

# Leitlinien für die Umsetzung von regulatorischen Anforderungen für Pandemie Apps

## COMPASS Arbeitspaket 3\_Regulatorische Anforderungen

Dieses Dokument soll dem Umstand Rechnung tragen, dass regulatorische Anforderungen bei Pandemie-Apps ungleich wichtiger sind als bei bisherigen Medical Apps, da viel schneller verlässliche Ergebnisse gefordert werden. Ebenso ist die Entwicklung solcher Apps nicht immer auf die Weise planbar wie das bei Medical Apps außerhalb einer Pandemie der Fall ist.

Version 1.0 \_ 30.04.2021

## Inhaltsverzeichnis

---

<b>Abbildungsverzeichnis</b>	4
<b>Ziele des Dokumentes</b>	5
<b>Kurzzusammenfassung</b>	5
<b>Einleitung</b>	5
<b>Regulatorische Grundlagen</b>	7
Sammlung gängiger Normen im Healthcare-Bereich	8
Normen-Vergleich hinsichtlich wichtiger regulatorischer Themen	8
Fazit	9
<b>Planung des regulatorisch konformen App-Erstellungsprozesses („Konzept“)</b>	11
Anforderungen des Softwarelebenszyklus	11
Rollen sowie Zuordnung zu Projektteam-Mitgliedern	11
Entwicklungsphasen/-Dokumente, Verantwortlichkeiten und Reviews	11
Beteiligte Dienstleister, Schulungen und notwendige SOPs	13
Tracing Prozesse	14
Testkonzept	14
Schnittstellen	14
Freigabe	15
Risikomanagementprozess/-analyse	15
<b>Projektpläne durchgeführter App-Entwicklungsprojekte (2020)</b>	17

<b>Erfahrungen der Teammitglieder</b>	20
<b>Empfehlung</b>	21
<b>Referenzen</b>	22
<b>Anhang</b>	22

## Abbildungsverzeichnis

---

### **Abbildung 1:**

CHART Regulatorische Konformität- Iterativer Ansatz – Hohe Entwicklungsgeschwindigkeit – Effiziente Fehlervermeidung. **7**

### **Abbildung 2:**

Vergleich der relevanten regulatorischen Normen. **9**

### **Abbildung 3:**

Vorgehensmodell der Softwareentwicklung "Wasserfall". **12**

### **Abbildung 4:**

Vorgehensmodell der Softwareentwicklung "agil". **13**

### **Abbildung 5:**

Ergebnisse des Risikomanagementprozesses im Projekt Corona Health. **16**

### **Abbildung 6:**

Entwicklungszeiten Corona Check und Corona Health. **18**

## Ziele des Dokumentes

Dieses Dokument soll dem Umstand Rechnung tragen, dass regulatorische Anforderungen bei Pandemie-Apps ungleich wichtiger sind als bei bisherigen Medical Apps, da viel schneller verlässliche Ergebnisse gefordert werden. Ebenso ist die Entwicklung solcher Apps nicht immer auf die Weise planbar wie das bei Medical Apps außerhalb einer Pandemie der Fall ist.

Regulatorische Anforderungen bestehen dabei nicht nur aus einer Medizinproduktekonformitätseinhaltung, viele weitere Aspekte sind zu berücksichtigen: Barrierefreiheit, Datenschutz, Infektionsschutz, nationale Gesetze, internationale Gesetze, wissenschaftliche Initiativen, Guidelines und mehr.

Das Dokument basiert auf den Erfahrungen der Entwicklung der Medical Apps *Corona Check Screening* und *Corona Health*, sowie auf der Erfahrung und Einschätzungen durch langjährige Tätigkeiten im Bereich regulatorischer Softwareentwicklungsprojekten. Erläutert wird, neben der heutigen Notwendigkeit solcher Medical Apps, ebenso die Wichtigkeit regulatorischer Anforderungen an diese Projekte, um eine sichere, effiziente und schnelle Entwicklung, Verfügbarkeit und Nutzung solcher Apps in Zukunft zu ermöglichen.

Obwohl dieses Dokument einen Erfahrungsbericht bezüglich zweier Apps in diesem Bereich widerspiegelt, kann es als Leitfaden für zukünftige, ähnliche Projekte interpretiert werden. Dieses Dokument gibt einen Überblick über die Vorgehensweise für die Entwicklung einer App unter regulatorischen Anforderungen. Betrachtet werden dabei die einzelnen Projektabschnitte, die beteiligten Personen, sowie die dabei (für die Entwicklung sowie Freigabe/Abnahme) entstehenden Dokumente. Es wird verdeutlicht und visualisiert wie sich regulatorische Anforderungen auf die Entwicklung einer App auswirken bzw. auswirken können. Darauf aufbauend werden entsprechende Empfehlungen und Hinweise für einen reibungslosen Ablauf gegeben.

## Kurzzusammenfassung

Dieser Leitfaden, entstanden in Begleitung der beiden medizinprodukte-konformen Apps *Corona Check Screening* und *Corona Health* zeigt, dass die Entwicklung von Medical Apps unter Einhaltung regulatorischer Anforderungen nicht generell zu einem enormen Mehraufwand oder zu enormen Mehrkosten führen muss. Im Gegenteil, durch genaueste Planungen im Voraus und eine agile Realisierung bzw. Implementierung konnten viele Fehler- und Risikoquellen ausgemerzt werden bevor diese wirklich ins Gewicht fielen. Der Aufwand und die Kosten wurden dadurch auf ein Minimum reduziert. Parallel wurden alle Zielsetzungen erfolgreich erreicht und Fehlerkosten praktisch eliminiert.

Eines der größten Probleme bei der Realisierung und Verbreitung besagter Apps bestand darin, dass die beiden großen Store-Betreiber, für diese Apps, aufgrund der pandemischen Lage zum damaligen Zeitpunkt und kürzlich aktualisierten Richtlinien, einen unverhältnismäßig langen Prüfprozess starteten. Diese lange Phase der Bereitstellung der Apps, in welcher eigentlich keine weitere Entwicklung stattfand, sondern lediglich auf die Freigabe und Indexierung (für die Suchfunktion in den Stores) seitens der Appstores gewartet wurde, sollte bei ähnlichen Entwicklungen ebenfalls berücksichtigt werden. Eine frühzeitige Kontaktaufnahme mit Google und Apple, sowie die eventuelle Einbindung von offiziellen Stellen (wie z. B. entsprechende Ministerien, das RKI oder ähnliche Behörden), erhöhen den Druck auf jene Store-Betreiber und sollten besagtes Vorgehen beschleunigen. Somit sollten zukünftig qualitativ hochwertige, regulatorisch korrekte und kostengünstige Medical Apps möglichst schnell, möglichst vielen Nutzern zur Verfügung gestellt werden können.

## Einleitung

Unabhängig von Pandemie Apps sind regulatorische Anforderungen im Bereich von Medical Apps sehr lange vernachlässigt worden. Im Jahr 2016 listet das Portal Statista 2.5 Millionen Medical Apps in den beiden

prominenten App Stores von Apple und Google. Die Zahl erhöht sich nahezu täglich in Größenordnungen, an die vor 10 Jahren nicht zu denken gewesen wäre. Da bislang regulatorische Rahmenbedingungen für Medical Apps noch nicht besonders im Fokus von Behörden–Audits stehen, kann jeder ohne große Verantwortung eine Medical App in die Stores einstellen.

