

Hochschule Worms  
Fachbereich Informatik  
Studiengang Angewandte Informatik - dual (B.Sc)

Abschlussarbeit

# Bachelorarbeit

Digitalisierung der Anmeldung und Zustellung von Weinen  
zu Weinproben des Regionalverbands für Weine in der  
Weinregion Mosel als TYPO3-Extension

Version 1.0

Vorgelegt von

Leon Etienne, 676838  
inf3642@hs-worms.de  
mit Medienagenten Stange & Ziegler oHG  
Im Sommersemester 2023  
Bad Dürkheim

bei Volker Schwarzer  
schwarzer@hs-worms.de  
Worms

# Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist noch nicht in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung vorgelegt worden.

Bad Dürkheim, 22. Dezember 2022

Leon Etienne \_\_\_\_\_

---

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>I</b>
<b>Tabellenverzeichnis</b>	<b>II</b>
<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Glossar</b>	<b>IV</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Problemstellung . . . . .	1
1.2. Zielsetzung . . . . .	2
<b>2. Stand der Forschung</b>	<b>3</b>
2.1. Modell nach Parviainen et al . . . . .	3
2.2. Modell nach Verhoef et al . . . . .	6
<b>3. Stand der Technik</b>	<b>7</b>
3.1. Die bestehende Webseite . . . . .	7
3.2. QR-Codes erstellen . . . . .	7
3.2.1. Javascript-Implementationen . . . . .	7
3.2.2. PHP-Implementationen . . . . .	9
3.2.3. Vergleich in Bezug auf die Problemstellung . . . . .	10
3.2.4. Fazit . . . . .	14
<b>4. Anforderungserfassung</b>	<b>15</b>
4.1. Interview mit Product Owner . . . . .	16
4.2. Online-Fragebögen für Stakeholder . . . . .	16
4.3. Ergebnisse . . . . .	17
<b>5. Konzeption und Entwicklung</b>	<b>18</b>
<b>6. Ergebnisse</b>	<b>19</b>

---

<b>7. Ausblick</b>	<b>20</b>
<b>Literaturverzeichnis</b>	<b>21</b>
<b>Anhang</b>	<b>25</b>
<b>A. Interview-Fragebogen</b>	<b>26</b>
<b>B. Online-Fragebogen Weingüter</b>	<b>28</b>
<b>C. Interview-Protokoll</b>	<b>31</b>
<b>D. Pflichtenheft</b>	<b>35</b>

---

# Abbildungsverzeichnis

2.1. Model for tackling digital transformation . . . . .	4
2.2. Plan-Do-Check-Act (PDCA) . . . . .	5
3.1. Vergleichsergebnisse QR-Code Bibliotheken . . . . .	12

# Tabellenverzeichnis

4.1. Initial bekannte Anforderungen . . . . .	15
-----------------------------------------------	----

# Abkürzungsverzeichnis

**WM** “Weinland Mosel” e.V.

**PDCA** Plan-Do-Check-Act

**SWOT** Strength-Weakness-Opportunity-Threat

**FA** Funktionale Anforderung

**NFA** Nichtfunktionale Anforderung

**ELN** Einlieferungsnummer

**JAP** Jahresauswahlprobe

# Glossar

## Frontend

*“Das Frontend einer Webseite ist der Teil der Webseite, der Endnutzern präsentiert wird. Ein Frontend besteht aus HTML und Cascading Style Sheets” [Udjaja, 2018].*



# 1. Einleitung

Der Regionalverband für Weine “Weinland Mosel” e.V. (WM) lässt Weine in organisierten Weinproben, sog. Jahresauswahlproben (JAPs), von Juroren bewerten. Teilnehmende Weingüter registrieren ihre Weine in verschiedenen Kategorien und schicken diese auf dem Postweg ein. Dieser Prozess bildet sich in Form von ausgedruckten Formularen, die von Hand ausgefüllt und von Hand in eine Excel-Tabelle übertragen werden, ab.

## 1.1. Problemstellung

Die Teilnehmenden Weingüter schicken ihre Weine zusammen mit Formularen über den Postweg zu WM. Es ist der Normalfall, dass ein teilnehmendes Weingut *mehrere* Weine zur Bewertung anbringt. In diesem Fall ist für jeden anzumeldenden Wein ein solches Formular erneut auszufüllen. Hierbei werden sämtliche auf das Weingut bezogene Daten redundant ausgefüllt. Diese Daten sind redundant, da sie keine Eigenschaften der Weine, sondern die des Weingutes selbst sind. Da sich das Weingut zwischen den Weinen nicht ändert, ändern sich die darauf bezogenen Daten auch nicht. Sie müssen aber für jeden Wein erneut ausgefüllt werden. Abgesehen davon, dass solche Redundanzen auf Weinguts- und Verbandsseite die hedonische Qualität schädigen, bietet so ein Workflow Freiraum für Fehler und Inkonsistenzen. Dieser Workflow mit den zuvor genannten Nachteilen wird auf Verbandsseiten, nach Zustellung der Weine, weiter fortgeführt: WM erfährt erstmalig mit der Zustellung eines Weines von dessen Teilnahme. Das erschwert das Planen der Logistik, da im Voraus keine konkrete Zahl der zu erwartenden Flaschen bekannt ist. Geht eine Flasche auf dem Postweg verloren, könnte das unbemerkt bleiben, da der Prozess für das Weingut mit dem Versand endet und der Prozess für WM erst mit dem Erhalt des der Flasche beiliegendem Formulars beginnt. Der Postweg stellt somit eine Lücke zwischen diesen

Prozessen dar. Kommt ein teilnehmender Wein bei WM an, wird das beiliegende Formular von Hand in eine Excel-Tabelle übertragen. Diese Schnittstelle ist besonders ressourcenaufwändig und fehleranfällig, weil es oft vorkommt, dass die zuteils dysgraphisch verfassten Formulare nur schwer, mehrdeutig, oder gar nicht dechiffriert werden können. In diesem Prozess wird der Teilnahme des Weines eine inkrementell aufsteigende Einlieferungsnummer (ELN) zugewiesen, die anschließend in Form eines Aufklebers an der Flasche befestigt wird. Anschließend wird die Flasche im Lager verstaut.

### 1.2. Zielsetzung

Ziel dieser Arbeit ist es, in Erfahrung zu bringen, wie der zuvor genannte Prozess bestmöglichst, im Rahmen bestimmter Constraints und funktionalen- sowie nichtfunktionalen Anforderungen, digitalisiert werden kann. Während die Constraints bereits bekannt sind, werden detaillierte Anforderungen im Rahmen der Anforderungstechnik ausgearbeitet [Heinemann, 2021]. Die Constraints besagen, dass der Anmeldeprozess in die existierende Internetpräsenz des Weinverbandes integriert werden muss. Bei dieser Internetpräsenz handelt es sich um ein TYPO3-Redaktionssystem. Sämtliche Interaktionen zwischen Akteuren, die nicht WM oder dem System zugehörig sind, müssen im Frontend der Webseite stattfinden. Oberflächen für Mitarbeiter von WM dürfen in der TYPO3-Backend-Oberfläche implementiert werden.

