vorgehensmodell OpenUP gehalten.

1.1.4. Eingesetzte Techniken

Pair Programming

Code Review

1.2. Durchführung

Im Folgenden sind, in chronologischer Reihenfolge, die Ziele, Aktivitäten, Ergebnisse und Entscheidungen Stichpunktartig aufgelistet. 2.x Steht für Iteration x im zweiten Semester.

Iteration 2.1

Start: 30.03.2020

Ende: 12.04.2020

Ziele:

- Projekt wieder anschieben
- Auf die Corona Situation anpassen und besprochene
- Alternativen nutzen
- Implementierung von GitHub Issues als Work Item List
- Vorbereiten auf Kundengespräch
- Haupt Prototyp wird erstellt und iterativ angepasst

Aktivitäten:

- Iterations Meeting
- Einrichtung eines Discord Servers
- Struktur und Richtlinie für GitHub Probleme festgelegt

Ergebnisse:

- Eigenständige Organisation von Rollen Meetings funktioniert gut
- Sammlung einiger ungeklärter Fragen an Kunden
- Team ist trotz des verzögerten starts zuversichtlich
- Teammoral ist sehr positiv und motiviert

Entscheidungen:

- Die wegfallende Rolle des Analysten wird nicht neu besetzt
 - Ein Analyst reicht für das Projekt aus
- Developer sind unterteilt in frontend und backend
 - Devs müssen nicht immer wieder einarbeiten in Quellcode
- Meetings werden künftig in Discord abgehalten
 - Ist ein umfangreiches und zugleich kostenloses Tool
- Es wird, bis auf weiteres, keine Präsenz Meetings geben
 - Gesetzgebung lässt es nicht zu und das Infektionsrisiko soll klein gehalten werden

Probleme:

- Meetings haben spät begonnen

- Die Idee eine Iteration kürzer zu machen wurde verworfen. Wir glauben immer noch gut in der Zeit zu liegen
- Dokumentation ist umständlich
 - Die Dokumentation wird etwas aufgeteilt und teils in Google Drive gespeichert

Iteration 2.2

Start: 13.04.2020

Ende: 27.04.2020

Ziele:

- Prototyp weiterentwickeln damit eine Präsentation an den Kunden möglich ist
- Meetings strukturierter abhalten
- Kundenmeeting planen

Aktivitäten:

- Iterations Meeting
- Haupt Prototyp wurde erstellt und entspricht bereits grob dem Wireframe
- Noch offene Fragen an den Kunden wurden gesammelt und mit Anforderungen gegengeprüft
- Meeting Protokoll aus 2.1 wurde überarbeitet
- Dev-Meeting für die exakte Zuordnung von Aufgaben
- Kundenmeeting geplant und vorbereitet

Entscheidungen:

- Kürzeres Feedback Intervall durch Devs in dieser Iteration
 - Damit Fertigstellung bis Kundenmeeting sichergestellt ist
- Alle Mitglieder sind für GitHub Issues verantwortlich und müssen darauf achten, dass die festgelegten Richtlinien von allen eingehalten werden
 - Es entsteht sonst Undurchsichtigkeit und Aufwand

Probleme:

- Haupt Prototyp ist noch nicht ausreichend für ein Kundenmeeting
 - Rückmeldung der Devs nach einer Woche, sollte die Velocity nicht ausreichen → Zuordnung von mehr Ressourcen
- Aufgabenzuordnung schwierig da in dieser Iteration nicht viele Aufgaben existierten
 - Keine Künstlich geschaffenen Probleme generieren um Aufgabengestaltung fairer zu machen. Wurde im Meeting besprochen, alle haben dem Umgang zugestimmt
- Backend Entwicklung kann nicht nicht erfolgen da Objektmanagement innerhalb des Programms noch unklar ist. Es bedarf noch mehr Einarbeitung in Qt5
 - Backend Devs sollen sich mit Qt5 beschäftigen um ein Verständnis für das Framework zu entwickeln

Iteration 2.3

Start: 27.04.2020

Ende: 10.05.2020

Ziele:

- Backend entwicklung planen
- Datenstruktur fertigstellen
- U-Berechnung fertigstellen
- Feedback der Kundin mit höchster Priorität verarbeiten
- Prototyp zum Druckauftrag erstellen
- Testskripte für U-Berechnung schreiben
- Code Review vereinbaren (Frontend)
- Risk List überarbeiten
- Erstes Dokument zu Programmbedienung erstellen
- Prototyp in logischer Programmierung erstellen (dynamisches Berechnen)
- Branching einführen
- Analyse der neuen Informationen vom Kunden (mehr Beispielaufgaben) auf mehrwert

Aktivitäten:+

- Iterations Meeting
- Discord beim Kunden eingerichtet
- Kundenfeedback in Anforderungen übernommen
- Kundenmeeting abgehalten
- Datenstruktur wurde bearbeitet
- Frontend angepasst
- Farbdesign entworfen

Entscheidungen:

- Dev meeting am Ende der Woche um fertigstellung von Backend zu gewährleisten
 - Backend jetzt von hoher Priorität weil weitere Tests auf Berechnungen unvollständig sind
- (Vorläufige Entscheidung) Logische Programmierung kommt nicht zum Einsatz weil der Aufwand sehr hoch ist und die Anforderung optional

Probleme:

- Frontend hat keinen guten Eindruck hinterlassen weil es nicht ansprechend genug war
 - Beim Endkunden Meeting (besonders bei der Vorstellung von Produkt Oberfläche) hätten wir Farben und Design einfließen lassen sollen und nicht nur die Anordnung/Existenz von Button bzw. Interaktionsfelder

Iteration 2.4

Start: 11.05.2020

Ende: 25.05.2020

Ziele:

- Berechnungen fertigstellen
- Berechnungen testen
- Import / Export hinzufügen
- Import / Export testen
- Druckauftrag bearbeiten
- Die wichtigsten Anforderungen erfüllen um das Programm demnächst an Kunden zu übergeben (Betaphase)
- Backend testen

Aktivitäten:

- Iterations Meeting
- Frontend wurde an neues Design angepasst
- Alle bisherigen Frontend Issues bearbeitet
- Deployment Plan aktualisiert
- Tests wurden vorbereitet
- Design mit Icons vervollständigt
- Code Review abgehalten

Entscheidungen:

- Das Backend wird nun von Ersatzrolle Dev und Tester erstellt
 - Backend Dev sagte er habe wenig Zeit, würde es aber bis Iterationsende fertigstellen. Dann meldete er sich nicht → Neuzuteilung
- Logische Programmierung wird endgültig verworfen
 - Logische Programmierung ist auch am zweiten Versuch als Sinnvolle Lösung gescheitert

Probleme:

- Backend Dev hat wenig Zeit
 - Seine Aufgaben werden nun auf die anderen Mitglieder verteilt
- Einige Ziele der Letzten Iteration wurden nicht erreicht
 - Verschiebung in der Priorität

Iteration 2.5

Start: 26.05.2020

Ende: 16.06.2020

Ziele:

- Druckauftrag fertigstellen
- Kundenfeedback aus Betaphase umsetzen

- Neudesign der Startoberfläche
- Persistente Speicherung fertigstellen
- Dev Meeting planen
- Flyer und benutzerhandbuch aktualisieren
- Zweite Betaversion an Kunden übergeben

Aktivitäten:

- Iterations Meeting
- Erstes Deployment von Betaversion des Programms
- Dev Meeting abgehalten
- Kundenfeedback aus Beta in Anforderungen aufnehmen

Entscheidungen:

- Neue Betaversion wird an Kunden übergeben
 - Einige, sehr wichtige, Funktionen fehlen noch in der ersten Version

Probleme:

- Backend Dev weiterhin verhindert
 - Erstmal keine unmittelbare wichtige Aufgaben zugeordnet
- Nur wenige Teammitglieder sind in der Lage Programmcode zu schreiben
 - Verschieben mehrerer Tätigkeiten, sodass Devs zu Verfügung stehen
 - Code Review hat dieses Problem nicht verhindern können

Iteration 2.6

Start: 16.06.2020

Ende: 29.06.2020

Ziele: * Neudesign der Startoberfläche diese Iteration lösen * Optionale Anforderungen und Quality of Life Features hinzufügen * Verbesserungsvorschläge des Teams evaluieren * Auf Kundenfeedback reagieren und Dokumente anpassen * Kleine Optische veränderungen um dem Design zu entsprechen * Bugfixing(Druckauftrag, Zahlendarstellung, Graph Skalierung)

Aktivitäten:

- Iterations Meeting

Entscheidungen:

- Einige Optionale Anforderungen die wenig Gewichtung für den Kunden haben werden verworfen
 - Kunde hat auch bei einer erneuten Rücksprache nur wenig Interesse an diesen Änderungen gezeigt

Probleme:

- Kommunikation zu Backend Dev ist nur schwierig aufzubauen

- Mittlerweile ist die Dev Rolle bereits vergeben. Zuordnung von kleineren Aufgaben (CSS Anpassungen)

Iteration 2.7

Start: 29.06.2020

Ende: 12.07.2020

Ziel:

- Fehlende CSS Anpassungen neu verteilen
- letzte Testfehler bearbeiten
- Dokumentation für Belegabgabe vorbereiten
- Finale Übergabe des Programms