Dies bringt vor allem zwei Probleme mit sich:

- Zunächst kann wenig darüber ausgesagt werden, ob eine ausgewählte App auch tatsächlich hilft bzw. wirksam ist. Regulatorische Anforderungen können in dieser Hinsicht zwar nicht abschließend unterstützen, aber sie würden dennoch einen Anhaltspunkt geben, ob sich eine infrage kommende App an Standards bezüglich Validierung, Verifikation und die Grundlagen der Datenintegrität hält.
- Das zweite Problem ist, dass Betroffene bei der unüberschaubaren Menge von angebotenen Medical Apps nicht wissen können, welche App tatsächlich die richtige für das vorliegende Bedürfnis ist. Regulatorische Rahmenbedingungen können hier ein Qualitätsmerkmal darstellen und gleichzeitig auch ein Anhaltspunkt sein, um Betroffenen eine bessere Navigation zu ermöglichen und eine zuverlässigere Einschätzung einer App zu bieten.

Zwei Medical Apps, die sich an die gleichen Vorgaben halten, sind somit besser vergleichbar. Daher kann auch ihre Wirkweise besser verglichen werden. Glücklicherweise kümmern sich sowohl der Gesetzgeber als auch viele Initiativen (VDI, DIN, DKE, etc.) zunehmend darum, regulatorische Rahmenbedingungen für Medical Apps zu schaffen. Es existieren auch zunehmend mehr Initiativen, die sich um eine Navigationshilfe für Medical Apps einsetzen, wie z. B. AppRadar. Da jedoch solche Initiativen und Richtlinien erst seit Kurzem existieren, hat der App-Markt (sowohl in Forschung und Wissenschaft, als auch im medizinischen Nutzungsbereich) noch wenig damit begonnen, diese Richtlinien umzusetzen. Dies könnte auch dadurch begründet sein, dass noch zu Wenige dieser Richtlinien wirklich von den entsprechenden Behördenvertretern als Grundlage für deren Audits verwendet und durchgesetzt werden.

Regulatorische Voraussetzungen/Anforderungen dürfen dabei keinen Hemmschuh darstellen, denn Geschwindigkeit ist bei der Entwicklung zukünftiger Apps neben der Verlässlichkeit, den Kosten und der Akzeptanz ein weiterer sehr wichtiger Faktor. Es ist nicht zielführend, wenn eine App gut funktioniert und akzeptiert wird, aber etwaige Regularien und deren Anwendung dafür sorgen, dass die App erst einsatzfähig ist, wenn die eigentliche Pandemie bereits abgeebbt ist.

Die regulatorischen Anforderungen müssen so formuliert werden, dass Sie verständlich sind und die kommende Entwicklungsarbeit nicht beeinträchtigen, sondern diese unterstützen.

Im Folgenden wird ein Erfahrungsbericht inklusive eines abgeleiteten Realisierungskonzepts dargestellt, wie zukünftig bei kommenden Pandemien, Apps schnell, kostengünstig und funktionsfähig entwickelt werden können. Gleichzeitig soll ein hohes Qualitätslevel erreicht werden, um bei der Bevölkerung eine größtmögliche Akzeptanz zu erzeugen.

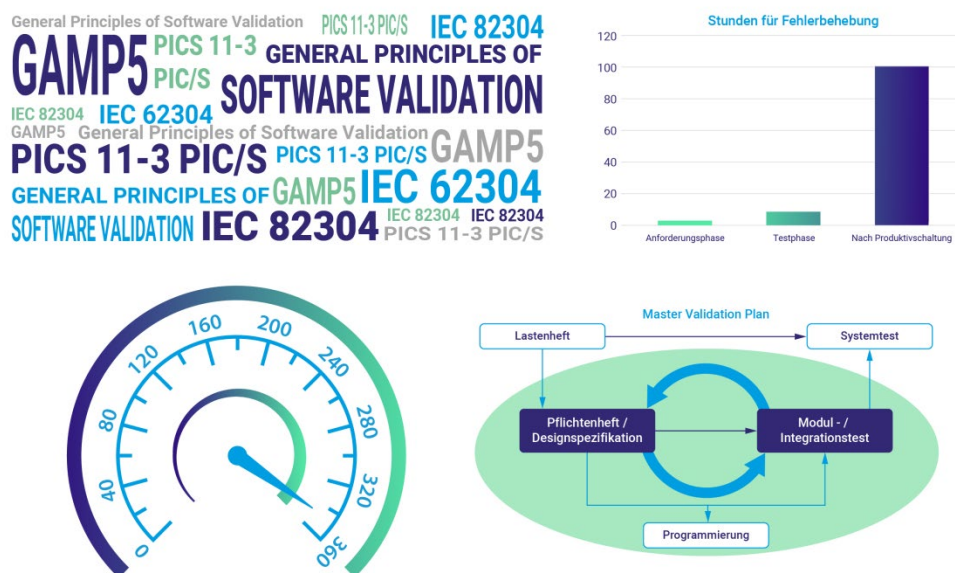
Konkrete Zielsetzungen waren hierbei:

- Aufbau eines harmonisierten regulatorischen Ansatzes, der aus allen Regeln und Normen das Beste herauszieht und somit verhindert, dass „überreguliert wird“, dennoch aber **Konformität mit der Medizinprodukte-Verordnung** aufweist.
- Möglichkeit eines **Iterativen Entwicklungsansatzes** bei gleichzeitiger Einhaltung der notwendigen Regularien.

- Schaffung eines einfachen und machbaren Vorgehens, damit die **hohe Entwicklungsgeschwindigkeit** nicht gebremst wird, gleichzeitig jedoch alle notwendigen Anforderungen in einer App-Lösung umgesetzt werden.

Und als eine Art Nebeneffekt – die Schaffung eines Ansatz mit dessen Hilfe durch eine ausführlichere Anforderungsphase eine effiziente Fehlervermeidung möglich wird:

- Verlässliche Projektzeitpläne können mit zuverlässigen Kostenschätzungen erstellt werden.
- Reduzierung von Fehlerkosten (d. h. eine ausführliche Anforderungsphase verhindert dabei Fehler im Test oder nach Produktivschaltung).



**Abbildung 1: CHART Regulatorische Konformität– Effiziente Fehlervermeidung– Hohe Entwicklungsgeschwindigkeit – Iterativer Ansatz**

Die folgenden Kapitel zeigen zunächst auf, welche existierenden Normen sinnvoll sind und sich als regulatorische Basis für die Entwicklung von Medical Apps im Allgemeinen und einer Pandemie App im Speziellen eignen. Die Schaffung/Umsetzung eines pragmatischen Vorgehensmodells ist dabei die Grundlage um regulatorisch korrekt-erstellte Apps in den Verkehr zu bringen und so das Vertrauen der Bürgerinnen und Bürger zu gewinnen.

## Regulatorische Grundlagen

Da sich weder Medizinproduktegesetz (MPG) noch die seit 2017 in Kraft getretene Medizinprodukte Verordnung (MPV) näher zum Thema Anforderungen an eine Medical App äußern, galt es zunächst gängige Normen und Regelwerke zu ermitteln und hinsichtlich relevanter Themengebiete zu analysieren bzw. miteinander zu vergleichen. Hieraus wurde eine geeignete und harmonisierte, regulatorische Grundlage geschaffen, auf deren Basis Projektleiter, Entwickler und Qualitätsmanager gemeinsam eine App entwickeln konnten, die vollständig, (nahezu) fehlerfrei und qualitativ hochwertig war.

Die folgenden Regelwerke aus dem Bereich Healthcare wurden dabei analysiert.