Somit lautet die **Forschungsfrage**:

*Was ist die beste Möglichkeit, um die Anmeldung und Zustellung von Weinen zu Weinproben des Regionalverbunds für Weine in der Weinregion Mosel als TYPO3-Erweiterung zu realisieren, um davon zu profitieren?*

## 2. Stand der Forschung

Der Stand der Forschung beleuchtet verschiedene Erkenntnisse zur Digitalisierung und zur Digitalen Transformation.

### 2.1. Modell nach Parviainen et al

*“The importance of digitalization is becoming understood, but the question now is how to do it in practice in order to best benefit from it.”*  
[Parviainen et al., 2022].

Parviainen et al. stellten sich diese Frage und entwickelten in ihrer Forschungsarbeit einen konzeptionellen Rahmen, um zu verstehen, wie die Digitalisierung in der Praxis umgesetzt werden kann und welche Vorteile sich daraus ergeben. Dieser Rahmen basiert auf dem Plan-Do-Check-Act (PDCA) -Prinzip.

Dieses Rahmenwerk sieht anhand des PDCA-Prinzips vier Schritte vor:

**Im ersten Schritt** wird definiert, wie weit die Digitalisierung für das Unternehmen gehen kann und welche Position das Unternehmen dabei anstrebt. Dieser Schritt kann in vier Teilschritte unterteilt werden: Ausmaße, Treiber, Szenarien und Ziele. Für die Bestimmung der Ausmaße ist die Analyse aktueller Trends und deren Relevanz für die Domäne des Unternehmens wichtig. Ebenfalls ist wichtig, wie weit diese Trends bereits im Fachgebiet verankert sind. Zur Einordnung eignen sich Strength-Weakness-Opportunity-Threat (SWOT) -Analysen. Diese Analysen sind die Grundlage um ein Unternehmen in der Digitalisierung aufzustellen.

Aus den Ergebnissen der Trendanalysen sollten dann Treiber identifiziert werden. Diese Treiber sollten auf der Grundlage zukünftiger Ergebnisse skalierbar sein: Beispielsweise könnten drastische Maßnahmen erforderlich sein, um drastische Auswirkungen zu verhindern oder dras-

tische Verbesserungen zu erreichen.

Für die relevantesten Treiber sollten Zukunftsszenarien untersucht werden. Dies ist wichtig, um zu wissen, welche Auswirkungen bestimmte Trends in welcher Ausprägung haben werden. Relevant sind hier die Vorteile der Umsetzung des Szenarios, die Kosten der Umsetzung sowie die Risiken, das Szenario nicht umzusetzen oder doch umzusetzen. Auf dieser Basis kann das beste Szenario ausgewählt werden.

Aus diesem Szenario werden schließlich die Ziele der Digitalisierung abgeleitet. Diese Ziele müssen so formuliert sein, dass sie mit der Ausgangssituation verglichen werden können.

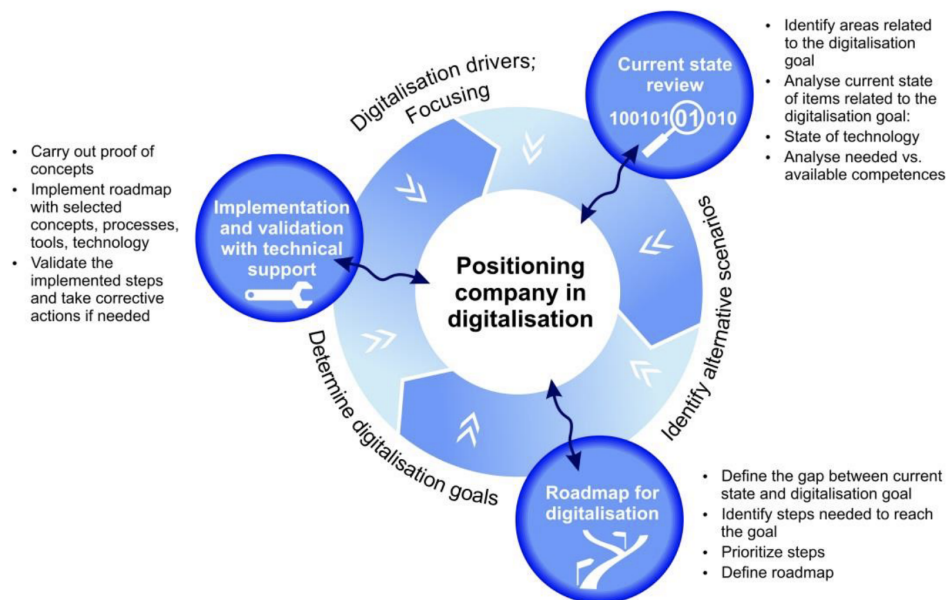


Abbildung 2.1.: Model for tackling digital transformation

Quelle: [Parviainen et al., 2022]

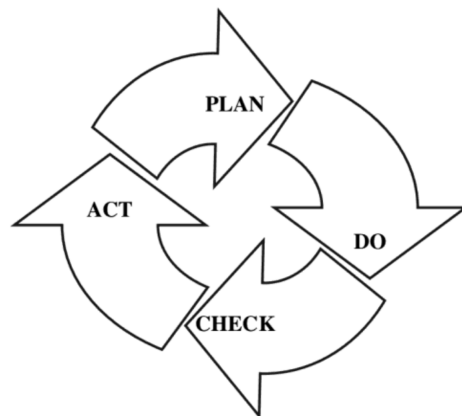


Abbildung 2.2.: Plan-Do-Check-Act (PDCA)

Quelle: [Abraham, 2005]

**Im zweiten Schritt** wird der Ist-Zustand des Unternehmens ermittelt. Dazu wird die aktuelle Positionierung des Unternehmens im Hinblick auf den Zielzustand mit Fokus auf die Digitalisierungsziele betrachtet. Dazu wird der Ist-Zustand im Kontext des Soll-Zustandes anhand definierter Fragen bewertet. Die Auswahl der Fragen unterscheidet sich je nach Art der Ziele. Der gesamte Fragenkatalog kann im Detail der Ausarbeitung von Parviainen et al. entnommen werden.

**Der dritter Schritt** ist die Festlegung der konkreten Schritte, die für den Übergang vom Ist-Zustand zum Soll-Zustand erforderlich sind. Dazu muss zunächst die Lücke zwischen dem Ist- und dem Soll-Zustand identifiziert werden. Relevant ist dabei der aktuelle Stand der Technik und welche Veränderungen notwendig sind, um den Zielzustand zu erreichen. Anschließend sollten die konkreten Schritte identifiziert werden, die erforderlich sind, um diese Lücke zu schließen. Wenn zum Beispiel ein Treiber “interne Effizienz” ist, könnten die Schritte darin bestehen, neue digitale Werkzeuge zu integrieren. Schließlich werden diese Schritte analysiert und priorisiert. Prädestiniert dafür sind Kosten-Nutzen-Analysen, Analysen der Umsetzbarkeit, des Wartungsaufwands und der

Mitarbeiterschulung.

