

Management-Applikation für Schokoladen-3D-Drucker CHOCOFORMER mit ASP.NET Core und React



CHOCOFORMER[®]

BACHELORARBEIT

ABTEILUNG INFORMATIK

HOCHSCHULE FÜR TECHNIK RAPPERSWIL

Frühjahrssemester 2020

Autoren: Marc Weber und Vanessa Janknecht
Betreuer: Olaf Zimmermann
Projektpartner: IWK
Experte: Reto Bättig, CEO M&F Engineering AG
Gegenleser: Thomas Corbat

ABSTRACT

Der CHOCOFORMER ist ein an der HSR entwickelter Schokoladen-3D-Drucker und dient als Vorzeigemodell an Messeständen. Der Druckauftrag ist in mehrere Schritte aufgeteilt, welche technisches Knowhow, unter anderem über CAD- und Slicing-Software, erfordern. Zuerst wird ein 3D-Modell erstellt, welches anschliessend in einer separaten Applikation in mehrere, druckbare Schichten aufgeteilt wird; schliesslich muss die Datei manuell auf den CHOCOFORMER kopiert und der Auftrag dort gestartet werden. Daher kann ein Messebesucher den CHOCOFORMER nicht allein bedienen. Der gesamte Druckauftrag für eine persönliche Grussbotschaft müsste von einer Fachperson durchgeführt werden, welche am Messestand jedoch mit Verkaufsgesprächen beschäftigt ist. Mit der CHOCOFORMER App wird der Druckauftrag zu einem Schritt vereinfacht, so dass nur noch ein zweidimensionales Sujet von Hand erstellt werden muss und die Applikation den bestehenden Druckauftrag automatisch ausführen kann. Die Web-Applikation ist im Back-End mit ASP.NET Core und im Front-End mit React und TypeScript implementiert und wird auf einem Raspberry Pi ausgeführt, welcher direkt mit dem Duet3D-Steuerungs-Board des CHOCOFORMERS verbunden ist. Somit ist es für Messebesucher ohne technische Vorkenntnisse möglich, einen personalisierten Gruss aus Schokolade zu drucken. Die Validierung der Software erfolgte mit Unit-, Integration- und Usability-Tests. Zukünftig können noch zusätzliche Gestaltungsmöglichkeiten für den Benutzer eingebaut oder die Applikation an einen Web-Shop angebunden werden.



INHALT

1	Glossar und Abkürzungsverzeichnis.....	8
2	Management Summary.....	10
2.1	Problemstellung, Vision	10
2.2	Ziele	10
2.3	Rahmenbedingungen	10
2.4	Methode / Vorgehen.....	11
2.5	Stand der Technik – Was gibt es schon?.....	11
2.5.1	Konkurrenzanalyse	11
2.6	Eigener Lösungsansatz	11
2.7	Resultate	12
3	Anforderungsspezifikation	13
3.1	Anforderungen	13
3.2	Annahmen	13
3.3	Use Cases.....	14
3.3.1	Use Cases - Brief Format	14
3.3.2	Use Cases – Fully Dressed	16
3.4	Weitere mögliche Funktionen.....	18
3.5	System-Sequenzdiagramme.....	18
3.6	Nicht-Funktionale Anforderungen	19
3.6.1	Funktionalität	19
3.6.2	Zuverlässigkeit.....	19
3.6.3	Benutzbarkeit	20
3.6.4	Effizienz	20
3.6.5	Änderbarkeit	20
3.6.6	Übertragbarkeit.....	21
4	Design.....	22
4.1	Architektur	22
4.1.1	Verantwortlichkeiten und Interaktionen	26
4.2	Konzepte.....	28
4.2.1	Druck-Aufträge	28

4.2.2	Locking-Mechanismus.....	29
4.3	UI-Design	30
4.3.1	Benutzer Front-End	30
4.3.2	Administrator-Front-End.....	30
5	Programmiersprachen, Frameworks und Tools.....	31
5.1	Front-End / GUI	31
5.2	Back-End.....	32
5.3	Datenbank	33
5.4	Slicer	34
5.5	CAD.....	35
5.6	Duet 2 vs. Duet 3	36
5.7	Netzwerk	37
6	Implementation und Entwicklung.....	38
6.1	Erläuterungen wichtiger konkreter Klassen	38
6.2	Konfigurationen.....	38
6.3	Lizenzen	39
6.3.1	Verwendete Software mit Lizenzen	40
6.4	Sicherheit.....	41
6.5	Fehlerbehandlung	41
6.5.1	Fehlerverhalten nachgelagerter Software	41
6.5.2	Fehlerverhalten des CHOCOFORMERS.....	42
6.6	Design Patterns und Best Practices.....	42
6.6.1	Dependency Injection	42
6.6.2	Material Design	43
6.6.3	Presentational und Container Components	43
6.7	CI/CD.....	44
6.7.1	Qualitäts-Sicherung.....	44
6.7.2	Release	45
6.7.3	Installation.....	45
6.7.4	Betrieb unter DuetPi	46
7	Testing.....	47
7.1	Ziele und Testkonzept	47

7.2	Automatische Testverfahren.....	47
7.2.1	Integrationstests	47
7.3	Manuelle Testverfahren	47
7.3.1	Testen am CHOCOFORMER.....	48
7.3.2	Testen ohne 3D-Drucker	48
7.4	Usability Tests	49
7.4.1	Wissensziele	49
7.4.2	Testpersonen rekrutieren	49
7.4.3	Ablauf	50
7.4.4	Testaufbau.....	51
7.4.5	Testszenarien	51
7.4.6	Nachinterviews.....	51
7.4.7	Durchführung der Tests	52
7.4.8	Auswertung der Testergebnisse.....	52
7.4.9	Erarbeiten von Verbesserungsvorschlägen.....	55
8	Resultate und Weiterentwicklung	57
8.1	Resultate	57
8.2	WebSocket vs. HTTP.....	57
8.3	Vergleich Wireframes - Resultat	58
8.3.1	Benutzer Front-End	58
8.3.2	Administrator Front-End	60
8.4	Welche Use Cases wurden erreicht?	61
8.5	Möglichkeiten der Weiterentwicklung	62
8.5.1	Benutzer-Front-End.....	62
8.5.2	Admin-Front-End.....	63
8.5.3	Konfiguration.....	63
8.5.4	Weitere Möglichkeiten.....	64
9	Projektmanagement – Planung/Soll	65
9.1	Prozessmodell	65
9.2	Team, Rollen und Verantwortlichkeiten	65
9.3	Aufwandschätzung	66
9.4	Zeitplan.....	66

9.5	Meilensteine.....	66
9.6	Projektplan	67
9.7	Risiken	69
9.8	Chancen.....	70
9.9	Eliminierte Risiken.....	71
9.9.1	R3 - Integration des Slicers funktioniert nicht wie geplant.....	71
9.10	Eingetretene Risiken	71
9.10.1	R9 - COVID-19 Quarantäne	71
10	Projektmonitoring - Ist	73
10.1	Soll-Ist-Zeitvergleich.....	73
10.2	Codestatistik.....	75
10.2.1	Verteilung der Programmiersprachen	75
10.2.2	Metriken.....	76
10.2.3	Testabdeckung	77
11	Softwaredokumentation	78
11.1	Firmware Updaten	78
11.2	Installation und Konfiguration	78
11.2.1	Getting started with development	78
11.2.2	Using the app on a Raspberry Pi	79
11.2.3	Structure.....	84
12	Reflexion.....	85
12.1	Marc Weber	85
12.2	Vanessa Janknecht	85
13	Literatur- und Quellenverzeichnis.....	87
14	Verzeichnisse.....	90
14.1	Abbildungen	90
15	Erklärung zur Urheberschaft	91
16	Anhang	92
16.1	Wireframes.....	92
16.1.1	Wireframes Benutzer Front-End.....	92
16.1.2	Wireframes Administrator Front-End	98
16.2	Getting Started with Duet 3 & Raspberry Pi [33].....	100



16.2.1	Was gebraucht wird	100
16.2.2	Aufbau des 3D-Druckers	100
16.2.3	Duet 3 Mainboard	101
16.2.4	Vorbereitung	101
16.2.5	Aufsetzen.....	102
16.3	Versuchspläne	103
16.3.1	Versuchsplan 13.05.2020 – ab 9:00 – 2.112	103
16.3.2	Versuchsplan 27.05.2020 – ab 9:00 – 2.112	108
16.4	Ausgefüllte Fragebogen	114

1 GLOSSAR UND ABKÜRZUNGSVERZEICHNIS

Administrator	Eine geschulte Person (m/w/d), welche den CHOCOFORMER einrichten und konfigurieren kann. Zumindest bis zum Abschluss der Bachelorarbeit wird dies hauptsächlich das Maintainer-Team sein, also Mitarbeiter aus den HSR-Instituten. Zu einem späteren Zeitpunkt könnten dies auch externe Betreiber dieser Plattform sein.
Benutzer	Eine Person (m/w/d), welche die CHOCOFORMER App bedient und keine Vorkenntnisse zum CHOCOFORMER oder 3D-Druck im Allgemeinen hat. Benutzer könnten zum Beispiel Messebesucher oder potenzielle Kunden des CHOCOFORMER-Produkts sein. Da der CHOCOFORMER als Vorzeigeobjekt für die HSR dienen soll, werden auch Berufsschüler bzw. zukünftige Studierende angesprochen, also jüngere Personen, welche möglicherweise noch kein Vorwissen zum Thema haben.
CLI	Command-line Interface. Software-Programme können text-basierte Optionen anbieten, mit welchen das Programm gestartet werden kann. Neben Input-Dateien können so auch Konfigurationen oder Ausgabe-Pfade festgelegt werden. Programme mit einem CLI eignen sich zur Automatisierung, da kein Eingriff in eine Benutzeroberfläche notwendig ist.
DCS	Duet Control Server. Die Hauptanwendung des Duet Software Frameworks, welche die Inter-Prozess-Kommunikation übernimmt und mit der RepRap-Firmware kommuniziert.
Druckauftrag	Eine Anweisung, welche durch den Benutzer erstellt und den Administrator ausgelöst wird, um ein erstelltes Sujet durch den CHOCOFORMER ausdrucken zu lassen.
DSF	Duet Software Framework. Eine Sammlung von Programmen, mit welcher das Duet-Board über DuetPi betrieben werden kann [1].
Duet3D	Die Firma, welche hinter der Entwicklung der Controller-Boards steht. Zudem ist dies auch der Überbegriff für die unterschiedlichen Boards, wie dem Duet 2 oder dem Duet 3.
Duet 3	Das zentrale Controller-Board, welche die Motoren des CHOCOFORMERs steuert. Während einige Chocofomer-Drucker noch die Vorgänger-Plattform Duet 2 verwenden, werden neue Drucker mit diesem Board bestückt.
DuetPi	Ein Raspbian-basiertes Betriebssystem, welches auf dem Raspberry Pi laufen gelassen werden kann und mit dem DSF vorkonfiguriert ist, um ein Duet-Board anzusteuern.
DWC	Duet Web Control. Eine in Vue.js implementierte Web-Oberfläche, mit welcher das Duet-Board bzw. die RepRapFirmware gesteuert und verwaltet werden kann [2].
DWS	DuetWebServer. Ein in ASP.NET Core implementiertes Back-End, welches HTTP-Endpoints für den DWC zur Verfügung stellt.

Einlegebauteil	Die Platte, die in den CHOCOFORMER eingesetzt wird und auf der anschließend das gewählte Sujet gedruckt wird. Es gibt verschiedene Einlegebauteile mit Halterungen für Kekse und Schokoladentafeln, damit diese an der richtigen Stelle positioniert sind und nicht während eines Druckes verschoben werden.
G-Code	Eine Befehlssprache, welche verwendet wird, um 3D-Drucker anzusteuern [3]. Diese Befehle werden aus einem Buchstaben, typischerweise "G", sowie einer Zahl gebildet. Damit werden unter anderem der Druckkopf angesteuert, die Motoren bewegt oder aber auch Daten von der SD-Karte des Duet-Boards manipuliert.
Homing	Der Kalibriervorgang der Achsen eines 3D-Druckers. Bevor er anfangen kann zu drucken, müssen seine Achsen gehomet werden. Das bedeutet, alle Achsen werden an ihren Null-Punkt gefahren. Dadurch wird sichergestellt, dass das Koordinatensystem des 3D-Druckers richtig eingestellt wurde.
ILT	Institut für Laborautomation und Mechatronik. Das HSR-Institut, welches die Entwicklung des CHOCOFORMERs insbesondere mit der Konfiguration des Duet Controller Boards unterstützt.
IWK	Institut für Werkstofftechnik und Kunststoffverarbeitung. Das HSR-Institut, in welchem der CHOCOFORMER entwickelt und hergestellt wird.
Macro	Eine Sequenz von G-Code-Befehlen, welche verwendet werden, um den 3D-Drucker zu steuern [4]. Diese Macros werden in einzelnen Dateien auf der SD-Karte des Duets abgelegt.
PLA	Polylactide. Ein von 3D-Druckern normalerweise verwendetes Material.
PrusaSlicer	Verwendete Slicing-Software. Modelle im STL-Format werden hiermit in Schichten aufgeteilt, welche dann während dem Druck aufeinander aufgebaut werden. Zudem werden auch Hilfsstrukturen zum Beispiel in den ansonsten hohlen Innenräumen von 3D-Objekten aufgebaut. Das Resultat dieser Software wird als G-Code abgespeichert.
Raspberry Pi	Ein Einplatinencomputer bzw. Single Board Computer, welcher während dieser Arbeit in der Version 4B als Laufzeitumgebung für das Endergebnis verwendet wird. Oft wird darauf Raspbian eingesetzt, ein Linux-basiertes Betriebssystem.
RepRap Firmware	Objekt-orientierte C++ Software, geschrieben für 32-Bit Controller. Duet verwendet diese Firmware zur Steuerung ihrer Boards.
STL	Standard Triangulation Language. Ein Dateiformat, welches von vielen CAD-Programmen, wie zum Beispiel dem Siemens NX, generiert wird.
Sujet	Ein erstelltes Motiv, das der Benutzer an den CHOCOFORMER sendet.
SVG	Scalable Vector Graphics. Zweidimensionale Vektorgrafik basierend auf XML. Der Vorteil von SVG-Dateien ist, dass sie ohne Qualitätsverlust skaliert werden können. Wird von allen relevanten Browsern unterstützt.

2 MANAGEMENT SUMMARY

2.1 PROBLEMSTELLUNG, VISION

Das IWK hat im Rahmen einer Bachelorarbeit einen Druckkopf entwickelt, der flüssige Schokolade drucken kann, und diesen dann in einen bereits bestehenden 3D-Drucker namens CHOCOFORMER 1 eingebaut. Im Jahr 2019 wurde der Drucker durch das IWK und das ILT neu entwickelt. Die Erfahrungen aus der ersten Version konnten in der CHOCOFORMER 2 eingearbeitet werden.

Die Bedienung dieser Drucker ist momentan noch sehr mühselig und mit vielen Arbeitsschritten verbunden:

- Erstellen eines 3D-Modells im CAD
- Manuelles slicen der 3D-Datei im Slicer
- Übertragen der gesliceten Datei auf den 3D-Drucker
- Starten des Drucks via Webinterface oder Panel

Da der CHOCOFORMER vor allem für Messeauftritte und dergleichen verwendet werden soll, ist dieser Prozess zu aufwändig, um durch den Messebesucher personalisierte Sujets drucken zu lassen. Das Ziel ist eine Applikation, die all diese Schritte selbständig ausführen kann. Somit wird ein Besucher einer Messe, nachfolgend als Benutzer bezeichnet, auch in der Lage sein, ohne Einführung in die Funktionsweise der Plattform, einen Druck auszulösen. Dadurch können sich die Administratoren während eines Messeauftritts zum Beispiel auf Verkaufsgespräche konzentrieren und gleichzeitig wird den Benutzern durch die Erstellung eines eigenen Sujets ein positives Erlebnis geboten. Darüber hinaus entfällt für die Administratoren des CHOCOFORMERs das Wechseln zwischen mehreren Applikationen, wie zum Beispiel der CAD-Software oder dem Slicer.

2.2 ZIELE

Hauptziel ist, dass Benutzer ihren Namen oder einen Gruss in der Applikation schreiben können (mit einem vorhandenen Alphabet oder Freihand) und dieser anschliessend auf eine Tafel Schokolade oder einen Keks gedruckt wird.

Optionale Ziele sind, dass auch 3-dimensionale Drucke unterstützt werden.

Ein langfristiges Ziel für diese Applikation, welches nicht Teil dieser Bachelorarbeit ist, wird ein Webshop sein. Damit können 3D-Drucke direkt aus einer im Internet verfügbaren Plattform bestellt werden.

2.3 RAHMENBEDINGUNGEN

Die Entwicklung der CHOCOFORMER App darf eigentlich nichts kosten. Tools und Frameworks sollten möglichst Open Source sein. Die benötigte Hardware (zum Beispiel das Duet Controller Board oder der Raspberry Pi) wird vom IWK für diese Arbeit zur Verfügung gestellt.

Das System muss sehr robust sein. Es gibt kein Know-How im IWK, um allfällige Software-Probleme zu lösen. Die Applikation muss also ohne Wartung zuverlässig funktionieren.

2.4 METHODE / VORGEHEN

Das finale Produkt wird nach der agilen Methode Scrum erarbeitet, in welcher sich das Entwicklerteam iterativ im Zwei-Wochen-Rhythmus fortbewegt.

2.5 STAND DER TECHNIK – WAS GIBT ES SCHON?

2.5.1 KONKURRENZANALYSE

Kurzbeschreibung

ChocEdge [5] ist ein Unternehmen aus China und dem Vereinigten Königreich, das nach einem ähnlichen Prinzip arbeitet wie der CHOCOFORMER. Es ist ein spezieller 3D-Drucker, der Schokolade drucken kann. Sie bieten eine Software, mit der Benutzer ihre eigenen 3D-Modelle erstellen können. Dabei können die Benutzer zwischen 3 verschiedenen Applikationen wählen. Die erste bietet vorgefertigte Sujets, aus denen der Benutzer eines auswählen kann, die zweite kann Texte drucken und die dritte ermöglicht dem Benutzer, eine Zeichnung zu erstellen.

Defizite

Der ChocEdge bietet viele Möglichkeiten, seinen persönlichen 3D-Druck zu erstellen. Man kann zwischen drei verschiedenen Anwendungen wählen, jedoch kann man diese nicht miteinander kombinieren. Insbesondere die Möglichkeit, die verschiedenen Anwendungen miteinander zu kombinieren, wäre sehr mächtig, da so zum Beispiel ein Geburtstagsglückwunsch-Sujet vorbereitet werden kann, welches der Benutzer anschliessend mit einem Namen personalisieren könnte.

Auch besitzt der ChocEdge nur einen Druckkopf, somit können alle Sujets nur einfarbig gedruckt werden. Mit mehreren Druckköpfen wären die Kreativität der Benutzer weniger eingeschränkt. Zudem werden alle Sujets direkt auf die Druckplatte des 3D-Druckers gedruckt, es gibt keine Unterlagen, auf die gedruckt werden kann. Wenn zum Beispiel Kekse als Unterlage möglich wären, so könnten mit einer dünneren Schokoladenschicht zusätzliche Gestaltungsideen umgesetzt werden.

2.6 EIGENER LÖSUNGSANSATZ

Konzept für die Umsetzung

Für die CHOCOFORMER App sind zwei Single Page Applikationen geplant. Eine Applikation für den Benutzer, der ein Sujet erstellen und drucken möchte. Und eine für den Administrator, der für den Betrieb des CHOCOFORMERS verantwortlich ist, um Konfigurationen vorzunehmen und Druckaufträge zu verwalten. Der Benutzer kann mit Hilfe von verschiedenen Tools ein Sujet erstellen. Nachdem er dieses gesendet hat, reiht sich der Druckauftrag in eine Queue ein. Der Administrator kann diese Queue sehen und steuern. Sobald ein Druck beendet wurde, kann er die Druckplatte im CHOCOFORMER

auswechseln und den nächsten starten. Dadurch müssen die Benutzer nicht warten, bis ein Druck abgeschlossen wurde, um den nächsten Druck zu erstellen.

Wichtige Elemente

Der Benutzer kann sein Sujet mit einer Zeichnung und einem Text gestalten. Zudem kann er bis zu zwei Schokoladensorten auswählen, in denen das Sujet gedruckt werden soll.

Der Administrator kann über das Dashboard die Queue verwalten und neue Druckaufträge starten. Auch hat er einen Überblick über die wichtigsten Parameter des CHOCOFORMERs, wie die Temperatur der Druckköpfe. In einem Konfigurationspanel kann er zudem verschiedene Einstellungen bezüglich der Optionen, mit denen der Benutzer sein Sujet gestalten kann, anpassen.

2.7 RESULTATE

Zusammenfassung

Das Endprodukt reicht über das Minimum Viable Product (MVP) hinaus. Die Kernelemente der Applikation konnten implementiert werden.

Zielerreichung

Die wichtigsten Use Cases konnten erfolgreich umgesetzt werden. Der Benutzer ist in der Lage ein Sujet zu erstellen. Dabei stehen ihm zwei Tools zur Verfügung, die er sogar miteinander kombinieren kann. Da der CHOCOFORMER zwei Druckköpfe besitzt, ist es möglich, jeden mit einer anderen Schokoladensorte zu füllen. Der Benutzer kann nicht nur zwischen den Schokoladensorten wählen, er kann auch beide Schokoladen in einem Druck gleichzeitig verwenden. Nachdem der Benutzer seinen Druck fertiggestellt und abgeschickt hat, ist seine Aufgabe erledigt und er muss nur noch auf seinen fertigen Druck warten.

Der Administrator hat eine eigene Applikation erhalten, in der er den CHOCOFORMER überwachen und die Druck-Jobs koordinieren kann. Alle Druck-Jobs befinden sich in einer Queue, dabei werden dem Administrator verschiedene Informationen über die Jobs mitgeteilt: die ID, die ungefähre Druckdauer, die Zeit, bis der Druck starten kann und welche Druckunterlage verwendet wird. Zudem kann der Administrator jeden Druck mehrfach starten, falls etwas nicht funktioniert hat. Bei Fehlverhalten signalisiert die Applikation farblich, dass zum Beispiel ein Sensorwert oder ein Status fehlerhaft ist. Somit kann der Administrator direkt sehen, falls etwas nicht in Ordnung ist.

Ausblick

Es gibt noch sehr viele Möglichkeiten die CHOCOFORMER App zu erweitern, sowohl auf der Benutzer- als auch auf der Administrator-Seite. Insgesamt sind beide Applikationen so aufgebaut, dass sie einfach erweiterbar sind, da die Komponenten wiederverwendbar sind. Man kann die Optionen, mit denen der Benutzer sein Sujet erstellen kann, beliebig erweitern, unter anderem mit vorgefertigten Sujets, Textbearbeitungskomponenten oder verschiedenen Pinseln.

3 ANFORDERUNGSSPEZIFIKATION

Der CHOCOFORMER wird von zwei Arten von Personen bedient, einem Benutzer und einem Administrator. In diesem Projekt soll der Ablauf automatisiert werden, welcher es vereinfacht, den CHOCOFORMER an einem Messestand zu konfigurieren und zu bedienen. In den nachfolgenden Unterkapiteln ist beschrieben, welche Rahmenbedingungen und Anforderungen für dieses Projekt gelten.

3.1 ANFORDERUNGEN

Der Druck startet erst, nachdem ein Mitarbeiter bzw. ein Administrator bestätigt hat, dass der Druck starten kann (über Touchpanel am CHOCOFORMER oder aus der Applikation). So wird gewährleistet, dass der Druck auf der richtigen Oberfläche stattfindet.

Der Benutzer soll Schritt für Schritt (evtl. mit Hilfe eines digitalen Assistenten) durch die Anwendung geführt werden.

3.2 ANNAHMEN

1. Die Konfiguration direkt an der Hardware des CHOCOFORMERs wird als korrekt angesehen, die Applikation überprüft dies nicht.
2. Die Positionierung von zu bedruckenden Lebensmitteln wird als korrekt angenommen, es findet keine Überprüfung statt.

Die CHOCOFORMER App muss keinen erfolgreichen Druck gewährleisten, wenn Annahme 1 und/oder 2 nicht stimmen.

3.3 USE CASES



Abbildung 1 Use Case Diagramm

3.3.1 USE CASES - BRIEF FORMAT

UC01: Druckstatus verwalten

Der Administrator überwacht den CHOCOFORMER (Temperatur, Geschwindigkeit, Fehler, etc.). Die Applikation informiert den Administrator über den Status eines Drucks, welches Einlegebauteil verwendet wird und eingesetzt werden muss und der Administrator autorisiert die Freigabe eines Drucks.

UC02: Druckköpfe verwalten

Der Administrator verwaltet, welche Druckköpfe eingesetzt sind und welche Schokolade bzw. welche Düsengrösse die Druckköpfe verwenden. Das System stellt damit die passenden Konfigurationsdateien für den Slicer bereit, wobei diese bereits vorgängig vom Administrator hinterlegt wurden.

UC03: Schriftarten verwalten

Der Administrator lädt Vorlagen für Schriftarten hoch oder löscht diese. Dabei wählt er auch aus, welche Schriftarten dem Benutzer zur Verfügung stehen sollen und aus welchen Optionen (Fett, Kursiv, ausgefüllt, etc.) dieser wählen kann.

UC04: 3D-Modelle verwalten

Der Administrator lädt Vorlagen für 3D-Modelle hoch oder löscht diese. Dabei wählt er auch aus, welche 3D-Modelle dem Benutzer zur Verfügung stehen sollen.

UC05: Einlegebauteile verwalten

Der Administrator lädt verschiedene Templates von Einlegebauteile hoch oder löscht diese. Dabei wählt er auch aus, welche Einlegebauteile dem Benutzer zur Verfügung stehen sollen.

UC06: Text/Gruss eingeben

UC06.1: Text/Gruss mit vorgegebener Schriftart erstellen

Der Benutzer wählt eine Schriftart aus und schreibt einen Text. Er verändert den Text mit den vom Administrator vorgegebenen Optionen.

UC06.2: Text/Gruss Freihand erstellen

Der Benutzer zeichnet oder schreibt Freihand. Er verwendet dabei keine vorgegebene Schriftart. Die Applikation lässt diese Interaktion nur in einem bestimmten Bereich zu, welcher der Form des ausgewählten Einlegebauteils entspricht.

UC07: 3D-Modell auswählen (optional)

Der Benutzer wählt ein 3D-Modell aus einer Bibliothek aus. Er verändert das 3D-Modell mit den vom Administrator vorgegebenen Optionen.

UC08: Druckauftrag senden

Der Benutzer sendet sein Motiv/Modell/Schriftzug zum Drucken an den CHOCOFORMER. Die Applikation teilt dem Benutzer seine Druck-ID und die Wartezeit, bis der Druck abgeschlossen ist, mit.

UC09: Bilddatei in Druckdatei transformieren (optional)

Der Benutzer nimmt ein Bild mit einer Kamera auf und verändert oder löscht dieses. Er verändert das Bild mit den vom Administrator vorgegebenen Optionen.

3.3.2 USE CASES – FULLY DRESSED

UC01: DRUCKSTATUS VERWALTEN

Primärer Akteur	Administrator
Beschreibung	Der Administrator erhält eine Übersicht über den CHOCOFORMER, den laufenden Druck und den Druckaufträgen in der Queue.
Vorbedingungen	Die Applikation muss eine Verbindung zum CHOCOFORMER aufgebaut haben.
Nachbedingungen	Die Applikation muss eine Verbindung zum CHOCOFORMER haben.
Ablauf	<p>Druckauftrag abgeschlossen</p> <ol style="list-style-type: none"> Dem Administrator wird signalisiert (mit Ton vom CHOCOFORMER, visuell), dass ein Druckauftrag abgeschlossen wurde und vom CHOCOFORMER entnommen werden kann. <p>Druckauftrag auslösen</p> <ol style="list-style-type: none"> Dem Administrator wird signalisiert (mit Ton vom CHOCOFORMER, visuell), dass ein Druckauftrag bereit ist zum Drucken. Dem Administrator wird angezeigt, welches Einlegebauteil verwendet werden soll und wie lange der Druck dauert. <ol style="list-style-type: none"> Wenn die Einstellung vorgenommen wurde, dass der Administrator das Sujet erst freigeben muss, bevor es gedruckt werden kann, erscheint zusätzlich das Sujet auf der Anzeige. Nachdem der Administrator das richtige Einlegebauteil in den CHOCOFORMER eingesetzt hat, löst er den Druck mit einem Klick auf «Druck ausführen» aus. <ol style="list-style-type: none"> Möchte der Administrator das Sujet nicht drucken, kann er auf «Druck nicht ausführen» klicken. <p>CHOCOFORMER Monitoring</p> <ol style="list-style-type: none"> Der Administrator erhält eine Übersicht über die Parameter des CHOCOFORMER. Der Administrator kann Parameter, falls nötig, ändern
Ausnahmen	<p>Falls der CHOCOFORMER sich fehlerhaft verhält (der CHOCOFORMER sendet Fehlermeldungen, diese können abgefangen und dargestellt werden), wird dies dem Administrator mitgeteilt.</p> <p>Falls die Verbindung zum CHOCOFORMER getrennt wird, wird dies dem Administrator mitgeteilt.</p>
Häufigkeit des Ereignisses	Ungefähr 20 Mal pro Stunde an einer Messe.
Hinweise	Der Administrator muss wissen, welche Parameter er dem CHOCOFORMER senden darf. Die Applikation prüft nicht, ob die Daten valide sind oder den Druckvorgang stören könnten.

UC06.1: TEXT/GRUSS MIT VORGEGEBENER SCHRIFTART ERSTELLEN

Primärer Akteur	Benutzer
Beschreibung	Der Benutzer kann mit Hilfe einer Tastatur einen Text/Gruss eingeben. Es besteht die Möglichkeit das Geschriebene zu bearbeiten.
Vorbedingungen	<ol style="list-style-type: none"> 1. Der Benutzer muss ein Einlegebauteil ausgewählt haben. 2. Der Benutzer muss die Einstellung «Mit Tastatur schreiben» gewählt haben.
Nachbedingungen	Die Applikation zeigt den geschriebenen Text/Gruss an.
Ablauf	<ol style="list-style-type: none"> 1. Der Benutzer wählt eine Schriftart aus. 2. Der Benutzer wählt die Schokolade, mit der der Text gedruckt werden soll, aus. 3. Der Benutzer schreibt seinen gewünschten Text mit der Tastatur. 4. Der Benutzer kann seinen Text mit den gegebenen Optionen verändern
Ausnahmen	Der Benutzer ist nicht in der Lage seinen Text ausserhalb des Bereichs für das entsprechende Einlegebauteil zu schreiben.
Häufigkeit des Ereignisses	Ungefähr 20 Mal pro Stunde an einer Messe.
Hinweise	Stürzt das Gerät, auf dem die Applikation läuft, während dieses Vorgangs ab, werden die Änderungen des Benutzers nicht gespeichert.

UC06.2: TEXT/GRUSS FREIHAND ERSTELLEN

Primärer Akteur	Benutzer
Beschreibung	Der Benutzer kann einen Text/Gruss über das Touchpad (mit einem Finger oder Stift) eingeben. Es besteht die Möglichkeit das Geschriebene zu bearbeiten.
Vorbedingungen	<ol style="list-style-type: none"> 1. Der Benutzer muss ein Einlegebauteil ausgewählt haben. 2. Der Benutzer muss die Einstellung «Freihand schreiben» gewählt haben.
Nachbedingungen	Die Applikation zeigt den geschriebenen Text/Gruss an.
Ablauf	<ol style="list-style-type: none"> 1. Der Benutzer wählt die Schokolade, mit der der Text gedruckt werden soll, aus. 2. Der Benutzer schreibt seinen gewünschten Text mit dem Finger/einem Stift. 3. Der Benutzer kann seinen Text mit den gegebenen Optionen verändern
Ausnahmen	Der Benutzer ist nicht in der Lage seinen Text ausserhalb des Bereichs für das entsprechende Einlegebauteil zu schreiben.
Häufigkeit des Ereignisses	Ungefähr 20 Mal pro Stunde an einer Messe.
Hinweise	Stürzt das Gerät, auf dem die Applikation läuft, während dieses Vorgangs ab, werden die Änderungen des Benutzers nicht gespeichert.

3.4 WEITERE MÖGLICHE FUNKTIONEN

- Fehlerbehandlung bei Schrittverlust des Druckers
- Anzeige, wieviel Schokolade noch ungefähr vorhanden ist
- Wenn eine Slicer-Konfiguration im Administrator Front-End eingelesen wird, besteht die Möglichkeit, die Konfiguration im Slicer zu überprüfen.

3.5 SYSTEM-SEQUENZDIAGRAMME

Im nachfolgenden Sequenzdiagramm wird erklärt, wie der typische Ablauf eines Drucks an einem Messestand aussehen kann. Der Ablauf beginnt beim Benutzer, welcher auf dem Benutzer Front-End das Sujet vorbereitet und zum Druck sendet. Anschliessend ist Back-End dafür verantwortlich, die Eingaben des Benutzers in Befehle für den Drucker umzuwandeln. Die folgenden Schritte sind ein Zusammenspiel von Administrator, Administrator Front-End und dem CHOCOFORMER. Der Administrator sorgt dafür, dass der Druck korrekt abläuft und erhält dafür die benötigten Informationen durch das Administrator Front-End sowie dem CHOCOFORMER selbst.

In diesem Ablauf sind unter anderem die Schritte nicht enthalten, welche notwendig sind, um den Drucker an einem Messestand einzurichten. Die Einrichtung erfolgt durch den Administrator am CHOCOFORMER und im Administrator Front-End. Dabei werden für den Benutzer verfügbare Optionen aktiviert bzw. deaktiviert. Diese Auswahl erfolgt nicht in einer vordefinierten Reihenfolge.