### **Sammlung gängiger Normen im Healthcare-Bereich**

- IEC 62304 – Medizingeräte -Software – Software-Lebenszyklus-Prozesse
  - Die Norm IEC 62304 legt die Anforderungen an Software-Lebenszyklus-Prozesse für die Entwicklung von medizinischer Software und Software in Medizinprodukten fest. Die in dieser Norm vorgeschriebenen Prozesse, Aktivitäten und Aufgaben bilden einen gemeinsamen Handlungsrahmen für alle Lebenszyklusprozesse für Software im Bereich Medizinprodukte.
- GAMP5 – „Good Automated Manufacturing Practice Supplier Guide for Validation of Automated Systems in Pharmaceutical Manufacture“
  - Dieser Leitfaden hat sich zum Standardregelwerk für die Validierung computergestützter Systeme in der pharmazeutischen Industrie (Hersteller und Zulieferer) entwickelt. Das GAMP5-Regelwerk hat jedoch keine gesetzliche Bindung. Deshalb sind davon abweichende Formen der Validierung von computerisierten Systemen möglich, was bei vielen Systemen sinnvoll ist.
- General Principles of Software Validation – Regelwerk der amerikanischen Behörde Food and Drug Administration (FDA) – US-Behörde für Lebens- und Arzneimittel
  - Dieser Leitfaden umreißt allgemeine Validierungsprinzipien, die die FDA als anwendbar für die Validierung von Software für medizinische Geräte oder die Validierung von Software, die für den Entwurf, die Entwicklung oder die Herstellung von mediz. Geräten verwendet wird, erachtet.
- PICS 11-3 PIC/S Guide to Good Manufacturing Practices (GMP)
  - Der PIC/S Guide to GMP ist die Grundlage für GMP-Inspektionen. Insbesondere sein Anhang 11, "Computergestützte Systeme", wird bei der Inspektion solcher Systeme verwendet. Zweck dieses Dokuments ist es, Empfehlungen und Hintergrundinformationen über computergestützte Systeme bereitzustellen, die den Inspektoren bei der Ausbildung und während der Inspektion computergestützter Systeme behilflich sein sollen. Das Dokument soll allen "Good Practice"-Inspektoren dienen, die für die Inspektion von Anwendungen im regulierten pharmazeutischen Sektor zuständig sind.
- IEC 82304 Gesundheitssoftware – Teil 1: Allgemeine Anforderungen für die Produktsicherheit
  - Diese Norm behandelt zum einen „Software als Medizinprodukt“ inklusive Mobile Medical Apps, sowie weitere Health Software (außerhalb der Anwendung der Verordnung (EU) 2017/745 über Medizinprodukte). Durch die IEC 82304-1 wurde eine Lücke im Bereich der normativen Abdeckung der Validierung und Dokumentation von Software geschlossen, die ohne spezielle Hardware in Verkehr gebracht wird. Die Schlüsselpunkte dieser Norm sind Anforderungen an die Produktsicherheit (SAFETY) als auch an die Informationssicherheit (SECURITY) von Health Software Produkten. Dies beinhaltet ebenso Forderungen an die Gebrauchstauglichkeit (USABILITY) und Gebrauchsanweisung (INSTRUCTION FOR USE). Die Norm fordert die Einhaltung eines Software-Entwicklungsprozesses und verweist normativ auf die IEC 62304/A1:2015.

### **Normen-Vergleich hinsichtlich wichtiger regulatorischer Themen**

Nach der erfolgreichen Sammlung der Regelwerke wurde im Team ein Themenkatalog erstellt, in dem alle wichtigen Themen und regulatorischen Forderungen zum Thema App-Entwicklung gesammelt wurden. Man einigte sich auf folgende Punkte: Softwarelebenszyklus mit den Themen Validierungsplan/Dokumentation (inkl. Verifikation), Risikomanagement, sowie die Einbeziehung von Dienstleistern und die Schulung von Mitarbeitern. Im Anschluss daran wurde untersucht, ob die genannten Themenschwerpunkte von den verschiedenen Regelwerken behandelt wurden. Wenn dies für ein Regelwerk zutraf, dann wurden die Anforderungen des



jeweiligen Themenschwerpunkts im betreffenden Regelwerk auf Umfang und Verhältnismäßigkeit der dort geforderten Aktivitäten überprüft.

Zusammengefasst kann gesagt werden, dass alle genannten Regelwerke die festgelegten Themen behandelten und in Umfang und Verhältnismäßigkeit der dort geforderten Aktivitäten jeweils sehr ähnlich waren.

Die folgende Tabelle soll dazu einen Gesamtüberblick geben.

Subjects/Standard	General Principles of SW-Validation (2002)	IEC 62304-2015	PIC/S 11-3-2007	IEC 82304-2016	AAMI TIR36:2007	GAMP5:2008
1) Software Life Cycle – general, incl. validation plan	Sections 4.4, 5.1	Sections 5, 5.1	Section 9	Section 6.1	Chapter 4	Sections 2, 3, 4, 3.3, 6.1.6, M1
2) Software Life Cycle – V-model documentation (Planning-RE-Design-Test)	Sections 5.1-5.2.6	Sections 5.2-5.8	Sections 5-14	Sections 5, 6	Sections 4.1-4.3	Section 4
3a) Risk Management (classification)	Section 5.2	Section 4.3	Section 15 (reference to GAMP)	Section A3 (subitem to 5.), as well as reference to IEC 62304 and thus to section 7	Section 7B	Sections 4.2.6, M4
3b) Risk Management (continuous risk management process)	Section 5.2	Section 7	Sections 5-14	Section A3 (subitem to 5.), as well as reference to IEC 62304 and thus to section 7	Section 7B	Sections 5, M3
4) Software Lifecycle – Software Maintenance and Support (CR Development)	Sections 4.7, 5.2.7	Sections 6.1-6.3	Sections 17-18	Sections 1.2, 8.2 as well as reference to IEC 62304 and thus to sections 6.1-6.3	Section 4.4	Sections 3.1, 4.2, 4.3
5) Software Life Cycle – Decommissioning	Section 5.1	-	Sections 14.2, 17.2	Sections 4.2, 7.2.2.9, 8	Section 4.5	Sections 3.1, 4.2, 4.4, M10
6) Involvement of software service providers	Section 6.3	Referred in Section 4.1 – by reference to QM system	Section 11	Reference to IEC 62304 and thus indirectly to section 4.1 – reference to QM system	Indirectly in section 2.2 – reference to QM system	Sections 6.1.4, 7
7) Training of employees	Section 2.4 – Reference to QM system	Referred in Section 4.1 – by reference to QM system	Section 22	Chapters 6.1.e, as well as reference to IEC 62304 and thus indirectly to section 4.1	Indirectly in section 2.2 – reference to QM System	Section 6.1.3

**Abbildung 2: Vergleich der relevanten regulatorischen Normen**

## Fazit

An den zuvor definierten Themenschwerpunkten wurden auch nach Beendigung der Analyse festgehalten. Nach weiterer Untersuchung entschied man, dass mit der IEC 82304 ein Regelwerk verwendet werden konnte, welches genau für solch einen Fall (Standalone Softwaresysteme und Gesundheits-Apps) konzipiert wurde. Ein wesentliches Problem war jedoch, dass die Forderungen dieser Norm immer noch recht umfangreiche Folgen auf den Umfang einer möglichen Dokumentation und die Entwicklungsgeschwindigkeit eines App Projekts besaßen.

