



Portfolio

Technische Projektplanung (DLMDWWTPL01)

im Master Studiengang Data Science

**Rettungsdienst Berliner Rotes Kreuz
MachineLearning-Einsatz in der Bereitschaftsplanung
Projektplan**

Verfasser: Georg Grunsky

Matrikelnummer: IU14072015

Tutor: Florian Wacker

Datum: 25. Februar 2025

I Abstract

Der Berliner Rot Kreuz Rettungsdienst (DRK Berlin) projiziert den Einsatz eines Machine Learning Systems zur effizienten Planung eines Bereitschaftsdienstplanes für Einsatzfahrer:innen. Auf Basis einer erfolgten Use Case Analyse, stellt diese Arbeit den Projektplan vor und behandelt sowohl die Projektstruktur und -teamplanung als auch die Zeitplanung, die Kostenschätzung, die Stakeholderanalyse und das Risikomanagement des Projektes.

Für den Projektmanagement-Ansatz wird eine hybride Vorgehensweise verfolgt. Agile Methoden, wie Scrum, werden mit klassischen Elementen des V-Modell XT kombiniert. Das schafft eine klare Projektstruktur mit definierten Phasenübergängen und Planungswerkzeugen und gewährleistet gleichzeitig die Vorteile der flexiblen und iterativen Entwicklung. Das V-Modell XT wurde auf das notwendigste beschränkt um ein kleines Projektteam, einen geringen „Overhead“ und eine modulare Struktur zu ermöglichen.

Mit den verwendeten Elementen des V-Modell XT wurde in manchen Bereichen zwar über die Vorgaben der Aufgabenstellung hinausgeschossen, die Dokumentation des Rahmenprozesses bietet jedoch eine hervorragende Unterstützung bei gleichzeitiger Ermutigung zur Flexibilität. Man bekommt das Gefühl, für den Projektstart „bereit“ zu sein. Der, in der Aufgabenstellung verlangte, Projektstrukturplan kommt in der Dokumentation des V-Modell XT (Angermeier et al. 2024) nicht vor. Leider ist auch in den vorgestellten Modellen des Studienmoduls „Management von IT-Projekten“ (IU Internationale Hochschule 2024), das diesem Modul üblicherweise vorangeht, kein Projektstrukturplan abgebildet. Den ersten zufriedenstellenden Anhalt dazu lieferte Wikipedia (2022). Beim Projektplan finden die Anforderungen der Aufgabenstellung sowie das V-Modell XT aber wieder zusammen. Als mögliche Ansatzpunkte wurden weiters The PM Minimalist Quick Start Guide (Greer 2011) sowie die Lektüre von How Big Things Get Done (Flyvbjerg / Gardner 2023) herangezogen, diese lieferten jedoch nicht die gehoffte Unterstützung für die Aufgabe.

Die inhaltliche Vorgabe des Projektes ist es, kosteneffizient die tägliche Anzahl des benötigten Bereitschaftspersonals vorherzusagen, gleichzeitig jedoch nie zu wenig Einsatzfahrende als Reserve vorzusehen. Das System und das Projekt bauen sich rund um diese Vorgabe auf. Die Planung sieht vor mit einem Entwicklungsteam in mehreren, teils parallelen Sprints, diese Minimalanforderung umzusetzen, sinnvoll um weitere Funktionen zu erweitern und das System, mit entsprechender Hardware und benutzerfreundlichen Frontends, in eine stabile Umgebung einzubetten, die sich gut in den IT-Betrieb des DRK Berlin integrieren lässt. Die wahrscheinlich langen Lieferzeiten der Hardwarekomponenten geben der Entwicklung einen zeitlichen Rahmen für die Umsetzung der Sprints. Inhaltlich wird dieses Projekt nicht als „zeitkritisch“ eingestuft. Gegebenenfalls wirkt sich daher eine Verzögerung zwar auf die, der Kostenstelle des Projektes zugeschriebenen, Personalkosten jedoch nicht auf sonstige Aspekte und Abhängigkeiten der Umsetzung aus. Der Fokus der Entwicklung liegt, auch in der Priorisierung der Arbeitspakete, auf den, für das Machine Learning Modell relevanten Aufgaben. Das Vorhersagemodell selbst bildet das Herzstück der Aufgabe. In der Zeit- und Kostenschätzung wurde

daher der Einsatz eines/r weiteren Data Scientist zwar erwogen, jedoch auch wieder verworfen.

Anhand der exemplarisch dargestellten Risiken, ist auch ersichtlich, dass nicht nur ein potentieller Reputationsschaden für das DRK (aufgrund einer zu niedrig prognostizierten Anzahl nötiger Bereitschaftsfahrender und damit mangelnder Einsatzbereitschaft) das Projekt gefährden kann, sondern auch eine mögliche Ablehnung durch die Nutzer, die durch die Umsetzung mehr Verantwortung übernehmen müssen als in den derzeitigen Arbeitsprozessen. Dies sind u.a. zwei wesentliche Punkte, denen im Rahmen der Projektumsetzung, bei der Gestaltung des Modells und der Regelung der zukünftigen Arbeitsprozesse, unbedingt Rechnung getragen werden muss. Die zukünftigen Nutzer werden daher bereits ab Projektbeginn sowohl für das Requirements Engineering als auch für die Qualitätssicherung hinzugezogen.

I.1 Making of

Ausgangspunkt dieser Arbeit war ein MachineLearning Szenario, das dem Modul „Model Engineering“ (Pak 2024) dieses Studienganges entnommen wurde. Dank der Zustimmung der zuständigen Tutoren wurde es möglich, dasselbe Thema in drei unabhängigen aber inhaltlich dennoch aneinander anschließenden Modulen zu bearbeiten. Dadurch ergab sich eine durchgängige Bearbeitung der Problemstellung durch mehrere Entstehungsphasen eines Machine Learning Systems hindurch.

1. **Modul „Data Science UseCase“** : Analyse des Anwendungsfalls (ML-Canvas) bis zur Präsentation für die Freigabe durch das Präsidium.
2. **Modul „Technische Projektplanung“** : die gegenständliche Arbeit
3. **Modul „Model Engineering“** : Die Umsetzung des Vorhersagemodells mit bereitgestellten Trainingsdaten

Die ganzheitliche Betrachtungsweise der Thematik ließ bereits jetzt einen besseren Einblick in die Prozesse und Herausforderungen der Softwareprojektentwicklung mit Machine Learning Aspekten entstehen.

Für die technische Bearbeitung der Aufgabenstellung wurden mehrere Werkzeuge verwendet. In vorangegangenen Modulen hat sich bereits die Kombination eines Git-Repositories für die LATEX-Dateien der Bearbeitung bewährt. Zusätzlich wurde probeweise das Repository auch einem Git-Projekt hinzugefügt um die Zeitschätzung des Projektes in Form von Iterations, Meilensteinen und Issues über das Git-Projekt zu gestalten. Dies erwies sich zwar als machbar, aber nicht als „sauber“, da im Repository, die tatsächlichen Issues der Portfoliophasen mit den fiktiven Issues des Projektes vermischt wurden. Eine Testversion der Software „objectiF RPM“ (microTool GmbH 2024) unterstützte zwar die Entwicklung eines Verständnisses für die Herausforderungen in der Projektplanung, eine sinnvolle Herangehensweise und auch für Details der Umsetzung (wie zB Reviewmöglichkeiten für Anforderungen und „Slicing“ von Use Cases), war jedoch für die tatsächliche Verwendung im Rahmen der Portfolioaufgabe zu umfangreich und hätte den Rahmen gesprengt. Das GANTT-Diagramm wurde anschließend mit dem relativ intuitiv gehaltenem Tool „YouTrack“ von JetBrains erstellt. Gerade, die Möglichkeit die Arbeitspakete hierarchisch zu strukturieren und zeitlich einfach anzupassen, machten das GANTT-Diagramm zu einem wertvollen Werkzeug, dass gute Einblicke in die Erfordernisse der Projektentwicklung bot.

Inhaltsverzeichnis

I	Abstract	II
I.1	Making of	III
1	Konzeptvorstellung	1
2	Projektstrukturplan und Projektteam	2
2.1	Projektstrukturplan	3
2.2	Team	7
3	Projektplan	8
3.1	Zeitplanung	8
3.1.1	Definition von Meilensteinen	8
3.1.2	Erstellung eines GANTT-Diagramms	12
3.1.3	Änderung der Teamkonfiguration	14
3.2	Definition von „fertig“	14
3.3	Kostenschätzung	15
4	Stakeholder Analyse	17
5	Risikomanagement	19
5.1	Risikoidentifikation	19
5.2	Risikomatrix	19
5.3	Risikoanalyse und -bewertung	21
5.3.1	Risiko 1: Reputationsschaden	21
5.3.2	Risiko 2: Mangelnde Akzeptanz bei Nutzern	22
5.3.3	Risiko 5: Lieferverzug der Hardware	23
6	Literaturverzeichnis	24
7	Abbildungsverzeichnis	25
8	Tabellenverzeichnis	26

1 Konzeptvorstellung

Die DRK Rettungsdienst Berlin GmbH ist Teil der Notfallrettung in Berlin und Partner des Landes Berlin. Der Dienst besetzt Rettungs- und Intensivtransportwagen und absolviert jährlich tausende Alarmeinsätze im Berliner Stadtgebiet. Täglich sind hierbei mehrere Einsatzfahrende im Dienst. Akutell werden pro Tag zusätzlich 90 weitere Fahrer:innen in Bereitschaft gehalten, um bedarfsgerecht agieren zu können. Aufgrund einer Annahme der Planungsstelle, dass der Bedarf an Personal in Bereitschaft saisonalen Schwankungen unterliegt und hier Kosten gespart werden könnten, wurde eine Use Case Analyse für den Einsatz eines Machine Learning Ansatzes zur effizienteren Gestaltung des Bereitschaftsdienstplanes erstellt (Grunsky 2024). Die Umsetzung des Systems wurde nun vom Präsidium des Landesverbandes Berliner Rotes Kreuz projektiert. Nachdem vor Kurzem, zur Optimierung des Datenmanagements, ein Data Scientist eingestellt wurde, hat das Präsidium die „Make or Buy“ Entscheidung bereits getroffen und dieses Projekt zur Feuertaufe des Data Scientist erklärt.