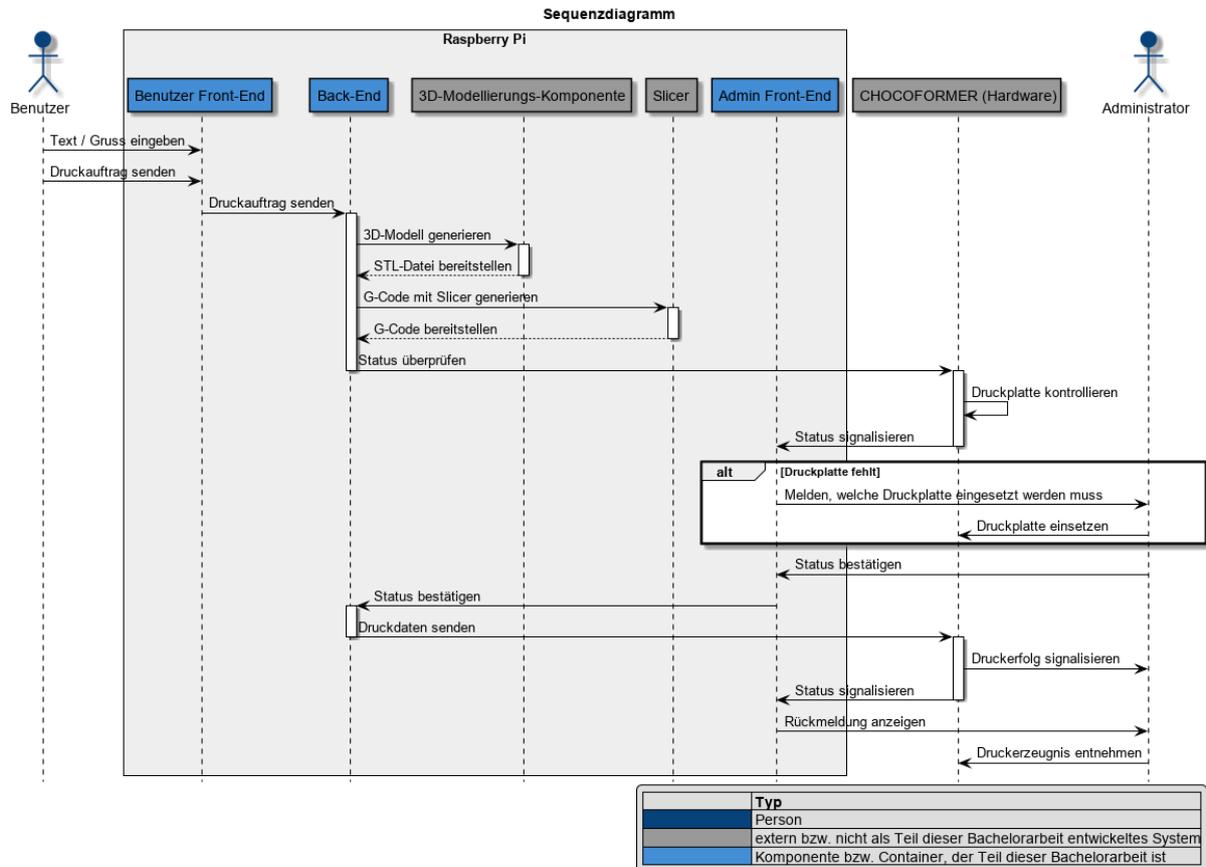


Abbildung 2 Ablauf und Interaktionen eines Druckauftrags

3.6 NICHT-FUNKTIONALE ANFORDERUNGEN

Anforderungen werden nach Priorität umgesetzt. Wobei «Muss» die höchste Priorität hat und «Kann» die geringste.

3.6.1 FUNKTIONALITÄT

Sicherheit

Nur eine autorisierte Person ist in der Lage den Administrator Bereich aufzurufen.

Soll

3.6.2 ZUVERLÄSSIGKEIT

Fehlertoleranz

Der Drucker druckt weiter, selbst wenn das Front-End unerwartet vom CHOCOFORMER getrennt wird.

Muss

Fehlerhafte Eingaben führen nicht zum Absturz der Anwendung.

Muss

Wiederherstellbarkeit

Administrator-Einstellungen bleiben erhalten bei Absturz oder Neustart der Anwendung.	Soll
Benutzer-Einstellungen (geschriebene Texte, Modelle, etc.) werden nicht serverseitig gespeichert.	Kann

3.6.3 BENUTZBARKEIT

Verständlichkeit

Die Beschriftung der Elemente in der Applikation erfolgt in Deutsch.	Muss
Es besteht die Möglichkeit, die Einstellung für die Beschriftung der Elemente in der Applikation auf Englisch zu wechseln.	Soll
Fehlermeldungen werden für den Benutzer verständlich, ohne Fachbegriffe, formuliert und farblich hervorgehoben in der Benutzeransicht angezeigt.	Soll

Erlernbarkeit / Bedienbarkeit

Der Administrator bedient das Administrator-Front-End nach einer 20-minütigen Einweisung.	Muss
Der Benutzer benötigt keine Hilfe vom Administrator, um die Applikation in der Benutzeransicht zu bedienen.	Soll

3.6.4 EFFIZIENZ

Zeitverhalten

Das Speichern und Abrufen von Daten sowie Antwortzeiten auf Benutzerinteraktionen sollten sowohl für den Benutzer als auch für den Administrator innerhalb von maximal 10 Sekunden erfolgen.	Soll
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------

3.6.5 ÄNDERBARKEIT

Analysierbarkeit

Das serverseitige Logging wird so aufgebaut, dass der aktuelle Status des zu bearbeitenden Drucks der entsprechenden Log-Datei entnommen werden kann.	Soll
Fehlermeldungen werden so geloggt, dass sie einem eindeutigen Ort im Code zugeordnet werden können.	Muss

Modifizierbarkeit

Alle Attribute und public-Methoden im Back-End sind nach den offiziellen C# Guidelines dokumentiert.	Muss
Alle Funktionen im Front-End sind nach den offiziellen JavaScript Coding Conventions implementiert.	Muss

Stabilität

Es wird ein stabiles Netzwerk, wie Ethernet, für die Kommunikation zwischen CHOCOFORMER, Back-End und Front-End an Messen verwendet.	Muss
--------------------------------------------------------------------------------------------------------------------------------------	------

Prüfbarkeit

Alle Änderungen am Code sind bei jedem Pull Request vom anderen Mitglied des Development Teams manuell sowie über automatisierte Tests kontrolliert worden.	Muss
-------------------------------------------------------------------------------------------------------------------------------------------------------------	------

3.6.6 ÜBERTRAGBARKEIT

Anpassbarkeit

Die Applikation ist in der Lage, mit dem Duet-Board zu kommunizieren, solange keine inkompatiblen API-Änderungen an der Firmware gemäss Semantic Versioning [5] gemacht werden.	Muss
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------

Installierbarkeit

Die benötigten Dateien für den Raspberry Pi, mit Betriebssystem DuetPi, lassen sich mit Hilfe eines einzelnen Scripts installieren.	Soll
Die Applikation lässt sich über eine IP-Adresse aufrufen, nachdem das Endgerät mit dem Raspberry Pi verbunden wurde.	Soll

Austauschbarkeit

Die Applikation ist Cross-Browser kompatibel: <ul style="list-style-type: none"> • Firefox 74.0, Chrome 80.0 • Edge 80.0, Apple Safari 13.0 	Muss Soll
Die Applikation kann auf unterschiedlichen Endgeräten verwendet werden <ul style="list-style-type: none"> • Tablet • PC • Smartphone 	Muss Muss Kann

4 DESIGN

4.1 ARCHITEKTUR

Das nachfolgende Kontextdiagramm zeigt die Ausgangslage dieser Bachelorarbeit. Wie daraus zu entnehmen ist, bestand die Lösung aus verschiedenen, unabhängigen Systemen, welche jeweils manuell von Administrator bedient werden mussten. Daher lässt sich das System nicht von unerfahrenen Benutzern bedienen.

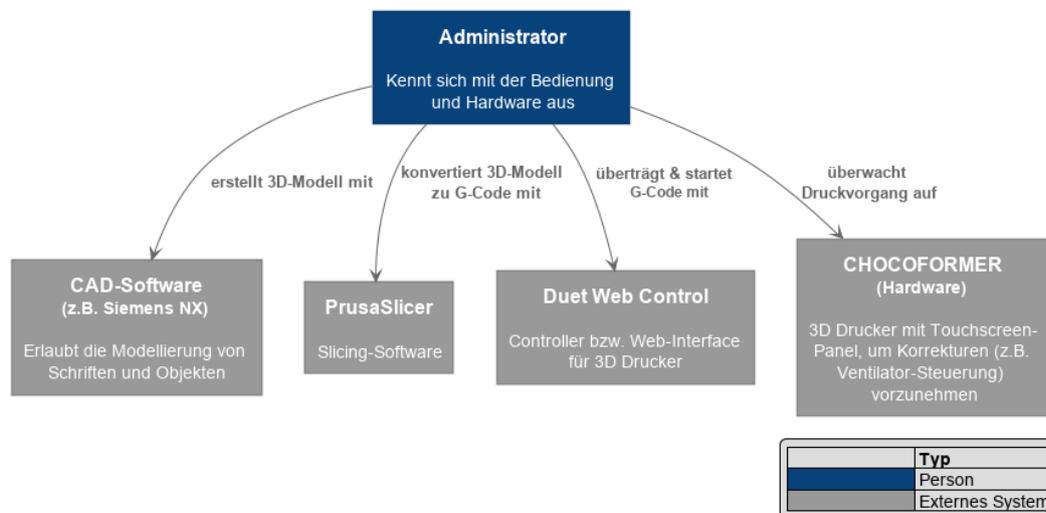


Abbildung 3 C4 Kontext-Diagramm für den bisherigen Zustand des CHOCOFORMERs (vorher)

Die Lösung dieser Bachelorarbeit soll die wichtigsten Komponenten zusammenfassen, so dass das Gesamtsystem auch von ungeschulten Benutzern bedient werden kann. Die CAD-Software bleibt als optionales externes System erhalten, so dass daraus vorgefertigte Sujets für den Benutzer bereitgestellt werden können. Die Funktionalität, fertige Sujets durch den Administrator drucken zu lassen, konnte in dieser Bachelorarbeit zwar wegen mangelnder Zeit nicht implementiert werden, jedoch ermöglicht die Architektur eine einfache Anpassung des Admin-Front-Ends, um solche Drucke zu ermöglichen.

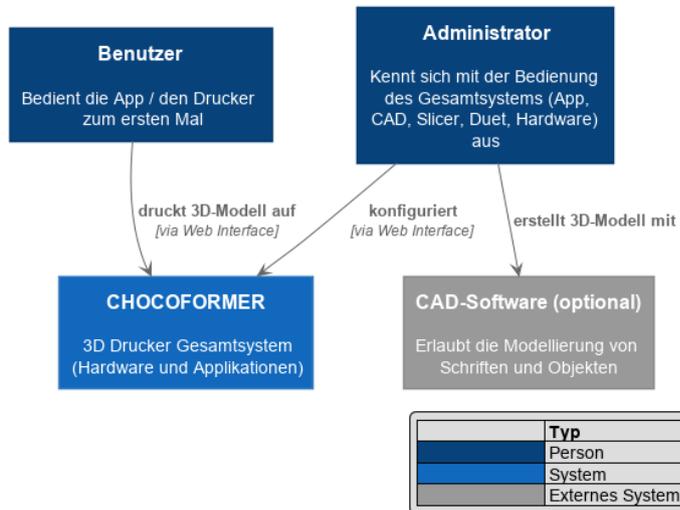


Abbildung 4 C4 Kontext-Diagramm für den CHOCOFORMER mit der CHOCOFORMER App (nachher)

Das neue CHOCOFORMER-System soll dabei auf einem Raspberry Pi gemeinsam mit den ehemals externen Applikationen laufen. Da die Lösung von zwei Benutzergruppen mit unterschiedlichen Bedürfnissen bedient wird, werden auch zwei separate Front-End-Applikationen entwickelt, welche dann jeweils mit dem Back-End-Service kommunizieren. Dieser übernimmt die Ansteuerung des Slicers und des Duet-Boards.

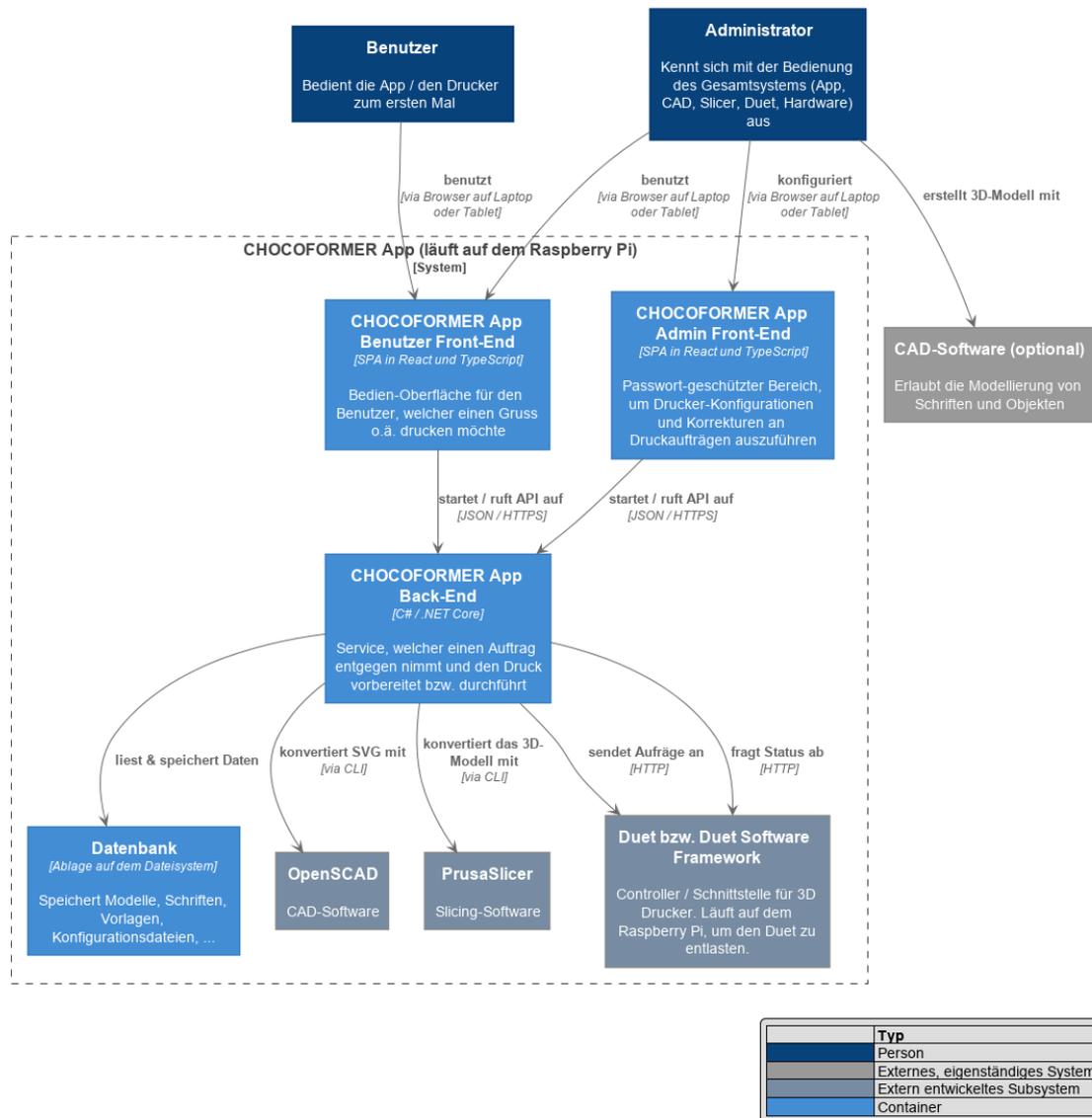


Abbildung 5 C4 Container-Diagramm für die CHOCOFORMER App

Das Back-End wiederum ist in mehrere Schichten aufgeteilt, welche dem Komponenten-Diagramm entnommen werden können. Die Web-Service-Komponenten bilden die oberste Schicht und damit den Einstiegspunkt für das jeweilige Front-End. In der darunterliegenden Schicht befindet sich die Applikationslogik, welche den Druck-Prozess von der Erstellung des Modells bis zur Auslösung des G-Codes auf dem Duet-Board übernimmt. Schlussendlich steht auch ein Data Access Layer bereit, durch welchen der Zugriff auf gespeicherte Daten vereinheitlicht wird.

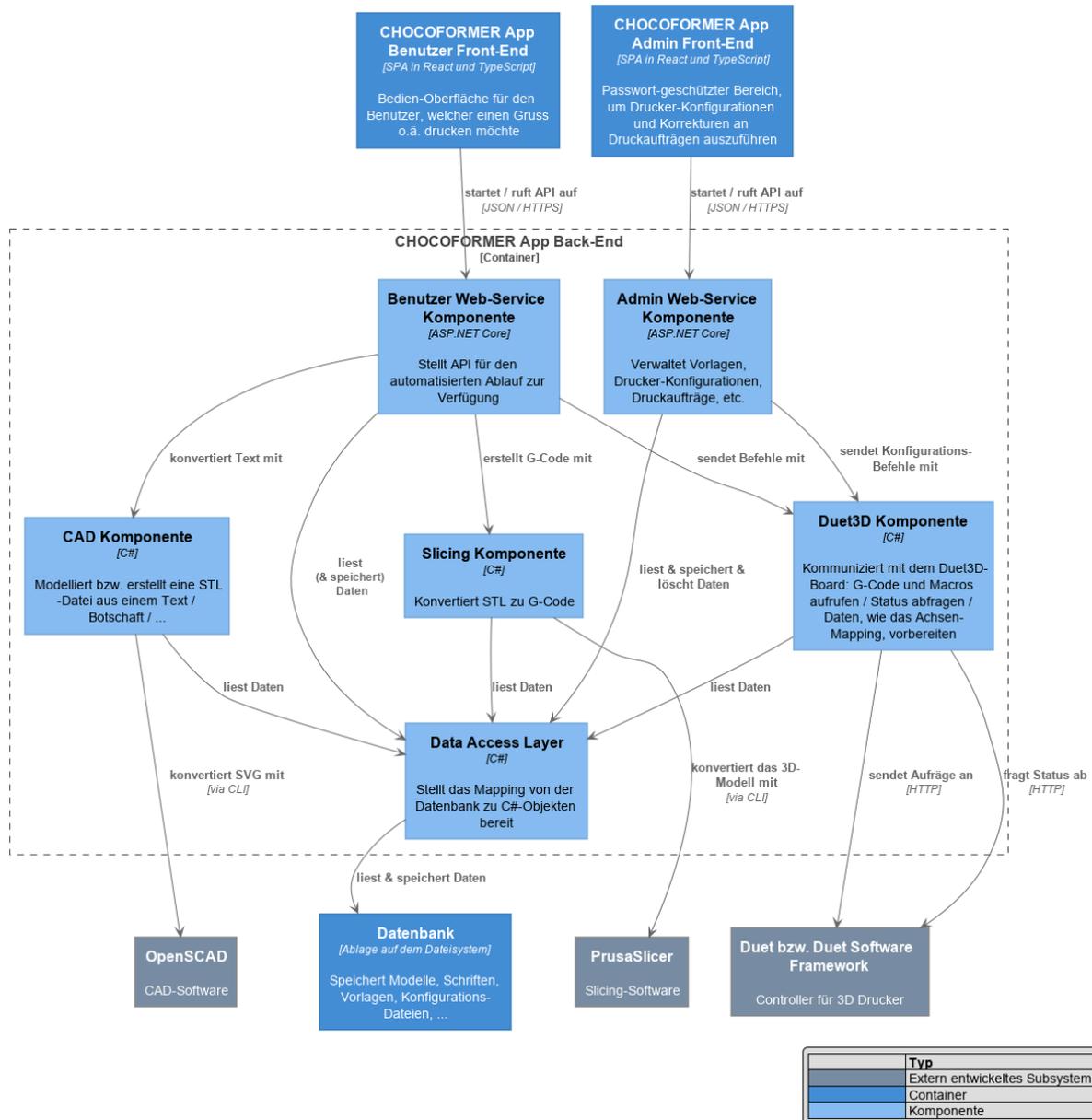


Abbildung 6 C4 Komponenten-Diagramm für das CHOCOFORMER App Back-End

4.1.1 VERANTWORTLICHKEITEN UND INTERAKTIONEN

Benutzer Web-Service-Komponente

- Stellt die API für das Benutzer-Front-End via ASP.NET Core zur Verfügung.
- Das Front-End sendet dieser Komponente das Sujet, welches gedruckt werden soll. Dazu gehören entweder ein Text, inklusive Schriftart, oder eine Vektorgrafik, welche vom Benutzer Freihand gezeichnet wurde.
- Aus dieser Komponente wird der gesamte Druckauftrag angestoßen, sobald der Benutzer sein Sujet zum Druck absendet. Dazu werden die benötigten Komponenten in folgender Reihenfolge aufgerufen: CAD Komponente, Slicing Komponente, Duet3D Komponente.

Admin Web-Service-Komponente

- Stellt die API für das Administrations-Front-End via ASP.NET Core zur Verfügung.
- Informationen über manuelle Arbeitsschritte (zum Beispiel Druckplatte einsetzen) bzw. Status-Details, welche von Duet gemeldet werden, werden an das Front-End weitergeleitet, so dass der Administrator während dem laufenden Betrieb des Druckers den Überblick über den Status behalten kann.
- Folgende Konfigurationen könnten aus dem zugehörigen GUI festgelegt werden: Slicer-Informationen (zum Beispiel Extruder-Einstellungen), Feature Toggles (Aktivierung einzelner Funktionalitäten, wie zum Beispiel welche Druckplatten verfügbar sind, mit welcher Schokolade gedruckt wird, wie viel Text gedruckt werden darf, etc.), Macros anpassen und hinzufügen.

CAD Komponente

- Erstellt die 3D-Modelle in Form einer STL-Datei aus den gegebenen Daten.
- Zu den Input-Daten gehören Text, wie zum Beispiel der Name des Benutzers oder eine Gruss-Botschaft, und eine Schriftart. Weiterhin können auch Vektor-Grafiken zu einem 3D-Modell konvertiert werden.
- Der Output ist ein 3D-Modell, wobei dieses auch zwei-dimensional sein kann, so dass der Drucker nur eine einzelne Schicht drucken muss. Die Output-Datei wird anschliessend via Service-Komponente für den nächsten Arbeitsschritt weitergereicht.
- Diese Komponente ist so modelliert, dass die darunterliegende CAD-Software jederzeit ausgetauscht werden kann. Die einzige Voraussetzung dafür ist, dass die neue CAD-Software ebenfalls SVG-Dateien und Text über das CLI entgegennehmen kann und ein 3D-Modell im STL-Format erzeugt.

Slicing Komponente

- Erstellt die Befehle, welche benötigt werden, um den Druckkopf zu steuern.
- Die Input-Datei im STL-Format wird in die benötigten Schichten unterteilt und daraus werden die Bewegungsabläufe in Form von G-Code generiert.
- Um das Slicing durchzuführen wird ein bereits bestehender Slicer via Command Line Interface angebunden. Durch die Auslagerung dieser Funktionalität, wird die korrekte Generierung von G-Code sichergestellt.
- Das CLI ermöglicht, den Slicer jederzeit und ausschliesslich durch Änderung von Konfigurationsdateien anzupassen. Sofern eine andere Slicer-Software ebenfalls über vergleichbare Konfigurationsdateien gesteuert wird, kann diese ohne Anpassung der Slicing Komponente ausgetauscht werden.

Duet3D Komponente

- Stellt die bidirektionale Kommunikation mit dem Duet Controller sicher.
- Druckbefehle sollen als G-Codes via HTTP-Requests an den Duet gesendet werden, um so Druckkopf-Bewegungen, Macros und weitere Steuerungsbefehle auszulösen.
- Status-Meldungen vom Duet Controller Board werden via WebSockets empfangen und nach Informations-Typ (zum Beispiel Temperatur-Meldungen, Fehler, etc.) gefiltert.
- Gegebenenfalls müssen die Daten zuvor für den einzelnen Drucker angepasst werden. Da der Drucker mit zwei Druckköpfen ausgerüstet sein kann, muss vor dem Senden der Druckdaten ein Mapping vom Druckkopf auf die passende Achse im G-Code durchgeführt werden.
- Da die gesamte Kommunikation mit dem Duet Controller über diese Komponente läuft, müsste nur diese Komponente neu implementiert werden, um Befehle an einen anderen Controller zu senden.

Data Access Layer

- Bietet einen vereinfachten Zugriff auf die darunterliegende Datenverwaltung.
- Sämtliche Komponenten greifen über diesen Layer auf die benötigten Daten zu. Damit wird eine Entkopplung von der darunterliegenden Datenhaltungs-Technologie erzielt.
- Der Zugriff auf die Daten auf dem Dateisystem erfolgt via der von der Laufzeitumgebung zur Verfügung gestellten API.

OpenSCAD

- Bietet mit einer eigenen Domain-Specific Language eine Möglichkeit an, die CAD-Software über die Kommandozeile zu bedienen.
- Als Ergebnis eines Aufrufs wird eine STL-Datei erstellt, welche das zu druckende 3D-Modell enthält.

PrusaSlicer

- Führt das Slicing eines 3D-Modells in einzelne Schichten und vom Drucker ausführbare Maschinen-Befehle durch.
- Wurde bereits vor dieser Arbeit vom IWK als Slicing-Software verwendet. Dadurch ist bereits Erfahrung betreffend Verhalten des Slicers und der optimalen Konfiguration vorhanden.
- Über die Kommandozeile können 3D-Objekte im STL-Format importiert und rudimentär angepasst werden. Konfigurationen, zum Beispiel für Material und Grössenangaben, können in einer INI-Datei mitgegeben werden.
- Erzeugt Maschinen-lesbare Befehle im G-Code-Format. Zusätzliche Informationen, wie die geschätzte Druckzeit, werden als Kommentare in die gleiche Datei geschrieben.

Duet Software Framework

- Stellt die Steuerungssoftware auf dem DuetPi-Betriebssystem bereit. Dazu gehören unter anderem das Duet Web Control (DWC), dessen HTTP-Endpoints im DuetWebServer und das Mapping der SD-Karte im Duet-Board auf das Linux-Dateisystem.
- Die HTTP-Endpoints des DuetWebServers können verwendet werden, um G-Code an den Drucker zu senden und Status-Informationen abzurufen.

4.2 KONZEPTE

4.2.1 DRUCK-AUFTRÄGE

Die Daten, welche für einen Druck anfallen, sind hauptsächlich temporäre Dateien, welche von den einzelnen externen Applikationen verarbeitet werden. Während des Prozesses, den fertigen G-Code zu generieren, müssen verschiedene Dateien erstellt werden: svg-, scad-, stl- und gcode-Dateien. Diese werden von den verschiedenen Komponenten verarbeitet und erstellt. Alle Dateien für einen Job werden in einen für diesen Job bestimmten Ordner gespeichert. Der Ordnername lautet job-{id}. Alle ähnlichen Dateien von den verschiedenen Jobs können so den gleichen Namen tragen.

Je nachdem, wie viele Drucke stattfinden, können sich viele Daten ansammeln und über die Zeit die SD-Karte füllen. Um zu verhindern, dass der Speicher überfüllt wird, werden die Dateien wieder gelöscht, sobald der Druck nicht mehr benötigt wird. Dies ist der Fall, sobald der nächste Druck aus der Queue ausgelesen wird. Das Maintainer-Team möchte die Möglichkeit haben, einen Druck mehrfach auszuführen, falls das Sujet mehr als einmal gedruckt werden soll. Zudem kann es vorkommen, dass während dem Druck Probleme auftreten, der Chocoformer druckt nicht sauber oder die Schokoladenfüllung ist leer. In diesen Fällen ist es von Vorteil, wenn der Druck wiederholt werden kann. Daher soll der Administrator bewusst entscheiden können, wann der Druck nicht mehr benötigt wird.

Dateien, welche zu einem einzelnen Druckauftrag gehören, werden in einem eindeutig identifizierten Ordner abgelegt. Innerhalb eines solchen Ordners können die Dateinamen konstant gehalten werden. Dem Ordner wird mittels einer ID ein eindeutiger Name zugewiesen. Die ID wird vom Back-End für jeden Druck erstellt, von 1 aufwärts zählend. Zunächst war geplant, dass die ID aus einem Zeitstempel,

der auf Sekunde genau angegeben ist, bestehen soll. Da die Applikation nur von einem Benutzer auf einmal verwenden kann, ist ein dichtes Aufkommen von Druckjobs nicht gegeben. Allerdings hat sich herausgestellt, dass sich dieses Vorgehen in doppelter Hinsicht nicht eignet. Zum einen wird die ID dem Benutzer präsentiert, damit er weiss, wann sein Druck abgeschlossen wurde, zum anderen eignet sich diese ID nicht für automatisierte Tests. Da die Tests parallel ablaufen, können pro Sekunde mehrere Tests ausgeführt werden. Dadurch überschreiben sich die Tests gegenseitig ihre Testdateien und können so nicht erfolgreich durchgeführt werden.

4.2.2 LOCKING-MECHANISMUS

Ein Locking Mechanismus wird benötigt, wenn verschiedene Prozesse auf die gleichen Daten schreiben können. Es wird damit verhindert, dass Daten verloren gehen. Mit einem Locking-Mechanismus wird dafür gesorgt, dass andere Prozesse eine Datei nicht verändern können, wenn diese bereits von einem anderen Prozess verwendet wird, das bedeutet, die Datei wurde gelockt.

Die Daten der Jobs für die Queue werden in einer JSON-Datei auf dem Dateisystem abgelegt. Da das Benutzer- und das Administrator Front-End mit Schreibbefehlen auf diese Datei zugreifen, muss die Datei entsprechend gelockt werden, um Datenverlust zu vermeiden.

Zuerst wurde die EventWaitHandle Klasse [6] verwendet. Diese Klasse bietet einen Event Handler, der mehreren Prozessen erlaubt miteinander zu kommunizieren. Jedoch wird diese Klasse nur auf Windows unterstützt. Da die Applikation später auf dem Raspberry Pi laufen muss, der mit Linux betrieben wird, musste ein anderes Framework gefunden werden.

Da ASP.NET Core keinen vorgefertigten Locking-Mechanismus unter Linux bietet, musste ein eigenes Locking-Verfahren implementiert werden. Jedes Mal, wenn ein Prozess einen Schreibzugriff auf die JSON-Datei durchführt, muss dieser eine Lock-Datei setzen. Solange diese Lock-Datei vorhanden ist, darf kein anderer Prozess auf die JSON-Datei schreiben. Sobald der Schreibvorgang abgeschlossen wurde, entfernt der Prozess die Lock-Datei und die anderen Prozesse können wieder Schreibzugriffe auf die JSON-Datei ausführen.

Dieser Locking-Mechanismus ist nicht performant, da viele Schreibzugriffe auf die SD-Karte des Raspberry Pi vorgenommen werden. Diese Schreibzugriffe sind jedoch vertretbar, da sie etwa zwei- bis viermal in der Minute ausgeführt werden und somit keine Performance-Einbussen entstehen. Auch wenn es nicht sehr wahrscheinlich ist, dass die beiden Prozesse des Benutzer- und des Administrator Front-Ends gleichzeitig auf die JSON-Datei schreiben, ist es doch wichtig, möglichen Datenverlust zu verhindern.

4.3 UI-DESIGN

Die Entwürfe für das Front-End befinden sich im Anhang 16.1 WIREFRAMES.

4.3.1 BENUTZER FRONT-END

Das Benutzer Front-End ist eine Single Page Applikation, das bedeutet der Browser lädt dynamische die neuen Elemente nach, statt eine ganze neue Seite zu rendern. Die Grundlage der Seite besteht aus der Navigation. Je nachdem, welcher Punkt in der Navigation gewählt wird, wird das entsprechende Element in die Seite geladen. Der Benutzer startet mit einem Begrüssungsbildschirm. In der Navigation kann er entscheiden, was er als Nächstes tun möchte. Um zu verhindern, dass der Benutzer einen Gruss schreiben kann, ohne ein Einlegebauteil gewählt zu haben, wird der entsprechende Navigationspunkt erst verfügbar, wenn ein Einlegebauteil gewählt wurde. Ebenso kann der Benutzer den Druck erst starten, wenn er sein Sujet erstellt hat.

Der Benutzer kann aus einer Auswahl von Einlegebauteilen jenes wählen, welches ihm am meisten zusagt. Um sein Sujet zu schreiben, werden dem Benutzer verschiedene Tools angeboten. So kann er auf dem Touchscreen sein Sujet malen oder er kann die Tastatur verwenden, um etwas zu schreiben.

4.3.2 ADMINISTRATOR-FRONT-END

Das Administrator-Front-End ist in zwei Bereiche unterteilt. Einerseits gibt es eine Übersicht, in welcher die Druckaufträge aufgelistet sind. Diese Aufträge können aus dem GUI freigegeben werden. Zudem ist für den Administrator ersichtlich, welche Druckplatte eingesetzt werden muss.

Der zweite Bereich dient dazu, die Konfigurationen vorzunehmen. Hier können also die entsprechenden Dateien hochgeladen bzw. gelöscht werden. Ausserdem werden in dieser Ansicht Features je nach Bedarf oder Besuchermasse aktiviert oder deaktiviert.

5 PROGRAMMIERSPRACHEN, FRAMEWORKS UND TOOLS

Für die Umsetzung der Applikation folgt ein Vergleich einiger geeigneter Programmiersprachen, Frameworks und Tools, die dem Development Team bekannt sind.

5.1 FRONT-END / GUI

Variante	Vorteile	Nachteile
Single Page Application mit React und TypeScript	<ul style="list-style-type: none"> • Läuft in den gängigen, aktuellen Browsern und somit auf allen wichtigen Betriebssystemen • Unterstützt Multi-Touch • Zeichnungsfläche ist einfach umzusetzen • Viel Erfahrung im Team • Eigentlich alles möglich umzusetzen 	<ul style="list-style-type: none"> • Abhängigkeiten können schnell ändern, was die Wartung erschweren kann. Es darf also nur auf wenige, bekannte Bibliotheken gesetzt werden.
WPF Windows Applikation	<ul style="list-style-type: none"> • Unterstützt Multi-Touch 	<ul style="list-style-type: none"> • Nur unter Windows verfügbar • (Design-) Richtlinien können sich immer wieder ändern. • Es können Lizenz-Kosten anfallen.
Android Smartphone / Tablet App	<ul style="list-style-type: none"> • Unterstützt Multi-Touch 	<ul style="list-style-type: none"> • Nur unter Android verfügbar
Xamarin Mobile App (für Android & iOS)	<ul style="list-style-type: none"> • Gleicher Code wird für Android und iOS verwendet. 	<ul style="list-style-type: none"> • Eine iOS-App benötigt dennoch einen Computer mit macOS. • Es können Lizenz-Kosten anfallen.

Die Applikation wird als Single Page Web Applikation umgesetzt. Das Development Team hat in diesem Bereich am meisten Erfahrung und es gibt viele Möglichkeiten, eine attraktive Applikation zu realisieren. Es wird darauf geachtet, nicht zu viele und nur stark etablierte Bibliotheken zu verwenden, um die kritische Anforderung an die Robustheit nicht zu gefährden und spätere Wartungsarbeiten möglichst zu vereinfachen. Eine Web Applikation ist nicht abhängig von der Plattform (Windows, macOS, iOS, Android), dem Endgerät (Tablet, PC, Smartphone) oder dem Browser (Chrome, Firefox, Safari, Edge) auf dem sie läuft. Somit kann jedes beliebige Gerät für die Ansteuerung des CHOCOFORMERs verwendet werden.

Die Umsetzung als Web App entspricht auch den Vorstellungen des Stellvertretenden Product Owners und der Maintainer. Zu einem späteren Zeitpunkt könnte die Applikation somit auch in eine Webseite integriert werden. Aus diesem Grund werden die React-Komponenten so implementiert, dass diese einfach in zukünftigen Web-Applikationen eingebaut werden können.