Es wurden deshalb wesentliche Punkte aus den weiteren Normen im geplanten Entwicklungsprozess aktualisiert und dieser wurde danach in einem Risikobasierten Ansatz zusätzlich noch sogenannt „getailort“, also noch final zugeschnitten (man könnte auch sagen „entschlackt“). Somit wurde den anderen Normen Rechnung getragen und trotzdem konnten die regulatorischen Vorgaben für ein App-Entwicklungsprojekt auf ein sinnvolles Mindestmaß an verursachendem Aufwand reduziert werden. D. h. der initiale Entwicklungsprozess beinhaltete das „Gute“ aus allen zuvor dargelegten Normen und enthält dennoch alle relevanten Vorgaben der IEC 82304-1. Dieses konkrete konzeptionelle Vorgehen ist im Kapitel „Planung des regulatorisch konformen Softwareerstellungsprozesses“ detailliert beschrieben. Es kann als Vorlage für kommende Projekte verwendet werden.

Ergänzend ist zu sagen, dass die Einhaltung dieses regulatorisch-konformen Entwicklungsprozesses letztlich zwingend notwendig war, damit die entwickelten Medical Apps in den Stores von Google (Google Play) und Apple (App Store) aufgenommen werden konnten. Dies geschah allerdings nicht in der Form, dass man ein etwaiges Freigabe-Zertifikat direkt an Google oder Apple schickte, sondern dieses Freigabe-Zertifikat bildete wiederum die Basis um

- einerseits ein positives Votum einer Ethikkommission (sie prüft den Inhalt letztlich u. a. nach ethischen, medizinischen und wissenschaftlich Gesichtspunkten)
- und andererseits die Unterstützung einer öffentlichen Stelle (RKI, LGL, ...)

zu erhalten. Beides waren letztlich zentrale Bedingungen der Store-Betreiber um eine Medical App als solche in die Stores aufzunehmen.

Um jedoch die finale Zustimmung von Ethikkommission und öffentlicher Stelle zu erhalten, musste neben einem Freigabe-Zertifikat noch ein weiterer geeigneter Nachweis erbracht werden, dass die zuvor genannte(n) Norm(en) - inkl. des harmonisierten Zuschnitts, bei einem Softwareprojekt auch tatsächlich eingehalten wurden.

Dies geschah in Form von sogenannten Validierungsdokumenten, Schulungsnachweisen oder auch Standard Operating Procedures (SOPs). Sie mussten diesen Stellen als „Beweis“ für die Erfüllung von MPG-/ MPV-Vorgaben bzw. deren (oben genannten) zugrundeliegenden Normen vorgelegt werden. Wie sich dieser Prozess gestaltet bzw. welche Dokumente gefordert werden und wie deren grundlegende Inhalte sind, beschreibt das folgende Kapitel.

## Planung des regulatorisch konformen App-Erstellungsprozesses („Konzept“)

Der Nachweis, dass regulatorische Anforderungen bei einem App-Entwicklungsprojekt erfüllt wurden, geschieht durch das Bereitstellen der im Kapitel zuvor genannten Dokumente. Damit diese ordnungsgemäß erbracht werden können, ist es erforderlich, dass das Entwicklungsprojekt regulatorisch geplant wird.

Da sämtliche hier genannten Regelwerke einen risikobasierten Ansatz empfehlen, bildet dieser auch in der Planung die Basis für das weitere Vorgehen.

Näheres ist in Kapitel „Risikomanagementprozess/-analyse“ zu finden.

### Anforderungen des Softwarelebenszyklus

Bei der Planung sind neben den Entwicklungsphasen und den korrespondierenden Dokumenten (z. B. Anforderungsphase = Lastenheft, Abnahmetest = Systemtest, etc.), Dienstleister, Schulungen, Testkonzept und das Vorgehen beim Requirement-Tracing (Sicherstellung, dass alle Anforderungen auch am Ende tatsächlich in der Software enthalten sind) zu definieren. Zuvor müssen jedoch erst einmal die Rollen, deren Verantwortlichkeiten im Projekt sowie deren Zuordnungen zu den aktuellen Projektmitgliedern festgelegt werden.

### Rollen sowie Zuordnung zu Projektteam-Mitgliedern

Folgende Rollen sollte es in einem SW-Entwicklungsprojekt MINDESTENS geben:

- **Projektmanagement:**  
Verantwortlich für die Durchführung und Leitung der gesamten Produktentwicklung.
- **Qualitätsmanagement:**  
Verantwortlich für die Sicherung der Softwarequalität, der regulatorischen Anforderungen, sowie die Archivierung der Dokumentation
- **Entwicklung:**  
Verantwortlich für Entwicklungsplanung sowie die zugehörigen Tests (außer Systemtests)
- **Test:**  
Verantwortlich für den gesamten Systemtest (Planung und Durchführung) sowie das Risikomanagement. Hinweis: Risikomanagement kann durchaus eine eigene Rolle darstellen.
- **Hauptverantwortlicher:**  
Haupt-Verantwortlicher für das Gesamte Projekt. Erteilung der finalen Freigabe

Aktuelle Teammitglieder müssen ihre jeweilige Rolle im Projekt erhalten. Notwendige Schulungen müssen dabei noch festgelegt werden (s. dazu auch unten den Punkt „Beteiligte Dienstleister, Schulungen und notwendige SOPs“).

Wenn dies alles definiert wurde, gilt es nun die Entwicklungsphasen (inkl. den korrespondierenden Dokumenten) und die Rollen-Verantwortlichkeiten für Dokumente sowie die Review-Planung festzulegen.

### Entwicklungsphasen/-Dokumente, Verantwortlichkeiten und Reviews

Da App-Entwicklungsprozesse normalerweise eine überschaubare Komplexität besitzen, können bei der Planung regulatorisch geforderte Entwicklungsschritte in der Regel zusammengefasst werden.

Als Basis für die weitere Planung des Entwicklungsprozesses wurde ein harmonisierter Ansatz aus der IEC 62304 Klasse B und dem GAMP5 Kat. 4 (konfigurierbare Software) verwendet.

Die Entwicklungsphasen des Projekts mit den zugehörigen Dokumenten gliedern sich demnach wie folgt:

- Planungsphase (Dokumenttitel: Softwarequalitätsmanagementplan oder auch Master Validierungsplan)
- Anforderungserhebung (Dokumenttitel: Lastenheft oder Anforderungsspezifikation),
- Zusammengefasste Funktions-Beschreibung + Design (Dokumenttitel: kombiniertes Pflichtenheft-/Designspezifikation),
- Programmierung (Dokumenttitel: hier entsteht kein eigenes Dokument)
- Zusammengefasster Modul-/Integrationstest (Dokumenttitel: Kombiniertes Modul-/ Integrationstest mit Spezifikation/Protokoll/Bewertung)
- System-Test (Dokumenttitel: Systemtest mit Spezifikation/Protokoll/Bewertung)
- Freigabe (Dokumenttitel: Freigabeprotokoll z. B. für Ethikkommission und Storebetreiber)

Der normale Wasserfallprozess (hier das sogenannte Vorgehensmodell der Softwareentwicklung oder auch V-Modell genannt) mit den korrespondierenden Dokumenten würde sich in der Regel wie folgt darstellen:

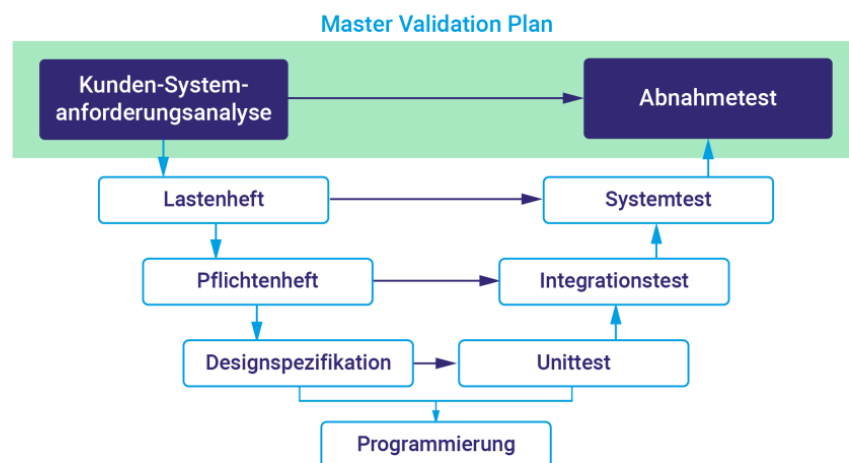


Abbildung 3: Vorgehensmodell der Softwareentwicklung "Wasserfall"

Da wir jedoch aus (Entwicklungs-)Geschwindigkeitsgründen einen agilen Softwareentwicklungsprozess empfehlen, wird der tatsächliche Entwicklungsprozess wie folgt geplant:

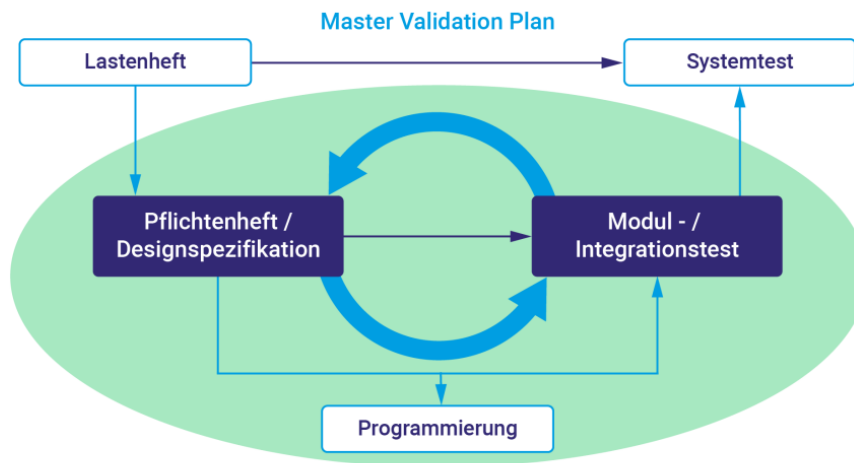


Abbildung 4: Vorgehensmodell der Softwareentwicklung "agil"

In der Planung wird dabei definiert, dass zunächst ausführlich die Anforderungen erhoben werden müssen, um dann die Basis (das sogenannte Backlog) zu besitzen, um diese Anforderungen in einem agilen Prozess „Design – Programmierung – Test“ (grünlich hinterlegter Bereich) umzusetzen. Sobald alle Anforderungen in den diversen Sprints (kurze Entwicklungszyklen) umgesetzt und getestet wurden, müssen diese final im Systemtest getestet werden. Alle Validierungs- und Verifikationsdokumente können dabei solange im Entwurfsstatus gelassen werden, bis die finale Freigabe ansteht. Zwischentestergebnisse nach jedem Sprint werden dabei dokumentiert. Gegen Ende des Projekts können diese Dokumente schrittweise freigegeben werden.

Nachdem nun Rollen, Entwicklungsprozess und Verantwortlichkeiten definiert wurden, müssen die Anforderungen an Dienstleister, Schulungen und erforderliche SOPs formuliert werden.

#### **Beteiligte Dienstleister, Schulungen und notwendige SOPs**

Beteiligte Dienstleister werden gelistet, deren Anforderungen an Qualifikation und Verfügbarkeit formuliert und ggf. organisatorische Rahmenbedingungen definiert. Dabei ist vor Beginn des Projekts vom Projektleiter sicherzustellen, dass die beteiligten Mitarbeiter (sowohl eigene als auch Mitarbeiter des Dienstleisters) ausreichend Erfahrung besitzen und in den regulatorischen Grundlagen zum Thema App-Entwicklung/-Validierung geschult sind.

Ebenso sollen alle Schulungen definiert werden, die von den jeweiligen Projektmitgliedern (gemäß ihren Rollen) gefordert sind. U. a. sollten die Teammitglieder das hier im Master Validation Plan (oder auch Softwarequalitätsmanagementplan genannt) beschriebene Vorgehen gelesen und verstanden haben. Ergebnisse müssen in einem Protokoll oder in Schulungsausweisen/-pässen dokumentiert werden.

Zusätzlich werden alle SOPs dargestellt, die für die Nutzung und den Betrieb einer App notwendig sind.

Wenn dies alles geschehen ist, muss nun sichergestellt werden, dass die App auch tatsächlich alle geforderten Requirements erfüllt. Die Überprüfung geschieht dabei formal in einem sogenannten Requirement Tracing Prozess.

## Tracing Prozesse

Es ist zu planen, dass das Requirement Tracing (Verfolgung der Anforderungen über den Entwicklungsprozess hinweg) in drei Durchgängen durchgeführt werden muss. Somit ist das Vorhandensein aller Lasten im kombinierten Pflichtenheft/Designspezifikation und je mindestens einem Test in den jeweiligen Testspezifikationen (Modul-/ Integrationstestspezifikation - Systemtestspezifikation) gesichert.

Es entstehen dabei folgende Dokumente:

- Trace-Matrix Lastenheft – Pflichtenheft/Designspezifikation
- Trace-Matrix Lastenheft – Systemtestspezifikation
- Trace-Matrix Pflichtenheft/Designspezifikation – Modul-/Integrationstestspezifikation

Nachdem die Forderungen an den Tracing Prozess abgeschlossen sind, muss noch zu guter Letzt ein schlankes Testkonzept formuliert werden.

## Testkonzept

Die durchzuführenden Tests wurden bereits im Entwicklungsprozess geplant und die zugrundeliegenden Dokumente genannt. Die Modul-/Integrationstests können in einer simulierten Umgebung stattfinden (empfohlen wird hier baldmöglichst im Produktivumfeld zu testen). Die finalen Systemtests müssen in einer Produktivumgebung passieren.

Die Tests werden in der Regel 2-stufig durchgeführt. In Phase 1 sollen die sogenannten Modul-/ Integrationstests erfolgen und in Phase 2 die Systemtests (oder auch Abnahmetests genannt). Dabei werden die Modul-/Integrationstests pro Iteration immer wieder vollständig definiert (und freigegeben), ausgeführt und die Testergebnisse protokolliert und bewertet. Können in einer Iteration KEINE vollständigen Tests durchgeführt werden, müssen die durchzuführenden Regressionstests vorab festgelegt werden.

Die ausgedruckte Testspezifikation stellt dabei das Testprotokoll dar, welches von den Testern vervollständigt werden muss. Nach Abschluss der Tests überprüft dann jeder Tester die Vollständigkeit und Gültigkeit der Einträge und bestätigt dies durch Datum, Kurzzeichen, Name und seiner Unterschrift auf der ausgedruckten Testspezifikation bei jeder der durch ihn abgetesteten Anforderungen. Alternativ, kann die Zuordnung von Unterschrift und Kurzzeichen auch einmalig tabellarisch erfasst werden, dann würde es ausreichen wenn der Tester die Vollständigkeit und Gültigkeit der Einträge durch Datum und Kurzzeichen pro Testfall bestätigt.

Hinweis: Wo sinnvoll, ist das Testergebnis noch durch einen oder mehrerer Screenshot(s) zu belegen.