Diese Software berechnet bis zum 10. Tag jedes Monats den täglichen Bedarf an Bereitschaftspersonal für den Folgemonat aufgrund von historischen Daten des Rettungsdienstes. Wichtig ist vor allem, dass niemals zu wenig Personal in Bereitschaft gehalten wird, aber trotzdem nur so viel wie notwendig um Kosten zu sparen. Aufgrund der Use Case Analyse soll die projektierte Software zu Beginn des Monats aktualisierte Daten zur Anzahl an Notfällen, eingesetzten Einsatzfahrer:innen, etc. den bestehenden historischen Daten hinzufügen, daraus ein Regressionsmodell lernen, dieses anschließend auf den Folgemonat anwenden und die Anzahl der benötigten Bereitschaftsfahrenden pro Tag der Planungsstelle zur manuellen Weiterbearbeitung bereitstellen. Das Modell soll darüber hinaus möglicherweise noch mit weiteren Features angereichert werden. Zum Beispiel können Tagestemperaturdaten die Saisonalität um eine weitere Dimension ergänzen und gegebenenfalls transparentere Abhängigkeiten aufzeigen. An Wochen- und Feiertagen verhält sich die Bevölkerung anders als an Arbeitstagen. Das gilt es auch in einem Vorhersagemodell zu berücksichtigen. Besondere Wetterbedingungen führen zu einer kurzfristigen Anpassung der Vorhersage und einer Meldung an die Planungsstelle, damit diese rechtzeitig reagieren kann. Und zu guter Letzt erhöhen bevorstehende Veranstaltungen (nach Größe kategorisiert) ebenso den Bedarf an Bereitschaftspersonal und können somit zusätzliche Merkmale für die Vorhersage liefern. Eine laufende Selbstevaluierung der getätigten Vorhersagen des Modells unter Einbeziehung aktueller Wetterprognosen und geplanter Veranstaltungen für das aktuelle Monat korrigiert gegebenenfalls die vorab getroffene Vorhersage und das Planungspersonal wird mit einer Notification hingewiesen, den Bereitschaftsdienstplan kurzfristig anzupassen.

Alles in allem hilft das Machine Learning System dem Rettungsdienst mit einer möglichst akkuraten Vorhersage, unnötige StandBy-Kosten zu reduzieren und dem Anspruch der Wirtschaftlichkeit gerecht zu werden. Diese ist ein essentieller Faktor, um den Rettungsdienst zu erhalten, und somit sicherzustellen, dass auch in Zukunft Menschen in Not geholfen wird.

2 Projektstrukturplan und Projektteam

Im folgenden Abschnitt werden die Teilaufgaben und Arbeitspakete des Projektstrukturplans (Wikipedia 2022) aufgelistet. In Klammern neben der Aufgabe ist eine Zuordnung derselben zu einer betrieblichen Organisationseinheit angeführt. Daraus lässt sich in weiterer Folge die notwendige Zusammenstellung eines Teams ableiten.

Neben den grundlegenden und, gemäß Aufgabenstellung, vorgegebenen Projektplanungswerkzeugen orientiert sich die Projektplanung an einer vereinfachten Variante des Softwareprozessmodell-Rahmenwerks V-Modell XT und verbindet dieses mit Scrum-Elementen. Das Rahmenwerk bietet gut dokumentierte und nachvollziehbare Projektabwicklungsprozesse während ein agile Entwicklung rasche, ergebnisgetriebene Projektprodukte ermöglicht. Aufgrund der relativ kleinen Projektgröße wird der Projektleitung gleichzeitig die Aufgabe des Productowners zugewiesen.

Aus Sicht des Rahmenwerkes handelt es sich beim gegenständlichen Projekttyp um ein Systementwicklungsprojekt (AG/AN), in der Variante „AG-AN-Projekt mit Entwicklung, Weiterentwicklung oder Migration“, das keine Trennung zwischen der Auftraggeber- und Auftragnehmerseite vorsieht. Die durch das Rahmenwerk vorgegebenen Vorgehensbausteine für diese Projekttypvariante und weitere anzuwendende Projektmerkmale sind:

- Projektmanagement
- Qualitätssicherung
- Konfigurationsmanagement
- Problem- und Änderungsmanagement
- Anforderungsfestlegung
- Systemerstellung
- Lieferung und Abnahme
- Kaufmännisches Projektmanagement
- Projektgegenstand: Softwareentwicklung
- Betriebsübergabe

Die Grundidee dieser Projekttypvariante ist, dass Anwenderanforderungen bereits zu Beginn des Projekts relativ umfassend abgesteckt werden und Änderungen über das Änderungsmanagement bzw. über die Iterationen der Systemerstellung geplant werden. Damit wird das System in einzelnen Inkrementen entworfen, realisiert und ausgeliefert. Das hat den Vorteil, dass der Anwender frühzeitig in den Besitz einer Vorstufe des Systems gelangt und sich damit die Prinzipien einer agilen Entwicklung anwenden lassen (Angermeier et al. 2024, S.280).

Da eine vollumfängliche Anwendung der oben genannten Vorgehensbaustände vermutlich den Rahmen des Projektes übersteigen würde, lehnt sich der nachfolgende Projektstrukturplan lediglich an diese an und versucht gleichzeitig den Projektoverhead auf ein Minimum zu beschränken.

Die Arbeitspakete für die Entwicklungssprints werden auf Basis der vorausgegangenen Use Case Analyse abgeleitet und implizieren jeweils auch die dazugehörigen Unit-Tests. Erweitert werden diese

Arbeitspakete um Entwicklungsbesonderheiten aus den Machine-Learning Operations, wie die u.a. Abbildung 1 zeigt (Tamburri 2020, S.3). Zur Priorisierung sind die Arbeitspakete der Entwicklung im Projektstrukturplan bereits nach der MoSCoW-Methode bewertet und farbkodiert hervorgehoben (*must*), (*should*), (*could*), (*won't*).

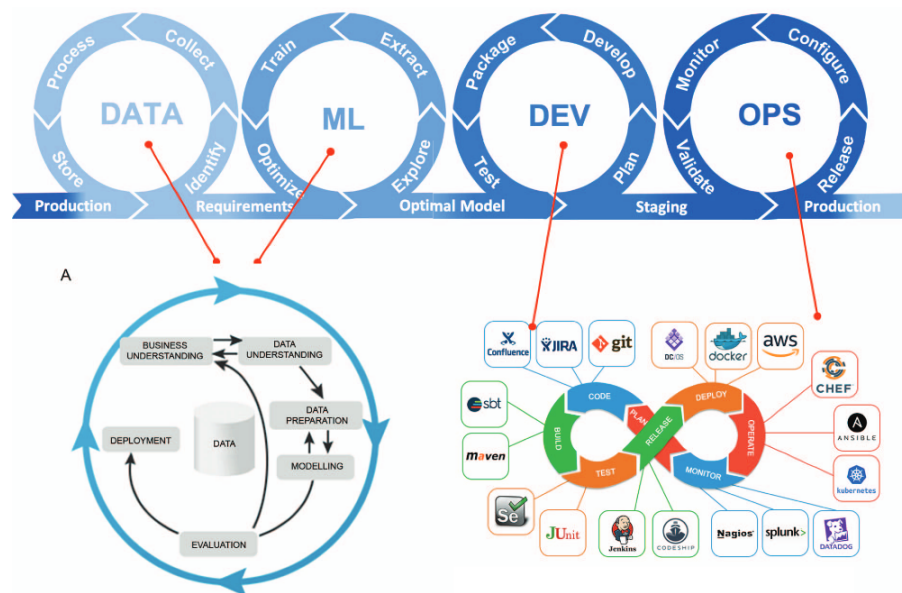


Abb. 1: ML-Ops Workflow
(Tamburri 2020, S.3)

2.1 Projektstrukturplan

1. Projektmanagement

1.1. Planung und Steuerung

- 1.1.1. Stakeholder-Analyse erstellen (*Projektleitung*)
- 1.1.2. Anforderungen festlegen (*Projektleitung, Stakeholder*)
- 1.1.3. Review der Anforderungen organisieren (Anforderungsbewertung) (*Projektleitung*)
- 1.1.4. Liste der Arbeitspakete erstellen (*Projektleitung*)
- 1.1.5. Arbeitspakete priorisieren (*Projektleitung*)
- 1.1.6. Entscheidungspunkte / Projektphasenübergänge definieren (*Projektleitung*)
- 1.1.7. Projektdurchführungsstrategie definieren (*Projektleitung*)
- 1.1.8. Projektteam zusammenstellen (*Projektleitung*)
- 1.1.9. Zeitplanung erstellen (*Projektleitung, beteiligte Fachbereiche*)
- 1.1.10. Kaufmännisches Projektmanagement
 - 1.1.10.1. Kostenschätzung erstellen (*Projektleitung, beteiligte Fachbereiche, Controlling*)
 - 1.1.10.2. Projektkostenkontrolle organisieren (*Projektleitung, Controlling*)