**Der vierte Schritt** Der vierte Schritt befasst sich mit der Umsetzung der in Schritt 3 geplanten Maßnahmen und der Bewertung der erzielten Ergebnisse. Diese Bewertung der Ergebnisse sollte z.B. soziokulturelle Barrieren berücksichtigen, die sich aus den Reaktionen bestimmter Stakeholder ergeben, die möglicherweise negativ auf bestimmte Veränderungen reagieren oder Schwierigkeiten bei der Einführung neuer Technologien haben. Wenn diese Analyse zeigt, dass die Ziele der Digitalisierung nicht erreicht wurden, sollten Korrekturmaßnahmen in früheren Phasen ergriffen werden.[Parviainen et al., 2022].

## 2.2. Modell nach Verhoef et al

Nach Verhoef et al lässt sich der hier so genannte “Prozess der Digitalisierung” in drei Phasen unterteilen. Diese drei Phasen sind *Digitization*, *Digitalization* und *Digital Transformation* [Verhoef et al., 2021].

Die Phase *Digitization* befasst sich mit der Umwandlung analoger Datenstrukturen und Modelle in digitale Datenmodelle, so dass diese digital, z.B. in Form von Nullen und Einsen, gespeichert und elektronisch weiterverarbeitet werden können [Dougherty and Dunne, 2012, Loebbecke and Picot, 2015].

*Digitalisierung* beschreibt den Prozess der Veränderung bestehender Geschäftsprozesse, um mit digitalen Werkzeugen und Datenmodellen zu arbeiten [Li et al., 2016]. Die letzte Phase, die *Digitale Transformation* beschreibt eine firmenweite Veränderung, die beispielsweise Ergründungen neuer Geschäftsmodelle mit sich bringen könnte [Pagani and Pardo, 2017].

## 3. Stand der Technik

Der Stand der Technik bezieht sich auf bestehende praktische Umsetzungen der erforderlichen Technologien. Im Wesentlichen gibt es zwei Arten von Technologien, die untersucht werden müssen: Die bestehende Website von WM und eine PHP-Bibliothek zur Erzeugung von QR-Codes.

### 3.1. Die bestehende Webseite

Als Mitentwickler des Projektes ist dem Author dieser Ausarbeitung bekannt, dass die bestehende Webseite von WM ein TYPO3-Redaktionssystem ist. Das Frontend der Webseite wird mit Webpack und Sass übersetzt. Webpack ist ein Modulbundler [Zimmermann, 2017] und Sass ein CSS-Präprozessor [W3Schools, 2023].

### 3.2. QR-Codes erstellen

Um mit QR-Codes zu arbeiten, ist es unabdinglich, QR-Codes zu erstellen, da dieselben sonst nicht vorhanden sind. Prinzipiell gibt es zwei Möglichkeiten QR-Codes zu erstellen: Auf Browserseite in JavaScript und auf Serverseite in PHP. Das ist so, da es lediglich diese zwei Domänen gibt.

#### 3.2.1. Javascript-Implementationen

##### **jquery-qrcode**

*jquery-qrcode* ist ein Plugin für JQuery um dynamisch QR-Codes auf Browserseite zu generieren. Jedoch verweist diese Bibliothek selbst auf ihren desolaten Zustand und empfiehlt stattdessen “kjua” zu verwenden [Jung, 2020a]. Somit scheidet *jquery-qrcode* für nähere Evaluationen aus, da die Software nicht mehr gepflegt wird. *jquery-qrcode* wurde bis zum heutigen Tage 853 mal von Nutzern mit einem Stern markiert

[Jung, 2020a]. Wenn ein Nutzer ein Github-Repository mit einem Stern markiert, interessiert sich der Nutzer weit genug, um über Neuigkeiten auf dem Laufenden gehalten werden zu wollen [Github, 2023]. JQuery-qrcode ist MIT-lizenziert [Jung, 2020a].

#### **kjua**

*Kjua* ist eine Javascript-Bibliothek, um dynamisch QR-Codes auf Browserseite zu generieren. Im Gegensatz zu *jquery-qrcode* funktioniert *Kjua* auch ohne JQuery. Es werden diverse Stilattribute für gestaltete QR-Codes unterstützt [Jung, 2020b]. *Kjua* kann QR-Codes über HTML-Canvas, Bilder und Vektorgrafiken umsetzen. Das ist bei näherer Betrachtung der *Kjua* Tech Demo (<https://larsjung.de/kjua/latest/demo>) ersichtlich, jedoch nicht explizit erwähnt. Nach dem aktuellen Stand wurde *Kjua* von 243 Nutzern mit einem Stern markiert. *Kjuas* letzte Codeänderung liegt circa zwei Jahre zurück. Es gibt eine unbeantwortete, sieben Jahre alte Pull-Request und Issues scheinen ignoriert zu werden. *Kjua* ist MIT-lizenziert [Jung, 2020b].

#### **soldair/node-qrcode**

*Soldair/node-qrcode* ist eine node.js-basierte Implementation eines QR-Code Generators und bietet somit Funktionalität serverseitig, als CLI, sowohl als auch Browserseitig an. Die Readme-Datei zeugt von Länge, ist reich an Beispielen und detailreichen Erklärungen. Der letzte Commit ist zu diesem Zeitpunkt knapp älter als ein halbes Jahr. Somit macht das Projekt einen moderat gepflegten Eindruck. Die Readme-Datei verweist auf Unit Tests bei Travis, jedoch lief die letzte Pipeline vor circa zwei Jahren, Februar 2021, durch und schlug zudem fehl. Einige Pull-Requests und Issues werden seit Jahren ignoriert [Day, 2020]. Die Bibliothek wurde 74 Millionen mal heruntergeladen mit 6308 Sternen markiert. *Soldair/node-qrcode* ist MIT-lizenziert [npm, inc, 2020].



### 3.2.2. PHP-Implementationen

#### **chillerlan/php-qrcode**

Eine arrivierte Lösung für QR-Code Generierung in PHP ist *chillerlan/php-qrcode*, umgesetzt durch Eugen Rochko und weitere. Das zeigt sich durch den zu diesem Zeitpunkt auf 5 Millionen stehendem Downloadzähler, mehrere Entwickler und den letzten Commit, der zu diesem Zeitpunkt nicht älter als einen Monat ist. Insgesamt erfolgten bis dato 808 Commits von 6 Entwicklern. Das Projekt verfügt über Unit-Tests, die 90% der Zeilen in der Codebase abdecken. Rochko übernahm Teile der Codebase aus dem Java-Projekt “ZXing” und übersetzte diese zu PHP. Issues und Pull Requests sind alle bearbeitet. Es gibt zu diesem Zeitpunkt keine unbeantworteten Issues oder Pull-Requests. *chillerlan/php-qrcode* basiert auf einer angepassten Version von *kazuhikoarase/qrcode-generator*. Einzig auffällig sind die Commitnachrichten, die zuteils nur aus einem (1) Emoji bestehen. *chillerlan/php-qrcode* wurde von 1212 Github-Nutzern mit einem Stern markiert. Die Bibliothek ist MIT-lizenziert [Rochko, 2022b].

