
MASTERARBEIT

Herr
Martin Urragg B.A.

**Vor- und Nachteile der
Einführung von SAFe in
internationalen Großprojekten
der Medizintechnik**

Mittweida, 2021

Fakultät Ingenieurwissenschaften

MASTERARBEIT

Vor- und Nachteile der Einführung von SAFe in internationalen Großprojekten der Medizintechnik

Autor:

Herr Martin Urragg

Studiengang:

Industrial Management

Seminargruppe:

ZM 19sA1

Erstprüfer:

Dekan Prof. Dr. Andreas Schmalfuß

Zweitprüfer:

Prof. Dr. Johannes Stelling

Einreichung:

Kriens, 16. November 2021

Verteidigung/Bewertung:

Mittweida, 2021

MASTER THESIS

Advantages and disadvantages of introducing SAFe in a large international medical engineering project

author:

Mr. Martin Urragg

course of studies:

Industrial Management

seminar group:

ZM 19sA1

first examiner:

Dekan Prof. Dr. Andreas Schmalfuß

second examiner:

Prof. Dr. Johannes Stelling

submission:

Kriens, 16. November 2021

defence/ evaluation:

Mittweida, 2021

Bibliografische Beschreibung:

Urragg, Martin:

Vor- und Nachteile der Einführung von SAFe in internationalen Großprojekten der Medizintechnik. 89 S., Mittweida, Hochschule Mittweida, Fakultät Ingenieurwissenschaften, Masterarbeit, 2021

Referat:

Die vorliegende Masterarbeit befasst sich mit etablierten klassischen und agilen Projektmanagementmethoden, insbesondere mit SAFe. Anhand eines beispielhaften Projekts der Medizintechnik werden ausgewählte Untersuchungsfelder von SAFe auf ihre Vor- und Nachteile, im Vergleich zu bisher angewendeten klassischen Projektmanagementmethoden, untersucht. Zu den ausgewählten Untersuchungsfeldern gehören die Projektplanung und Organisation, die Anforderungsdefinition, sowie der Testablauf und dessen Freigabeprozess.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	III
Abkürzungsverzeichnis	V
1 Einführende Bemerkungen	1
1.1 Einführung in die Problemstellung	1
1.2 Abgrenzung des Untersuchungsgegenstandes	1
1.3 Gang der Untersuchung und methodisches Vorgehen	2
1.4 Definitive Grundlagen	2
2 Projektmanagement	3
2.1 Was charakterisiert Projekte.....	5
2.1.1 Charakterisierung anhand der Projektgröße	5
2.1.2 Charakterisierung anhand der Projektart	6
2.1.3 Charakterisierung anhand der Komplexitätsstufe	7
2.2 Herausforderungen in Entwicklungsprojekten.....	9
2.2.1 Standortübergreifenden Projekte	13
2.2.2 Großprojekte	15
2.2.3 Entwicklung von Medizinprodukten	20
3 Vorgehensmodelle im Projektmanagement	24
3.1 Klassische Vorgehensmodelle.....	24
3.1.1 Wasserfall Modell	25
3.1.2 V-Modell	26
3.2 Agile Vorgehensmodelle.....	29
3.2.1 SCRUM	31
3.2.2 Kanban.....	34
3.2.3 SAFe	36
3.3 Auswahl eines geeigneten Vorgehensmodells	44
4 Vorstellung des Entwicklungsprojekts	46
4.1 Projekt Charakteristika	46

4.2	Angewandter Entwicklungsprozess.....	47
4.3	Projektaufbau.....	50
5	Anwendung von SAFe	52
5.1	Projektplanung und Organisation	53
5.1.1	Bisheriger Ansatz - unter Anwendung klassischer Methoden	54
5.1.2	Anwendung von SAFe	57
5.1.3	Vorteile / Nachteile der Anwendung von SAFe.....	61
5.2	Anforderungsdefinition	64
5.2.1	Bisheriger Ansatz - unter Anwendung klassischer Methoden	65
5.2.2	Anwendung von SAFe	67
5.2.3	Vorteile / Nachteile der Anwendung von SAFe.....	70
5.3	Testablauf und Freigabeprozess.....	72
5.3.1	Bisheriger Ansatz - unter Anwendung klassischer Methoden	72
5.3.2	Anwendung von SAFe	75
5.3.3	Vorteile / Nachteile der Anwendung von SAFe.....	76
6	Zusammenfassende Bemerkungen	79
6.1	Zusammenfassung und Fazit.....	79
6.2	Ausblick und weiterer Forschungsbedarf.....	80
Literatur	81
Selbstständigkeitserklärung	83

Abbildungsverzeichnis

Abbildung 1 - Visualisierung der Abgrenzung Produkt, Projekt, Prozess	3
Abbildung 2 – Beispiel einer Klassifizierung nach Projektgröße	5
Abbildung 3 - Einteilung der Komplexitätsstufe anhand der Stacey Matrix	7
Abbildung 4 - Anwendungsbereich klassischer oder agilen Methoden	9
Abbildung 5 - Beispiel eines länderübergreifenden Entwicklungsprojekts.....	14
Abbildung 6 - Beispielhafte Darstellung der Organisationsform Projektkoordination	16
Abbildung 7 - Beispielhafte Darstellung der Organisationsform "Reine Projektorganisation"	17
Abbildung 8 - Beispielhafte Darstellung der Organisationsform "Matrix- Projektorganisation"	19
Abbildung 9 - Eigendarstellung eines beispielhaften Wasserfall Modells	25
Abbildung 10 - Beispielhafte Darstellung eines V-Modells	27
Abbildung 11 - Eigendarstellung eines SCRUM Ablaufs.....	31
Abbildung 12 - Beispielhafte Darstellung eines Kanban Boards	34
Abbildung 13 - The SAFe big picture	37
Abbildung 14 - Essential SAFe	40
Abbildung 15 - Large Solution SAFe	41
Abbildung 16 - Portfolio SAFe	42
Abbildung 17 - Full SAFe.....	43
Abbildung 18 - Aufbau des Produkt Entwicklungsprozesses	47
Abbildung 19 - V-Modell in Kombination mit den Projektphasen.....	50

Abbildung 20 - Darstellung der Projektstruktur	51
Abbildung 21 - Problemstellungen der Projektplanung und Organisation in Projekt Titan	53
Abbildung 22 - Zeitpunkt der Definition von Arbeitspaketen und deren Planungshorizont	54
Abbildung 23 - Darstellung Integrationshäufigkeit	55
Abbildung 24 – Teilnehmer am Change Request Board System Level	55
Abbildung 25 - Implementierung des Solution Agile Release Train	57
Abbildung 26 - Abstimmungen des Solution Agile Release Train, angelehnt an.....	58
Abbildung 27 - Kontinuierlicher Prozess eines Agile Release Trains.....	59
Abbildung 28 - Lösungsansätze mit der Anwendung von SAFe	62
Abbildung 29 - Problemstellungen der Anforderungen in vorliegendem Projekt	64
Abbildung 30 - Überblick über die Projektmeilensteine und ihre zugehörigen Phasen	65
Abbildung 31 - Anforderungs- und Architekturdokumente in respektiven Projektphasen .	66
Abbildung 32 - Beispiel eines Features	69
Abbildung 33 - Beispiel einer User Story in SAFe	69
Abbildung 34 - Lösungsansätze mit der Anwendung von SAFe	71
Abbildung 35 - Problemstellungen des Testablaufs und Freigabeprozesses.....	72
Abbildung 36 - Integrationspunkte und Zeitpunkte finaler Verifikation und Validation	73
Abbildung 37 - Ansatz System Integration & Testing.....	74
Abbildung 38 – Ablauf mit Einbindung virtueller Tests.....	75
Abbildung 39 - Aufbau des Solution Integration Environments	75
Abbildung 40 - Lösungsansätze mit der Anwendung von SAFe	77

Abkürzungsverzeichnis

ART	Agile Release Train
bzw.	beziehungsweise
CAD	Computer-Aided Design
CE	Communauté Européene
CHF	Schweizer Franken
CoP	Community of Practice
CP	Checkpoint
CP FuMo	Checkpoint Funktionsmodell
CP Pilot	Checkpoint Pilot
CP Proto	Checkpoint Prototyp
CP Series	Checkpoint Series
EU	European Union
FDA	Food and Drug Administration
FTE	Full Time Equivalent
GCS	Global Customer Support
IEC	International Electrotechnical Commission
IPMA	International Project Management Association
IPMA	International Product Manager
ISO	Internationale Organisation für Normung
IT	Informationstechnologie
IVDR	In Vitro Diagnostica Regulation
Lean UX	Lean User Experience
MDR	Medical Device Regulation
Medtech	Medizintechnik
Mio.	Million
NFR	Non Functional Requirements
PI	Program Increment
PI Planning	Program Increment Planning
PMI	Project Management Institute
PMO	Project Management Office
QMS	Quality Management System
SAFe	Scaled Agile Framework
SPC	SAFe Program Consultant
z.B.	zum Beispiel

1 Einführende Bemerkungen

Das einleitende Kapitel dient zur Beschreibung der Motivation, sowie der Aufgabenstellung dieser Masterarbeit und gibt den Lesern eine Übersicht der folgenden Kapitel.

1.1 Einführung in die Problemstellung

Agile Entwicklungsmodelle sind aus der Softwareentwicklung nicht mehr wegzudenken. Sie helfen Unternehmen, näher mit Kunden zusammenzuarbeiten und dadurch Kunden besser zu verstehen, wodurch Unternehmen auch mehr auf den Kunden und deren Wünsche eingehen können. Des Weiteren unterstützen agile Entwicklungsmodelle kontinuierliche und stabile Lieferungen, welche durch konsequente Priorisierung und schlanke Prozesse erzielt werden können. Das hilft Unternehmen, sich schnell auf Marktveränderungen anpassen zu können.

Betrachtet man Entwicklungsprojekte in der Medizintechnik, so umfassen diese Projekte meist Software und Hardware, gelten als starr und folgen meist etablierten Prozessen und herkömmlichen Projektmanagement Ansätzen. Diese Produkte unterliegen hohen regulatorischen Anforderungen, welche einen hohen Definitions- und Dokumentationsgrad erfordern.

Die Anforderungen an das Produkt, sowie dessen technische Umsetzungen werden bei vorliegendem Unternehmen bereits in einer frühen Projektphase definiert. Während der Entwicklung von Projekten können Marktveränderungen oder technische Probleme auftreten. Beides resultiert in ungültigen Anforderungen die nachträglich aufwändig angepasst werden müssen.

Ein weiteres Problem bei vorliegendem Projekt ist, dass es durch dessen Größe enorme Abstimmungen benötigt und es bei zugehöriger Kommunikation oftmals Informationsverluste zu beklagen gibt.

1.2 Abgrenzung des Untersuchungsgegenstandes

Die vorliegende Arbeit prüft, inwiefern sich durch die Anwendung von agilen Methoden Vor- bzw. Nachteile bei der Entwicklung eines ausgewählten Medizintechnik Projektes

erzielen lassen. Im Konkreten soll das Scaled Agile Framework (SAFe¹) genauer geprüft werden.

Eine komplette Evaluierung von SAFe im Vergleich mit bisher verwendeter klassischer Projektmanagement Modelle kann innerhalb dieser Arbeit nicht abgehandelt werden. Zu diesen Zweck wurden die drei Bereiche Projektplanung und Organisation, Anforderungsdefinition und Testablauf und Freigabeprozess weiters evaluiert. Diese Bereiche wurden ausgewählt, da es dort häufig zu lösende Problemstellungen gibt oder diese als kritisch für den Erfolg eines Projekts eingestuft werden können.

1.3 Gang der Untersuchung und methodisches Vorgehen

Das erste Kapitel dient der Erläuterung der Problemstellung und grenzt den Umfang der Masterarbeit ab.

Im zweiten Kapitel werden ausgewählte theoretische Grundlagen des Projektmanagement und dessen Charakteristiken erläutert. Zudem beschreibt es eine Auswahl an Herausforderungen des Projektmanagements. Im folgenden dritten Kapitel werden klassische und agile Vorgehensmodelle vorgestellt.

Kapitel vier umfasst das Entwicklungsprojekt, anhand dessen in Kapitel fünf die Anwendung von SAFe geprüft wird. Dabei wird jeweils der Vergleich zwischen bisher angewandten Projektmanagement Methoden und der Anwendung von SAFe gezogen, dabei wird beurteilt welche Vor- bzw. Nachteile sich durch die Anwendung von SAFe ergeben.

Das abschließende Kapitel beinhaltet eine Zusammenfassung, sowie ein Fazit der Arbeit. Des Weiteren liefert es einen Ausblick und definiert den weiteren Forschungsbedarf.

1.4 Definitiorische Grundlagen

Das Projektmanagement dient einem Projektteam als Leitfaden für die Projektdurchführung. Klassisches Projektmanagement plant und steuert das Vorgehen innerhalb eines Projektes und definiert dessen Ziele anhand von Inhalt, Budget und Zeit. Agile Vorgehensmodelle übertragen dem Team mehr Eigenverantwortung und versuchen mittels stetigem Hinterfragen und Anpassen dem Kunden ein optimales Produkt zur Verfügung stellen zu können. SAFe vereint viele bekannten Methoden in einem Framework und ermöglicht es Unternehmen, je nach Größe entsprechend Projektteams zu skalieren. Wie auch bei anderen agilen Methoden ist der Kundennutzen im Zentrum aller Überlegungen.

¹ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

2 Projektmanagement

Projektmanagement, in der klassischen Form, entwickelte sich in der Raumfahrt und dem Anlagebau in den Fünfzigerjahren. Viele der damaligen Methoden haben auch heute noch Bedeutung, beispielsweise die Netzplantechnik, anhand dessen man den kritischen Pfad innerhalb der Abläufe ermitteln kann. Auch Tools des 21. Jahrhunderts bauen auf diesen Erkenntnissen auf, beispielsweise Microsofts Project.

Projektmanagement im Generellen umfasst eine Vielzahl von Charakteristika. Einerseits bildet es eine temporäre Projektorganisation, die die notwendigen Ressourcen im Projekt einbindet, andererseits werden Kompetenzen, Verantwortlichkeiten und Kommunikationspfade klar definiert². Anhand vorhandener Prozesse können Aufgaben definiert und abgearbeitet werden. Während des gesamten Projektablaufs gilt es im Projektmanagement anhand ausgewählter Kennzahlen stets zu prüfen, ob man Abweichungen zu den zuvor definierten Planwerten hat. Anhand dieser, kann ein Projekt zum Erfolg gesteuert werden.

Ein Projekt erstreckt sich von der Idee eines Produkts bis hin zum fertigen Produkt. Aus der Idee eines Produkts wird ein Projektziel formuliert. Prozesse werden innerhalb eines Projektes angewendet und unterstützen das Projekt in der Definition und Abarbeitung von Arbeitspaketen oder Arbeitsschritten. In größeren Unternehmen sind diese Prozesse hoch standardisiert und werden je nach Projektgröße entsprechend auf das Projekt angepasst.

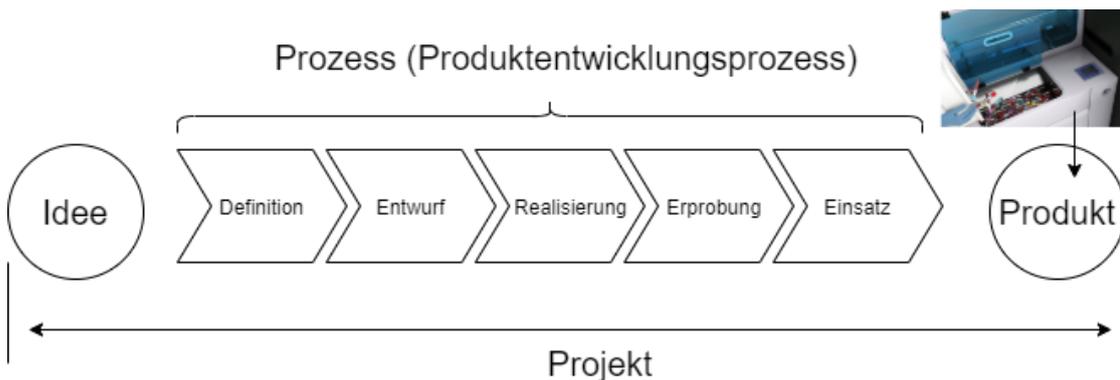


Abbildung 1 - Visualisierung der Abgrenzung Produkt, Projekt, Prozess³

² (Kuster, et al., 2019, S. 2)

³Eigendarstellung in Anlehnung an (Burghardt, 2013, S. 18)

Im Wesentlichen gilt es im Projektmanagement drei eng miteinander verbundene Begriffe voneinander abzugrenzen: Produkt, Projekt, Prozess.

Ein Produkt kann generell ein materielles oder immaterielles Gut sein. In der Medizintechnik handelt es sich meist um eine Kombination aus Hardware und Software die dem Kunden als gesamtheitliches Produkt verkauft wird. Das Projektziel bzw. die Definition eines künftigen Produktes geschieht durch die Analyse der Kundenbedürfnisse unter Einbeziehung der potenziellen Marktveränderungen.

Ein Projekt definiert sich als zielorientiertes „Vorhaben, das im Wesentlichen durch Einmaligkeit der Bedingungen in seiner Gesamtheit gekennzeichnet ist, wie z.B.

- Zielvorgabe
- Zeitliche, finanzielle, personelle und andere Begrenzungen
- Abgrenzungen gegenüber anderen Vorhaben
- Projektspezifische Organisation“⁴

Prozesse kennzeichnen sich durch einen klaren Ablauf von definierten, sich wiederholenden Aufgaben. In der Produktentwicklung bzw. dem zugehörigen Produktentwicklungsprozess gibt es meist eine weitere Untergliederung des Hauptprozesses in einzelne Subprozesse, wie zum Beispiel Definition, Entwurf, Realisierung, Erprobung und Einsatz. Arbeitspakete oder Arbeitsschritte innerhalb des Prozesses haben allgemein gültige Inputs, Ablauf des Arbeitsschritts und erwarteter Output definiert. Innerhalb des Prozesses werden Meilensteine und Checkpoints festgelegt, an denen der Projektstatus mittels Soll / Ist Vergleich erhoben wird. Anhand dieser Erhebungen kann ein Projektfortschritt ermittelt und bei Abweichungen entsprechend gesteuert werden.

Bei erfolgreicher Anwendung unterstützt das Projektmanagement das Unternehmen zuvor festgelegte strategische und operative Ziele zu erreichen.

Obwohl viele klassische Methoden in der Praxis noch immer weit verbreitet sind, stoßen viele Ansätze an ihre Grenzen. In den Bereichen der Produktentwicklung bzw. der Softwareentwicklung geht der Trend seit Jahren in Richtung agiler Entwicklung. Die Gründe dafür sind vielschichtig, einerseits möchte man schneller auf Marktveränderungen und sich ändernde Prioritäten reagieren können, andererseits verändert sich die Art der Mitarbeiterführung. Während in den Fünfzigerjahren vor allem autoritär geführt wurde, so geht der Trend mehr in Richtung Einbindung des einzelnen Mitarbeiters in Entscheidungen. Agile Entwicklungsansätze bauen auf der Selbstorganisation der Teams auf und räumen

⁴ (Burghardt, 2013, S. 19)

insofern dem Individuum mehr Eigenverantwortung ein, was eine Steigerung der Motivation der Mitarbeiter bewirken kann⁵.

2.1 Was charakterisiert Projekte

Da der Begriff Projekt sehr weitgreifend ist, gibt es auch die unterschiedlichsten Ausprägungen von Projekten. Eine Einteilung in Kategorien hilft dem Unternehmen bei der Definition der Projektstruktur und dem Etablieren aller notwendigen Strukturen. Projekte können auf unterschiedliche Weisen charakterisiert werden.

2.1.1 Charakterisierung anhand der Projektgröße

Eine Möglichkeit der Kategorisierung ist die Einteilung der Projektgröße. Hierfür können Kriterien definiert werden, wann ein Projekt in welche Größe einzugliedern ist. Abbildung 2 – Beispiel einer Klassifizierung nach Projektgröße zeigt eine beispielhafte Einteilung in die Größen Micro-, Klein-, Mittel- und Großprojekte. Für die herangezogene Bewertung sind die Gesamtgrößen Personalaufwand und die damit verbundenen Projektkosten ausschlaggebend. In diesem Beispiel werden alle anderen Projektkosten ebenfalls in den Personalaufwand pro Mitarbeiter eingerechnet, beispielsweise die Beschaffung neuer Hardware, IT-Infrastruktur, Arbeitsplatzkosten oder Materialkosten. Für dieses Beispiel wurde ein jährlicher Personalaufwandsatz von 250.000 CHF herangezogen.

Projektgröße	Projektdauer	Jährlicher Personalaufwand	Gesamt Personalaufwand	Gesamt Projektkosten
Micro	< 1 Jahr	< 20 FTE	< 20 FTE	< 5 Mio. CHF
Klein	1 – 2 Jahre	20 – 60 FTE	20 – 120 FTE	5 – 30 Mio. CHF
Mittel	2 – 4 Jahre	60 – 120 FTE	120 – 480 FTE	30 – 120 Mio. CHF
Groß	> 4 Jahre	> 120 FTE	> 480 FTE	> 120 Mio. CHF

Abbildung 2 – Beispiel einer Klassifizierung nach Projektgröße

⁵ (SwissQ - Trends & Benchmarks Report Switzerland, aufgerufen am 05.01.2021, S. 23)

Die Projektgröße kann nicht allgemein gültig definiert werden, da diese unternehmens- und branchenabhängig ist, so würde beispielsweise ein kleines Softwareentwicklungsunternehmen vermutlich ein Großprojekt bei einem wesentlich geringeren gesamt Projektkosten Aufwand festlegen als ein Medtech Unternehmen, dessen Entwicklung Hardware und Software beinhaltet.

Ebenfalls unternehmensspezifisch muss festgelegt werden, welche Regeln bei entsprechender Projektgröße gelten. Einerseits könnte definiert werden, dass bei größeren Projekten Personen mit mehr Seniorität in Schlüsselpositionen engagiert bzw. zugewiesen werden sollten, andererseits sollte das Projektcontrolling in größeren Projekten weit umfangreicher durchgeführt werden.

2.1.2 Charakterisierung anhand der Projektart

Eine weitere Möglichkeit der Klassifizierung ist die **Projektart**, wobei Projekte nach ihrem Zweck kategorisiert werden. Es gibt eine Vielzahl an verschiedenen Projektarten, Beispiele für Bekannte Vertreter sind Forschungsprojekte, Entwicklungsprojekte, Bauprojekte, Rationalisierungsprojekte oder Change Projekte.

Forschungsprojekte werden durch ein eher unklares Forschungsziel charakterisiert. Es besteht eine vage Vorstellung des Ziels, allerdings werden durch Testreihen oder durch exploratorisches Arbeiten Erfahrungen in einem zu erforschendem Gebiet gesammelt. Forschungsprojekte werden meist in einer separaten Abteilung durchgeführt, da die notwendige Kreativität zur Ideenfindung einerseits Freiraum benötigt, andererseits die Ergebnisse nicht streng vorausgeplant werden können.

Entwicklungsprojekte hingegen haben ein klar definiertes Projektziel, beispielsweise die Entwicklung eines neuen Blutmessgerätes im Hochdurchsatz Segment. Das Ziel des Projektteams ist es ein fertig entwickeltes Produkt aus Hardware und Software an die Produktion zu übergeben. Zumeist werden hier dem Unternehmen schon bekannte Elemente bzw. bekannte Technologien verwendet.

In vielen Unternehmen werden Forschungs- und Entwicklungsprojekte kombiniert, wobei das Projektziele ebenfalls klar bestimmt ist, allerdings können beispielsweise während der Entwicklung neue Technologien erforscht bzw. entwickelt werden. Um Projektfortschritte messbar zu machen, werden hier oftmals Zwischenziele definiert, bis zu denen Machbarkeitsstudien durchgeführt werden.

Die Klassifizierung der Projektart dient dem Unternehmen zum Anwenden gleicher oder vergleichbarer Methoden. So würde es vermutlich wenig Sinn ergeben, die gleichen Projektmanagement Methoden in einem Forschungsprojekt, sowie in einem Bauprojekt zu verwenden. Obwohl dieser Vergleich hinkt, zeigt er die Wichtigkeit der zu verwendenden

Methoden auf. Indem das Unternehmen Projekte kategorisiert, können diese auch entsprechend mit ähnlichen Projekten verglichen werden. Das kann einerseits innerhalb eines Unternehmens stattfinden, kann aber auch mit konkurrierten Projekten anderer Unternehmen erfolgen, sofern vergleichbare Kennzahlen zur Verfügung stehen.

2.1.3 Charakterisierung anhand der Komplexitätsstufe

Eine andere Art der Klassifizierung ist die Stacey Matrix. Sie wurde zur Einteilung der Komplexitätsstufe entwickelt und unterteilt diese in Simpel, Kompliziert, Komplex und Chaos. Inhaltlich ähnlich dem Cynefin Framework, soll die Stacey Matrix im Projektalltag genutzt werden können. Die Einteilung soll zudem helfen, dass eine geeignete Entwicklungsmethode den jeweiligen Projekten zugewiesen werden kann.

Die Stacey Matrix zeigt auf der x-Achse die Klarheit der anzuwendenden Technologie und auf der y-Achse die Klarheit der damit verbundenen Anforderungen.