Aktivitäten:

- Iterations Meeting

Entscheidungen:

- Alle anderen Bearbeitungen erfolgen nach der Prüfung
 - Software so gut wie fertig und alle Mitglieder sind jetzt in Prüfungsstress

Probleme:

- Backend Dev meldet sich nicht mehr und nimmt nicht an Meetings teil
 - Projekt bereits so gut wie abgeschlossen. Keine Gefahr mehr für Teamerfolg

1.2.1. Gesprächsprotokoll 1

2. Protokoll: Kundengespräch 2020-05-06

2.1. Anwesend

1. Tester
2. Manager
3. Analyst
4. Dev (Yewgenij)
5. Deployment
6. Kunde

2.2. Ziel des Meetings: Erklärung und Vorstellung des Prototypen / Feedback zu bisherigem Stand

2.3. Das wurde dem Kunden erklärt:

1. Tab System
2. Wo sind die Eingabemöglichkeiten

3. Überblick zum Schichtsystem (dynamisches einfügen und löschen)
4. Schichttemperatursystem
5. Funktion des Moduswechsels
6. Erhalt der Werte bei Moduswechsel
7. Verlauf des Projektes und die Zielstellung einen Prototypen zu bauen der an den Kunden übergeben wird

2.3.1. Bemerkung des Kunden:

- Bezeichnung "Breite" ändern zu "Dicke"
- Zweiteilung der Eingabemaske und Schichtsystem deutlich machen
- U Wert Berechnung bei Temp Berechnung nicht notwendig
- Werte sollen bei Wechsel zu Temp Modus erhalten bleiben
- Wichtige Werte bei Temp Berechnung sollen angezeigt werden: R_{si} , R_{se} , U - Wert, R_t , R Gesamt
- im Formlayout $R_{\{Nummer\}}$ statt R_i schreiben (Reihenfolge)
- Programm soll etwas farblicher und ansprechender sein

2.4. Fragen an den Kunden

1. Soll eine Fehleingabe durch ein Popup signalisiert werden?
 - Nein, Fehleingabe unterdrücken reicht aus.
2. Wie sollen die einzelnen Tabs gespeichert werden?
 - Ein Tab (eine Berechnung) wird unter einem Namen als Datei gespeichert
3. Welches Layout soll der Druckauftrag einer Berechnung haben
 - Soll ähnlich aussehen wie das Layout im Programm, Mehrere Tabellen mit Materialdaten/Eingangsdaten/Allgemeine Daten/ Grenzwerttabelle
 - Kunde wird noch eine Beispieltabelle als Vorlage anlegen
4. Soll das Programm rekursive Berechnung (Fehlender Parameter wird errechnet) bieten?
 - Wäre gut aber hat keine Priorität

2.5. Fazit

- Kunde ist Grundsätzlich zufrieden, Anmerkungen waren überwiegend visuelle Wünsche
- Es ist davon auszugehen dass unsere Vorstellung des Prototypen das Problem des Kunden lösen kann

2.5.1. Gesprächsprotokoll 2

Iterations Meeting am 10.05.2020

Beginn 16:00

Fehlende Teilnehmer: Max Ullmann

Was haben wir erreicht?

- Alle Issues wurden behandelt, Datenstruktur und Funktionen sind hochgeladen
- Testskripte / testlogging sind geschrieben
- Farbliche Idee entworfen

- Benutzerhandbuch angefangen

Welche Probleme hab es?

- Problem in Datenstruktur, Objektgestaltung von Export unklar
- Windows 10 findet executable file unsicher
- Backend Dev weiterhin verhindert, zuarbeit wird jedoch benötigt
- Meeting musste verschoben werden [10.05 → 19.05]

Was sind unsere Ziele für die nächste Iteration

- Entscheidung ob logische Programmierung verwendet wird
- Coach Meeting planen?
- Berechnung fertigstellen und testen
- Erklärung der GUI im Handbuch

2.5.2. Gesprächsprotokoll 3

Iterations Meeting am 30.03.2020

Beginn: 16:00

Fehlende Teilnehmer: -

Was waren die Probleme in SE1?

- chaotische Abarbeitung der Aufgaben. Gesamtübersicht hat gefehlt
- Lösung durch GitHub Issues

Ideen für weitere Verbesserungen?

- Keine weiteren Vorschläge
- Gemeinsame Besprechung und Entscheidung das Discord sinnvolle, zentrale Kommunikationseinheit ist
- Gemeinsame Besprechung und erörterung wie GitHub Issues aussehen sollen

Wie soll die Entwicklung ablaufen?

- Architekt überblickt Gesamtsystem und verteilt requirements als Issue an Devs
- Dev entwickelt Prototyp, welcher durch Tester geprüft wird
 - Ist der Test ok wird der Prototyp in Gesamtsystem integriert

Sonstige

- Team war mit Iterationsdauer zufrieden.
- Zwei Wochen Rhythmus bleibt auch für SE2 bestehen
- Jedes Mitglied kann trotz der Pandemie am Projekt arbeiten
- Teammoral ist sehr Positiv

2.5.3. Gesprächsprotokoll 4

Iterations Meeting am 12.07.20

Beginn: 16:00

Fehlende Teilnehmer: Max Ullmann, Christian Grieb, Christian Lehmann

Was haben wir erreicht?

- Fast alle Tests sind erfolgreich gewesen
- Anforderungsanalyse angepasst

Welche Probleme gab es?

- Das HTML Dokument und der PNG Export des pyqtgraph sind noch nicht flexibel einsetzbar. Funktioniert nur bedingt
- Max hat sich immer noch nicht gemeldet und auch die CSS wird langsam benötigt

Was wollen wir nächste Iteration erreichen?

- Dokumentation für Beleg machen
- Produkt Übergeben
- Projekt abschließen
- Finale Änderungen in den Dokumenten

3. Überarbeitetes Use Case Model

4. Use-Case Model BauphysikSE1

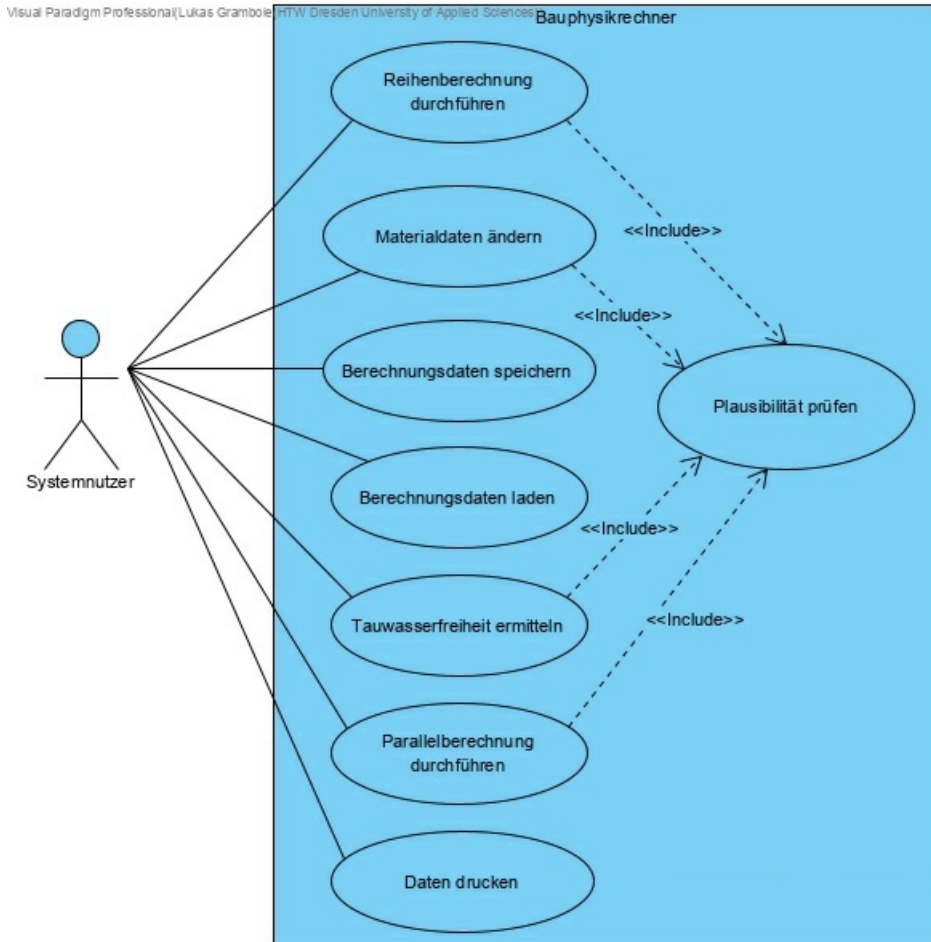
4.1. Identifizierte Use Cases

Hinweis: Die Use Cases wurden nach ihrer Priorität sortiert.