#### **kreativekorp/barcode**

*kreativekorp/barcode* ist eine PHP-Bibliothek zur Generierung von QR-Codes, bereitgestellt von *Kreative Software*, R.G. Bettencourt. Diese Implementation umfasst eine Vielzahl an Barcode-Formaten und unterstützt eine Vielzahl an Anpassungsmöglichkeiten. Das Projekt wurde bis zum heutigen Tage 189 mal mit einem Stern markiert [Bettencourt, 2018]. Die letzte Codeänderung dieses Projektes ist auf 2018 datiert [Bettencourt, 2018]. Das liegt zu diesem Zeitpunkt vier Jahre in der Vergangenheit. Vier von den sechs Commits erfolgten innerhalb von zwei Tagen in 2016. Ein weiterer Commit erfolgte wenige Tage später. Der aktuellste Commit wurde knapp zwei Jahre später, 2018, veröffentlicht. Damit ist diese Bibliothek de-facto sechseinhalb Jahre alt und wurde seitdem ein mal um Features erweitert. Issues und Pull-Requests werden weitestgehend ignoriert. Die Bibliothek verwendet die MIT-Lizenz [Bettencourt, 2018].

#### **Bacon/BaconQRCode**

*Bacon/BaconQRCode* ist eine PHP-Bibliothek zur Generierung von QR-Codes, bereitgestellt von Ben Scholzen, hinter der Github-Organisation “Bacon”, dessen einziges Mitglied Scholzen darstellt. Verlinkt ist eine Homepage, die zu einer Nginx-“Hello World”-Seite führt. Begleitet wird BaconQRCode von etlichen weiteren Bacon-Applikationen wie beispielsweise *BaconPdf*, *BaconStringUtils* und *BaconUser* um nur einige zu nennen. Insgesamt machen die stichprobenartig betrachteten Projekte einen desolaten Eindruck mit zuteils aktuellsten Commits von vor 10 Jahren. BaconQRCode stellt das beliebteste und gepflegteste Projekt mit 1508 Sterne-Markierungen und einem aktuellsten Commit von vor zwei Monaten dar. BaconQRCode erfreut sich an 17 Entwicklern, die jeweils zumindest einen Commit beigetragen haben. Insgesamt fanden zu diesem Zeitpunkt 177 Commits statt. Githubs DependencyGraph verzeichnet nahezu 80.000 Projekte die BaconQRCode verwenden [Scholzen, 2018] und Packagist spricht von 50 Millionen Downloads [Packagist, 2023]. Eine Dokumentation neben der Readme-Datei existiert nicht und die diese ist bescheiden. BaconQRCode kann QR-Codes als Rasterbild und Vektorgrafiken (SVG und EPS) generieren. Spezielle Styles sind nicht erwähnt. Ein Großteil der Issues und Pull-Requests wurden behandelt. BaconQRCode ist mit einer BSD-2-Clause-Lizenz lizenziert [Scholzen, 2018].

#### **3.2.3. Vergleich in Bezug auf die Problemstellung**

Um eine Bibliothek als “die Beste” für einen Anwendungsfall zu kurieren, müssen die konkreten Anforderungen und Constraints für diesen Anwendungsfall beachtet werden. Das ist so, da verschiedene Eigenschaften der Bibliotheken verschiedene Auswirkung in Gewichtung und Richtung je nach Anwendungsfall aufweisen. Hierfür werden die zuvor vorgestellten Bibliotheken zur Erstellung von QR-Codes in den folgenden Attributen gegenübergestellt:

##### **Funktionalität**

Der Umfang der für diese Problemstellung relevanten Funktionen

in Annahme dessen, dass die Bibliothek syntaktisch und pragmatisch korrekt [Heinemann, 2021] ist.

#### **Gepflegtheit**

Das Ausmaß, in dem das Projekt aktiv gepflegt wird und ordnungsgemäß entwickelt zu sein scheint. Hierzu zählen beispielsweise: Bearbeitung von Issues, Bearbeitung von Pull-Requests, Präsenz von Tests, Präsenz einer angemessenen Dokumentation, häufige Commits, mehr als nur ein Contributor, Anzahl der Sterne auf Github (Ausmaß an tieferem, öffentlichem Interesse [Github, 2023]), sowie der Anzahl der Downloads bzw Installationen, falls verfügbar.

#### **Workflow-Eignung**

Die Eignung einer Bibliothek in existierende Workflows und Constraints übernommen zu werden. Maßgeblich, ob und mit wie viel Aufwand eine Bibliothek in das Projekt übernommen werden kann. Ebenfalls ist relevant, ob die Lizenz einer Bibliothek überhaupt deren Einbindung gestattet, bzw. welche Bedingungen gelten.

Hierfür werden den verschiedenen Bibliotheken Punkte  $([0, 10])$  in den zuvor genannten Kategorien vergeben. Die Kumulativpunktzahl einer Bibliothek beschreibt deren Gesamteignung.

#### **kjua**

Kjua ist funktionell für dieses Projekt gut aufgestellt, da es optisch ansprechende QR-Codes mit Logo unterstützt. Das wird mit acht Punkten in “Funktionalität” bemessen. Kjua macht andererseits einen eher desolaten Eindruck und lässt somit begründete Zweifel an dessen Zukunftssicherheit zu. Eine knappe Dokumentation ist verfügbar. Für die Dokumentation werden zwei Punkte in “Gepflegtheit” angerechnet. Kjua ist als Javascript-Bibliothek nur schwer mit den Anforderungen vereinbar, da der QR-Code in einem PDF eingebunden werden soll. Hierfür wäre ein serverseitiger Generator prädestiniert. Kjuas Lizenz erlaubt Verwendung in kommerziellen, Closed-Source Projekten [Opensource.org, b].

Die Exklusivität für Nutzung in Webbrowsern schließt eine Einbindung in den Workflow aus.