1.2. Berichtswesen

- 1.2.1. Projektfortschrittsmeetings planen (*Projektleitung*)
- 1.2.2. Konfigurationsmanagement
 - 1.2.2.1. Projektdokumentation organisieren (*Projektleitung*)
 - 1.2.2.2. Produktbibliothek verwalten (*Projektleitung*)
- 1.2.3. Projektstatusbericht erstellen (*Projektleitung*)
- 1.2.4. Projekttagbuch bzw. -protokolle führen (*Projektleitung*)
- 1.2.5. Projektabschlussbericht erstellen (*Projektleitung*)
- 1.2.6. Projekthandbuch / Qualitätssicherungshandbuch erstellen (*Projektleitung*)
- 1.2.7. Qualitätssicherungsbericht erstellen (*Projektleitung*)
- 1.2.8. Änderungsanträge erfassen (*Projektleitung*)
- 1.2.9. Prüfberichte erstellen (*Projektleitung*)
- 1.3. Risikomanagement
 - 1.3.1. Risikoliste erstellen (*Projektleitung*)
- 2. Systemerstellung
 - 2.1. Entwicklungsplanung
 - 2.1.1. Detailliertes Requirements Engineering (*Entwicklung*)
 - 2.1.2. Backlog erstellen (*Productowner/Projektleitung*)
 - 2.1.3. Allgemeine Sprintplanung vorbereiten (*Entwicklung*)
 - 2.1.4. Softwaredokumentation organisieren (*Entwicklung*)
 - 2.1.5. Code-Review-Prozesse implementieren (*Entwicklung*)
 - 2.1.6. SCRUM-Organisation
 - 2.1.6.1. Änderungen aus Änderungsanträgen beschließen (*Entwicklungsteam*)
 - 2.1.6.2. Sprint-Backlog erstellen (*Entwicklungsteam*)
 - 2.2. Datenmanagement
 - 2.2.1. Externe (betriebsfremde) Datenquellen spezifizieren und dokumentieren (*Data Scientist*)
 - 2.2.2. Datenschema für DRK Daten erstellen (*Planungsstelle*)
 - 2.2.3. Datenschema (extern) für Tagestemperaturdaten erstellen (*Data Scientist*)
 - 2.2.4. Datenschema (extern) für Wetterdaten erstellen (*Data Scientist*)
 - 2.2.5. Datenschema (extern) für Wochenend- und Feiertagsdaten erstellen (*Data Scientist*)
 - 2.2.6. Datenschema (extern) für besondere Wetterbedingungen erstellen (*Data Scientist*)
 - 2.2.7. Datenschema (extern) für Veranstaltungsdaten (*Data Scientist*)
 - 2.2.8. Daten gem. Datenschema automatisch abrufen/erfassen (*Entwicklung*)
 - 2.2.9. Daten gem. Datenschema vorverarbeiten (*Data Scientist*)

2.2.10. Vorverarbeitete Daten speichern (Entwicklung)

2.2.11. Datenbankdesign (Entwicklung)

2.3. Machine-Learning (ML) System entwickeln

2.3.1. Beschaffung von Testdaten aus externen Quellen (Data Scientist)

2.3.2. Beschaffung von Testdaten aus internen Quellen (Data Scientist, Entwicklung)

2.3.3. Datenexploration (Data Scientist)

2.3.4. Verknüpfung externer und interner Daten (Data Scientist)

2.3.5. Konzeption des ML-Ansatzes (Data Scientist)

2.3.6. Evaluierung verschiedener ML-Algorithmen (Data Scientist)

2.3.7. Modelltraining (Data Scientist, Entwicklung)

2.3.8. Modelltests und -optimierung (Data Scientist)

2.3.9. Modellanwendung (Entwicklung)

2.3.10. Ergebnisausgabe (Entwicklung)

2.3.11. Selbstevaluierung

2.3.11.1. temporäres hinzufügen von Prognosedaten Wetter (Entwicklung)

2.3.11.2. temporäres hinzufügen von Prognosedaten Veranstaltungen (Entwicklung)

2.3.11.3. Starten des Vorhersagemodells für den aktuellen Monat (Entwicklung)

2.3.11.4. Vergleich der Ausgabedaten mit aktueller Vorhersage (Entwicklung)

2.3.11.5. Notifikation bei wesentlichen Vorhersageänderungen an Planungsstelle (Entwicklung)

2.4. Anwendungsentwicklung

2.4.1. Mockups bzw. Prototypen designen

2.4.1.1. manuelle Dateneingabe interner Daten (GUI) (Entwicklung)

2.4.1.2. manuelle Dateneingabe Veranstaltungsdaten (GUI) (Entwicklung)

2.4.2. Systemelemente erstellen

2.4.2.1. Start eines Modelltrainings (Entwicklung)

2.4.2.2. Start einer Vorhersage durch das Modell (Entwicklung)

2.4.2.3. täglicher Start des Selbstevaluierungsalgorithmus (Entwicklung)

2.4.2.4. Ergebnisausgabe (Entwicklung)

2.4.3. Genehmigte Prototypen umsetzen und entwickeln (Entwicklung)

2.4.4. Schnittstelle für Dateneingaben über GUI entwickeln (Entwicklung)

2.4.5. Schnittstelle für Format der Datenausgabe entwickeln (Entwicklung)

2.4.6. Sicherheitstest für Datenschnittstellen entwickeln (Entwicklung)

2.4.7. Qualitätssicherung

2.4.7.1. Testfälle generieren (*Productowner*)

2.4.7.2. Integrationstests durchführen (*Entwicklung/Tester:in*)

2.4.7.3. Regressionstests nach Sprints durchführen (*Entwicklung/Tester:in*)

2.4.7.4. Testreports dokumentieren (*Entwicklung/Tester:in*)

2.4.7.5. Bug-Issues erstellen (*Entwicklung/Tester:in*)

2.4.8. Manual/Nutzungsdokumentation erstellen

2.4.9. Ausbildungsunterlagen erstellen

3. Logistik

3.1. Hardware Anforderungen spezifizieren (*Entwicklung, IT-Betrieb, Productowner, Data Scientist*)

3.2. Beschaffung der Hardwarekomponenten (*IT-Betrieb, Einkauf*)

3.3. Ersatzteilibewirtschaftung (Lagerung, Nachbestellung, ...) (*IT-Betrieb*)

3.4. Hardwareablöse planen (*IT-Betrieb*)

4. Betrieb

4.1. Aufbau der Hardware (*IT-Betrieb*)

4.2. Integration der Hardware (IP-Adressen, Anschlüsse, Stromversorgung, Kühlung, ...) (*IT-Betrieb*)

4.3. Release roll-outs planen und durchführen (*IT-Betrieb, Entwicklung*)

4.4. Monitoring umsetzen (*IT-Betrieb, Entwicklung*)

4.5. Benutzerschulung organisieren (*Planungsstelle*)

4.6. Vorgaben zum IT-Betrieb festlegen und dokumentieren (*IT-Betrieb*)

4.7. Wartung organisieren (*IT-Betrieb*)

5. Lieferung und Abnahme

5.1. Abnahmeerklärung vorbereiten (*Projektleitung*)

5.2. Lieferumfang prüfen (*Projektleitung*)

6. Adaptierung von Arbeitsprozessen

6.1. Workshops organisieren (*Planungsstelle*)

6.2. Workshops durchführen (*Planungsstelle, Leiter Bereitschaft, IT-Betrieb*)

6.3. Prozesshandbuch erstellen (*Planungsstelle*)

7. Kommunikation

7.1. Kommunikationsplan erstellen (*Projektleitung, PR-Abteilung*)

7.2. externe Kommunikation durchführen (Pressemitteilung über Erstellung und Erfolg) (*PR-*

Abteilung)

- 7.3. interne Kommunikation durchführen (Ankündigung, Fortschrittsinformationen, Schulungspläne, ...) (*Projektleitung, PR-Abteilung, Planungsstelle*)
- 7.4. Feedback-System entwickeln (*Entwicklung*)

2.2 Team

Wie sich anhand der Aufgabenliste gezeigt hat sollte das Projektteam, neben der Projektleitung, aus Mitarbeitenden der Bereiche

- PR
- IT-Betrieb
- Entwicklung
- Bereitschaftsplanung (Nutzer)
- Data Science (sofern diese nicht Teil der Entwicklung sind)

bestehen.

Aufgrund der geringen Größe des Projektes sind manche eigenständigen Aufgaben, wie zB das Änderungsmanagement, zu Teilaufgaben des Projektmanagements geworden. Die Projektleitung hat neben diesen gebündelten administrativen Aufgaben jedoch auch die Funktion des Productowners zu erfüllen und die Kommunikation zu internen und externen Stakeholdern zu führen. In Summe ergibt sich dadurch eine sehr hohe Arbeitslast, weshalb die Projektleitung mit drei Personen abgebildet wird, nämlich dem Projektleiter, der administrativen Assistenz der Projektleitung sowie einer stellvertretenden Projektleitung mit QS-Verantwortung. Das Entwicklungsteam ist ein Subelement des Projektteams und setzt sich zusammen aus einem/r SCRUM-Master, zwei Entwickler:innen und einem/r Data Scientist mit Entwicklungskenntnissen. Für manche Aufgaben besteht zusätzlicher Informationsbedarf aus zB dem Einkauf oder der Personalabteilung, Dieser erfordert allerdings nicht eine Beteiligung der jeweiligen Abteilung am Projektteam.

Geplant ist demnach ein Projektteam, bestehend aus neun Personen, das sich wie angeführt zusammensetzt:

- Projektleitung
- stellvertretende Projektleitung (aus PR-Abteilung, übernimmt QS-Verantwortung)
- administrative Assistenz der Projektleitung
- Entwicklungsteam (vier Personen, wie oben beschrieben)
- Requirements + QS Mitarbeit (Schnittstelle Nutzer; aus Abteilung Planungsstelle)
- Requirements + QS Mitarbeit (Schnittstelle IT-Betrieb, aus Abteilung IT-Betrieb)

3 Projektplan

Wie bereits im Konzeptionsteil dieser Arbeit beschrieben wurde, lehnt sich die Planung dieses Projektes an das Softwareprozess-Rahmenmodell V-Modell XT an (Angermeier et al. 2024). Die untere Grafik zeigt daher die Entscheidungspunkte des Projektes in einer Projektdurchführungsstrategie, die ebenso für die Definition der Meilensteine herangezogen wird. Die abgebildeten Zahlen nummerieren diese und werden in dieser Arbeit gleichzeitig als Referenz auf den jeweiligen Meilenstein weitergeführt. Mit Hilfe der nachfolgenden Zeitplanung werden die Entwicklungsschritte zeitlich hinterlegt und ermöglichen so, möglicherweise noch weitere Optimierungen hinsichtlich Ablauf und Teamkonfiguration.