Die Testergebnisse sind dann durch die jeweiligen Verantwortlichen zu bewerten. Dies kann in einem extra Dokument passieren. Nach Abschluss der erfolgreichen Systemtests kann die Software freigegeben werden. (s. Kapitel „Freigabe“).

## Schnittstellen

Nicht jede App besitzt Schnittstellen zu anderen Systemen. Ist dies jedoch der Fall, dann müssen auch diese Schnittstellen „validiert und verifiziert“ werden. Dies passiert nun ausnahmsweise auf Basis des Regelwerks GAMP5. Es wird hier das dort beschriebene Vorgehen bei „Kategorie 3: Nicht-konfiguriertes Produkt“ angewendet. Dies bedeutet, dass Anforderungen an die relevante Schnittstelle und die korrespondierenden Systemtestfälle formuliert werden müssen. Die Dokumentation kann in einem einzigen Dokument, in einem

sogenannten Project Summary (auch „Fact File“ oder „Projektsteckbrief“ genannt) geschehen. Dieses Dokument enthält dabei einen kurzen Organisationsteil, das Lastenheft, die Abnahmetestspezifikation inkl. Protokoll, die Traceability Matrix und Risikomanagement-Elemente. Inhaltlich müssen vor allem die zu importierenden Daten (Datenstruktur) definiert, Übertragungsart/-weg (u. a. sicherer/zeitnaher Daten-Transfer, Interpretationslogik, und Konfiguration) festgelegt und die Orte der Datenspeicherung (Datenübernahme bzw. -Ablage) formuliert werden.

### **Freigabe**

Nach Abschluss der durchgeführten Schulungen und Erstellung der zuvor genannten Dokumente, inkl. der erfolgreichen Systemtests, sowie der Formulierung notwendiger SOPs und ggf. erstellter Project Summaries für Schnittstellen, kann die App freigegeben werden: Dazu muss als Nachweis für Behörden ein offizielles Freigabedokument erstellt werden. Es muss eine Liste aller entstandener und freigegebener (Nachweis-)Dokumente enthalten, sowie von allen hauptverantwortlichen Projektteilnehmern unterschrieben werden.

### **Risikomanagementprozess/-analyse**

Der für ein Entwicklungsprojekt im regulierten Umfeld stattfindende Risikomanagementprozess sollte immer mindestens auf Basis der ICH Q9 Qualitätsrisikomanagement (ICH = International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use) passieren. Dabei werden zu Beginn des Projekts und während des Entwicklungsprozesses ständig Risiken

- gesammelt,
- bewertet,
- mitigiert und
- überwacht.

Hinweis: Der hier dargestellte Prozess ist zu Erklärungszwecken vereinfacht dargestellt.

Dabei sind Risiken mindestens in folgende Gruppen zu unterteilen:

- Produktrisiken
- Patientenrisiken
- Risiken in Bezug auf Datenintegrität

Im Risikomanagementprozess sollten alle Stakeholder vertreten sein. Dies können z. B. Projektleiter, Entwickler, Q-MA aber auch Anwender, Behördenvertreter oder Personen mit IT-Entwicklungserfahrung sein.

Wenn diese Risiken nicht IT-technisch mitigiert werden können, kann auch angestrebt werden die Risiken durch Schulungen und ausführliche SOPs (Standard Operational Procedures) auf ein erforderliches Minimum zu reduzieren. Da die eigentlichen Risiken von App zu App und Situation zu Situation unterschiedlich sein können, wurde auf eine Darstellung oder Zusammenfassung möglicher Risiken verzichtet. Um jedoch einen Eindruck der Effektivität und Effizienz des Risikomanagement-prozesses zu geben, wurde im Folgenden beispielhaft dessen Ergebnisse im Projekt *Corona Health* tabellarisch dargestellt:

Schweregrad / Häufigkeit	1 Theoretisch möglich	2 Selten	3 Gelegentlich	4 Wahrscheinlich	5 Oft
3 Katastrophal	0	0	0	0	0
2 Kritisch	1	9	3	0	0
1 Wichtig	0	0	1	0	0

**Matrix vor der Minimierung**

Schweregrad / Häufigkeit	1 Theoretisch möglich	2 Selten	3 Gelegentlich	4 Wahrscheinlich	5 Oft
3 Katastrophal	0	0	0	0	0
2 Kritisch	0	0	0	0	0
1 Wichtig	14	0	0	0	0

**Matrix nach der Minimierung**

**Abbildung 5: Ergebnisse des Risikomanagementprozesses im Projekt Corona Health**

Die linke Tabelle zeigt die Einstufung der Risiken vor der Mitigierung, die rechte Tabelle nach der Risikomitigierung. Dabei ist darauf zu achten, dass keines der Risiken in den roten definierten Bereich (nicht zu tolerierender Bereich) fällt und die Risiken soweit als möglich auf ein tolerierbares Minimum zu reduzieren sind.

Das Risikomanagement MUSS in zukünftigen IT-Entwicklungsprojekten für Medical Apps einen wesentlichen Bestandteil darstellen.

Nachdem bisher die Planungen des regulatorisch konformen Softwareerstellungs- und des Risikomanagementprozesses beschrieben wurden, sollen die folgenden Kapitel zeigen, wie sich die gründliche risikobasierte Planung auf den Projektverlauf und den Projekterfolg der Apps *Corona Check Screening* und *Corona Health* ausgewirkt haben.



## Projektpläne durchgeführter App Entwicklungsprojekte (2020)

Im Jahr 2020 entstanden nacheinander die Apps *Corona Check Screening* und *Corona Health*. Beide Projekte durchliefen den zuvor beschriebenen Planungs- und Softwareerstellungsprozess. Abfolgen und interne Laufzeiten waren bei beiden Projekten (abhängig vom Projektumfang) sehr ähnlich. Was die dennoch sehr stark voneinander abweichenden, unterschiedlichen Projektdauern verursachten, waren externe Faktoren.

Hier sind zuerst die Freigaben von Apple und Google für deren Stores zu nennen. Diese waren mit 11-50 Tagen sehr lange. Übertroffen wurden diese Zeiten jedoch durch die darauf folgenden Freischaltungen von zentralen Suchbegriffen wie z. B. „Corona“ und „Covid“.

Von der Freigabe durch die Ethikkommission bis zu dem Moment als die Apps im Google Play Store unter den Suchbegriffen „Corona“ oder „Covid“ gefunden wurden, vergingen bei der 2. App (*Corona Health*) bis zu 82 Tage.

(Hinweis: Die reine App Entwicklungszeit inkl. Genehmigung durch die Ethikkommission dauerte 86 Tage).

Wie ist das erklärbar?

Zwischen 03/2020 – 08/2020 änderten Google und Apple die Store-Zulassungsbedingungen sehr stark. Stellten die beiden Konzerne zunächst keine außergewöhnlichen Anforderungen an sogenannte Corona Apps, so änderte sich dies bereits nach wenigen Wochen und das Votum einer Ethikkommission wurde verpflichtend. Nach weiteren 3-4 Monaten war auch dies nicht mehr ausreichend und eine Empfehlung einer sogenannten Offiziellen Stelle (in unserem Fall das RKI) wurde benötigt.