5.2 BACK-END

Variante	Vorteile	Nachteile
C# mit xUnit.net, Entity Framework Core, ASP.NET Core	<ul style="list-style-type: none"> • Einfach wartbare Programmiersprache • Genügend Erfahrung im Development Team • Know-How im Maintainer-Team • Das Duet Software Framework ist auch mit C# und ASP.NET Core implementiert und läuft bereits auf DuetPi. • Erfahrungen im Maintainer Team vorhanden 	<ul style="list-style-type: none"> • Keine Erfahrung bezüglich Entwicklung mit .NET Core unter Linux (Anbindung von Bibliotheken, ...) • Es können Lizenz-Kosten anfallen.
Java mit JUnit, Spring (Web Services, Data)	<ul style="list-style-type: none"> • Portabilität auf verschiedene Betriebssysteme ist gewährleistet. • Einfach wartbare Programmiersprache • Genügend Erfahrung im Development Team 	<ul style="list-style-type: none"> • Wenig Erfahrung im Development Team mit den entsprechenden Spring Frameworks
C++	<ul style="list-style-type: none"> • Geeignet für Hardware-nahe Entwicklung • Erfahrungen im Maintainer Team vorhanden 	<ul style="list-style-type: none"> • Einarbeitung in die Programmiersprache ist mit hohem Aufwand verbunden • Keine Erfahrung mit geeigneten Frameworks

Das Back-End wird mit .NET Core bzw. C# implementiert. Das Entwicklerteam hat sich für diese Variante entschieden, da hier die meiste Erfahrung vorhanden ist. Einerseits kommen beide Entwickler gut mit dieser Programmierumgebung zurecht, andererseits gibt es sowohl am ILT als auch am IFS Ansprechpersonen, welche C#-Erfahrung mitbringen und dadurch Wartungsarbeiten an der CHOCOFORMER App durchführen oder unterstützen können.

Ein weiterer wichtiger Punkt, welcher für C# spricht, ist das Duet Software Framework selbst. Dieser Bestandteil der Duet-Plattform ist in C# implementiert und läuft bereits auf dem DuetPi-Betriebssystem. Somit existiert eine Art Referenz-Implementation, an welcher sich die CHOCOFORMER App bezüglich Framework-Kompatibilität und Konfiguration für den Raspberry Pi orientieren könnte.

Von der Verwendung von C++ wurde abgeraten. Da allgemein im Entwickler- und im Wartungsteam wenig Erfahrung vorhanden ist, wäre die Einarbeitungszeit in diese Sprache zu hoch. Andererseits wäre C++ sehr gut für Hardware-nahe Entwicklung geeignet. Wie sich jedoch bereits früh im Projektverlauf herausstellte, hätte diese Eigenschaft keinen Vorteil erbracht. Die Duet-Plattform setzt auf Protokolle, welche in anderen Programmiersprachen ebenfalls bestens unterstützt sind.

5.3 DATENBANK

Variante	Vorteile	Nachteile
PostgreSQL	<ul style="list-style-type: none"> Erfahrung im Development Team 	<ul style="list-style-type: none"> Zu schwergewichtig für die Anwendung
SQLite	<ul style="list-style-type: none"> Leichtgewichtige Datenbank 	
Dateisystem	<ul style="list-style-type: none"> Benötig wenig Speicherplatz und Arbeitsspeicher Ausreichend, da keine Mehrnutzerfähigkeit benötigt wird (nur eine Applikation greift darauf zu) 	<ul style="list-style-type: none"> Spätere Erweiterungen (z.B. als Web-Shop) erfordern grössere Anpassungen

In diesem Projekt müssen vor allem Dateien und boolesche Datenwerte gespeichert werden. Dateien umfassen sowohl Schriftarten und Konfigurationsdateien für den Slicer und das Duet-Board, als auch temporäre Dateien, welche für den Druckvorgang generiert werden müssen. Weiterhin muss abgespeichert werden, welche Features im Admin-GUI aktiviert wurden. Der Datenbestand dieser Applikation besteht als vornehmlich aus Dateien und Konfigurationen, welche als Key-Value-Pairs abgelegt werden können. Komplexe Abfragen des Datenmodells sind nicht notwendig.

Aus den oben genannten Gründen wird für die CHOCOFORMER App eine einfache Dateiablage als Datenbank verwendet. Es ist einfacher, Dateien direkt auf dem Dateisystem abzulegen, als diese in ein möglicherweise starres Modell in einer Datenbank einzupassen. Zudem wird ohne Datenbank weniger Arbeitsspeicher verwendet, welcher den Raspberry Pi zusätzlich belasten würde.

Da die CHOCOFORMER App in einem späteren Schritt um einen Web-Shop ergänzt werden könnte, sollte der Zugriff auf die Datenbank möglichst einfach austauschbar sein. Deshalb wird ein Data Access Layer eingebaut. Für den eigentlich einfachen Zugriff auf das Dateisystem kann diese Lösung möglicherweise ein Mehraufwand darstellen, da sämtliche Aufrufe durch eine weitere Komponente geleitet werden. Dafür ist sichergestellt, dass der Zugriff auf die Datenhaltungs-Lösung für die restlichen Komponenten transparent ist. Somit kann die Datenbank ausgetauscht bzw. das Entity Framework für die entsprechenden Zugriffe eingefügt werden, ohne dass dabei die bestehenden Komponenten hinsichtlich des Datenzugriffs angepasst werden müssen. Da die Aufteilung in einen solchen Data Access Layer den Empfehlungen von anerkannten Persönlichkeiten aus dem Software-Umfeld [8] [9] entspricht, wird es für zukünftige Entwickler an dieser Applikation auch einfacher, sich im Source Code zurechtzufinden und allfällige Migrationen durchzuführen.

5.4 SLICER

Folgende Slicer sind kompatibel mit dem CHOCOFORMER:

Tool	Vorteile	Nachteile
PrusaSlicer [8]	<ul style="list-style-type: none"> • Kann via Command Line von anderen Programmen verwendet werden. • Libslic3r kann unabhängig verwendet werden. • Team hat bereits Erfahrung (Empfehlung) • Open Source • Kostenlos • Verfügbar für Windows, Linux, Mac • Aktive Community 	
Ultimaker CURA [9]	<ul style="list-style-type: none"> • Verfügbar für Windows, Linux, Mac • CAD plugin (SolidWorks, Siemens NX, Autodesk Inventor, etc.) • Open Source • Kostenlos • Grosse Community 	<ul style="list-style-type: none"> • Keine Informationen zur Benutzung via Command Line
Pathio [10]	<ul style="list-style-type: none"> • Scripting mit GCode könnte zur Automatisierung verwendet werden • Verfügbar für Windows, Linux, Mac • Momentan kostenlos 	<ul style="list-style-type: none"> • Ist momentan nur als Beta-Version verfügbar • In Zukunft können (teilweise) Kosten anfallen
Craftbot CraftWare [11]	<ul style="list-style-type: none"> • Verfügbar für Windows, Linux, Mac, Chrome OS • Momentan Kostenlos 	<ul style="list-style-type: none"> • Keine Informationen zur Benutzung via Command Line • Beta-Phase
Simplify3D [12]	<ul style="list-style-type: none"> • Verfügbar für Windows, Linux, Mac • Viele Funktionen • Guter Support über Entwickler und Forum 	<ul style="list-style-type: none"> • Internet-Verbindung während der Verwendung notwendig • Kostenpflichtig (150\$)

Die Entwicklung der CHOCOFORMER App hat zwei kritische Faktoren betreffend Software und Tools, die verwendet werden sollen: Erstens sollte die Entwicklung möglichst wenig kosten und zweitens sollte die Lösung möglichst robust sein. Anhand dieser Kriterien wird der Slicer für die Applikation ausgewählt.

Simplify3D ist ein sehr guter und robuster Slicer, jedoch benötigt er eine Internetverbindung (dies kann an Messen ein Problem sein, an dem der CHOCOFORMER vor allem betrieben wird) und er ist kostenpflichtig. Daher kommt der Simplify3D nicht in Frage.

Pathio und CraftWare sind beide noch in der Beta-Phase. Beide können noch Bugs enthalten und könnten kostenpflichtig werden, sobald sie fertiggestellt werden. Daher kommen Pathio und CraftWare ebenfalls nicht in Frage.

Ultimaker CURA erfüllt viele der Bedürfnisse. Es ist Open Source, hat eine grosse Community, läuft auf allen wichtigen Betriebssystemen und bietet eine CAD Einbindung. Problematisch ist, dass es wahrscheinlich nicht über die Command Line verwendet werden kann.

Der PrusaSlicer ist ebenfalls Open Source, hat eine aktive Community und unterstützt alle wichtigen Betriebssysteme. Es hat jedoch keine CAD Einbindung, dafür kann man den PrusaSlicer via Command Line verwenden oder die Bibliothek direkt anbinden. Ausserdem ist diese Software vom Team empfohlen worden, da momentan auch damit gearbeitet wird, bei Fragen sind somit direkte Ansprechpersonen vorhanden.

Für diese Applikation wird anhand der oben genannten Gründe, der PrusaSlicer verwendet.

5.5 CAD

Für die Erstellung von 3D-Modellen musste eine CAD-Software gefunden werden, die SVG-Dateien einlesen und STL-Dateien ausgeben kann. Als besondere Herausforderung muss diese Software auch über die Kommandozeile ansprechbar sein und nicht über eine grafische Benutzeroberfläche.

Die meistens freien CAD-Applikationen sind, da es sich um ein sehr visuelles Anwendungsgebiet handelt, nur mit einer grafischen Oberfläche ausgestattet und sind daher nicht geeignet, um in die CHOCOFORMER App eingebettet zu werden.

Von den restlichen Applikationen war es schwierig, eine Lösung zu finden, welche mit den passenden Dateien umgehen konnte. Zum Beispiel bietet FreeCAD eine Python-Schnittstelle an, jedoch können keine SVG-Dateien importiert werden und der Export erzeugt bei einem Test unter DuetPi kein gültiges STL.

Die erste Applikation, die den Anforderungen genügte, war OpenSCAD [13]. Dabei handelt es sich um eine Anwendung, welche die 3D-Modelle durch eine Domain Specific Language (DSL) generieren kann. Mit dieser Sprache können eigene Objekte und Texte platziert werden, aber auch SVG- oder PNG-Bilder importiert werden. Durch verschiedene Funktionen können diese Elemente anschliessend skaliert, verschoben oder extrudiert werden.

OpenSCAD hat den Vorteil, dass eine Benutzeroberfläche bereitsteht, mit welcher die Modelle in einer Vorschau angezeigt werden können, aber gleichzeitig auch über die Kommandozeile als externer Prozess angestossen werden kann. Der Nachteil an dieser Variante ist, dass es durch die DSL schwieriger wird, fehlerhafte Eingaben zu finden. Dies ist zum Beispiel der Fall, wenn Grafiken importiert werden. Diese sind standardmässig im 2-dimensionalen Raum und können nicht sofortig mit 3D-Objekten kombiniert werden.

Eine weitere Herausforderung an OpenSCAD ist, dass für die armhf-CPU-Architektur des Raspberry Pi kein Softwarepaket bereitsteht. Glücklicherweise liess sich ein vorbereitetes Paket für eine aktuelle Version von OpenSCAD sowie eine Anleitung, um OpenSCAD für den Raspberry Pi zu kompilieren, auf GitHub finden [14]. Um zu verhindern, dass diese Anleitung nicht mehr auf GitHub zu finden ist, kann sie auch auf dem CHOCOFORMER-GitLab [15] gefunden werden.

5.6 DUET 2 VS. DUET 3

Für die Ansteuerung des CHOCOFORMERs wird Duet3D [16] verwendet. Duet3D bietet Hardware für die Ansteuerung von 3D-Druckern, welche mit der Open Source Firmware RepRapFirmware [17] betrieben wird.

Das IWK hat sich beim Bau des ersten CHOCOFORMERs für die Verwendung von Duet3D entschieden. Daher wird auch in dieser Arbeit mit Duet3D gearbeitet. Das Know-How ist bereits vorhanden und der Unterhalt gewährleistet.

Board	Vorteile	Nachteile
Duet 2	<ul style="list-style-type: none"> • Momentan im Einsatz beim CHOCOFORMER 	<ul style="list-style-type: none"> • Kein WebSocket, sondern HTTP-Polling • Zu wenig Steckplätze für Motoren. Es werden Extension Board benötigt.
Duet 3	<ul style="list-style-type: none"> • Einfache Verbindung mit Raspberry Pi möglich • Eigenes Betriebssystem (DuetPi) für Raspberry Pi • Durch das Verwenden eines Raspberry Pi wird die CPU des Duet 3 entlastet, da der DWC und der DWS auf dem Raspberry Pi laufen. Die CPU des Duets ist nur noch für die Motorensteuerung zuständig. • Status-Informationen werden über WebSocket gesendet • Genügend Steckplätze für Motoren • Aktuellere Version, wird länger supported 	<ul style="list-style-type: none"> • Neue Boards müssen angeschafft werden, was teurer für bestehende CHOCOFORMER Drucker ist

Für diese Arbeit empfiehlt sich das Duet 3 Board. Die Schnittstelle zum Raspberry Pi ist schon vorhanden und es existiert ein eigenes Betriebssystem für den Raspberry Pi von Duet3D. Auf dem Raspberry

Pi soll ebenfalls die CHOCOFORMER App laufen. Zudem ist die Auswertung der Daten einfacher als mit dem Duet 2 Board, da es die Daten über das WebSocket-Protokoll sendet.

Die endgültige Entscheidung, welches Board verwendet wird, liegt nicht in den Händen des Development Teams. Der Product Owner trifft diese Entscheidung.

5.7 NETZWERK

Der Raspberry Pi 4B verfügt über zwei verschiedene Netzwerk-Schnittstellen: Einerseits steht eine RJ-45-Buchse bereit, um eine Verbindung über Ethernet aufzubauen. Andererseits ist auch ein WLAN-Modul integriert.

Im Rahmen dieser Arbeit wird Ethernet verwendet, da es grundsätzlich die stabilere Lösung ist. Zudem ist es einfacher, die Geräte innerhalb eines solchen Netzwerkes zu kontrollieren und macht damit die Lösung mindestens als ersten Schritt sicherer.

Ein weiterer Vorteil der Lösung mit dem Raspberry Pi ist, dass es einfach ist, in einem späteren Schritt die Applikation doch über Wi-Fi bereitzustellen. Die Schnittstelle bzw. der Web-Server sind unabhängig von der verwendeten Variante, so dass es sich bei der Umstellung prinzipiell nur um eine Konfigurationsänderung auf dem DuetPi-Betriebssystem handeln sollte.

6 IMPLEMENTATION UND ENTWICKLUNG

6.1 ERLÄUTERUNGEN WICHTIGER KONKRETER KLASSEN

Für die Implementation in C# wird grundsätzlich für jede Komponente ein eigenes Projekt erstellt. Eine Ausnahme stellen die Komponenten für die Applikationslogik dar. Da diese Komponenten vor allem externe Software aufrufen und dadurch einen eher kleinen Umfang haben, ist es sinnvoll, diese Komponenten in ein Projekt zusammenzufassen. Um eine Unterteilung der Komponenten aufrecht zu erhalten, werden diese in jeweils einem separaten Namensraum implementiert.

Die erste Komponente in der Applikationslogik ist in der Klasse `CadService` implementiert. Diese Klasse nimmt den Text des Benutzers und die SVG-Daten entgegen. Zu den SVG-Daten zählen sowohl die Zeichnung in einer der gewählten Farben als auch ein Rechteck, welches den Rand markiert. Dieses Rechteck ist notwendig, damit die Positionierung der Zeichnung in der Slicing-Software erhalten bleibt. Die entgegengenommenen Daten werden der von OpenSCAD eingesetzten Sprache hinterlegt. Damit wird anschliessend OpenSCAD gestartet und das 3D-Modell im STL-Format generiert.

In der Klasse `SlicerService` ist die nächste Komponente implementiert. Diese Klasse nimmt den Speicherort der STL-Datei sowie die Konfigurationsdatei für den gewählten Druckkopf entgegen. Damit wird dann PrusaSlicer gestartet. PrusaSlicer positioniert das Sujet absolut in seinem Koordinatensystem. Das dafür verwendete Rechteck wird nicht den Druck übernommen, da die Linie so dünn gewählt wurde, dass sie von PrusaSlicer verworfen wird. Das Ergebnis dieser Klasse wird in einer G-Code-Datei abgespeichert.

Die letzte wichtige Komponente bzw. Klasse in der Applikationslogik-Schicht nennt sich `Duet3DService`. Diese Klasse übernimmt die Kommunikation mit Duet, indem die entsprechenden HTTP-Endpoints aufgerufen werden. Einerseits kann eine beliebige G-Code-Zeichenkette mitgegeben werden, um damit Instruktionen an das Duet-Board zu senden. Andererseits können mit einer weiteren Methode die aktuellen Status- und Sensor-Informationen vom Duet-Board geladen werden.

In der Benutzer Web-Service Komponente werden die ersten zwei oben genannten Komponenten zusammengeführt. Die Klasse `WorkflowService` ruft dazu diese Klassen in der richtigen Reihenfolge auf. Da die Zeichnungen für jede Farbe einzeln behandelt werden, werden die Klassen für das CAD und den Slicer potenziell mehrfach aufgerufen. Abschliessend werden die daraus entstandenen G-Code-Dateien zusammengeführt.

Da der Druck durch den Administrator bestätigt werden muss, wird der letzte Schritt durch die `WorkflowService`-Klasse in der Admin Web-Service Komponente durchgeführt. Damit wird der G-Code der Duet3D-Komponente übergeben und damit der Druck gestartet.

6.2 KONFIGURATIONEN

In den Konfigurationsdateien befinden sich verschiedene Einstellungsmöglichkeiten, welche typischerweise nur selten geändert werden müssen. Dazu zählen zum Beispiel die Pfade zu den externen Applikationen und deren Parameter, aber auch die URLs zu den Schnittstellen des DuetWebServers.

Während der Entwicklung und des Testens werden andere Konfigurationen bzw. Pfade benötigt, da die Entwicklung unter Windows stattfindet und nicht auf dem DuetPi selbst. Um die Änderung dieser beiden Mengen an Konfigurationen zu vereinfachen, wurden zwei separate Konfigurationsdateien erstellt. Eine Datei ist für die Entwicklung unter Windows gedacht, wobei die Pfade wo möglich relativ angegeben werden, um eine Anpassung pro Entwicklerarbeitsplatz zu vermeiden. Die zweite Datei dient der schlussendlichen Konfiguration auf dem DuetPi.

Ebenso sind die Integrationstests davon betroffen. Einerseits benötigen auch diese die Konfiguration für Windows, andererseits werden Test-Projekte anders ausgeführt als ASP.NET-Projekte. Dies führt dazu, dass die relativen Pfade zu den Applikationen hier in einer weiteren Variante vorliegen müssen.

Um sicherzustellen, dass alle Werte in den Konfigurationsdateien spezifiziert wurden, wird jeweils beim Einlesen der Daten ein kurzer Test durchgeführt. Einerseits wird überprüft, ob alle Attribute aus der Konfigurationsdatei in die entsprechende Klasse eingelesen werden können. Andererseits sind die Properties in der Klasse als Pflichtfelder markiert, wodurch verhindert wird, dass diese Felder leer bleiben.

Da diese Überprüfung jeweils beim Einlesen der benötigten Datei durchgeführt wird, bedeutet dies auch, dass jeweils nur eine Datei überprüft wird. Für die Unit Test Projekte ist dies kein Problem, da jeweils die entsprechende Datei geladen wird, solange Tests mit Zugriff auf die Konfiguration ausgeführt werden. Für die Applikation selbst heisst das, dass der Fehler jeweils beim ersten Zugriff auftritt. Da ein kurzer Testlauf vor einem Messeauftritt und insbesondere bei Änderungen an der Applikation sowieso empfehlenswert sind, wird es so keine Überraschungen geben, falls ein Feld erst in einem Spezialfall gelesen werden sollte.

Im Abschnitt 11.2.2.7 CONFIGURATION wird genauer beschrieben, wie die Konfigurationen gehandhabt werden und welche Auswirkungen sie auf die Applikation haben.

6.3 LIZENZEN

Als Lizenz wird Apache 2.0 [18] [19] für die gesamte CHOCOFORMER App verwendet. Diese Lizenz erlaubt es, die Software zu verwenden und verändern, auch im privaten Umfeld. Applikationen, welche die CHOCOFORMER App verwenden, müssen nicht unter der Apache Lizenz stehen. Dies kann ein Vorteil sein, wenn die Applikation zum Beispiel um einen Web-Shop ergänzt und dieser als separates Projekt implementiert wird.

Bei der Auswahl einer Lizenz gab es keine namentlichen Wünsche von Seiten Maintainer-Team. Solange die Applikation bzw. der Source Code noch nicht publik gemacht werden, wird die Lizenz auch nicht als hohe Priorität behandelt. Es soll sich allerdings um eine freie Lizenz handeln.

Der Entscheid fiel für die Apache 2.0 Lizenz aus, da diese einerseits im Open Source Guide von GitHub [20] empfohlen wird und andererseits einige der verwendeten Software auch unter dieser Lizenz veröffentlicht wurde.

6.3.1 VERWENDETE SOFTWARE MIT LIZENZEN

Name	Version	Lizenz
@typescript-eslint/eslint-plugin	2.34.0	MIT
@typescript-eslint/parser	2.34.0	BSD 2-Clause
coverlet.collector	1.2.0	MIT
ESLint	6.8.0	MIT
Immutable	4.0.0-rc.12	MIT
Material UI	4.9.12	MIT
Microsoft.AspNetCore.SignalR.Client	3.1.3	Apache-2.0
Microsoft.AspNetCore.SpaServices.Extensions	3.1.2	Apache-2.0
Microsoft.Extensions.Logging	3.1.3	MIT
Microsoft.NET.Test.Sdk	16.5.0	Microsoft .NET Library Software License [21]
Microsoft.TypeScript.MSBuild	3.8.3	Microsoft Visual Studio 2015 Add-Ons Software License [22]
Moment	2.24.0	MIT
Moq	4.12.1	BSD 3-Clause
Newtonsoft.Json	12.0.3	MIT
NLog	4.7.0	BSD 3-Clause
NLog.Web.AspNetCore	4.9.0	BSD 3-Clause
NWebsec.AspNetCore.Middleware	3.0.0	BSD 3-Clause
OpenSCAD	2019.05	GNU GPL v2
PrusaSlicer	2.2.0	GNU AGPL v3.0
React	16.13.1	MIT
TypeScript	3.7.5	Apache-2.0
xunit	2.4.0	Apache-2.0
xunit.runner.visualstudio	2.4.0	MIT

6.4 SICHERHEIT

Die CHOCOFORMER App ist so aufgebaut, dass Benutzer und Administrator jeweils eine unterschiedliche, unabhängige Web-Applikation verwenden. Sowohl das Front-End als auch das Back-End ist für den jeweiligen Verwendungszweck getrennt. Zudem finden keine Zugriffe über die Applikationsgrenzen hinweg statt, also zum Beispiel greift das Benutzer-Front-End nie auf das Back-End für den Administrator zu. Durch diese architektonische Trennung der Applikationen wird verhindert, dass ein Implementationsfehler einem unberechtigten Benutzer Zugriff auf die Konfigurationsseite für den CHOCOFORMER gewährt.

Um die Installation von DuetPi zu sichern, können die Empfehlungen für den Raspberry Pi [23] umgesetzt werden. Auf dem von dem Maintainer bereitgestellte DuetPi wurde bereits das Passwort geändert. Damit die aktuellen Software-Pakete mit den neuesten Sicherheits-Patches installiert sind, werden aus dem Installations-Skript der CHOCOFORMER App die Pakete aktualisiert.

Die einflussreichste Massnahme betreffend Netzwerk-Sicherheit durch das Maintainer-Team ist, dass der CHOCOFORMER zusammen mit dem Raspberry Pi grundsätzlich nicht mit einer aktiven Internet-Verbindung betrieben werden. Die einzige Ausnahme von dieser Regel ist die Aktualisierung des Betriebssystems und die Installation der CHOCOFORMER App, bei welcher Abhängigkeiten via APT installiert werden müssen.

An Messeständen wird der CHOCOFORMER zusammen mit den Notebooks für den Benutzer und den Administrator in einem lokalen Netzwerk betrieben. Um auch hier die Angriffsfläche zu minimieren, ist auf dem Raspberry Pi eine Firewall konfiguriert. So werden alle Ports blockiert, mit Ausnahme von SSH, was für den Zugriff und Datenaustausch durch den Administrator benötigt wird, und die Ports für die Web-Front-Ends der CHOCOFORMER App sowie des Duet Web Controls.

Der HTTP-Endpoint, welcher die Bilder für die Einlegebauteile nachlädt, greift direkt auf das Dateisystem zu. Auch wenn grundsätzlich nur auf PNG-Dateien zugegriffen werden kann, muss dieser besonders geschützt werden. Um zu verhindern, dass beliebige Pfade auf dem Dateisystem zugegriffen werden können, werden nur Pfade im Endpoint zugelassen, welche aus Buchstaben, Zahlen, Punkten, Bindestrichen und Leerzeichen bestehen. Pfade mit anderen Zeichen werden nicht zugelassen.

6.5 FEHLERBEHANDLUNG

6.5.1 FEHLERVERHALTEN NACHGELAGERTER SOFTWARE

Es ist wichtig, aber ebenso schwierig, Fehlerfälle zu behandeln, welche von den externen Applikationen verursacht werden. Sowohl PrusaSlicer [24] als auch OpenSCAD [25] bieten Statusmeldungen, welche typisch sind für Konsolen-Applikationen. Beide Applikationen arbeiten in einer einfachen Weise mit Exit-Codes. Bei erfolgreicher Durchführung wird 0 zurückgegeben, sonst 1. Eine einfache Analyse der Einstiegsklassen deutete auf keine anderen Exit-Codes. Ebenso werden Fehlermeldungen auf die Standard-Fehlerausgabe ausgegeben. Dies kann verwendet werden, um aus der CHOCOFORMER App

aussagekräftigere Fehlermeldungen zu generieren und in die Log-Datei bzw. dem Administrator anzuzeigen.

6.5.2 FEHLERVERHALTEN DES CHOCOFORMERS

Wichtig für die Entwicklung einer robusten Applikation mit einem externen Gerät ist zu wissen, wie das Fehlerverhalten des Gerätes ist. Der CHOCOFORMER wird mit Hilfe eines Duet3D-Boards gesteuert, dieses kennt vier Zustände [1]: "OK", "Generic error", "Incompatible DCS version" und "DCS is unavailable". Für die Kommunikation mit der Applikation sind vor allem die Zustände "Ok" und "Generic error" wichtig. Dadurch ist es schwierig herauszufinden, was das eigentliche Problem ist, da bei einem Fehler ein "Generic error" gesendet wird.

Durch einige Tests mit dem Test-3D-Drucker und in Kommunikation mit dem Maintainer-Team, wurde folgendes Verhalten beim CHOCOFORMER festgestellt:

- Wird dem CHOCOFORMER G-Code gesendet und er ist nicht gehomet, verwirft der CHOCOFORMER diese Anfrage und bewegt sich nicht, es wird keine Fehlermeldung gesendet.
- Erleidet der Chocoformer einen Schrittverlust, bedeutet, er bleibt aus welchem Grunde auch immer irgendwo stecken und befindet sich dadurch nicht mehr auf der geplanten Bahn, arbeitet der CHOCOFORMER normal weiter. Es wird keine Fehlermeldung gesendet.
- Wird dem CHOCOFORMER ein G-Code Befehl gesendet, während er bereits am Abarbeiten eines anderen Befehls ist, wird der neue Befehl für eine gewisse Zeit gespeichert. Ist der CHOCOFORMER mit dem alten Befehl fertig, führt er anschliessend den gespeicherten Befehl aus. Nach einem Timeout wird der Befehl verworfen. Es wird keine Fehlermeldung gesendet.
- Wird dem CHOCOFORMER ein ihm fehlerhafter G-Code Befehl geschickt, verwirft er den Befehl und macht nichts. Es wird keine Fehlermeldung gesendet.
- Wird dem CHOCOFORMER ein ihm unbekannter G-Code Befehl geschickt, verwirft er den Befehl und macht nichts. Es wird keine Fehlermeldung gesendet.
- Wird dem CHOCOFORMER ein G-Code gesendet, der eine Achse ausserhalb ihrer Reichweite fahren lässt, zum Beispiel wenn der X-Achse eine negative Koordinate mitgegeben wird, so wird der Befehl nicht ausgeführt.

Da der CHOCOFORMER bei allen, für diese Applikation relevanten Fehlern, nur bei einem Fehler ein potenzielles Fehlverhalten hat, muss die Applikation vor allem dafür sorgen, dass keine neuen Druck-Befehle gesendet werden, solange der Chocoformer am Drucken ist. Bei allen anderen Fehlern verwirft der CHOCOFORMER den Befehl und tut nichts.

6.6 DESIGN PATTERNS UND BEST PRACTICES

6.6.1 DEPENDENCY INJECTION

In der CHOCOFORMER App gibt es verschiedene Klassen, welche voneinander abhängig sind. Dabei verwenden diese Klassen wiederum andere Klassen mit verschiedenen Lebensdauern. Zudem verwendet jede Klasse eine eigene Logger-Instanz, was für die Lokalisierung von Log-Meldungen wichtig ist.

Die Instanziierung von einzelnen Objekt-Bäumen, insbesondere mit den Logger-Spezialisierungen, wurde bereits früh im Projekt komplex. Um dieses Problem zu lösen, wird Dependency Injection [28] eingesetzt. Dependency Injection wird verwendet, um Abhängigkeiten zwischen Klassen auf ein Minimum zu reduzieren. Dies sorgt für einfach wartbaren und weniger fehleranfälligen Code. Durch die gute Integration in ASP.NET Core kann Dependency Injection auch ohne grossen Konfigurationsaufwand einfach eingesetzt und konfiguriert werden.

Auch der Einstiegspunkt für die Konfigurationsdaten wurde mit dieser Massnahme vereinfacht. Nun wird von dieser Klasse an einem zentralen Ort eine Instanz erstellt und dabei darauf geachtet, dass je nach Umgebung die richtige Konfigurationsdatei angezogen wird. Darüber hinaus gibt es nun nur noch eine Instanz, welche durch das Dependency Injection Framework bereitgestellt wird, anstelle von vielen unkontrollierten, lokal erstellten Instanzen.

6.6.2 MATERIAL DESIGN

Für die Gestaltung der Benutzeroberfläche wird Material UI [26], eine Implementation von Material Design [27] für React, verwendet. Durch die Verwendung einer Design-Sprache und einer entsprechenden Bibliothek wird das Styling der Front-Ends vereinfacht und vereinheitlicht. So sind neben dem Aussehen der einzelnen Komponenten auch die Abstände zwischen einzelnen Komponenten und die Formulierung von Texten geregelt.

Die Farben und Abstände sind zentral geregelt und können in den jeweiligen Styling-Definitionen referenziert werden, wodurch sich das Aussehen einfach anpassen lässt. Für die Entwickler ist es somit zum Beispiel einfacher, ein Vielfaches des Basis-Abstands zu referenzieren. So werden Abstände konsistent, ohne dass eine separate Tabelle mit den zu verwendenden Abständen geführt werden müsste.

Als Farbschema wird die Farbtabelle des HSR Corporate Design [28] verwendet. Da die Farben zentral in einem JavaScript-Objekt festgehalten sind, können diese einfach verändert werden. Dies kann zum Beispiel notwendig sein, wenn die Farben im Rahmen der Zusammenlegung der Fachhochschulen zur OST [29] angepasst werden sollen oder falls für den CHOCOFORMER ein eigenes Farbschema festgelegt wird.

Für das Benutzererlebnis der CHOCOFORMER App ist die Verwendung von Material Design sinnvoll, da viele Benutzer durch die Verwendung von Google-Produkten bereits mit den Komponenten vertraut sind und für die tendenziell kurze Nutzung der Applikation keine überraschenden, neuen Verhaltensweisen erlernen müssen.

6.6.3 PRESENTATIONAL UND CONTAINER COMPONENTS

Um die Verantwortlichkeiten besser aufzuteilen und um somit für höhere Wartbarkeit zu sorgen, sind einzelne Komponenten, wie zum Beispiel der Bereich für die Drucker-Statusdaten, in zwei Dateien aufgeteilt.

Einerseits gibt es Presentational Components. In diesen Dateien wird ausschliesslich die Darstellung von Daten behandelt. Diese Dateien beinhalten also keine Logik, welche den Zustand der Applikation

verändert. Die Darstellung beziehungsweise das Styling in eine separate Komponente zu kapseln ist sinnvoll, da bei der Verwendung von externen Styling-Bibliotheken viel Code entstehen kann, welcher für die eigentliche Funktionalität einer React-Komponente nicht relevant sein kann. Gerade weil solche Presentational Components insgesamt sehr einfach aufgebaut sind und keine Logik verwalten müssen, lohnt es sich, diese als Funktionskomponenten zu implementieren.

Auf der anderen Seite gibt es die Container Components. Diese Komponenten verwenden jeweils die Presentational Component und ergänzen diese um die notwendige Logik, wie unter anderem die Verwaltung des Lebenszyklus, das Nachladen von Daten aus dem Back-End oder die Behandlung eines Button-Klicks. Dadurch, dass in diesen Dateien kein Styling behandelt werden muss, können sich Entwickler auf die eigentliche Logik konzentrieren. Um den Lebenszyklus zu verwalten, eignen sich Klassenkomponenten besser als Funktionskomponenten mit Hooks, da die einzelnen Phasen einfacher kontrolliert werden können.

Um die Verwendung der beiden Implementationsvarianten innerhalb des CHOCOFORMER App Projekts konsistent zu machen, werden Presentational Components immer als Funktionskomponenten umgesetzt und Container Components immer als Klassenkomponenten umgesetzt. Somit wird ein konsistenter Stil über alle Dateien erreicht, auch wenn sich der andere Stil für einzelne Komponenten besser eignen würde.

6.7 CI/CD

Die Continuous Integration erfolgt direkt über ein in GitLab integriertes Tool. Die Pipeline besteht aus zwei Schritten, dem Build und dem Test. Jedes Mal, wenn ein Push auf das Repository geschieht, wird die Pipeline durchlaufen. Damit soll sichergestellt werden, dass die Änderungen kompiliert werden können und die automatischen Tests durchlaufen. Diese Tests sind in die zwei Kategorien Unit-Test und Integrations-Test aufgeteilt, wie im nachfolgenden Kapitel 7.2 AUTOMATISCHE TESTVERFAHREN erläutert wird.

In diesem Projekt gibt es kein Continuous Delivery. Die CHOCOFORMER App läuft nicht auf einem zentralen Server, sondern wird auf einzelnen 3D-Drucker installiert. Wie bei Desktop-Applikationen, die auf unterschiedliche Geräte installiert werden, könnten die 3D-Drucker bei Bedarf auch in unterschiedlichen Versionen betrieben werden. Zudem ist es auch nicht garantiert, dass die Zielplattform, also der am CHOCOFORMER angeschlossene Raspberry Pi, eine Verbindung zu GitLab hat.