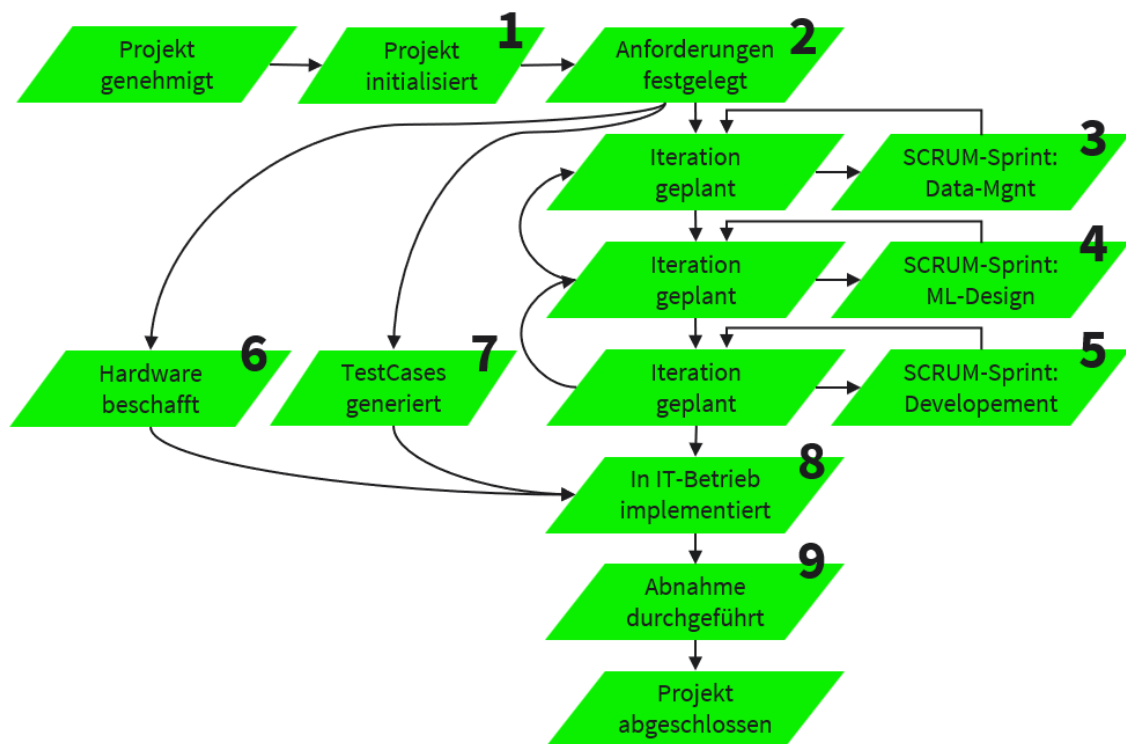


Abb. 2: Projektdurchführungsstrategie

3.1 Zeitplanung

3.1.1 Definition von Meilensteinen

Wie bereits angeführt sind die o.a. nummerierten Meilensteine aus der groben Durchführungsstrategie des Projektes abgeleitet. Im Nachfolgenden werden wesentliche Arbeitspakete des Projektstrukturplans, die den jeweiligen Meilensteinen zugeordnet sind, mit Zeitschätzungen hinterlegt. Diese sind nach dem Schema (*Mindestbedarf/Erwartungswert/Maximalbedarf*) in Tagen notiert. Für die Aufgaben der Entwicklungssprints wurden die Arbeitspakete des Punktes Systemerstellung der Konzeptionsphase

zusammengefasst. Ein Sprint wird hierbei mit jeweils zehn Arbeitstagen (zwei Wochen) veranschlagt.

1. **Projekt initialisiert** (6/9/12)

- 1.1. Projektteam zusammenstellen (2/3/5)
- 1.2. Zeitplanung erstellen (2/2/4)
- 1.3. Kostenschätzung erstellen (1/2/2)
- 1.4. Qualitätshandbuch erstellen (4/7/9)
- 1.5. Risikoliste erstellen (2/3/4 Tage)

Insgesamt werden für das Erreichen dieses Meilensteines mindestens elf Personaltage, aber maximal 24 Personaltage veranschlagt. Der erwartete Wert liegt bei 17 Personaltagen. Abgesehen von der Zuarbeit durch Fachbereiche liegen diese Aktivitäten mit Masse im Bereich der Projektleitung. Diese besteht aus zwei Personen und einer administrativen Assistenz. Die Tätigkeiten sollten daher in ca. sechs bis zwölf Tagen zu bewerkstelligen sein. Die erwartete Dauer liegt bei etwa neun Tagen.

2. **Anforderungen festgelegt** (18/25/31)

- 2.1. Workshop „neue Arbeitsprozesse“ organisieren (3/4/8)
- 2.2. Workshop „neue Arbeitsprozesse“ durchführen (2/2/2)
- 2.3. Requirements Engineering und Backlog erstellen (10/14/16)
- 2.4. Hardware spezifizieren (3/4/5)

Die Anforderungen eines Projektes sollen detailliert und ausreichend tief spezifiziert sein. Je genauer hier gearbeitet wird und umso mehr Stakeholder an dieser Stelle mitwirken können, desto zufriedenstellender und zielgerichteter wird das Ergebnis aussehen. An diesem Meilenstein, wird daher das gesamte Projektteam beteiligt sein, und gegebenenfalls auch noch weitere externe Stakeholder hinzuziehen. Eine „Arbeitsteilung“ scheint hierbei jedoch wenig sinnvoll, weshalb ein Zeitbedarf von 18/24/31 Tagen für diese Aufgaben geschätzt wird. Wie im späteren Auszug aus dem GANTT-Diagramm (siehe Abb.3) ersichtlich sind mehrere weitere Aufgaben von den Ergebnissen des Requirements Engineering abhängig. Optimalerweise wird dieses bereits während des Workshops mit dem Team der Planungsstelle begonnen, wo die Anforderungen der zukünftigen Nutzer diskutiert, aufgenommen und dokumentiert werden können. Durch diese Überlappung reduziert sich die erwartete Gesamtdauer bis zur Erreichung dieses Entscheidungspunktes auf 22 Tage.

3. **Scrum Sprint: Data-Management** (1/2/2 Sprints)

- 3.1. Sprintorganisation (Änderungsanträge, Sprintbacklog, ...) (je Sprint 1/1/1)
- 3.2. Externe Datenquellen: Recherche und Spezifikation (2/3/5)
- 3.3. Datenschema „DRK Daten“ erstellen (1/2/3)
- 3.4. Datenschema „Wetterdaten“ erstellen (1/2/3)
- 3.5. Datenschema „Wochenend- und Feiertagsdaten“ erstellen (1/2/3)
- 3.6. Datenschema „besondere Wetterbedingungen“ erstellen (1/2/3)

3.7. Datenschema „Veranstaltungsdaten“ erstellen (1/2/3)

3.8. Funktion: Externe Daten abrufen (2/4/5)

3.9. Funktion: Daten vorverarbeiten und speichern (4/6/7)

3.10. Datenbank designen (5/7/10)

Die Data-Management Sprints bilden in vielerlei Hinsicht die Grundlage für die weitere Entwicklung. Deshalb arbeitet das gesamte Scrum-Team an diesen Aufgaben. Mit drei Personen liegt der Zeitbedarf für diese Aufgaben bei 6/10/14 Tagen, also ein oder zwei Sprints. Ein durchdachtes Datenmanagement ist für die nachfolgenden Schritte förderlich, daher werden zwei Sprints für diese Arbeitspakete als Zeitbedarf angenommen. Wobei sich der erste Sprint vorwiegend mit der Datenbeschaffung und der zweite mit dem Datenbankdesign beschäftigt.

4. **Scrum Sprint: Machine Learning Design** (4/5/6 Sprints)

4.1. Sprintorganisation (Änderungsanträge, Sprintbacklog, ...) (je Sprint 1/1/1)

4.2. Entwicklungsdatensatz „interne Daten“ generieren (1/1/1)

4.3. Entwicklungsdatensatz „externe Daten“ generieren (1/1/1)

4.4. Datenexploration, -dokumentation und -aggregation (2/4/5)

4.5. Konzeption verschiedener ML-Ansätze bzw. Algorithmen (5/7/10)

4.6. Evaluierung der ML-Ansätze (5/8/10)

4.7. Modelltrainings (10/12/15)

4.8. Modelltests und -optimierung (5/6/9)

4.9. Modellanwendung (Prediction) parametrisieren („flexible prediction scope“) (3/4/5)

Für das Machine Learning Design ist in erster Linie die/der Data Scientist verantwortlich. Hierfür ist eine Person im Scrum-Team vorgesehen. Die ML-Sprints lassen sich daher teilweise parallel zu den nachfolgenden Development-Sprints umsetzen. Der summierte Zeitbedarf würde mit 31/43/56 Tagen, also vier bis sechs Sprints geschätzt werden. Einige der Aufgaben haben jedoch Überschneidungspunkte und laufen in ähnlichen Denkprozessen ab, weshalb auch ein/e einzelne/r Data Scientist teilweise Arbeitspakete parallel bearbeiten kann. Modelltrainings, -tests und -optimierungen sind ein gutes Beispiel hierfür. Im GANTT-Diagramm (siehe Abb.3) konnten die Arbeitspakete sinnvoll auf nur drei Sprints aufgeteilt werden, wobei der Fokus des Ersten auf der Konzeptionierung, des Zweiten auf der Evaluierung verschiedener Ansätze und des dritten Sprints auf den Trainings- und Optimierungen des Modells gelegt wird.