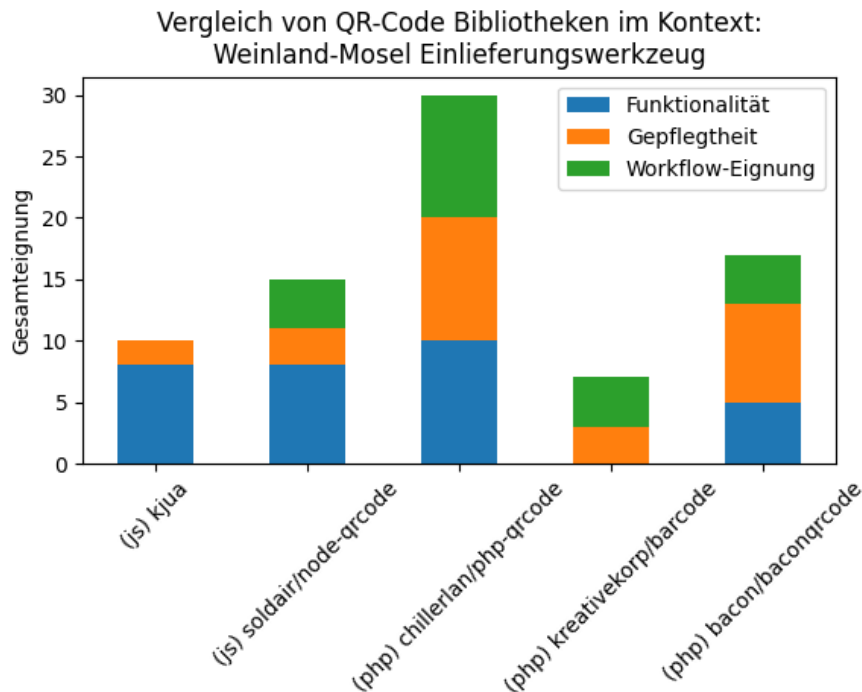


Abbildung 3.1.: Vergleichsergebnisse QR-Code Bibliotheken

Quelle: Eigene Darstellung

#### **soldair/node-qrcode**

Soldairs Lösung sieht dokumentativ und funktional vielversprechend aus. Hieraus bilden sich 8 Punkte in “Funktionalität”. Da diese Bibliothek eine Node.js-Bibliothek ist und Node.js ohne Browser, als Kommandozeilenapplikation ausgeführt werden kann [OpenJS Foundation, 2023], böte sich Soldair/node-qrcode als serverseitige Anwendung an. Auch die Lizenz spräche nicht gegen eine Verwendung. Dennoch wäre es ein vermeidbarer Mehraufwand und ggf. imperformant generierte QR-Codes aus einer Node.JS-Applikation in eine PHP-Umgebung zu übertragen. Das bildet sich mit vier Punkten in “Workflow-Eignung” ab. Darüberhinaus macht Soldair/node-qrcode einen verbesserungswürdigen Ein-

druck der Projektpflege, wofür es lediglich drei Punkte gibt.

#### **chillerlan/php-qrcode**

Rochkos Lösung macht einen aktiv gepflegten Eindruck und wird von großen Downloadzahlen gestützt. Rochko stellt eine ausführliche Dokumentation und Codebeispiele bereit. Hierfür wird die volle Punktzahl in “Gepflegtheit” vergeben. Der Funktionsumfang übersteigt weit die Anforderungen und gestattet so ausführlich gestaltete QR-Codes [Rochko, 2023], die somit ebenso die hedonische Qualität fördern [König, 2022]. Bewertet wird das mit einer vollen Punktzahl in “Funktionalität”. Chillerlan/php-qrcode lässt sich nahtlos in das existierende Projekt eingliedern, da es als PHP-Bibliothek über Composer eingebunden werden kann und eine eigene API bereitstellt [Rochko, 2022b]. Die von Rochko verwendete Lizenz gestattet eine unkomplizierte Verwendung. Chillerlan/php-qrcode funktioniert sowohl mit PHP 7.4 und 8.0. Die Bibliothek benötigt zwei weitere Abhängigkeiten. Eine dieser Abhängigkeiten ist ebenfalls von Rochko bereitgestellt [Rochko, 2022c] und weist eine ähnlich gute Projektpflege auf [Rochko, 2022a]. Das wird mit der Maximalwertung in “Workflow-Eignung” berechnet.

#### **creativekorp/barcode**

Kreativekorp beeindruckt durch Nutzungsbeispiele und Dokumentation in der Readme-Datei, sowie einer Vielzahl unterstützter Barcode-Formate, darunter auch QR-Codes und einiger improvisierter Tests. Hierfür werden drei Punkte in “Gepflegtheit” vergeben. 0 Punkte in “Workflow-Eignung” rechtfertigen sich durch die Abwesenheit jeglicher Unterstützung für Paketmanager, wodurch eine saubere Verwendung in dem Brownfield-Projekt Änderungen an der Bibliothek selbst erfordern würde. Die Funktionalität wurde aufgrund der desaströsen Gepflegtheit und Eignung nicht näher untersucht, da eine Verwendung selbst mit guter Funktionalität nicht infrage käme.

#### **Bacon/BaconQRCode**

BaconQRCode nennt keine speziellen Optionen um näheren Einfluss auf den generierten QR-Code auszuüben. Es werden Rasterbilder und verschiedene Vektorformate als Renderziel unterstützt. Aufgrund der fehlenden Optionen, die andere Bibliotheken bereitstellen, werden lediglich fünf von 10 Punkten für “Funktionalität” vergeben. Die Projektgepflegtheit ist inkonsistent. Manche Stellen, beispielsweise die Mitwirkenden, Alter des neuesten Commits, Tests und Nutzerzahlen machen einen guten Eindruck, während Punkte wie die Dokumentation, die Organisationswebseite und andere Projekte Sorgen begründen. Für die “Ge pflegtheit” werden 8 Punkte vergeben. Da es sich hierbei um eine PHP-Bibliothek handelt, die über Composer in PHP- 7.4 und 8.0 Projekte geladen werden kann und eine API bereitstellt, ist die “Workflow-Eignung” gut. Die BSD-2-Clause-Lizenz verkompliziert eine Integration, da dadurch eine Copyright-Notiz an Nutzer gezeigt werden muss [Opensource.org, a]. Dadurch werden drei Punkte einer vollkommenen Workflow-Eignung abgezogen, wodurch sieben Punkte vergeben werden.

#### **3.2.4. Fazit**

Nach Evaluation der verschiedenen QR-Code-Bibliotheken im Kontext der vorliegenden Problemstellung erweist sich *chillerlan/php-qrcode* als die am besten geeignete Bibliothek. Somit wird *chillerlan/php-qrcode* als QR-Code Technologie verwendet werden.

## 4. Anforderungserfassung

Obwohl bereits ein grober Anriss des Zielsystems bekannt ist, ist es unabdinglich eine Anforderungsanalyse durchzuführen, um Details auszuarbeiten [Ebert, 2019]. Hierbei ist es wichtig, kein exzessives Pflichtenheft aufzubauen, denn letztendlich zählt nur, was dem Kunden geliefert wird. Nicht, wie viele gar nicht benötigte Anforderungen umgesetzt wurden. *“Zu viele oder falsche Anforderungen ruinieren Budgets, Termine und die Qualität.”* [Ebert, 2019]. Die Anforderungen eines Produktes sind in drei Kategorien einzuteilen: Funktionale Anforderungen (FAs), Nichtfunktionale Anforderungen (NFAs), und Constraints [Heinemann, 2021]. Wie oben erwähnt, sind bereits die Constraints und einige funktionale und nichtfunktionale Anforderungen bekannt. Diese sind:

Art	Beschreibung
Constraint	Einbau in brown-field TYPO3
Constraint	Mitarbeiter-UI in TYPO3-Backend
Constraint	Kunden-UI im Frontend
NFA	Aus dem Papierformular soll ein Webform werden
NFA	Angaben zum Weingut des Weines sollen aus dem Accountdatensatz anstatt aus dem Webform kommen
NFA	Registrierte Weinteilnahmen bestehen aus einem Wein und einem Zustand(Ausstehend,Eingegangen).
FA	Beim Erstellen einer Weinteilnahme soll ein QR-Code als PDF generiert werden, der den Wein identifiziert
FA	Durch scannen des QR-Codes soll dem Wein der Status “Eingegangen” zugewiesen werden

Tabelle 4.1.: Initial bekannte Anforderungen

Um nähere Anforderungen zu ermitteln, werden die Befragungstechniken “Interview” und “Fragebogen” verwendet [Heinemann, 2021].