Durch den Verzicht auf Continuous Delivery wird auch die Stabilitäts-Anforderung sichergestellt. Das Deployment einer neuen Version der CHOCOFORMER ist eine bewusste Entscheidung der Maintainer. Damit wird also verhindert, dass eine Version automatisiert installiert wird, welche potenziell zu Problemen an einem Messestand führen kann.

6.7.1 QUALITÄTS-SICHERUNG

Für die Front-End-Projekte werden Überprüfungen mit ESLint durchgeführt. Dabei wird einerseits auf die korrekte bzw. konsistente Formatierung geachtet. Andererseits wird der Code auch auf potenzielle Probleme überprüft, wie zum Beispiel ob Variablen und Parameter unbenutzt sind.

Die Überprüfung mit ESLint findet dabei während des Builds auf GitLab statt. Eine alternative Umsetzung wäre, bei jedem Start der Web-Applikation die Überprüfung durchzuführen. Die gewählte Variante bietet den Vorteil, dass die Entwickler sich auf das Kernproblem des aktuellen Features bzw. das Debugging konzentrieren können, ohne dass sie durch die Linting-Meldungen unterbrochen werden. Obwohl so diese Überprüfung sehr spät stattfindet, vereinfacht dies das Setup für die einzelnen Entwickler und wird konsistent bei jedem Build auf GitLab ausgeführt. Entwickler können dennoch selbstständig das Linting durchführen, indem sie das entsprechende Kommando (`npm run lint`) aus der Konsole heraus aufrufen.

Für die Formatierung des Back-Ends werden die C# Coding Conventions [30] von Microsoft verwendet. Diese Conventions helfen den Code verständlicher und einfacher wartbar zu machen. Es werden Guidelines für Namensgebung, Layout, Kommentare, usw. vorgegeben. Arbeitet man mit Visual Studio, werden Verstöße gegen diese Guidelines und entsprechende Lösungsvorschläge angegeben. Zudem formatiert Visual Studio den Code direkt. Auch andere IDEs bieten diese Guidelines, wenn man in C# programmiert.

6.7.2 RELEASE

Neben den beiden oben erwähnten Pipeline-Schritten gibt es für Pipelines auf dem Master-Branch einen weiteren Schritt. Dieser muss von einem Maintainer manuell angestoßen werden und erstellt damit das Release Artifact. Dieses Release Artifact besteht aus der CHOCOFORMER App, welche für die Raspberry Pi Plattform kompiliert wurde, den vordefinierten Konfigurationsdateien und einem Installations-Skript.

Da das Erstellen eines Releases bei manuell ausgelieferten Applikationen eher selten geschehen kann und damit Details in Vergessenheit geraten können, ist dieser Prozess zusammen mit dem Installationsprozess auch in der README.md-Datei erklärt.

6.7.3 INSTALLATION

Das Installations-Skript sorgt dafür, dass sämtliche Daten aus dem Release Artifact in das jeweils passende Verzeichnis kopiert werden. Zudem werden die Abhängigkeiten via `apt install` und mittels Downloads installiert, sowie auch die Services für die beiden Web-Applikationen der CHOCOFORMER App registriert und gestartet.

Als weitere Funktionalität enthält das Skript auch eine Update-Option. Hier werden ausschliesslich die Web-Applikationen aus dem Release Artifact aktualisiert und die Services entsprechend neu gestartet. Weitere Update-Schritte müssen manuell ausgeführt werden. Zum Beispiel wäre es für das Update-Skript schwierig bis unmöglich zu entscheiden, welche Konfigurationsdateien aktualisiert werden müssen, welche Dateien unbedingt beibehalten werden sollten und wo allenfalls Teile aus beiden Dateien zusammengefügt werden müssen.

6.7.4 BETRIEB UNTER DUETPI

Die Web-Applikation wird als eigenständige Anwendung (auf Englisch self-contained executable genannt) bereitgestellt und als Konsolenapplikation gestartet. Dies hat den Vorteil, dass die Applikation auf den Raspberry Pi kopiert werden kann und dann ohne die zusätzliche Installation von .NET läuft. Da das Duet Software Framework ebenfalls auf .NET setzt, wird so zwar mehr Speicherplatz verwendet, dafür können potenziell Versionskonflikte vermieden werden. Die beiden Front-End-Applikationen befinden sich in einen geteilten Ordner, so dass wenigsten innerhalb der CHOCOFORMER App keine duplizierten .NET-Dateien vorhanden sind.

Da die Applikation beim Start des CHOCOFORMERs mitgestartet werden soll und bei Abstürzen ein erneutes Aufstarten der Applikation versucht werden sollte, sind die beiden Web-Anwendungen als Service in systemd registriert.

Eine Best Practice von Microsoft [31] sieht vor, dass ein Webserver wie Nginx oder Apache verwendet wird. Im Rahmen dieser Bachelorarbeit ergibt sich daraus allerdings kein Vorteil. So werden die automatisierten Neustarts bereits durch systemd abgehandelt. Weitere Funktionen, wie das Caching von Dateien und das Load Balancing sind für die CHOCOFORMER App aktuell nicht notwendig, da die App von einem einzelnen Browser aufgerufen wird und keine hohe Last durch viele Anfragen erzeugen wird. Ganz im Gegenteil entsteht so mehr Konfigurations- und Wartungsaufwand sowohl für die Entwickler als auch für die Maintainer, da ein solcher Webserver immer wieder aktualisiert und gegen mögliche Sicherheitsprobleme geschützt werden muss.

7 TESTING

7.1 ZIELE UND TESTKONZEPT

Mit Hilfe des Testing soll die Robustheit und Fehlerfreiheit der Applikation sichergestellt werden. Die Applikation kommuniziert mit einem externen Gerät und externer Software und wird von Menschen ohne Kenntnisse über die Applikation bedient. Daher sind verschiedene Testkonzepte nötig, um die Stabilität der Applikation zu gewährleisten.

In den nachfolgenden Kapiteln ist festgehalten, mit welchen unterschiedlichen Verfahren getestet wird. Durch die Kombination von automatischen und manuellen Testverfahren auf unterschiedlichen Ebenen kann sichergestellt werden, dass sowohl einzelne Teile als auch das Gesamtsystem zuverlässig funktionieren.

7.2 AUTOMATISCHE TESTVERFAHREN

Die automatischen Tests werden mit xUnit.net implementiert. Damit werden Unit-Tests und Integrations-Tests umgesetzt. Für jede Komponente, die in diesem System umgesetzt wird, müssen die Schnittstellen getestet werden.

Da einige Komponenten auf externe Programme zugreifen, werden die Zugriffe, wann immer möglich mit einem Mocking-Framework gemockt, um damit möglichst viele Abhängigkeiten zu minimieren.

7.2.1 INTEGRATIONSTESTS

Einige Komponenten lassen sich gut als Integrationstest zusammen mit den externen Applikationen testen. So kann eine vorgefertigte Input-Datei bereitgestellt werden und damit die entsprechende Applikation aufgerufen werden. So lässt sich sofort feststellen, ob die Parameter für den Aufruf richtig konfiguriert werden.

Ein Problem dieser Integrationstests ist, dass die aufgerufenen Applikationen plattform-abhängig sind. Aus diesem Grund werden die Integrationstests nicht via Continuous Integration auf GitLab überprüft. Anstelle dessen ist es in der Verantwortung des Entwicklers, diese Tests durchzuführen.

Auf GitLab werden für jeden Build-Vorgang die Abhängigkeiten erneut heruntergeladen. Aus dieser Sicht ist der Ausschluss der Integrationstests ein Vorteil, da so nicht für jeden einzelnen Build-Vorgang alle Abhängigkeiten erneut heruntergeladen werden müssen. Dadurch wird eine grosse Anzahl von Downloads der Applikationen vermieden und die GitLab-Infrastruktur der HSR nicht weiter belastet.

7.3 MANUELLE TESTVERFAHREN

Szenarien, welche nicht automatisch getestet bzw. gemockt werden können, werden entsprechend dokumentiert und manuell getestet. Zudem werden auch End-to-end-Tests manuell durchgeführt. Damit wird sichergestellt, dass der gesamte Druckprozess auch dann funktioniert, wenn keine Komponenten durch ein Mocking-Framework ersetzt werden.

7.3.1 TESTEN AM CHOCOFORMER

Um zu testen, wie sich die Applikation im Zusammenspiel mit dem CHOCOFORMER verhält, wurden Tests direkt mit dem CHOCOFORMER durchgeführt. Die Tests waren vor allem wichtig, um das Zusammenspiel der beiden Druckköpfe zu testen. Da der 3D-Drucker für die Entwicklung nur einen Druckkopf besitzt, konnte diese Funktion vorher nie getestet werden.

Aufgrund der speziellen Situation mit dem Coronavirus, konnten die Tests nur nach Absprache mit der Schulleitung erfolgen. Hygienemassnahmen mussten eingehalten werden und es wurde auf Social Distancing geachtet. Zusätzlich musste der Test später durchgeführt werden als ursprünglich geplant. Der Test sollte mit dem neuen CHOCOFORMER erfolgen, der im Rahmen einer Bachelorarbeit dieses Semesters gebaut werden sollte. Da wegen der Ausnahmesituation der Bau jedoch nicht weitergeführt werden konnte, musste der bestehende CHOCOFORMER zuerst umgebaut werden. Zudem war es Studenten zuerst nicht gestattet, die Laborräume der HSR zu betreten. Es konnten somit nur zwei Tests am CHOCOFORMER direkt durchgeführt werden: am 13.05.2020 und am 27.05.2020.

Vor den Tests wurden die Wissensziele festgehalten, sprich welche Funktionen getestet werden sollen. Dabei wurde das Ziel des Tests, die Vorbedingungen, die Interaktion, das erwartete Resultat und die geschätzte Zeit festgehalten. Während des Tests wurden das tatsächliche Resultat, einzuplanende Aktionen, Umsetzung und der Erfolg des Tests dokumentiert. Am Ende wurde eine Auswertung über alle Tests aufgestellt. Die Versuchspläne für die Tests befinden sich im Anhang unter 16.3 VERSUCHSPLÄNE.

7.3.2 TESTEN OHNE 3D-DRUCKER

Die statischen Anzeige-Funktionalitäten des Administrator-Dashboards können auch ohne angeschlossenen 3D-Drucker getestet werden. Dafür steht im Repository ein Projekt für SoapUI bereit. SoapUI erlaubt es, HTTP-Requests zu behandeln und eine anpassbare Antwort zurückzugeben, ohne dass das Back-End angepasst werden muss.

Als Beispiel ist diese Möglichkeit besonders geeignet, um die Anzeige von verschiedenen Zuständen der Sensordaten zu testen. Sensordaten können einen Status, wie "active" oder "fault" haben. Zudem könnte die zurückgelieferte Temperatur den Bereich verlassen, welcher für Schokolade geeignet ist. Solche Fehlerzustände bzw. deren unterschiedlichen Kombinationen sind schwierig einzustellen oder können mit einem herkömmlichen 3D-Drucker nicht nachgestellt werden. Daher ist es einfacher, diese Daten mit SoapUI zu mocken. Die Konfiguration muss in SoapUI nicht neu geladen oder kompiliert werden und steht daher sofort für den nächsten Request bereit.

Um SoapUI zu nutzen, muss das bereitstehende Projekt in SoapUI geöffnet und gestartet werden. Da damit ein eigener Webserver gestartet wird, müssen die entsprechenden Requests aus dem Front-End auf den richtigen Port umgeleitet werden. Dafür ist es ausreichend, wenn den URLs in den Dashboard-Komponenten "http://localhost:8080" vorangestellt wird.

Die weiteren Funktionalitäten der Front-Ends profitieren nicht von einem solchen Mocking. Das Benutzer-Front-End bzw. der Konfigurations-Bildschirm für den Administrator sind nicht von einem

angeschlossenen Drucker abhängig. Diese Daten können direkt über das Admin-Front-End bzw. die Dateien auf dem Datenträger eingesehen und angepasst werden.

7.4 USABILITY TESTS

Die Usability-Tests werden mit Hilfe von Testpersonen, welche vordefinierte Szenarien durchlaufen, durchgeführt. Dies ist eine qualitative Herangehensweise. Für diesen Usability Test gibt es zwei verschiedene Arten von Testpersonen. Die Benutzer, welche die Applikation testen und die Administratoren, die das Dashboard und die Konfigurationen testen.

7.4.1 WISSENSZIELE

Für die Usability-Tests rücken drei Fragen besonders in den Vordergrund:

Benutzer Front-End

- Können die Benutzer den Abläufen der Applikation folgen?
- Ist den Benutzern klar, was sie tun müssen?
- Ist den Benutzern klar, was passiert, wenn sie ihren Druck abgeschickt haben?

Admin-Front-End

- Versteht der Administrator was auf dem Dashboard passiert (Queue, Monitoring, momentaner Job)?
- Kann der Administrator ein Einlegebauteile hinzufügen und löschen?
- Weiss der Administrator, wie er die Jobs managen kann?

7.4.2 TESTPERSONEN REKRUTIEREN

Für den Usability Test müssen Testpersonen rekrutiert werden. Da ab einer gewissen Anzahl Testpersonen der Wert der neuen Erkenntnisse nicht mehr signifikant ist, macht es irgendwann keinen Sinn mehr mit weiteren Personen zu testen. Usability Test sind teuer und zeitintensiv, daher sollte die Anzahl Testpersonen gut gewählt sein. Anhand der untenstehenden Grafik ist zu sehen, dass eine gute Anzahl Testpersonen bei etwa sechs liegt, daher werden für diesen Usability Test fünf Testpersonen evaluiert.

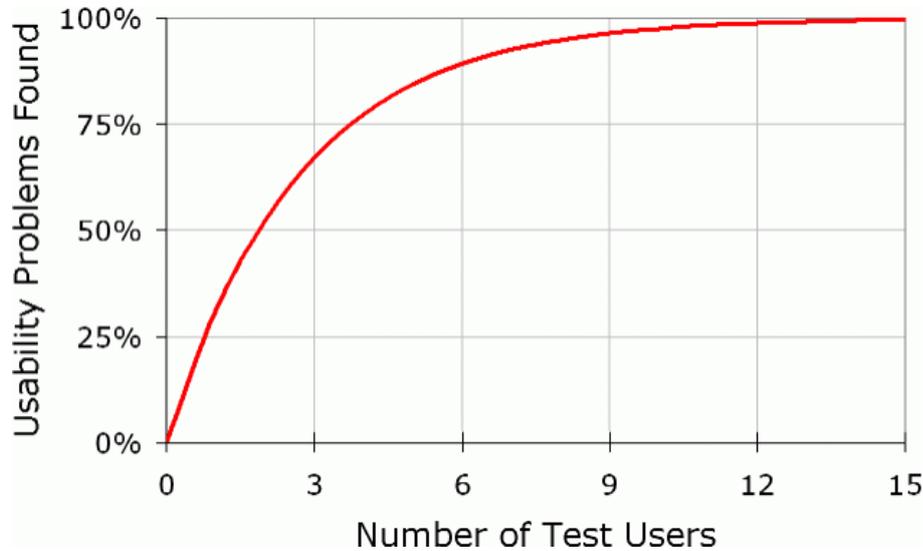


Abbildung 7 Vergleich Anzahl Testpersonen und Anzahl gefundener Usability Probleme [32]

Die Testpersonen für die Benutzer werden mit Hilfe eines Doodles gesucht. Aufgrund der momentanen Situation mit dem Coronavirus ist Auswahl der Testpersonen auf die Mitarbeiter an der HSR beschränkt. Die Testpersonen sollten möglichst unterschiedlich sein, um ein realistisches Publikum zu simulieren. Die wichtigsten Unterscheidungsunkte sind Alter, Beruf, und technisches Wissen.

Für das Testen des Administrator-Front-Ends werden die Testpersonen aus dem Kreis des Maintainer-Teams gewählt. Da der Administrator eine Einweisung vor der Benutzung erhält und zu dem Zeitpunkt des Tests nur zwei Personen aus dem Maintainer-Team anwesend sein können, wird mit diesen zwei Personen der Test durchgeführt.

7.4.3 ABLAUF

Aufgrund der momentanen Lage mit dem Coronavirus und den daraus folgenden Empfehlungen und Bestimmungen müssen die Usability-Tests darauf angepasst werden.

In der Planung wird die Gesundheit aller Beteiligten in den Vordergrund gestellt. Die Empfehlung des Bundes sollen auf jeden Fall eingehalten werden. Zu den Hygienemassnahmen gehören unter anderem: Es dürfen sich maximal fünf Personen am CHOCOFORMER befinden und der Mindestabstand von zwei Meter ist einzuhalten. Daher werden die Testpersonen gestaffelt zum Test gebeten. Es stehen zwei Zeiträume mit jeweils drei Testpersonen zur Verfügung: 10:30-11:00 und 11:30-12:00. Jede Testperson hat somit 10 Minuten Zeit, sich mit der Applikation auseinander zu setzen. Ein Mitglied der Testleitung übernimmt dabei die Rolle des Administrators.

Anschliessend wird der Administrator getestet. Ein Mitglied der Testleitung übernimmt dabei die Rolle des Benutzers

7.4.4 TESTAUFBAU

Der CHOCOFORMER wird mittels Ethernets mit zwei Laptops verbunden. Auf dem ersten Laptop wird die Applikation für die Benutzer und auf dem zweiten wird die Applikation für den Administrator gestartet.



Abbildung 8 Aufbau für den Test am CHOCOFORMER

7.4.5 TESTSZENARIEN

Damit die Benutzer unsere Software möglichst unvoreingenommen testen können werden sie nur mit den nötigsten Informationen versorgt. Dadurch stellen wir sicher, dass keine Unklarheiten überdeckt werden. Die Administratoren werden vorher in die Applikation eingeführt. Dies aus dem Grund, da auch später der Administrator immer vor Benutzung der Applikation geschult wird. Um die definierten Wissensziele zu erreichen, müssen die Benutzer und Administratoren folgende Testszenarien absolvieren.

Benutzer:

1. Du bist auf einer Messe und möchtest den CHOCOFORMER ausprobieren.
2. Du möchtest gerne einen Schokoladendruck mit nach Hause nehmen.

Administrator:

1. Es steht eine weitere Messe an und ihr wollt die Benutzer einen herzförmigen Keks bedrucken lassen. (Bild wird vorgegeben)
2. Es befindet sich ein Einlegebauteil im CHOCOFORMER, welches nicht mehr benötigt wird.
3. Du bist an einer Messe und mehrere Besucher haben die Applikation ausprobiert.
4. Du möchtest überprüfen, ob der CHOCOFORMER noch korrekt funktioniert.
5. Es wird eine neue Schokoladenart an einer Messe verwendet.

7.4.6 NACHINTERVIEWS

In den Nachinterviews soll zuerst die generelle Einstellung und Stimmung gegenüber der Applikation abgeholt werden. Auch konkrete Probleme und Unstimmigkeiten sollen hier noch einmal aufgenommen und festgehalten werden. Die Testpersonen werden einen kurzen Fragebogen ausfüllen. Die

ausgefüllten Fragebogen der Testpersonen befinden sich im Anhang unter 16.4 AUSGEFÜLLTE FRAGEBOGEN

7.4.7 DURCHFÜHRUNG DER TESTS

Bei der Durchführung der Tests gilt es einige wichtige Punkte zu beachten:

- Der Testperson wird mitgeteilt, dass der Test anonym abläuft und keine persönlichen Daten erfasst werden.
- Die Testperson wird darauf aufmerksam gemacht, dass die Applikation und nicht sie selbst das Testobjekt sind.
- Die Testperson wird darum gebeten, während dem Prozess/Test laut zu denken.
- Die Testperson wird darauf hingewiesen, dass die Aufgaben möglichst selbständig durchgeführt werden und nur in Ausnahmefällen gefragt werden soll.
- Der Testleiter sollte während des Tests möglichst nichts sagen.

7.4.8 AUSWERTUNG DER TESTERGEBNISSE

Den Testpersonen war es möglich, alle Testszenarien erfolgreich zu absolvieren. Die Wissensziele konnten beantwortet werden. Insgesamt hinterliess die Applikation einen positiven Eindruck. Es können jedoch einige wichtige Verbesserungen vorgenommen werden. Anschliessend befinden sich die Protokolle der Testversuche und eine Zusammenfassung des Fragebogens. Einiges an Feedback war zu erwarten, wie etwa fehlende Funktionen für die Gestaltung, die aus zeitlichen Gründen nicht mehr implementiert werden konnten. Funktionen und Aspekte, die bereits im Kapitel 8.5 MÖGLICHKEITEN DER WEITERENTWICKLUNG behandelt werden, wird in diesem Abschnitt nicht noch einmal angesprochen.

PROTOKOLLE

Testperson 1

- gute Übersicht
- Einlegebauteil Bildschirm: "Teil" zu ungenau, ändern zu "Unterlage" oder ähnliches. Genauer Beschreiben was zu tun ist
- Mit einem Einlegebauteil verwirrend, da Auswahl, aber man hat keine Auswahl --> z.B. Ausgegraute Dummies
- Freihand / Text nicht klar was gewählt wurde, da Feedback fehlt → (z.B. Farbe des Buttons ändert sich dauerhaft solange ausgewählt)
- Logisch gestaltet, erinnert an Bestellvorgang, sehr intuitiv
- Weitere Informationen nach Abschicken des Drucks: Anzeigen ungefähre Zeit, wann Druck abgeschlossen ist.

Testperson 2

1. Nicht klar was passiert, wenn auf «Freihand» gedrückt wird → Button anders anschreiben oder Beschreibung dazu

2. Verwirrt, dass Freihand zeichnen nicht mit Maus funktioniert
3. Schwierig zuerst malen und dann Text, da Text festgelegten Platz hat.
4. Denkt, man kann noch etwas machen, nachdem Einlegebauteil gewählt wurde, nicht direkt daran gedacht auf weiter zu klicken → evt. beschreiben
5. Schön gestaltet

Testperson 3

- Stepper ist sehr gut für Übersicht was passiert
- Probiert zu zeichnen obwohl noch im Schritt 2, erste Intuition, direkt reinschreiben
- Text leider nicht verschiebbar
- Character Beschränkung im Text ist gut
- Google Layout ist bekannt und schön gestylt
- Touch Support ist nice
- Schritt 2 ist etwas verwirrend, da nur eine Option vorhanden --> evt. erste Option als Default laden

Testperson 4

- 3-d oder drei-dimensional verwenden, kein Gemisch
- Was tun nachdem Keks ausgewählt wurde?
- Auswahl von Einlegebauteilen und Zeichnen/Text verdeutlichen --> evt. mit Radio Buttons, damit verständlicher ist, was gewählt wurde.
- Nicht klar, was man tun muss, wenn man auf Freihand gedrückt hat, beim Text sieht man, dass etwas passiert
- Je nachdem, wie gross das Einlegebauteil ist, kann mit dem Text darüber hinaus geschrieben werden
- Bevor der Druck abgeschickt wird, könnte eine Vorschau gezeigt werden, was gedruckt wird

Testperson 5

- Mit nur einem Einlegebauteil ist die Auswahl bescheiden.
- Mehr Auswahl wäre schön.
- Nur mit dem Finger zeichnen ist ungenau.
- Ein Stift für die Eingabe ist unbedingt notwendig.

Was uns aufgefallen ist während dem Test

- Nur eine Schicht wäre besser, da der Druck sonst zu lange dauert
- Zurück-Swipen (in Google Chrome) besser verbieten
- Grösse -10 rechnen, damit nichts über den Rand hinausgeht
- Schnelles Malen ist nicht so vorteilhaft → Hinweis in Applikation

AUSWERTUNG FRAGEBOGEN

Wie würdest du die Applikation als Ganzes benoten? (1 = mangelhaft, 6 = sehr gut)

1	2	3	4	5	6

Durchschnitt: 5.4

Wie gut hat dir die Gestaltung gefallen? (1 = mangelhaft, 6 = sehr gut)

1	2	3	4	5	6

Durchschnitt: 5.4

Welche Elemente der Gestaltung haben dir besonders gefallen / nicht gefallen?

Positiv	Negativ
<ul style="list-style-type: none"> - Bekannte Bestelldarstellung - Einfacher Aufbau - Straight Forward 	<ul style="list-style-type: none"> - Nur ein Objekt zur Auswahl ist verwirrend - Kleine Zeichnungsfläche - Manchmal unklar, was ausgewählt wurde, z.B. welcher Unterlage oder welches Tool

Wie gut hast du dich zurechtgefunden? (1 = mangelhaft, 6 = sehr gut)

1	2	3	4	5	6

Durchschnitt: 5.0

Was hat dir am besten an der Applikation gefallen?

- Struktur der Bestellung
- Logischer Aufbau
- Die Druckunterlage wird als Bild gezeigt
- Kein Filter
- Generelle Idee der Applikation
- «Assistent-Design» wie Schritt für Schritt vorgegangen wird

Was hat dich am meisten gestört?

- Schriftarten wirken unterschiedlich
- Wechselnder Gebrauch von Maus und Touchpad → ohne vorhandene Maus wäre das kein Problem

Hast du Informationen oder Funktionen vermisst? Wenn ja, welche?

- Zeitangabe zum Druck
- Feedback beim Auswählen
- Den Text verschieben, ähnlich wie Word
- Informationen: welche Unterlage ausgewählt ist, welches Tool
- Vorschau des Drucks vor dem Drucken

7.4.9 ERARBEITEN VON VERBESSERUNGSVORSCHLÄGEN

VERBESSERUNG FÜR DAS ENDPRODUKT

Die Buttons für das Freihand zeichnen und das Text schreiben müssen umbenannt werden. Jede der Testpersonen war zuerst irritiert und nicht sicher, was sie zu tun hat.

Die Buttons im "Sujet erstellen"- und "Unterlage wählen"-Tab werden durch Toggle Buttons ersetzt. Dadurch wird dem Benutzer ersichtlich, welche Option er im Moment ausgewählt hat.

Die Beschreibungen im "Sujet erstellen"- und "Unterlage wählen"-Tab werden genauer spezifiziert.

VERBESSERUNGEN FÜR SPÄTERE PROJEKTE

Nachdem der Benutzer seinen Druck abgeschickt hat, wird ihm seine ID präsentiert. Zusätzlich könnte noch die geschätzte Wartezeit, bis der Druck fertig ist, angezeigt werden.

Bevor der Benutzer seinen Druck abschickt, könnte eine Vorschau des Drucks dem Benutzer präsentiert werden.

Wenn im Tab "Unterlage wählen" nur ein Einlegebauteil zur Auswahl steht, sollte dieses von der Applikation direkt verwendet werden. Der Benutzer muss in diesem Fall keine Unterlage wählen und geht direkt weiter. Möglicherweise kann der zweite Schritt auch direkt von der Applikation übersprungen werden.

GENERELLE EMPFEHLUNGEN FÜR DEN GEBRAUCH DER APPLIKATION

Die Tests wurden mit einem Microsoft Surface mit Tastatur durchgeführt. Vor allem betreffend des Freihand zeichnen, gab es einige Schwierigkeiten mit der Bedienung der Applikation. Einige Testpersonen haben versucht, mit dem Mousepad zu zeichnen, die Zeichenfunktion reagiert jedoch nur auf Touch. Daher ist für den Gebrauch an Messen ein Tablet, ohne oder mit zurückklappbarer Tastatur zu empfehlen.

Wird Google Chrome als Browser für die Applikation gewählt, können die Swipe-Funktionen, vor allem beim Freihand zeichnen, stören. Diese können folgendermassen deaktiviert werden: unter `chrome://flags/#overscroll-history-navigation` «Overscroll history navigation» auf disabled setzen.

Wenn Mozilla Firefox verwendet wird, muss der Computer in den Tabletmodus gesetzt werden. Ansonsten wird die Bildschirmtastatur nicht automatisch aktiviert. Dieser Modus kann aktiviert werden,

indem die Benachrichtigungsleiste auf der rechten Seite geöffnet wird und die Tabletmodus-Aktion ausgewählt wird.

Während des Tests wurde mit einer Extruder-Höhe von 1.2mm gearbeitet. Dies entspricht zwei gedruckten Schichten. Dadurch betrug die Druckzeit von einigen Sujets fast zehn Minuten, obwohl die Motive relativ klein ausfielen. Daher ist es empfehlenswert, die Extruder-Höhe auf 0.6mm, also auf eine Schicht, zu setzen, um zu lange Druckzeiten zu vermeiden.

8 RESULTATE UND WEITERENTWICKLUNG

Für die Versionsverwaltung des Projekts wurde das interne GitLab der HSR¹ verwendet (TODO: Link auf Repo). Die Entwicklung wurde mit Hilfe von Feature-Branches durchgeführt. Nur Code, der von beiden Entwicklern begutachtet und für richtig empfunden wurde, ist auf den master-Branch gepusht worden. Mit Hilfe dieser Reviews wird die Qualität des Source Codes überprüft. Zudem erlaubt die Unterteilung in Feature-Branches, dass Features erst dann in den master-Branch gelangen, wenn diese einsatzbereit sind. So ist der master-Branch stets in einem stabilen Zustand, welcher für einen Release auf den CHOCOFORMER verwendet werden kann.

8.1 RESULTATE

In der Zeit, welche zur Verfügung stand, konnte ein optisch ansprechendes, funktionierendes Produkt entwickelt werden, welches für Messen einsatzbereit ist. Insbesondere der Ablauf, um einen Druck zu erstellen, ist vollständig implementiert und enthält eine Queue, welche diesen Ablauf an Messeständen optimiert. Damit geht das entwickelte Produkt bereits über ein Minimum Viable Product (MVP) hinaus.

Das Produkt als Ganzes ist gut erweiterbar aufgebaut. Die Erweiterung um zusätzliche Formatierungsoptionen für den Benutzer ist soweit vorbereitet, dass die zusätzlichen Konfigurationen einfach eingefügt werden können. Auch um die Konfiguration gänzlich über das Administrator Front-End abzuhandeln ist bereits im Grundsatz implementiert, so dass nur noch die weiteren Bereiche analog zur existierenden Konfiguration der Einlegebauteile ergänzt werden müssen.

Darüber hinaus sind die meisten Bereiche der Applikation konfigurierbar. Die existierenden Schriftart-Optionen, die Kommandozeilen-Schalter für das CAD und den Slicer sowie auch verschiedene weitere Konfigurationen können vorgenommen werden, ohne dass die Applikation neu kompiliert und installiert werden muss.

Usability-Tests wurden erfolgreich durchgeführt und die Probanden konnten die Applikation ohne zusätzliche Hilfe bedienen, somit sollte es für Besucher einer Messe auch kein Problem sein, die Applikation selbstständig zu benutzen.

8.2 WEBSOCKET VS. HTTP

Zu Beginn der Arbeit war angedacht, dass die Status-Daten des Duet-Controllers über WebSocket des DuetWebServers ausgelesen werden. Da der WebSocket kontinuierlich Daten von Sensoren sendet, welche in der CHOCOFORMER App nicht verwendet werden, wurde eine andere Lösung umgesetzt. Der DuetWebServer bietet ebenso eine HTTP-Schnittstelle an, welche alle Daten in einem Durchgang liefert. Das Entwickler-Team hat sich für diese zweite Variante entschieden, da diese Variante den Raspberry Pi von dem kontinuierlichen Datenfluss des WebSockets entlastet. Zudem ist es damit einfacher, die Daten in einem konsistenten Zustand anzuzeigen, da ansonsten Daten, die

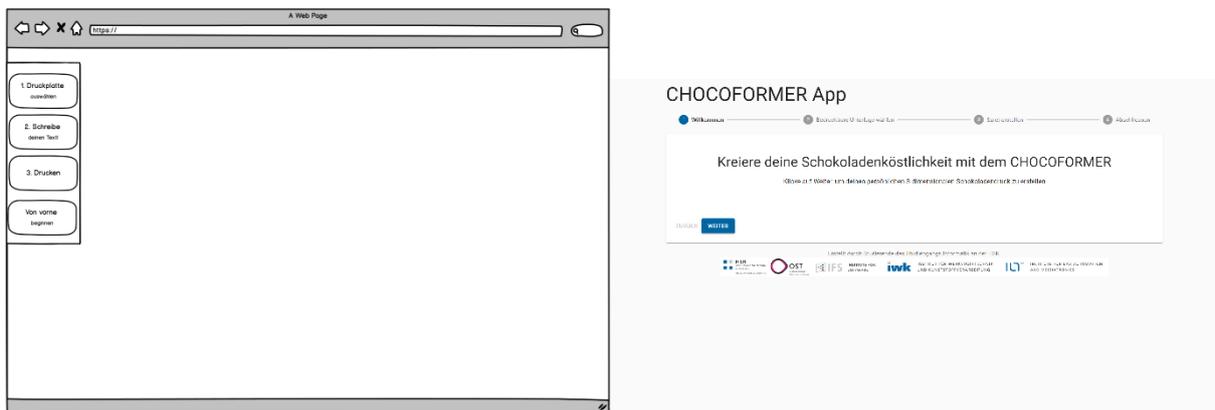
¹ <https://gitlab.dev.ifs.hsr.ch/chocoformer/chocoformer-app>

zusammengehören potenziell separat geladen werden und dadurch inkonsistente Warnungen ausgegeben werden könnten.

8.3 VERGLEICH WIREFRAMES - RESULTAT

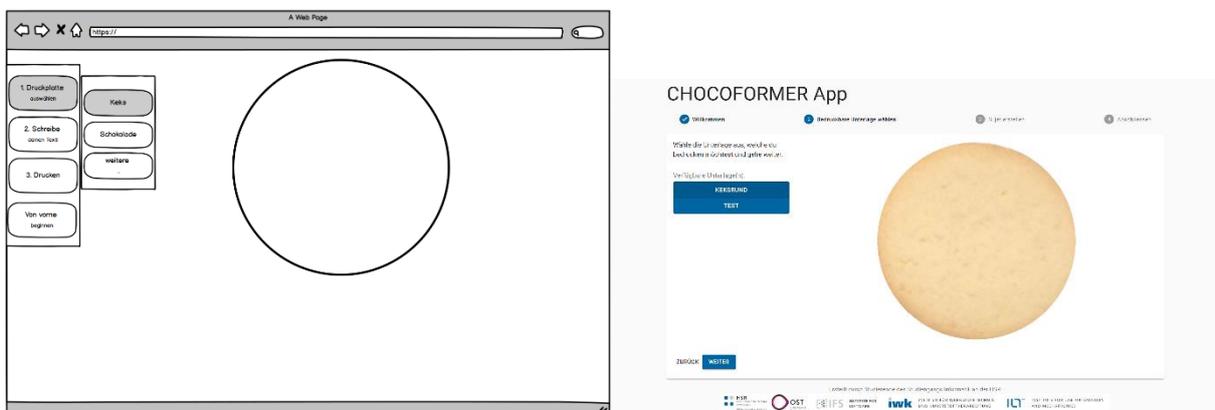
Die Gestaltung des Front-Ends konnte entsprechend der erstellten Wireframes umgesetzt werden. Es wurden nur einig kleine Änderungen vorgenommen.