5. **Scrum Sprint: Developement** (1/2/3 Sprints)

5.1. Sprintorganisation (Änderungsanträge, Sprintbacklog, ...) (je Sprint 1/1/1)

5.2. Eingabe-GUI (interne Daten) designen (5/10/12)

5.3. Eingabe-GUI (externe Daten) designen (5/10/12)

5.4. Funktion: Modelltraining starten (1/1/1)

5.5. Funktion: Modellanwendung (Prediction) starten (1/1/1)

- 5.6. Funktion: Ergebnisse ausgeben (1/1/1)
- 5.7. Funktion: Ergebnisausgabe mit bestehenden, älteren Vorhersagen vergleichen (1/2/2)
- 5.8. Funktion: Notification (3/5/6)
- 5.9. Funktion: Sicherheitschecks für Datenschnittstellen (1/1/1)
- 5.10. Funktion: Userfeedback aufnehmen (3/5/6)

Die Arbeitspakete dieses Sprints liegen mit Masse im Aufgabenbereich der Entwickler:innen. Davon wurden in der Teamkonstellation zwei vorgesehen. Der summierte und anschließend halbierte Zeitbedarf beträgt daher 8/18/21 Tage. Das entspricht ein bis drei Sprints. Nachdem Entwicklung zumeist länger dauert als angenommen, werden hier für die Planung drei Sprints vorgesehen. Jede angeführte, entwickelte Funktion beinhaltet des weiteren die dazugehörigen Unit-Tests, die Eintragungen in die Systemdokumentation und die Bedienungsanleitung, sowie die entsprechenden Log-Ausgaben für ein Systemmonitoring. Die Sprints der Anwendungsentwicklung können bereits parallel zu jenen des Machine-Learning Designs begonnen werden.

6. **Hardware beschafft** (47/70/138)

- 6.1. Angebote einholen (2/5/10)
- 6.2. Beschaffung einleiten (2/2/2)
- 6.3. Lieferzeit berücksichtigen (40/60/120)
- 6.4. Lieferung abnehmen (1/1/1)
- 6.5. Hardware logistisch erfassen (1/1/2)
- 6.6. Ersatzteilibewirtschaftung und Hardwareablöse planen (1/1/3)

Das Beschaffen der Hardware wird der/die Mitarbeiter:in des IT-Betriebes übernehmen. Die zu erwarteten Lieferzeiten sind hierbei ein kritischer und schwer zu kalkulierender Faktor. Deshalb sollte das Einleiten der Beschaffung so bald als möglich stattfinden. Der Zeitbedarf wird an dieser Stelle mit 47/70/138 Tagen geschätzt.

7. **TestCases generiert** (5/7/10)

- 7.1. Testfälle generieren (5/7/10)

Test Cases zur Überprüfung der fertigen Software basieren auf den User Stories, also auf Anwendungsfällen, mit denen die Nutzer konfrontiert sind und die es zu bewerkstelligen gilt. (microTool GmbH 2017b, S.12). Der diesem Projekt zugrunde liegende Use Case (Grunsky 2024) ist vergleichsweise mit anderer Software relativ einfach in der Benutzer-System-Interaktion. Dennoch liegt es in der Verantwortung des Productowners (in diesem Fall, der Projektleitung), in der Phase der Anforderungserstellung die User Stories herauszuarbeiten und die entsprechenden Test Cases für die Integration zu generieren. Optimalerweise beginnt dieser Prozess bereits während, oder unmittelbar nach den Workshops mit den Mitarbeitern der Planungsstelle (siehe Meilenstein: „Anforderungen festgelegt“). Für das erstellen von Testfällen wird ein Bedarf von 5/7/10 Tagen angenommen.

8. **In IT-Betrieb implementiert** (7/14/22 einmalig, 3/4/7 iterativ)

- 8.1. Release roll-out plan erstellen (1/2/3 einmalig)
- 8.2. Aufbau der Hardware (1/2/4 einmalig)
- 8.3. Software auf neuer Hardware bereitstellen (2/2/4)
- 8.4. Monitoring Grundgerüst/Dashboard umsetzen (5/10/15 einmalig)
- 8.5. Integrationstests durchführen (2/2/3 iterativ)
- 8.6. Monitoring umsetzen (1/2/4 iterativ)

Die Aufgaben der Integration übernimmt der IT-Betrieb, der mit einer Person im Projektteam vertreten ist. Die angeführten Arbeitspakete lassen sich in zwei Kategorien unterteilen. Jene, die nur einmalig ausgeführt werden müssen, wie zB der Aufbau der Hardware, und die anderen, die iterativ nach jedem „ausrollbaren“ Softwarestand zu erledigen wären. Die einmaligen Aufgaben, können teils bereits zu einem frühen Zeitpunkt, unmittelbar nach dem Requirements Engineering begonnen werden (wie zB das Erstellen eines Release Roll-out Plans oder des Monitoring Grundgerüsts), das spart Zeit und schafft Reserven. Andere finden sich nahezu an letzter Position des Zeitplans, wie zB der Aufbau der gelieferten Hardware. Wie im GANTT-Diagramm (siehe Abb.3) ersichtlich, bestimmen vor allem die Hardwarelieferzeiten die Gesamtdauer des Projektes. Ein frühzeitiges Release von funktionierenden Teilkomponenten wird daher erwartungsgemäß obsolet, da mit Aufbau der Hardware die Software bereits fertiggestellt sein sollte. Dennoch finden sich ein Release Roll-out Plan, sowie die iterativen Aufgaben in der Planung wieder, um gegebenenfalls (zB bei erhöhtem Zeitbedarf in der Entwicklung) darauf zurückgreifen zu können. Erwartungsweise wird jedoch nur ein Release mit anschließenden Integrationstests und Implementierung des Monitorings notwendig sein.

9. **Abnahme durchgeführt** (9/12/17)

- 9.1. Schulungsunterlagen erstellen (3/4/6)
- 9.2. Benutzerschulung durchführen (2/2/2)
- 9.3. Projektdokumentation abgeschlossen (4/5/8)
- 9.4. Abnahmeerklärung gefertigt (1/1/1)

Sind alle vorangegangenen Aufgaben ordnungsgemäß erledigt worden, werden für die Abnahme nur noch die Projektleitung und die/der Mitarbeitende der Planungsstelle benötigt. Hierbei können die großen Punkte „Schulung“ und „Dokumentation“ parallel bearbeitet werden. Der geschätzte Zeitaufwand beträgt daher 5/7/9 Tage.

3.1.2 Erstellung eines GANTT-Diagramms

Das in Abb.3 dargestellte GANTT-Diagramm zeigt die zeitlich hinterlegten Meilensteine und die Arbeitspakete in ihren Abhängigkeiten zueinander, ihrer Bearbeitungsreihenfolge, und den Möglichkeiten der parallelen Bearbeitung. Für den Zeitbedarf der einzelnen Arbeitspakete wurde jeweils der oben beschriebene und begründete Erwartungswert herangezogen. Der so berechnete Gesamtzeitbedarf für dieses Projekt wird mit 23 Wochen angenommen. Im Diagramm begann die Bearbeitung am 03. Februar 2025 und sollte gemäß Planung bis zum 14. Juli 2025 abgeschlossen werden können.

Zur exemplarischen Darstellung der Vorgehensweise wurden in der u.a. Grafik die Meilensteine „Anforderungen festgelegt“ und „In IT-Betrieb implementiert“ mitsamt ihren Arbeitspaketen angezeigt. Man erkennt hier gut die Bedeutung des Requirements Engineering aufgrund der nachvollziehbaren Abhängigkeiten weiterer Meilensteine und Arbeitspakete. Ebenso deutlich zu sehen ist die der Grund für den augenscheinlich großen Zeitbedarf der Implementierung in den IT-Betrieb. Schuld daran ist die „Aufspreizung“ der Arbeitspakete auf nahezu den gesamten Entwicklungsprozess. Zu guter Letzt ist auch die Parallelität der Entwicklungs- und Machine Learning Sprints erkennbar, sowie der Gewinn hinsichtlich des Gesamtzeitbedarfs, der sich daraus ergibt. Das ermöglicht auch, dass in der ersten Planung nur wenig auf die, in der Konzeptionsphase getroffenen, Priorisierung der Arbeitspakete eingegangen werden muss. Die vorhandene Zeit lässt auch die planmäßige Umsetzung von niedriger priorisierten Funktionen der Software zu.