Kurzbezeichnung	Name	Akteur	Beschreibung
UC1	Programmstart	Systemnutzer	Start des Programms
UC2	Reihenberechnung durchführen	Systemnutzer	Berechnung und Ausgabe von j , R_{ges} , R_i , R_T , U , $\Delta\vartheta_k$ und ϑ_k für in Reihe angeordnete Wandschichten
UC3	Temperaturverlaufs berechnung durchführen	Systemnutzer	zusätzlich zu den in UC1 berechneten Werten Berechnung von Grenztemperaturen und Temperaturverlauf
UC4	Plausibilität prüfen	Systemnutzer	Überprüfung der Gültigkeit von Eingabedaten
UC5	Berechnungsdaten speichern	Systemnutzer	Speicherung von Eingabe- und Ergebnisdaten

UC6 Kurzbezeichnung	Berechnungsdaten laden Name	Systemnutzer Akteur	Laden von gespeicherten Beschreibung Eingabe- und Ergebnisdaten
UC7	Daten drucken	Systemnutzer	Druck berechneter Informationen

4.2. Use Case Diagramm



4.3. Ausgearbeitete Use Cases

4.4. Use Case: Programmstart

4.4.1. Kurzbeschreibung

Der Use Case beschreibt den Programmstart für das Softwaresystem.

4.4.2. Kurzbeschreibung der Akteure

Systemnutzer Möchte das System nutzen um Berechnungen ausführen zu können.

4.4.3. Vorbedingungen

Der Systemnutzer ist im Besitz des Softwaresystems und führt das System unter Windows 10 auf einem handelsüblichen Rechner aus der mit dem Internet verbunden ist. Die Berechnungssoftware ist auf einem internen oder extern angeschlossenen Medium verfügbar.

4.4.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Systemnutzer das Softwaresystem aufruft
2. Die Berechnungssoftware startet fehlerfrei

3. *WHILE* Das System lädt Hintergrunddaten a.. *INCLUDE* Plausibilität prüfen
4. Der Systemnutzer wählt erwünschten Vorgang aus.
5. Der Use Case ist abgeschlossen.

4.4.5. Alternative Abläufe

Der Systemnutzer startet das Programm auf einem Computer, der nicht mit dem Internet verbunden ist. In dem Fall soll ebenfalls der Standardablauf gelten.

4.4.6. Wesentliche Szenarios

- **SC1:** Der Systemnutzer ist mit seinem PC online und startet das Programm. Anschließend wählt er die gewünschte Berechnung aus. Die Software lädt im Hintergrund die benötigten Daten um die Berechnung zu ermöglichen. Der Programmstart wurde erfolgreich abgeschlossen.
- **SC2:** Der Systemnutzer ist mit seinem PC online und startet das Programm. Anschließend wählt er die gewünschte Berechnung aus. Die Software lädt im Hintergrund die benötigten Daten um die Berechnung zu ermöglichen. Der Programmstart wurde erfolgreich abgeschlossen.
- **SC3:** Der Systemnutzer startet das Programm, ohne dass die Software vorher installiert wurde. Die Berechnungssoftware startet fehlerfrei. Der Programmstart wurde erfolgreich abgeschlossen.

4.4.7. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case muss folgende Nachbedingungen erfüllt sein:

- Softwaresystem ist fehlerfrei gestartet und kann zur Berechnung verwendet werden.

4.5. Use Case: Reihenberechnung durchführen

4.5.1. Kurzbeschreibung

Der Use Case beschreibt einen Berechnungsvorgang für die Wärmewiderstandsdaten von in Reihe geschalteten Wandschichten.

4.5.2. Kurzbeschreibung der Akteure

Systemnutzer

will Wärmewiderstandsberechnung durchführen.

Vorbedingungen

Der Systemnutzer hat das Softwaresystem gestartet.

Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Systemnutzer die Funktion der Reihenberechnung ausgewählt hat.
2. *WHILE* Das System bietet eine Eingabemöglichkeit für die Materialdaten.
 - a. *WHILE* Der Systemnutzer gibt Materialdaten ein.
 - b. Das Softwaresystem berechnet direkt die Wärmewiderstandsdaten.
 - c. Das Softwaresystem gibt die berechneten Wärmewiderstandsdaten aus.
 - i. *INCLUDE* Plausibilität prüfen
3. *IF* Systemnutzer will einen Temperaturverlauf ermitteln
4. Systemnutzer wechselt in den Modus Temperaturverlaufs Berechnung

- a. *WHILE* Der Systemnutzer gibt Temperaturdaten ein.
 - b. Das Softwaresystem ermittelt den Temperaturverlauf.
 - c. Das Softwaresystem gibt den Temperaturverlauf aus.
 - i. *INCLUDE* Plausibilität prüfen
5. Der Use Case ist abgeschlossen.

Alternative Abläufe

Alternativer Ablauf #2.2

Wenn der Systemnutzer im Schritt 2 des Standardablaufs weitere Materialschichten hinzufügen möchte, dann

- i. Der Systemnutzer fügt eine Schicht hinzu.
- ii. Das System schafft Eingabemöglichkeiten für die Daten einer zusätzlichen Materialschicht.
- iii. Der Use Case wird im Schritt 2 des Standardablaufs fortgesetzt.

Alternativer Ablauf #2.3

Wenn der Systemnutzer im Schritt 2 des Standardablaufs eine Materialschicht entfernen möchte, dann

- i. Der Systemnutzer entfernt eine Materialschicht.
- ii. Das System entfernt die Eingabemöglichkeiten für die Daten einer Materialschicht.
- iii. Der Use Case wird im Schritt 2 des Standardablaufs fortgesetzt.

Wesentliche Szenarios

- **SC1:** Der Systemnutzer wählt die Funktion der Wärmewiderstandsberechnung aus und gibt die Materialdaten von zwei Schichten ein. Während der Systemnutzer die Berechnungsdaten eingibt, berechnet das Softwaresystem die Wärmewiderstandsdaten und gibt diese aus. Die Berechnung wurde erfolgreich abgeschlossen.
- **SC2:** Der Systemnutzer wählt die Funktion der Wärmewiderstandsberechnung aus und gibt die Materialdaten von zwei Schichten ein. Während der Systemnutzer die Berechnungsdaten eingibt, berechnet das Softwaresystem die Wärmewiderstandsdaten und gibt diese aus. Anschließend wechselt der Systemnutzer in den Modus Temperaturverlaufsberechnung, gibt Temperaturdaten ein und lässt einen Temperaturverlauf ermitteln. Die Berechnung wurde erfolgreich abgeschlossen.
- **SC3:** Der Systemnutzer wählt die Funktion der Wärmewiderstandsberechnung aus und gibt direkt die Wärmedurchlasswiderstände von zwei Schichten ein. Während der Systemnutzer die Werte eingibt, berechnet das Softwaresystem die Wärmewiderstandsdaten und gibt diese aus. Die Berechnung wurde erfolgreich abgeschlossen.
- **SC4:** Der Systemnutzer wählt die Funktion der Wärmewiderstandsberechnung aus und gibt die Materialdaten von drei oder mehr Schichten ein. Während der Systemnutzer die Berechnungsdaten eingibt, berechnet das Softwaresystem die Wärmewiderstandsdaten und gibt diese aus. Die Berechnung wurde erfolgreich abgeschlossen.

4.5.3. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case muss folgende Nachbedingungen erfüllt sein:

- Die Berechnungsergebnisse werden angezeigt.

4.5.4. Wireframes

Der Use Case beschreibt einen Berechnungsvorgang für den Temperaturverlauf von in Reihe geschalteten Wandschichten.

4.6.2. Kurzbeschreibung der Akteure

Systemnutzer

will Temperaturverlaufsrechnung durchführen.

Vorbedingungen

Der Systemnutzer hat das Softwaresystem gestartet.

Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Systemnutzer die Funktion der Temperaturverlaufsrechnung ausgewählt hat.
2. *WHILE* Das System bietet eine Eingabemöglichkeit für die Materialdaten.
 - a. *WHILE* Der Systemnutzer gibt Materialdaten ein.
 - b. Das Softwaresystem berechnet direkt die Wärmewiderstandsdaten.
 - c. Das Softwaresystem gibt die berechneten Wärmewiderstandsdaten aus.
 - i. *INCLUDE* Plausibilität prüfen
3. *WHILE* Das System bietet eine Eingabemöglichkeit für Außen- und Innentemperatur.
 - a. *WHILE* Systemnutzer gibt Temperaturdaten ein.
 - b. Das Softwaresystem ermittelt den Temperaturverlauf.
 - c. Das Softwaresystem gibt den Temperaturverlauf aus.
 - i. *INCLUDE* Plausibilität prüfen
4. Der Use Case ist abgeschlossen.

Alternative Abläufe

Alternativer Ablauf #2.2

Wenn der Systemnutzer im Schritt 2 des Standardablaufs weitere Materialschichten hinzufügen möchte, dann

- i. Der Systemnutzer fügt eine Schicht hinzu.
- ii. Das System schafft Eingabemöglichkeiten für die Daten einer zusätzlichen Materialschicht.
- iii. Der Use Case wird im Schritt 2 des Standardablaufs fortgesetzt.

Alternativer Ablauf #2.3

Wenn der Systemnutzer im Schritt 2 des Standardablaufs eine Materialschicht entfernen möchte, dann

- i. Der Systemnutzer entfernt eine Materialschicht.
- ii. Das System entfernt die Eingabemöglichkeiten für die Daten einer Materialschicht.
- iii. Der Use Case wird im Schritt 2 des Standardablaufs fortgesetzt.