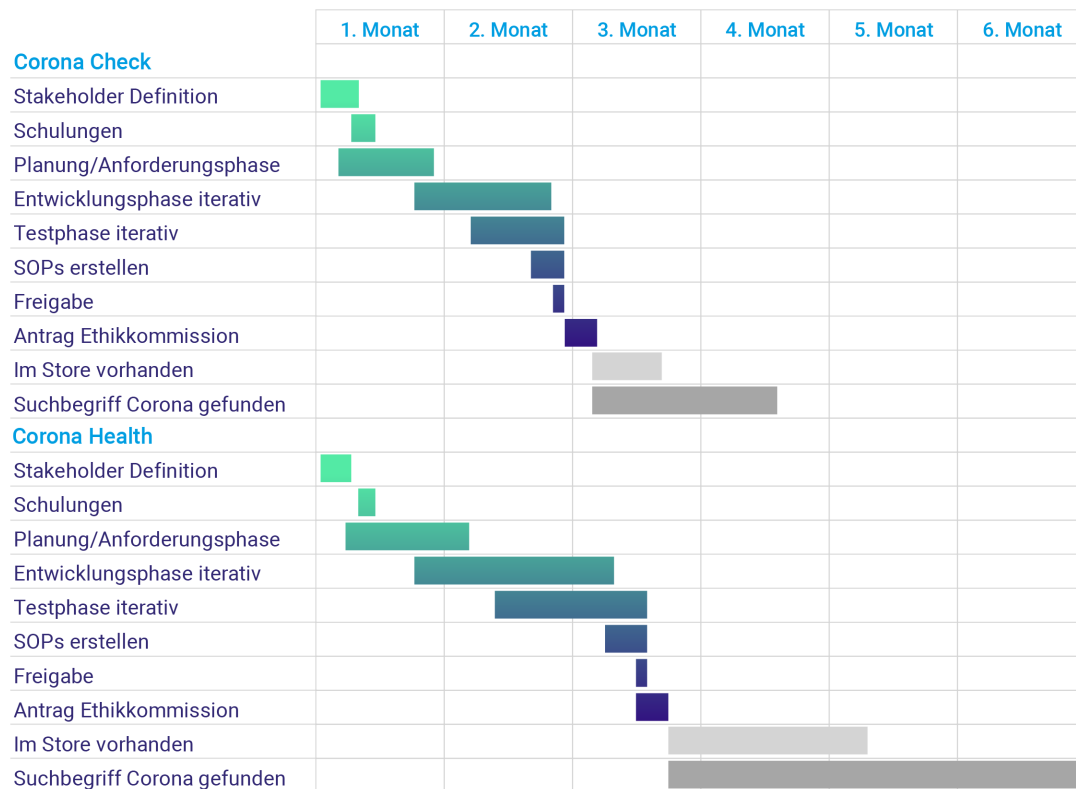
Beide Voraussetzungen wurden rechtzeitig durch die Projektleitung (Prof. Pryss) geplant und erfüllt, und dennoch führten dann plötzlich „interne Freigaberunden“ der Store-Betreiber (Dauer jeweils zwischen 5-6 Tagen) dazu, dass zwischen Erfüllung der Vorgaben und tatsächlicher Storefreigabe (inkl. Suchbegriffen) letztlich 3 Monate (!) vergingen.

Warum ist die Freischaltung für zentrale Suchbegriffe so wichtig?

Erscheint eine App bei einem allgemeinen Suchbegriff wie z. B. „Corona“ nicht in der Trefferliste, dann wird Sie praktisch überhaupt nicht in den Stores gefunden. In unserem Fall musste im Google Play Store der Suchbegriff „CoronaHealth App“ eingegeben werden, damit die App *Corona Health* überhaupt gefunden und angezeigt wurde. Als die App schließlich für die zentralen Suchbegriffe freigeschaltet wurde, vervielfachten sich die Download-Zahlen.

Auf den Verlauf dieser internen Freigaberunden konnte das Projektteam nur sehr begrenzt Einfluss nehmen! Erst nach massiver Intervention aller Projektbeteiligten, vor allem von Prof. Pryss, sowie dem Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung, unter Empfehlung von Bundesminister Gerd Müller, gelang es auf einer sogenannten Whitelist im Google Play Store gelistet zu werden. Damit wurden die beiden Apps *Corona Health* und *Corona Check Screening* plötzlich zwei von insgesamt nur sieben Apps, die bei Google (weltweit) angezeigt werden, wenn Nutzer die Suchbegriffe Corona oder auch Covid eingegeben haben.

Berücksichtigt man die hier genannten Zeiten, und stellt sie alle zueinander ins Verhältnis, dann ergibt sich folgendes Schaubild:



**Abbildung 6: Entwicklungszeiten Corona Check und Corona Health**

Es ist schematisch gut zu sehen, dass der Unterschied der Entwicklungszeiten beider App-Entwicklungsprojekte nur etwa zwei Wochen beträgt. Dies ist an dem blauen Balken der Zeilen „Antrag Ethikkommission“ erkenntlich. Es können an dieser Stelle bereits gut die beiden „Treiber“ der langen Projektlaufzeit identifiziert werden und eine der Hauptempfehlungen wird bereits an dieser Stelle deutlich. Baldmöglichste Kontaktaufnahme mit Google- und Apple-Verantwortlichen an höchster Stelle um nach Freigabe der Software und Ethikvotum sofort in den Stores aufgenommen und auch gefunden zu werden. Beim oben genannten App-Projekt *Corona Health* hätte dies, wie zuvor angedeutet, die Projektlaufzeit um nahezu 50% verkürzt!

Berücksichtigt man nun ausschließlich den Entwicklungszeitraum, so ergibt sich auf Basis der beiden vergangen Projekte für eine App mit mittleren Umfang und Komplexität der folgende Zeitplan (die Firmenauswahl muss hier bereits stattgefunden haben):

- Bestimmung der Projektbeteiligten (Tag 1– Tag 8)
- Schulungsphase (Tag 9 – Tag 11)
- Planungsphase und Requirement Phase (Tag 8 – Tag 35)
- Entwicklungsphase (Tag 30 – Tag 72)
- Testphase (Tag 44 – Tag 79)
- SOPs erstellen und freigeben (Tag 70 – Tag 79)
- Freigabe (Tag 79)
- Antrag Ethikkommissionen (Tag 79 – Tag 86)
- Freigabe durch die Store-Betreiber (Tag 86 – Tag ?)
- Unter Suchbegriff Corona bei Google gefunden (Tag 86 – Tag ?)

Bei dem hier gezeigten Projektablauf ist bemerkenswert, dass eine relativ lange Anforderungsphase und regulatorische Rahmenbedingungen die tatsächliche Entwicklungsgeschwindigkeit kaum bremsen. Im Gegenteil, durch die relativ lange Anforderungsphase und regulatorischen Vorgaben können Unklarheiten und Probleme rechtzeitig erkannt und behoben werden. Daraus resultierende potentielle Fehler(-quellen) werden rechtzeitig erkannt und ausgemerzt (s. dazu Abbildung 1: CHART Regulatorische Konformität – Effiziente Fehlervermeidung – Hohe Entwicklungsgeschwindigkeit – Iterativer Ansatz). Somit wird eine solide, widerspruchsfreie, testbare und vollständige Anforderungsdokumentation geschaffen auf deren Basis folgendes gelingt:

Kurze Entwicklungszyklen in Verbindung mit umfangreichen Tests und schnellen Feedbackschleifen, so dass in nur 3 Iterationen die App Entwicklung abgeschlossen werden kann.

Die Validierungsdokumentation hilft bei Weiterentwicklungen der Apps, zusätzlich etwaige Einarbeitungszeiten zu verkürzen. Bestehende Tests können durch vorhandene umfangreiche Testspezifikationen detailliert geplant und ggf. erweitert werden und bilden somit die Grundlage, um die Tests schließlich erfolgreich durchzuführen.

Entwicklungszeiten und Erfüllung regulatorischer Anforderungen sind allerdings bei einem Projekt nur die eine Seite der Medaille. Akzeptanz der Projektmitglieder und Bewertung des Entwicklungserfolgs sind die andere Seite.