## 4.1. Interview mit Product Owner

Zunächst wird ein Interview mit dem Product Owner geführt. Ziel dieses Interviews ist es, konkrete Fragen zu Anforderungen zu beantworten und somit konkrete Anforderungen zu formulieren. Aufgrund der individuellen Gesprächsführung wurde sich für ein “Teil-standardisiertes Interview” entschieden. Bei einem Teil-standartisiertem Interview gibt es vordefinierte Fragen, aber auch Freiraum für Improvisation und Persönlichkeit. Für ein gutes Interview ist gute Vorbereitung unabdinglich. Daher wurden bereits sämtliche wichtigen Fragen in einem Fragebogen festgehalten. Dieser Interview-Fragebogen liegt in *⟨⟨A Interview-Fragebogen⟩⟩* anbei. Ebenso ist Vorbereitung auf Seiten des Interviewteilnehmers wichtig, weshalb das Interview einen Tag zuvor angesprochen wurde. Um möglichst objektive und unvorbeeinflusste Antworten zu gewährleisten wird darauf geachtet, keine Suggestivfragen oder ähnliches zu stellen [Die SOPHISTen, 2016].

## 4.2. Online-Fragebögen für Stakeholder

Um Fragebögen für Stakeholder formulieren zu können, muss zunächst bekannt sein, wer die Stakeholder sind. “*Ein Stakeholder eines Systems ist eine Person oder Organisation, welche (direkt oder indirekt) Einfluss auf die Anforderungen des betrachteten Systems hat.*” [Pohl and Rupp, 2015]. Daraus ergeben sich die Stakeholdergruppen: “Mitarbeiter WM” und “teilnehmende Weingüter”. Jede dieser Stakeholdergruppen sieht das System aus einer anderen Perspektive [Die SOPHISTen, 2016]. Daher ist es wichtig für jede Stakeholdergruppe einen eigenen Fragebogen zu entwerfen und die bestimmten Perspektiven beim Entwurf der Fragebögen zu beachten. Ebenso ist es wichtig, die wichtigsten Fragen am Anfang zu stellen, da Formulare nicht immer vollständig ausgefüllt werden [Die SOPHISTen, 2016]. Sämtliche Fragen an die Stakeholdergruppe “Mitarbeiter WM” wurden bereits im Interview mit dem Product Owner beantwortet und als Anforderungen festgehalten. Insofern gibt es



schlichtweg keine offenen Fragen, die diese Stakeholdergruppe beantworten könnten. Damit fällt ein Onlinefragebogen für “Mitarbeiter WM” weg. Der Fragebogen der Stakeholdergruppe “teilnehmende Weingüter” liegt in  $\langle\langle B \text{ Online-Fragebogen Weingüter} \rangle\rangle$  bei.

### 4.3. Ergebnisse

Aus dem Interview mit dem Product Owner und dem Fragebogen an die Winzer ergibt ein Pflichtenheft. Dieses ist im Anhang unter  $\langle\langle D \text{ Pflichtenheft} \rangle\rangle$  zu finden. Das Interviewprotokoll und Fragebogenergebnisse sind im Anhang unter  $\langle\langle C \text{ Interview-Protokoll} \rangle\rangle$  und XXXX zu finden.

## 5. Konzeption und Entwicklung

## 6. Ergebnisse

## 7. Ausblick

# Literaturverzeichnis

- [Abraham, 2005] Abraham, M. (2005). *Sustainability Science and Engineering: Defining Principles*. ISSN. Elsevier Science.
- [Bettencourt, 2018] Bettencourt, R. G. (2018). barcode.php - Generate barcodes from a single PHP file. MIT license. <https://github.com/kreativekorp/barcode>. Zugriff: Januar 2023.
- [Day, 2020] Day, R. (2020). QR code/2d barcode generator. <https://github.com/soldair/node-qrcode>. Zugriff: Januar 2023.
- [Die SOPHISTen, 2016] Die SOPHISTen (2016). Die kleine re-fibel. 3.
- [Dougherty and Dunne, 2012] Dougherty, D. and Dunne, D. D. (2012). Digital science and knowledge boundaries in complex innovation. *Organization Science*, 23(5):1467–1484.
- [Ebert, 2019] Ebert, C. (2019). Vorwort zu systematisches re.
- [Github, 2023] Github (2023). Saving repositories with stars. <https://docs.github.com/en/get-started/exploring-projects-on-github/saving-repositories-with-stars>. Zugriff: Januar 2023.
- [Heinemann, 2021] Heinemann, E. (2021). *Vorlesung: Requirements Engineering*. Hochschule Worms.
- [Jung, 2020a] Jung, L. (2020a). jquery-qrcode readme.md. <https://github.com/lrsjng/jquery-qrcode/blob/master/README.md>. Zugriff: Januar 2023.
- [Jung, 2020b] Jung, L. (2020b). kjua dynamically generated QR codes. <https://larsjung.de/kjua/>. Zugriff: Januar 2023.
- [König, 2022] König, W. (2022). *Vorlesung: User Experience*. Hochschule Worms.

- [Li et al., 2016] Li, F., Nucciarelli, A., Roden, S., and Graham, G. (2016). How smart cities transform operations models: a new research agenda for operations management in the digital economy. *Production Planning & Control*, 27(6):514–528.
- [Loebbecke and Picot, 2015] Loebbecke, C. and Picot, A. (2015). Reflections on societal and business model transformation arising from digitization and big data analytics: A research agenda. *The Journal of Strategic Information Systems*, 24(3):149–157.
- [npm, inc, 2020] npm, inc (2020). QR code/2d barcode generator. <https://www.npmjs.com/package/qrcode>. Zugriff: Januar 2023.
- [OpenJS Foundation, 2023] OpenJS Foundation (2023). Command-line API. <https://nodejs.org/api/cli.html#command-line-api>. Zugriff: Januar 2023.
- [Opensource.org, a] Opensource.org. The 2-Clause BSD License. <https://opensource.org/licenses/BSD-2-Clause>. Zugriff: Januar 2023.
- [Opensource.org, b] Opensource.org. The MIT License. <https://opensource.org/licenses/MIT>. Zugriff: Januar 2023.
- [Packagist, 2023] Packagist (2023). BaconQRCode is a QR code generator for PHP. <https://packagist.org/packages/bacon/bacon-qr-code>. Zugriff: Januar 2023.
- [Pagani and Pardo, 2017] Pagani, M. and Pardo, C. (2017). The impact of digital technology on relationships in a business network. *Industrial Marketing Management*, 67:185–192.
- [Parviainen et al., 2022] Parviainen, P., Tihinen, M., Kääriäinen, J., and Teppola, S. (2022). Tackling the digitalization challenge: how to benefit from digitalization in practice. *International Journal of Information Systems and Project Management*, 5(1):63–77.