Wesentliche Szenarios

- **SC1:** Der Systemnutzer wählt die Funktion der Temperaturverlaufsrechnung aus und gibt die Materialdaten von zwei Schichten ein. Außerdem gibt er Werte für Innen- und Außentemperatur ein. Während der Systemnutzer die Berechnungsdaten eingibt, berechnet das Softwaresystem die Wärmewiderstandsdaten, die Temperaturen an den

Grenzflächen, sowie den Temperaturverlauf und gibt diese aus. Die Berechnung wurde erfolgreich abgeschlossen.

- **SC3:** Der Systemnutzer wählt die Funktion der Temperaturverlaufsberechnung aus und gibt direkt die Wärmedurchlasswiderstände von zwei Schichten ein. Außerdem gibt er Werte für Innen- und Außentemperatur ein. Während der Systemnutzer die Werte eingibt, berechnet das Softwaresystem die Wärmewiderstandsdaten, die Temperaturen an den Grenzflächen, sowie den Temperaturverlauf und gibt diese aus. Die Berechnung wurde erfolgreich abgeschlossen.
- **SC4:** Der Systemnutzer wählt die Funktion der Temperaturverlaufsberechnung aus und gibt die Materialdaten von drei oder mehr Schichten ein. Außerdem gibt er Werte für Innen- und Außentemperatur ein. Während der Systemnutzer die Berechnungsdaten eingibt, berechnet das Softwaresystem die Wärmewiderstandsdaten, die Temperaturen an den Grenzflächen, sowie den Temperaturverlauf und gibt diese aus. Die Berechnung wurde erfolgreich abgeschlossen.

4.6.3. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case muss folgende Nachbedingungen erfüllt sein:

- Die Berechnungsergebnisse werden angezeigt.

4.7. Use Case: Plausibilität prüfen

4.7.1. Kurzbeschreibung

Der Use Case beschreibt einen Vorgang zur Prüfung der Gültigkeit von Eingabedaten.

4.7.2. Kurzbeschreibung der Akteure

Systemnutzer

will korrekte Eingabedaten eingeben und das Eingaben entsprechend auf Korrektheit geprüft werden.

4.7.3. Vorbedingungen

Der Systemnutzer hat einen Use Case gestartet, der eine Dateneingabe erfordert.

4.7.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Systemnutzer Eingabedaten eingibt.
2. *WHILE* Systemnutzer gibt Eingabedaten ein
 - a. Das Softwaresystem prüft die Gültigkeit der Eingabedaten.
 - b. *IF* Softwaresystem erkennt eine ungültige Eingabe
 - i. Das Softwaresystem verhindert die Eingabe.
 - ii. *IF* Systemnutzer gibt 0,0 für Lambda ein, wird Feld rot hinterlegt
3. Der Use Case ist abgeschlossen.

4.7.5. Wesentliche Szenarios

- **SC1:** Der Systemnutzer gibt d_i ein. Der Eingabewert beträgt 2,60. Es wird keine ungültige Eingabe erkannt. Der Use Case ist abgeschlossen.
- **SC2:** Der Systemnutzer gibt d_i ein. Er versucht 1o einzugeben. Das Softwaresystem verhindert die Eingabe des Buchstaben o. Der Use Case ist abgeschlossen.
- **SC3:** Der Systemnutzer gibt 0,0 für die Wärmeleitfähigkeit (Lambda) einer Schicht ein. Das Softwaresystem markiert das betreffende Feld rot. Der Use Case ist abgeschlossen.
- **SC4:** Der Systemnutzer gibt eine negative Schichtdicke ein und bestätigt seine Eingabe. Das Softwaresystem verhindert die

Bestätigung und in dem betreffenden Feld steht 0,00.

4.7.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case muss folgende Nachbedingungen erfüllt sein:

- Die eingegebenen Daten wurden im Hinblick auf ihre Plausibilität validiert.

4.8. Use Case: Berechnungsdaten laden

4.8.1. Kurzbeschreibung

Der Use Case beschreibt einen Vorgang zum hineinladen von gültigen Datensätzen aus einer Datei.

4.8.2. Kurzbeschreibung der Akteure

Systemnutzer

möchte Daten aus einer Datei in das Softwaresystem importieren.

4.8.3. Vorbedingungen

Das Softwaresystem startet fehlerfrei und befindet sich im Startbildschirm. Der Systemnutzer hat den Use Case gestartet, wenn er im Startbildschirm auf den Button Öffnen klickt.

4.8.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Systemnutzer das System fehlerfrei gestartet hat und den Button Öffnen betätigt.
2. Der Systemnutzer möchte eine Datei einbinden.
 - a. *WHILE* Systemnutzer sucht Datei heraus
 - b. Anzeige von Dateien mit .baup-Endung
 - c. *IF* Systemnutzer wählt Datei aus
 - d. Softwaresystem überprüft die Gültigkeit der Daten und der Datei
 - e. *INCLUDE* Plausibilitätsprüfung
 - i. *IF* Softwaresystem findet einen Fehler/ kann Daten oder Datei nicht verarbeiten
 - ii. Das Softwaresystem verhindert das Laden.
 - iii. *ELSE* Daten werden übernommen.
3. Der Use Case ist abgeschlossen.

4.8.5. Wesentliche Szenarios

- **SC1:** Der Systemnutzer lädt eine Datei. Es wird eine gültige Datei erkannt. Die Daten werden im aktuellen Tab geladen. Der Use Case ist abgeschlossen.
- **SC2:** Der Systemnutzer lädt eine Datei. Dabei wird eine ungültige Datei erkannt. Das Softwaresystem verhindert das Laden der Datei und verbleibt im Startbildschirm. Der Use Case ist abgeschlossen.

4.8.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case muss folgende Nachbedingungen erfüllt sein:

- Die eingegebenen Daten wurden im Hinblick auf ihre Plausibilität validiert.

4.9. Use Case: Berechnungsdaten speichern

4.9.1. Kurzbeschreibung

Der Use Case beschreibt einen Vorgang zum speichern von gültigen Datensätzen in einer Datei.

4.9.2. Kurzbeschreibung der Akteure

Systemnutzer

möchte Daten in einer Datei speichern, um diese zu einem späteren Zeitpunkt erneut aufrufen zu können.

4.9.3. Vorbedingungen

Der Systemnutzer hat den Use Case gestartet, wenn er seine Datei speichern möchte.

4.9.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn die Berechnung erfolgreich war.
2. Der Systemnutzer möchte seine Berechnungsdaten speichern.
3. Der Systemnutzer betätigt im Auswahlménü "Datei" den Punkt "Speichern" oder "Speichern unter"
4. Der Systemnutzer wählt den Pfad aus, in dem er seine Datei speichern möchte.
5. Der Systemnutzer gibt eine Bezeichnung für die zu speichernde Datei ein.
6. Die Daten werden nach den eingegebenen Einstellungen lokal gespeichert.
7. Der Use Case ist abgeschlossen.

4.9.5. Wesentliche Szenarios

- **SC1:** Der Systemnutzer speichert seine Berechnung. Es wird kein ungültiges Argument bei der Überprüfung erkannt. Der Systemnutzer wählt den Namen und den Pfad der Datei zur Speicherung der Daten. Der Use Case ist abgeschlossen.

4.9.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case muss folgende Nachbedingungen erfüllt sein:

Die Daten sind lokal in einer Datei gespeichert und können geöffnet, sowie vom Softwaresystem geladen werden.

4.10. Use Case: Daten drucken

4.10.1. Kurzbeschreibung

Der Use Case beschreibt die Erteilung eines Druckauftrags für den Druck von Berechnungsdaten.

4.10.2. Kurzbeschreibung der Akteure

Systemnutzer

will Eingabedaten und Berechnungsergebnisse drucken.

4.10.3. Vorbedingungen

- Es liegen Berechnungsergebnisse vor.
- Es ist ein Drucker oder ein PDF-Drucker verfügbar.

4.10.4. Standardablauf (Basic Flow)

1. Der Use Case beginnt, wenn der Systemnutzer die Druckfunktion auswählt.
2. Das Softwaresystem öffnet eine Druckvorschau zum Überblick über das zu druckende Dokument.
3. Der Systemnutzer löst den Druckauftrag aus.

4. Das Softwaresystem sendet den Druckauftrag an das Betriebssystem.

5. Der Use Case ist abgeschlossen.

4.10.5. Wesentliche Szenarios

- **SC1:** Der Systemnutzer wählt die Druckfunktion aus. Es wird die Druckvorschau angezeigt. Er löst den Druckauftrag aus. Das Softwaresystem übermittelt den Druckauftrag an das Betriebssystem. Der Druckauftrag wurde erfolgreich erteilt.
- **SC3:** Der Systemnutzer wählt die Druckfunktion aus und lässt sich eine Druckvorschau anzeigen. Er lässt das Dokument in eine pdf Drucken. Die pdf wird lokal gespeichert. Der Druckauftrag wurde erfolgreich abgeschlossen.

4.10.6. Nachbedingungen

Bei erfolgreicher Durchführung des Use Case muss folgende Nachbedingung erfüllt sein:

- Der Druckauftrag wurde an das Betriebssystem gesendet.