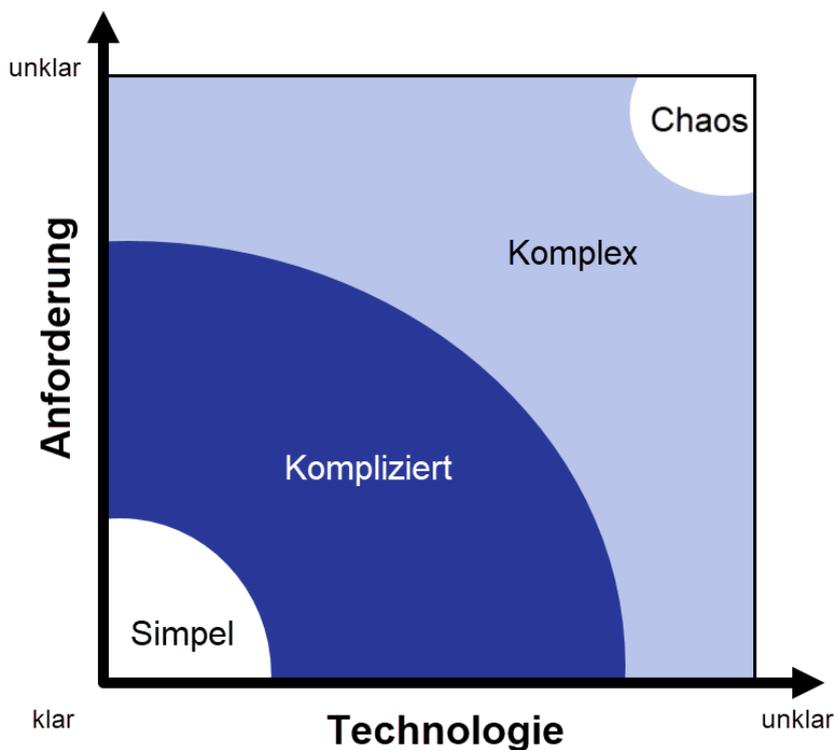


Abbildung 3 - Einteilung der Komplexitätsstufe anhand der Stacey Matrix

Komplexitätsstufe „Kompliziert“

Eine komplizierte Aufgabenstellung ist dann zutreffend, wenn Technologie wie auch die Anforderungen als nicht einfach eingeschätzt werden können. Gleiche oder ähnliche Technologien wurden bereits verwendet, die Anforderungen wurden im Wesentlichen verstanden und können effizient angewendet werden.

Durch das Wissen innerhalb des Unternehmens oder Projektteams werden bestimmte Inputs erwartet, deren Auswirkungen sind bekannt und können in planbare Outputs transformiert werden. Anhand einer detaillierter Planung können Ressourcen abgestimmt und optimal eingesetzt werden.

Komplexitätsstufe „Komplex“

Das komplexe Umfeld wird gekennzeichnet durch schwer oder gar nicht abschätzbare Abhängigkeiten. Viele Faktoren können zugleich Einfluss auf das Projekt nehmen, deshalb eine Planung oft nur eine kurze Gültigkeit hat, wodurch eine Detailplanung oftmals wirkungslos ist oder eventuell sogar behindernd wirkt. Oftmals stehen konkrete Vorgaben oder Best Practices nicht zur Verfügung oder können nicht entsprechend angewendet werden.

Komplexitätsstufe „Chaos“

Handelt es sich um die Komplexitätsstufe Chaos so können selbst kleine Änderungen nicht mehr korrekt abgeschätzt und deren Auswirkung vorab definiert werden. Dieser Bereich ist nach Möglichkeit zu vermeiden und ist nur schwer bis gar nicht managebar.

Anwendung verschiedener Methoden je nach Komplexität

Die Einordnung eines Projektes zwischen der Komplexität kompliziert oder komplex ist für die Anwendung einer geeigneten Methode essenziell.

Im simplen oder komplizierten Bereich sind die dahinterliegenden Aufgaben eines Projektes plan- und berechenbar und klare sowie überprüfbare Vorgaben existieren. Durch diese Eigenschaften können die Methoden des klassischen Projektmanagements zum Einsatz kommen, so kann beispielsweise der optimale Ressourceneinsatz bestimmt werden.

Im komplexen Bereich sind die dahinterliegenden Aufgaben unberechenbar und schwer planbar, die dazugehörigen Anforderungen sind unklar oder ändern sich schnell während des Projektverlaufs. In diesem Bereich entfalten die agilen Entwicklungsmethoden ihre Stärken, indem sie durch permanentes Hinterfragen und dem Mut zum Experimentieren stetig den Status quo anzweifeln und sich dadurch steile Lernkurven entwickeln.

Eine entsprechende Aufteilung wird in Abbildung 4 visualisiert.

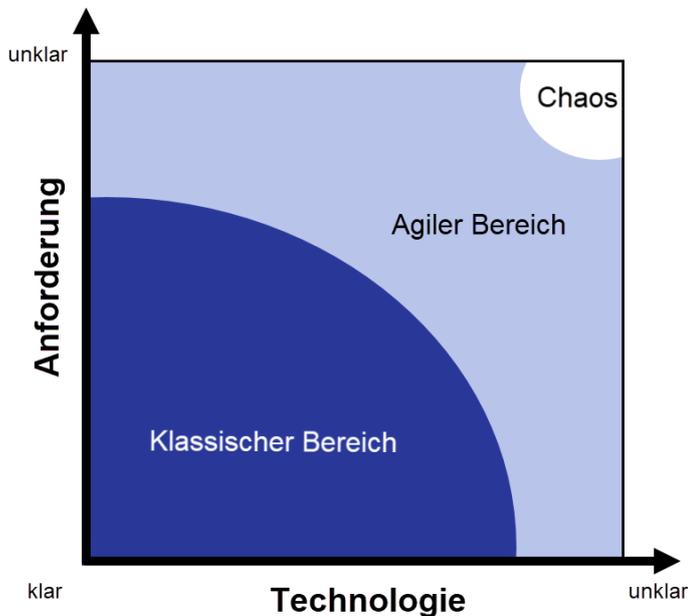


Abbildung 4 - Anwendungsbereich klassischer oder agilen Methoden

Eine inkorrekte oder nicht durchgeführte Einteilung kann bei Anwendung agiler Methoden zu Ressourcenverschwendung führen, während klassische Methoden diese optimal eingesetzt hätten. Werden andererseits im komplexen Bereich klassische Methoden angewendet, so können diese zu Scheinsicherheiten führen. Veränderungen in diesem Bereich müssen allerdings akzeptiert werden, was zu aufwändigen und teureren Änderungsanträgen führt, was oftmals eine Überschreitung des Budgets oder einer Überschreitung der veranschlagten Zeit zur Folge hat.

2.2 Herausforderungen in Entwicklungsprojekten

Bei der Analyse von Literatur und Quellen findet man eine große Vielfalt an Gründen für das Scheitern von Projekten im Allgemeinen. Das liegt vermutlich daran, dass die Gründe von der Unternehmensgröße, der Rahmenbedingungen von Projekten, aber auch von der Branche in dem sich das Unternehmen befindet abhängig sind.

Sehr oft findet man folgende vier Gründe für das Scheitern von Projekten:

- Mangelnde oder schlechte Kommunikation
- Unklare und sich ständig ändernde Anforderungen / Ziele
- Schlechte Projektplanung und Projektcontrolling
- Mangelnde qualifizierte Mitarbeiter und deren Verfügbarkeit

Mangelnde oder schlechte Kommunikation

Dadurch, dass Projekte befristet sind und Projektmitarbeiter in dieser Konstellation oftmals nur über den befristeten Zeitraum zusammenarbeiten müssen, aber auch die Tatsache, dass bei virtueller Kommunikation oftmals die persönliche Ebene verloren geht, tragen dazu bei, dass Kommunikationslücken entstehen und handelnde Akteure demotiviert werden. In Projekten ist es fast unumgänglich, dass es zu Informationslücken, Spannungen und damit verbundenen Konflikten kommt.⁶ Häufige Ursachen für erfolglose Kommunikation können unter anderem folgende Punkte sein:

- Unzureichende Einbindung der relevanten Kommunikationspartner
- Wahl eines ungeeigneten Kommunikationsinstruments
- Fehlendes Vertrauen zwischen den Kommunikationspartnern
- Kommunikation ist belastet durch bestehende Emotionen und Konflikte
- Es existieren vorgefasste Meinungen, wodurch der Inhalt nur selektiv wahrgenommen wird

Das Projektmanagement sollte klare Informationsflüsse und Zuständigkeiten definieren, sicherstellen, dass Besprechungen effektiv und effizient abgehalten werden und Kommunikation zu anderen Organisationsabteilungen oder Abteilungen aktiv betrieben wird.

Eine klare Kommunikationsstrategie hilft die relevanten Zielgruppen und die entsprechenden Kommunikationsinhalte festzulegen. Die Erstellung der Kommunikationsstrategie sollte in Abstimmung mit den Projektstrukturen und der gelebten Kommunikationskultur erfolgen. Wird die Kommunikation nicht ausgewogen gewählt, so kann es zu einer hohen Ressourcenbindung innerhalb des Projektes führen, andererseits wird der Kommunikationspartner durch eine enorme Datenflut überlastet und übersieht die wesentlichen Informationen.

Dem Projektmanagement stehen Tools, wie die Stakeholder Analyse und der davon abgeleitete Kommunikationsplan, zur Verfügung, in denen die Projektkommunikation und deren Inhalte festgehalten werden können. Anhand ausgearbeiteter Kommunikationspläne kann der Kommunikations- bzw. Informationsfluss geprüft und gesteuert werden.

⁶ (Wastian, Braumandl, & von Rosenstiel, 2009, S. 65)

Unklare und sich ständig ändernde Anforderungen / Ziele

Bei der Erstellung von Anforderungen und Zielen von Projekten gibt es eine Vielzahl unterschiedlicher Punkte zu beachten. Zu Beginn eines Projektes sollten sich beteiligte Stakeholder auf die Zielsetzung des Projektes einigen, abgeleitet davon entstehen Anforderungen, welche im klassischen Projektmanagement in einem Lastenheft entsprechend zusammengefasst werden⁷. Anschließend befasst sich das Projektteam mit der Frage, wie diese Anforderungen umgesetzt werden sollen, das Resultat dieser Phase wird im Pflichtenheft dokumentiert. In dieser Phase werden die initialen Anforderungen ermittelt und entsprechend dokumentiert. Achtet das Projektmanagement nicht darauf, dass die Anforderungen mit den Auftraggebern abgestimmt werden, läuft das Projekt Gefahr, etwas zu entwickeln, was gar nicht in deren Sinne ist. Zusätzlich muss auch darauf geachtet werden, dass die Anforderungen von handelnden Personen innerhalb des Projekts verstanden werden. Idealerweise beinhalten die initialen Anforderungen bereits beide Aspekte. Müssen die Anforderungen nachträglich angepasst oder verfeinert werden, muss ebenfalls darauf geachtet werden, dass die Anforderungen noch immer im Sinne des Auftraggebers definiert sind. Ein weiterer Aspekt unklarer oder zu wenig detaillierter Anforderungen kann darin resultieren, dass handelnden Personen Interpretationsspielraum gelassen wird, was ebenfalls zu einer abweichenden Entwicklung führen kann.

Unklare Anforderungen können aber auch bedingen, dass der Auftraggeber bei Rückfragen seine Meinung geändert hat und somit unbewusst Anforderungen dem Projekt hinzugefügt werden, was unter Umständen in zusätzlichen Aufwand bei gleichbleibenden Budget enden kann.

Das Projektmanagement muss zudem sorgfältig darauf achten, dass im Lastenheft definierte Anforderungen nicht stetig verändert werden. Durch konstantes Ergänzen oder Ändern der Ziele können definierte Parameter, wie Budget oder ein vereinbarter Fertigstellungstermin, unter Umständen nicht eingehalten werden. Besteht der Auftraggeber auf Änderungen, sollten auch das Lastenheft und entsprechende Parameter angepasst werden.

Schlechte Projektplanung und Projektcontrolling

Das klassische Projektmanagement sieht drei Etappen vor, die Definitionsetappe, die Planungsetappe und die Steuerungsetappe.⁸

In der Definitionsetappe werden die Zielvereinbarungen, wie zuvor beschrieben, zwischen dem Auftraggeber und den Projektverantwortlichen festgelegt.

⁷ (Kuster, et al., 2019, S. 86)

⁸ (Kuster, et al., 2019, S. 204)

Die Planungsetappe leitet aus zuvor definierten Zielen konkrete Pläne ab. Dazu gehören im klassischen Projektmanagement ein Phasenplan, der Projektstrukturplan, ein Terminplan, ein Ressourcenplan, sowie ein Kostenplan.

Die Steuerungsetappe baut auf den zuvor definierten Zielen der Planungsetappe auf und vergleicht den Soll Zustand des Projekts mit dem tatsächlichen Ist Zustand. Durch das Erkennen von Abweichungen können entsprechende Gegenmaßnahmen ergriffen werden.

Je früher Abweichungen zwischen Soll und Ist erkannt werden, desto kostengünstiger können entsprechende Gegenmaßnahmen ergriffen werden. Sollte in einem Projekt zu wenig oder gar kein Controlling vorhanden sein, so läuft das Projektmanagement Gefahr Abweichungen zu spät oder gar nicht zu erkennen, was zu immensen Abweichungen in Zeit, Budget oder Inhalt führt. Versucht das Projektmanagement das Controlling zu detailliert einzuführen, so läuft man Gefahr Datenfriedhöfe zu erschaffen oder man hat Daten zur Verfügung, die von handelnden Personen gar nicht mehr interpretiert werden können. Insofern sollte ein ausgewogenes Controlling in Projekten etabliert werden.

Mangelnde qualifizierte Mitarbeiter und deren Verfügbarkeit

Projektteams können nur dann effektiv und effizient arbeiten, wenn ihnen die benötigten Ressourcen zur Verfügung stehen. Während Inhalt, Zieltermine und Zielbudget durch gekonnte Verhandlungen erzielt werden können, benötigt es dennoch geeignete Mitarbeiter, die die geforderten Aufträge auch entsprechend durchführen können. Bei der Gewinnung solcher Mitarbeiter, ist aber nicht nur das Gehalt ausschlaggebend, sondern auch andere Faktoren wie Unternehmensstandort, Entwicklungsmöglichkeiten im Unternehmen oder die Außenwirkung des eigenen Unternehmens, aber auch die mit einem Unternehmen konkurrierenden Unternehmen. Eine in der Schweiz durchgeführte Umfrage⁹ zeigt, dass Unternehmen vor allem im Bereich IT, Ingenieurswesen aber auch in Forschung & Entwicklung einen Fachkräftemangel sehen. Das bedeutet, dass Unternehmen den Bedarf an qualifizierten Mitarbeiter haben, diese aber nur schwierig oder gar nicht bekommen.

Möglicherweise muss ein Projekt aber nicht nur außerhalb des Unternehmens um Fachkräfte werben, sondern auch innerhalb eines Unternehmens. In großen Unternehmen ist es nicht unüblich, dass erfahrene Mitarbeiter auch in anderen Projekten begehrt sind. Oft werden Projekte innerhalb des Managements priorisiert, wodurch die wichtigsten Projekte die Ressourcen bekommen, die sie benötigen, auf Kosten des Erfolgs anderer Projekte.

Ein weiterer Faktor ist, dass erfahrene, qualifizierte Mitarbeiter eine gewisse Verhandlungsposition haben, welche in schwierigeren Zeiten vom Mitarbeiter auch ausgenutzt werden könnte. Gibt es beispielsweise disziplinäre Probleme, könnte der Mitarbeiter durch

⁹ (Rundstedt & Partner, 2018), abgerufen am 07-Feb-2021

seine generell gestärkte Verhandlungsposition damit drohen, in ein anderes Projekt oder gar das Unternehmen zu wechseln. In solchen Fällen muss das Projektmanagement Vor- und Nachteile abwägen, da auch auf das Klima innerhalb eines Projektteams geachtet werden muss.

Alle aufgeführten Problemfaktoren können durch richtiges Anwenden von Methoden und den richtigen Personen an den richtigen Stellen gut bewältigt werden, schließlich gibt es auch viele Beispiele für gut geführter und erfolgreicher Projekte.

In den nachfolgenden Unterkapiteln werden weitere Einflüsse auf Projekte durchleuchtet und geprüft, ob diese eher zum Erfolg oder eher zum Misserfolg eines Projektes führen.

2.2.1 Standortübergreifenden Projekte

Durch die Globalisierung und den Fortschritt der Telekommunikationstechnologien werden Projekte vermehrt über mehrere Standorte aber auch über mehrere Länder entwickelt. Dabei ergeben sich für Unternehmen Chance und Risiken.

Ein Unternehmen kann an einem Standort ein Kompetenzzentrum errichten und bündelt dadurch entsprechende Kompetenzen zentral. Abgesehen von den eigenen Stärken des Unternehmens an einem Standort könnte eine entsprechende Wahl durch das Unternehmensumfeld gesteuert werden. Hierbei könnte die Nähe zu wichtigen Lieferanten eine Rolle spielen. Ein weiterer ausschlaggebender Faktor spielt das Engagieren neuer qualifizierter Mitarbeiter. Während es Regionen gibt, an denen es einen erhöhten Fachkräftemangel gibt, gibt es auch Standorte, die das Recruiting von Mitarbeitern begünstigt. Ein Beispiel hierfür ist die Nähe zu einer renommierten Universität, mit der man durch langjährige Kooperationen geeignete Fachkräfte engagieren kann. Weitere Faktoren zur Auswahl eines Standorts können aber auch geltende Arbeitsgesetze, sowie unterschiedliche Mitarbeiterlöhne sein.

Zu einer ausgewogenen Betrachtung der Standortauswahl müssen auch eventuell anfallende Risiken bzw. Schwierigkeiten betrachtet werden. Entwicklungen über verschiedene Länder oder gar Kontinente, können zu unterschiedlichsten Problemen führen. Die Abstimmung zwischen den Teams kann durch die verschiedenen Zeitzonen erschwert werden, außerdem können die Charakteristiken der verschiedenen Arbeitsweisen bzw. kulturellen Probleme herbeiführen.

Abbildung 5 zeigt ein Projekt das über Deutschland, Spanien und die Schweiz entwickelt wird. Am Standort Spanien wird Software entwickelt, die mit Hardware aus Deutschland interagieren muss. Zusätzliche Hard- und Software wird in der Schweiz entwickelt.

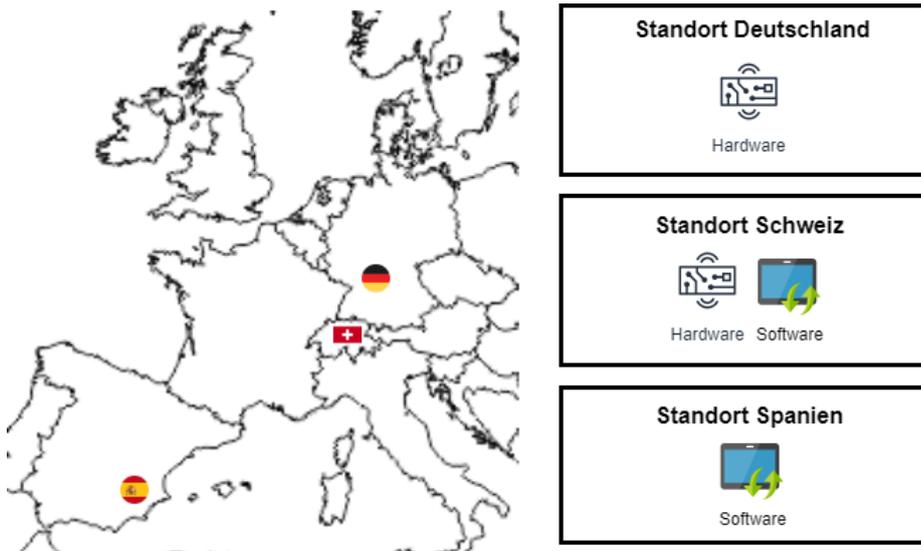


Abbildung 5 - Beispiel eines länderübergreifenden Entwicklungsprojekts

Die sich zu stellende Frage ist, ob länderübergreifende Entwicklungsprojekte die vier zuvor genannten Gründe für das Scheitern von Projekt eher begünstigen oder nicht.

Während in kleinen Projekten, die am selben Standort durchgeführt werden, Mitarbeiter und verschiedene Ebenen des Projekts leicht und mit kurzen Wegen kommunizieren können, ist es bei länderübergreifender Entwicklungen schwerer. Einerseits besteht eine sprachliche Barriere die Kommunikation mindert, andererseits sind diese Mitarbeiter schwerer für ihre Kollegen greifbar, da sie deren Abläufe und Gewohnheiten nicht kennen. So kann beispielsweise eine einfache, aber wichtige Problemstellung nicht innerhalb von kurzer Zeit geklärt werden, da die Mitarbeiter auf das nächste Abstimmungsmeeting warten. Durch den Einsatz von Chats können Mitarbeiter zwar schneller aufeinander zugehen, allerdings birgt dies die Gefahr, dass schnell der Fokus verloren werden kann.

Verfügt das Projekt über nicht ausreichend gute Anforderungen, so Bedarf es zusätzlicher Kommunikation zwischen den Teams und verschiedenen Stakeholdern, damit ein einheitliches Verständnis der Anforderungen geschaffen werden kann. Ähnlich verhält es sich mit konstant wechselnden Anforderungen. Es besteht die Gefahr, dass nur Teile des Teams von den Änderungen erfahren und die Teams in unterschiedliche Richtungen entwickeln. Durch zusätzliche Abstimmungen und gutes Projektmanagement können diese Probleme kontrolliert werden.

Verfügt ein Projekt über schlechte Projektplanung oder ein schlechtes Projektcontrolling, wirkt sich die Entwicklung über mehrere Ländern negativ aus, da selbst kleinere Abweichungen leicht unbemerkt bleiben können. Während bei Entwicklungen am gleichen Standort ein guter Abgleich zwischen den Mitarbeitern kleinere Fehler möglicherweise kompensieren können, ist es in einem solchen Setup nahezu unmöglich.

Besteht ein Mangel an qualifizierten Mitarbeitern, besteht in diesem Umfeld die Chance, dass an einem anderen Standort andere Voraussetzungen gelten und gemeinsam nach

einer Lösung gesucht werden kann. Hier besteht die Möglichkeit der Trennung von Verantwortlichkeiten oder durch gezielte Verlagerung einzelner Aspekte. Fachkräftemangel oder die fehlende Verfügbarkeit der Mitarbeiter ist bei länderübergreifende Entwicklungen eher als positiv beeinflussend zu bewerten.

Nimmt man dieses beispielhafte Entwicklungsprojekt und kombiniert es mit den vier zuvor genannten Gründen für das Scheitern von Projekten, dann kann man erkennen, dass äußerste Vorsicht geboten ist. Während sich die Aufteilung auf mehrere Länder und Standorte positiv auf das Finden von qualifizierten Mitarbeitern auswirken könnte, so können die anderen drei Gründe sich tendenziell zusätzlich negativ auswirken, da diese stark miteinander verknüpft sein können.

2.2.2 Großprojekte

Wie bereits in Kapitel 2.1.1 erläutert, gibt es keine allgemein gültige Definition eines Großprojekts. Für dieses weiterführende Kapitel, wird deshalb die Definition von Kapitel 2.1.1 herangezogen, was eine Projektdauer von mehr als vier Jahren oder einem Budget von mehr als 120 Millionen Schweizer Franken bedeutet. Damit verbunden sind neben dem finanziellen Aufwand auch das Einbinden und Steuern von mehr als 120 Mitarbeitern in das Projekt.

Während in kleinen Projekten, durch klare Verantwortlichkeiten und kurze Kommunikationswege, die Wahl der richtigen Projektorganisation keine große Wichtigkeit hat, gilt die sinnvolle Verteilung der Verantwortlichkeiten und die damit verbundene Projektorganisation als elementar.

Zur Auswahl der geeigneten Projektstruktur müssen unter anderem folgende Punkte berücksichtigt werden:

- Wie schnell und wie häufig müssen Informationen fließen
- Wie kann der Informationsfluss sichergestellt werden
- Zwischen welchen Projektpartnern müssen Informationen ausgetauscht werden
- Wie sollen Konflikte ausgetragen werden und wohin sollen sie eskaliert werden
- Wird ein reger Austausch zwischen Abteilungen benötigt oder ist ein Silodenken sogar gewünscht
- Wie können Warte- und Liegezeiten zwischen Projektmitarbeitern vermieden werden

Im Allgemeinen können Projektorganisationen in drei Formen unterteilt werden, die Projektkoordination, die reine Projektorganisation und die Matrixorganisation¹⁰. Die unterschiedlichen Organisationsformen unterscheiden sich im Wesentlichen in der Aufteilung der Führungs- und Entscheidungskompetenz zwischen der Linienorganisation und der Projektorganisation.

Bei der Organisationsform **Projektkoordination** handelt es sich um die Minimalform einer Projektorganisation, wobei der Projektleiter in einer Stabstelle koordinativ tätig ist, wodurch die Autorität in der Linienorganisation liegt. Der Projektleiter ist für den Ablauf des Projektes, den Informationsfluss in die Linienorganisation und das abgestimmte Vorgehen aller allokierten Mitarbeiter verantwortlich. Aktuelle und künftige Schritte und Maßnahmen müssen immer mit der Linie abgestimmt werden, wodurch ihre konstruktive Beteiligung vorausgesetzt werden muss.