- [Pohl and Rupp, 2015] Pohl, K. and Rupp, C. (2015). *Basiswissen Requirements Engineering*. dpunkt.verlag, Heidelberg, 4., überarbeitete auflage edition. Description based upon print version of record.
- [Rochko, 2022a] Rochko, E. (2022a). A container class for immutable settings objects. Not a DI container. <https://github.com/chillerlan/php-settings-container>. Zugriff: Januar 2023.
- [Rochko, 2022b] Rochko, E. (2022b). A QR code generator and reader with a user friendly API. PHP 7.4+. <https://github.com/chillerlan/php-qrcode>. Zugriff: Januar 2023.
- [Rochko, 2022c] Rochko, E. (2022c). composer.json. <https://github.com/chillerlan/php-qrcode/blob/main/composer.json>. Zugriff: Januar 2023.
- [Rochko, 2023] Rochko, E. (2023). SVG QR Code Generator. <https://smiley.codes/qrcode>. Zugriff: Januar 2023.
- [Scholzen, 2018] Scholzen, B. (2018). barcode.php - Generate barcodes from a single PHP file. MIT license. <https://github.com/bacon/baconqrcode>. Zugriff: Januar 2023.
- [Tabrizi et al., 2019] Tabrizi, B., Lam, E., Girard, K., and Irvin, V. (2019). Digital transformation is not about technology. *Harvard business review*, 13(March):1–6.
- [Udjaja, 2018] Udjaja, Y. (2018). Ekspanpixel bladsy stranica: Performance efficiency improvement of making front-end website using computer aided software engineering tool. *Procedia Computer Science*, 135:292–301. The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life.
- [Verhoef et al., 2021] Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Qi Dong, J., Fabian, N., and Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. *Journal of Business Research*, 122:889–901.

- [W3Schools, 2023] W3Schools (2023). Sass Introduction. [https://www.w3schools.com/sass/sass\\_intro.php](https://www.w3schools.com/sass/sass_intro.php). Zugriff: Januar 2023.
- [Zimmermann, 2017] Zimmermann, J. (2017). Webpack - A Detailed Introduction. <https://www.smashingmagazine.com/2017/02/a-detailed-introduction-to-webpack/>. Zugriff: Januar 2023.



# Anhang

# A. Interview-Fragebogen

Wie stellen Sie sich den Prozess des Einscannens der QR-Codes beim Entgegennehmen der Flaschen vor? Beschreiben Sie den Ablauf.

Mit was soll dieser Code gescannt werden? Soll der Scanner in der Applikation eingebaut sein, oder soll das System auch mit Drittanbieter-Apps funktionieren?

Von welchem Endgerät wird gescannt?

Welche Fallbacks soll es geben, sollte ein Code nicht scanbar sein? Z.B.: Der Code-Inhalt in Text unter dem Code, der auch von Hand eintippbar sei.

Gegen welche Missbrauchsszenarien sollte der QR-Code geschützt sein? Sollte ggf. ein Passwort nach dem Einscannen verlangt werden? -> Diskurs über versch. Authentifizierungsmethoden und Abwägung über Aufwand der dadurch entsteht.

Was soll der Kunde beim Versand selbst machen?

Wenn es ein PDF zum Ausdruck zum Aufbringen des Empfängers gibt, sollte man auch ein PDF zum Ausdruck des Absenders generieren, falls die relevanten Daten vorliegen?

In welcher Form sollten Mitarbeiter die ausstehenden und eingegangenen Weine sehen? Reicht eine einfache Liste, oder sind Export- und Filtermöglichkeiten erwünscht? Wenn ja: Welche Filter (auch Sortierungen)? Welche Exportformate (Excel kann auch csv öffnen)?

Wenn nun ein Wein als “eingegangen” vermerkt ist, sollte ein Mitarbeiter das rückgängig machen können? Sollte ein Mitarbeiter Weine löschen können? Wenn eines der beiden ja: Einzeln, oder als Bulk-Action? (Bulk-Actions sind teuer/aufwändig umzusetzen)

Welche Informationen soll der Kunde über seine Sendunge(n) sehen?

Soll auch ein Kunde in der Lage sein, seine eigene Weinsendung(en) aus dem System zu löschen oder zu verändern? (Eventuell vertippt man sich) Wenn ja, bis wann? Bis zur Eingangsbestätigung? Ist hierbei eine Bulk-Action wichtig?

Die ELN ist eine inkrementell erhöhte Ganzzahl. Ist der tatsächliche Wert dieser wichtig, oder ist es lediglich wichtig, dass sie eindeutig ist? Der einfachste/günstigste Weg wäre es, sie in der Datenbank als "auto\_increment" zu deklarieren. Dann hätte man niemals, auch über x Auswahlproben hinweggehend, die selbe ELN zwei mal. Das verkürzt und vereinfacht die Entwicklung, den entstehenden Code, und die Nutzererfahrung auf Seiten von Weinland Mosel.

## B. Online-Fragebogen Weingüter

# Jahresauswahlprobe - Registrierungstool

1. Sind Sie mit der aktuellen Arbeitsweise der Anmeldung ihrer Weine zur Jahresauswahlprobe zufrieden?

*Markieren Sie nur ein Oval.*

- ☐ Ja  
☐ Nein

2. Wünschen Sie sich eine Digitalisierung dieser Arbeitsweise, die ihre Formulare verkürzt, an den Rechner verschiebt und es ihnen ermöglicht zu verfolgen, ob ihre Sendungen eingegangen sind?

*Markieren Sie nur ein Oval.*

- ☐ Ja  
☐ Nein

3. Wie viele Weine stellen Sie pro Jahresauswahlprobe ungefähr auf?

---

4. Mit welchem Gerät greifen Sie primär auf das Internet zu?

*Wählen Sie alle zutreffenden Antworten aus.*

- ☐ iPhone
- ☐ Android Phone
- ☐ iPad
- ☐ Android Tablet
- ☐ Laptop/Macbook
- ☐ Desktop-Rechner

5. Haben Sie Zugriff auf einen Drucker?

*Markieren Sie nur ein Oval.*

- ☐ Ja
- ☐ Nein

6. Haben Sie sonstige Wünsche und Anregungen bezüglich der Anmeldung zu Jahresauswahlproben?

---

---

---

---

---

---

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt.

Google

## C. Interview-Protokoll

Interviewer	Leon Etienne (Entwickler)
Befragter	Jochen Stange (Product Owner)
Zeit	27.01.2022, 10:00
Ort	Zoom
Notiz	Dieses Protokoll ist nicht wortlautkorrekt, sondern zeigt lediglich zeitgleich geschriebene Notizen

**Wie stellen Sie sich den Prozess des Einscannens der QR-Codes beim Entgegennehmen der Flaschen vor? Beschreiben Sie den Ablauf.** - Paket kommt an - Wird aufgemacht - Ziehen qr code raus - Haben entweder Handscanner oder handy und scannen den code ein - system gibt an, welche ELN aufgeklebt werden muss - wein wird eingelagert

Die ELN muss im Verein mit der JAP unique sein. Die ELN ist dann im Format ELN-Jahresauswahlprobennummer z.b. 19-1 gesetzt. Das steht für ELN 19, JAP 1. Auf Datenkbankebene unique. Es wird bei 1 anfangen zu zählen.

**Mit was soll dieser Code gescannt werden? Soll der Scanner in der Applikation eingebaut sein, oder soll das System auch mit Drittanbieter-Apps funktionieren?** Smartphone ist am sinnvollsten, da feedback von Webseite Backup-Funktion die ELN händisch einzugeben. Das ginge dann auch mit einem Handscanner.

**Von welchem Endgerät wird gescannt?** Es gibt dedizierte Scanhardware, die sich vor Ort befindet. Das genaue Gerät wird in Kürze festgestellt.

**Welche Fallbacks soll es geben, sollte ein Code nicht scanbar sein? Z.B.: Der Code- Inhalt in Text unter dem Code, der auch von Hand eintippbar sei.** ELN händisch im typo3 backend suchen und bearbeiten.