Im Anschluss an die Projekte wurden die Teammitglieder zu Ihren Erfahrungen befragt.

## Erfahrungen der Teammitglieder

Die wichtigsten Stakeholder des Entwicklungsprozesses waren Entwickler, Projektleitung, Risikomanager/QM, sowie die Mitarbeiter der sogenannten offiziellen Stellen wie dem RKI oder die Mitarbeiter der Universitäten Würzburg, Regensburg und Ulm.

Im Folgenden wurden die einzelnen Vertreter zu Ihren Erfahrungen in den beiden App-Entwicklungsprojekten *Corona Health* und *Corona Check Screening* befragt:

- Prof. Rüdiger Pryss (Projektleiter):
  - „Meine bisherigen Erfahrungen mit der Entwicklung von Software im regulierten Umfeld waren, dass dies zeitaufwendig und sehr dokumentationsintensiv ist. Der bei der Entwicklung von *Corona Health* und *Corona Check Screening* genutzte regulatorische Ansatz hat im Gegensatz dazu kaum zu Mehraufwand geführt, gleichzeitig Fehler vermieden und somit die Qualität der Software erhöht. Die Akzeptanz beim Freigabeprozess durch die Ethikkommissionen hat dies beide Male bestätigt.“
- Winfried Schlee (Risikomanager/Qualitätsmanager):
  - „Durch die ausgedehnte Anforderungsphase und das umfangreiche Risikomanagement traten während der Entwicklung kaum Fehler auf.“
- Carolin Cordes (RKI):
  - „Für eine IT-fachfremde Projektbeteiligte wie mich, war es beeindruckend anzusehen, wie schnell aus unseren Anforderungen eine fertige App entstanden ist.“
- Carsten Vogel (Entwicklung IOS):
  - „Ich habe zu Beginn gedacht, dass uns die Regularien sehr viel Zeit und Aufwand kosten, aber das Gegenteil war der Fall. Durch das iterative Vorgehen und die kompetente Projektleitung, waren wir genauso schnell wie geplant und das ist bei einem App-Entwicklungsprojekt dieses Ausmaßes sehr erstaunlich gewesen.“
- Fabian/Julian Haug (Entwicklung Android)
  - „Durch die gute Anforderungs-/Planungsphase von Prof. Pryss mussten die Anforderungen nur einmal umgesetzt werden und Nacharbeiten wurden weitestgehend vermieden.“
- Johannes Schobel (Entwicklung Anwendung/Datenbanken)
  - „Dadurch, dass wir auf einem bestehenden Entwicklungsframework aufgesetzt haben, musste das Rad nicht komplett neu erfunden werden. Die Implementierungsphase war dadurch im Verhältnis äußerst kurz.“

Zusammenfassend zu den hier genannten Erfahrungen ist hervorzuheben, dass die Projektmitglieder durch Ihr Know-How und Ihre Erfahrungen, sowie Ihr außerordentliches Engagement einen großen Teil dazu beitrugen, dass beide App-Entwicklungsprojekte im zeitlich vorgegeben Rahmen und innerhalb der geplanten Kosten abgeschlossen werden konnten. Nach Analyse der beiden Projekte ergibt sich daraus für zukünftige App-Projekte im regulierten Umfeld folgende Empfehlung.

## Empfehlung

Auf Grund jahrelanger Erfahrungen im Bereich App-Entwicklung sowie regulatorischer Softwareentwicklung im Allgemeinen, konnten die Verantwortlichen der Projekte *Corona Health* / *Corona Check Screening* die tatsächlichen Entwicklungsdauern auf ein Minimum begrenzen (s. oben) und einem gesamten Entwicklungsaufwand von deutlich unter 1 Mio. Euro erreichen. Die vom Aufwand her komplexere *Corona-Warn-App* hatte laut Telekom<sup>1</sup> im Vergleich dazu eine Realisierungszeit von ca. 50 Tagen, allerdings ohne regulatorische Konformität und mit einem Budget in Höhe von ca. 20 Mio. Euro (inkl. Wartung/Betreuung).

Als Mitarbeiter der Normungsgruppe 811-3 der DKE (Deutsche Kommission für Elektrotechnik und Elektronik) (seit 2012) und Verantwortlicher in mehr als 200 regulatorisch relevanten Softwareentwicklungsprojekten schätze ich zukünftige App Entwicklungsprojekte wie folgt ein:

Bei richtiger Planung und unter Berücksichtigung der regulatorischen Konformität, sowie einem guten Requirement Engineering Prozess sollte sich die Entwicklungsdauer durch die Einhaltung der regulatorischen Vorgaben max. um ca. 10% verlängern. Dabei zeigen die Erfahrungen, dass sich durch die Zusammenarbeit erfahrener App-Entwickler, guter RE-Fachleute, sowie regulatorisch erfahrener und pragmatischer Qualitätsmanager, Entwicklungszeiten sogar beschleunigen lassen und dies bei gleichzeitig verbesserter Produktqualität.

Die folgenden, konkreten Zielsetzungen, die sich die Projektleitung zu Beginn beider Projekte gestellt haben, konnten vom Team alle erreicht werden:

- **Konformität mit der Medizinprodukte-Verordnung**
- **Iterativer Entwicklungsansatz** und damit verbundene
- **hohe Entwicklungsgeschwindigkeit**
- **Verlässliche Projektzeitpläne** und
- **niedrige Kosten** bei praktisch keinen Fehlerkosten

In Summe sind die beiden Corona Apps jeweils in weniger als 4 Monaten entstanden, mit weniger als 1 Mio. Euro Kosten, inkl. Genehmigungen deutscher Ethikkommissionen.

Eine frühzeitige Kontaktaufnahme mit Google und Apple, um einerseits schnell in die Stores aufgenommen zu werden und andererseits, um gut von potentiellen Anwendern gefunden zu werden, ist der einzige Punkt der zu Beginn der beiden Projekte nicht ausreichend geplant werden konnte.

Wenn nun noch offizielle Stellen (wie z. B. das RKI oder das Bundesministerium für Gesundheit) rechtzeitig in den Entwicklungsprozess eingebunden werden und politischen Druck auf Google und Apple ausüben können, dann sollten der Bevölkerung neuartige Medical Apps in zukünftigen Krisensituationen

- schnell,
  - qualitativ hochwertig,
  - regulatorisch korrekt und
  - kostengünstig
- zur Verfügung gestellt werden können!

---

<sup>1</sup> Quelle: <https://www.telekom.com/de/medien/medieninformationen/detail/telekom-sap-corona-warn-app-in-50-tagen-programmiert-602162> (letzter Zugriff 29.04.2021)

## Referenzen

<https://www.corona-health.net/> (letzter Zugriff 29.04.2021)

<https://www.coronacheck.science/de/> (letzter Zugriff 29.04.2021)

## Anhang

**Folgende Universitätskliniken des  
Netzwerks Universitätsmedizin  
nehmen am COMPASS-Projekt teil:**

Charité – Universitätsmedizin Berlin  
Universitätsmedizin Göttingen  
Universitätsmedizin Mainz  
Universitätsklinikum Würzburg  
Uniklinik Köln  
Universitätsklinikum Münster  
Universitätsklinikum Regensburg  
Universitätsklinikum Ulm  
Universitätsklinikum Erlangen

**Ansprechpartner für weitere Fragen:**

COMPASS Koordinierungsstelle  
[compass@unimedizin-mainz.de](mailto:compass@unimedizin-mainz.de)



<https://num-compass.science>



@CompassNum