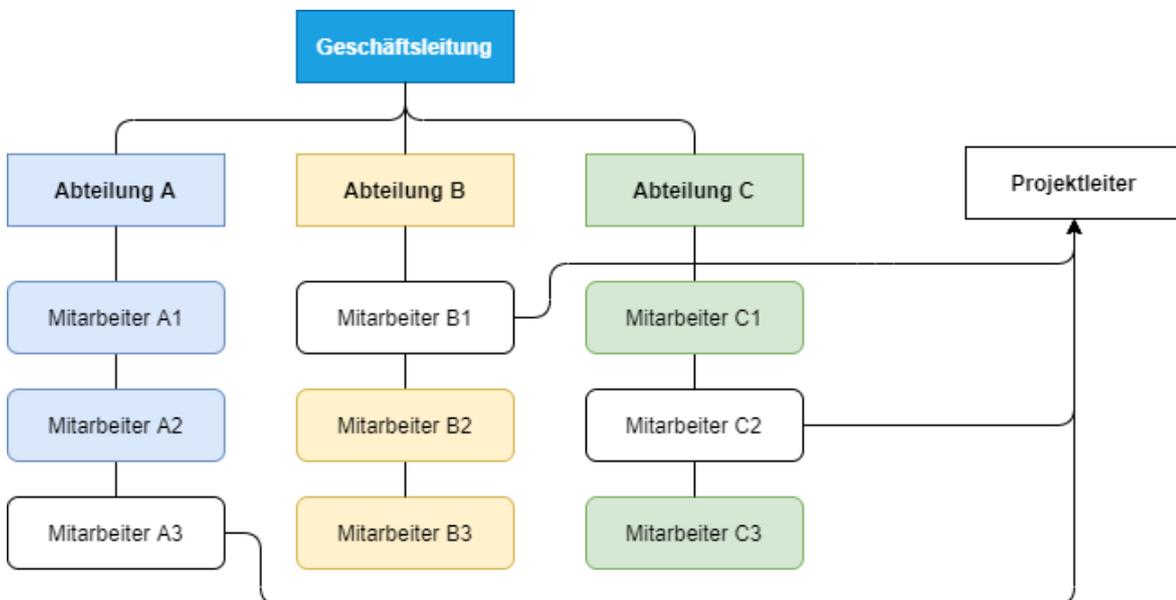


Abbildung 6 - Beispielhafte Darstellung der Organisationsform Projektkoordination

Die Vorteile dieser Organisationsform liegen darin, dass keine Anpassung der Organisationsstruktur vorgenommen werden muss, Erfahrung leicht innerhalb der Abteilung ausgetauscht werden kann und die Organisation eine hohe Flexibilität im Personaleinsatz hat.

Zu den Nachteilen gehört, dass durch das Fehlen einer eigenen Projektorganisation sich kein richtiger Teamgedanke etabliert, wodurch sich möglicherweise niemand dem Projekterfolg verschrieben fühlt. Ein weiterer Nachteil ist, dass durch den koordinativen Auf-

¹⁰ (Kuster, et al., 2019, S. 120)

wand, dem Überzeugen und Abstimmen mit der Linie die Reaktionsgeschwindigkeit beeinträchtigt sein kann.

Die Projektkoordination hat vor allem ihre Stärken, wenn es sich um kleine und hoch standardisierte Projekte handelt, bei Großprojekten ist von dieser Organisationsform abzuraten.

Die **reine Projektorganisation** bildet für das Projekt eine eigenständige Organisationseinheit. Das gesamte Projektteam inklusive dem Projektleiter arbeiten zu hundert Prozent im Projekt selbst, der Projektleiter erhält Führungsaufgaben und alle Entscheidungskompetenzen. Die Mitarbeiter werden somit aus den jeweiligen Abteilungen losgelöst und der neuen Organisationsstruktur angehängt. Die dabei entstandenen Lücken in der Stammorganisation müssen entsprechend nachbesetzt werden, was zu Mehrkosten führt.

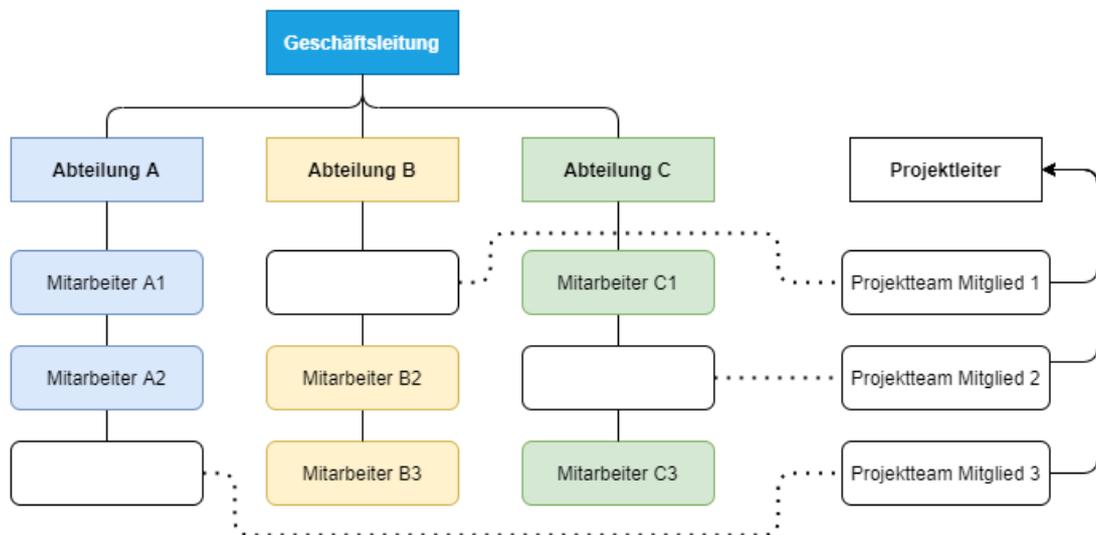


Abbildung 7 - Beispielhafte Darstellung der Organisationsform "Reine Projektorganisation"

Die Vorteile liegen vor allem darin, dass eine effiziente Organisation gebildet wird, die bei Problemen nur kurze Reaktionszeiten benötigt und einem Projektteam, das klar auf die Projektziele ausgerichtet werden kann. Zudem ist es in einer solchen Projektorganisation leichter den Teamgedanken zu fördern. Durch die volle Entscheidungskompetenz des Projektleiters können zudem auch Entscheidungen schneller erzielt und umgesetzt werden, die Entscheidungskompetenz ist zudem klar verteilt.

Der Nachteil dieser Organisationsform aus Sicht des Projekts liegt nur darin, dass es wenig Flexibilität in der Personalwahl hat, da nicht zeitweise Spezialisten für Teilaufgaben eingesetzt werden können. Alle Mitarbeiter müssen für die reine Projektorganisation entsprechend aus der eigenen Organisation oder neu rekrutiert werden. Während aus Projektsicht die Vorteile dieser Organisationsform klar überwiegen, liegen die Nachteile dieser eher auf Seite des Mitarbeiters. Nach Abschluss eines Projektes, bleibt offen, wie der Mitarbeiter weiter beschäftigt wird. Natürlich ist es im Interesse des Unternehmens Mitar-

beiter längerfristig zu beschäftigen, allerdings ist am Ende des Projektes der Mitarbeiter prinzipiell ohne konkrete Aufgaben, bis er einem neuen Projekt zugewiesen werden kann. Ein weiterer Nachteil aus Sicht des Mitarbeiters ist, dass wenn es zu Problemen innerhalb des Projekts kommt, es keine neutrale Anlaufstelle gibt, die hier vermitteln kann oder den Mitarbeiter gar in ein anderes Projektumfeld zuweisen kann. Das Interesse des Projektleiters besteht nämlich vor allem darin, mit dem Projekt Erfolge zu erzielen und weniger darin, Mitarbeiterförderung zu betreiben. Finden Mitarbeiter ein solch negatives Umfeld über Jahre vor, kann bei ihnen der Eindruck entstehen ausgebeutet zu werden, wodurch sie möglicherweise das Unternehmen verlassen könnten.

Die reine Projektorganisation wird bei zeitkritischen Projekten, die eine erhöhte Durchschlagskraft benötigen empfohlen. Idealerweise haben diese Projekte wenig Berührungspunkte mit Aufgaben oder Bezugspunkten außerhalb der Projektorganisation. Für Großprojekte ist diese Organisationsform geeignet, wobei das Unternehmen darauf achten soll, Mitarbeiter geeignete Entwicklungsmöglichkeiten bieten zu können, damit diese auch langfristig dem Unternehmen zugehörig bleiben und ihre gewonnenen Erfahrungen aus dem Großprojekt in andere Projekte wieder einfließen lassen können.

Die **Matrix Projektorganisation** ist eine Mischform aus der Projektkoordination und der reinen Projektorganisation. Die Projektleitung und die Linienorganisation teilen sich die Verantwortlichkeiten und Kompetenzen entsprechend auf. Die getroffene Aufteilung sollte vor allem nach den Bedürfnissen des Projekts ausgerichtet werden, eine genaue Aufteilung muss aber zwischen Projekt und Linie getroffen werden. Wichtig ist dabei, dass sich alle ihren Rollen und Aufgaben bewusst sind und sich entsprechend danach verhalten. Mitarbeiter können Projekten in dieser Form auch nur teilweise oder voll zu kürzerer Dauer eingesetzt werden, wodurch diese Form eine hohe Abstimmung zwischen den Führungspositionen fordert, wobei es in der Praxis oftmals zu Konflikten führen kann. Prinzipiell können Mitarbeiter auch mehreren Projekten zugewiesen werden, was zu zusätzlichen Spannungen führen kann, sollte es zu einer Verschiebung des Mitarbeiterbedarfs kommen. Eine klare Priorisierung der Projekte kann hier vorgenommen werden, sofern diese von allen Beteiligten respektiert wird.

Diese Organisationsform ist im klassischen Projektmanagement am häufigsten in der Praxis anzufinden.

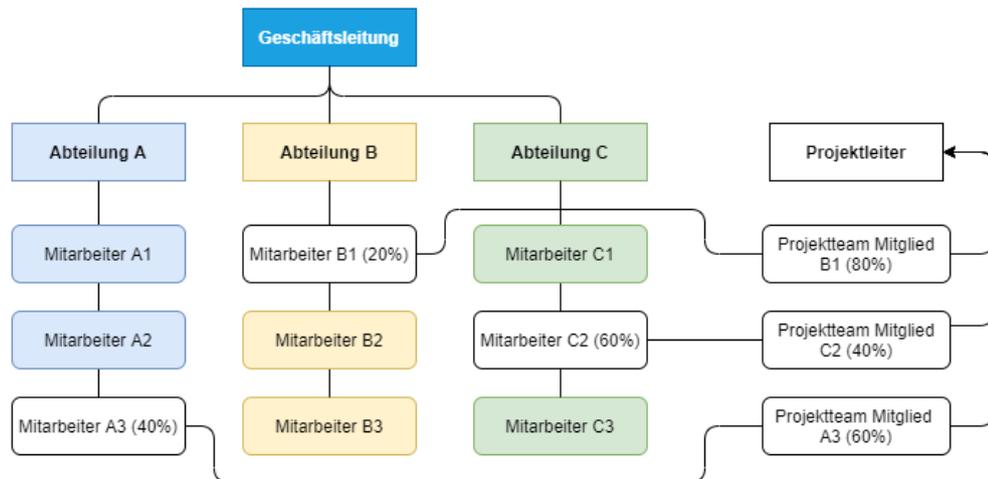


Abbildung 8 - Beispielhafte Darstellung der Organisationsform "Matrix-Projektorganisation"

Der Vorteil der Matrix Projektorganisation ist, dass sich alle Projektmitarbeiter dem Projekterfolg verpflichtet fühlen. Zudem liegt die volle Entscheidungskompetenz bei dem Projektleiter, der flexibel neues Personal, auch nur für eine begrenzte Zeit, von der Linie anfordern kann. Der Mitarbeiter hat zudem den Vorteil, dass die Linienorganisation vermittelnd bei Konflikten wirken kann, sich aber auch für die Weiterbildung und Entwicklung des Mitarbeiters interessiert. Im Falle eines Projektes das zu Ende geht, vermittelt die Linie den Mitarbeiter in ein neues Projekt, wodurch eine ständige und konstante Auslastung gegeben sein sollte.

Die Matrix Projektorganisation hat vor allem zwei wesentliche Nachteile. Erstens können nicht klar definierte Verantwortlichkeiten zwischen der Linien- und der Projektautorität zu Kompetenzkonflikten in der Führung führen, die aber auch zu verunsicherten Mitarbeitern führen kann, da diese in dieser Organisationsform zwei Vorgesetzte haben. Sollten beide unterschiedliche Ziele an den Mitarbeiter geben, führt das zu großen Verunsicherungen. Zweitens benötigt die Matrix Projektorganisation eine erhöhte Kommunikationsbereitschaft. Es muss zu jeder Zeit allen Beteiligten klar sein, welche Informationen, über welches Medium an wen fließen müssen. Sollte eine Abstimmung der relevanten Informationen zwischen Linie und Projekt nicht gegeben sein, kann es wiederum zu unterschiedlichen Zieldefinitionen kommen.

Diese Projektform kann für Großprojekte geeignet sein, sofern die Verantwortlichkeiten klar aufgeteilt sind, die Linie die Kapazitätsplanung bzw. Ressourcenplanung mit dem Projekt abstimmt, Konflikte offen angesprochen werden können und alle Unternehmensbereiche am Projekterfolg interessiert sind und sich nicht gegenseitig konkurrieren.

Generell werden Projektorganisationen oftmals zu schnell gebildet, ohne ausgewogen auf die Vor- und Nachteile einer jeden zu achten. Wichtig ist zudem die Projektbedürfnisse konkret zu charakterisieren, um diese bei der Umsetzung entsprechend einfließen lassen zu können.

Für Großprojekte stehen generell die reine Projektorganisation und die Matrix Projektorganisation als geeignete Modelle zur Verfügung. In beiden Modellen sollte darauf geachtet werden, dass die Organisation möglichst schlank gehalten wird, um den damit verbundenen Koordinations- und Abstimmungsaufwand zu minimieren. Des Weiteren sollte vermieden werden, dass Mitarbeiter Mehrfachrollen zugewiesen werden, da dadurch die Verantwortlichkeiten eventuell in Konflikt treten können oder nicht mehr klar ersichtlich sind. Durch den grundsätzlich hohen Kommunikationsbedarf in Großprojekten muss in beiden Modellen genau darauf geachtet werden, dass der Informationsfluss gewährleistet wird.

2.2.3 Entwicklung von Medizinprodukten

Für Hersteller kann es einerseits attraktiv wirken, ein Produkt als Medizinprodukt zu kennzeichnen, da dessen Vermarktung dadurch unter Umständen ansprechender oder hochwertiger auf potenzielle Kunden wirken könnte, andererseits bringt das Entwickeln von Medizinprodukten erhöhte Anforderungen mit sich. Medizinprodukte, die in der Europäischen Union neu zugelassen werden sollen, unterliegen der Medizinprodukteverordnung (MDR) und In-vitro-Diagnostika-Verordnung (IVDR). Die MDR wurde im Amtsblatt der Europäischen Union im April 2017 veröffentlicht.¹¹

Gemäß MDR¹² Artikel 2 wird ein Medizinprodukt wie folgt definiert:

Ein „Medizinprodukt bezeichnet ein Instrument, einen Apparat, ein Gerät, eine Software, ein Implantat, ein Reagenz, ein Material oder einen anderen Gegenstand, das dem Hersteller zufolge für Menschen bestimmt ist und allein oder in Kombination einen oder mehrere der folgenden spezifischen medizinischen Zwecke erfüllen soll:

- Diagnose, Verhütung, Überwachung, Vorhersage, Prognose, Behandlung oder Linderung von Krankheiten,
- Diagnose, Überwachung, Behandlung, Linderung von oder Kompensierung von Verletzungen oder Behinderungen,
- Untersuchung, Ersatz oder Veränderung der Anatomie oder eines physiologischen oder pathologischen Vorgangs oder Zustands,
- Gewinnung von Informationen durch die In-vitro-Untersuchung von aus dem menschlichen Körper — auch aus Organ-, Blut- und Gewebespenden — stammenden Proben ...“

¹¹ (Amtsblatt der Europäischen Union, 2017)

¹² (Amtsblatt der Europäischen Union, 2017), Artikel 2

Handelt es sich bei dem zu entwickelnden Produkt um ein Medizinprodukt, so muss schon zu Beginn der Entwicklungsphase darauf geachtet werden, dass das Produkt Gesetzes-, Verordnungs- und Normenkonform entwickelt wird.

Die europäischen Medizinprodukte Verordnungen (MDR und IVDR) werden mit nationalen Gesetzen und Verordnungen weiter konkretisiert und ergänzt. Die Verordnungen geben die Rahmenbedingungen für die Entwicklung aller Medizinprodukte, wodurch die zu erfüllenden Anforderungen an die Hersteller nicht spezifisch sind. Die Europäische Union erlaubt es Herstellern den Nachweis der Verordnungen über harmonisierte Normen zu bringen. Harmonisierte Normen werden von der EU im Amtsblatt entsprechend publiziert.

Beispiele für in der Vergangenheit harmonisierte Normen in der Medizinentwicklung sind:

- ISO 13485 – Medizinprodukte – Qualitätsmanagement
- ISO 14971 – Anwendung des Risikomanagements
- IEC 62304 – Medizingeräte Software und deren Lebenszyklus

Die harmonisierten Normen helfen dem Hersteller eines Medizinprodukts somit, detaillierte Rahmenbedingungen zur Entwicklung des Medizinproduktes zu bekommen.

Eine der wichtigsten Normen dabei ist die ISO 13485, die definiert welche Anforderungen an das Qualitätsmanagementsystem des Medizinprodukte Herstellers gelten. Die ISO 13485 fordert unter anderem die Erstellung eines Qualitätsmanagementhandbuchs, das definiert, wie der Hersteller konkret Medizinprodukte entwickeln möchte, die dafür benötigten Ressourcen dazu bereitstellen möchte und wie die dafür benötigten Prozesse überwacht werden sollen.

Die ISO 13485 ist der Norm ISO 9001 generell sehr ähnlich. Beide Normen definieren die Anforderungen eines zu etablierenden Qualitätsmanagement. Die ISO 13485 ist branchenspezifisch im Bereich der Medizinproduktentwicklung.

Die ISO 13485 fordert weitere Aktivitäten, deren Planung und Durchführung wiederum in weiterführenden harmonisierten Normen beschrieben werden, die aber je nach Produkt ebenfalls durchgeführt werden müssen. Beispielsweise fordert die ISO 13485 die Durchführung des Produktrisikomanagements, das wiederum in der ISO 14971 näher beschrieben wird. Das Produktrisikomanagement betrachtet potenzielle Risiken für Patienten, den Anwender des Medizinprodukts oder Dritte. Dabei gilt es, bereits während der Entwicklung eventuelle Risiken zu identifizieren und entsprechende Gegenmaßnahmen im Produktdesign vorzunehmen.

Neben dem Produktrisikomanagement sind unter anderem die Nachverfolgbarkeit der Produkte und die Validierung der eingesetzten Software während der Entwicklung und Produktion wesentliche ergänzende Anforderungen an den Hersteller von Medizinprodukten, im Vergleich zur ISO 9001.

Bei der Entwicklung von Software wird meist gemäß IEC 62304 entwickelt, damit der Hersteller das Einhalten der gesetzlich verpflichtenden Forderung des Software Lebenszyklusprozess nachweisen kann. Besondere Vorsicht ist für Hersteller geboten, sollte die notwendige Klassifizierung tiefer ausfallen. Die IEC 62304 fordert hier weniger Dokumentationsaufwand während des Entwicklungsprozesses. Möchte der Hersteller allerdings ebenfalls in den USA sein Produkt zulassen, so muss eine entsprechende Evaluierung des „level of concern“ durchgeführt werden¹³. Während die Kriterien zur Evaluierung weitestgehend mit jenen der IEC 62304 deckungsgleich sind, unterscheiden sich die darauffolgenden Konsequenzen deutlich. Die FDA fordert, dass alle Dokumentation unabhängig des „level of concern“ im Entwicklungsprozess erzeugt werden, es müssen aber bei der initialen Einreichung, während des Zulassungsverfahrens, nicht alle Dokumente mitgeliefert werden. Bei Rückfragen müssen diese Dokumente aber ebenfalls vorgewiesen werden.

Der Hersteller muss sich somit bereits zu Beginn der Entwicklungsphase im Klaren sein, was der geplante Einsatzzweck seines Produktes ist und in welchen Regionen er das Produkt als Medizinprodukt zulassen möchten. Darauf abgestimmt, müssen entsprechend die Entwicklungsprozesse abgestimmt und Normen bzw. Verordnungen eingehalten werden.

Während in den USA ein Medizinprodukt zugelassen werden muss, müssen Hersteller in Europa das Medizinprodukt nicht zulassen, sondern in einem Konformitätsbewertungsverfahren nachweisen können, dass sie entsprechende Verordnungen eingehalten haben.

Ein idealtypisches Vorgehen zur Erlangung der CE-Kennzeichnung könnte in 6 Schritten definiert werden:

1. **Festlegung der Zweckbestimmung**

Der Hersteller legt fest, wer sein Produkt in welchem Umfeld verwenden soll. Dabei wird auch geregelt, ob es sich um ein Medizinprodukt handelt, oder nicht.

2. **Ermittlung der gesetzlichen Anforderungen und Klasse des Medizinprodukts**

Anhand der festgelegten Zweckbestimmung, können die dafür relevanten Verordnungen identifiziert und eine entsprechende Klassifizierung des Produkts gemacht werden. Die Klassifizierung definiert die später relevanten Aktivitäten, die entsprechend im Qualitätsmanagement System verankert werden müssen und ein entsprechendes Konformitätsbewertungsverfahren einleiten.

3. **Etablierung des Qualitätsmanagement Systems**

Das Qualitätsmanagement System beinhaltet eine Vielzahl an Vorgabedokumenten wie Verfahrens- und Arbeitsanweisungen, sowie deren Aufzeichnungen. Abgeleitet vom geltenden Konformitätsbewertungsverfahren, muss das Qualitätsmanagement

¹³ (FDA, 2005, S. 4)

System zertifiziert werden. In großen Unternehmen gibt es meist ein großes Qualitätsmanagement System, das alle Medizinprodukte des Herstellers beinhaltet. Da sich die Produkte oftmals wesentlich in ihrer Zweckbestimmung und somit auch in ihren gesetzlichen Anforderungen unterscheiden, können die Hersteller definieren, dass gewisse Bereiche ihres Qualitätsmanagements Systems nur von bestimmten Produkten oder Produktgruppen durchgeführt werden müssen.

4. Entwicklung des Produkts, sowie Erstellung der technischen Dokumentation

Während der Entwicklung des Produkts, müssen die Vorgaben des Qualitätsmanagements Systems eingehalten werden, Ergebnisse sowie Abweichungen müssen nachvollziehbar dokumentiert werden. Als Bestandteil der Entwicklungsphase müssen auch Nachweise über die korrekte Implementierung der Funktionen des Produkts erbracht werden, die anhand der zuvor erstellten Dokumentation verifiziert werden. Ebenfalls muss unter anderem die Effektivität der Risikomitigierungen des Medizinprodukts in dieser Phase nachgewiesen werden.

5. Erklärung der Konformität des Produkts

Der Hersteller erklärt seine Konformität des Medizinprodukts gegenüber dem geltenden Qualitätsmanagement System. Die relevanten Daten bzw. Akten werden einer Benannten Stelle (staatlich überwachte Auditier- und Zertifizierungsstellen) zur Verfügung gestellt, das Produkt wird registriert, sodass anschliessend das CE-Kennzeichen auf dem Produkt ausgewiesen werden darf.

6. Produkt am Markt überwachen

Nachdem das Produkt am Markt verkauft werden darf, enden die Pflichten eines Medizinherstellers nicht. Während des gesamten Produktlebenszyklus muss der Hersteller den Markt auf Rückmeldungen und aufgetretenen Problemen untersuchen und je nach Schweregrad geeignete Maßnahmen ergreifen. Diese können von einer Behebung des Problems bis hin zum Produktrückruf reichen.

Zusammenfassend kann gesagt werden, dass das Einreichen von Produkten als Medizinprodukt einen großen Aufwand für die Hersteller mit sich bringt, was gerade kleine Firmen, wie auch Startups oftmals vor große Herausforderungen stellt. Etabliert der Hersteller ein geeignetes Qualitätsmanagement System, zertifiziert dieses und arbeitet gemäß der Vorgaben, so kann ein Medizinprodukt erfolgreich zugelassen werden.

3 Vorgehensmodelle im Projektmanagement

Das klassische Projektmanagement, sowie die klassische Produktentwicklung waren ein wesentlicher Bestandteil erfolgreicher Meilensteine der Vergangenheit. In den letzten 20 Jahren wurde es in der Produkt- und Softwareentwicklung allerdings immer schwieriger, mit den vorhandenen Prozessen und Methoden, auf die Schnelligkeit des Markts zu reagieren.

Parallel dazu haben vor allem in der Softwareentwicklung agile Entwicklungsmethoden begonnen sich zu etablieren. Agile Methode wurden oftmals aus anderen Methoden weiterentwickelt und an die neu entstanden Bedürfnisse angepasst.

In der Praxis existieren heutzutage nach wie vor klassische wie auch agile Vorgehensmodelle, das vorliegende Kapitel erklärt ausgewählte Methoden.

3.1 Klassische Vorgehensmodelle

Anfang des 20. Jahrhunderts gab es keine etablierten oder standardisierten Methoden.

Der Ursprung der klassischen Vorgehensmodellen liegt Anfang des 20. Jahrhunderts als Henry Laurence Gantt das heute noch immer populäre Gantt Chart entwickelt. Mitte des 20. Jahrhunderts wurden durch die Weltraumentwicklungen neue Ablaufmodelle entwickelt, die einerseits eine zügige Erreichung von einmaligen Projektzielen anstrebten und andererseits hohe Qualität sicherstellen konnten. In etwa zur selben Zeit wurden, durch die das Project Management Institut PMI und die International Projekt Management Association IMPA Vorgehensmodelle entwickelt und Methoden standardisiert.¹⁴

Klassische Vorgehensmodelle haben einen plangetriebenen Projektablauf und gelten heutzutage oftmals als starr und behäbig. Zu Beginn des Projekts werden Ziele für den Umfang bzw. Inhalt des Projektes, die zeitliche Komponente und die damit verbundenen Kosten definiert. Abgeleitet von diesen Zielen werden Pläne erstellt, die während der Projektdurchführung entsprechend durch entsprechende Gremien und mittels Projekt Controlling überwacht werden. Gibt es Abweichungen zu den definierten Planzielen, wird über verschiedene Methoden versucht frühzeitig gegenzusteuern. Wenn möglich wird versucht das Projekt wie ursprünglich geplant durchzuführen, sollte es notwendig sein wird nochmals durch eine Erhöhung der Ressourcen oder eine Verschiebung der Markteinführung versucht gegenzusteuern.