8.3.1 BENUTZER FRONT-END

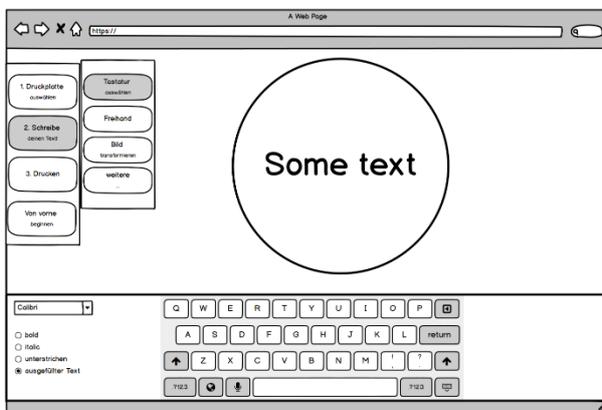
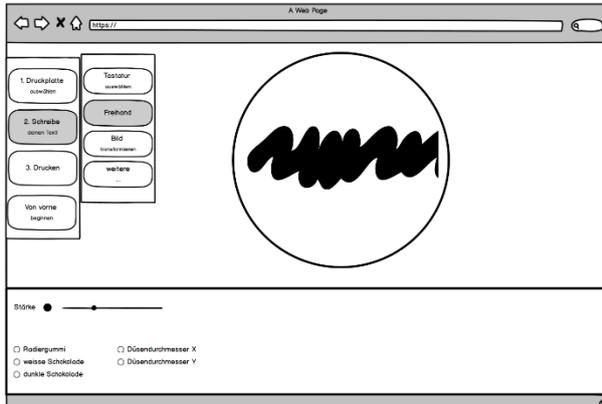


Anstatt die Navigation mit Leiste mit Buttons am linken Rand zu realisieren, wurde ein vorgefertigter Stepper verwendet. Dadurch kann der Benutzer bewusst durch die Applikation geführt werden. Zudem erhöht das Benutzen eines Steppers die Verständlichkeit für den Benutzer, da dieser in vielen Anwendungen, wie z.B. dem Checkout in einem Online Shop, verwendet wird. Der Benutzer hat somit eine Vertrautheit mit der Applikation, obwohl er sie noch nie bedient hat.

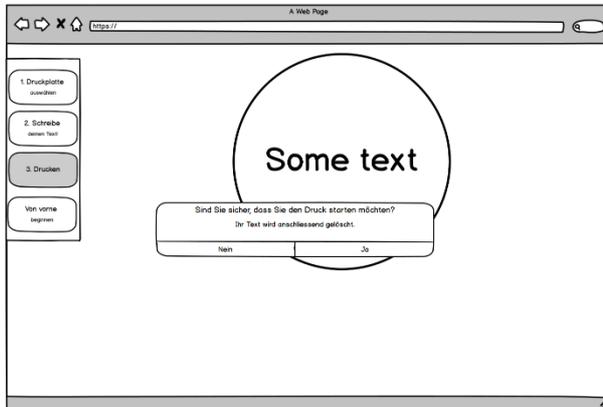
Zusätzlich hat die Applikation einen Titel und einen Willkommenstext erhalten, um den Benutzer abzuholen. Auch wurde ein Banner mit den Logos der Institute und der Hochschule eingefügt.



Die Auswahl des Einlegebauteils ist nach dem gleichen Prinzip aufgebaut wie in den Wireframes. Zudem wurde eine Einführungstext geschrieben, damit der Benutzer weiss, was er machen kann.

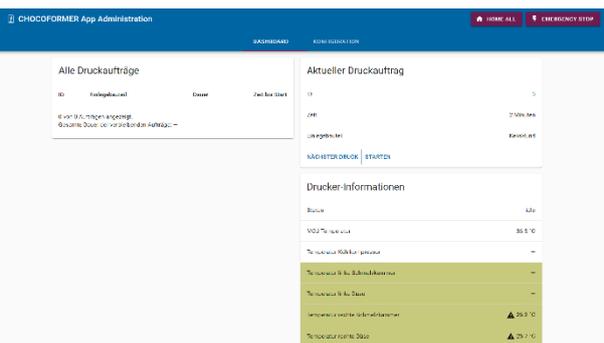
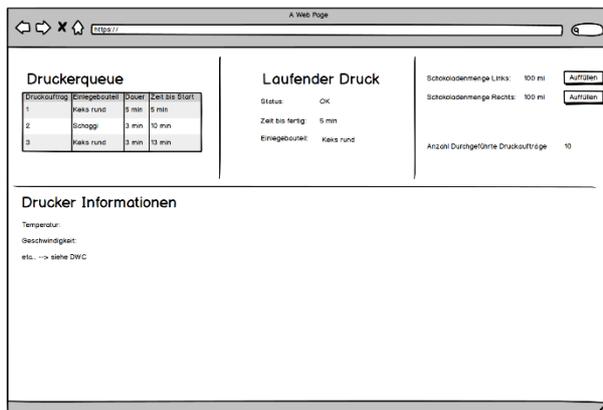


Für das Zeichnen mit dem Touchscreen wurde der Button von «Freihand» auf «Male mit dem Stift» umbenannt. In den Usability Test wurde klar, dass so gut wie alle Testpersonen mit Freihand nichts anfangen können. Ebenso beim Button «Tastatur», dieser wurde auf «Schreibe mit der Tastatur» umbenannt. Die Optionen zum Anpassen der Schrift oder der Dicke des Pinsels wurden aus Zeitgründen nicht implementiert. Jedoch ist es dem Administrator möglich, über ein Konfigurations-File die Schriftart, die Schriftgröße und die Pinselstärke anzupassen. Da der CHOCOFORMER zwei Druckköpfe besitzt, ist dem Benutzer die Möglichkeit gegeben mit zwei verschiedenen Farben sein Sujet gestalten.

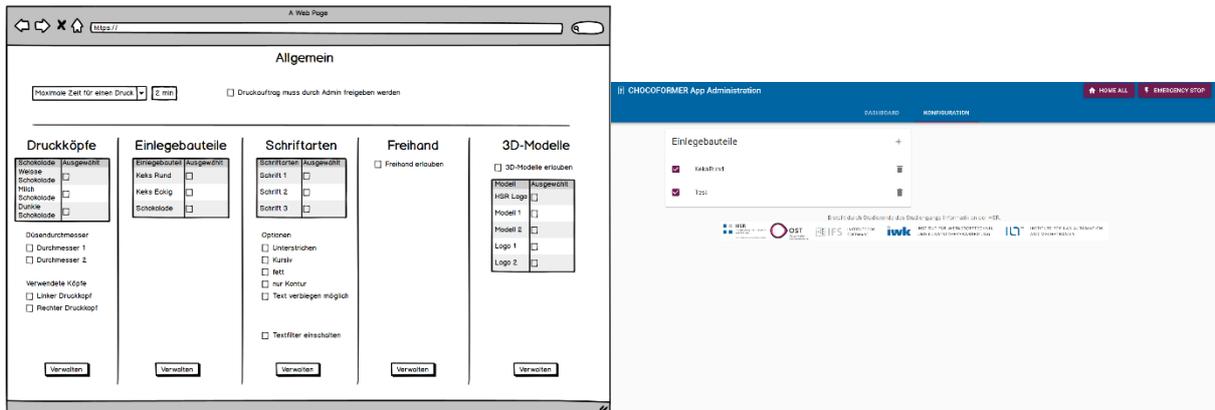


Das Starten des Drucks wurde als einziges Element stark umgestaltet und hat zwei Fenster erhalten. Der Benutzer kann den Druck nicht selbst starten, sondern er reiht ihn in eine Queue ein und der Administrator startet die Drucke anschliessend. Auch erscheint kein Pop-up mehr mit einer Bestätigung, der Benutzer wird stattdessen bevor er auf Drucken klickt informiert. Nachdem der Benutzer den Druck abgeschickt hat, wird ihm eine ID mitgeteilt. Mit dieser kann er herausfinden, wann sein Druck fertiggestellt ist.

8.3.2 ADMINISTRATOR FRONT-END



Das fertige Administrator Front-End entspricht sehr genau dem erstellten Wireframe. Nur die Anordnung der verschiedenen Karten hat sich während der Entwicklung verändert. Eine Karte, mit der Anzeige, wieviel Schokolade noch übrig ist, konnte aus zeitlichen Gründen nicht mehr implementiert werden. Die Basis dieser Karte existiert bereits, da sie jedoch keine Logik enthält, wird sie im fertigen Produkt nicht angezeigt. Dafür konnten ein «Home»-Button und ein «Emergency Stop»-Button, welche vom Maintainer-Team während der Implementation gewünscht wurden, eingebaut werden.



Auf der Konfigurationsseite konnte nur das Management für Einlegebauteile implementiert werden. Andere Konfigurationen konnten aus Mangel an Zeit nicht mehr umgesetzt werden. Die Applikation ist jedoch soweit vorbereitet, dass weitere Konfigurationen einfach hinzufügbare sind.

8.4 WELCHE USE CASES WURDEN ERREICHT?

Von den Use Cases für den Administrator konnten UC01 (Druckstatus verwalten) und UC05 (Einlegebauteile verwalten) im Front-End umgesetzt werden. UC02 (Druckköpfe verwalten) und UC03 (Schriftarten verwalten) sind über eine Konfigurationsdatei im Back-End erfüllt. Aus Zeitgründen konnte UC04 (3D-Modelle verwalten) nicht mehr implementiert werden. Dieser Use Case hatte zudem die niedrigste Priorität, da er unter anderem nicht für den minimalen Ablauf eines Druckprozesses notwendig ist.

Von den Use Cases für den Benutzer konnten UC06 (Text/Gruss eingeben), UC06.1 (Text/Gruss mit vorgegebener Schriftart erstellen), UC06.2 (Text/Gruss Freihand erstellen) und UC08 (Druckauftrag senden) umgesetzt werden. Die optionalen Use Cases UC07 (3D-Modell auswählen) und UC09 (Bilddatei in Druckdatei transformieren) sind aus Zeitgründen nicht mehr implementiert worden.

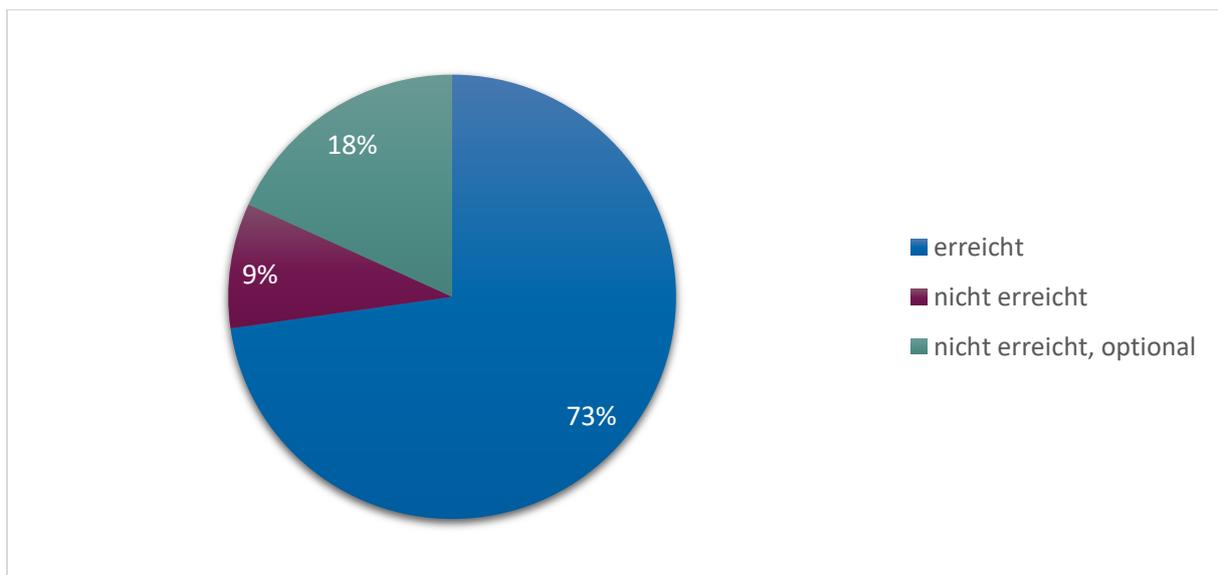


Abbildung 9 Erreichte vs. nicht erreichte Use Cases

Neben den erreichten Use Cases wurde auch eine Queue für die Druckaufträge realisiert. Nachdem die Anforderungen bereits abgenommen wurden, kristallisierte sich aus den Gesprächen mit dem Maintainer-Team immer mehr heraus, dass eine solche Queue für den Betrieb an Messeständen notwendig ist. So beschleunigt die Queue auch den Ablauf, da ein Besucher bereits ein Sujet erstellen kann, bevor der Druck der vorherigen Besucher abgeschlossen wurde. Aus diesem Grund wurde die Queue gegenüber der vollständigen Konfiguration über das Admin-Front-End und dem Use Case mit der niedrigsten Priorität priorisiert.

8.5 MÖGLICHKEITEN DER WEITERENTWICKLUNG

In den nachfolgenden Unterkapiteln ist erwähnt, welche Features und Optimierungen zu einem späteren Zeitpunkt in die CHOCOFORMER App eingebaut werden könnten. Diese Listen beinhalten sowohl Features, welche in dieser Bachelorarbeit nicht mehr umgesetzt werden konnten, als auch Ideen, welche vom Development Team oder dem Maintainer-Team erwähnt wurden.

Die Features, welche aus Sicht des Entwickler-Teams den grössten Nutzen bringen, sind: Alle Konfigurationen können über das Admin-Front-End vorgenommen werden, das Benutzer-Front-End erhält mehr Formatierungsoptionen und die Applikation wird besser geschützt. Zusätzliche Details zu diesen Features befinden sich in den nachfolgenden Unterkapiteln.

8.5.1 BENUTZER-FRONT-END

- Freihand-Zeichnen könnte mit mehreren Pinselgrössen ausgestattet werden. Eine dickere Linie könnte dann dazu führen, dass der CHOCOFORMER den Pfad mehrfach abfahren muss. Dies verlängert die Druckzeit, ist jedoch die flexibelste Lösung. Alternativ könnten mit einem solchen Feature auch unterschiedliche Düsen angesprochen bzw. selektiert werden.
- Für die Text-Eingabe sind mehr Formatierungsoptionen möglich, z.B. fette bzw. kursive Schriften, mehrere Schriftarten und freie Positionierung des Texts auf der Zeichenfläche. Für die Unterstützung von Schriftarten müssen die Schriftart-Dateien möglicherweise in OpenSCAD installiert werden.
- Bei runden Einlegebauteile wäre eine Formatierungsoption denkbar, bei welcher der Text gebogen entlang der Kante verläuft. Die Herausforderungen dieses Features liegen darin, dass das CAD Text verformen muss und vor allem, dass das Eingabefeld im Front-End eine entsprechende Vorschau anzeigt und gleichzeitig editierbar bleibt.
- Als alternative Eingabemöglichkeit könnten auch Bilder verwendet werden. Diese müssten mit einer Webcam aufgenommen werden. Die grösste Herausforderung an diesem Feature ist, dass ein farbiges Bild auf zwei Farben reduziert werden muss, so dass dieses immer noch erkennbar ist.
- Es könnten auch vom Administrator vorgefertigte Sujets verwendet werden. Der Benutzer könnte dann diese Schriften und Bilder auswählen und frei auf der Zeichenfläche positionieren.

8.5.2 ADMIN-FRONT-END

- Die Druckaufträge könnten mit einer Vorschau bestätigt oder abgelehnt werden. Diese Funktion wäre besonders für Ausstellungen mit Schülern gedacht, da dort eher unangebrachte Sujets erwartet werden könnten. Mit einer solchen visuellen Vorschau könnten Druckaufträge entsprechend entfernt werden.
- Aus dem Dashboard könnten verschiedene Drucker-Befehle gesendet werden. Eine Option wäre, dass der Administrator beliebige G-Code-Befehle oder zumindest Macros aufrufen kann. Für den G-Code existiert bereits das Duet Web Control. Falls diese Funktionalität benötigt wird, muss wahrscheinlich sowieso das DWC verwendet werden. Für den Aufruf von Macros müsste ein Ablauf entworfen werden, welcher es erlaubt, die Macros einfacher aufzurufen als über das DWC oder das Display am CHOCOFORMER.
- Aus dem Dashboard könnten vorbereitete 3D-Modelle gedruckt werden. So könnte der Administrator ein 3D-Modell vollständig vorbereiten, die G-Code-Datei hochladen und drucken lassen. Mit der Funktionalität könnten die Möglichkeiten des CHOCOFORMERs besser demonstriert werden, allerdings schränkt dies die persönlichen Drucke ein, da grössere 3D-Modelle viel Zeit beanspruchen können. Um solche Drucke in die bestehende Applikation einzubetten, empfiehlt es sich, diese Druckaufträge in die Queue einzureihen und damit einen Grossteil der Infrastruktur wiederzuverwenden.
- Der aktuelle Schokoladen-Füllstand wird angezeigt. Der Administrator weiss momentan nicht, wieviel Schokolade noch in den Druckköpfen vorhanden ist. Während der Betreuung eines Messestandes kann die Nachfüllung schnell einmal vergessen werden. Daher wäre es äusserst nützlich, wenn der Administrator benachrichtigt wird, wenn die Schokolade zu Neige geht. In der ursprünglichen Form wurde eine solche Anzeige für diese Arbeit angedacht. Ein entsprechendes GUI ist dafür in der Datei PrinterMonitoring.tsx vorbereitet, jedoch fehlt die Applikationslogik und die Anbindung an diese. Die Berechnung müsste wahrscheinlich als Schätzung implementiert werden. Einige Daten könnten zudem aus der vom Slicer generierten G-Code-Datei gelesen werden.

8.5.3 KONFIGURATION

- Die benötigten Konfigurationen können mit der in dieser Bachelorarbeit erstellten CHOCOFORMER App via Konfigurationsdateien angepasst werden ohne dass dabei die Applikation neu kompiliert werden muss. Für einige dieser Konfigurationen wäre es wünschenswert, wenn diese über den Konfigurationsbildschirm des Admin-Front-Ends bearbeitet werden könnten. Insbesondere wichtig ist die Konfiguration der Druckköpfe für den Slicer. Beide Druckköpfe benötigen unterschiedliche Slicer-Konfigurationen bzw. das entsprechende Slicer-Profil, falls mit einer einzelnen Konfigurations-Datei für beide Druckköpfe gearbeitet werden soll.
- Als Alternative zum vorherigen Punkt wäre eine weitere Option, dass die Konfigurationsdateien automatisch neu eingelesen werden, sobald diese verändert werden. Bislang müssen die Applikationen neu gestartet werden, wenn eine manuelle Änderung vorgenommen wird.
- Mit einem Button könnte die Konfiguration getestet werden. Wenn der Administrator auf diesen Button klickt, werden das CAD und der Slicer mit einem vorbereiteten, statischen Modell

ausgeführt. Falls diese Schritte erfolgreich sind, so kann der Administrator davon ausgehen, dass die vorgenommenen Konfigurationen stimmen.

- Die maximale, erlaubte Druckzeit könnte angegeben werden. Wenn der Administrator ein hohes Besucheraufkommen feststellt, so könnte die Grösse des Sujets limitiert werden, um damit mehr Besuchern zu ermöglichen, einen persönlichen Druck zu gestalten. Eine solche Limitierung kann bedeuten, dass der Benutzer zum Beispiel nur eine bestimmte Länge an Strichen zur Verfügung hat oder dass die bemalte Fläche einen bestimmten Wert nicht überschreiten darf.

8.5.4 WEITERE MÖGLICHKEITEN

- Da die Applikation teilweise unbeaufsichtigt betrieben werden kann, muss das Administrations-Front-End mit einem Passwort geschützt werden. Bislang ist ein rudimentärer Schutz eingebaut, indem die beiden Front-End-Bereiche nur über separate Ports erreichbar sind.
- Wenn die Applikation weiter ausgebaut wird und aus dem Internet erreichbar sein soll, ist es wichtig, dass die Applikation entsprechend geschützt wird. Dafür müssen Zertifikate eingerichtet werden, damit die Applikation über HTTPS betrieben werden kann.

9 PROJEKTMANAGEMENT – PLANUNG/SOLL

9.1 PROZESSMODELL

Als Vorgehensmodell wird in dieser Bachelorarbeit Scrum in einer leicht abgeänderten Version verwendet. Zum einen ist der (Stellvertretende) Product Owner nicht im Team und erscheint nur zu den wöchentlichen Meetings, bei Fragen ist er jedoch per Mail erreichbar. Zum anderen hat eine Bachelorarbeit ein hartes Ende, sprich die Arbeit endet nach 16 Wochen. Dies ist unüblich für Scrum, da normalerweise immer weiter iteriert wird und es kein Ende gibt.

Diese Entscheidung beruht auf erfolgreichen Anwendungen in vergangenen Projekten und bereits vorhandenem Wissen. Zudem eignet sich Scrum bei diesem Projekt, da die Aufgabenstellung im komplexen Bereich liegt und Komponenten für das Development Team unbekannt sind: Details und Funktionen sind zu Beginn noch unbekannt und Features müssen erprobt werden.

Beratung und Unterstützung erfolgt über das zukünftige Maintainer Team. Da nur eine geringe Erfahrung in Software Engineering Bereich vorhanden ist, soll eine möglichst einfache wartbare Lösung gefunden werden.

Die Sprints haben eine Länge von 2 Wochen und beginnen jeweils am Montag. Jeden Montag findet das Sprint Review statt, einmal nur das Development Team und einmal mit allen Verantwortlichen des Projekts. Somit entfällt die Suche nach einem zusätzlichen, regelmässig wiederkehrenden Termin, was mit bis zu sieben Teilnehmern eine anspruchsvolle Aufgabe ist.

Das Daily Scrum wird grundsätzlich um 8 Uhr durch das Development Team durchgeführt. Bei Terminüberschneidungen durch Lektionen an der Hochschule, wird das Daily Scrum verschoben.

9.2 TEAM, ROLLEN UND VERANTWORTLICHKEITEN

In der nachfolgenden Tabelle sind die Rollen und Verantwortlichkeiten sowohl vom Bachelorarbeitsteam als auch den involvierten Mitarbeitern aus den Instituten aufgelistet:

Rolle	Name
Product Owner	Daniel Schwendemann
Product Owner Stellvertretung	Patrick Fässler
Scrum Master	Marc Weber
Development Team	Vanessa Janknecht, Marc Weber
Projektleitung CHOCOFORMER	Patrick Fässler
Projektleitung Stellvertretung CHOCOFORMER	Rahel Hasler
Ansprechperson Software/Elektronik	Marco Rossi
Maintainer Team	Patrick Fässler, Rahel Hasler, Marco Rossi

9.3 AUFWANDSCHÄTZUNG

Der Aufwand einer Bachelorarbeit wird auf 360 Stunden pro Studenten geschätzt. Somit stehen 720 Stunden für diese Arbeit zur Verfügung. Dieser Aufwand verteilt sich über die gesamten 14 Wochen des Semesters, hinzu kommen zwei Wochen nach Ende des Semesters, welche ausschliesslich für die Bachelorarbeit gedacht sind. Damit ergibt sich einen durchschnittlichen Aufwand von 22.5 Stunden pro Woche und Student.

Von diesem zur Verfügung stehenden Zeitbudget wird nach Abschätzungen der HSR davon ausgegangen, dass 50% dieser Zeit für Dokumentation verwendet wird.

9.4 ZEITPLAN

Die zeitliche Planung wird über GitLab gestaltet. Die Planung ist agil.

Es gibt folgende festgelegte Termine in der Planung:

- **22.05.2020:** Feature Freeze
- **29.05.2020:** Semesterende. Abgabe an IWK ist durchgeführt.
- **05.06.2020:** Abgabe Abstract und A0-Poster an Betreuer.
- **10.06.2020 bis 10 Uhr:** A0-Poster per E-Mail an das Studiengangsekretariat senden.
- **12.06.2020 bis 12 Uhr:** Hochladen aller verlangten Dokumente. Abgabe des Berichts an den Betreuer. Präsentation und Ausstellung der Bachelorarbeiten, 16 bis 20 Uhr

Dazu kommen die folgenden Termine, welche während dem Semester festgelegt wurden:

- **06.04.2020 ab 13 Uhr:** Zwischenpräsentation zusammen mit dem Betreuer, Experten und Gegenleser.
- **25.06.2020 ab 16 Uhr:** Mündliche Prüfung der Bachelorarbeit

9.5 MEILENSTEINE

Datum	Meilenstein	Beschreibung
M1 15.03.2020	Anforderungen und Architektur sind spezifiziert und abgenommen	Anforderungen sind abgeklärt, zu erstellende Komponenten sind identifiziert.
M2 29.03.2020	Schnittstelle zwischen Applikation und CHOCOFORMER ist funktionsfähig	Schnittstelle / Kommunikation mit dem Drucker funktioniert aus der Backend-Applikation heraus.
M3 19.04.2020	Druck-Prozess ist vollständig automatisiert	Der Druck-Prozess ist durchgehend abgebildet, von der Erstellung eines einfachen Modells (in Form eines einfachen Texts) bis zum Druck. Ein sporadisches GUI existiert, mit welchem ein End-to-end-Test zusammen mit dem Product Owner durchgeführt wurde.

M4 03.05.2020	Benutzerfreundliches GUI ist lauffähig	Die Benutzeroberfläche ist anschaulich und vom Product Owner abgenommen und kann für Messeauftritte verwendet werden.
M5 12.06.2020	Produkt / Dokumentation ist fertiggestellt	Die formelle Übergabe an das Maintainer-Team ist durchgeführt und das Produkt ist lauffähig.

Die Meilensteine werden am Montag nach Abschluss des Sprints im Sprint-Review vom (Stellvertretenden) Product Owner abgenommen, mit Ausnahme des Meilensteins M5, diese Abnahme geschieht spätestens am Freitag den 12.06.2020.

9.6 PROJEKTPLAN

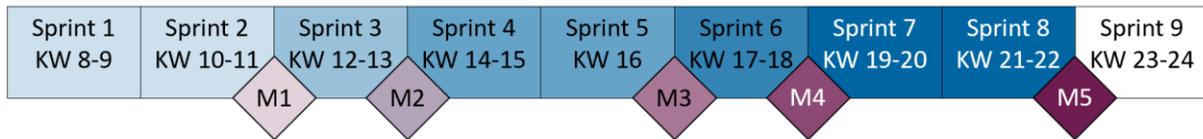


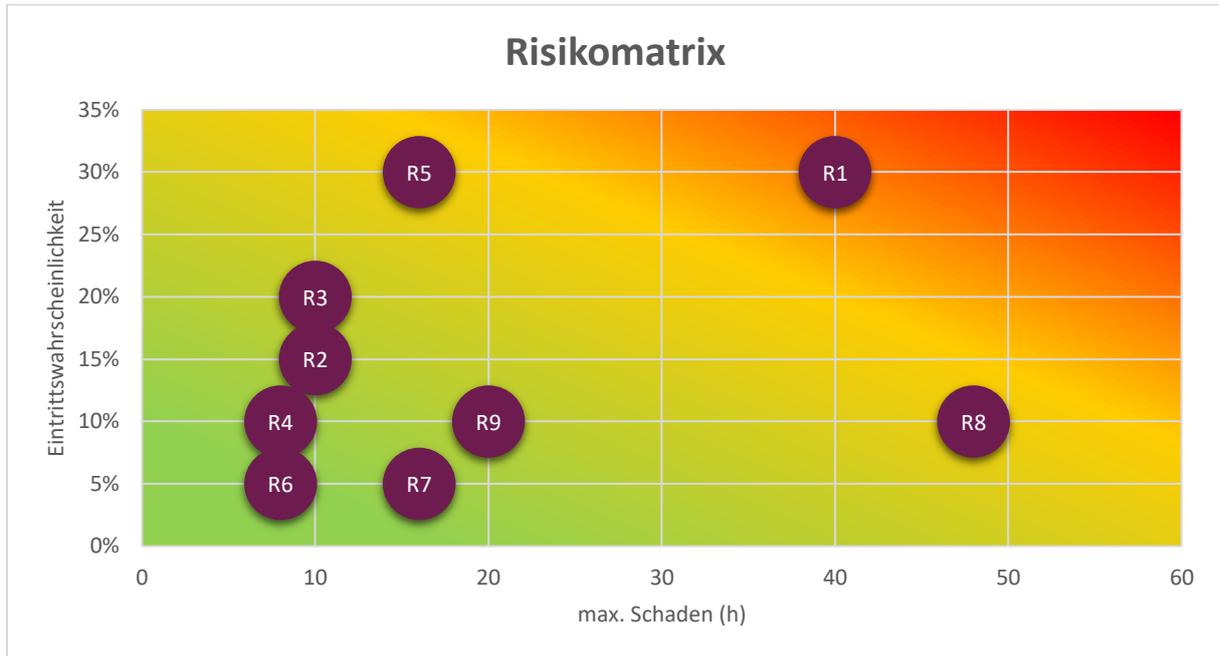
Abbildung 10 Zeitplanung mit Meilensteinen

Sprint 1	Start: 17.02.2020 Ende: 01.03.2020	<ul style="list-style-type: none"> Fragenkatalog ist erstellt Anforderungen sind geklärt Einarbeitung mit Duet ist geschehen Organisatorisches zur Abgabe ist geklärt Erster Architektur-Entwurf ist erstellt Risikobewertung ist erstellt Zeitplan ist erstellt Meilensteine sind vorerst festgelegt
Sprint 2	Start: 02.03.2020 Ende: 15.03.2020	<ul style="list-style-type: none"> Weitere Tests mit Duet 3 sind durchgeführt Anforderungen / Meilensteine / Zeitplan / Architektur / generell die Dokumentation mit Betreuer und Maintainer Team sind besprochen und angepasst Erster Entwurf des GUI Designs ist erstellt und besprochen <p>M1 Anforderungen und Architektur sind spezifiziert und abgenommen</p>
Sprint 3	Start: 16.03.2020 Ende: 29.03.2020	<ul style="list-style-type: none"> 3D-Drucker für Tests zusammenbauen und in Betrieb nehmen CI/CD einrichten VS Projekt aufsetzen Schnittstelle implementieren <p>M2 Schnittstelle zwischen Applikation und CHOCOFORMER ist funktionsfähig</p>
Sprint 4	Start: 30.03.2020 Ende: 12.04.2020	<ul style="list-style-type: none"> Rudimentäres Front-End erstellen Kommunikation von Front-End bis Drucker möglich

		<ul style="list-style-type: none"> • Robustheit überprüfen • Zwischenpräsentation • WebSocket-Kommunikation läuft • Konfigurationsdateien Handhabung implementiert
Sprint 5	Start: 13.04.2020 Ende: 19.04.2020	<ul style="list-style-type: none"> • SVG-Datei aus Frontend extrahieren • CI/CD erweitern • WorkflowService erstellen • Testabdeckung • Konfiguration für Slicer und CAD Komponente erweitern <p>M3 Druck Prozess ist vollständig automatisiert</p>
Sprint 6	Start: 20.04.2020 Ende: 03.05.2020	<ul style="list-style-type: none"> • Datenablage erstellen • Styling • Optionale Features? • Admin Front-End • Robustheit <p>M4 Benutzerfreundliches GUI ist lauffähig</p>
Sprint 7	Start: 04.05.2020 Ende: 17.05.2020	<ul style="list-style-type: none"> • Erster Versuch am CHOCOFORMER • Admin Dashboard • Installations Script erstellen • Debian Package erstellen • Code Refactor des Front-Ends
Sprint 8	Start: 18.05.2020 Ende: 31.05.2020	<ul style="list-style-type: none"> • <i>22.05.2020: Feature Freeze</i> • Zweiter Versuch am CHOCOFORMER • Admin Konfigurationen • Usability Test organisieren • Funktionalität erweitern: 2-farbig drucken • Übersetzung der Front-End auf Deutsch
Sprint 9	Start: 01.06.2020 Ende: 12.06.2020	<ul style="list-style-type: none"> • Usability Test durchführen und auswerten • Dokumentation abschliessen • Readme ergänzen • Installationsanleitung fertigstellen • Poster erstellen • Abstract schreiben <p>M5 Produkt / Dokumentation ist fertiggestellt</p>

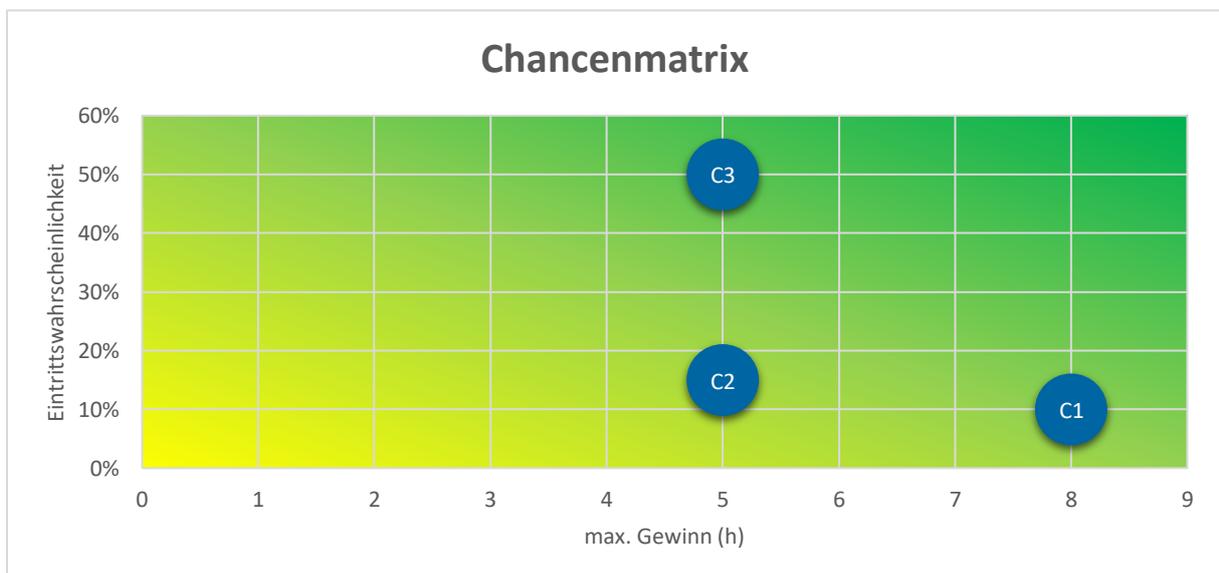
9.7 RISIKEN

Nr	Titel	max. Schaden (h)	Eintrittswahrscheinlichkeit	Verhalten beim Eintreten
R1	Fehlendes Know-How über die Hardware	40	30%	Mit den Mitarbeitern aus den Instituten eine gemeinsame Lösung bzw. Verständnis schaffen. Ggf. mit den Entwicklern des Boards Kontakt aufnehmen.
R2	Kommunikationsprotokoll ungeeignet	10	15%	Workarounds für das Problem erarbeiten oder herausfinden, ob sich ev. der Wechsel auf eine Alternative lohnt (Abwägung von Aufwand und Nutzen).
R3	Integration des Slicers funktioniert nicht wie geplant	10	20%	Auf eine andere Integrationsvariante wechseln.
R4	Planungsfehler	8	10%	Absprache halten. Features entsprechend umplanen.
R5	Material / Ansprechpersonen sind nicht verfügbar	16	30%	Mit Stellvertretung kommunizieren. Andere, pendente Aufgaben angehen.
R6	Konflikte im Development Team	8	5%	Das Problem muss sachlich und ruhig ausdiskutiert werden. Eventuell muss ein Mediator hinzugeschaltet werden.
R7	Datenverlust	16	5%	Auf die Backups bzw. den Stand vom anderen Teammitglied zugreifen.
R8	Ausfall eines Developers wegen dem Corona-Virus	48	10%	Absprache mit dem Maintainer Team, wie weiter verfahren wird. Je nach dem muss der Scope gekürzt werden.
R9	Schliessung der HSR wegen dem Corona-Virus.	20	10%	Alle wichtigen Gerätschaften werden mit nach Hause genommen. Die Sitzungen finden über Telefonkonferenz statt.