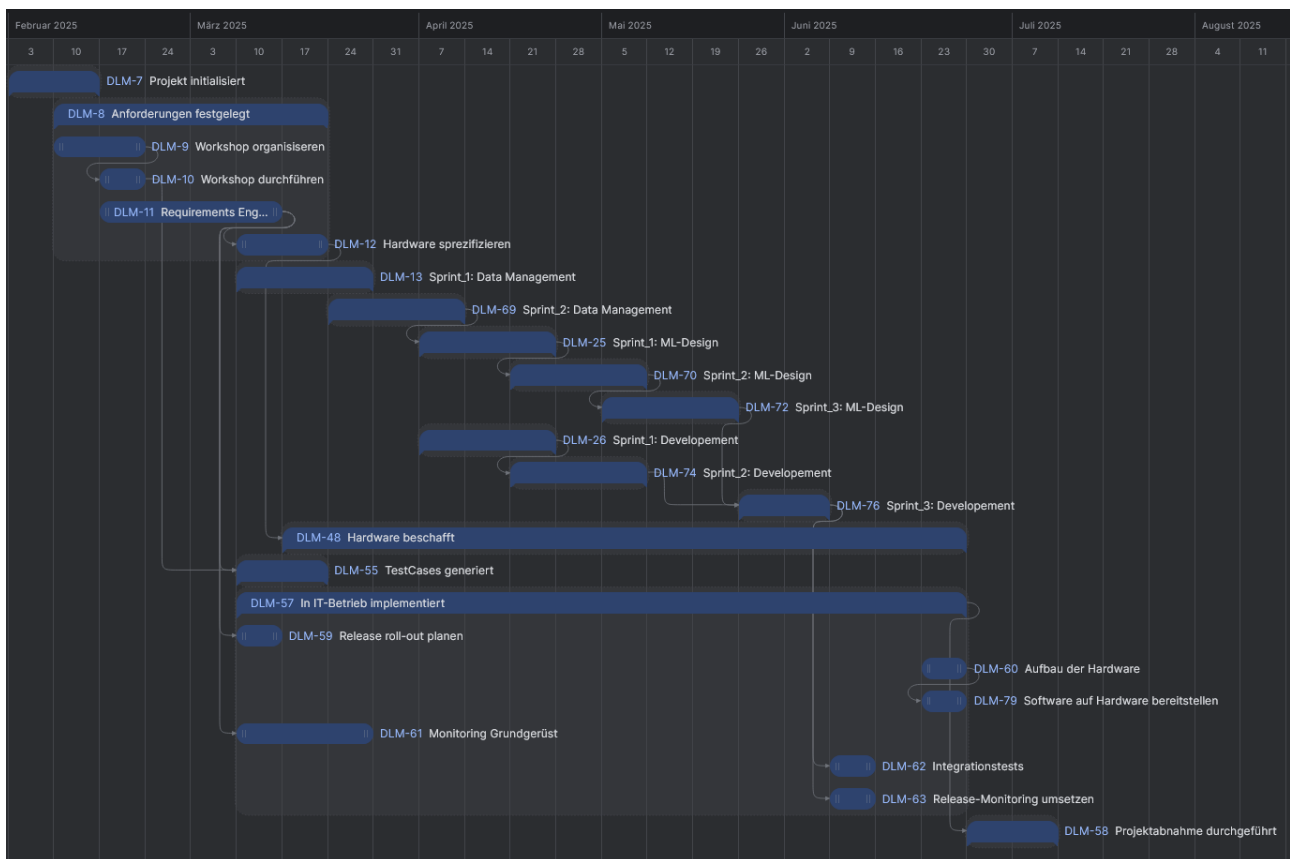


Abb. 3: GANTT-Diagramm

Die nachfolgende Grafik zeigt die Anpassung und Erweiterung der Projektdurchführungsstrategie aus Abb.2. Die hinzugefügten schwarzen Kästen in der rechten unteren Ecke der Entscheidungspunkte zeigen den geschätzten Zeitaufwand (in Wochen) für die Umsetzung dieses Meilensteins. Die kumulierte Dauer der Pfade bis zu Beginn der Tätigkeiten für den Entscheidungspunkt „In IT-Betrieb implementieren“ wurden farbig herausgehoben. Der rot eingefärbte Kasten und die dazugehörigen Pfeile stellen den kritischen Pfad dar, der - gut erkennbar - maßgeblich von der Beschaffung der Hardware abhängig ist. Der Pfad der Entwicklung folgt dem kritischen Pfad jedoch dicht, mit nur einer Woche Reserve. Eine Verzögerung in diesem Bereich ließe das Projekt daher genauso schnell in der Gesamtdauer anwachsen wie ein Lieferverzug der Hardware.

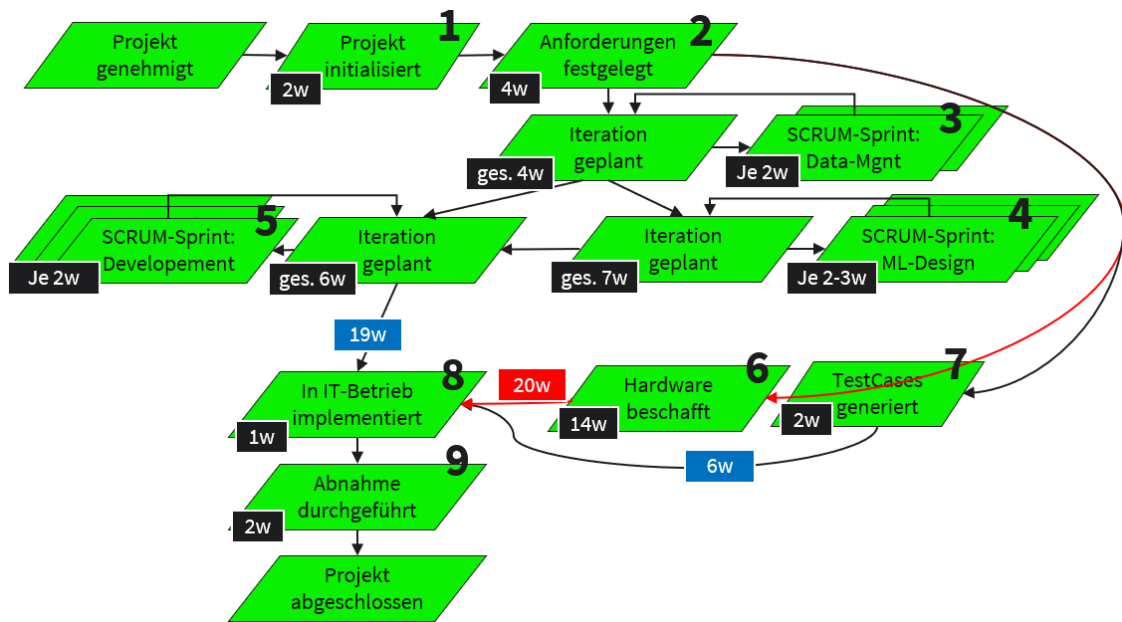


Abb. 4: Projektdurchführungsstrategie inkl. Zeitschätzung

3.1.3 Änderung der Teamkonfiguration

Durch das Hinzufügen eines/einer weiteren Data Scientist, könnte der Zeitbedarf der Machine-Learning Sprints vermutlich reduziert werden. Das hätte Auswirkungen auf den gesamten Entwicklungsprozess. Hier könnten gem. GANTT-Diagramm etwa zwei Wochen eingespart werden. Das SCRUM-Team wäre dann mit seiner Arbeit fertig und würde nicht mehr gebraucht. Im Gegenzug dazu müsste für 15 Wochen ein/e weitere/r Data Scientist bezahlt werden. Ob durch diese Maßnahme die Projektkosten reduziert werden können bleibt daher noch zu erwägen.

Aufgrund des kritischen Pfades der Hardwarebeschaffung wäre eine Verkürzung des Entwicklungsprozesses und damit auch eine Änderung der Teamkonfiguration aus zeitlichen Gründen nicht zwingend notwendig.

3.2 Definition von „fertig“

Der Use Case zu diesem Projekt (Grunsky 2024) definiert die benötigte Kernfunktionalität (und damit auch den ersten „fertigen“ Zustand) der Software damit, dass diese bis zum zehnten Tag jedes Monats eine Liste mit der tageweisen Anzahl benötigter Bereitschaftsfahrer:innen für den Folgemonat ausgeben soll. Hierfür ist die Eingabe interner Daten der Bereitschaftsfahrten (als Lernmaterial), das Machine Learning Modell (Berechnung) und die Ausgabe für die Planungsstelle (Ergebnisliste) erforderlich. Vereinfacht ausgedrückt wird dies mit zwei GUIs, einer Datenbank, und einem passenden Zeitreihen-Vorhersagemodell abgedeckt. Die damit verbundenen Funktionen und Arbeitspakete wurden in der Konzeptionsphase als **must** priorisiert. Die restlichen Aufgaben, sowie die Beschaffung der Hardware bleiben dadurch jedoch unberührt.

Das Hinzuziehen weiterer externer Daten wie Veranstaltungs-, Temperatur- oder Wetterdaten, sowie die laufende Selbstevaluierung der eigenen Vorhersage, ist für die Grundfunktionalität nicht zwingend erforderlich. Im Falle eines höheren Zeitbedarfs als geplant richtet sich die Priorität der Arbeitspakete

nach der Grundfunktionalität, diese ist mit dem ersten Release und bei vorhandener Hardware nach einer Gesamtzeit von etwa 23 Wochen zu erreichen. Zusatzfunktionalitäten können dann bei ausreichend vorhandenem Budget ggf. über einen Release Roll-out Plan nachgezogen werden.

3.3 Kostenschätzung

Die geschätzten Kosten für dieses Projekt setzen sich im Grunde aus drei Bereichen zusammen:

1. Personalkosten für Projektabwicklung und Softwareentwicklung
2. Kosten für die geplante Hardware
3. Administrative Kosten inkl. Arbeitsmaterial, Telefongebühren, etc.

Für den letzten Punkt wurde der Einfachheit halber, bei einem Projektteam von insgesamt neun Teammitgliedern und einer geschätzten Gesamtprojektdauer von etwa einem halben Jahr, ein Betrag von € 10.000,- angenommen.

Die Kosten für die Hardware können im Detail erst nach dem Arbeitspaket „Hardware spezifizieren“ festgelegt werden. Schätzungsweise wird eine performante aber möglichst einfache Server- und Storagelösung für diesen Use Case bevorzugt. Überschlagsmäßig ist hierfür mit Kosten in der Höhe von etwa € 60.000,- zu rechnen (Thomas-Krenn.AG 2025).

Für die Schätzung der Personalkosten wurden durchschnittliche Jahresgehälter, sowie Arbeitgebernebenkosten (ca. 29%) in Deutschland herangezogen und unter der Annahme von 220 Arbeitstagen im Jahr auf einen Tagessatz für die einzelnen Projektteammitarbeiter:innen heruntergebrochen (ChatGPT 2025).

Der Zeitbedarf anhand der Zeitplanung sieht vor, dass alle Teammitglieder für 17 Wochen durchgehend an dem Projekt arbeiten werden, der/die Mitarbeiter:in aus dem IT-Betrieb noch eine weitere zusätzliche Woche, sowie die Mitglieder der Projektleitung noch weitere 6 Wochen. Die unten angeführte Tabelle zeigt die angeführten Berechnungen. Eine Woche besteht aus fünf Arbeitstagen.