¹⁴ (Kuster, et al., 2019, S. 51)

3.1.1 Wasserfall Modell

Das Wasserfall Modell ist ein sequenzielles Vorgehensmodell und sieht vor, dass zuerst eine Phase abgeschlossen wird, bevor die nächste gestartet wird. Für jede Phase wird genau definiert, welche Arbeitspakete darin erledigt und welche Fortschritte erzielt werden sollen. Sobald eine Phase komplett abgeschlossen wird, kann mit der nächsten Phase begonnen werden, eine Überlappung der Phasen ist nicht vorgesehen. Findet man in einer späteren Projektphase ein Problem, das aus einem Ergebnis einer früheren Phase resultiert, so wird in diese Phase zurückgesprungen. Ein Beispiel hierfür könnte sein, dass man während der Testphase identifiziert, dass eine Anforderung der Analyse Phase unklar formuliert und damit ein falsches Design und eine falsche Entwicklung umgesetzt wurde. In diesem Beispiel muss die entsprechende Anpassung der Anforderung der Analyse Phase durchgeführt und alle folgenden Phasen neu durchlaufen werden.¹⁴

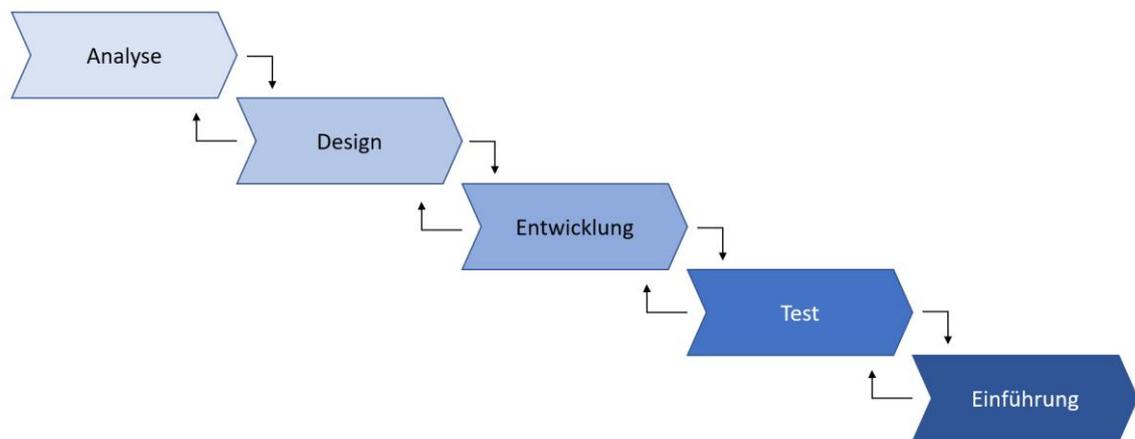


Abbildung 9 - Eigendarstellung eines beispielhaften Wasserfall Modells

Bevor eine Phase abgeschlossen wird, werden Reviews durchgeführt. Die Teilnehmer eines Reviews unterscheiden sich von Unternehmen zu Unternehmen und können von Fachpersonen zur Prüfung von inhaltlichen Themen, aber auch vom Management zur Prüfung der Wirtschaftlichkeit des Projekts besetzt sein. Eine Mischung aus beidem wäre empfehlenswert, denn nur wenn beide Aspekte erfolgsversprechend beurteilt werden, ist eine erfolgreiche Umsetzung und Einführung am Markt sinnvoll und rentabel.

Wie bereits erwähnt, sieht dieses Modell es prinzipiell nicht vor, dass eine Phase startet, bevor eine andere erfolgreich abgeschlossen wird. Viele Unternehmen lassen aber dennoch einen vorbehaltlichen Abschluss der Phase zu und verknüpfen diesen mit weiteren Auflagen. Ein Beispiel könnte sein, dass das Review ergibt, dass ein Paket an Anforderungen in der Analyse Phase vergessen wurde und definiert, dass dieses innerhalb von

¹⁴ (Timinger, 2017, S. 38)

vier Wochen dem Review Team nachgereicht werden muss. Dennoch kann mit den restlichen Anforderungen die Design Phase entsprechend beginnen.

Der Vorteil an diesem Modell wird oftmals darin gesehen, dass man sich schwieriger an Details verzetteln kann. Das liegt vor allem daran, dass Unternehmen ein Modell über Jahre entwickelt und verfeinert haben. Mitarbeiter, die bereits länger im Unternehmen sind, kennen das Modell bereits und können es wie in anderen Projekten zuvor abarbeiten. Das Wasserfall Modell erlaubt zudem eine strukturierte Planung bereits sehr frühzeitig im Projekt, was es erlaubt Abschätzungen über die benötigten Ressourcen und einem geplanten Einführungsdatum zu geben.

Als problematisch am Wasserfall Modell kann gesehen werden, dass es von der Projektidee bis hin zur Markteinführung sehr lange dauern kann. Gerade in Zeiten der Globalisierung, in denen es oftmals neue Märkte für Unternehmen aber auch neue Mitbewerber gibt, ist es schwierig Kunden oftmals über Jahre hinzuhalten. Ein weiteres Problem des Wasserfall Modells ist die Schwierigkeit sich an neue Situationen anzupassen. Ändert sich der Markt während der Projektdurchführung, so stellt sich dem Unternehmen die Frage, ob das Projekt wie geplant durchgeführt werden soll, oder ob man das Projekt stoppt und eventuell neu plant.

3.1.2 V-Modell

Diese Vorgehensweise findet vor allem im reguliertem Umfeld seine Anwendung, wo das Endprodukt hohe Sicherheitsansprüche erfüllen muss. In der Medizintechnik ist die Anwendung des V-Modells weit verbreitet und heutzutage nach wie vor üblich.

Bei dem V-Modell handelte es sich um ein sequenzielles Vorgehensmodell, bei dem auf der linken Seite des V-Modells von der Grobkonzeption bis hin zur Detailkonzeption der Projektgegenstand entworfen wird. Dabei kann je nach Systemgröße auch in unterschiedliche Ebenen unterteilt werden.

Stakeholder können betriebsinterne oder betriebsexterne Interessensgruppen sein, deren Einbindung in die Anforderungsphase des Produkts entweder vertraglich geregelt, gesetzlich vorgegeben oder vom Unternehmen als wichtig erachtet wird.

Beispiel für betriebsinterne Stakeholder und deren Interessen:

- Kundenservice: Wartbarkeit des Produkts
- Produktion: möglichst einfache Produktion, klare verständliche Baupläne, Einhaltung der Produktionskostenrichtlinien
- Design Team: Einheitliches Aussehen der Produkte, Benutzer Freundlichkeit

Beispiele für betriebsexterne Stakeholder und deren Interesse:

- Kunde: problemlösendes Produkt, Innovation, finanzierbares Produkt
- Behörden: Einhaltung der notwendigen Richtlinien

Die gesammelten Stakeholder Anforderungen müssen konsolidiert werden und anschließend durch das Unternehmen auf ihre Relevanz für das Projekt bewertet werden.

Die Anforderungen der System Ebene leiten sich aus den gesammelten Anforderungen der Stakeholder ab und definieren eine Grobkonzeption des künftigen Produkts. In diesem Zusammenhang werden erste Architektur Entscheidungen getroffen und diese mittels Dokumenten und Anforderungen definiert und festgehalten.

In der Komponenten Ebene werden die definierten System Anforderungen weiter verfeinert und mit technischen Inhalten kombiniert. Technologische Limitierungen sollten evaluiert und in den Designentscheidungen berücksichtigt, sowie in den technischen Anforderungen dokumentiert werden.

Während des Modul Designs werden konkrete Überlegen zur Umsetzung der Vorgaben gemacht. Üblicherweise werden durch weiteres Runterbrechen der Anforderungen mehrere Module je Komponente gebildet. Die Module müssen untereinander klar erkennbare Schnittstellen haben. Einer der wichtigsten Faktoren ist, dass sie wartungsfähig, änderungsfähig und testbar sind, da ansonsten aus regulatorischer Sicht eine Sicherstellung der Qualität auch bei künftigen Wartungen oder Weiterentwicklungen, nach Produkteinführung, nicht als gewährleistet gesehen wird.

Sind alle Architekturentscheidungen gefallen und die dazu gehörigen Anforderungen schriftlich dokumentiert, findet eine Implementierung von Hardware und oder Software statt.

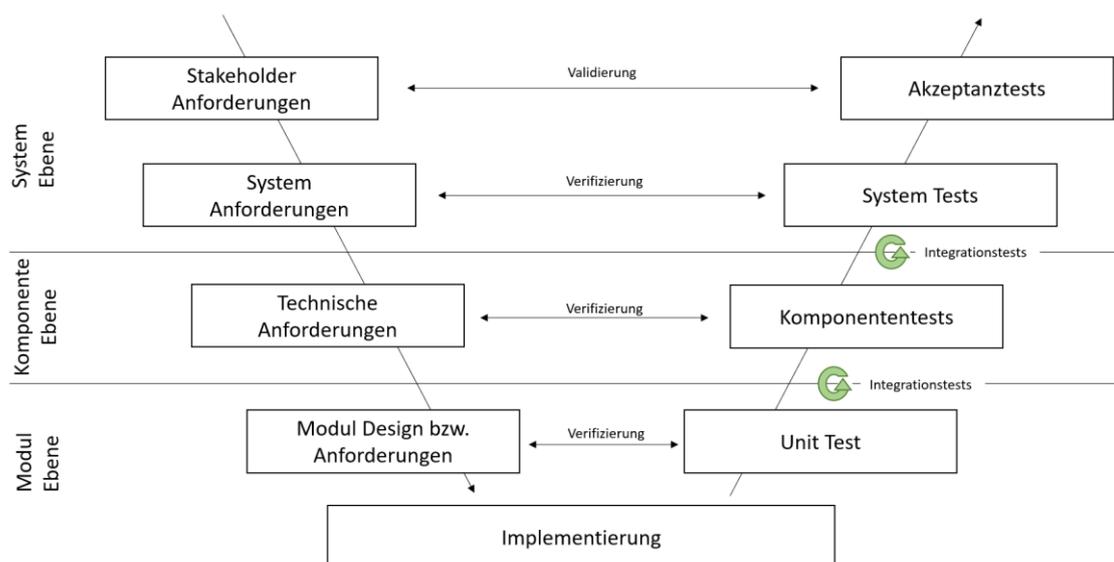


Abbildung 10 - Beispielhafte Darstellung eines V-Modells

Auf der rechten Seite des V-Modells finden die entsprechenden Verifizierungs- und Validierungsaktivitäten statt, wobei hier von sehr detailliert aufwärts Richtung grob detailliert verifiziert wird. Generell werden die zuvor definierten Anforderungen durch Testfälle abgetestet, um die beschriebene Funktionalität auf ihre Korrektheit zu prüfen. Desto früher ein Problem oder Fehler in der Entwicklung entdeckt werden kann, desto günstiger kann das Problem gelöst werden. Werden Probleme erst in den System Tests gefunden, so müssen die Änderungen wieder auf Korrektheit und potenzielle Nebeneffekte auf der kompletten rechten Seite des V-Modells durchgetestet werden, was zu weit höheren Kosten führt.

Unit Tests finden meist durch die Entwicklungsabteilung selbst statt. Dabei können unterschiedliche Methoden angewendet werden. In der Softwareentwicklung werden beispielsweise Code Reviews oder automatisierte Tests durchgeführt. In der Hardwareentwicklung können beispielsweise Dauer und Belastungstests durchgeführt werden.

Bevor alle Module gemeinsam in den Komponententests verifiziert werden, wird ein Modul nach dem anderen Schrittweise integriert und geprüft, ob sich durch dessen Integration neue und offensichtliche Fehlverhalten entdecken lassen. Sollten erste Integrationstests erfolgreich abgeschlossen worden sein, so kann mit den Komponententests fortgefahren werden. Komponenten werden gegen ihre Anforderungen getestet, meist durch Unterstützung von Simulatoren, die das Verhalten der zu diesem Zeitpunkt fehlenden anderen Komponenten imitieren.

Nach erfolgreichem Abschluss der Komponententests werden alle Komponenten zu einem Gesamtsystem integriert und durch erste Integrationstests getestet. Die Durchführung der System Tests findet meist ohne Simulatoren statt und prüft, ob alle zuvor definierten System Anforderungen eingehalten werden können.

Während des gesamten Prozesses werden Fehler erfasst und auf ihre Kritikalität bewertet. Sollte das Projektteam zum Schluss kommen, dass nach Durchführung der System Tests alle schwerwiegenden Fehler behoben wurden und verbleibende Fehler unkritisch bzw. akzeptabel sind, so kann mit der nächsten Phase gestartet werden.

Hier werden die Stakeholder Anforderungen mit Akzeptanztests validiert, was zumeist mit den jeweiligen Stakeholdern durchgeführt wird. Gibt es bisher unentdeckte Fehler in der Implementierung, so müssen diese eventuell vor Markteinführung behoben werden. Weitaus schlimmer gestaltet sich das Szenario, in dem die Stakeholder Bedürfnisse falsch verstanden oder falsch dokumentiert wurden und somit über alle Ebenen falsch umgesetzt wurden. In solchen Fällen ist mit größeren Zeit und Budgetabweichungen zu rechnen.

Sollten die Akzeptanztests positiv verlaufen, kann eine Einführung des Produktes in die Wege geleitet werden.

3.2 Agile Vorgehensmodelle

Agile Vorgehensmodelle haben ihren Ursprung in den Neunzigerjahren in der Softwareentwicklung, als immer komplexer werdende Softwarelösungen zu immer häufigeren Verzögerungen und den damit verbundenen Budgetüberschreitungen führten. Ein weiterer negativer Aspekt war, dass entwickelte Software oftmals größere Qualitätsprobleme beinhalteten.

Als eine mögliche Ursache für diese Entwicklung könnte, die häufig während der Entwicklung ändernden Kundenbedürfnisse und die damit verbundenen Änderungen genannt werden. Die Einführung des Internets ermöglichte einerseits das einfache Erschließen neuer Märkte, aber andererseits boten sich dadurch auch neue Konkurrenten. Zu dieser Zeit wurden erste agile Methoden wie Scrum und Extreme Programming entwickelt und veröffentlicht. Mitbegründer dieser Modelle trafen sich im Jahr 2001, um zusätzlich über eine Alternative zu den bisher sehr schwergewichtigen und dokumentationsgetriebenen Software-Entwicklungsprozessen zu diskutieren.

Dabei entstanden vier agile Leitwerte und zwölf Prinzipien, die bis heute existieren und worauf viele Methoden und Modelle auch noch heutzutage beruhen.

Die folgenden vier agilen Leitwerte wurden von 17 Personen um Kent Beck definiert, unterzeichnet und sind öffentlich unter www.agilemanifest.org zugänglich:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- **Individuen und Interaktionen** mehr als Prozesse und Werkzeuge
- **Funktionierende Software** mehr als umfassende Dokumentation
- **Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung
- **Reagieren auf Veränderung** mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“¹⁵

Die Leitwerte zeigen, dass das agile Entwicklungsteam den Kunden und dessen Wünsche im Fokus aller Überlegungen haben möchte. Es zeigt außerdem, dass starre und aus Sicht des Entwicklungsteam sinnlose Aufgaben und Dokumente nicht der Hauptfokus sein sollten.

Zusätzlich zu den vier agilen Leitwerten wurden auch außerdem 12 Prinzipien definiert, die weiterführend den Gedanken verfeinern sollen.

¹⁵ (agilemanifest.org, abgerufen am 03. Oktober 2021)

„Wir folgen diesen Prinzipien:

- Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
- Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
- Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
- Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
- Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
- Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
- Funktionierende Software ist das wichtigste Fortschrittsmaß.
- Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
- Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
- Einfachheit -- die Kunst, die Menge nicht getaner Arbeit zu maximieren -- ist essenziell.
- Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
- In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an. ¹⁶

Im Fokus agiler Vorgehensmodelle liegen somit die Kundenwünsche, die durch flache Hierarchien, eine dynamische Organisation und kurzen Kommunikations- und Entscheidungswegen in hoher Produktqualität erzielt werden sollen.

Die Einführung von agilen Vorgehensmodellen kann unterschiedliche Beweggründe haben. Ein möglicher Grund eines Unternehmens für die Einführung ist, dass das Unternehmen schneller Produkte auf den Markt bringen möchte und gegebenenfalls dadurch auch schneller auf Marktänderungen reagieren kann. Ein weiterer Grund aus Unternehmenssicht ist, dass agile Entwicklung auch eine iterative Lieferung von Resultaten und Prototypen beinhaltet, wodurch die Wahrscheinlichkeit minimiert werden soll, dass kritische Probleme erst am Ende eines Projekts sichtbar werden.

Aus individueller Sicht beinhaltet ein Wechsel in die Agilität, dass Mitarbeiter mehr Eigenverantwortung bekommen. In agilen Vorgehensmodellen steht die Selbststeuerung eines Einzelnen im Vordergrund. Führungsaufgaben werden nicht mehr nur von einer oder wenigen Personen durchgeführt, sondern auf das komplette Team aufgeteilt bzw. durch dieses abgedeckt, wodurch bei dem Team Denken und Handeln zusammengeführt wird.¹⁷

¹⁶ (agilemanifest.org, abgerufen am 03. Oktober 2021)

¹⁷ (Kuster, et al., 2019, S. 46)

3.2.1 SCRUM

Scrum ist eine Methode, die vor allem in der Softwareentwicklung bereits früh Fuß gefasst hat. Scrum orientierte sich an japanischen Produktentwicklungen des Lean Managements, in dem mit schlanken und innovativen Wegen neue Produkte von der Idee zur Markteinführung realisiert wurden.

Scrum sieht vor, dass am Ende einer zeitlichen Periode ein funktionstüchtiges Produkt Inkrement entsteht, welches eigenverantwortlich vom Scrum Team entwickelt und getestet wurde.

Eine Periode wird als Sprint bezeichnet, welche vorab definiert wird und konstant gleich bleibt, wodurch Arbeitsinhalte vorgängig entsprechend geschätzt und eingeplant werden müssen (time-boxed development). In vielen Entwicklungsteams wird die Dauer eines Sprints auf zwei bis vier Wochen festgelegt, wobei oft auch die Komplexität des bearbeiteten Produkts ausschlaggebend ist. Nachdem ein Sprint abgeschlossen wurde, startet sofort der nächste Sprint mit der gleichen zuvor definierten Kadenz.

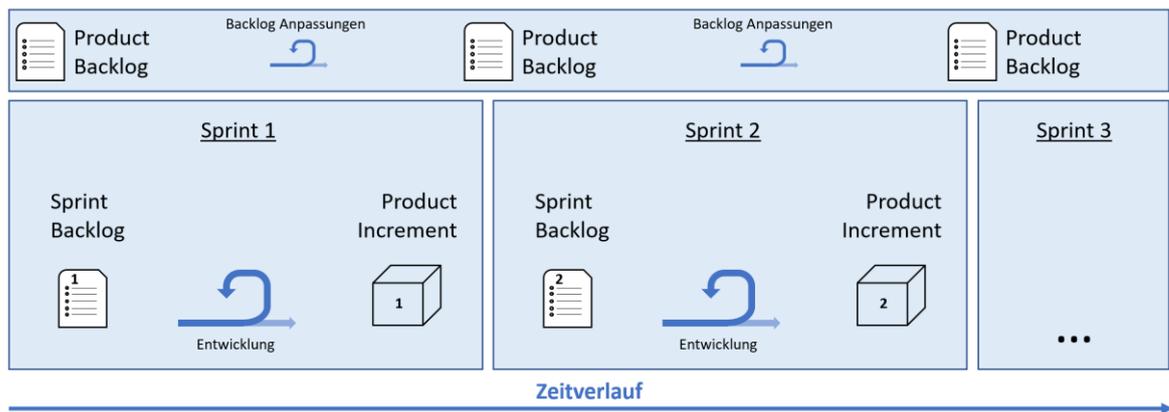


Abbildung 11 - Eigendarstellung eines SCRUM Ablaufs

Die Inhalte werden vor Start eines jeden Sprints priorisiert im Product Backlog definiert. Das Scrum Team ist für entsprechende Aufwandsabschätzungen, so genannte Story Points selbst verantwortlich und plant mittels pull Prinzip den Inhalt des Sprints selbst.

Am Ende eines jeden Sprints entsteht ein Produkt Inkrement, welches funktionstüchtig ist und auch an Kunden potenziell ausgeliefert werden könnte. Zusätzlich findet am Ende des Sprints ein Sprint Review und eine Retrospektive statt. Das Sprint Review dient dazu, das Arbeitsergebnisse des Sprints einer erweiterten Gruppe von Stakeholdern zu präsentieren. Dabei geht es einerseits darum, die erfolgte Arbeit sichtbar zu machen, aber auch um sicherzustellen, dass das Backlog korrekt implementiert wurde. Sollte es Unklarheiten gegeben haben oder weiterer Überarbeitungsbedarf notwendig sein, so kann ein Product Backlog nochmals überarbeitet oder verfeinert werden, um im neuen Sprint korrekt umgesetzt zu werden.

In der Retrospektive hinterfragt sich das gesamte Team, um sich stetig verbessern zu können. Notwendige Anpassungen können sofort ab dem nächsten Sprint umgesetzt werden.

Scrum sieht im Wesentlichen drei Rollen vor: den Product Owner, den Scrum Master und das Projektteam, ein klassischer Projektleiter ist hier nicht vorgesehen. Seine Rolle wird auf den Product Owner und den Scrum Master aufgeteilt.

Der Product Owner führt die fachliche und inhaltliche Steuerung durch und vertritt die Interessen der Stakeholder. Wie bereits erläutert, sollen innerhalb von Scrum die Wege schlank gehalten werden, insofern ist es essenziell, dass der Product Owner die Kundenbedürfnisse kennt und versteht, sich regelmäßig mit ihnen und anderen Stakeholdern austauscht und sein Wissen entsprechend direkt dem Entwicklungsteam vermitteln kann. Der Product Owner arbeitet nahe mit dem Team zusammen, ist aber nicht Teil davon. Er gibt mit dem Product Backlog das „was“ vor, das Team entscheidet aber selbst über das „wie“.

Das Product Backlog wird durch den Product Owner erstellt, er nimmt ebenfalls dessen Priorisierung vor. Product Backlog Items mit hoher Priorität werden im Product Backlog Oben geführt, während unwichtigere Items weiter Unten geführt werden. Product Backlog Items können als eine Art Anforderungen gesehen werden und werden für das Entwicklungsteam als User Stories geschrieben. Eine User Story folgt typischerweise dem Prinzip: Wer, Was und Warum? Ein Beispiel dafür könnte sein: Als Kunde (wer), brauche ich eine Auswertung meiner angefallenen Kosten patientenspezifisch (was), damit ich die Kosten der Versicherung oder dem Patienten korrekt verrechnen kann (warum).

Der Scrum Master ist Methodenspezialist und unterstützt vereinfacht gesagt das Team, die Anforderungen des Product Owners entsprechend umsetzen zu können.

Der Scrum Master unterstützt durch Moderieren das Beseitigen von Hindernissen. Das Moderieren des Scrum Masters besteht darin, dass er dem Team die Scrum Prozesse und Inhalte von Meetings klar macht und dem Team darin hilft zum Erfolg zu kommen. Ein Beispiel dafür könnte sein, dass ein Teammitglied ein unlösbares Problem schildert und der Scrum Master durch das Zerlegen des Problems anderen Teammitgliedern hilft, so dass sie Lösungsansätze anbieten können. Die Kunst des Scrum Masters besteht darin, zwar einerseits moderierend das Team zu unterstützen, sich andererseits nicht in inhaltliche Teamaufgaben einzumischen. Das Beseitigen von Hindernissen kann viele Facetten haben und ist ein weiterer wichtiger Aspekt des Scrum Masters. Ein Beispiel könnte sein, dass dem Team notwendige Informationen fehlen, ohne die ein Teammitglied blockiert ist. Ein strukturelles Problem könnte sein, dass dem Team notwendige Lizenzen nicht mehr oder nicht in ausreichender Form zur Verfügung stehen oder, dass andere Abteilungen immer versuchen, dem Team andere Aufgaben zu geben, was sich negativ auf die Produktivität auswirkt.

Die Aufgabe des Scrum Master ist es eine Umgebung zu schaffen, in denen das Team auch solche Probleme anspricht, damit darauf reagiert werden kann. Am Besten werden

die Probleme festgehalten, es wird nach Lösungen gesucht und die Probleme werden entsprechend eliminiert, damit sich das Team auf den Inhalt des Sprints konzentrieren kann.

Der Vorteil von Scrum für Teams ist, dass sie sich auf das Wesentliche konzentrieren können, Kreativität Raum geboten wird und es die Möglichkeit für eine offene und transparente Kultur schafft. Aus Unternehmens- und Kundensicht sind die Vorteile, dass nach Sprint Ende immer ein Product Increment entsteht, das dem Kunden ausgeliefert werden könnte oder ausgeliefert wird. Die frühe Einbindung des Kunden auch in kleinere Aspekte birgt die Chance, dass der Kunde genau das bekommt, was er auch benötigt und minimiert das Risiko für das Unternehmen, am Ende ein gescheitertes Projekt zu haben.

Als Nachteil könnte gesehen werden, dass Scrum sich zwar für kundenspezifische Lösungen bestens eignet, allerdings hinterfragt werden kann, ob dies dann auch für einen Massenmarkt tauglich ist. Des Weiteren kann bei Scrum der Nachteil gesehen werden, dass der Gesamtüberblick über ein Gesamtprojekt fehlt, da immer nur von Sprint zu Sprint geplant wird, wodurch es bei großen Projekten zu sehr langen Gesamtentwicklungsphasen kommen kann.

3.2.2 Kanban

Kanban wurde ursprünglich von Toyota entwickelt, um mehr Flexibilität und eine Erhöhung der Effizienz in deren Produktion zu erzielen. Das Grundsätzliche Motto lautet „Stop starting, start finishing“¹⁸.