Die ELN muss also auch unter dem qr code stehen, aber das ist für uns out-of-scope.

Das ist nicht der QR-Code, den wir generieren, sondern der, der beim Einlagern auf die Flasche geklebt wird. Diese lässt Weinland Mosel im Voraus drucken.

**Gegen welche Missbrauchsszenarien sollte der QR-Code geschützt sein? Sollte ggf. ein Passwort nach dem Einscannen verlangt werden? ->Diskurs über versch. Authentifizierungsmethoden und Abwägung über Aufwand der dadurch entsteht.** Verwenden wir einen API-Key. Übermittlung via POST. Eingetragen in qr-code app. Es gibt einen api key. Der soll im backend änderbar sein. Gehashed mit typo3 argon2i. Gespeichert in der Datenbank.

**Gibt es im Lager Wlan?** Out of scope

**Wie wollen Sie Weinanmeldungen zwischen Jahresauswahlproben im Frontend trennen? Wichtig: Aktuelle Proben nach Anmeldeschluss sollten immer noch sichtbar sein und alte Proben ggf. gar nicht mehr angezeigt werden.** Es gibt eine Liste mit JAPs. Da werden verschiedene aufgelistet. Backenduser können die Anmeldung (Weine an JAPs anmelden) an- und abschalten. Vergangene Proben werden nicht angezeigt. Die Probe gilt als vergangen, wenn die Proben probiert wurden. Die Anmeldeöglichkeit und "Probe ist vorbei"-Feststellung im Anmeldetool ist ein Clone vom „active until“ im „access-panel. Mit Dateselector.

**Was soll der Kunde beim Versand selbst machen?** Wenn weinland-mosel-mitglied: Ggf account erstellen mit Mitgliedsnummer <schon implementiert> <muss händisch von WM freigeschalten werden> In account einloggen

Wenn Nichtmitglied: Ggf account erstellen ohne Mitgliedsnummer <muss händisch freigeschalten werden>

Eine JAP auswählen Einen Wein anmelden, Webform zu Weindaten ausfüllen. PDF downloaden, ausdrucken im Paket dem Wein beilegen. Das PDF muss Daten zum Wein beinhalten, und den qr-code mit der ELN. (zur Zuordnung) Nummernschwund, falls Weine gelöscht werden, ist kein Problem. 196-1 darf auf 194-1 folgen. Adressierung wird selbst gemacht. Frankierung auch.



**In welcher Form sollten Mitarbeiter die ausstehenden und eingegangenen Weine sehen? Reicht eine einfache Liste, oder sind Export- und Filtermöglichkeiten erwünscht? Wenn ja: Welche Filter (auch Sortierungen)? Welche Exportformate (Excel kann auch csv öffnen)?** - CSV soll ausgabbar sein (excel format, semicolon separaten) - Filter nicht notwendig, aber probenspezifisch

**Wenn nun ein Wein als “eingegangen” vermerkt ist, sollte ein Mitarbeiter das rückgängig machen können? Sollte ein Mitarbeiter Weine löschen können? Wenn eines der beiden ja: Einzeln, oder als Bulk-Action? (Bulk-Actions sind teuer/aufwändig umzusetzen)** - Was TYPO3 by-default kann, nichts eigenes bauen

**Welche Informationen soll der Kunde über seine Sendung(en) sehen?** - ob sie eingegangen ist, mit emailbenachrichtigung

**Soll auch ein Kunde in der Lage sein, seine eigene Weinsendung(en) aus dem System zu löschen oder zu verändern? (Eventuell vertippt man sich) Wenn ja, bis wann? Bis zur Eingangsbestätigung?** Nach Einreichung (nicht Eingang bei WM) nicht mehr veränderbar und auch nicht löschar.

**Die ELN ist eine inkrementell erhöhte Ganzzahl. Ist der tatsächliche Wert dieser wichtig, oder ist es lediglich wichtig, dass sie eindeutig ist? Der einfachste/günstigste Weg wäre es, sie in der Datenbank als “auto\_increment” zu deklarieren. Dann hätte man niemals, auch über x Auswahlproben hinweggehend, die selbe ELN zwei mal. Das verkürzt und vereinfacht die Entwicklung, den entstehenden Code und die Nutzererfahrung auf Seiten von Weinland Mosel.** - siehe oben

**Weiteres** Es soll einen Single-View für Weine geben, der anhand einer ELN aufrufbar ist. Hier werden bekannte Details gezeigt.

**Zum Formular, das Winzer ausfüllen:** Das Formular besteht aus zwei Schritten:  
Schritt 1: Daten ausfüllen Schritt 2: Zusammenfassung anzeigen entweder zurück,  
weiter bearbeiten, oder submit “wein verbindlich anmelden”

## D. Pflichtenheft

Art	Beschreibung
Constraint	Einbau in brown-field TYPO3
Constraint	Kunden-UI im Frontend
FA	Mitglieder bei WM können sich einen Nutzeraccount erstellen
FA	Nicht-Mitglieder bei WM können sich einen Nutzeraccount erstellen
FA	Angemeldete Nutzer sehen eine Übersicht aller aktiven JAPs
FA	Angemeldete Nutzer können mehrere Weine bei aktiven JAPs anmelden
FA	Eine Weinanmeldung fragt Weindaten mit einem Webform ab
FA	Vor Abschluss des Formulars wird eine Übersicht der Daten gezeigt
FA	Das Formular lässt sich mit einem “Jetzt verbindlich anmelden”-Button abschicken
FA	Angemeldete Nutzer sehen angemeldete Weine für JAPs inklusive Versandstatus
FA	Für angemeldete Weine lassen sich PDF-Dateien herunterladen, die die ELN als Barcode und eine Beschreibung des Weines enthalten
NFA	Das Formular mit Weindaten und dem QR-Code lässt sich auch direkt nach Anmeldung eines Weines herunterladen
FA	Mitarbeiter von WM können eingetragene Weine als TYPO3-Objekte manipulieren
FA	Mitarbeiter von WM können JAPs im Backend zeitgesteuert für Anmeldungen freigeben
FA	Mitarbeiter von WM können JAPs im Backend zeitgesteuert für die Einsicht freigeben
FA	Mitarbeiter von WM können einen gesicherten API-Link anhand einer ELN aufrufen um einen Wein als “eingegangen” zu markieren
FA	Wird ein Wein als “eingegangen” markiert, bekommt der Winzer eine Email
NFA	Der API-Link ist mittels eines API-Keys gesichert
NFA	Der API-Key wird in POST-Daten übergeben
NFA	Die ELN wird url-kodiert übergeben
FA	Mitarbeiter von WM können diesen API-Key im Backend verändern
NFA	Nach Aufruf eines API-Links wird gezeigt, welcher Wein gescannt wurde
NFA	Der Prozess soll auf ein GERÄT_XYZ spezialisiert sein

<b>Art</b>	<b>Beschreibung</b>
FA	Es gibt einen Single-View für Weine, der über eine ELN in der URL erreichbar ist
NFA	Der Wein-Single-View zeigt bekannte Weindaten
FA	Mitarbeiter von WM können Weineintragungen je nach JAP in Semicolon-CSV exportieren