Projektteammitglied	durchschnittlicher Jahresbruttogehalt	Arbeitgeber- nebenkosten	summierte Jahreskosten	Tagessatz	Zeitbedarf in Wochen	Kosten
Projektleitung	€ 75 000,00	€ 21 750,00	€ 96 750,00	€ 439,77	23	€ 50 573,86
stellvertretende Projektleitung	€ 70 000,00	€ 20 300,00	€ 90 300,00	€ 410,45	23	€ 47 202,27
Administrative Assistenz	€ 40 000,00	€ 11 600,00	€ 51 600,00	€ 234,55	23	€ 26 972,73
SCRUM-Master	€ 70 000,00	€ 20 300,00	€ 90 300,00	€ 410,45	17	€ 34 888,64
Entwickler:in 1	€ 60 000,00	€ 17 400,00	€ 77 400,00	€ 351,82	17	€ 29 904,55
Entwickler:in 2	€ 60 000,00	€ 17 400,00	€ 77 400,00	€ 351,82	17	€ 29 904,55
Data Scientist	€ 65 000,00	€ 18 850,00	€ 83 850,00	€ 381,14	17	€ 32 396,59
Mitarbeiter:in Planungsstelle	€ 55 000,00	€ 15 950,00	€ 70 950,00	€ 322,50	17	€ 27 412,50
Mitarbeiter:in IT-Betrieb	€ 55 000,00	€ 15 950,00	€ 70 950,00	€ 322,50	18	€ 29 025,00
Gesamtpersonalkosten						€ 308 280,68

Abb. 5: Personalkostenschätzung

Die unter Punkt 3.1.3 angedachte Änderung der Teamkonfiguration lässt sich an dieser Stelle kostenmäßig gut vergleichen. Der Zeitbedarf für die Entwicklungsteams und den/die Mitarbeiter:in der Planungsstelle reduziert sich auf 15 Wochen dafür wird für ein/e weitere/r Data Scientist für 15 Wochen hinzugerechnet.

Projektteammitglied	durchschnittlicher Jahresbruttogehalt	Arbeitgeber- nebenkosten	summierte Jahreskosten	Tagessatz	Zeitbedarf in Wochen	Kosten
Projektleitung	€ 75 000,00	€ 21 750,00	€ 96 750,00	€ 439,77	23	€ 50 573,86
stellvertretende Projektleitung	€ 70 000,00	€ 20 300,00	€ 90 300,00	€ 410,45	23	€ 47 202,27
Administrative Assistenz	€ 40 000,00	€ 11 600,00	€ 51 600,00	€ 234,55	23	€ 26 972,73
SCRUM-Master	€ 70 000,00	€ 20 300,00	€ 90 300,00	€ 410,45	15	€ 30 784,09
Entwickler:in 1	€ 60 000,00	€ 17 400,00	€ 77 400,00	€ 351,82	15	€ 26 386,36
Entwickler:in 2	€ 60 000,00	€ 17 400,00	€ 77 400,00	€ 351,82	15	€ 26 386,36
Data Scientist 1	€ 65 000,00	€ 18 850,00	€ 83 850,00	€ 381,14	15	€ 28 585,23
Data Scientist 2	€ 65 000,00	€ 18 850,00	€ 83 850,00	€ 381,14	15	€ 28 585,23
Mitarbeiter:in Planungsstelle	€ 55 000,00	€ 15 950,00	€ 70 950,00	€ 322,50	15	€ 24 187,50
Mitarbeiter:in IT-Betrieb	€ 55 000,00	€ 15 950,00	€ 70 950,00	€ 322,50	18	€ 29 025,00
Gesamtpersonalkosten						€ 318 688,64

Abb. 6: Personalkostenschätzung mit zusätzlicher/m Data Scientist

Die Personalkosten steigen in dieser Variante um € 10.000,- und, wie beschrieben, wird auch kein zeitlicher Vorteil durch diese Personalverstärkung erreicht. Die Teamkonfiguration wird demnach wie geplant weitergeführt.

Die Gesamtkosten des Projekts belaufen sich, aufgrund der Schätzungen auf etwa € 380.000,-, wie die u.a. Tabelle zusammenfassend darstellt. Im Pitch Deck des Use Cases wurde prognostiziert, dass mit einer Vorhersage des Bereitschaftspersonalbedarfs anhand der Software bis zu € 14.000,- pro Tag eingespart werden können. Mit dieser Prognose würden sich die Projektkosten vmtl. innerhalb von ein bis zwei Monaten amortisieren.

Kostenart	Kosten
Personalkosten (aufgerundet)	€ 310 000,00
Hardwarekosten	€ 60 000,00
Administrative Kosten, etc	€ 10 000,00
Gesamtpersonalkosten	€ 380 000,00

Abb. 7: Gesamtkosten

4 Stakeholder Analyse

In der vorausgehenden Use Case Analyse (Grunsky 2024) wurden drei unmittelbare Stakeholder des Machine Learning Projektes angeführt.

1. **Die Bereitschaftsfahrenden:** Es ist davon auszugehen, dass das neue, bedarfszentrierte System üblicherweise weniger Personal in StandBy bereithalten wird, als das vorherige, welches fix 90 Bereitschaftsfahrende pro Tag vorsah. Demnach wird das Bereitschaftspersonal über einen längeren Zeitraum deutlich weniger Bereitschaften zu besetzen haben. Dafür können, diese erst Mitte des Vormonats, also vermutlich deutlich kurzfristiger, bekannt gegeben werden. Grundsätzlich scheint ein erfolgreich arbeitendes Vorhersagesystem den Arbeitsbedingungen der Bereitschaftsfahrer:innen zugute zu kommen. Zumindest in zeitlichem Aspekt. Finanziell wird das Bereitschaftspersonal vermutlich mit Einbußen rechnen müssen. Die Haltung dieser Stakeholder gegenüber dem neuen System ist daher schwer einzuschätzen. Langfristig werden sie allerdings damit leben müssen und haben keinen direkten Einfluss auf den Erfolg des Projektes.
2. **Der Rettungsdienst** (Vertreten durch das Präsidium): Die Leitung des Rettungsdienstes ist daran interessiert, die Leistung des Dienstes auf einem hohen Niveau zu halten und als zuverlässiger Partner für die Stadt und das Land zu gelten. Gleichzeitig ist es unerlässlich wirtschaftlich zu arbeiten und unnötige Kosten zu vermeiden. Gerade im Bereitschaftsdienst ist das eine Gratwanderung. Eine zwölfstündige Bereitschaft kostet den Rettungsdienst, gem. DRK-Reformtarifvertrag etwa € 165,- (Deutsches Rotes Kreuz 2023). Bei 90 eingeteilten Bereitschaften sind das € 14.850,- pro Tag. Sind die ersten Ansätze vielversprechend, ist davon auszugehen, dass das Präsidium der Umsetzung dieses Projektes positiv und geduldig gegenübersteht, weil langfristig hohe Kosteneinsparungen erzielt werden können. Zeigen die Prototypen des Modells allerdings Schwächen bzw. eine zu hohe Chance zu wenig Bereitschaftsfahrende für einen Tag vorherzusagen, könnte das Projekt vermutlich bereits zu einem frühen Zeitpunkt gestoppt werden um das Budget nicht zu belasten. Wahrscheinlich fordert das Präsidium daher frühzeitig erfolversprechende Ergebnisse eines prototypischen Vorhersagemodells.
3. **Die Planungsstelle:** Die Mitarbeitenden der Planungsstelle sind die Nutzer des Systems und haben hierdurch am meisten zu verlieren. Derzeit werden, per Anordnung, 90 Bereitschaftsfahrende pro Tag fix vorgesehen. Ein/e Mitarbeiter:in der Planungsstelle ist nicht verantwortlich dafür, ob diese Anzahl ausreicht oder nicht. Es wurde so angeordnet. Mit dem neuen System ist das anders. Ein Vorhersagemodell ermittelt den wahrscheinlichen Bedarf an Bereitschaftspersonal pro Tag und die Mitarbeiter:innen der Planungsstelle übernehmen diesen Wert ihre Planung. Sie sind jedoch auch befugt diesen (protokolliert) zu verändern, wenn sie der Meinung sind, das mehr oder weniger Bereitschaftsfahrende notwendig sind. Mit der Übernahme bestätigen sie, Kraft ihrer Erfahrung, somit auch die Plausibilität der Vorhersage und übernehmen die Verantwortung. Nachdem eine zu geringe Planung, gemäß Vorgabe, in jedem Fall verhindert

werden muss, wird das Präsidium selbst mit größtem Interesse auf solche „Fehler“ achten. Es ist ratsam, die Verantwortlichkeit in diesen Fällen genau und klar zu regeln um den Mitarbeitenden der Planungsstelle Sicherheit zu geben. Außerdem ist die Einbeziehung dieser Stakeholder bereits zu Beginn des Projektes in Form eines Workshops, sowie mit der Bestellung eines Projektteammitglieds zur Mitarbeit an der Qualitätssicherung im Projektplan vorgesehen. Es ist wichtig dieses Personal mit an Bord zu holen um das Projekt nicht Aufgrund einer Ablehnung durch die Nutzer zum Scheitern zu bringen. Innovation und die Möglichkeit der gemeinsamen Gestaltung sollen der Motor zu Akzeptanz und gewollter Verantwortungsübernahme sein.

microTool GmbH (2017a) führt, neben Nutzern, Besitzern, und abhängigen Unternehmen, in einem Leitfaden noch die **Kunden** einer Organisation als Stakeholder an. Im Fall des Roten Kreuzes sind dies die Betroffenen der gemeldeten Notfälle, die in einer solchen Situation ausschließlich an schneller Hilfe interessiert sind. Ob ein Bereitschaftsfahrender von einem Machine Learning System auf StandBy gesetzt wurde oder nicht, ist hier vollkommen egal. Hauptsache es ist jemand verfügbar. Unzufriedene „Kunden“ sind der entscheidende Faktor, wenn es um das erste, unten angeführte, Risiko geht, den Reputationsschaden. Dieser soll auf jeden Fall vermieden werden.