„Kanban beruht auf sechs Praktiken:

- Mach Arbeit sichtbar.
- Limitiere den Work in Progress, also die Menge an begonnener Arbeit.
- Manage Flow.
- Mach Prozessregeln explizit deutlich.
- Implementiere (häufige) Feedbackmechanismen.
- Führe gemeinschaftlich Verbesserungen durch. “¹⁹

Kanban zielt auf konzentriertes Erledigen einer selbstbestimmten Aufgabe hin. Mehrere Aufgaben sollen nicht von einem Mitarbeiter gleichzeitig erledigt werden, eine Aufgabe soll nach der nächsten Abgeschlossen werden.

Generell werden allerdings keine Abläufe oder Strukturen dem Team vorgegeben, Kanban zielt wie andere agile Entwicklungsmethoden auf die Selbstorganisation der Mitarbeiter bzw. des Teams. Das Team verfügt über ein Backlog an Aufgaben, die erledigt werden müssen. Mitarbeiter, die eine Aufgabe abgeschlossen haben, nehmen eine passende Aufgabe und starten deren Bearbeitung. Sobald diese Aufgabe erledigt wurde, wird sie als solches markiert und kann zum nächsten Bearbeitungsschritt freigegeben werden.

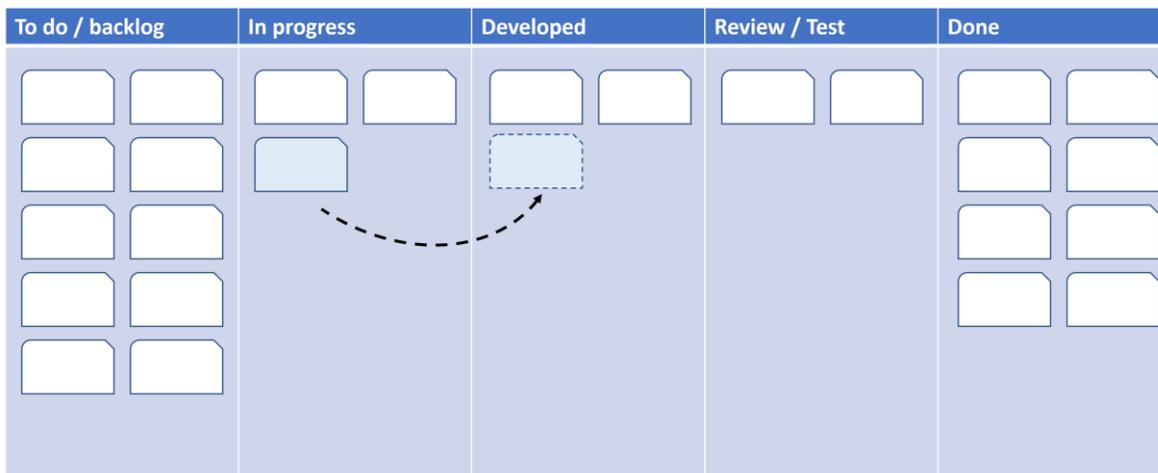


Abbildung 12 - Beispielhafte Darstellung eines Kanban Boards

¹⁸ (Kusay-Merkle, 2018, S. 44)

¹⁹ (Kusay-Merkle, 2018, S. 43)

Typischerweise werden die Aufgaben an Kanban Boards sichtbar gemacht. Die beispielhafte Darstellung eines Kanban Boards in Abbildung 12 zeigt einen Aufbau über 5 Spalten. Die Gestaltung des Boards ist jedem Team selbst überlassen, sodass es auf dessen Bedürfnisse angepasst werden kann. Abbildung 12 wäre ein Beispiel für die Organisation eines Teams aus Softwareentwicklung und Softwaretest. Ein Mitarbeiter nimmt eine Aufgabe aus dem backlog und startet deren Bearbeitung, indem er die Karte der Aufgabe vom Kanban Board löst und sie in die nächste Spalte klebt. Sobald er die Aufgabe erledigt hat, wird die Aufgabe als developed markiert bzw. in dessen Spalte geklebt. Sobald ein Tester des Teams verfügbar ist, kann er starten die Aufgabe zu prüfen, indem er das Post It in die Spalte Review / Test verschiebt. Nach erfolgreichem Abschluss kann die Aufgabe in Done verschoben werden.

Der Vorteil von Kanban ist, dass es sich ohne großen Aufwand innerhalb eines kleineren Teams anwenden lässt. Obwohl es unterstützende IT Tools für Kanban Boards gibt, müssen diese nicht verwendet werden, meistens kann man mittels Whiteboard die Arbeit sogar besser sichtbar machen. Ein weiterer Vorteil von Kanban ist, dass man es leicht an seine Anforderungen anpassen kann. Sollte das Team nach erster Anwendung Verbesserungswünsche haben, können diese jederzeit eingearbeitet werden.

Kanban selbst gibt weder eine Vorgabe zur Größe der Aufgaben, noch definiert es die Größe oder Entwicklungsdauer von potenziellen Lieferungen an den Kunden. Diese Tatsache macht es möglich, Kanban einerseits in sehr vielen unterschiedlichen Bereichen einzusetzen, andererseits ist es auch möglich Kanban mit anderen Methoden zu kombinieren.

3.2.3 SAFe

Die Abkürzung SAFe steht für Scaled Agile Framework und wurde von Dean Leffingwell im Jahr 2011 veröffentlicht. Seit der Einführung wird das Modell stetig überarbeitet, so dass es mittlerweile fünf große Updates der Methode gegeben hat, bei denen bekannte Aspekte verfeinert und neue Aspekte eingeführt wurden.

Bei SAFe handelt es sich um ein Framework, das unter der Website <https://www.scaledagileframework.com>²⁰ kostenlos aufgerufen werden kann. Den Überblick über alle darin verwendeten Methoden und Prinzipien gibt das in Abbildung 13 dargestellte big picture. Alle darauf angezeigten Icons können angeklickt werden, wodurch man zu jeweiligen weiterführenden Informationen gelangt. Die auf der Website publizierten Inhalte beziehen sich auf die SAFe Version 5.1.

Weiteres wird durch die Autoren Richard Kastner und Dean Leffingwell das Buch SAFe Distilled²¹ veröffentlicht, das die Informationen aufgreift und erklärend versucht sie miteinander in einen Kontext zu setzen. Das Buch bezieht seinen Inhalt auf SAFe Version 5.0.

Zusätzlich gibt es eine Vielzahl an, durch Scaled Agile Framework, zertifizierte Trainingsangebote, bei denen man sich auf SAFe, dessen Einführung im Unternehmen oder auf spezielle Rollen schulen lassen kann. Unternehmen können selbst entscheiden, ob sie nur wenige Personen ausbilden wollen, die wiederum weitere Personen im Unternehmen schulen und die Einführung unterstützen, oder aber, ob sie das Unternehmen durch Consultants in dem Change Prozess begleiten lassen.

SAFe ist eine Methode die viele bekannte Methoden aus Lean Management bzw. Produktentwicklung, agilen Methoden und Systemdenkweise anwendet und versucht diese miteinander in Einklang zu bringen. SAFe ermöglicht es Unternehmen, Aspekte von SAFe, je nach deren Bedürfnissen und Teamgrößen, durch Skalierung anzuwenden oder wegzulassen. SAFe hat es zum Ziel Unternehmen bei der Transformation ins digitale Zeitalter zu unterstützen, damit diese kompetitive Produkte und Lösungen ihren Kunden in kürzester Zeit anbieten können. Die Methoden unterstützen die Abstimmung und Zusammenarbeit von mehreren agilen Teams.

²⁰ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

²¹ (Leffingwell & Knaster, SAFe distilled; Achieving Business Agility with the Scaled Agile Framework, 2020)

empfohlen werden. Das Framework erlaubt es Unternehmen aber, dass je nach deren Bedürfnissen, Punkte hinzugefügt oder entfernt werden.

Die *Vision* beschreibt die Zukunft der künftigen Gesamtlösung und definiert künftige Kunden- und Stakeholder Anforderungen.

Die *Roadmap* zeigt die bereits definierte Abfolge von geplanten Aktivitäten, Ereignissen und Meilensteinen.

Bei *Meilenstein* werden Fortschritte zu einem definierten Zeitpunkt sichtbar gemacht, Meilensteine in SAFe sind festgelegte Zeitpunkte, Produktinkremente oder Lernerfolge.

Die *Shared Services* sind geteilte Ressourcen zwischen den Entwicklungsteams und bieten den Team Services und Tools an, die zur Realisierung des Projekterfolgs benötigt werden, beispielsweise Sicherheitsexperten.

CoP (Community of Practice) ist eine informelle Gruppe an Personen, die praktische Erfahrungen besitzen und diese mit anderen Mitgliedern teilen möchten. Diese Aspekte können technischer, aber auch geschäftlicher Natur sein.

Das *System Team* ist ein agiles Team, das Unterstützung bei dem Aufbau und der Verwendung der continuous delivery pipeline bietet und selbst end to end Tests durchführt, wo notwendig.

Bei *Lean UX (Lean User Experience)* handelt es sich um einen kontinuierlichen Verbesserungsprozess, der es zum Ziel hat, Prinzipien zur einheitlich konstanten Benutzererfahrung den Entwicklungsteams anbieten zu können.

Metrics. Das höchste messbare Ziel in SAFe ist die Messbarkeit von funktionierenden Produkten. Zusätzlich bietet SAFe eine Reihe an Kennzahlen, die den Fortschritt anhand Zwischenzielen und Langzeitzielen der ARTs (Agile Release Trains) und des Portfolios messbar machen.

3. Die Foundation (Abbildung 13 - grüner Rahmen)

Die Foundation bildet die Basis an unterstützenden Prinzipien, Werten und Mindsets die für die Implementierung und Anwendung von SAFe benötigt werden. Diese unterstützen das Unternehmen dabei Produkte erfolgreich abzuliefern und der Skalierbarkeit. Die Basis ist für alle SAFe Konfigurationen gleich.

Lean-Agile leaders sind Führungspersonen, die SAFe täglich den Teams vorleben, anderen dabei helfen sich zu entwickeln und durch ihr Coaching dabei helfen, die bestmöglichen Ergebnisse zu erzielen.

Die *Core Values* stellen die Ideale und Überzeugungen in SAFe dar und bilden die Grundlagen dessen, worauf SAFe aufbaut: Ausrichtung, verinnerlichtes Qualitätsbewusstsein, Transparenz und Programmausführung.

Lean-Agile mindset beinhaltet, dass Lean Agile Leaders lebenslang Neues dazu lernen wollen und wollen ihr Wissen über Lean und Agile Prinzipien im Unternehmen verbreiten und dazu ermutigen es entsprechend umzusetzen.

SAFe bietet Unternehmen eine *Implementation Roadmap* an, die aufzeigen soll, wie sich ein existierendes Unternehmen in ein Lean-Agile Technologie Unternehmen wandeln kann.

Die SAFe Program Consultants (SPCs) sind Personen, die über ein fundiertes Wissen über SAFe verfügen und das Unternehmen beim Wandel unterstützen können.

SAFe baut auf zehn *SAFe principles* auf, die als Basis allen Tuns stehen:

- Prinzip 1: Nehme eine ökonomische Sichtweise ein
- Prinzip 2: Denke in Gesamtsystemen
- Prinzip 3: Sei variabel; behalte Optionen aufrecht
- Prinzip 4: Entwickle inkrementell mit schnellen und integrierten Lernzyklen
- Prinzip 5: Meilensteine sollen auf objektiven Bewertungen der Arbeitssystemen beruhen
- Prinzip 6: Visualisiere und limitiere Arbeitspakete, verkleinere Batches und manage Warteschlangenlängen
- Prinzip 7: Verwende einen festen Rhythmus und synchronisiere dich mit Bereich übergreifenden Planungen
- Prinzip 8: Fördere die intrinsische Motivation von wertvollen Mitarbeitern
- Prinzip 9: Dezentralisiere die Entscheidungsfindung
- Prinzip 10: Wertschöpfung als Basis allen Tuns

Essential SAFe configuration

Essential SAFe ist die kleinste Konfiguration und beinhaltet die notwendigsten Rollen, Meetings und Artefakte, die dazu benötigt werden, dass ein Unternehmen kontinuierlich Business Lösungen liefern kann. Als Basis dienen Prinzipien und Praktiken des Lean Agile Leaderships, technische Agilität, Team Agilität und agile Produktentwicklung.

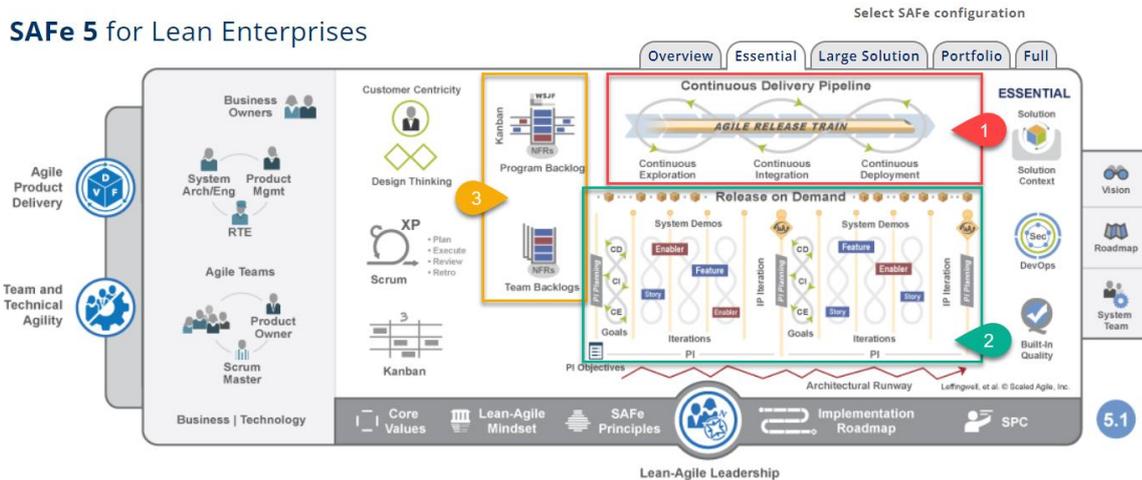


Abbildung 14 - Essential SAFe²³

Wie bereits erwähnt, bedient sich SAFe vieler bekannten agilen Methoden und vereint sie miteinander. In Abbildung 14 kann beispielsweise Kanban und Scrum gefunden werden, die als etablierte Methoden in den agilen Teams verwendet werden.

Der Agile Release Train (siehe Abbildung 14, rote Markierung [1]), auch ART genannt, ist das Herzstück in allen SAFe Konfigurationen und ermöglicht Business Agilität. Er umfasst typischerweise zwischen 50 – 125 Personen und vereint dabei unterschiedliche Funktionen wie Business, technische Teams, Key Stakeholders und andere Ressourcen zu einer dezidierten und wichtigen Solution Mission. Der Agile Release Train verfügt über eine fix definierte Kadenz, klar definierte Rollen, Pakete und Synchronisierungspunkte (siehe Abbildung 14, grüne Markierung [2]). Der Inhalt der jeweiligen Pakete wird über Backlogs definiert und priorisiert (siehe Abbildung 14, orange Markierung [3]). Eine detailliertere Beschreibung des Agile Release Trains, sowie ein Vergleich zum klassischen Ansatz wird in Kapitel 5 weiter ausgeführt.

²³ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

Large Solution SAFe configuration

Large Solution SAFe baut auf Essential SAFe auf und erweitert es um Rollen und Artefakte in der Spinning Palette (siehe Abbildung 15, orange Markierung [3]) und führt die Enterprise Solution Delivery (siehe Abbildung 15, grüne Markierung [2]) ein. Die Enterprise Solution Delivery unterstützt die Entwicklung der größten und komplexesten Produktlösungen, die die Anwendung von mehreren ARTs und die Einbindung von Zulieferern (siehe Abbildung 15, rote Markierung [1]) benötigen. Diese Konfiguration inkludiert allerdings noch keine Portfolio Betrachtungen, was typischerweise in der Luftfahrt-, Verteidigungs- oder Automobilindustrie, oder auch bei Behörden, der Fall ist.

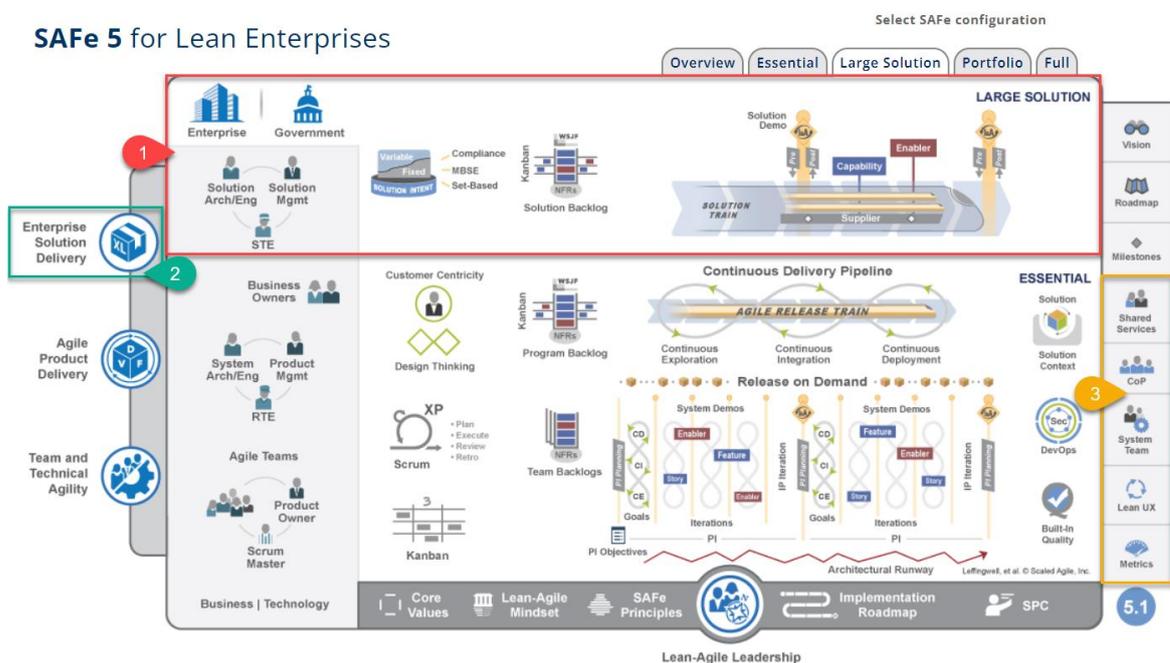


Abbildung 15 - Large Solution SAFe²⁴

Die Anwendung dieser Konfiguration wird näher für die Umsetzung in der Medizintechnik, unter Berücksichtigung von vorliegendem Entwicklungsprojekt, in Kapitel 5 geprüft.

²⁴ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

Portfolio SAFe configuration

Portfolio SAFe ist die Minimalkonfiguration für Unternehmen, die Business Agilität erreichen wollen. Portfolio SAFe baut auf Essential SAFe auf, fügt bekannte Rollen und Artefakte der Spinning Palette hinzu (siehe Abbildung 16, orange Markierung [3]) und erweitert es um Aspekte des Lean Portfolio Managements, organisatorische Agilität und um die Kultur des kontinuierlichen Lernens (siehe Abbildung 16, grüne Markierung [2]).

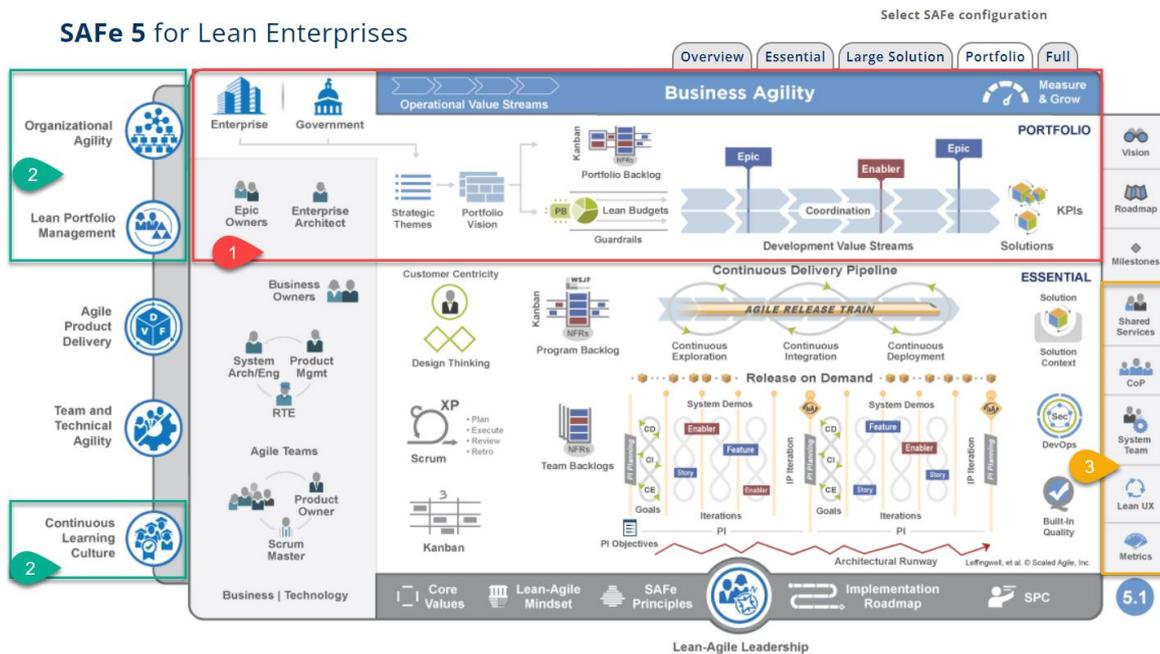


Abbildung 16 - Portfolio SAFe²⁵

Die Konfiguration vereint die Ausführungen des Portfolios und der unternehmerischen Strategie und versucht die Entwicklung an den Rhythmus von einem oder mehreren Wertschöpfungsketten anzupassen (siehe Abbildung 16, rote Markierung [1]).

Diese Konfiguration soll Unternehmen dabei unterstützen, dass die Wertschöpfung und die damit verbundenen Agile Release Trains sich unter optimalen Einsatz von Ressourcen auf die Ausführung ihrer Entwicklung konzentrieren können und dennoch die strategischen Ziele erreichen können.

²⁵ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

Full SAFe configuration

Full SAFe ist die umfassendste Konfiguration und ergänzt die Aspekte aus der Konfiguration Large Solution (siehe Abbildung 17, grüne Markierung [1]) und Portfolio Solution (siehe Abbildung 17, rote Markierung [2]). Die volle Konfiguration zielt auf die größten Unternehmen und Behörden ab, sodass es unter Umständen notwendig ist, dass mehrere Instanzen von SAFe Konfigurationen verwendet werden.

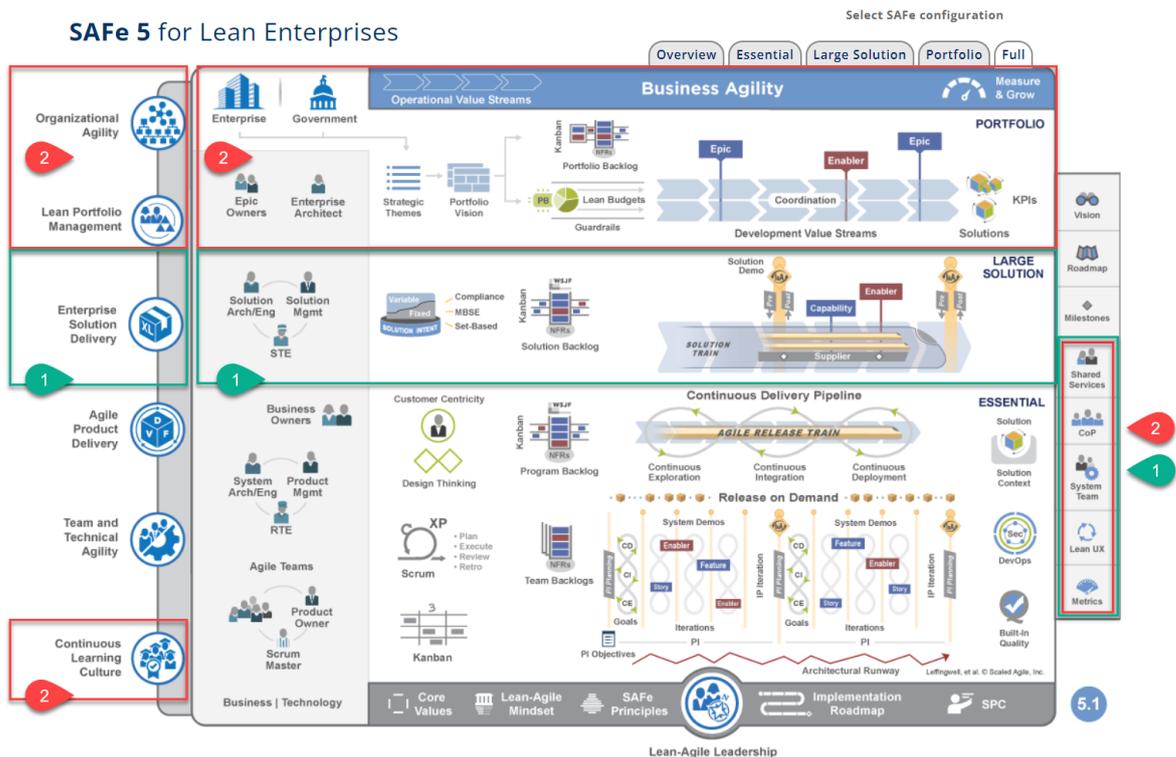


Abbildung 17 - Full SAFe²⁶

Jedes Unternehmen kann sich selbst dafür entscheiden, in welchem Detaillierungsgrad und welcher Konfiguration es SAFe implementieren möchte. Viel hängt von individuellen Faktoren ab, beispielsweise die Branche in dem das Unternehmen tätig ist, die Unternehmensgröße, aber auch die Fähigkeit des Unternehmens für einen bevorstehenden Wandel. Auch für größere Unternehmen könnte es eine Möglichkeit sein, zuerst mit einer kleinen Konfiguration, wie Essential SAFe zu starten und erst später den Aspekt der Business Agilität hinzufügt.