5. Anwenderdokumentation

5.1. Benutzerhandbuch

[Benutzerhandbuch](#)

5.2. Flyer

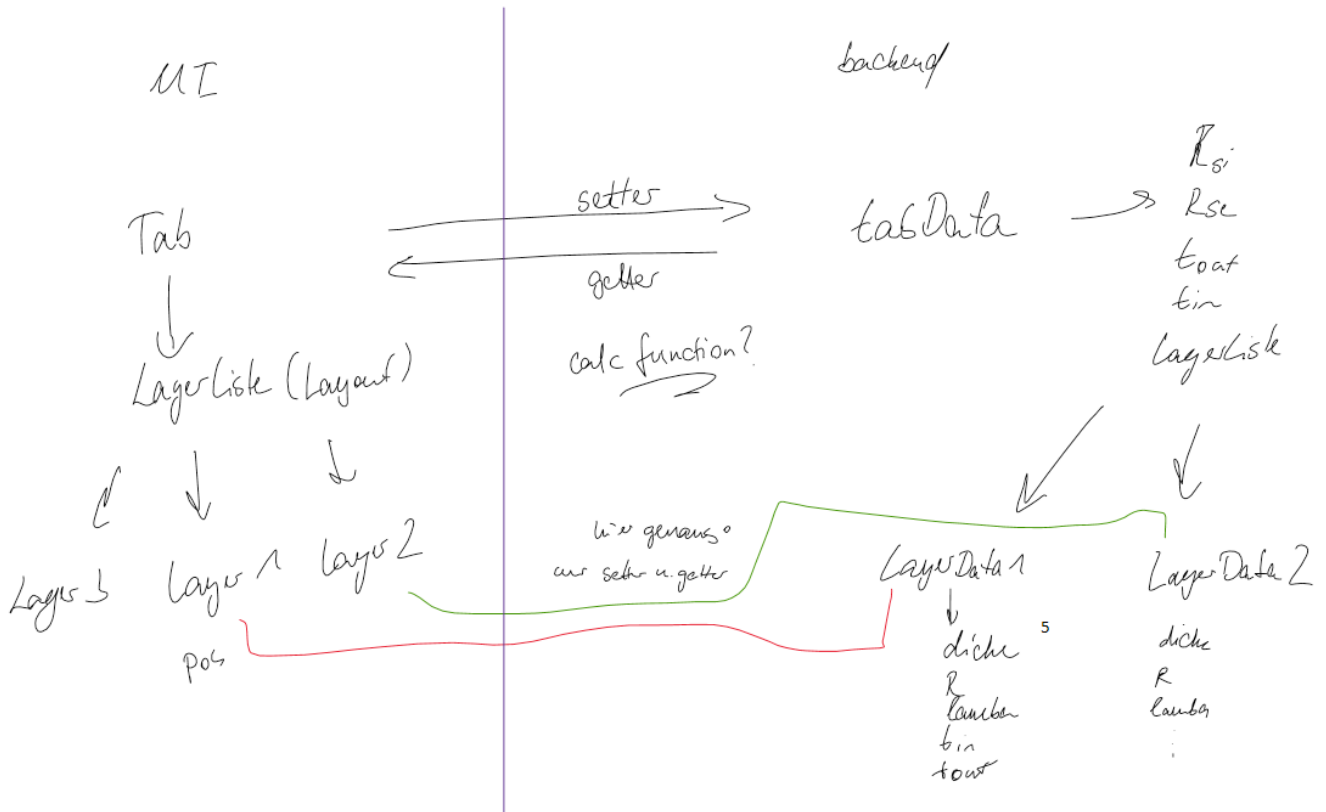
[Flyer](#)

6. Entwicklerdokumentation

Entwurfsdokumentation

Yewgenij Baburkin <yewgenij.baburkin@htw-dresden.de> 1.0, 14.08.2020 :toc: :toc-title: Inhaltsverzeichnis :sectnums:

1. Systemarchitektur



include Calculation

TabData

Attribute

- double rsi
- double rsc
- double rges
- double u
- double tin
- double toat
- <List> layers
- int need

Calculation keine Instanz!

Plot

LayerCalc (LayerData e)

tabCalc (tabData t)

at 1 → R

nicht variabel

true

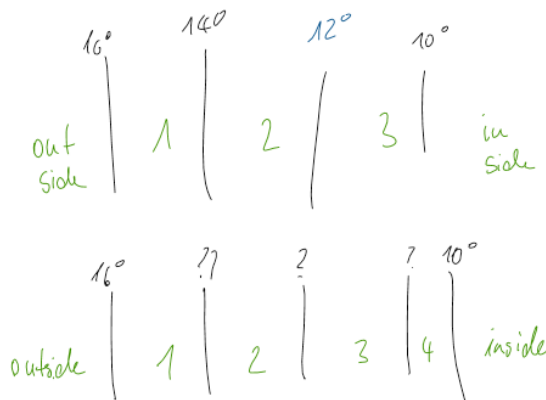
LayerData

Attribute

- double r
- double Lambda
- double dicke
- double temp in
- double temp out

next, previous? Punkt auch aus wie
für die Liste realisieren wollen...

mit material ID



bei Einfügen einer neuen Layer → neue Zwischen-Temps!

Abbildung 1. Grobe Entwurfsskizze

In einem Dev-Meeting kam die obere, durchaus informelle Skizze zustande. Ziel dieser war die Struktur der Klassen sowie deren Anordnung festzuhalten. Zu sehen ist, dass in einem Tab eine Liste von Layern geführt wird, genauer genommen im Layout des Bereichs des Tabs, in welchen Layer(Widgets) eingefügt werden. Jede Layer hat hierbei ein zugehöriges LayerData Objekt, welches die Informationen einer Layer beinhaltet. Jeder Tab besitzt ein tabData Objekt, welches alle zur Berechnung und Speicherung notwendigen Informationen beinhaltet. In diesem wird auch eine Liste geführt, welche LayerData Objekte verwaltet. Die Berechnung sollte ausgelagert werden, jedoch nicht instanziiert. Je nach dem, wie variabelntreu (pass by value

oder pass by reference) Python ist, sollte darüber nachgedacht werden, wie man diese Berechnungsfunktionen inkludiert.

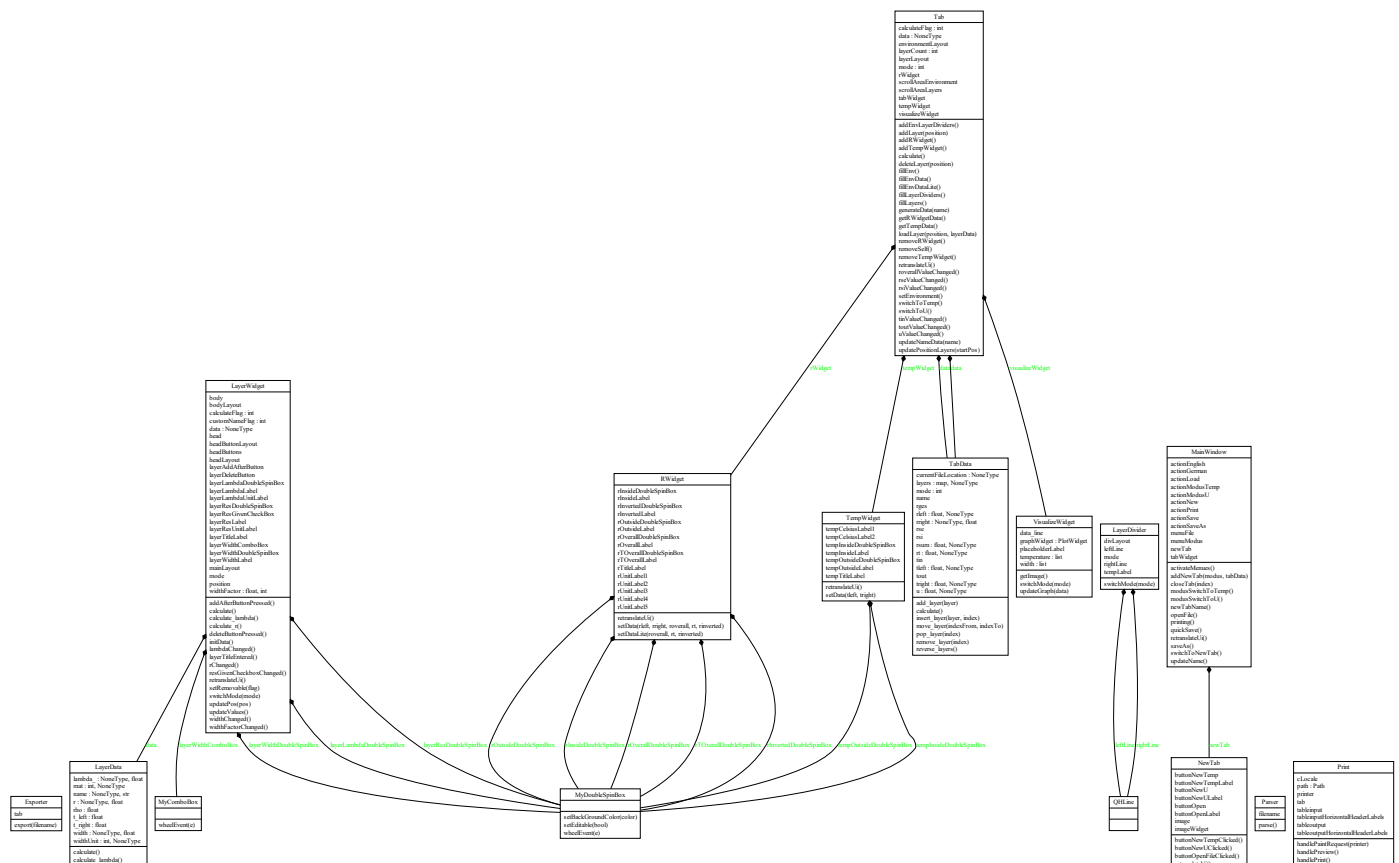


Abbildung 2. UML-Class Diagram

Dies ist die aktuelle Ansicht der benutzten Klassenstruktur. Zu erkennen ist, dass der vorherige Entwurf dementsprechend umgesetzt wurde. Kleinere Klassen sind hinzugekommen (z.B. MyDoubleSpinBox), welche jedoch einfach nur von denentsprechenden QT-Klassen abgeleitet wurden und deren Verhalten in bestimmten Situationen angepasst wurde.