5 Risikomanagement

5.1 Risikoidentifikation

Die unten angeführte Liste von zehn identifizierten Risiken des Projektes erhebt bei weitem keinen Anspruch auf Vollständigkeit, sondern dient der exemplarischen Darstellung.

1. Reputationsschaden, wenn zu wenige Bereitschaftsfahrende eingeteilt werden
2. Ablehnung des Projektes bei den Nutzern (Planungsstelle)
3. Überschreitung des Projektbudgets
4. Projektstopp durch das Präsidium (zu geringe Erfolgsaussichten)
5. Lieferverszug im Bereich der Hardware (externe Faktoren)
6. Ungenügend bzw. mangelnde historische Testdaten zur Entwicklung des Prototyps
7. Unmut des Bereitschaftspersonals aufgrund Verdienstentgangs
8. Ausfall/Abgang von Projektteammitgliedern während des Projektes
9. Abgang von Personal das für die Wartung/Anpassung des Systems benötigt wird
10. Fehlende Vorhersage aufgrund Systemausfällen

Für eine nähere Betrachtung in den nachfolgenden Unterabschnitten, angelehnt an die Norm ISO 31000 (Wikipedia 2024), werden die Risiken Nummer 1, 2 und 5 ausgewählt und einer Risikoanalyse und -bewertung aufgrund einer zuvor definierten Risikomatrix zugeführt.

5.2 Risikomatrix

Anhand einer Risikomatrix kann das Risikoniveau eines identifizierten Risikos eingeordnet werden. Die Bewertung erfolgt nach dem Schema:

Eintrittswahrscheinlichkeit des Risikos x Auswirkung des Schadens

Vor allem bei qualitativen Risikoabstufungen unterstützt die, vorab für ein Projekt definierte, Risikomatrix bei einer nachvollziehbaren und einheitlichen Risikobeurteilung. Welche Risikoniveaus toleriert werden können, Maßnahmen erfordern oder als inakzeptabel eingestuft werden ist üblicherweise in einem Risikohandbuch deklariert. Für die Bewertung der unten angeführten Risikobeispiele gilt:

- **Risikoniveau Niedrig:** Das Risiko wird in Kauf genommen
- **Risikoniveau Mittel:** Sowohl eine Risikoverminderung als auch ein Risikotransfer sind zulässige Maßnahmen. Der kalkulierbare Risikotransfer ist allerdings vorzuziehen.
- **Risikoniveau Hoch:** Im besten Fall wirkt eine Kombination von Risikoverminderung oder -vermeidung und Risikotransfer den Risikofaktoren entgegen. In jedem Fall ist ein dokumentiertes Monitoring für dieses Risikoniveau vorgeschrieben.

- **Risikoniveau Kritisch:** Das Risiko ist für das Projekt ein „Show Stopper“ . Risikovermeidung und ein dichtes Monitoring des Risikos sind die einzig zulässige Strategie

Auswirkung / Wahrscheinlichkeit	Unwahrscheinlich	Mittel	Hoch	Sehr wahrscheinlich
Kritisch	Hoch	Kritisch	Kritisch	Kritisch
Hoch	Mittel	Hoch	Kritisch	Kritisch
Mittel	Niedrig	Mittel	Hoch	Hoch
Gering	Niedrig	Niedrig	Mittel	Mittel

Tab. 1: Risikomatrix

5.3 Risikoanalyse und -bewertung

5.3.1 Risiko 1: Reputationsschaden

Risikofaktor	Beschreibung
Risikonommer	1
Risikobeschreibung	Reputationsschaden, wegen zu wenigen Bereitschaftsfahrenden
Risikoanalyse	Ein Notruf kann nicht, oder nur verspätet, bearbeitet werden, weil eine ungenügende Anzahl an Bereitschaftsfahrer:innen für diesen Tag vorgesehen wurden. Das ML-Modell hat für diesen Tag eine zu niedrigen Bedarf vorhergesagt.
Eintrittswahrscheinlichkeit	Mittel; Je mehr Daten vorliegen und mit laufender Modelloptimierung wird die Eintrittswahrscheinlichkeit sinken
Auswirkung	Kritisch; Vor allem in der Anfangsphase des Projektes, können Schwächen in diesem Bereich zu einem Projektstopp durch das Präsidium führen
Risikoniveau	Kritisch
Risikosteuerung	Risikovermeidung
Steuerungsmaßnahmen	Vorhersagemodelle zu Beginn deutlich „überschießen“ lassen; Anfangstestdatensatz modifizieren, um mit höherem Tagesbedarf zu trainieren
Steuerungsverantwortung	Data Scientist
Risikoüberwachung	Ausführliche, dokumentierte und mit Projektleitung besprochene Modelltests in der Entwicklungsphase, periodische Modelltests und -optimierungen in ersten beiden Jahren der Betriebsphase mit historischen Daten (dokumentiert)
Überwachungsverantwortung	Data Scientist

Tab. 2: Bewertung von Risiko 1

5.3.2 Risiko 2: Mangelnde Akzeptanz bei Nutzern

Risikofaktor	Beschreibung
Risikonummer	2
Risikobeschreibung	Ablehnung des Systems durch die Nutzer
Risikoanalyse	Höhere Verantwortung für Ergebnisse eines „Black-Box-Systems“
Eintrittswahrscheinlichkeit	Hoch
Auswirkung	Kritisch; Das System wird nicht verwendet und es werden weiterhin weit überschießende Fixwerte eingetragen; Die Mitarbeiter:innen der Planungsstelle verändern sich oder machen schlechte Stimmung
Risikoniveau	Kritisch
Risikosteuerung	Verminderung
Steuerungsmaßnahmen	Erhöhung der Identifikation mit dem System bei Mitarbeitenden der Planungsstelle (Workshop); klare Regelung der Verantwortung (nicht ausschließlich bei Planungsstelle); Angeordneter Wert (wie bisher) bei unplausiblen Werten, Zweifel oder Ausfall des Systems (zB 90 Bereitschaftsfahrende, Dokumentation erforderlich)
Steuerungsverantwortung	Projektleitung
Risikoüberwachung	„Führung halten“ mit Planungsstelle, Wöchentliche Dokumentation von Bedenken und Stimmung der Mitarbeitenden (Planungsstelle) während der Entwicklungsphase
Überwachungsverantwortung	Projektleitung

Tab. 3: Bewertung von Risiko 2

5.3.3 Risiko 5: Lieferverzug der Hardware

Risikofaktor	Beschreibung
Risikonummer	5
Risikobeschreibung	Lieferverzug der Hardware
Risikoanalyse	Externe Einflussfaktoren, Projektdauer und Projektkosten erhöhen sich dadurch
Eintrittswahrscheinlichkeit	Hoch
Auswirkung	Gering; Die Umsetzung des Systems ist nicht zeitkritisch, Mit der verlängerten Projektdauer steigen jedoch die Personalkosten, die auf dieses Projekt gebucht werden.
Risikoniveau	Mittel
Risikosteuerung	Risikotransfer
Steuerungsmaßnahmen	Die erhöhten Kosten sind kalkulierbar und sollten sich in den ausgehandelten Lieferbedingungen (zB Pönale) wiederfinden. Das Risiko wird dadurch abgewälzt
Steuerungsverantwortung	Projektleitung
Risikoüberwachung	Dokumentation durch Auftragsvergabe der Hardwarebeschaffung, kein weiteres Monitoring nötig
Überwachungsverantwortung	Projektleitung

Tab. 4: Bewertung von Risiko 5

6 Literaturverzeichnis

Angermeier, D./Bartelt, C./Bauer, O./Beneken, G./ Wittmann, M. (2024): *V-Modell-XT Das deutsche Referenzmodell für Systementwicklungsprojekte Version 2.4.*

ChatGPT (2025): *Schätzung der Tageskosten für ein Projektteam in Deutschland.*

Deutsches Rotes Kreuz (2023): *DRK-Reformtarifvertrag.*

Flyvbjerg, B./ Gardner, D. (2023): *How Big Things Get Done: The Surprising Factors That Determine the Fate of Every Project, from Home Renovations to Space Exploration and Everything in Between.* Currency, New York.

Greer, M. (2011): *The PM Minimalist Quick Start Guide: The Absolute Least You Can Do!*

Grunsky, G. (2024): *Rettungsdienst Berliner Rotes Kreuz MachineLearning-Einsatz in der Bereitschaftsplanung.*

IU Internationale Hochschule (2024): *Management von IT-Projekten.*

microTool GmbH (2017a): *Stakeholder Compact.*

microTool GmbH (2017b): *The Use Case Handbook.*

microTool GmbH (2024): *objectiF RPM Benutzerhandbuch.*

Pak, M. (2024): *Aufgabenstellung_DLMDWME01.*

Tamburri, D. A.: Sustainable MLOps: Trends and Challenges. In: *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, IEEE, Timisoara, Romania (2020), 17–23.

Thomas-Krenn.AG (2025): *Rack-Server.*
<https://www.thomas-krenn.com/de/produkte/rack-server.html>.

Wikipedia (2022): *Projektstrukturplan.*

Wikipedia (2024): *ISO 31000.*

7 Abbildungsverzeichnis

1	ML-Ops Workflow	3
2	Projektdurchführungsstrategie	8
3	GANTT-Diagramm	13
4	Projektdurchführungsstrategie inkl. Zeitschlätzung	14
5	Personalkostenschätzung	15
6	Personalkostenschätzung mit zusätzlicher/m Data Scientist	16
7	Gesamtkosten	16

8 Tabellenverzeichnis

1	Risikomatrix	20
2	Bewertung von Risiko 1	21
3	Bewertung von Risiko 2	22
4	Bewertung von Risiko 5	23