²⁶ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

3.3 Auswahl eines geeigneten Vorgehensmodells

Bei der Auswahl eines geeigneten Vorgehensmodells, müssen Unternehmen eine Vielzahl an Kriterien beachten, einige Beispiele dafür sind:

- Was ist meine Stellung am Markt
- Wie leicht können neue Konkurrenten in den Markt eindringen
- Wie schnell verändert sich der Markt und wie schnell muss ich darauf reagieren können
- Entwickle ich Massenprodukte oder Auftragsprodukte
- Wie setze ich meine vorhandenen Ressourcen am besten ein
- Welche Technologien setze ich ein und wie Erfahren ist mein Unternehmen darin
- Wie ist die Zusammenstellung meines Entwicklungsteams

Klassische Methoden sind seit vielen Jahrzehnten bewährt, haben große Erfolge verzeichnen können und werden auch immer wieder durch neue Methoden erweitert. In einem eingespielten Team, mit erfahrenen Technologien und Mitarbeitern können mit einem optimalen Ressourceneinsatz sehr gute Ergebnisse erzielt werden. Projektleiter haben bewährte Methoden, Tools und Kennzahlen zur Hand und können so das Projekt zum Erfolg führen. Typischerweise ist bei klassischen Projekten der Scope festgelegt und entsprechend in Lasten- und Pflichtenheft fixiert, die internen Entwicklungskosten und der Arbeitsaufwand für die Erledigung der Arbeitspakete basiert auf Abschätzungen und kann bei unvorhergesehenen Ergebnissen zu Abweichungen führen. Häufig gelten klassisch geführte Projekte deshalb als träge, kostenintensiv und mit einem hohen Risiko auf Projektverzögerungen. Zudem empfinden jüngere Mitarbeiter die darin verwendeten Methoden und Hierarchien als veraltet und drängen mehr zur Eigenverantwortung.

Agile Methoden sind bereits seit vielen Jahren im Einsatz und wurden in der Vergangenheit mehrheitlich in der Softwareentwicklung eingesetzt. Bei korrektem Einsatz bestechen sie durch eine hohe Flexibilität, was es Unternehmen ermöglicht schnell auf Marktveränderungen zu reagieren. Der Kunde steht im Zentrum aller Überlegungen und soll eine optimale Erfahrung mit dem Produkt erleben können, was agile Teams durch Einbinden des Kunden erzielen. Kleine Entwicklungsintervalle und konstantes Erstellen funktions-tüchtiger Software helfen, dass Kinderkrankheiten frühzeitig entdeckt und ausgebessert werden können. Bei agilen Entwicklungsprojekten sind typischerweise der Ressourceneinsatz und die Zeit bzw. Kadenz der Entwicklung klar fixiert, während der Scope eine veränderbare Größe ist. Agile Projekte könnten unter dem Motto: „Wir wissen noch nicht wie das Produkt im Detail aussieht, das wir liefern, wir garantieren aber, dass es das bestmögliche Produkt mit den vorhanden Ressourcen ist“ durchgeführt werden. Agile Methoden bieten neben vielen Vorteilen und Chancen aber auch einige Gefahren. Viel Flexibilität gegenüber dem Kunden kann aber auch dazu führen, dass der Kunden sehr oft die Meinung ändert und man bereits fertig gestellte Software wiederrum anpassen muss oder sich sehr oft im Kreis dreht. Agile Teams können aber auch dazu neigen, das Rad immer wieder neu erfinden zu wollen. Das kann vor allem dann passieren, wenn Mitarbeiter im Team oft wechseln und nicht auf die Erfahrungen der Vergangenheit aufbauen können.

Das Experimentieren mit Neuem kann auch dazu führen, dass sich Teammitglieder in Details verzetteln und dadurch ungenügenden Output liefern. Generell muss bei der Teamzusammenstellung von agilen Teams darauf geachtet werden, dass die darin arbeitenden Mitarbeiter mit der gebotenen Freiheit auch umgehen können und diese zu dem Allgemeinwohl nutzen. Gerade bei der Transformation zu agilen Teams, kann es vorkommen, dass es auch Mitarbeiter gibt, bei denen Widerstand entsteht, da sie gerne nach genauen Vorgaben arbeiten oder in der Vergangenheit anders gearbeitet haben und sich nicht auf die Veränderungen einstellen wollen. In solchen Fällen kann es zu großen sozialen Unstimmigkeiten innerhalb des Unternehmens kommen, wodurch auch motivierte Mitarbeiter während der Veränderung möglicherweise ihre Motivation verlieren. Eine weitere Gefahr ist, dass Unternehmen eine Art Scheinagilität einführen, bei welchen man bekannte Methoden, wie das V-Modell, einfach öfter ausführt.

Für viele Unternehmen sind die Charakteristiken ihrer Projekte nicht klar pro klassische Methoden und nicht klar pro agile Methoden. Für solche Fälle oder für eine sukzessive Einführung agiler Methoden eignet es sich sogenannte hybride Vorgehensmodelle zu verwenden. Ein möglicher Einsatz wäre, dass zu Beginn eines Projekts die Anforderungen nur grob erfasst werden und die weitere Ausarbeitung mit agilen Methoden erfolgt, zu einem späteren Zeitpunkt im Projekt aber wieder mit klassischen Methoden entwickelt und gesteuert wird. Ein weiterer Ansatz könnte es sein, dass Teile von Projekten agilen Methoden folgen, so kann beispielweise die Hardwareentwicklung nach klassischen Methoden erfolgen, während die Softwareentwicklung agile Methoden anwendet.²⁷

Alle klassischen und agilen Methoden haben Vor- und Nachteile und müssen im Unternehmenskontext evaluiert werden. Viele Nachteile und Risiken agiler Methoden sind unter Umständen kein Problem, wenn im Unternehmen geeignete Rahmenbedingungen vorliegen. Wichtig ist, dass sich die Unternehmen mit bekannten, wie auch mit neuen Methoden auseinandersetzen und sich dabei deren Vor- und Nachteile bewusst machen. Nachfolgend sollte eine gezielte Entscheidung für ein oder mehrere Methoden, unter Berücksichtigung der Rahmenbedingungen, getroffen werden. In größeren Unternehmen ist es zudem möglich, dass nicht alle Projekte denselben Methoden folgen können oder müssen. Eine Einführung agiler Methoden empfiehlt sich Schrittweise und nicht als Big Bang. Der Veränderungsprozess muss vom Unternehmen begleitet werden, sodass auf Ängste und Bedürfnisse der Mitarbeiter eingegangen werden kann, denn schlussendlich liefern nur motivierte und engagierte Mitarbeiter einzigartige und hochwertige Produkte.

²⁷ (Kuster, et al., 2019, S. 28)

4 Vorstellung des Entwicklungsprojekts

Im vorliegenden Kapitel wurden das Entwicklungsprojekt, dessen Charakteristika und das Projektumfeld beschrieben. In Kapitel 5 wird auf beschriebenes Projekt Bezug genommen und ausgewählte Methoden werden anhand des Projekts evaluiert.

Bei dem vorliegenden Projekt handelt es sich um ein beispielhaftes, aber repräsentatives Projekt einer Entwicklung in der Medizintechnik, welches Weiters in dieser Arbeit als Projekt Titan bezeichnet wird.

Am Ende des Projekts soll ein Produkt entstehen, das in Kombination mit anderen Medizintechnik Produkten verwendet werden kann. Zu den verbundenen Produkten gehören Software- aber auch Hardwarelösungen, die Kunden gemeinsam in einem Labor nutzen können. Bei den angebunden Produkten handelt es sich um eine Mischung aus existierenden Produkten, Produkten anderer Firmen und neu entwickelte Produkte, die parallel und in Abstimmung zu Projekt Titan entwickelt werden.

4.1 Projekt Charakteristika

Bei Projekt Titan handelt es sich um ein Forschungs- und Entwicklungsprojekt, das sich aus Software- und Hardwareentwicklung zusammensetzt. Bei einem Teil der eingesetzten Technologie wird während der Entwicklungsdauer noch Forschung betrieben, da es sich um eine Methode handelt, welche erstmals in einem Medizintechnik Produkt zum Einsatz kommen soll. Dabei muss die Frage beantwortet werden, ob Ergebnisse zuverlässig und korrekt erzeugt werden können und ob es negative Auswirkungen der Technologie auf Patientenproben gibt. Diese Fragen sollen in frühen Phasen der Produktentwicklung erforscht und mit verschiedenen Hardwareständen stetig verbessert werden.

Projektgröße und Projektdauer

Bei vorliegendem Entwicklungsprojekt handelt es sich um ein Projekt mit 200 Vollzeit Mitarbeitern, einem Projektbudget von ca. 60 Millionen Schweizer Franken und einer geplanten Entwicklungsdauer von sechs Jahren. Verwendet man eine Einteilung wie in Kapitel 2.1.1 Charakterisierung anhand der Projektgröße, so kann das Entwicklungsprojekt als Großprojekt bezeichnet werden.

Komplexität des Projekts

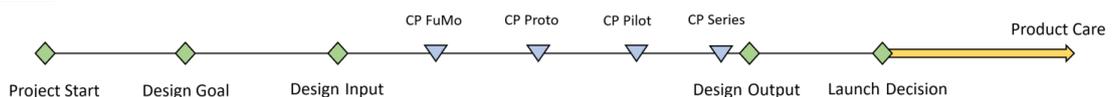
Die Komplexität des Projekts kann gemäß Stacey Matrix aus Kapitel 2.1.3 Charakterisierung anhand der Komplexitätsstufe als komplex eingeteilt werden. Teile der angewendeten Technologie wurden bisher in einem ähnlichen Kontext noch nicht verwendet und

müssen während des Projektzeitraums erforscht, stabilisiert und serienreif entwickelt werden. Die Anforderungen an das Produkt sind Großteils klar, allerdings handelt es sich bei Projekt Titan auch um ein Produkt, das neue Marktsegmente erschließen soll, in dem das Unternehmen bislang nur wenig vertreten war. Ebenfalls kann die Entwicklungsdauer von sechs Jahren dazu führen, dass anfangs klare Anforderungen sich ändern oder über die Zeit unklar werden. Das Unternehmen in dem Projekt Titan entwickelt wird, verwendet Großteils klassische Projekt Management Methoden. Vor allem auf höheren Managementebenen, aber auch in der Projektleitung finden klassische Methoden ihren Einsatz. Den Entwicklungsteams selbst wird zwar mittels Entwicklungsprozess vorgeschrieben, in welcher Phase welche Arbeitspakete bearbeitet werden müssen, allerdings sind die Methoden wie dies erzielt werden kann teils offen. Das bedeutet, dass vor allem in Abteilungen mit reiner Softwareentwicklung überwiegend agile Methoden zur täglichen Arbeitssteuerung verwendet werden.

4.2 Angewandter Entwicklungsprozess

In einem Projekt gibt es eine Vielzahl an Prozessen, viele davon beschreiben wie Teams oder Individuen miteinander agieren oder beispielsweise wie die Kommunikation innerhalb des Projekts stattfinden soll. In der Entwicklung von Medizintechnikprodukten ist aber der formale Prozess besonders wichtig, denn er beschreibt, wie ein Unternehmen vorgeht und reproduzierbar zuverlässige Ergebnisse erzielen möchte. Dieser Prozess wird für das gesamte Unternehmen oder für Teilbereiche definiert und muss entsprechend während der Projektdurchführung eingehalten werden. Die Prozesse und deren Einhaltung können durch unterschiedliche Behörden oder andere externe Partner jederzeit geprüft werden.

Checkpoints



Milestones

Abbildung 18 - Aufbau des Produkt Entwicklungsprozesses

Abbildung 18 stellt eine vereinfachte Darstellung des Entwicklungsprozesses des Unternehmens dar und zeigt insgesamt fünf Milestones. Vor einem Milestone liegt eine gleich heißende Projektphase, in denen hinterlegte Arbeitspakete bearbeitet und abgeschlossen werden müssen. Zu einem Milestone werden definierte Projektziele geprüft und entschieden, ob das Projekt in die nächste Phase übergehen kann.

Der Milestone Project Start ist der offizielle Start des Projektes, in dem die Vision des Produktes bereits präsent ist und das Projektteam gebildet wird.

In der Phase Design Goal werden alle Stakeholder Anforderungen gesammelt und das Projektteam legt die Architektur des Systems fest.

Die Phase Design Input startet nach erfolgreichem Abschluss des vorherigen Milestones, bestehende Stakeholder Anforderungen und Architektur dienen als Input für System Anforderungen und definieren das System lösungsneutral. Die Phase endet durch Fertigstellung der technischen Anforderungen und dem Review der dazu gehörenden Architektur mit dem erfolgreichen Abschluss im Milestone Meeting Design Input. Zu diesem Zeitpunkt werden auch die Zeitpunkte für den Milestone Design Output und Launch Decision festgelegt. Ab diesem Zeitpunkt darf, laut Unternehmensphilosophie, die Entwicklung des Produkts an Kunden kommuniziert und potenzielle Kunden dürfen beworben werden.

Die Phase zwischen dem Milestone Design Input und dem Milestone Design Output ist die längste Phase, in der Modul Anforderungen geschrieben werden, die Umsetzung der Anforderungen durch Implementierung von Software und Entwicklung von Hardware erfolgt und Tests auf verschiedenen Integrationsebenen des V-Modells durchgeführt werden. Diese Phase gilt als äußerst wichtig und wird wegen ihrer Phasenlänge in weitere Checkpoints unterteilt, in denen der Fortschritt und der Erfüllungsgrad der Arbeitspakete gemessen wird. Die Checkpoints heißen FuMo, Proto, Pilot, Series und sind nach den entsprechenden Hardware-Entwicklungsständen benannt. Sollten sich in der Phase zu einem Checkpoint technische Änderungen ergeben, so können diese bis zum nächsten Checkpoint angepasst bzw. überarbeitet werden. Ein Beispiel hierfür wäre, dass Langzeittests zeigen, dass ein ausgewähltes Material nicht robust genug ist, wodurch man sich für ein anderes robusteres Material entscheidet, oftmals mit einem höheren Preis verknüpft. Bis zum CP FuMo (Checkpoint Funktionsmodell) kann innerhalb des Projekts noch Grundlagenforschung betrieben werden, es wird mittels Machbarkeitsstudien geprüft, ob eine Technologie prinzipiell technisch realisierbar und marktfähig wäre.

CP Proto (Checkpoint Prototyp) prüft erstmals, ob das Design und die Anforderungen auch im Produktionsumfeld zu bestimmten Preisen gebaut werden können. Hierzu werden in Einzelproduktion Teile gefertigt und zusammengebaut, es wird geprüft, ob die Materialien den Anforderungen genügen. Ein weiterer Bestandteil des CP Proto ist, dass Dauertests und Zuverlässigkeitstests zeigen, dass das Design auch langfristig gemäß Erwartungen in erforderlicher Qualität arbeiten kann.

Zu CP Pilot (Checkpoint Pilot) müssen alle Tests erstmals abgeschlossen worden sein, damit man ein komplettes Bild über die Qualität des Produkts hat. Zu diesen Tests gehören neben funktionalen und nicht funktionalen Tests auch Kompatibilitätstests, Lagertests, Transporttests, Höhentests und Klimatests. Nach Abschluss der Tests müssen die Geräte noch immer in vollem Umfang funktionstüchtig sein. Ein weiterer Aspekt ist, dass zu diesem Zeitpunkt die Produktionslinien manuell oder automatisiert in der Lage sein müssen, das Produkt in größerem Maße zu fertigen.

Typischerweise, so der Qualitätsanspruch, sind mit Abschluss des Checkpoints Pilot nur noch wenige bis gar keine Änderungen mehr notwendig. Sollten Änderungen erfolgen, da ein Test zuvor fehlgeschlagen ist, so müssen die fehlgeschlagenen Tests erneut durchgeführt werden. Der CP Series (Checkpoint Series) hat somit ein verkaufsfähiges und produzierbares Produkt als Review Gegenstand. Zu diesem Zeitpunkt muss auch jegliche Dokumentation, wie Testberichte und Benutzer Dokumentation, abgeschlossen werden.

Nach erfolgreichem Abschluss des letzten Checkpoints wird der Erfolg und das Erreichen der kompletten Phase Design Output nochmals geprüft, bei positivem Verlauf gilt der Milestone Design Output als bestanden. Ab diesem Zeitpunkt kann die Produktion für den Verkauf der Produkte gestartet werden.

Größere Abweichungen am Produkt bzw. Anpassungen der Systemanforderungen sind während der Phase vom Milestone Design Input bis zum Milestone Design Output nicht mehr bzw. nur mit Zustimmung aller Stakeholder möglich. Ein Beispiel hierfür könnte sein, dass während der Software Implementierung festgestellt wird, dass eine Anbindung an eine externe Kundensoftware technisch nicht möglich ist, bedingt durch zuvor falsch getroffene Architekturentscheidungen sein. Diese Änderung würde bedeuten, dass der Kunde sein Produkt nicht wie erwartet verwenden kann oder, dass getroffene Entscheidungen überarbeitet werden müssen, was mehrheitlich mit großen Terminabweichungen verbunden ist. In jedem Fall kann eine solche Entscheidung nicht mehr durch das Projektteam allein getroffen werden, eine Einbindung aller Stakeholder ist notwendig.

Zu Milestone Launch Decision wurden alle Akzeptanztest durchgeführt und das Ergebnis geprüft. Treten schwerwiegende Probleme auf, so muss beurteilt werden, ob das Produkt dennoch in den Markt eingeführt werden soll, oder ob noch Anpassungen erfolgen müssen. Weiters werden zu diesem Zeitpunkt erste Kundeninstallationen durchgeführt, welche zur Evaluierung der Kundenzufriedenheit durchgeführt werden. Beim so genannten first customer monitoring können beispielsweise noch Probleme gefunden werden, die sich durch kundenspezifische Anbindungen ergeben. Hier kann durch kleinere Softwareanpassungen noch Kundenzufriedenheit für den globalen Launch des Produkts geschaffen werden.

Nachdem alle Milestones erfolgreich durchlaufen sind, kann das Produkt final am Markt eingeführt werden. Zu diesem Zeitpunkt beginnt die Product Care Phase für einen Hersteller eines Medizintechnik Produkts. In dieser Phase müssen die ausgelieferten Produkte am Markt gewartet und überwacht werden. Findet der Hersteller heraus, dass das Produkt nicht so funktioniert, wie es von ihm bei den Behörden eingereicht wurde, besteht sofortiger Handlungsbedarf. Beispiele hierfür können sehr unterschiedlich sein, von einer generellen Sicherheitslücke in einem verwendeten Tool oder Betriebssystem bis hin zu falschen oder ungenauen Messwerten. In allen Fällen müssen durch Experten die Auftretenswahrscheinlichkeit und der potenzielle Schaden beurteilt werden. Von dieser Beurteilung hängen auch die weiteren Schritte ab, die von einfachen Softwareupdates bis hin zum Entfernen des Produkts am Markt führen können. Produkte in der Medizintechnik werden typischerweise mindestens zehn Jahre vom Unternehmen verkauft, bevor sie durch neuere Produkte ersetzt werden könnten. Vertraglich sind Medizintechnik Hersteller aber meist verpflichtet, auch nicht mehr verkaufte Produkte weiterhin zu warten und entsprechende Verbrauchsmaterialien zu produzieren und zu verkaufen, damit Kunden die Produkte nochmals weitere Jahre verwenden können.

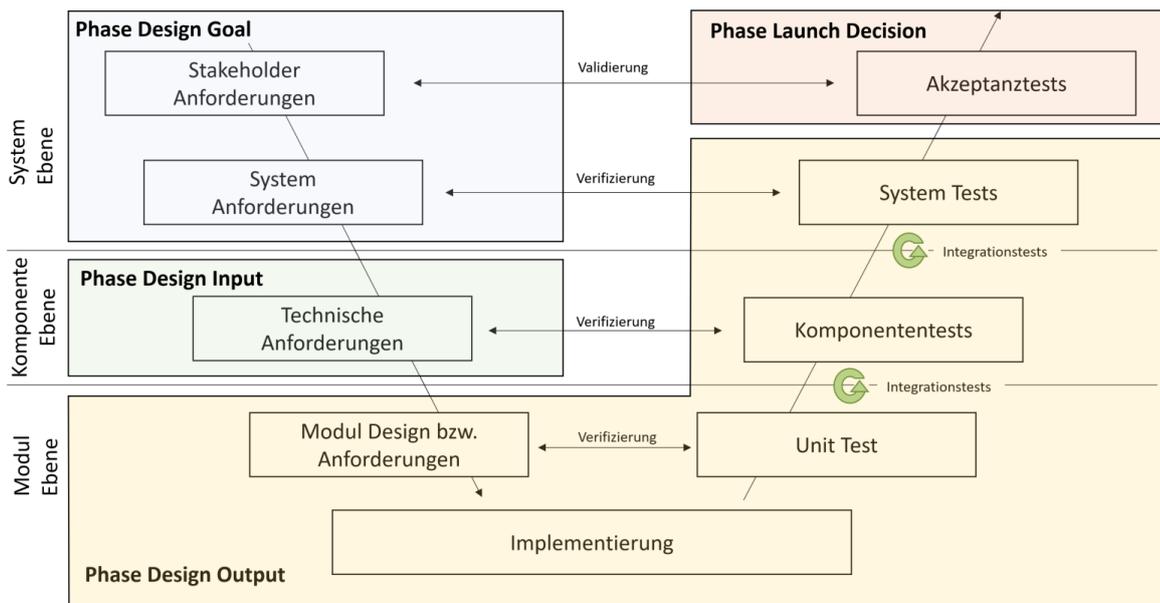


Abbildung 19 - V-Modell in Kombination mit den Projektphasen

Zusätzlich zum Entwicklungsprozess kommt das V-Modell zum Einsatz. Abbildung 19 zeigt, wie sich der Entwicklungsprozess und das V-Modell überschneiden. Jedes Level des V-Modells enthält eine entsprechende Verifizierung oder Validierung, die die erfolgreiche Umsetzung der Entwicklung im Vergleich zu den Anforderungen sicherstellt.

In Abbildung 19 ist ebenfalls ersichtlich, dass die Phase Design Output sich über mehrere Integrations- und Test Level erstreckt. Unit Tests, Komponententests und System Tests sollten zudem für jeden Hardwareentwicklungsstand durchgeführt werden, damit sichergestellt werden kann, dass das Produkt auch am Ende des Projekts funktionstüchtig ist und auch zusammengesetzt seine Zweckbestimmung erfüllt.

4.3 Projektaufbau

Das Projekt wird über drei Standorte entwickelt. In Deutschland findet die Entwicklung von drei Hardwarekomponenten statt, in Spanien findet reine Softwareentwicklung einer Komponente statt, in der Schweiz findet die Entwicklung einer weiteren Hardwarekomponente statt. Zusätzlich ist am Schweizer Standort, mit dem System Level, auch die Gesamtverantwortung der jeweiligen Disziplinen angesiedelt. Gesteuert werden alle Komponentenprojekte, sowie das System Level durch einen Programmleiter, der ebenfalls für die Kommunikation nach außen verantwortlich ist. Alle Hardwarekomponenten verstehen sich inklusive entsprechender Steuerungssoftware.

Das Projekt Titan muss ebenfalls unternehmensintern mit anderen Projekten und Programmen abstimmt werden, da es technische Abhängigkeiten zwischen den Produkten gibt. Nur wenn alle Produkte gemeinsam am Markt eingeführt und miteinander funktionstüchtig sind, können die Produkte gemeinsam am Markt erfolgreich sein.

Das Unternehmen ist in einer Matrix Struktur, wie in Kapitel 2.2.2 beschrieben, aufgebaut. Abbildung 20 zeigt die virtuelle Projektstruktur, alle Mitarbeiter sind von der Linien Struktur in die Projekt Struktur verliehen.

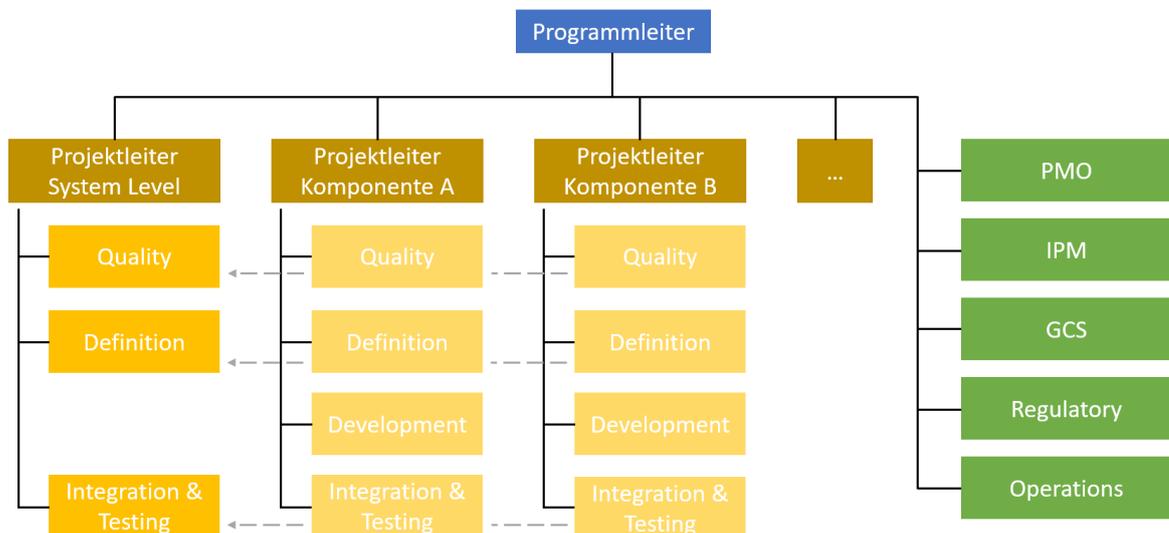


Abbildung 20 - Darstellung der Projektstruktur

Das Projekt wird übergeordnet durch einen Programmleiter geführt. Die braunen Kästen zeigen die Projektleiter der insgesamt fünf Komponenten Projektleiter und den Projektleiter der System Ebene. Der Projektleiter der System Ebene hat drei Disziplinen angehängt, Quality, Definition und Integration & Testing. Diese drei Disziplinen auf System Level (dunkelgelbe Kästen) sind Gesamtverantwortlich für ihren jeweiligen Bereich, was bedeutet, dass die entsprechenden Komponenten Disziplinen (hellgelbe Kästen) einerseits zu ihrem Projektleiter und andererseits zur Disziplin auf System Level rapportieren. Die Komponenten Projekte haben zusätzlich zu den drei erwähnten Disziplinen noch die Disziplin Development, in der die eigentliche Hardware- und Softwareentwicklung stattfindet.