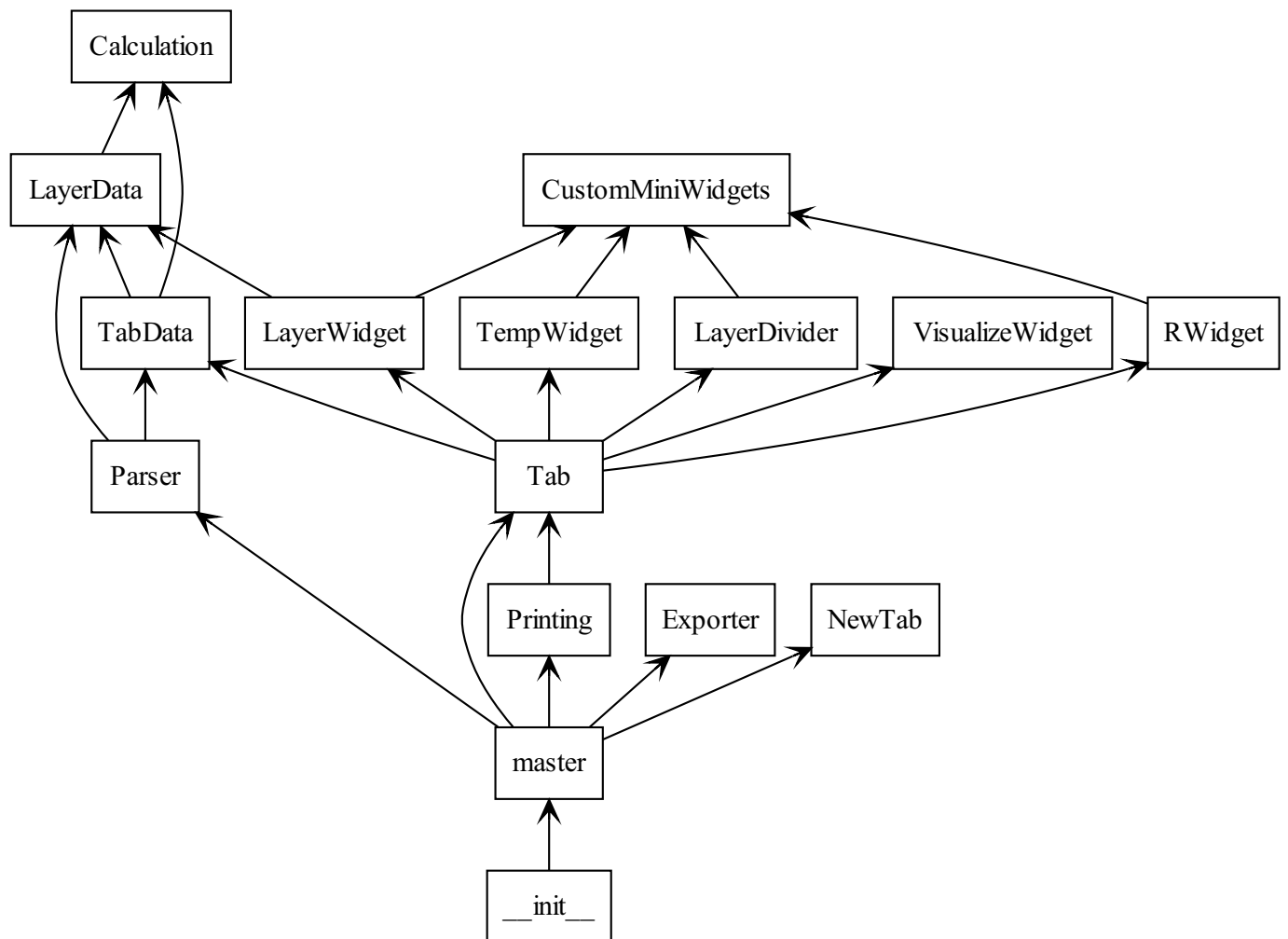


Abbildung 3. Package Diagram

In dieser Grafik kann man das Zusammenspiel zwischen den verschiedenen packages erkennen. Hierbei ist zu bemerken, dass `__init__` eigentlich `mainwindow.py` ist, jedoch im Rahmen der Generierung des Diagramms umbenannt werden musste.

2. Verwendete Frameworks/Module

- PyQt5
- numpy
- pyqtgraph

3. Struktur der gespeicherten Dateien

Tabelle 1. Tab-Struktur

Stelle	Einheit	Bezeichnung
name	String	Name des Tabs
rright	double	R- Wert rechts (siehe TabData doc)
rleft	double	R- Wert links (siehe TabData doc)
rsum	double	Summe R-Werte aller Layer

tricht Stelle	double Einheit	Temperatur rechts (siehe TabData doc) Bezeichnung
tleft	double	Temperatur links (siehe TabData doc)
mode	int	Modus, in welchem sich der Tab befindet
u	double	Rt invertiert
rt	double	Summe Rges und umgebende R-Werte
layers	"Liste"	Die einzelnen Layers

Tabelle 2. Layer-Struktur

Stelle	Einheit	Bezeichnung
1	double	Dicke der Schicht
2	int	Einheit der Dicke (0: m, 1: cm, 2: mm)
3	double	Lambda-Wert
4	double	R-Wert
5	double	linke Außentemperatur der Schicht
6	double	rechte Außentemperatur der Schicht
7	double	Rho-Wert der Schicht
8	int	Material-ID
9	String	Name der Schicht

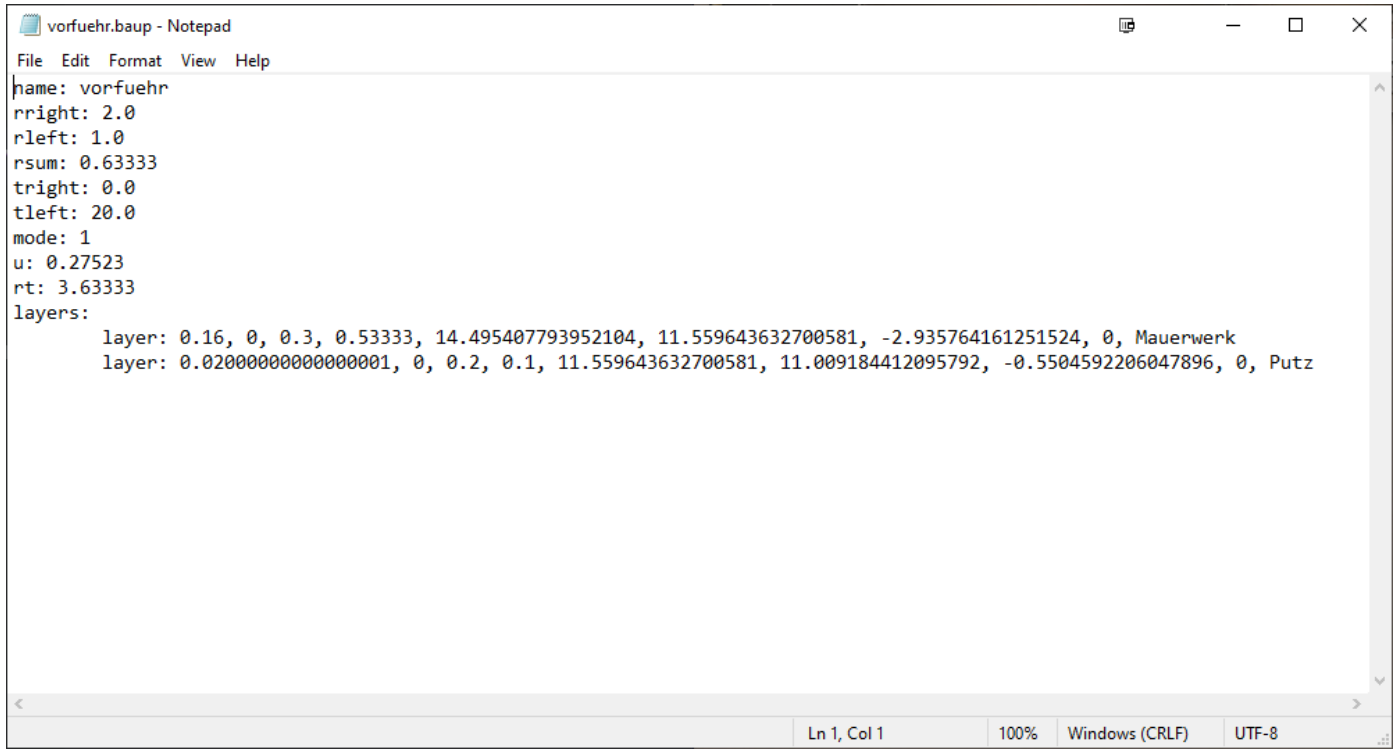


Abbildung 4. Beispiel einer gespeicherten Datei

4. Sequenzdiagramm

Da hier keine Client-Server oder Microservice o.Ä. Architektur vorliegt, wurde sich dazu entschieden, ein Sequenzdiagramm **nicht** anzufertigen. Es ergibt in Bezug auf den Aufwand keinen Mehrwert. Welche Klasse was aufruft, kann man auch dem UML-Klassendiagramm entnehmen.