9.8 CHANCEN

Nr	Titel	max. Gewinn (h)	Eintritts-wahr-scheinlichkeit	Verhalten beim Eintreten
C1	Material steht länger zur Verfügung als geplant	8	10%	Aufgaben ggf. um-priorisieren, um die Hardware so länger benutzen bzw. testen zu können.
C2	Features können einfacher genutzt werden	5	15%	Die zusätzliche Funktionalität implementieren.
C3	Duet3-Board anstelle des Duet2-Boards verwenden	5	50%	



9.9 ELIMINIERTEN RISIKEN

9.9.1 R3 - INTEGRATION DES SLICERS FUNKTIONIERT NICHT WIE GEPLANT

Die Integration des Slicers war einfacher als gedacht. Der PrusaSlicer kann einfach von der Kommandozeile angesprochen werden und läuft auf Linux. Dieses Risiko konnte somit eliminiert werden.

9.10 EINGETRETENE RISIKEN

9.10.1 R9 - COVID-19 QUARANTÄNE

Aufgrund der COVID-19 Pandemie musste die HSR den Präsenzunterricht ab dem 16.03.2020 bis voraussichtlich am 19.04.2020 komplett einstellen. Zudem war der Unterricht in der Woche vom 16.03.2020 bis zum 20.03.2020 eingestellt. In der Zeit durften auch die Bachelorarbeiten nicht betreut werden. Des Weiteren hat der Bund die Bevölkerung aufgefordert, das Haus möglichst nicht zu verlassen. Solange diese Massnahmen anhalten, werden alle Meetings und Präsentationen über Microsoft Teams abgehalten.

Durch die besondere Lage mussten sich alle umstrukturieren und von zu Hause arbeiten, dies hat einiges an Zeit gekostet, aber den Fortschritt der Arbeit nicht beeinträchtigt. Da Besprechungen nur noch über Telefonie möglich sind, dauern diese länger und das gemeinsame Debuggen im Code, wenn jemand Probleme hat, ist erschwert. Da bei dieser Bachelorarbeit mit spezieller Hardware gearbeitet wird, hat sich auch das Testen zu einer Herausforderung erschwert. Es wurde ein 3D-Drucker zur Verfügung gestellt. Dieser kann jedoch nur von einem Teammitglied betrieben werden, das andere Mitglied hat keinen Zugriff auf den Drucker.

UPDATE 08.04.2020:

Zu einem späteren Zeitpunkt soll auch am richtigen CHOCOFORMER getestet werden. Es ist noch nicht klar, wann und ob dies möglich ist, da eine andere Bachelorarbeit sich damit beschäftigt und noch stärkere Einschränkungen hat, da diese nur direkt im Labor durchgeführt werden kann und der Studierende momentan nicht ins Labor darf.

Die Massnahmen des Bundes gegen das Virus wurden bis zum 26.04.2020 verlängert.

UPDATE 09.04.2020

Der Antrag der Schulleitung, dass Studierende für ihre Bachelor- und Masterarbeiten weiterhin im Labor arbeiten dürfen, wurde vom Kanton abgelehnt. Somit kann an die Bachelorarbeit betreffend des neuen CHOCOFORMERs weiterhin nicht fortgesetzt werden.

Die Abgabe des Projektes wurde in Absprache mit den Stellvertretenden Product Owners auf spätestens den 12.06.2020 verschoben. Somit kann die Entwicklung der Applikation verlängert und der Test am CHOCOFORMER nach hinten geschoben werden. Der erste Test sollte bis spätestens am 15.05 durchgeführt werden. Betreffende Änderungen wurden im Projektmanagement angepasst.

UPDATE 04.05.2020

Der Umbau des CHOCOFORMERs auf das neue Duet 3 Board hat begonnen. Am 11.05. soll die Quarantäne gelockert werden. Für Arbeiten im Labor, muss man eine Bewilligung von der Schulleitung bekommen. Die Anfrage für eine Bewilligung muss diese Punkte enthalten: Grund, Studenten, Betreuer, Raum, Datum und Zeit auf Halbtage genau. In Absprache mit dem Maintainer-Team wurden die Daten 13.05., 20.05. und 27.05. für den ganzen Tag für Tests am CHOCOFORMER beantragt.

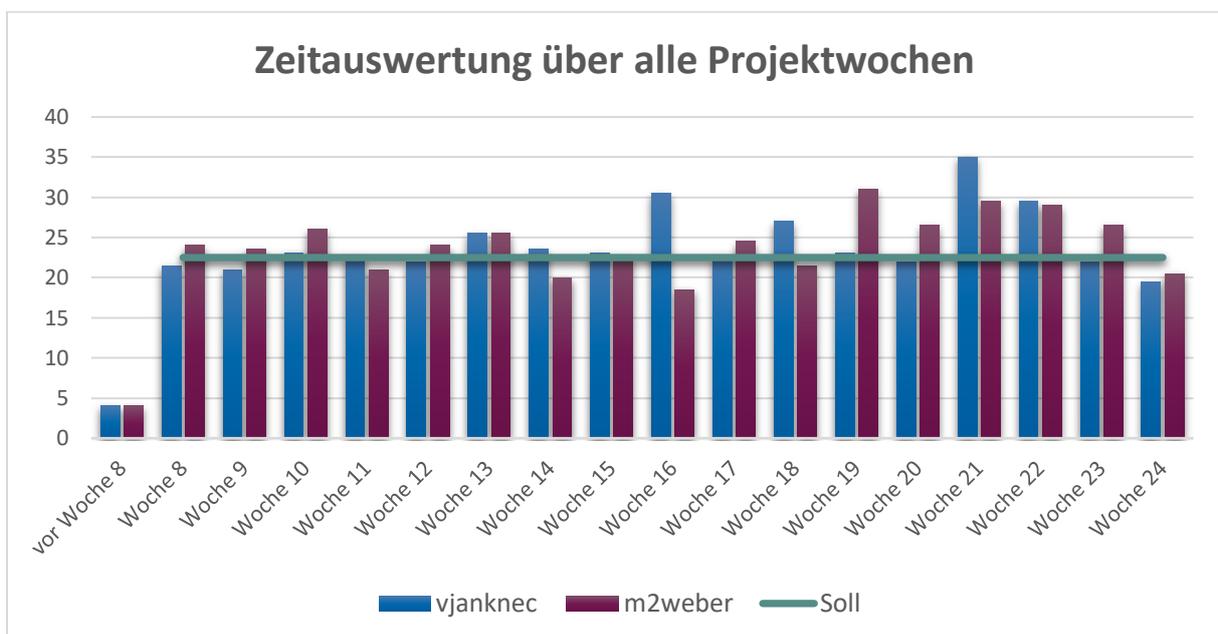
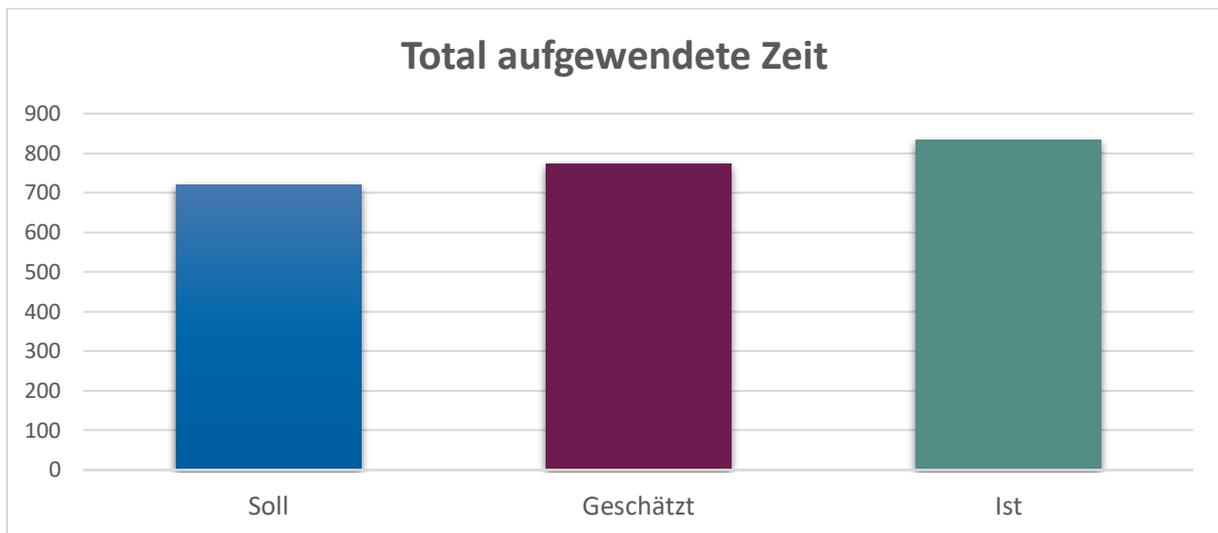
UPDATE 11.06.2020

Da der Stellvertretende Product Owner während den letzten drei Wochen der Bachelorarbeit nicht mehr anwesend war, kann die geplante Abnahme des letzten Meilensteins am 12.06. nicht wie geplant durchgeführt werden. Diese Abnahme kann zu einem späteren Zeitpunkt durchgeführt werden. Da die Applikation zweimal im Beisein des Maintainer-Teams getestet wurde, ist der grösste Teil der Implementation bereits bekannt. Über diesen Weg konnte das Entwickler-Team Feedback des Maintainer-Teams einholen.

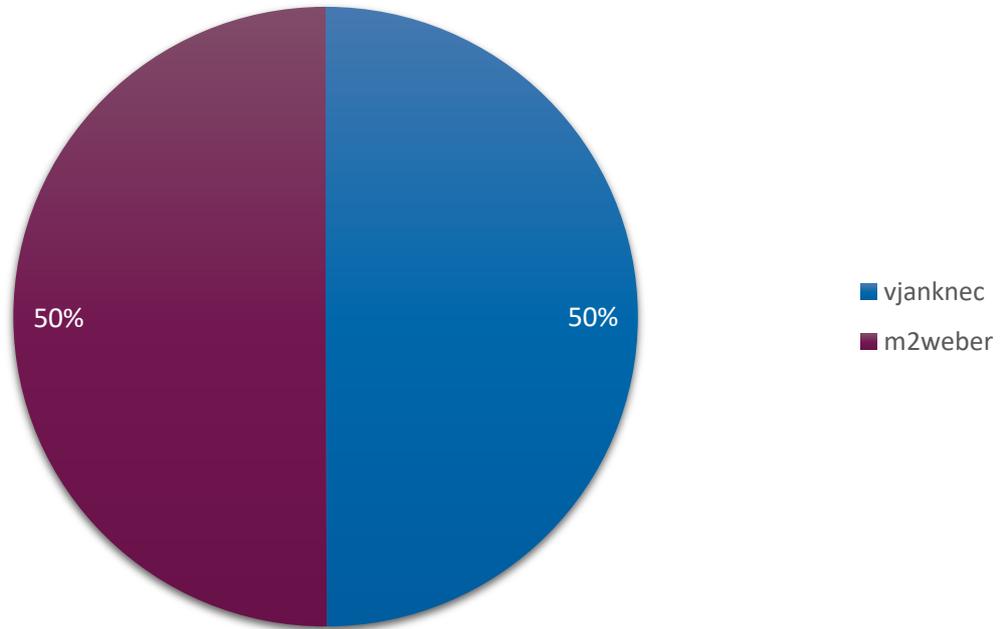
10 PROJEKTMONITORING - IST

10.1 SOLL-IST-ZEITVERGLEICH

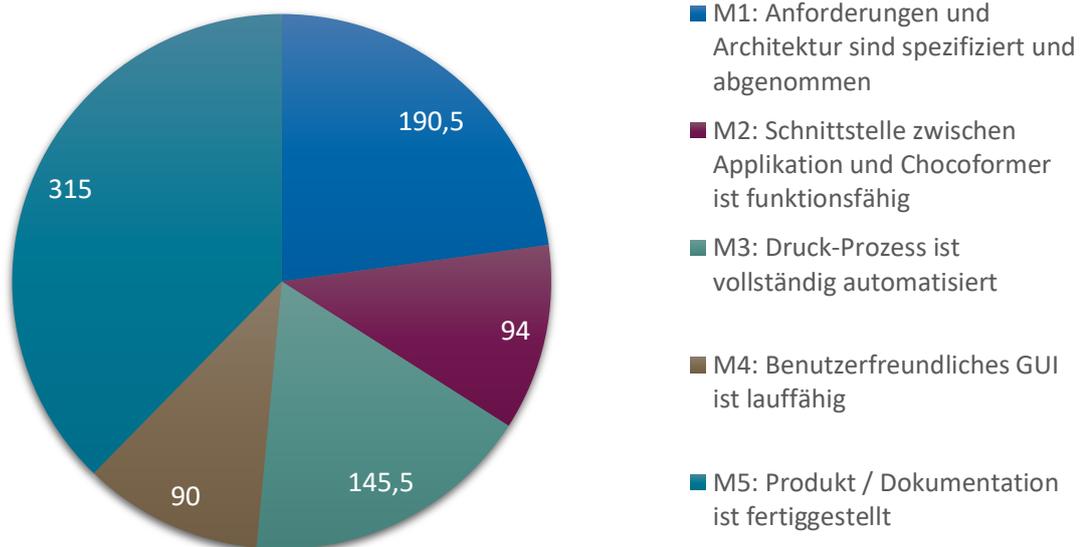
In den ersten Wochen konnte die Soll-Zeit gut eingehalten werden. Ab dem Lockdown wegen der Corona-Pandemie wurde von zu Hause gearbeitet. Daher gab es keine fixe Tagesstruktur mehr. Anstelle eines pünktlichen Feierabends wurden Implementationen noch am gleichen Tag versucht fertig zu stellen. Dadurch wurden am Abend immer häufiger zusätzliche Stunden geleistet. Auch an den Tagen vor und nach den Tests am CHOCOFORMER wurden zusätzliche Stunden geleistet, da die Möglichkeiten, am Drucker zu testen limitiert waren und es daher wichtig war, die Tests mit den geplanten Features durchzuführen.

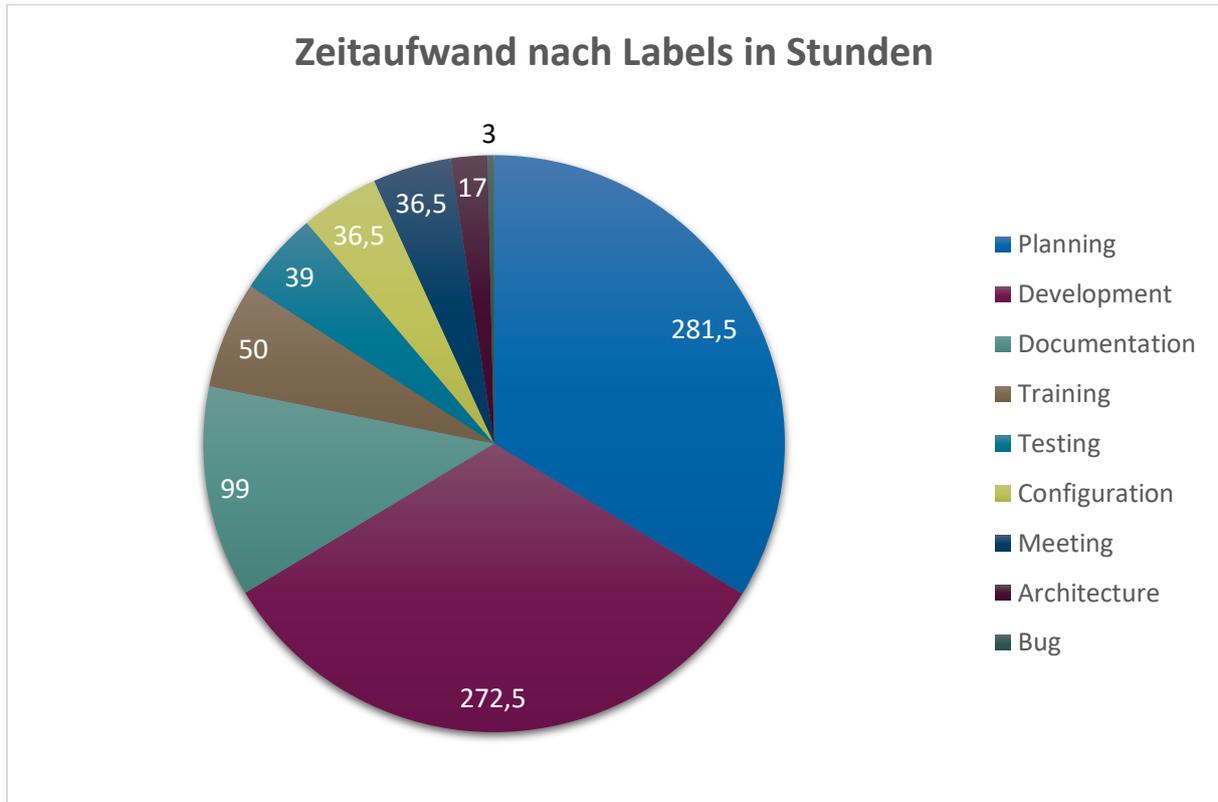


Aufteilung der Zeit zwischen Studenten



Zeitaufwand nach Meilensteinen in Stunden





Das Label Planning ist ein Sammellabel. Wenn das Entwicklerteam miteinander gearbeitet hat, wie z.B. Diskutieren, Beschlüsse fassen, Meetings vorbereiten, Dokumentation schreiben, Helfen beim Debuggen und Entwickeln, usw. wurde es schnell unübersichtlich, auf welches Issue die Zeit gebucht werden soll. Daher wurde für jeden Sprint ein Organisations-Issue erstellt, welches das Label Planning erhielt. Nur diese Organisations-Issues erhielten dieses Label. Immer wenn das Entwicklerteam zusammenge- arbeitet hat, wurde Zeit auf diese Issues gebucht.

10.2 CODESTATISTIK

10.2.1 VERTEILUNG DER PROGRAMMIERSPRACHEN

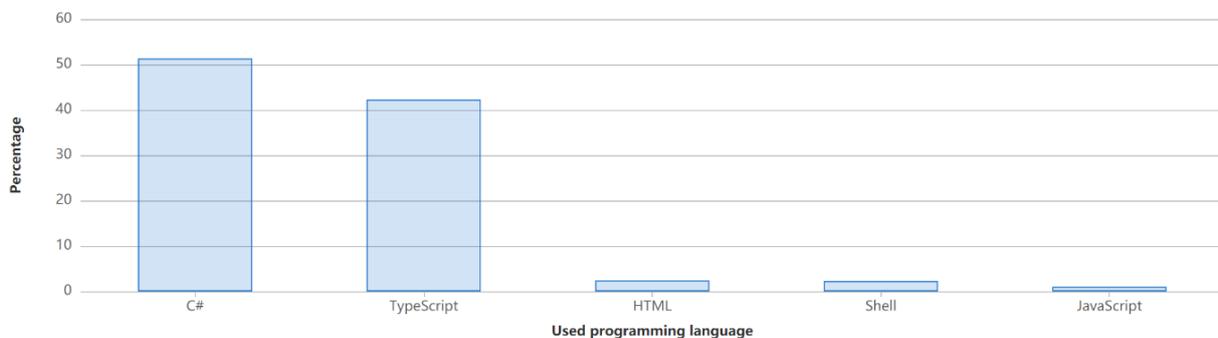


Abbildung 11 Auswertung der Programmiersprachen aus dem GitLab vom 10.06.2020

10.2.2 METRIKEN

Im C# wurden die Klassen, Interfaces und Funktionen gezählt.

	Klassen	Interfaces	Methoden
C# Back-End	22	5	38
C# Front-End	14	2	22
Total	36	7	60

Im TypeScript wurden die Klassen- und Funktions-Komponenten sowie die Methoden und Funktionen gezählt.

	Klassen-Komponenten	Funktions-Komponenten	Methoden / Funktionen
TypeScript	12	34	75

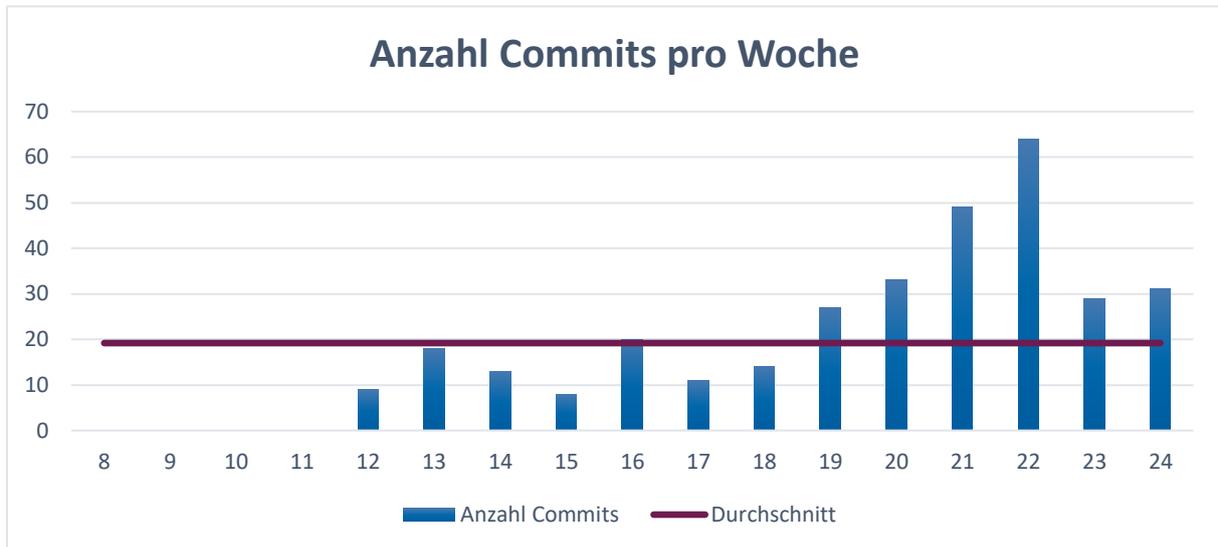
LINES OF CODE (LOC)

Hierarchy ^	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Source code	Lines of Executable code
▾ ■■ ApplicationLogic (Debug)	87	75	1	55	535	109
▾ ■■ ApplicationLogicTest (Debug)	83	32	1	27	357	101
▾ ■■ DataAccess (Debug)	92	126	1	37	619	106
▾ ■■ DataAccessTest (Debug)	86	16	1	13	133	50
▾ ■■ FrontEndAdmin (Debug)	80	131	4	123	633	148
▾ ■■ FrontEndAdminTest (Debug)	76	6	1	24	210	101
▾ ■■ FrontEndUser (Debug)	82	91	4	125	643	159
▾ ■■ FrontEndUserTest (Debug)	77	16	1	43	254	118

Abbildung 12 Code Metrics aus dem Visual Studio vom 10.06.2020

Programmiersprache	mit Tests	ohne Tests
C# Back-End	1'644	1'154
C# Front-End	1'740	1'276
TypeScript	3222	3222
Total	6606	5652

COMMITTS



Die Anzahl an Commits stieg im Laufe der Zeit an. Dies liegt einerseits daran, dass zu Beginn grössere Features implementiert wurden, welche als Ganzes committed wurden. In der späteren Projektphase wurden kleinere, dafür mehr einzelne Features implementiert. Dazu kommen sehr kleine Commits, welche insbesondere für die Tests am CHOCOFORMER notwendig waren und teilweise einzelne Zeilen änderten. In der letzten Woche lag die Anzahl der Commits immer noch über dem Durchschnitt. Zu diesem Zeitpunkt wurden aber keine neuen Features eingebaut, sondern der Code in kleinen Schritten aufgeräumt und die README-Datei mehrfach überarbeitet.

10.2.3 TESTABDECKUNG

Die durchschnittliche Testabdeckung für die Backend-Logik beträgt 66.12%. Die Testabdeckung für das Web-Service-Projekte beträgt 37.265%. Dabei ist anzumerken, dass die Berechnung Testabdeckung über den gesamten Code erfolgt, jedoch nur der C#-Code getestet wurde. Es hat fast doppelt so viele Zeilen TypeScript im Front-End wie C# im Back-End.

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
Surfy_DESKTOP-DTI1MR6 20...	671	30.38%	1538	69.62%
applicationlogic.dll	86	31.62%	186	68.38%
applicationlogictest.dll	8	4.02%	191	95.98%
dataaccess.dll	103	36.14%	182	63.86%
dataaccessstest.dll	2	2.25%	87	97.75%
frontendadmin.dll	237	59.85%	159	40.15%
frontendadmintest.dll	0	0.00%	239	100.00%
frontenduser.dll	229	65.62%	120	34.38%
frontendusertest.dll	6	1.58%	374	98.42%

Abbildung 13 Testcoverage aus dem Visual Studio vom 10.06.2020

11 SOFTWAREDOKUMENTATION

11.1 FIRMWARE UPDATEN

Um die Firmware auf dem Raspberry Pi manuell upzudaten, müssen folgende Befehle auf der Konsole ausgeführt werden:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Diese Befehle installieren die letzte stabile Version der Firmware und des DSF. Möchte man jedoch die letzte nicht stabile Version installieren, muss man dies vorher in den Einstellungen des Pi ändern:

```
wget -q https://pkg.duet3d.com/duet3d.gpg
```

```
wget -q https://pkg.duet3d.com/duet3d-unstable.list
```

```
sudo mv duet3d.gpg /etc/apt/trusted.gpg.d/
```

```
sudo mv duet3d-unstable.list /etc/apt/sources.list.d/duet3d-unstable.list
```

```
sudo chown root:root /etc/apt/trusted.gpg.d/duet3d.gpg
```

```
sudo chown root:root /etc/apt/sources.list.d/duet3d-unstable.list
```

```
sudo rm /etc/apt/sources.list.d/duet3d.list
```

Für Produktionszwecke ist es empfohlen, eine stabile Version zu verwenden.

11.2 INSTALLATION UND KONFIGURATION

Die Dokumentation, wie die CHOCOFORMER App installiert, bedient und konfiguriert wird, ist aus dem README im master-Branch des Repositories auf dem GitLab [35] übernommen. Das README wurde auf Englisch erstellt und ist für diese Dokumentation nicht übersetzt worden.

11.2.1 GETTING STARTED WITH DEVELOPMENT

11.2.1.1 PREREQUISITES

To set up the development environment including its dependencies run the script `Scripts\Setup-Development-Environment.bat` if you are on Windows. .NET Core has to be installed manually according to the environment you want to use. It could be downloaded from Microsoft's website². To run the tests OpenSCAD³ has to be installed additionally.

² <https://dotnet.microsoft.com/download/dotnet-core>

³ <https://www.openscad.org/>

To install all dependencies for the two front ends, call `npm install` from both directories (`FrontEndUser/ClientApp/` and `FrontEndAdmin/AdminApp/` respectively).

11.2.1.2 BUILDING

The C# solution can be compiled by calling `dotnet build` or from an IDE that can open solution files (e.g. Visual Studio or Rider).

To ensure a consistent style in the front-end projects ESLint can be used. To check the project call `npm run lint` in each project.

11.2.1.3 TESTING

The tests can be started by calling `dotnet test` in the corresponding directory or by running them from the IDE.

Each testing project contains its own `config.json`. This file will overwrite the default configuration file for the automated tests. By adjusting this file in the testing projects various parameters can be changed, like the path to the external slicer application.

These configurations are written in such a way that no adjustments are necessary in order to run the tests.

Note that integration tests, which may call platform-dependent programs like the slicer application, are excluded in the GitLab pipeline. These tests have to be invoked manually by a developer.

11.2.1.4 RUNNING LOCALLY

On Windows both front ends can be started in IIS Express. By default, the user front end will be started at `https://localhost:44391/` and the administration front end will be available at `https://localhost:44397/`.

11.2.2 USING THE APP ON A RASPBERRY PI

11.2.2.1 PREPARING A NOTEBOOK FOR THE CHOCOFORMER APP

Use the following rules to present the CHOCOFORMER App to customers using a notebook or tablet. These rules help to provide the best possible user experience for potential customers.

- Provide a touch screen stylus. Drawing with fingers can be inaccurate and thus provides an unsatisfying experience. Using a stylus the user can draw with much more detail.
- Remove the keyboard from the tablet. Most tablets are able to do this by either folding the keyboard back or detaching it from the screen. By doing so the user will not be tempted to use the mouse pad which may not responds as expected.
- Configure the web browser properly.
 - In Google Chrome there is a feature to navigate backwards by swiping. As this feature can interfere with the drawing screen it should be turned off. To do so navigate to

`chrome://flags/#overscroll-history-navigation` and disable "Overscroll history navigation".

- In Mozilla Firefox to activate the on-screen keyboard Windows 10 has to be put into tablet mode. This mode can be activated from the notification bar.

11.2.2.2 USING THE ADMINISTRATION DASHBOARD

11.2.2.2.1 POP-UP

If the CHOCOFORMER changes its status from any state to idle a pop-up appears and tells that the CHOCOFORMER is ready. This usually means the administrator can change the insert and start a new print. Because the CHOCOFORMER changes its state if it changes tools, it is possible that the pop-up then also appears, but it should disappear quickly.

11.2.2.2.2 MANAGE CURRENT JOB

The current job has 2 functionalities:

- next print: deletes the current print (all data from that print job will be lost!) and reads the next print from queue.
- start: starts the current print, if current print contains no job, nothing will happen. Do not press this button if a print is running. If the CHOCOFORMER is already printing and this button is pressed, the CHOCOFORMER will discard the request and keeps printing.

11.2.2.2.3 ADDING NEW INSERTS

Inserts can be uploaded using the configuration screen in the administration front end. The files that have to be prepared are:

- A square picture with a transparent background as a preview of the insert for the user. Its size should be at least 500 x 500 pixels.
- A G-code file for measuring the insert.

These files will be stored using the same name in the `data/inserts/pictures/` and `data/inserts/mesh_grids/` folders. Other associated data is stored in the `data/inserts/inserts_data.json` file.

11.2.2.3 CREATING A RELEASE

During the typical development on feature-branches there are two stages performed in a GitLab pipeline: build and test. As soon as these changes are merged into the master branch a third stage will be available. It has to be triggered manually by performing the following steps:

1. On GitLab navigate to the Pipelines page in the CI / CD category⁴
2. The changes to be released should currently show two successful stages and one pending stage.
3. Click on the play button on the right to start the release process. This process will take about 5 minutes.
4. Download the release artifact using the additional download button on the Pipelines page. This is the file to use for the installation on DuetPi.

The release stage is configured in such a way that release artifacts will be deleted after 1 week. To keep these files either back them up in a safe place or the GitLab wiki. Or keep them by using the GitLab Pipelines themselves:

1. Click on the third green stage icon to open the release stage of the current pipeline.
2. Click on "Keep" under the area "Job artifacts" in the sidebar.

11.2.2.4 INSTALLING ON DUETPI

To install the CHOCOFORMER App on a DuetPi system download the release artifact from GitLab to the Raspberry Pi. Ensure that during the installation a working internet connection is available as some packages have to be downloaded from this script.

1. Download the release artifact from GitLab to the Raspberry Pi (e.g. the home folder `/home/pi/`) or copy it using SSH or FTP.
2. Extract the files from the release artifact using `unzip artifacts.zip`.
3. Set the execution permission for the script: `chmod +x Package/install.sh`
4. Run the script with root privileges: `sudo Package/install.sh`
5. After the script is successfully finished, ensure that both web applications are accessible from a browser. Both parts should be available using e.g. `http://chocoformer:8080/` for the user front end and `http://chocoformer:8088/` for the admin front end.

If the CHOCOFORMER App is not available after the installation try the following steps:

- View the log file. If there are any problems with e.g. the configuration these errors will be listed here. `cat /opt/chocoformer-app/app/ChocoformerApp.log`
- Check if the services were started properly: `systemctl status chocoformer-app-user.service` and `systemctl status chocoformer-app-admin.service`. If one of the services is inactive or was not started, try `sudo systemctl enable <service-name>` and `sudo systemctl start <service-name>`.
- Restart the Raspberry Pi. If the Linux kernel was updated it is possible that e.g. the CAD does not recognize these changes.

⁴ <https://gitlab.dev.ifs.hsr.ch/chocoformer/chocoformer-app/pipelines>

11.2.2.5 UPDATING ON DUETPI

The installation script provides an option to update the web applications of the CHOCOFORMER App. This just replaces the app itself. Configuration files have to be updated manually.

1. Download the release artifact from GitLab to the Raspberry Pi or copy it using SSH or FTP.
2. Extract the files from the release artifact using `unzip artifacts.zip`.
3. Set the execution permission for the script: `chmod +x Package/install.sh`
4. Run the script with the update option: `sudo Package/install.sh -u`
5. Copy the configuration files from the release artifact to `/opt/chocoformer-app/data/` manually. Ensure that each file contains the changes previously made on the device itself as well as changes that may be introduction in the new release artifact.

11.2.2.6 DATA

All data is stored in a specific folder on the Raspberry Pi. The data folder is located `/opt/chocoformer-app/data/` on the Raspberry Pi. This folder should only be changed if you know what you are doing. Missing files potentially crash the application. The data folder contains four subfolders:

- **inserts:** stores the data for inserts
 - **mesh_grids:** contains the g-code data for measuring the inserts.
 - **picture:** contains the pictures of the inserts shown in the user front end.
 - **inserts_data.json:** holds the data of all inserts. The display name, the size in mm, the API call to the picture, the mesh grids name and if the insert is currently in use. It is recommended to use the configuration site on the administrator website to change this data.
- **jobs:** stores the folder for all print jobs available. The print job folders are temporarily and will be deleted if no longer used.
 - **Jobs.json:** holds the data of all print jobs in the queue. The job id, the insert that is used, the printing time and the seconds left until the print starts.
- **slicer_config:** stores data regarding the slicer.
 - **start:** g-code that is executed before the actual print will happen.
 - **end:** g-code that is executed after the actual print happened.
 - **slicer-config-left.ini:** a slicer configuration file. Specific for the chocolate used on the left head of the printer.
 - **slicer-config-right.ini:** a slicer configuration file. Specific for the chocolate used on the right head of the printer.