Ebenfalls an den Programmleiter rapportierend (grüne Kästen) sind PMO (Project Management Office), IPM (International Product Manager), GCS (Global Customer Support), Regulatory und Operations. In der Disziplin des PMO, werden unterstützende Tätigkeiten durchgeführt, beispielsweise das Projektcontrolling, Wartung von Projektplänen oder das Verwalten von Projektrisiken. IPMs und GCS sind unternehmensinterne Kundenvertreter, die Auskunft über Marktentwicklungen und Kundenbedürfnisse geben und dabei stetig versuchen, dass erfasste Anforderungen (als Bestandteil der Stakeholder Anforderungen) entsprechend der Erwartungen umgesetzt werden. Während die IPMs sich mehr auf Business Aspekte, wie der Verkaufsfähigkeit des Produkts konzentrieren, vertritt GCS Aspekte wie Wartbarkeit des Produkts und Verwendbarkeit bzw. Benutzerfreundlichkeit. Regulatory sorgt für die Einhaltung der zulassungsrelevanten Anforderungen. Operations überwacht während der Entwicklung, dass das Produkt zu einem späteren Zeitpunkt innerhalb der geforderten Produktionskosten erfolgreich produziert werden kann.

5 Anwendung von SAFe

Wie bereits beschrieben handelt es sich bei SAFe um ein skalierbares agiles Framework, welches für kleinere Unternehmen oder Teams, wie auch für Großkonzerne oder Großprojekte Anwendung findet. Das vorliegende Kapitel erläutert aus diesem Grund nur Teilaspekte von SAFe und vergleicht diese mit existierenden und im vorliegenden Projekt verwendete klassische Methoden.

Die Vielfältigkeit der Konfigurationen erlaubt es Unternehmen entsprechend ihrer Bedürfnisse zu skalieren, bei der Anwendung von SAFe anhand des vorliegenden Entwicklungsprojektes, wird die Large Solution Konfiguration betrachtet.

Alle Ausführungen zu SAFe beruhen auf den Ausführungen von Dean Leffingwell et. al. in seiner Publikation²⁸ und der Plattform von Scaled Agile Inc.²⁹.

Generell beinhaltet SAFe unterschiedliche Rollen als klassische Entwicklungsmethoden, was zu Widerständen zwischen den beteiligten Personen führen kann. Auf die Unterschiede der Rollen und die damit verbundenen Konflikte wird innerhalb dieser Arbeit nicht weiters eingegangen und bedarf bei Einführung von SAFe genauerer Betrachtung und Begleitung von geschultem Personals.

²⁸ (Leffingwell & Knaster, SAFe distilled; Achieving Business Agility with the Scaled Agile Framework, 2020)

²⁹ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

5.1 Projektplanung und Organisation

Die Planung von Arbeitspaketen und Abläufen innerhalb eines Projekts, sowie eine stetige Kommunikation sind die Grundlage für den Projekterfolg. Sie dienen als Basis für jegliche Zusammenarbeit und klären Verantwortlichkeiten, Kommunikationswege, Eskalationswege von Teams oder Individuen und sollen bei Einhaltung dem Projekt Stabilität verleihen.

Wie bereits in Kapitel 2 ausgeführt, gibt es eine Vielzahl an Gründen für ein Scheitern von Projekten. Abbildung 21 zeigt eine Sammlung von Problemstellungen, die die bisherige Projektplanung und Organisation mit sich bringen.

Nr.	Problemstellung	Auswirkung
1	Fehlende Kontrollbefugnisse der Disziplinen	<ul style="list-style-type: none"> • Qualitätsprobleme werden erst spät sichtbar • Anpassungen des Zielvorhabens werden nicht kommuniziert
2	Entwicklungsteams arbeiten in unterschiedlichen Projektteams	<ul style="list-style-type: none"> • Unterschiedliche Herangehensweisen führen zu unterschiedlicher Qualität • Zielkonflikt zwischen Projekt und Disziplin
3	Erschwerte Zusammenarbeit und mangelnder Informationsfluss, da Projektteams über mehrere Länder verteilt sind	<ul style="list-style-type: none"> • Ungenaue oder sich ändernde Informationen führen zu falschen Implementierungen • Reduzierter informeller Informationsfluss mangels fehlendem Vertrauensverhältnisses • Änderungen werden oft nur in kleinen Gremien besprochen und nur sporadisch an alle verteilt
4	Langfristiger Planungshorizont	<ul style="list-style-type: none"> • Stetige Anpassungen der Projektplanung • Aufkommende Schwierigkeiten haben Einfluss auf Zeit und Budget
5	Starre Einbindung externer Zulieferer	<ul style="list-style-type: none"> • Zulieferer können schwer oder gar nicht auf Anpassungen reagieren
6	Fehlendes Kundenverständnis	<ul style="list-style-type: none"> • Suboptimale Umsetzung von lösungsoffenen Anforderungen

Abbildung 21 - Problemstellungen der Projektplanung und Organisation in Projekt Titan

5.1.1 Bisheriger Ansatz - unter Anwendung klassischer Methoden

Der angewendete Entwicklungsprozess wurde bereits in Kapitel 4.2 beschrieben. In diesem sind bereits Arbeitspakete für den gesamten Projektablauf definiert und unterscheiden sich von Projekt zu Projekt nur in ihrem Umfang. Jedes Arbeitspaket ist somit schon zu Projektstart definiert und wird mittels work breakdown structure weiter verfeinert. Die Aufwände für alle Arbeitspakete werden von den Projektteam abgeschätzt. Der dazugehörige Planungshorizont erstreckt sich von Project Start bis Design Goal und von Design Goal bis zu Design Input. Zum Zeitpunkt Design Input muss das Projektteam ein Datum für die Markteinführung des Produktes festlegen. Die Schwierigkeit hierbei ist, dass zwischen den Meilensteinen Design Input und Design Output einerseits noch viele Design Entscheidungen getroffen, Design Spezifikationen geschrieben und jegliche Hard- und Software entwickelt und getestet werden muss. Innerhalb der verschiedenen Disziplinen, sowie im Projektmanagement werden deshalb Pufferzeiten nach Erfahrungswerten aufgeschlagen, sodass kleinere Abweichungen abgefangen werden können. Bei unvorhergesehenen größeren Problemen führt es allerdings zu einem Zielkonflikt des magischen Dreiecks und es müssen Scope, Budget und Timelines neu verhandelt werden. In vergleichbar großen Projekten ist es deshalb in späteren Projektphasen meist zum Engagement von externen Firmen gekommen, damit Scope und Timeline gehalten werden können, was in einem erheblichen Mehraufwand des Budgets resultierte.

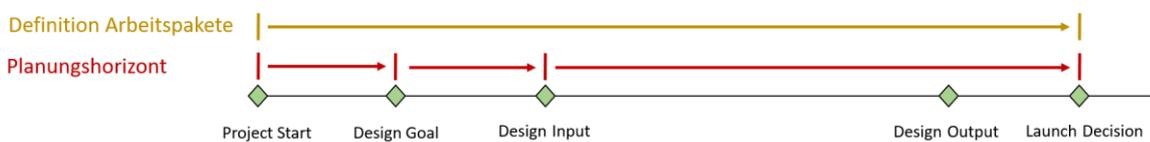


Abbildung 22 - Zeitpunkt der Definition von Arbeitspaketen und deren Planungshorizont

Wie in Kapitel 4.3 in Abbildung 20 aufgezeigt, wird jede Komponente, sowie das System Level als eigenes Projekt geführt, womit es neben dem System Projekt noch die Projekte Software, Buffer und Handover gibt. Die Komponenten Transport und Preparation wurden an externe Zulieferer abgegeben, allerdings werden diese dennoch intern von kleinen Projektteams betreut. Die dort verankerten Tätigkeiten dienen einerseits der Kommunikation und andererseits der Qualitätskontrolle der gelieferten Dokumente und Hardware. Das Projekt Titan kann nur in Kombination mit anderen System des gleichen Unternehmens betrieben und verkauft werden. Diese Systeme können unabhängig vom Produkt Titan verkauft werden und heißen Middleware, Pre Analytics und Post Analytics. Obwohl diese Systeme eigenständige Systeme sind, benötigt es für das Projekt Titan weitergehende Anpassungen, sodass diese miteinander in einer Solution betrieben werden können. Trotz dieser Abhängigkeit werden diese Systeme in eigenständigen Projekten entwickelt und unterliegen nicht der Reporting Line des Programms Titan.

Innerhalb des Projekts Titan wurde eine Integrationshäufigkeit von drei Monaten festgelegt. Die Projekte können zu diesem Zeitpunkt Software und, oder Hardware zur Integration an das System Level liefern. Zu Beginn der Entwicklungsphase, nach Design Input, ist

zu sehen, dass es erst verzögert zu einer ersten Hard- und Softwarelieferung gekommen ist, die anschließend integriert werden konnte.

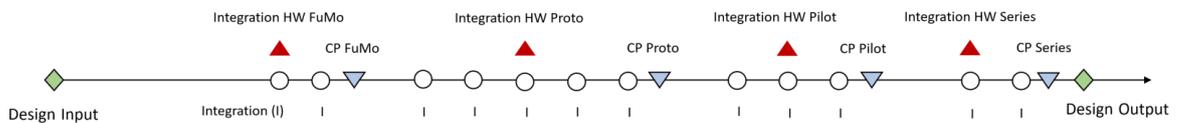


Abbildung 23 - Darstellung Integrationshäufigkeit

Die Inhalte der jeweiligen Lieferungen werden nach Möglichkeit zwischen dem System Projekt und den Komponenten Projekten abgestimmt, kommt es innerhalb eines Projekts zu Abhängigkeiten oder zu einem Ressourcenkonflikten, so bestimmen die Komponenten Projekte oftmals selbst, zu welchem Zeitpunkt ein Inhalt integriert wird.

Alle Anforderungen für das Projekt müssen bereits zum Meilenstein Design Input erstellt sein. In der Phase nach Meilenstein Design Input werden nur noch Modul Architekturentscheidungen getroffen und Design Spezifikationen für die Module der Komponenten geschrieben. Müssen Anforderungen nachträglich geändert werden, so muss das gesamte Projektteam und deren Stakeholder, in einem so genannten Change Request Board, den Änderungen zustimmen. Änderungsbedarf kann einerseits projektintern entstehen, indem neue Informationen der Entwicklung eine Anpassung der Anforderungen benötigen, oder von außerhalb durch Stakeholder an das Projekt herangetragen werden. In beiden Fällen werden diese Änderungen in dem Meeting durch den Requirements Manager aufgebracht und alle Anwesenden definieren zusammen die Auswirkungen und die Tragweite der Änderung. Wird einer Änderung zugestimmt, so müssen alle Personen, die diese Information benötigen informiert werden. Die Komponenten Teams werden durch ihre jeweiligen Projektleiter informiert, die ebenfalls am Change Request Board teilnehmen.

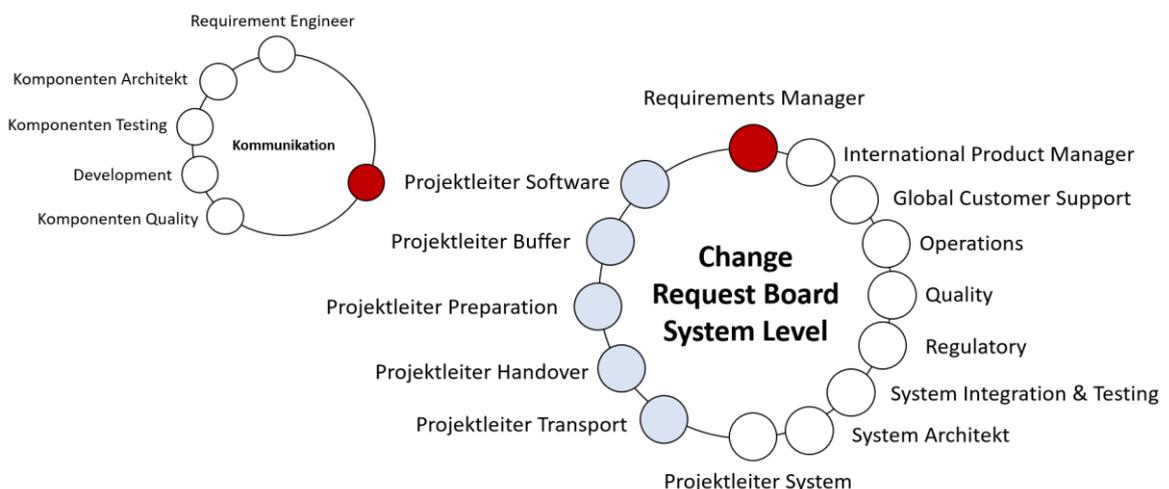


Abbildung 24 – Teilnehmer am Change Request Board System Level

Zusätzlich werden Entscheidungen in eigenen Dokumenten festgehalten und werden von allen Beteiligten unterzeichnet.

Das Beispiel des Change Request Boards in Abbildung 24 veranschaulicht den Aufwand der betrieben werden muss, um Änderungen zuzustimmen und relevante Personen nachträglich zu informieren. Zusätzlich zum Change Request Board, gibt es noch Architektur Boards und Change Request Boards auf Komponenten Ebene.

Weiters gibt es für jede Disziplin wöchentliche Besprechungen, in denen sich Manager der Disziplinen projektübergreifend abgleichen können. Jedes Projekt wird durch einen Repräsentanten vertreten, dieser muss neues Wissen wiederum innerhalb seines Projekts an relevante Personen verteilen. Beispiele für Meetings der Disziplinen sind: Integration & Testing, Definition, Quality und Development. Wöchentliche Besprechungen dienen einerseits der Fortschrittsmessung und andererseits dem Abgleich von technischen Problemstellungen. Normalerweise dienen diese Besprechungen nicht dem Abgleich von Methoden oder zur Definition von harmonisierten Arbeitsweisen. Wird während dieser Besprechungen ein Zielkonflikt zwischen den Komponenten und dem System Level identifiziert, so müssen diese Themen entsprechend zur Projektleitung eskaliert und in extremen Fällen über die Programmleitung gelöst werden.

Die Projektteams arbeiten vor allem mit Anforderungen und Dokumentationen, die bereits vor dem Meilenstein Design Input erstellt worden sind. Zu diesem Zeitpunkt sind einzelne Disziplinen bereits vertreten und beteiligen sich am Review, allerdings sind die Projektteams noch nicht voll besetzt, da in einer so frühen Phase beispielsweise noch keine Entwicklung oder entsprechende Tests stattfinden. Ihre Aufgabe ist es dennoch, diese Konzepte, Anforderungen und Architekturentscheidungen weiter zu verfeinern und entsprechend in Hardware und Software umzusetzen. Treten Unklarheiten oder Unstimmigkeiten auf, so sollten diese in entsprechenden wöchentlichen Besprechungen oder anderen technischen Boards aufgebracht werden.

5.1.2 Anwendung von SAFe

In SAFe werden wertschöpfende Tätigkeiten innerhalb eines oder mehreren Agile Release Trains (ART) erarbeitet. Alle Tätigkeiten werden auf den Kundennutzen ausgerichtet, sodass der Agile Release Train eine der wichtigsten Rollen im gesamten Framework hat.

Die Lieferungen des ART werden mittels Program Increments (PI) geplant und begrenzen die zu erbringende Leistungen auf die maximal verfügbare Kapazität des Trains. Der Zeitraum des PI ist immer gleich lange und ist für alle involvierten ARTs gleich terminiert.

Das Program Increment ist ein Zeitraum in dem die Planung, Entwicklung und Validierung einer oder mehrerer Funktionen erfolgt. Anhand einer Demo kann schnelles Feedback der Stakeholder gegeben werden und zeigt den Mehrwert der abgeschlossenen Periode für den Kunden. Die jeweiligen Program Increments benötigen sorgfältige Vorbereitungen und werden anschließend in einem Planungsmeeting mit den Teams abgestimmt und weiter verfeinert.

Solution Level

Der Solution Train umfasst mehrere Agile Release Trains und kann auch externe Zulieferer als eigene Trains einbinden. Abbildung 25 visualisiert die empfohlene Aufteilung der existierenden Projekte in verschiedene ARTs. In blau dargestellt sind die Projekte Software, Buffer und Handover angesiedelt. Diese Projekte werden innerhalb des Unternehmens entwickelt, umfassen zusammen etwa 100 Personen und sind über mehrere Standorte verteilt. Die gelben ARTs werden durch verschiedene externe Zulieferer abgedeckt und befinden sich deshalb in getrennten ARTs. In Grün dargestellt, sieht man jene ARTs, die innerhalb des Unternehmens entwickelt werden, aber ein Abhängigkeitsverhältnis seitens Projekt Titan zu diesen Projekten besteht. Die dort erstellten Produkte können allerdings autonom am Markt verkauft werden, aus diesem Grund sollten die Projekte in unterschiedliche ARTs aufgeteilt werden. Das System Projekt wird aufgelöst und ist wie die Rollen PMO, IPM, GCS, Regulatory und Operations integraler Bestandteil von SAFe.

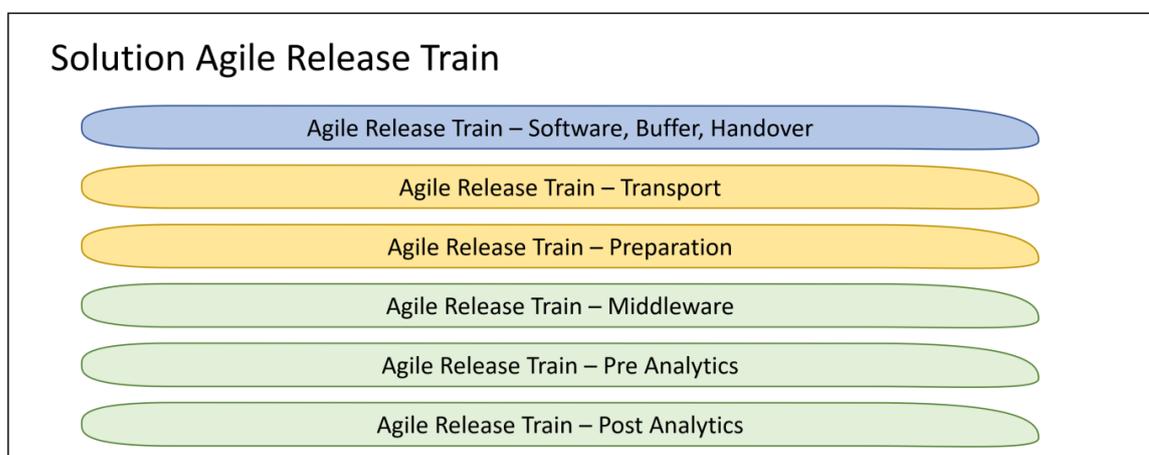


Abbildung 25 - Implementierung des Solution Agile Release Train

Das Ziel aller ARTs auf Solution Level ist es, dem Kunden ein Produkt zu erstellen, das zusammen funktionstüchtig ist und ihm einen Mehrwert bietet. Um das zu gewährleisten, bedarf es Abstimmungen in Planung und Synchronisierung während der Durchführung.

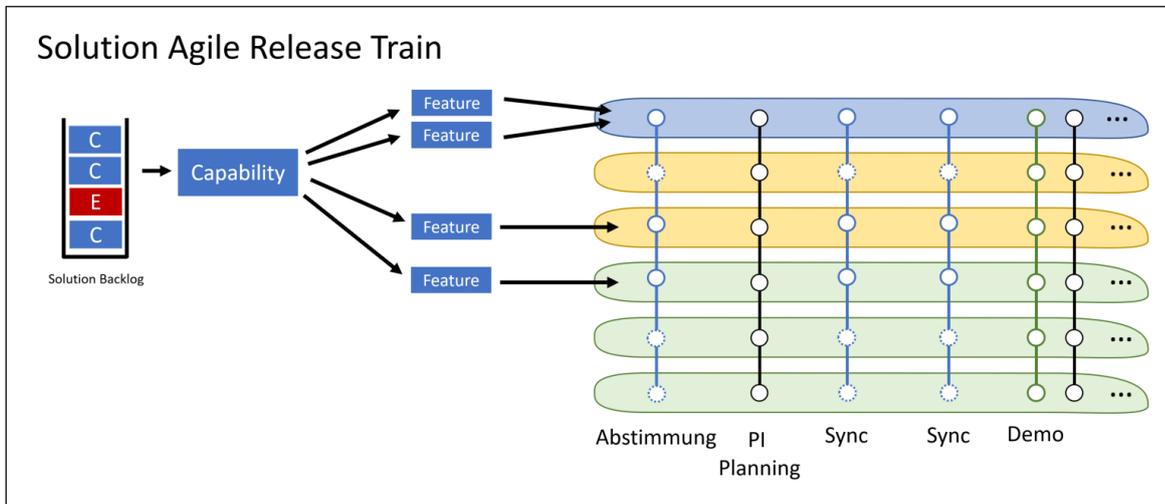


Abbildung 26 - Abstimmungen des Solution Agile Release Train, angelehnt an³⁰

Wie in Abbildung 26 zu sehen, umfasst das Solution Backlog Capabilities (C) und Enablers (E). Ein Program Increment umfasst normalerweise eine Vielzahl an Capabilities, die während der Planungsphase abgestimmt und in Features heruntergebrochen werden. Features können alle ARTs umfassen, müssen dies aber nicht, wie in Abbildung 26 zu sehen. Initiale Abstimmungen zu den Capabilities und Features finden bereits vor dem PI Planning statt und involvieren alle benötigten ARTs, inklusive Trains von externen Zulieferern. Abstimmungen zwischen ARTs erfolgen typischerweise mittels Product Managern, System Architekten und dem System Team. Während des Program Increments bedarf es oftmals auch weitere Synchronisierungen zwischen den Teams, um Abhängigkeiten zu managen und verbleibende oder neu auftauchende Unklarheiten zu managen.

Am Ende eines Program Increments findet eine Solution Demo statt, in der eine integrierte Lösung aller ARTs den Kunden und Stakeholdern präsentiert wird.

ART Level

Ein Agile Release Train (ART) besteht meistens aus 50 – 125 Personen und vereint verschiedene Disziplinen miteinander, die es zum Ziel haben dem Kunden eine gemeinsame wertschöpfende Solution liefern zu können. Wie in Abbildung 27 zu sehen erstellt der Agile Release Train kontinuierlich und inkrementell Lieferungen, wobei die Prozessschritte Definieren, Implementieren, Testen und Freigabe wiederholt werden.

³⁰ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

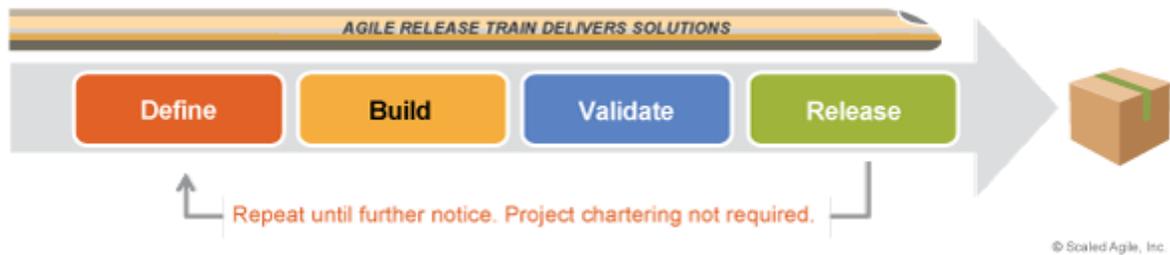


Abbildung 27 - Kontinuierlicher Prozess eines Agile Release Trains³¹

Program Increment (PI) Planning ist die Zusammenkunft aller, in einem Agile Release Train, beteiligter Personen, um ein einheitliches Verständnis des Program Increment zu bekommen. Typischerweise dauert das PI Planning zwei Tage, findet alle zwölf Wochen statt und versammelt alle Beteiligten eines Agile Release Trains in einem Raum.

Einleitend präsentiert ein Business Owner den Status quo der Kundenprobleme und teilt mit den Teilnehmern die Vision einer funktionierenden Solution, die dieses Problem lösen soll. Das Product Management zeigt den Teams auf, wie die ausarbeiteten Features dazu beitragen sollen, die Vision umzusetzen. System Architekten stellen die damit verbundene Vision der Architektur vor. Sollte sich das Entwicklungsframework verändert haben oder es wurde mittels Inspect & Adapt festgestellt, dass es Verbesserungsbedarf gibt, so können entsprechende Development Vertreter die Neuerungen ebenfalls präsentieren. Der Release Train Engineer präsentiert die bisherigen Planungsfortschritte und teilt die Erwartungen der anschließenden Detailplanung mit.

Der Hauptbestandteil des PI Plannings ist, dass jedes Agile Team in Breakout Sessions die Kapazität des Teams festlegt und sich überlegt, welche Backlog Items benötigt werden, um die zuvor präsentierten Features erfüllen zu können. Alle Teams erstellen vorläufige Pläne, die allen Teilnehmern des Plannings offen liegen. Während diesem Prozess definieren die Teams ebenfalls, welche Risiken und Abhängigkeiten zu anderen Teams sich ergeben. Die erarbeiteten Pläne stellt jedes Team den anwesenden Business Owners, dem Product Management, den anderen Teams und anderen Stakeholdern vor. Diese haben die Möglichkeit Fragen zu stellen oder Unklarheiten zu beseitigen.