5. Schnittstellen

None. Es handelt sich um eine Standalone-Anwendung mit keiner Aussicht auf Integration in andere Gebiete/Anwendungen.

6. Testdokumentation

7. Testkonzept Projekt Bauphysik

7.1. Testobjekte

Die wichtigsten und damit zu testenden Komponenten der Berechnungssoftware sind:

- Berechnungsfunktionen
- Benutzeroberfläche
- Eingabefelder bzw. Eingabemöglichkeiten
- Datenstruktur
- Importfunktion
- Exportfunktion
- Druckfunktion

7.2. Testmethoden

An dieser Stelle wird ein Überblick über die Testmethoden gegeben, die jeweils zum Test der einzelnen Testobjekte verwendet werden. Außerdem wird das "System unter Test" für die einzelnen Testobjekte spezifiziert. Zusätzlich wird aufgezeigt, in welchen TestCases, welches Testobjekt unter Testbedingungen geprüft wird.

Testobjekt	System unter Test	Testmethode	TestCase-ID
Berechnungsfunktionen	Calculation.py	manuell & automatisiert	001, 002
Benutzeroberfläche	Gesamtsystem	manuell	001 bis 008
Eingabefelder bzw. Eingabemöglichkeiten	UI	manuell	004, 007
Datenstruktur	TabData.py, LayerData.py	automatisiert	001, 002, 005, 006
Importfunktion	Gesamtsystem, Parser.py	manuell & automatisiert	006
Exportfunktion	Gesamtsystem, Exporter.py	manuell & automatisiert	005

7.3. Testplanung

7.3.1. Teststrategie

Die in diesem Projekt verfolgte Strategie bei den Tests ist, dass die wesentlichsten und wichtigsten Funktionen das meiste Gewicht für die Tests erhalten. Demzufolge entfallen die meisten Testabläufe auf automatisierte Tests der Berechnungsfunktion, sowie manuelle Tests von UI in Verbindung mit der Berechnungsfunktion. Die Richtigkeit der Berechnungen stellt den Kern der Software dar, besonders mit Blick auf das Einsatzgebiet etwa bei der Korrektur von Prüfungen ist absolute Korrektheit der Berechnungen unerlässlich.

Desweiteren ist es entscheidend, dass neben korrekten Berechnungen auch die Benutzeroberfläche mit den darin enthaltenen Eingabemöglichkeiten eindeutig ist und schon bei der Eingabe mögliche Fehler vermeidet. Aus diesem Grund entfallen viele Tests auf die Plausibilität der Eingaben. Mit sinkender Priorität wurden zusätzliche Funktionalitäten wie Drucken, sowie Export und Import von Berechnungen getestet.

Die Dokumentation der Testfälle (TestCase) und der zugehörigen Testabläufe (TestScript) erfolgt nach folgendem System. Ein TestCase beschreibt abstrakt, welcher UseCase diesem Testfall zugrunde liegt und wie der generelle Ablauf eines zugehörigen Tests ist. In einem TestCase sind mehrere TestScripte enthalten, die einen spezifischen Testablauf repräsentieren. Diese TestScripte werden dann jeweils Schritt für Schritt abgearbeitet.

7.3.2. Testumgebung

Als Testumgebung dient stets ein Desktop-Computer, auf dem das Betriebssystem Windows 10 läuft. * für die manuellen Tests wird die Berechnungssoftware selbst genutzt * für automatisierte Tests wurde vom Tester die IDE PyCharm Community Edition 2020.1 genutzt * für die Dokumentation wurde Visual Studio Code genutzt

7.3.3. Dokumentation

Die durchgeführten Tests werden in einem TestLog protokolliert. In diesem Dokument findet man eine Übersicht über alle verfügbaren Tests mit Kurzbeschreibung. Jeweils in tabellarischer Form gibt es dann für jeden Prototyp einer bestimmten Iteration eine Auflistung der an diesem Prototyp durchgeführten Testabläufe. In dieser Tabelle wird vermerkt, wer den Test durchgeführt hat und welches Ergebnis der Test hatte. Für zusätzliche Anmerkungen steht eine Kommentarspalte zur Verfügung, die speziell im Fall einer Abweichung genutzt werden muss. An dieser Stelle wird der Fehler im TestLog kurz beschrieben.

Für eine detailliertere Beschreibung des Fehlers gilt der Verweis auf das TestScript, bei dem der Fehler aufgetreten ist. Dort werden Fehler so detailgetreu wie möglich beschrieben und falls es dem Testenden möglich ist auch schon eine Analyse des Fehlers gemacht. Generell werden in jedem TestScript auch nochmal die konkreten Zeitpunkte eines Tests geloggt, unabhängig davon, ob Fehler passiert sind oder nicht.

7.3.4. Testorganisation

Die Analyse von Abweichungen erfolgt in der Regel in 3 Stufen. Abweichungen werden zu erst vom Tester festgestellt und demnach auch direkt zu erst vom Tester analysiert, nachdem der Fehler aufgetreten ist. An dieser Stelle geht es vorerst vor allem darum, ob das verwendete TestScript korrekt ist und auch korrekt angewendet wurde. Weiterhin kann der Tester im Rahmen seiner Möglichkeiten eine Analyse des vorliegenden Fehlers machen.

In jedem Fall wird der Tester dem Developer mitteilen, welche Abweichung aufgetreten ist und gegebenenfalls noch dazu angeben, wie diese Abweichung zu stande gekommen ist. Die Dokumentation und kommunikation erfolgt dabei über das für den Test der entsprechenden Komponente eröffnete Issue. Der Developer analysiert dann seinerseits die Abweichung.

Falls Tester und Developer den Grund der Abweichung nicht feststellen, wird im nächsten Meeting in der Gruppe die Abweichung diskutiert und analysiert, sodass im Team eine Lösung für den aufgetretenen Fehler gefunden wird.

Wenn Fehler existieren, wird die Beseitigung des Fehlers als Aufgabe mit in die nächste Iteration übernommen. Bis zum Ende dieser Iteration wird der Fehler dann beseitigt. Ob die Abweichung erfolgreich beseitigt wurde, wird dann getestet.

7.3.5. Testdurchführungsplanung

Für jede Iteration wird festgelegt, welche Funktionalität dem System hinzugefügt werden soll beziehungsweise welche Komponenten erweitert werden. Für diese neuen Komponenten, die gleichzeitig auch Testobjekte darstellen, werden in der Folge Testabläufe entsprechend der Teststrategie entwickelt. Wenn ein Developer eine Komponente aus seiner Sicht fertig gestellt hat, eröffnet er bei Github ein Issue für den Test der Komponente und weist dieses dem Tester zu. Dies geschieht in der Regel gegen Ende der Iteration, sodass die Komponente bis zum nächsten Kickoff-Meeting für die nächste Iteration bereits getestet ist. Der Tester testet dann mit den entsprechenden TestScripten die freigegebene Komponente.

Für jeden neuen Prototypen werden nicht nur die Tests für die neuen Funktionen oder Komponenten durchgeführt, sondern auch alle bereits an vorherigen Prototypen durchgeführten Testabläufe werden erneut an dem neuen Prototyp durchgeführt. Welche Testabläufe das sind, ist dem TestLog zu entnehmen.

8. Test-Log

8.1. Testscripte

An dieser Stelle wird eine Übersicht der verwendeten Testscripte gegeben, die über den entsprechenden Link direkt angesehen werden können.