11.2.2.7 CONFIGURATION

Important note: When editing the configuration files directly using a text editor the CHOCOFORMER App has to be restarted. Otherwise most changes will not come into effect or cause inconsistent behaviour. To restart the applications the following commands can be used:

```
sudo systemctl restart chocoformer-app-user.service
```

```
sudo systemctl restart chocoformer-app-admin.service
```

11.2.2.7.1 CONFIG.JSON

Configurations can be changed in the config.json file. This can change some behaviour and appearance in the user front end. Next will be shown all possible configurations and what they do. The config.json file can be found at `opt/chocoformer-app/app/config.json` on the Raspberry Pi. You should only edit this file if you know what you do.

CAD

- **AppName:** the CAD software location on the Raspberry Pi
- **Arguments:** CAD command-line arguments
- **ExtrusionHeight:** final print height in mm. Multiple of 0.6.

SLICER

- **AppName:** the slicer software location on the Raspberry Pi
- **Arguments:** slicer command-line arguments
- **Start:** g-code file location on the Raspberry Pi. Everything that happens before the print.
- **End:** g-code file location on the Raspberry Pi. Everything that happens after the print.
- **ConfigRightExtruder:** location on the Raspberry Pi of the slicer configuration for the right head. The file should be put at the location: `/opt/chocoformer-app/data/slicer_config/`.
- **ConfigLeftExtruder:** location on the Raspberry Pi of the slicer configuration for the left head. The file should be put at the location: `/opt/chocoformer-app/data/slicer_config/`.

DUET3D

- **Websocket:** DWC WebSocket URL
- **InstructionsUri:** URI where the g-code is sent to
- **StatusUri:** URI to get the printer status
- **MinHeadTemperature:** Minimal temperature the heads should have
- **MaxHeadTemperature:** Maximal temperature the heads should have

JOBS

- **TempFileLocation:** location on the Raspberry Pi where all the jobs are stored

INSERTS

- **pathToData:** the json file location on the Raspberry Pi where all inserts information is saved
- **pathToPicture:** folder location on the Raspberry Pi with all insert pictures
- **pathToMesh:** folder location on the Raspberry Pi with all mesh-grid g-code files.

DRAWING

- **Font:** the text will be printed in this font
- **fontSize:** size the text will be printed with
- **strokeWidth:** stroke width for the SVG drawing in mm. Multiply of 1.2

EXTRUDER

- **NameRightExtruder:** Right extruder name shown in front end. If this string is empty, the extruder is disabled. Can be used, if only one head is used.
- **NameLeftExtruder:** left extruder name shown in front end. If this string is empty, the extruder is disabled. Can be used, if only one head is used.
- **ColorRightExtruder:** right extruder SVG stroke color shown in front end
- **ColorLeftExtruder:** left extruder SVG stroke color shown in front end

11.2.3 STRUCTURE

The folder `Chocoformer App` contains the complete application itself. It includes the back-end project written in C# as well as the front ends implemented using React. The front-end projects are part of the corresponding services on the back-end side, e.g. the code of the front-end application for the administrator is in the project of the back-end service called `FrontEndAdmin`.

The other folders in the project are:

- **Binaries:** Contains external applications like PrusaSlicer and other binary files.
- **data:** Contains runtime data and configuration as described above.
- **Package:** Contains the files required for a release. Configuration and binary files for DuetPi and the installation script are stored in this repository. Other files (e.g. build output) will be copied here using GitLab.
- **Scripts:** Contains scripts for setting up the development environment and for testing the front end using SoapUI.

12 REFLEXION

12.1 MARC WEBER

Diese Bachelorarbeit hat mir sehr viel Spass gemacht, da die Aufgaben sehr interdisziplinär waren. Dazu gehörte alles von der Gestaltung einer Web-Oberfläche über die Implementation von Logik bis hin zur Konfiguration eines Raspberry Pis und dem Zusammenbauen eines kleinen 3D-Druckers für die ersten Tests.

Die Zusammenarbeit mit der Teampartnerin lief sehr gut, da wir oft miteinander kommuniziert haben und so Probleme frühzeitig lösen konnten. Um die Aufteilung der Meetings-Leitung fair zu gestalten, haben wir uns bei jedem Meeting abgewechselt, was gut funktioniert hat. Der zwischenmenschliche Aspekt hat gut funktioniert, auch wenn wir zuvor noch kein grösseres Softwareprojekt zusammen bearbeitet haben.

Auch die Zusammenarbeit mit den Projektpartnern war sehr spannend. Mit uns aus dem IT-Bereich waren Personen aus drei unterschiedlichen Bereichen zusammengekommen, um an einem Gesamtprodukt zu arbeiten. Die Arbeit hat besonders viel Freude gemacht, da die Zusammenarbeit reibungslos funktioniert hat. Die Projektpartner waren sehr zuverlässig und hilfsbereit bei all unseren Anliegen.

Ich bin auch froh, dass die Arbeit während der Corona-Pandemie gut weitergeführt werden konnte. Die Anforderungen und die Planung mit den Projektpartnern konnten wir bereits vor dem Lockdown abschliessen. Während des Lockdowns fiel der grösste Teil an Programmieraufgaben an. Dies war auch von Vorteil, da wir uns grösstenteils nur noch zu zweit absprechen mussten. Die Zusammenarbeit untereinander war auch dann noch intensiv und hat sehr gut funktioniert. Wir hatten uns täglich abgesprochen und uns gegenseitig bei Fragen unterstützt.

Rückblickend muss ich feststellen, dass diese Applikation ein sehr grosses Projekt ist. Wir konnten das Grundgerüst implementieren und es läuft bereits so, dass die Applikation an Messeständen eingesetzt werden kann. Jedoch sind noch viele Aufgaben offen, gerade in den Bereichen Qualitätssicherung und Konfiguration, da die Zeit für einige Punkte leider nicht mehr ausreichte. Dennoch bin ich der Überzeugung, dass wir für unsere Arbeit einen guten Mix aus benötigten Features und Testing gefunden haben.

12.2 VANESSA JANKNECHT

Die Bachelorarbeit war das grösste Softwareprojekt, dass ich bis dahin umgesetzt habe. Auch wenn die Arbeit anspruchsvoll und manchmal auch stressig war, hat mir die Arbeit an diesem Projekt sehr viel Spass gemacht.

Besonders gefallen hat mir, dass sich die Arbeit nicht nur um reine Software gedreht hat, sondern wir mit einer Maschine arbeiten durften. Dies brachte auch neue und andere Probleme mit sich, die gelöst werden mussten. Das Debugging war ab und zu sehr knifflig, da wir erst verstehen mussten wie sich der Drucker verhält und was er an Informationen zurückgibt. Auch waren wir so gut wie nicht eingeschränkt, wie wir das Projekt umsetzen sollen. Der Product Owner hat uns erzählt, was er sich vorstellt, wie das Ganze umgesetzt werden soll, war uns überlassen. Zudem durften wir auch eigene Ideen mit

einbringen. So war das Administrator Front-End unsere Idee, um den Betrieb des CHOCOFORMERS möglichst bequem zu gestalten. Auch welche Technologien zum Einsatz kamen, war uns überlassen.

Einige der Technologien, die für dieses Projekt verwendet wurden, kannte ich bereits, andere waren mir vollkommen neu. Es ist immer wieder spannend, etwas Neues kennen zu lernen. Vor allem die Kommunikation mit dem Raspberry Pi und dem Duet Board war vollkommen neu für mich.

Dadurch, dass wir so viele Verantwortlichkeiten hatten, wir waren nicht nur Entwickler, sondern auch die Teamleiter, haben ich bezüglich Teamführung und Projektplanung viel dazu gelernt. Auch wenn am Anfang viele Dinge noch etwas chaotisch abliefen, da wir uns erst an unsere Rolle gewöhnen mussten, wurde es im Verlauf des Projekts immer besser.

Leider haben wir etwas zu spät bemerkt, dass einige Features sehr viel mehr Zeit benötigten, als wir zu Beginn dachten. Vor allem die Queue, die zu Beginn nicht spezifiziert, aber später vom Product Owner sehr gerne gewünscht wurde. Daher blieben einige andere Features und auch das Testen auf der Strecke. Trotzdem bin ich sehr zufrieden mit unserem fertigen Produkt. Es ist eine fertige Applikation, die an Messen eingesetzt werden kann. Auch finde ich, ist die Gestaltung der Applikation uns gut gelungen, da dies für öffentliche Auftritte auch sehr wichtig ist. Zudem konnten wir die Applikation so vorbereiten, dass sie für die nächsten Entwickler einfach zu erweitern ist.

Da an diesem Projekt mehrere Institute der HSR beteiligt waren, bekamen wir eine andere Sichtweise auf die Probleme, als wir sie sonst aus der IT kannten. Das ganze Maintainer-Team war unglaublich freundlich und es hat gut miteinander funktioniert. Wir konnten uns auf unsere Ansprechpersonen verlassen und hatten insgesamt eine sehr harmonische Arbeitsbeziehung. Auch die Zusammenarbeit mit meinem Projektpartner lief ausgesprochen gut. Es gab keine Streitigkeiten und wir konnten Probleme und Entscheidungen sachlich diskutieren. Ebenso konnten wir uns gegenseitig helfen, wenn der andere einmal nicht weiterkam. Als die Corona-Krise ausbrach und wir alle nur noch von zu Hause arbeiten durften, hat sich die Zusammenarbeit nicht verschlechtert. Die Kommunikation war manchmal etwas schwierig, wenn das Mikrofon zu weit weg war oder es Störungen im Netzwerk gab, aber hat alles in allem sehr gut funktioniert. Bezüglich zwischenmenschlicher Interaktionen konnte ich mir kein besseres Projekt vorstellen.

13 LITERATUR- UND QUELLENVERZEICHNIS

[1]	«Duet Software Framework,» [Online]. Available: https://github.com/chrishamm/DuetSoftwareFramework . [Zugriff am 7. März 2020].
[2]	«Duet Web Control,» [Online]. Available: https://github.com/chrishamm/DuetWebControl . [Zugriff am 7. März 2020].
[3]	«G-Code,» [Online]. Available: https://duet3d.dozuki.com/Wiki/Gcode . [Zugriff am 12. März 2020].
[4]	«Macros,» [Online]. Available: https://duet3d.dozuki.com/Wiki/Macros . [Zugriff am 7. März 2020].
[5]	«ChocEdge,» [Online]. Available: http://chocedge.com/home.html . [Zugriff am 12. März 2020].
[6]	«Semantic Versioning,» [Online]. Available: https://semver.org/spec/v2.0.0.html . [Zugriff am 6. März 2020].
[7]	«EventWaitHandle,» [Online]. Available: https://docs.microsoft.com/en-us/dotnet/standard/threading/eventwaithandle . [Zugriff am 26. Mai 2020].
[8]	«Presentation Domain Data Layering,» [Online]. Available: https://martinfowler.com/bliki/PresentationDomainDataLayering.html . [Zugriff am 11. März 2020].
[9]	M. F. & D. Rice, Patterns of Enterprise Application Architecture, Boston: Addison-wesley, 2014.
[10]	«PrusaSlicer,» [Online]. Available: https://www.prusa3d.com/prusaslicer/ . [Zugriff am 18. Februar 2020].
[11]	«Ultimaker CURA,» [Online]. Available: https://ultimaker.com/software/ultimaker-cura . [Zugriff am 18. Februar 2020].
[12]	«Pathio,» [Online]. Available: https://pathio.xyz/ . [Zugriff am 18. Februar 2020].
[13]	«Craftbot CraftWare,» [Online]. Available: https://craftbot.com/craftware/ . [Zugriff am 18. Februar 2020].
[14]	«Simplify3D,» [Online]. Available: https://www.simplify3d.com/ . [Zugriff am 18. Februar 2020].
[15]	«OpenSCAD,» [Online]. Available: https://www.openscad.org/index.html . [Zugriff am 24. März 2020].
[16]	«openscad-raspberrypi,» [Online]. Available: https://github.com/koendv/openscad-raspberrypi . [Zugriff am 24. März 2020].
[17]	«GitLab Wiki - OpenSCAD on Raspberry Pi Build Notes,» [Online]. Available: https://gitlab.dev.ifs.hsr.ch/chocofomer/chocofomer-app/-/wikis/Guides/OpenSCAD-on-Raspberry-Pi-Build-Notes . [Zugriff am 9. Juni 2020].
[18]	«Duet3D Dokumentation,» [Online]. Available: https://duet3d.dozuki.com/ . [Zugriff am 4. März 2020].

[19]	«RepRap Firmware,» [Online]. Available: https://reprap.org/wiki/RepRap . [Zugriff am 4. März 2020].
[20]	«Apache2.0,» [Online]. Available: https://www.apache.org/licenses/LICENSE-2.0.txt . [Zugriff am 4. Juni 2020].
[21]	«Choose a License,» [Online]. Available: https://choosealicense.com/licenses/apache-2.0/ . [Zugriff am 4. Juni 2020].
[22]	«Open Source Guide - License,» [Online]. Available: https://opensource.guide/legal/#which-open-source-license-is-appropriate-for-my-project . [Zugriff am 3. Juni 2020].
[23]	«MICROSOFT SOFTWARE LICENSE TERMS - MICROSOFT .NET LIBRARY,» [Online]. Available: https://www.microsoft.com/web/webpi/eula/net_library_eula_enu.htm . [Zugriff am 3. Juni 2020].
[24]	«MICROSOFT VISUAL STUDIO 2015 ADD-ONS, VISUAL STUDIO SHELLS and C++ REDISTRIBUTABLE,» [Online]. Available: https://www.typescriptlang.org/License.html . [Zugriff am 3. Juni 2020].
[25]	«Raspberry Pi Security,» [Online]. Available: https://www.raspberrypi.org/documentation/configuration/security.md . [Zugriff am 2. Juni 2020].
[26]	«PrusaSlicer Fehlermeldungen,» [Online]. Available: https://github.com/prusa3d/PrusaSlicer/blob/0b629eb905cbd8e95d2edc86ae797880e01b315d/src/PrusaSlicer.cpp . [Zugriff am 10. April 2020].
[27]	«OpenSCAD Fehlermeldungen,» [Online]. Available: https://github.com/openscad/openscad/blob/3756d1367de6e4ec6cf8f395bf8b3d396220af12/src/openscad.cc . [Zugriff am 10. April 2020].
[28]	«Inversion of Control Containers and the Dependency Injection pattern,» [Online]. Available: https://martinfowler.com/articles/injection.html . [Zugriff am 4. Mai 2020].
[29]	«Material UI,» [Online]. Available: https://material-ui.com/ . [Zugriff am 29. Mai 2020].
[30]	«Material Design,» [Online]. Available: https://material.io/ . [Zugriff am 29. Mai 2020].
[31]	«HSR Corporate Design (nur mit HSR-Login erreichbar),» [Online]. Available: https://intranet.hsr.ch/Farben.7715.0.html?&L=0 . [Zugriff am 29. Mai 2020].
[32]	«OST Ostschweizer Fachhochschule,» [Online]. Available: https://www.ost.ch/ . [Zugriff am 29. Mai 2020].
[33]	«C# Coding Conventions,» [Online]. Available: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions . [Zugriff am 11. Juni 2020].
[34]	«Host and deploy ASP.NET Core,» [Online]. Available: https://docs.microsoft.com/en-us/aspnet/core/host-and-deploy/?view=aspnetcore-3.1#set-up-a-process-manager . [Zugriff am 15. Mai 2020].
[35]	«Anzahl Testpersonen für Usability Tests,» [Online]. Available: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ . [Zugriff am 1. Juni 2020].

[36]	«GitLab CHOCOFORMER App master,» [Online]. Available: https://gitlab.dev.ifs.hsr.ch/chocoformer/chocoformer-app/-/tree/master . [Zugriff am 11. Juni 2020].
[37]	«Getting_Started_With_Duet_3,» [Online]. Available: https://duet3d.dozuki.com/Wiki/Getting_Started_With_Duet_3 . [Zugriff am 12. März 2020].
[38]	«Duet_3_Mainboard_6HC_Wiring_Diagram,» [Online]. Available: https://duet3d.dozuki.com/Wiki/Duet_3_Mainboard_6HC_Wiring_Diagram . [Zugriff am 12. März 2020].
[39]	«Duet Software Framework API,» [Online]. Available: https://github.com/Duet3D/DuetSoftwareFramework#rest-api . [Zugriff am 11. Juni 2020].

14 VERZEICHNISSE

14.1 ABBILDUNGEN

Abbildung 1 Use Case Diagramm	14
Abbildung 2 Ablauf und Interaktionen eines Druckauftrags	19
Abbildung 3 C4 Kontext-Diagramm für den bisherigen Zustand des CHOCOFORMERs (vorher).....	22
Abbildung 4 C4 Kontext-Diagramm für den CHOCOFORMER mit der CHOCOFORMER App (nachher)	23
Abbildung 5 C4 Container-Diagramm für die CHOCOFORMER App	24
Abbildung 6 C4 Komponenten-Diagramm für das CHOCOFORMER App Back-End	25
Abbildung 7 Vergleich Anzahl Testpersonen und Anzahl gefundener Usability Probleme [32].....	50
Abbildung 8 Aufbau für den Test am CHOCOFORMER	51
Abbildung 9 Erreichte vs. nicht erreichte Use Cases.....	61
Abbildung 10 Zeitplanung mit Meilensteinen.....	67
Abbildung 11 Auswertung der Programmiersprachen aus dem GitLab vom 10.06.2020	75
Abbildung 12 Code Metrics aus dem Visual Studio vom 10.06.2020	76
Abbildung 13 Testcoverage aus dem Visual Studio vom 10.06.2020	77
Abbildung 14 Duet 3 Mainboard 6HC - Wiring Diagram v1.0 [34].....	101
Abbildung 15 Skizze der Neuordnung der Elemente im Administrator Dashboard	113

15 ERKLÄRUNG ZUR URHEBERSCHAFT

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne Hilfe Dritter angefertigt habe. Ich habe nur die Hilfsmittel benutzt, die ich angegeben habe. Gedanken, die ich aus fremden Quellen direkt oder indirekt übernommen habe, sind kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

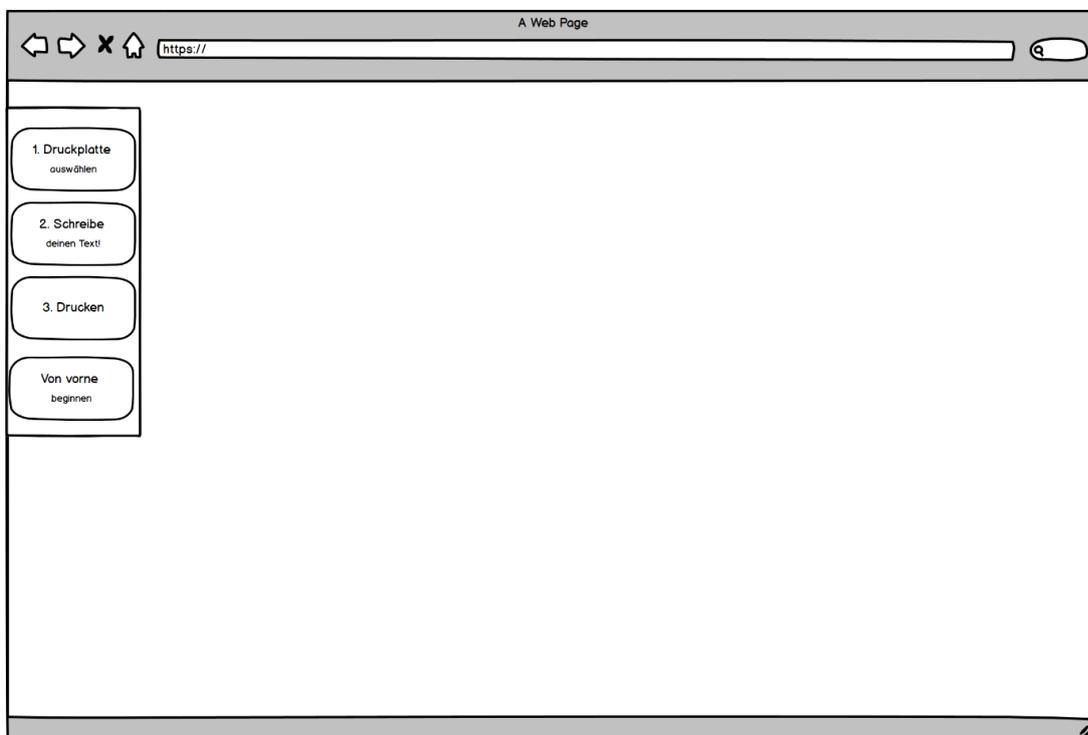
16 ANHANG

16.1 WIREFRAMES

Die Wireframes wurden mit dem Tool «Balsamiq Wireframes»⁵ erstellt. Balsamiq ermöglicht den Einsatz von «Clickable Wireframes», um einen ersten interaktiven Eindruck der Applikation erlauben.

16.1.1 WIREFRAMES BENUTZER FRONT-END

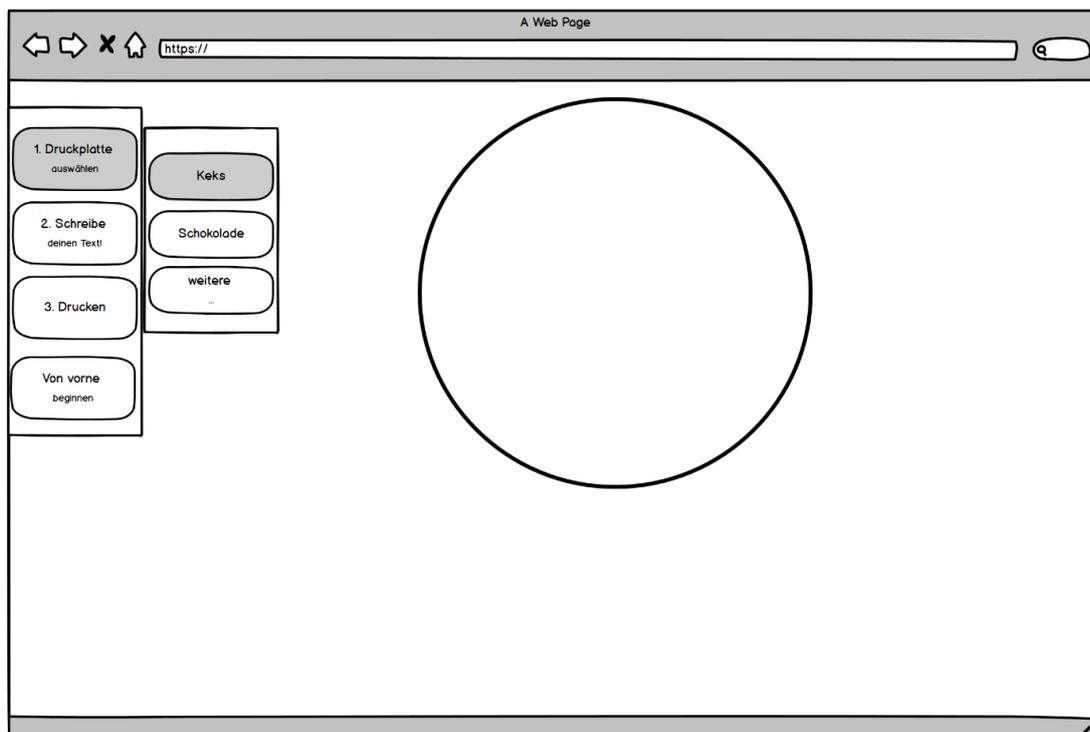
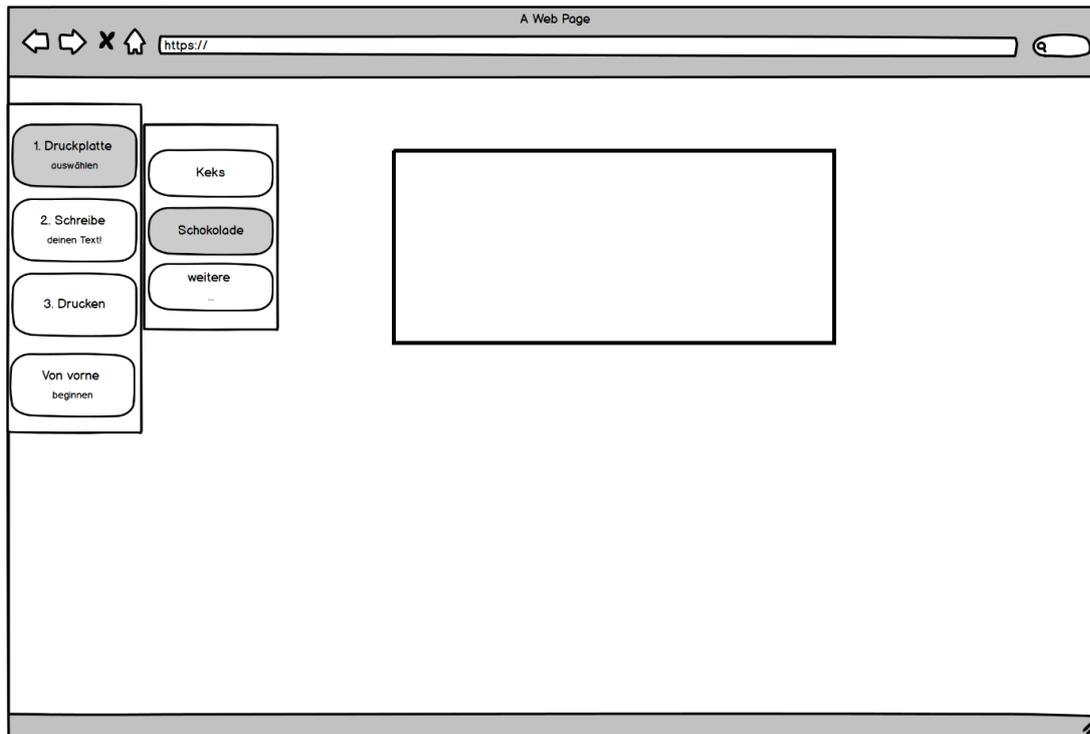
Zu Beginn befindet sich der Benutzer auf einer leeren Seite mit einer Toolbar. Evtl. wird für eine schönere Benutzeransicht der Drucker aus der Vogelperspektive angezeigt, dies ist jedoch optional.



⁵ https://balsamiq.com/?gclid=EAlaIQobChMlp9eO6sni6AIVCeR3Ch1L0wQ5EAAYAiAAEgKli_D_BwE

Druckplatte wählen

Wird der Button «1. Druckplatte» gedrückt, öffnet sich eine Auswahl mit verschiedenen Druckplatten, aus denen sich der Benutzer eine aussuchen kann. Die Druckplatte erscheint anschliessend auf dem Bildschirm als Referenz.

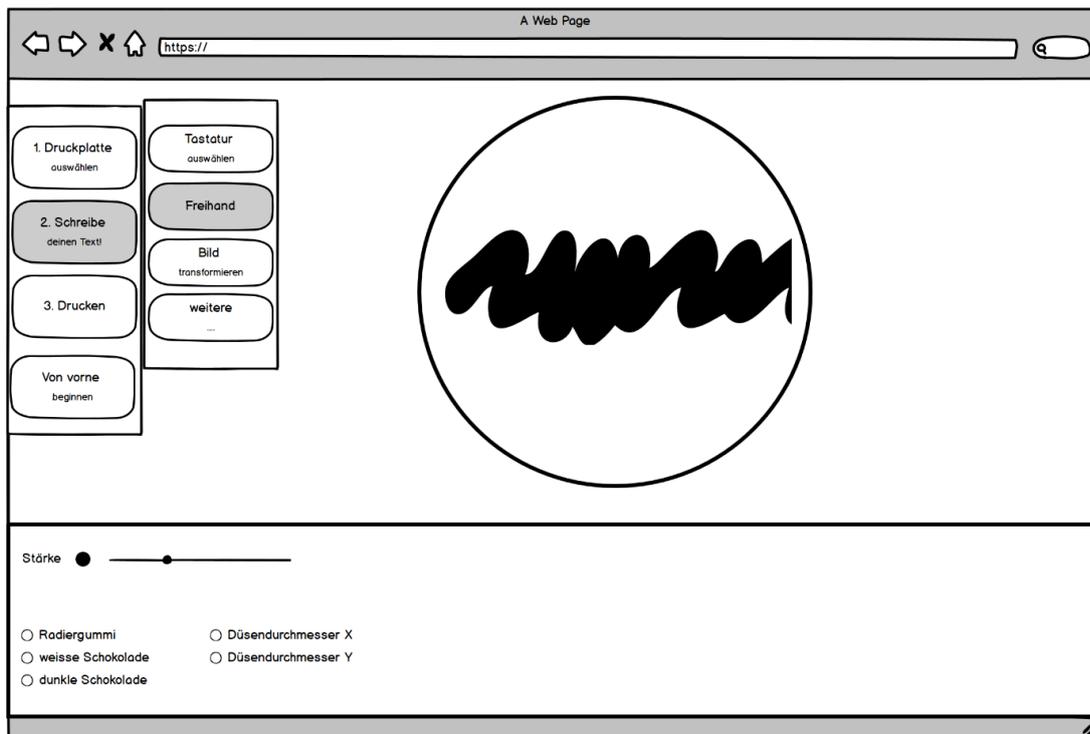


Grus / Text schreiben

Mit einem Klick auf den Button «2. Schreiben» öffnet sich eine Auswahl, aus welcher der Benutzer eine Option auswählen kann.

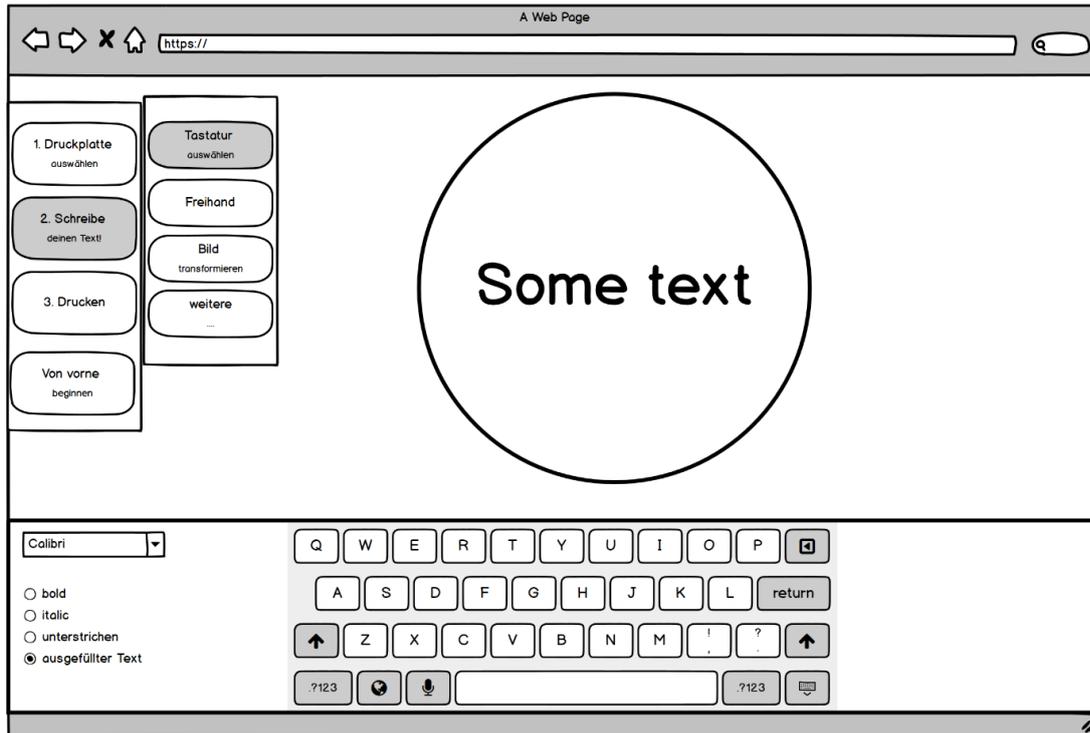
Freihand

Mit einem Klick auf Freihand, wechselt die Anwendung in den Freihand-Modus. Der Benutzer kann jetzt mit seinem Finger oder einem geeigneten Stift für Touch-Panels seinen Text verfassen. Dabei wird er auf die Form beschränkt, die er zuvor ausgewählt hat. Dabei stehen ihm verschiedenen Optionen in einem Fenster am unteren Rand des Bildschirms zur Verfügung.



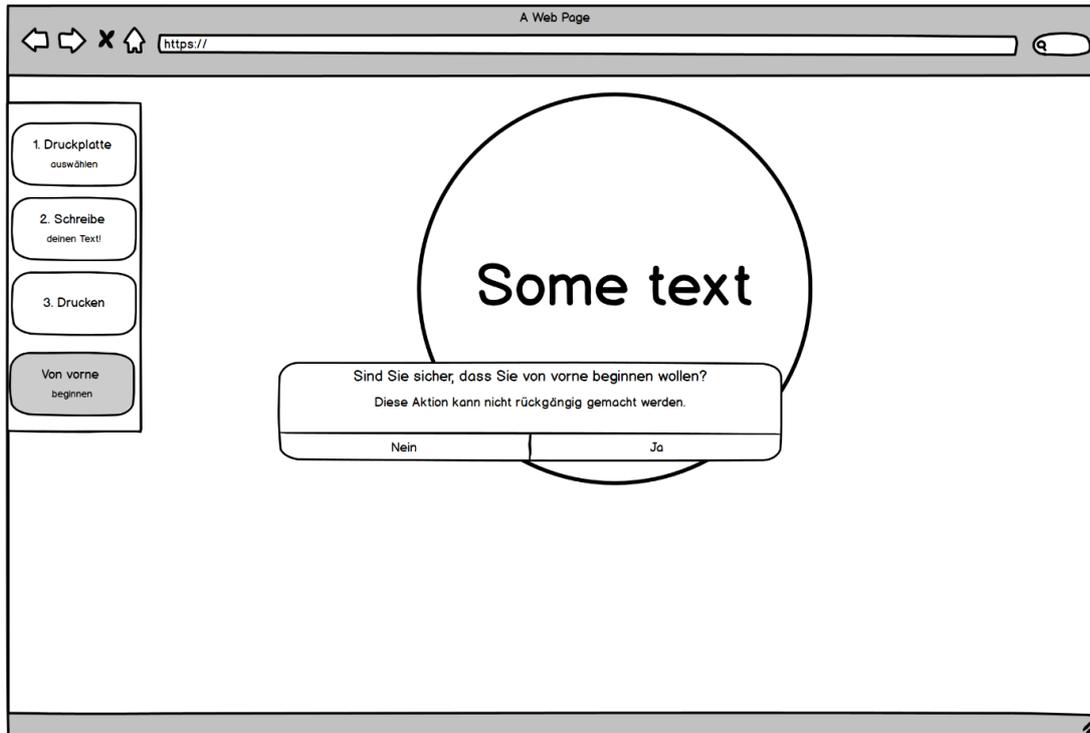
Tastatur

Wählt der Benutzer Tastatur, wechselt die Anwendung in den Tastatur-Modus. Dabei erscheint am unteren Bereich des Bildschirms eine Tastatur, mit der der Benutzer seinen Text schreiben kann. Es stehen auch verschiedene Optionen zur Veränderung des Texts zur Verfügung. Das Textfeld wird entsprechend der Druckplatte, die im ersten Schritt gewählt wurde, angepasst.