Testscript	Kurzbeschreibung	manuell/automatisch
Testscript_001	SW online starten	manuell
Testscript_002	SW offline starten	manuell
Testscript_003	Portabilität prüfen	manuell
Testscript_004	Plausibilität korrekt	manuell
Testscript_005	Plausibilität inkorrekt	manuell
Testscript_006	Plausibilität $\Lambda = 0$	manuell
Testscript_007	Plausibilität Schichtdicke negativ	manuell
Testscript_008	Plausibilität Temperatur negativ	manuell
Testscript_009	Reihenberechnung 4 Schichten	manuell
Testscript_010	Reihenberechnung 3 Schichten	manuell
Testscript_011	Reihenberechnung 5 Schichten	manuell
Testscript_012	Reihenberechnung 2 Schichten R_i bekannt	manuell
Testscript_013	Reihenberechnung 2 Schichten R_i bekannt V2	manuell
Testscript_014	Reihenberechnung 2 Schichten	manuell
Testscript_015	Reihenberechnung 2 Schichten V2	manuell

Testscript_016	Temperaturverlauf 3 Schichten	manuell
Testscript_017	Temperaturverlauf 2 Schichten R_i bekannt	manuell
Testscript_018	Temperaturverlauf 2 Schichten R_i bekannt V2	manuell
Testscript_019	Temperaturverlauf 2 Schichten R_i bekannt V3	manuell
Testscript_020	Temperaturverlauf 4 Schichten	manuell
Testscript_021	Reihenberechnung 4 Schichten	automatisiert
Testscript_022	Reihenberechnung 3 Schichten	automatisiert
Testscript_023	Reihenberechnung 5 Schichten	automatisiert
Testscript_024	Reihenberechnung 2 Schichten R_i bekannt	automatisiert
Testscript_025	Reihenberechnung 2 Schichten R_i bekannt V2	automatisiert
Testscript_026	Reihenberechnung 2 Schichten	automatisiert
Testscript_027	Reihenberechnung 2 Schichten V2	automatisiert
Testscript_028	Temperaturverlauf 3 Schichten	automatisiert
Testscript_029	Temperaturverlauf 2 Schichten R_i bekannt	automatisiert
Testscript_030	Temperaturverlauf 2 Schichten R_i bekannt V2	automatisiert
Testscript_031	Temperaturverlauf 2 Schichten R_i bekannt V3	automatisiert
Testscript_032	Temperaturverlauf 4 Schichten	automatisiert
Testscript_033	Export & Import	automatisiert
Testscript_035	Export von Daten	manuell
Testscript_036	Import von Daten	manuell
Testscript_037	Export von Daten in cm	manuell
Testscript_038	Import von Daten in cm	manuell
Testscript_039	Auswahl falscher Datei bei Import	manuell
Testscript_040	Eingabegeräte prüfen	manuell
Testscript_041	Systemsprache prüfen	manuell
Testscript_042	Schichten hinzufügen/löschen	manuell

Testscript_043	Modus wechseln	manuell
Testscript_044	4 Nachkommastellen	manuell
Testscript_045	Wechsel Einheit Schichtdicke	manuell
Testscript_046	Druckauftrag erteilen	manuell
Testscript_047	Druckauftrag erteilen U-Modus	manuell
Testscript_048	3 Nachkommastellen Lambda	manuell
TestSuite	Testet alle automatisierten Tests	automatisiert

8.2. Testergebnisse

Es folgt eine Übersicht über die in der jeweiligen Testphase am jeweiligen Prototypen durchgeführten Testabläufe. Dabei gilt zu beachten, dass sich die Testabläufe ebenso wie der Prototyp iterativ entwickeln und folglich nur die Testscripte angewendet werden, die in der jeweiligen Iteration der Überprüfung der erreichten Ziele der jeweiligen Iteration dienen.

8.2.1. Prototyp 1

Datum	Version	Testscript	Ergebnis	Verantwortlich	Kommentar
18.04.2020	0.1	001	bestanden	CL	am Heim-PC durchgeführt
18.04.2020	0.1	002	bestanden	CL	am Heim-PC durchgeführt
18.04.2020	0.1	040	bestanden	CL	am Heim-PC durchgeführt
18.04.2020	0.1	003	bestanden	CL	am Heim-PC durchgeführt
18.04.2020	0.1	041	Failure	CL	eine Beanstandung: Auswahlreiter für die Sprache war nicht auf deutsch ("Language")

8.2.2. Prototyp 2

Datum	Version	Testscript	Ergebnis	Verantwortlich	Kommentar
28.04.2020	0.2	001	bestanden	CL	am Heim-PC durchgeführt
28.04.2020	0.2	002	bestanden	CL	am Heim-PC durchgeführt
28.04.2020	0.2	040	bestanden	CL	am Heim-PC durchgeführt
28.04.2020	0.2	003	bestanden	CL	am Heim-PC durchgeführt

28.04.2020	0.2	041	bestanden	CL	am Heim-PC durchgeführt
28.04.2020	0.2	006	bestanden	CL	am Heim-PC durchgeführt
28.04.2020	0.2	042	bestanden	CL	am Heim-PC durchgeführt
28.04.2020	0.2	043	bestanden	CL	am Heim-PC durchgeführt

8.2.3. Prototyp 3

Datum	Version	Testscript	Ergebnis	Verantwortlich	Kommentar
20.05.2020	0.3	001	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	002	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	040	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	003	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	041	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	006	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	042	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	043	bestanden	CL	am Heim-PC durchgeführt
20.05.2020	0.3	048	Failure	CL	Wärmeleitfähigkeit hat nur 2 Nachkommastellen

8.2.4. Prototyp 4

Datum	Version	Testscript	Ergebnis	Verantwortlich	Kommentar
07.06.2020	0.4	001	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	002	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	040	bestanden	CL	am Heim-PC durchgeführt

07.06.2020	0.4	003	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	041	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	006	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	042	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	043	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	009	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	010	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	011	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	012	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	013	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	014	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	015	Failure	CL	das Ergebnis für U weicht um 0,0001 vom erwarteten Wert ab
07.06.2020	0.4	016	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	017	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	018	Failure	CL	berechnete Temperatur zwischen Schicht 1 und 2 weicht erheblich von Erwartung ab
07.06.2020	0.4	019	Failure	CL	berechnete Temperatur zwischen Schicht 1 und 2 weicht

					erheblich von Erwartung ab
07.06.2020	0.4	020	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	044	Failure	CL	Ergebniss nur auf 3 Nachkommstellen genau.
07.06.2020	0.4	045	bestanden	CL	am Heim-PC durchgeführt
07.06.2020	0.4	048	bestanden	CL	am Heim-PC durchgeführt

8.2.5. Prototyp 5

Datum	Version	Testscript	Ergebnis	Verantwortlich	Kommentar
22.06.2020	0.5	001	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	002	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	040	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	003	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	041	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	006	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	042	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	043	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	009	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	010	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	011	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	012	bestanden	CL	am Heim-PC durchgeführt

22.06.2020	0.5	013	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	014	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	015	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	021	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	022	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	023	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	024	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	025	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	026	bestanden	CL	am Heim-PC durchgeführt
22.06.2020	0.5	027	bestanden	CL	am Heim-PC durchgeführt
25.06.2020	0.5	033	bestanden	CL	am Heim-PC durchgeführt
25.06.2020	0.5	044	bestanden	CL	am Heim-PC durchgeführt
25.06.2020	0.5	045	bestanden	CL	am Heim-PC durchgeführt
25.06.2020	0.5	048	bestanden	CL	am Heim-PC durchgeführt

8.2.6. Prototyp 6

Datum	Version	Testscript	Ergebnis	Verantwortlich	Kommentar
30.06.2020	0.6	046	Failure	CL	Test ergab erhebliche Abweichungen vom Soll. Ausführlicher im TestScript_046 aufgeführt.
30.06.2020	0.6	047	Failure	CL	Test ergab erhebliche

					Abweichungen vom Soll. Ausführlicher im TestScript_047 aufgeführt.
30.06.2020	0.6	035	bestanden	CL	am Heim-PC durchgeführt
30.06.2020	0.6	036	bestanden	CL	am Heim-PC durchgeführt
30.06.2020	0.6	039	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	001	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	002	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	040	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	003	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	041	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	006	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	042	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	043	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	044	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	045	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	048	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	009	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	010	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	011	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	012	bestanden	CL	am Heim-PC durchgeführt

					durchgeführt
01.07.2020	0.6	013	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	014	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	015	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	016	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	017	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	018	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	019	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	020	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	021	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	022	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	023	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	024	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	025	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	026	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	027	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	028	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	029	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	030	bestanden	CL	am Heim-PC durchgeführt

01.07.2020	0.6	031	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	032	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	033	bestanden	CL	am Heim-PC durchgeführt
01.07.2020	0.6	Testsuite	bestanden	CL	am Heim-PC durchgeführt

8.2.7. Prototyp 7

Datum	Version	Testscript	Ergebnis	Verantwortlich	Kommentar
26.07.2020	0.7	046	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	047	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	035	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	036	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	039	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	001	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	002	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	040	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	003	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	041	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	006	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	042	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	043	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	044	bestanden	CL	am Heim-PC durchgeführt

					durchgeführt
26.07.2020	0.7	045	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	048	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	009	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	010	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	011	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	012	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	013	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	014	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	015	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	016	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	017	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	018	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	019	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	020	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	021	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	022	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	023	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	024	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	025	bestanden	CL	am Heim-PC

					durchgeführt
26.07.2020	0.7	026	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	027	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	028	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	029	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	030	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	031	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	032	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	033	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	Testsuite	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	037	bestanden	CL	am Heim-PC durchgeführt
26.07.2020	0.7	038	Failure	CL	grober Fehler aufgetreten, da Felder der Schichtdicke nicht mit Werten befüllt sind

9. Betriebsdokumentation