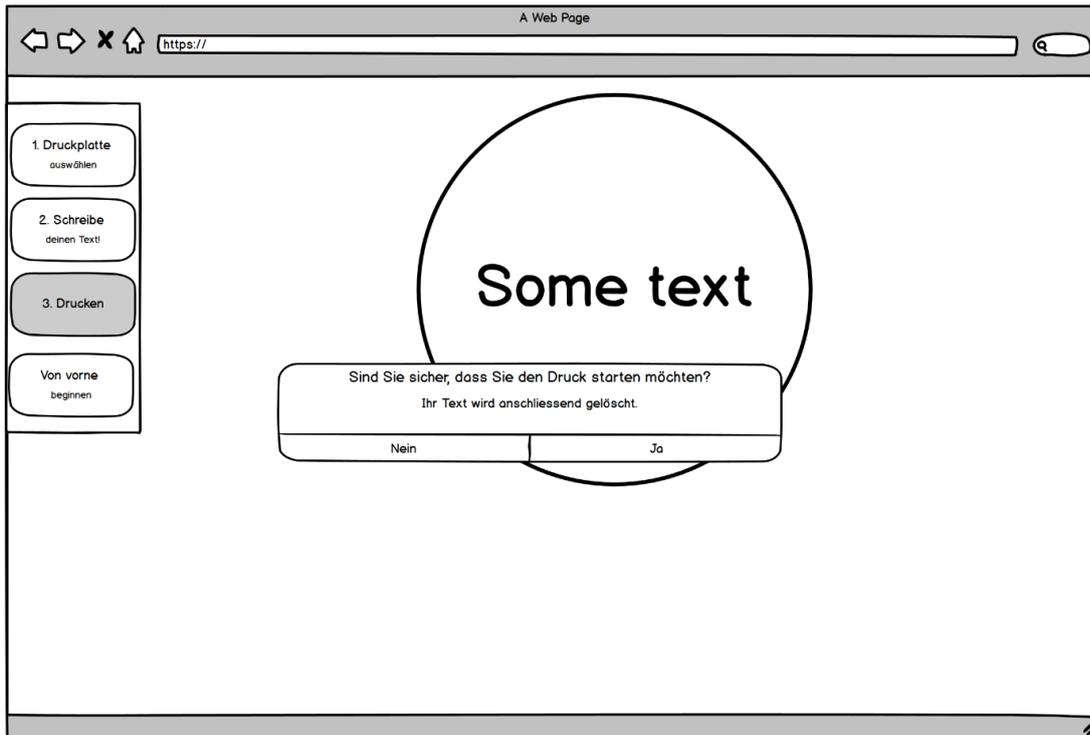
Löschen

Der Button «Von vorne beginnen» öffnet ein Textfeld mit der Benachrichtigung ob der Benutzer seine Änderungen wirklich löschen möchte.



Drucken

Wird der Button «3. Drucken» betätigt, erscheint ein Textfeld, das abfragt ob der Benutzer sein Sujet wirklich drucken möchte und dass dieses im Anschluss gelöscht wird. Wird dieser Button betätigt, wird der Druck am CHOCOFORMER gestartet. Der Benutzer kann nun ein weiteres Sujet erstellen oder die Applikation für die nächste Person freigeben.



16.1.2 WIREFRAMES ADMINISTRATOR FRONT-END

Der Administrator verfügt über zwei Ansichten. In der einen Ansicht kann er die Optionen für die Benutzeransicht festlegen. Mit einem Klick auf «Verwalten» öffnet sich ein Fenster, in dem der Administrator die entsprechende Spalte verändern kann, sprich Dateien löschen oder hinzufügen.

A Web Page

https:// 🔍

Allgemein

Maximale Zeit für einen Druck 2 min
 Druckauftrag muss durch Admin freigeben werden

Druckköpfe

Schokolade	Ausgewählt
Weisse	<input type="checkbox"/>
Schokolade	<input type="checkbox"/>
Milch	<input type="checkbox"/>
Schokolade	<input type="checkbox"/>
Dunkle	<input type="checkbox"/>
Schokolade	<input type="checkbox"/>

Düsendurchmesser

Durchmesser 1

Durchmesser 2

Verwendete Köpfe

Linker Druckkopf

Rechter Druckkopf

Verwalten

Einlegebauteile

Einlegebauteil	Ausgewählt
Keks Rund	<input type="checkbox"/>
Keks Eckig	<input type="checkbox"/>
Schokolade	<input type="checkbox"/>

Verwalten

Schriftarten

Schriftarten	Ausgewählt
Schrift 1	<input type="checkbox"/>
Schrift 2	<input type="checkbox"/>
Schrift 3	<input type="checkbox"/>

Optionen

Unterstrichen

Kursiv

fett

nur Kontur

Text verbiegen möglich

Textfilter einschalten

Verwalten

Freihand

Freihand erlauben

Verwalten

3D-Modelle

3D-Modelle erlauben

Modell	Ausgewählt
HSR Logo	<input type="checkbox"/>
Modell 1	<input type="checkbox"/>
Modell 2	<input type="checkbox"/>
Logo 1	<input type="checkbox"/>
Logo 2	<input type="checkbox"/>

Verwalten

In der zweiten Ansicht erhält der Administrator Informationen über den laufenden Druck und die Druckaufträge, die in der Queue warten. Zudem erhält er Informationen über den CHOCOFORMER, wie Temperatur oder Geschwindigkeit. Der Administrator wird informiert, wenn ein Druck fertig ist und wenn ein neuer Druck gestartet werden muss. Wird ein neuer Druck gestartet, wird ihm mitgeteilt welches Einlegebauteil er in den CHOCOFORMER einsetzen muss.

A Web Page

https://
🔍

Druckerqueue

Druckauftrag	Einlegebauteil	Dauer	Zeit bis Start
1	Keks rund	5 min	5 min
2	Schoggi	3 min	10 min
3	Keks rund	3 min	13 min

Laufender Druck

Status: OK

Zeit bis fertig: 5 min

Einlegebauteil: Keks rund

Schokoladenmenge Links: 100 ml

Schokoladenmenge Rechts: 100 ml

Anzahl Durchgeführte Druckaufträge: 10

Drucker Informationen

Temperatur:

Geschwindigkeit:

etc... --> siehe DWC

16.2 GETTING STARTED WITH DUET 3 & RASPBERRY PI [33]

16.2.1 WAS GEBRAUCHT WIRD

- Duet 3 Board
- Netzteil für Duet 3 Board (min. 12V. max. 32V, empfohlen 24V)
- Raspberry Pi Model 3B oder 4B
- HDMI auf Micro-HDMI Kabel
- Flachbandkabel (26-pin to 40-pin) 10-15cm lang
- Bildschirm
- Maus
- Keyboard
- Ethernet Anschluss
- 3D-Drucker

16.2.2 AUFBAU DES 3D-DRUCKERS

In der nachfolgenden Grafik ist ersichtlich, welche Teile von aussen und damit auch für den Benutzer zu sehen sind:



1. Druckplatte, auf welcher das gedruckte Objekt entsteht
2. Druckköpfe mit der jeweiligen Schokolade, je einer Druckdüse und einem Distanzsensor
3. Touchscreen für direkte Steuerung des CHOCOFORMERS
4. Motoren für die horizontale Verschiebung der Druckköpfe
5. Achse für die vertikale Verschiebung der Druckplatte

Der Duet 3 und der Raspberry Pi werden mit dem Flachbandkabel verbunden. **ACHTUNG:** Das Kabel muss unbedingt richtig positioniert angebracht werden! Die Position des Kabels erscheint seltsam, Grund dafür ist, dass der Raspberry Pi unter- oder oberhalb des Duet 3 aufgehängt werden soll, damit die Anschlüsse alle auf gleicher Höhe sind. Siehe Abbildung unten.

Sobald der Duet 3 mit dem Strom versorgt wird, startet der Raspberry Pi und das Duet Web Control öffnet sich.

16.2.5 AUFSETZEN

Nachdem der Raspberry Pi vollständig hochgefahren wurde soll die Kommandozeile geöffnet werden. Dort werden folgende Befehle ausgeführt:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Damit wird die aktuelle Version der RepRap Firmware heruntergeladen. Das System ist nun einsatzbereit. Der 3D-Drucker kann über das Duet Web Control gesteuert werden.

16.3 VERSUCHSPLÄNE

16.3.1 VERSUCHSPLAN 13.05.2020 – AB 9:00 – 2.112

Ziele für den gesamten Versuch:

- Die auf dem Prusa-3D-Drucker getestete Applikation läuft auch auf dem Chocoformer.
- Installations-Skript wird durchgeführt und installiert die Services.
- Konfigurationen werden auf den Chocoformer abgestimmt.

Vorbedingungen für den gesamten Versuch:

- Binärdateien (Release-Build-Output inklusive Konfigurationsdateien, optional mit Installationskript) stehen über GitLab oder einen USB-Stick/Netzlaufwerk bereit.
- DWC auf DuetPi ist mit Chocoformer verbunden.
- Testgeräte haben eine Verbindung zum DuetPi.
- Konfigurations-Datei für Slicer ist vorbereitet.

16.3.1.1 GEPLANTE VERSUCHE

ID	Ziel	Vorbedingung	Input / Interaktion	Erwartetes Resultat	Geschätzte Dauer in Minuten
1	Installations-Skript ist funktionsfähig	DuetPi ist im Ursprungszustand installiert	Skript wird mit Binaries heruntergeladen. Skript wird ausführbar gemacht. Skript wird ausgeführt.	Alle Abhängigkeiten sind installiert. OpenSCAD und PrusaSlicer sind vorhanden. Chocoformer App ist vorhanden. Admin / User GUI und DWC können aufgerufen werden.	20

2	Service ist richtig installiert	Installations-Skript wurde ausgeführt	Chocoformer neu starten.	Admin / User GUI können immer noch aufgerufen werden.	5
3	Maschinen-Daten im Admin-Dashboard werden korrekt angezeigt	Siehe #2	Admin-Dashboard öffnen	Alle Daten werden in der "Printer Information" Card angezeigt.	1
4	Maschinen-Daten im Admin-Dashboard werden korrekt aktualisiert	Siehe #2	Admin-Dashboard öffnen Mind. 10 Sekunden warten auf Refresh Ev. Temperaturen am Chocoformer anpassen.	Die Daten stimmen mit den Werten am Chocoformer-Display überein. Die Daten werden automatisch aktualisiert.	10
5	Gesamter Ablauf eines Drucks (Use Case UC08) funktioniert	Benutzung des Benutzer-Front-Ends	Ein Druck wird gestartet	Homing Ausmessung der Platte Drucken	10
6	Rechter Druckkopf kann angesteuert werden	Benutzung des Benutzer-Front-Ends	Rechten Druckkopf wählen und etwas drucken	Rechter Druckkopf reagiert	10
7	Linker Druckkopf kann angesteuert werden	Benutzung des Benutzer-Front-Ends	Linken Druckkopf wählen und etwas drucken	Linker Druckkopf reagiert	45
8	Text schreiben und drucken (UC06.1)	<ul style="list-style-type: none"> Benutzung des Benutzer-Front-Ends Schokolade 	Etwas schreiben	Text erscheint im richtigen Größenverhältnis und innerhalb der Abgrenzung	5
9	Mit Freihand schreiben und drucken (UC06.2)	<ul style="list-style-type: none"> Benutzung des Benutzer-Front-Ends Schokolade 	Etwas zeichnen	Schrift erscheint im richtigen Größenverhältnis und innerhalb der Abgrenzung	5

10	Herausfinden wie dünn eine Linie minimal sein darf, damit sie noch schön gedruckt wird?	<ul style="list-style-type: none"> Schokolade DWC 	Testdatei laufen lassen	Muster zum Auswerten	5
11	Herausfinden wie dick eine Linie maximal sein darf, damit es noch als eine Linie vom interpretiert wird?	<ul style="list-style-type: none"> Schokolade DWC 	Testdatei laufen lassen		5
12	Wie klein darf eine Schriftart minimal sein, damit sie noch leserlich ist?	<ul style="list-style-type: none"> Schokolade DWC 	Testdatei laufen lassen		5
13	Extrusion Height (CAD) für einen schönen Druck herausfinden	<ul style="list-style-type: none"> DWC Schokolade 	Testdatei laufen lassen		5
14	Emergency Stop betätigen	Ein Druck-Vorgang läuft	Im Admin GUI wird der Emergency Stop ausgelöst		Der Chocoformer bricht ab.
TOTAL					133

16.3.1.2 VERSUCHSPROTOKOLL

ID	Tatsächliches Resultat	Einzuplanende Aktionen	Umsetzung	Erfolgreich?
1	Für die Installation muss genügend Zeit eingerechnet werden, insbesondere wenn die Verbindung mit DuetPi auf dem entsprechenden Versuchs-PC zuvor noch nicht eingerichtet wurde.	Das Script muss die Berechtigungen für OpenSCAD und PrusaSlicer setzen.	Berechtigungen sind im Script angepasst.	Teilweise

	<p>Den externen Applikationen (OpenSCAD und PrusaSlicer) fehlen die benötigten Berechtigungen.</p> <p>Die Applikation an sich wurde richtig installiert.</p>			
2	Wie erwartet.	-	-	Ja
3	Wie erwartet.	-	-	Ja
4	<p>Die Daten sind plausibel und werden automatisch aktualisiert. Jedoch sind die Werte der beiden Sensoren in einem einzelnen Druckkopf fast gleich bzw. haben eine Abweichung von ca. 0.1 °C. Dadurch ist es schwierig, die korrekte Zuordnung der Werte im GUI zu überprüfen. Dies ist aber nicht kritisch, da sich die beiden Werte gemeinsam verändern.</p>	-	-	Ja
5	Ein Pfad war noch falsch angegeben. Druck hat funktioniert nach Anpassung.	Pfad im config.json korrigieren	Pfad ist korrigiert.	Ja
6	Wie erwartet.	Rechter Druckkopf wird mit G-Code „T0“ angesteuert	-	Ja
7	Test konnte erst ausgeführt werden, nachdem dem Entwicklerteam mitgeteilt wurde, wie der linke Druckkopf genau per G-code angesteuert wird, da am Testdrucker nur ein Kopf verwendet wird.	Linker Druckkopf wird mit G-Code „T1“ angesteuert.	Druckprozess ist für zwei Druckköpfe angepasst.	Ja
8	Text wurde gedruckt. Rotation im CAD musste angepasst werden, da Druck zuerst um 180° gedreht gedruckt wurde.	Rotation der Datei im CADService anpassen	Rotation ist angepasst.	Ja
9	Zeichnung wurde gedruckt. Bezüglich Rotation siehe oben.			Ja
10	Diese Testfälle wurden bereits vom IWK durchgeführt. Die Ergebnisse wurden dem Entwicklerteam zur Verfügung gestellt.	-	-	Nicht ausgeführt, da
11		-	-	

12		-	-	Ergebnisse vorhanden
13		-	-	
14	Der Chocoformer bricht alle Bewegungen ab und startet neu. Da die Steuerung über DuetPi läuft, dauert dies länger, als wenn der Drucker unabhängig von DuetPi laufen würde.	-	-	Ja
Zusatz	Ein zusätzlicher Test für ein Demo-Video konnte nicht durchgeführt werden, da die eingefüllte Schokolade nicht mehr die notwendige Temperatur hatte. Dieses Problem ist unabhängig von der Chocoformer App aufgetreten.	Die Temperaturen im Admin-Dashboard könnten hervorgehoben werden, wenn der erforderliche Temperatur-Bereich verlassen wird. So könnten Probleme möglicherweise schneller erkannt werden.	Temperatur-Bereich und fehlerhafte Stati werden hervorgehoben.	-

16.3.1.3 AUSWERTUNG

Der erste Verbindungsaufbau zum Raspberry Pi dauerte länger als angenommen, da die entsprechenden Tools (PuTTY bzw. WinSCP) noch nicht richtig konfiguriert waren. Zudem ist die Installation von DuetPi normalerweise nicht mit dem Internet verbunden, wodurch der Download von Abhängigkeiten mit dem ersten Versuch fehlschlug.

Vorherige Testläufe mit dem Installations-Skript wurden in einer virtuellen Maschine mit einem 64-bit Debian durchgeführt, da das für ARM-Prozessoren optimierte DuetPi nicht in VirtualBox aufgestartet werden konnte. Da DuetPi auf Raspbian basiert, welches wiederum auf Debian basiert, war kein grosser Unterschied zu erwarten. Der einzige Unterschied, welcher mit dem oben beschriebenen Versuch festgestellt wurde, ist ein unterschiedliches Berechtigungsmodell. Diese Fehler konnten direkt während des Versuchs behoben werden, so dass die Applikation problemlos installiert bzw. ausgeführt werden konnte.

Die weiteren, geplanten Versuche konnten ohne grössere Probleme von Seiten der Chocoformer App durchgeführt werden. Der Grossteil der gefundenen Unstimmigkeiten waren auf falsche Konfigurationen im Slicer zurückzuführen. Diese Anpassungen wurden jeweils durch das Maintainer-Team vorgenommen, welches uns bei diesen Tests tatkräftig unterstützt hat.

16.3.2 VERSUCHSPLAN 27.05.2020 – AB 9:00 – 2.112

Ziele für den gesamten Versuch:

- Update der Applikation kann durchgeführt werden.
- Fehlerzustände und zusätzliche Buttons (Home All, ...) funktionieren.
- Die Queue mit den Druckaufträgen wird korrekt angezeigt und Druckaufträge können gestartet werden.
- Beide Druckköpfe können aus einem Job heraus angesteuert werden.

Vorbedingungen für den gesamten Versuch:

- Binärdateien (Release-Build-Output inklusive Konfigurationsdateien) stehen über GitLab oder einen USB-Stick/Netzlaufwerk bereit.
- DWC auf DuetPi ist mit Chocoformer verbunden.
- Testgeräte haben eine Verbindung zum DuetPi.
- Konfigurations-Datei für Slicer ist vorbereitet.

16.3.2.1 GEPLANTE VERSUCHE

ID	Ziel	Vorbedingung	Input / Interaktion	Erwartetes Resultat	Geschätzte Dauer in Minuten
1	Installations-Skript kann Update durchführen	DuetPi ist mit einer älteren Version der Chocoformer App installiert	Neuester Release auf DuetPi laden Update via Skript starten Daten & Config-Files manuell kopieren	Admin / User GUI können aufgerufen werden. Die neue Version der Chocoformer App wird angezeigt.	34
2	Fehlende Einträge in der Chocoformer App-Konfiguration führen zu einem Fehler	Update (#1) ist durchgeführt	Zusätzliche Zeile einfügen (muss am Ende dieses Tests wieder entfernt werden)	Die Chocoformer App startet nicht und zeigt einen Fehler im Log an.	5

3	Admin-Dashboard wird im Ausgangszustand korrekt angezeigt	Update (#1) ist durchgeführt	Admin-Dashboard öffnen	Alle Daten werden in der "Printer Information" Card angezeigt.	2
4	Maschinen-Daten im Admin-Dashboard werden mit Fehler-Zuständen angezeigt	Maschinen-Daten werden geladen (#3)	Temperaturen am Chocoformer anpassen Druckköpfe einzeln entfernen	Bei problematischen Temperaturen (<31 und >36) wird eine entsprechende Meldung angezeigt. Bei fehlenden Druckköpfen wird eine entsprechende Meldung angezeigt. Fehlende Temperatur-Werte werden mit einem Strich dargestellt.	8
5	Homing kann aus dem Admin-Dashboard gestartet werden	Die Druckköpfe und die Platte wurden manuell verschoben	Im Admin-Dashboard auf den Home All Button klicken	Alle Achsen werden auf ihre Home-Position gefahren.	3
6	Jobs können aus dem Admin Dashboard gestartet werden	Es muss mindestens ein Job im Benutzer-Front-End erstellt werden.	Jobs in Queue. Admin startet Current Job	Current Job wird ausgeführt	13
7	Bestehender Job kann erneut gedruckt werden	Ein Druck wurde bereits ausgeführt	Admin startet Current Job nochmals	Current Job wird nochmals ausgeführt	8
8	Admin Dashboard zeigt mehrere Jobs an	Einzelne Jobs können erstellt und ausgelöst werden	Mindestens 3 Jobs im Benutzer-Front-End erstellen	Alle Jobs werden mit entsprechenden Zeitangaben angezeigt.	13
9	Es werden beide Druckköpfe während eines Jobs angesprochen	Es wurde ein 2-farbiger Job im Benutzer-Front-End erstellt	Admin startet den 2-farbigen Job	Der Job wird ausgeführt, beide Druckköpfe werden dabei verwendet.	8

10	Benutzerdefinierte Einlegebauteile werden angezeigt		Eigenes Einlegebauteil im Admin-Front-End hochladen. Bestimmte Anzahl von Einlegebauteilen aktivieren. Im Benutzer-Front-End Freihand zeichnen.	Die Bilder werden im Benutzer Front-End angezeigt. Das Ausmessen der Platte wird nach einem Druckstart durchgeführt.	8
TOTAL					102

16.3.2.2 VERSUCHSPROTOKOLL

ID	Tatsächliches Resultat	Einzuplanende Aktionen	Umsetzung	Erfolgreich?
1	Das Script läuft durch und arbeitet wie erwartet.	Zunächst gab es Unsicherheiten, wann die Konfigurationsdateien angepasst werden müssen. Das README muss entsprechend ergänzt werden.	README-Datei ist angepasst.	Ja
2	Wie erwartet.	-	-	Ja
3	Die erwarteten Status-Informationen werden angezeigt. Allerdings erscheint beim Start der Seite auch die Meldung "Druck fertig", da die Applikationen einen Status-Wechsel zu "Idle" interpretiert.	Der Text bei "Alle Druckaufträge" sollte auf zwei Zeilen aufgeteilt werden. Deutschsprachige Texte sind zu lang für eine Zeile. Für die Meldung "Druck fertig", siehe #5.	Zeilenumbruch und Meldung sind angepasst.	Ja
4	Wie erwartet.	-	-	Ja
5	Die beiden Köpfe und das Druckbett werden gehomed. Die Meldung "Druck fertig" erscheint auch,	Der Grund, warum "Idle" zurückgemeldet wird, kann nicht mit Sicherheit ermittelt	Die Meldung ist angepasst.	Teilweise

	wenn der angesteuerte Druckkopf gewechselt wird. Anscheinend wechselt der Status für einen kurzen Moment vor einem Wechsel auf "Idle". Da der Status von der Chocoformer App alle 10 Sekunden geladen wird, tritt dieses Verhalten scheinbar zufällig auf.	werden. Die Meldung soll auf "Drucker ist bereit" umgeändert werden. Damit ist die Meldung auch für #3 richtig.	Der Auslöser für die Meldung könnte weiter optimiert werden. Aufwand und Nutzen müssen aber dafür verglichen werden.	
6	<p>Für die Erstellung der einzelnen Job-Ordner wurden relative Pfade anstelle eines absoluten Pfades verwendet.</p> <p>Job erscheint zuerst in Queue, nicht in Jobs.json geschrieben. Druck-Start kann nicht ausgeführt werden, da der Job schon aus dem JSON gelöscht wurde und der Druck-Start direkt aus dem JSON liest.</p> <p>Zeit-Schätzung funktioniert nur im Sekunden-Bereich.</p> <p>Leere STL-Dateien können nicht gespeichert werden.</p>	<p>Pfad für Job-Ordner anpassen.</p> <p>Slicer-Config umbenennen ("dev" entfernen).</p> <p>Zeit-Schätzung des Slicers muss geparsed werden.</p> <p>Abfrage einbauen, ob das STL leer sein würde (Hat es Text- oder SVG-Inhalt?).</p> <p>Druck-Start anpassen (eventuell nach ID suchen).</p>	Alle Änderungen sind implementiert.	Teilweise
7	Wie erwartet.	-	-	Ja
8	Wie erwartet. Es werden alle Jobs angezeigt, welche auch angezeigt werden sollen.	-	-	Ja
9	Wie erwartet.	-	-	Ja
10	Läuft. Bilder mit einem anderen Seitenverhältnis werden verzerrt im Benutzer-Front-End dargestellt.	Die Erwartungen an das Bild im Upload-Di- alog und im README besser dokumentieren.	Die Dokumentation im GUI und im README sind nachgeführt.	Ja

Zusatz	Das Dashboard ist als Grid angeordnet. Wenn die Job Queue zu gross wird, haben die Drucker-Informationen möglicherweise kein Platz mehr auf der gleichen Bildschirmseite.	Zwischen dem aktuellen Druck-Job und den Drucker-Informationen auf der rechten Seite soll der Abstand entfernt werden.	Die Cards von Material Design sind jetzt platzsparender angeordnet.	-
---------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------	---

16.3.2.3 AUSWERTUNG

Einige eingeplante Versuche beanspruchten mehr Zeit als eingeplant, da kleine Korrekturen im Code notwendig waren und nicht durch eine Konfigurations-Änderung gelöst werden konnten. Für das Debugging wurde praktisch keine Zeit benötigt, da die Code-Stellen einfach zu finden waren und nur einzelne Zeilen angepasst werden mussten. Der zeitintensive Teil war, dass diese Änderungen durch die GitLab-Pipeline für einen erneuten Release kompiliert und anschliessend auf dem Raspberry Pi installiert werden mussten.

Die Ansteuerung von zwei Druckköpfen konnte nicht im Voraus getestet werden, da der verwendete Prusa-3D-Drucker nur einen Druckkopf besitzt. Am Versuchstag liessen sich beide Druckköpfe ohne Probleme aus einem Druckauftrag ansprechen. Das zweite, kritische Feature, die Druckauftrags-Queue, konnte nach den oben genannten Korrekturen am gleichen Versuchstag ebenfalls erfolgreich getestet werden.

Neues Styling: Die Card "Printer Monitoring" wurde entfernt, da diese nicht implementiert wurde. Die Queue benutzt die ganze linke Seite und Current Job und Printer Informationen benutzen die rechte Seite. Damit kann der vorhandene Platz auf dem Bildschirm besser ausgenutzt werden, auch wenn die Liste der Druckaufträge viel Platz beansprucht.

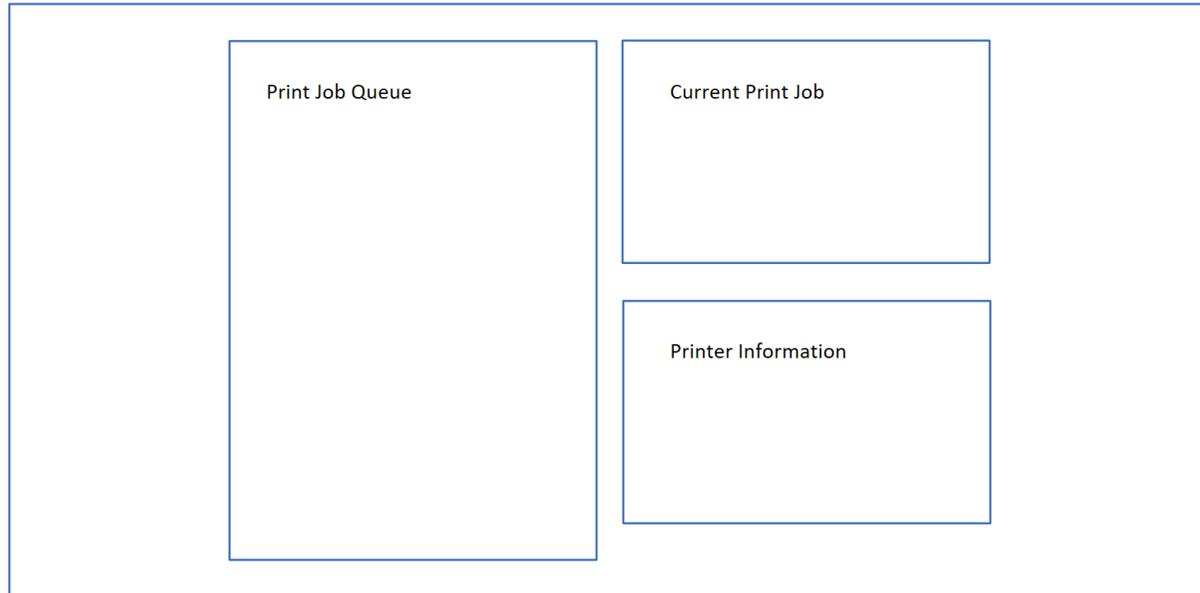


Abbildung 15 Skizze der Neuordnung der Elemente im Administrator Dashboard

16.4 AUSGEFÜLLTE FRAGEBOGEN

USABILITY-TEST CHOCOFORMER APP – FRAGEBOGEN FÜR DIE TESTPERSON

Mit deinem Feedback hilfst du uns, die Chocoformer App zu verbessern. Zum Abschluss bitten wir dich, diesen Fragebogen auszufüllen, um uns mitzuteilen, wie dir die Applikation gefallen hat.

Wie würdest du die Applikation als Ganzes benoten? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Wie gut hat dir die Gestaltung gefallen? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Welche Elemente der Gestaltung haben dir besonders gefallen / nicht gefallen?

- Bekannte Bestellstruktur
- Nur ein Objekt zur Auswahl ist verwirrend

Wie gut hast du dich zurechtgefunden? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Was hat dir am besten an der Applikation gefallen?

Strukturierung der Bestellung

Was hat dich am meisten gestört?

Schriftarten wirken unterschiedlich

Hast du Informationen oder Funktionen vermisst? Wenn ja, welche?

- Zeitangaben zum Druck
- Feedback beim Auswählen / Uchltasten

Kennst du vergleichbare Applikation / Webseiten?

- Online shops

Falls ja: Gefallen diese Applikationen dir besser oder schlechter als die, die du gerade getestet haben?

- Ähnliche Apps fühlen sich in der Bedienung gleich gut an



USABILITY-TEST CHOCOFORMER APP – FRAGEBOGEN FÜR DIE TESTPERSON

Mit deinem Feedback hilfst du uns, die Chocoformer App zu verbessern. Zum Abschluss bitten wir dich, diesen Fragebogen auszufüllen, um uns mitzuteilen, wie dir die Applikation gefallen hat.

Wie würdest du die Applikation als Ganzes benoten? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Wie gut hat dir die Gestaltung gefallen? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Welche Elemente der Gestaltung haben dir besonders gefallen / nicht gefallen?

+ einfaches Layout
- keine Zeichnungs-Fläche

Wie gut hast du dich zurechtgefunden? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Was hat dir am besten an der Applikation gefallen?

logischer Ablauf

Was hat dich am meisten gestört?

wechselnder Gebrauch von Maus & Touchpanel
↳ ohne vorhandene Maus wäre es kein Problem

Hast du Informationen oder Funktionen vermisst? Wenn ja, welche?

—

Kennst du vergleichbare Applikation / Webseiten?

nein

Falls ja: Gefallen diese Applikationen dir besser oder schlechter als die, die du gerade getestet haben?

USABILITY-TEST CHOCOFORMER APP – FRAGEBOGEN FÜR DIE TESTPERSON

Mit deinem Feedback hilfst du uns, die Chocoformer App zu verbessern. Zum Abschluss bitten wir dich, diesen Fragebogen auszufüllen, um uns mitzuteilen, wie dir die Applikation gefallen hat.

Wie würdest du die Applikation als Ganzes benoten? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Wie gut hat dir die Gestaltung gefallen? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Welche Elemente der Gestaltung haben dir besonders gefallen / nicht gefallen?

Das die Druckunterlage als Bild gezeigt wird

Wie gut hast du dich zurechtgefunden? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Was hat dir am besten an der Applikation gefallen?

Kein Filter 😊
Die Idee der Applikation

Was hat dich am meisten gestört?

-

Hast du Informationen oder Funktionen vermisst? Wenn ja, welche?

Den Text verschieben, ähnlich wie Word

Kennst du vergleichbare Applikation / Webseiten?

evt. Neme Webseiten
aber noch nie eine Seite gesehen, auf der man etwas 3D drucken kann

Falls ja: Gefallen diese Applikationen dir besser oder schlechter als die, die du gerade getestet haben?

Ist ähnlich aufgebaut, daher war Navigation sehr intuitiv

USABILITY-TEST CHOCOFORMER APP – FRAGEBOGEN FÜR DIE TESTPERSON

Mit deinem Feedback hilfst du uns, die Chocoformer App zu verbessern. Zum Abschluss bitten wir dich, diesen Fragebogen auszufüllen, um uns mitzuteilen, wie dir die Applikation gefallen hat.

Wie würdest du die Applikation als Ganzes benoten? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Wie gut hat dir die Gestaltung gefallen? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Welche Elemente der Gestaltung haben dir besonders gefallen / nicht gefallen?

+ straight forward
- manchmal unklar, was ausgewählt wurde, z.B. welcher Keks oder welches Tool

Wie gut hast du dich zurechtgefunden? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Was hat dir am besten an der Applikation gefallen?

"Assistent-Design" wie Schritt für Schritt vorgegangen wird

Was hat dich am meisten gestört?

Hab keine Schoggi bekommen

Hast du Informationen oder Funktionen vermisst? Wenn ja, welche?

Informationen: welcher Keks ausgewählt ist, welches Tool
↳ Vorschau des Keks vor dem Drück

Kennst du vergleichbare Applikation / Webseiten?

Falls ja: Gefallen diese Applikationen dir besser oder schlechter als die, die du gerade getestet haben?

USABILITY-TEST CHOCOFORMER APP – FRAGENBOGEN FÜR DIE TESTPERSON

Mit deinem Feedback hilfst du uns, die Chocoformer App zu verbessern. Zum Abschluss bitten wir dich, diesen Fragebogen auszufüllen, um uns mitzuteilen, wie dir die Applikation gefallen hat.

Wie würdest du die Applikation als Ganzes benoten? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Wie gut hat dir die Gestaltung gefallen? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Welche Elemente der Gestaltung haben dir besonders gefallen / nicht gefallen?

nicht nur einen runden keks
sonst i.o.

Wie gut hast du dich zurechtgefunden? (1 = mangelhaft, 6 = sehr gut)

1 2 3 4 5 6

Was hat dir am besten an der Applikation gefallen?

Was hat dich am meisten gestört?

von Hand zeichnen

Hast du Informationen oder Funktionen vermisst? Wenn ja, welche?

Stift

Kennst du vergleichbare Applikation / Webseiten?

NEIN

Falls ja: Gefallen diese Applikationen dir besser oder schlechter als die, die du gerade getestet haben?