Während der Reviews kann es dazu kommen, dass die vorgelegten Pläne Herausforderungen beinhalten, die von den Teams nicht selbst gelöst werden können. In solchen Fällen überprüft das Management die vorhandenen Möglichkeiten und entscheidet, wie die geplanten Herausforderungen bewältigt werden können.

Durch die Vielzahl an parallellaufenden ARTs benötigt es nach dem ersten Tag des PI Plannings noch eine weitere Abstimmung zwischen den Release Train Engineers. Dort werden Abhängigkeiten zwischen den ARTs diskutiert und aufgelöst.

³¹ (Leffingwell, Scaled Agile Framework, aufgerufen am 16.10.2021)

Am zweiten Tag des PI Plannings wird die Planung der Teams weitergeführt und die bestehende Planung, anhand erfolgter Diskussionen und Entscheidungen, weiter verfeinert. Die finalen Pläne werden in einem gleichen Setup wie am ersten Tag präsentiert, das Team zeigt verbleibende Risiken und Abhängigkeiten auf. Alle gesammelten Risiken beider Tage werden zudem darauf geprüft, ob sie zwischenzeitlich gelöst wurden, jemand das Risiko weiterverfolgen und im Nachgang lösen muss, das Risiko akzeptiert wird oder ob ein Team das Risiko abfedern kann.

Am Ende der Planung beurteilen alle Anwesenden, wie zuversichtlich sie sind, dass die definierten Ziele, unter Berücksichtigung der Risiken, erreicht werden können. Fällt die Abstimmung negativ aus, so bedarf es weiterer Planung und Erfassung zusätzlicher Risiken. Das PI Planning schliesst mit einer Retrospektive und reflektiert somit, was in Zukunft verbessert werden sollte.

Team Level

Ein Agiles Team ist eine Gruppe von fünf bis elf Personen, die während einer Iteration Themen Definieren, Implementieren und Testen. Die Teammitglieder haben unterschiedliche Fähigkeiten und erzeugen den Mehrwert eines Agile Release Trains. Alle Teams werden durch einen Product Owner und einen Scrum Master unterstützt. Wo sinnvoll, werden Teams mit Personen aus Hardware, Mechatronik und Software kombiniert gebildet. Ansonsten werden im Projekt Titan die Teams aufgeteilt in Team Software und Team Hardware und Mechatronik.

Eine Iteration ist ein Zeitraum von ein bis vier Wochen, in dem ein Team des Agile Release Trains inkrementell Mehrwert generiert, in dem es User Stories implementiert und diese anschließend erfolgreich testet. Im Projekt Titan wurde eine Iteration auf zwei Wochen festgelegt. In anderen agile Methoden wird diese Zeitspanne auch als Sprint bezeichnet.

Die Iteration Planning dient jeden agilen Team zur Planung der bevorstehenden Iteration. Der Product Owner definiert, in Abstimmung mit dem Commitment des Program Increments, vorläufige Iteration Goals. Die Iteration Goals definieren eine High Level Zusammenfassung der geplanten Business- oder technischen Ziele. Nach Außen dienen sie der Transparenz und der Abstimmung mit anderen Teams. Das Iteration Planning beginnt damit, dass der Product Owner und das Team die Ziele gemeinsam prüfen und zusammen mit prioritären Team Backlog Items, sowie weiteren Verbesserungsbedarf, bespricht. Der Product Owner definiert dabei was erledigt werden muss, das Team bespricht wieviel und die Art wie etwas erledigt werden kann. Dabei werden meist Optionen zur Implementierung, technische Herausforderungen und Abhängigkeiten besprochen. Zudem wird besprochen, wie die Ziele in Einklang mit den Non Functional Requirements gebracht werden können. Die Planung endet mit einem Plan für die kommende Iteration und dem Versprechen des Teams, die nötige Arbeit zu leisten, damit definierte Ziele erreicht werden können.

5.1.3 Vorteile / Nachteile der Anwendung von SAFe

Der Vorteil klassischer Herangehensweisen ist, dass bei simplen Projekten, mittels work break down structure und vorgefertigten Arbeitspaketen, das Projekt bereits bei Projektstart genau in Projektumfang, Budget und Zeit geplant werden kann. Zudem haben standardisierte Verfahren den Vorteil, dass man ein kontrolliertes Ergebnis erwarten kann. Der Nachteil klassischer Herangehensweisen ist, dass bei komplexen Sachverhalten, sowie bei technischen oder technologischen Herausforderungen, unvorhergesehene Kosten- oder Zeitexplosionen entstehen. Verändern sich der Markt oder die Kundenbedürfnisse während der Entwicklung, so müssen bereits fertig gestellte Arbeitspakete aufwändig überarbeitet werden.

Die bisherige Herangehensweise in Projekt Titan hat unter anderem das Problem, dass die Verteilung auf mehrere Unternehmen und mehrere Standorte in Kombination mit einer großen Anzahl an Projektmitarbeitern zu einem hohen Aufwand in der Abstimmung führt und häufig Informationen nicht alle Personen erreichen, die diese benötigen würden. Viele Prozesse und Anpassungen brauchen zudem große Entscheidungsgremien, die potenziell Einfluss auf das Zieldreieck des Projekts haben könnten.

Bei der Anwendung von SAFe werden Arbeitspakete um einen Mehrwert für den Kunden definiert. Arbeitspakete werden in entsprechenden Backlogs gesammelt und nach Priorität dem nächsten Program Increment zugewiesen. Änderungen in den Anforderungen bedeuten eine Überarbeitung des Program Increments und dies passiert typischerweise bevor das Backlog Item zur Implementierung freigegeben wird. Während dem PI Planning werden allen beteiligten Personen des Agile Release Trains neue Funktionen vorgestellt und der Kontext wird durch unterschiedliche Funktionen erörtert. Die Agilen Teams überlegen während der Planung, welche Abhängigkeiten sie zu anderen Teams haben und welche Aufgaben sie durchführen müssen damit sie die Ziele erreichen können. Am Ende des PI Plannings steht ein Plan für die kommenden drei Monate, der für alle Parteien verständlich und plausibel klingt. Während der Durchführung müssen bei Bedarf nur noch kleinere Abstimmungen zwischen den Teams erfolgen.

Aktuell werden andere Systeme des Unternehmens benötigt, die Abstimmung erfolgt aber nicht auf technischer Ebene, sondern erfolgt auf Ebene des Programleiters. Gegenseitige Anforderungen werden getauscht und werden ohne weitere Abstimmung zu einem freiwählbaren Zeitpunkt implementiert. Die Konsequenz daraus ist, dass oftmals Implementierungen innerhalb des Projekts Titan erfolgen, aber noch auf die Implementierung der anderen Systeme gewartet werden muss, um getestet werden zu können.

SAFe bindet alle benötigten Systeme, externe Zulieferer und internen Projekte in verschiedene Agile Release Trains. Der sich daraus ergebende Vorteil ist, dass alle Ziele für die kommende Periode aufeinander abgestimmt werden, sodass ein einheitliches Ziel verfolgt werden kann.

Nr.	Problemstellung	Lösungsansätze mit SAFe
1	Fehlende Kontrollbefugnisse der Disziplinen	<ul style="list-style-type: none"> • Teams bekommen mittels PI Planning ein einheitliches Verständnis • Teams definieren ihre Arbeit selbst und bestätigen, dass sie die Ziele erreichen können • Präsentation der Ergebnisse mittels Solution Demo
2	Entwicklungsteams arbeiten in unterschiedlichen Projektteams	<ul style="list-style-type: none"> • Alle Teams werden auf die gleichen Ziele ausgerichtet • Alle Teams haben die gleiche Kadenz • Abgestimmte Herangehensweisen sind integraler Bestandteil der Sync Points der Teams
3	Erschwerte Zusammenarbeit und mangelnder Informationsfluss, da Projektteams über mehrere Länder verteilt sind	<ul style="list-style-type: none"> • PI Planning als Teambuilding, da sich alle beteiligten Personen an einem Ort treffen • Unklarheiten, Rückfragen und Abhängigkeiten werden während des PI Plannings kollektiv besprochen • Verbleibende Risiken werden notiert und mittels Verantwortlicher Person weiter verfolgt
4	Langfristiger Planungshorizont	<ul style="list-style-type: none"> • Planung erfolgt mittels PI immer nur über einen Horizont von drei Monaten • Anforderungen werden erst verfeinert, wenn sie zur Implementierung benötigt werden
5	Starre Einbindung externer Zulieferer	<ul style="list-style-type: none"> • Externe Zulieferer werden mittels eigenen ART eingebunden • Externe Zulieferer werden in alle Planungsüberlegungen und konstanten Abgleich eingebunden
6	Fehlendes Kundenverständnis	<ul style="list-style-type: none"> • Das PI Planning teilt die Vision, wie der Kunde eine Funktion nutzen wird • Kunde im Fokus aller Überlegungen • Enger Austausch zwischen Product Owner und dem Entwicklungsteam

Abbildung 28 - Lösungsansätze mit der Anwendung von SAFe

Abbildung 28 zeigt, welche Ansätze SAFe verfolgt, um ausgewählte Problemstellungen zu lösen.

Vergleicht man den bisher verwendeten Ansatz mit der Anwendung von SAFe, so überwiegen die Vorteile bei der Anwendung von SAFe. Die Aufteilung der Projekte in Agile Release Trains in Kombination mit fixen Program Increments sorgen für eine ideale Umgebung für die Entwicklungsteams. Der Vorteil daran ist, dass mittels priorisierten Backlog neue Features über alle Agile Release Trains harmonisiert und implementiert werden. Die technische Ausarbeitung der Anforderungen und die Erarbeitung der relevanten Architektur erfolgt zeitnahe zur Entwicklung und verhindert dadurch, dass die Themen voneinander entkoppelt durchgeführt werden. Dieser Ansatz verhindert eine Vielzahl an ansonsten notwendigen Änderungen an Anforderungen.

Das PI Planning richtet den gesamten Agile Release Train auf ein gemeinsames Ziel aus und minimiert dadurch unnötig aufwändige Abstimmungen zwischen Disziplinen und Projekten. Aufkommende Fragen können bereits während der Planung geklärt oder als Folgeaktivität definiert werden.

Funktionen die bereits während eines vergangenen Program Increments implementiert wurden, werden weiterhin unter dem Einfluss von Marktveränderungen und sich ändernder Kundenbedürfnisse stehen. Diese Änderung bedarf einer Anpassung künftiger Backlog Items und benötigt weitere Nachbearbeitung in den Anforderungen, der Architektur und Implementierung.

Ein weiterer wichtiger Aspekt ist, dass es aus regulatorischer Sicht unbedingt notwendig ist, dass die Anforderungen und die Architektur vor der Implementierung erstellt und freigegeben werden. Eine Unterteilung der Dokumentation an die Anforderungen des Product Increments ist zu berücksichtigen. So können Dokumente mit jedem neuen Product Increment wachsen und entsprechend freigegeben werden.

5.2 Anforderungsdefinition

In jedem Projekt ist es elementar, dass ein einheitliches Verständnis von Anforderungen und die damit verbundenen Bedürfnisse der Stakeholder existiert. In klassischen Kundenauftragsprojekten definiert das Lastenheft die Bedürfnisse des Kunden an eine Softwareentwicklung, das Pflichtenheft beschreibt, wie das Unternehmen bzw. Projektteam die zuvor definierten Anforderungen in Funktionen umsetzen möchte. Bei Neuentwicklungen großer Medizintechnik Unternehmen definiert der Produktentwicklungsprozess, wie Anforderungen erfasst und umgesetzt werden müssen. Generell werden Lösungen in der Medizintechnik nicht für einen einzelnen Kunden entwickelt, sondern sind Forschungs- und Entwicklungsprojekte, die danach in Serie in hoher Stückzahl produziert werden. Je nach Größe der Instrumente und dem damit verbundenen Einsatzzweck variiert die Anzahl an Kundeninstallationen.

Abbildung 29 zeigt eine Sammlung von Problemstellungen aus der Anforderungsdefinition und führt auf, wie sich diese innerhalb des Projekts ausgewirkt haben.

Nr.	Problemstellung	Auswirkung
1	Sequenzielle Definition der Anforderungen	<ul style="list-style-type: none"> • Längere Definitionsphasen • Spätes Erkennen von Unklarheiten
2	Unklare technische Umsetzung während Definition der Anforderungen	<ul style="list-style-type: none"> • Schwierigkeiten während der Umsetzung • Hoher Aufwand zur Überarbeitung der Anforderungen
3	Lange Zeitspanne zwischen Definition und Implementierung	<ul style="list-style-type: none"> • Wissensverlust durch Abgänge im Projektteam • Hoher Dokumentationsaufwand um relevante Aspekte und Diskussionen festzuhalten
4	Frühes Definieren benötigt nachträgliches Anpassen der Anforderungen	<ul style="list-style-type: none"> • Hoher Aufwand um Architektur und Anforderungen anzupassen • Lange Wartezeiten bevor die Durchführung starten kann
5	Fehlende Anforderungen für Systeme, die miteinander interagieren müssen	<ul style="list-style-type: none"> • Hoher Abstimmungsaufwand • Große Abhängigkeiten zwischen den Systemen

Abbildung 29 - Problemstellungen der Anforderungen in vorliegendem Projekt

5.2.1 Bisheriger Ansatz - unter Anwendung klassischer Methoden

Der angewendete Entwicklungsprozess wurde bereits in Kapitel 4.2 beschrieben. Abbildung 30 zeigt die für den Entwicklungsprozess relevanten Meilensteine und die dazu gehörigen Phasen.



Abbildung 30 - Überblick über die Projektmeilensteine und ihre zugehörigen Phasen

Während der ersten Projektphase werden Stakeholder Requirements erstellt, parallel dazu findet die Erarbeitung der System Architektur statt.

In der nachgelagerten zweiten Projektphase werden die Requirements und Architektur aufgegriffen und in Product Requirements übersetzt. Die Erarbeitung technisch tiefer liegender Anforderungen findet sequenziell statt. Nach Abschluss der Definition der Product Requirements, werden diese in technische Requirements übersetzt. Technische Requirements gehen bereits konkret darauf ein, wie die technische Umsetzung erfolgen soll. Diese technische Umsetzung kann einerseits als Technical Requirements festgehalten werden, oder als Software Requirements. Technical Requirements beinhalten mechanische, mechatronische und elektronische Aspekte oder beinhalten eine Kombination derer. Software Requirements definieren ausschließlich Aspekte, die rein in Software implementiert werden. In dieser Phase werden ebenfalls Machbarkeitsstudien durchgeführt, sofern benötigt. Sollte die Verfeinerung der Requirements eine Anpassung der Architektur benötigen, so kann diese ebenfalls überarbeitet werden.

Ein Beispiel für ein Service Requirement könnte sein:

Das Produkt soll File (Up- und Download) und Screen Sharing unterstützen.

Das zugehörige Product Requirement könnte so lauten:

Ein externer Zugriff durch die Service Organisation auf das Produkt benötigt die Bestätigung eines Anwenders.

Abbildung 31 zeigt die benötigten Schritte in der Anforderungs- und Architekturdefinition zur jeweiligen Phase.

Phase	Anforderungen	Architektur
1	<ul style="list-style-type: none"> • Erstellen von Customer Requirements • Erstellen von Service Requirements 	<ul style="list-style-type: none"> • Erstellen der mechanischen Architektur • Erstellen der Softwarearchitektur • Erstellen der elektronischen Architektur • Erstellen der Gesamtarchitektur

		(Zusammenspiel aller Architektur) • Erstellen des Service Konzepts
2	<ul style="list-style-type: none"> • Erstellen von Product Requirements • Erstellen von Technical Requirements • Erstellen von Software Requirements 	<ul style="list-style-type: none"> • Durchführung Machbarkeitsstudien • Anpassung bestehender Architekturdokumente
3	<ul style="list-style-type: none"> • Erstellen von Design Spezifikationen bzw. Modul Anforderungen • Anpassungen bestehender Anforderungen (wenn notwendig) 	<ul style="list-style-type: none"> • Anpassung bestehender Architekturdokumente
4	<ul style="list-style-type: none"> • Keine neuen und keine Überarbeitung bestehender Anforderungen geplant 	<ul style="list-style-type: none"> • Keine neuen und keine Überarbeitung bestehender Architektur geplant

Abbildung 31 - Anforderungs- und Architekturdokumente in respektiven Projektphasen

In der dritten Projektphase werden detaillierte Anforderungen aus den zuvor definierten technischen Requirements geschrieben. Diese Anforderungen bzw. Spezifikationen sind sehr detailliert und beschreiben alle notwendigen Details zur Umsetzung des Requirements. Alle anderen Anforderungen und die Architektur bleiben bestehen und werden nur wenn unbedingt notwendig überarbeitet. In dieser Phase werden alle Anforderungen technisch in Form von Hardware oder Software umgesetzt.

Ein Beispiel für ein zugehöriges Software Requirement ist:

Ein Zugriff außerhalb des Gerätnetzwerks (network layer 452) kann durch die Service Organisation beantragt werden. Der Anwender des Produkts hat die Möglichkeit den Zugriff zu verweigern oder den Zugriff zu erlauben.

In der vierten Projektphase können keine neuen Anforderungen mehr erfasst werden, bestehende Anforderungen und Architektur sind nicht zur Überarbeitung geplant.

Alle Requirements haben eine n:n Verbindung, was bedeutet, dass ein Customer Requirement durch mehrere Product Requirements abgedeckt sein kann, eines dieser Product Requirements aber wiederum mehrere Customer Requirements abdecken kann.

Ein Beispiel für eine Design Spezifikation ist:

Bei positiv gewährtem Zugriff wird eine getunnelte Verbindung zur Service Organisation über Port 751 aufgebaut und der Service VNC für Screen Sharing gestartet.

Aus regulatorischer Sicht ist es wichtig, dass die Planung und Dokumentation von Anforderungen vor der Umsetzung erfolgt. Es muss nachvollziehbar dargestellt werden können, wie der Zusammenhang zwischen Requirements ist, was über eine so genannte Traceability Matrix abgedeckt wird.

Die Schwierigkeit dieser Art der Anforderungsanalyse und Definition ist, dass sich Probleme oftmals erst sehr spät ergeben, was bedeutet, dass man die komplette Kaskade an Anforderungen nochmals überarbeiten muss. Ein weiteres Problem ist das sequenzielle Abarbeiten der Requirements verschiedener Level, was oftmals durch verschiedene Personen durchgeführt wird. Vergeht zu viel Zeit zwischen der initialen Erfassung eines Requirements und dem Feedback, dass etwas angepasst werden muss oder etwas unklar definiert ist, kann es passieren, dass diese Personen bereits wieder in neuen Projekten eingeplant ist und dadurch die nötige Informationsquelle fehlt. Es wird versucht über detaillierte Review Zyklen dieses Risiko zu minimieren.

5.2.2 Anwendung von SAFe

In SAFe werden Anforderungen aller Level in entsprechenden Backlogs festgehalten. Die Large Solution Konfiguration sieht das Solution, Program und Team Backlog vor.

Das Solution Backlog beinhaltet künftige Capabilities und Enablers, die über mehrere Agile Release Trains umgesetzt werden können und erweitert die existierende Solution und deren Architectural Runway. Das Solution Backlog ist priorisiert und wird vom Solution Management erstellt.

Eine Capability ist mit einem Feature gleichzusetzen, allerdings umfasst die Umsetzung dessen meist mehrere Agile Release Trains und erstreckt sich typischerweise über mehrere Program Increments. Capabilities werden in mehrere Features zerlegt, damit die Umsetzung eines Features innerhalb eines Program Increment durchgeführt werden kann.

Enabler sind auf allen Ebenen von SAFe anzufinden und beschreiben einen architektonischen Klärungsbedarf, der benötigt wird, um eine Capability, Features oder User Story umzusetzen.

Die Enabler werden im jeweiligen Backlog erfasst und dem Architectural Runway zur Lösungsfindung übergeben.

Der Architectural Runway kümmert sich vorab um elementare Aspekte der Produktentwicklung und versucht Klarheit zu schaffen, bevor entsprechendes Backlog Item in einer Iteration im Team Backlog adressiert wird. Die Konsequenz einer nicht adäquaten Lösung vorab ist, dass der Agile Release Train verlangsamt wird und die Implementierung ins Stocken gerät. Weitere Auswirkungen, bei nicht adäquater Lösungsfindung, können exzessive Anpassungen im Design, Schwierigkeiten in der Integration und fehlende Testbarkeit oder ein Rückgang der Zuverlässigkeit bzw. Qualität des Produkts sein.

Generell unterscheidet SAFe zwischen Emergent design und Intentional Architecture.

Emergent design sind Architekturentscheidungen, die entweder kurz vor der Implementierung oder erst während der Implementierung getroffen werden müssen. Diese Entschei-

dungen können entweder vom Team selbst oder mit Hilfe eines System Architekten getroffen werden. Emergent design kann angewendet werden, wenn keine langwierige Vorbereitung gebraucht werden, Entscheidungen schnelllebig sein können oder nicht notwendigerweise zwischen den Teams konsistent sein müssen. Ein Beispiel hierfür wäre, die korrekte Einbindung eines Services in die existierende Software.

Intentional Architecture sind Architekturentscheidungen, die eher an klassische Entwicklungsmethoden und deren Architektur erinnern. Sie werden vorab definiert, haben meist Auswirkungen auf mehrere Teams oder Agile Release Trains, wodurch eine nachträgliche Anpassung nur mit hohen Kosten bewältigt werden kann. Ein Beispiel hierfür ist die Definition eines komplexen Interfaces zwischen zwei Systemen. Intentional Architecture benötigt oftmals auch eine Prüfung durch Machbarkeitsstudien. In solchen Fällen wird der Architekt zum Product Owner eines Entwicklungsteams und erarbeitet gemeinsam, ob die angedachte Lösung auch in der Realität umgesetzt werden kann.

Es ist wichtig, dass beide Methoden korrekt und ausgewogen zum Einsatz kommen. Verwendet das SAFe Team zu oft Intentional Architecture, so wird die Entwicklung starr und eher klassischer Natur. Auf neue Anforderungen kann nur noch mit hohem Aufwand eingegangen werden. Emergent design kann vor allem dazu verwendet werden, um kleine und überschaubare Architekturentscheidungen zu treffen, was dem SAFe Team viel Flexibilität bietet. Werden elementare Abstimmungen und Architekturentscheidungen allerdings fälschlicherweise dorthin verschoben, so wird die laufende Entwicklung verzögert und eine Abstimmung diverser Teams muss erfolgen.

NFRs, sogenannte Non Functional Requirements, sind die Basis eines jeden Backlogs und stellen sicher, dass Benutzerfreundlichkeit, Zuverlässigkeit und Skalierbarkeit die Basis alle funktionalen Backlog Einträge sind. Verfehlt die Produktentwicklung das Erreichen der NFRs, so kann die fehlende Qualität Auswirkungen auf Geschäfts-, Markt- oder Kundenbedürfnisse, sowie auf die Erfüllung von Vorgaben der Behörden oder Zulassungsstellen, haben. Die Spannweite der Auswirkung der nicht Erfüllung reicht von weniger Umsatz bis hin zur verweigerten Zulassung durch die Behörden oder deren Zulassungsstellen.

Das Program Backlog umfasst Features und Feature Enablers und definiert die Arbeit, die ein Agile Release Train im kommenden Program Increment oder einem folgenden Program Increment erbringen soll.

Ein Feature beschreibt eine Produkt- oder Servicefunktion und deckt einen Bedarf eines Stakeholders ab. Für die Erstellung der Features ist der Product Manager in Zusammenarbeit mit dem Product Owner und den Key Stakeholdern verantwortlich. Die Umsetzung eines Features darf sich über maximal ein Program Increment oder weniger erstrecken. Ein Feature setzt sich aus einer Beschreibung und einer Hypothese, die den Nutzervorteil oder den Businessvorteil beschreibt, zusammen.

Feature	Benefit Hypothesis
Externer Screen Sharing Zugriff	Schnelle Unterstützung für den Kunden bei technischen Problemen verbessert die Kundenzufriedenheit.

Abbildung 32 - Beispiel eines Features

SAFe Teams verwenden Agile Praktiken wie SCRUM und Kanban. Das erlaubt ihnen einerseits aufzuzeigen, welche User Stories gerade erledigt werden, zeigt andererseits auch, wo es Verbesserungsbedarf oder Engpässe gibt. Durch die Anwendung von Kanban Boards kann das Team jederzeit die Arbeit sichtbar machen. Jedes Item wird vom Team abgeschätzt und berücksichtigt Arbeitsaufwand, Komplexität und die Klar- bzw. Unklarheit des Arbeitspakets. Das Kanban Board umfasst alle Backlog Items des Teams für den aktuellen Sprint.

Das Team Backlog ist das, organisatorisch gesehen, tiefste Level an Backlogs. Es kann alle Arbeitspakete enthalten, die ein Team zur Entwicklung eines Systems benötigt, hauptsächlich sind dort aber User- und Enabler Stories vertreten.

User Stories beschreiben die benötigte Funktionalität und haben die Kundenbedürfnisse im Mittelpunkt ihrer Überlegungen. Die Erstellung verfolgt einer empfohlenen Schreibweise, siehe Abbildung 33.

Empfohlene Schreibweise	Angewandtes Beispiel
Als (Benutzer Rolle), möchte ich (Aktivität) damit ich (Business Wert)	Als Laborverantwortlicher, möchte ich meinen Bildschirm teilen können, damit ich Unterstützung bei Problemen anfordern kann und dadurch einen möglichst geringen System Ausfall habe.

Abbildung 33 - Beispiel einer User Story in SAFe

Der Vorteil dieser Schreibweise ist, dass die Teams besser die Benutzerprofile und deren Bedürfnisse verstehen lernen.

Enabler Stories beschreiben die benötigten Klärungen, die zur Erreichung einer User Story benötigt wird und können Themen wie bestehenden Softwarecode, Komponenten und technischer Infrastruktur beinhalten. Ein Beispiel hierfür ist:

Welcher Service und Port unterstützt Screen Sharing.

5.2.3 Vorteile / Nachteile der Anwendung von SAFe

Der Vorteil klassischer Herangehensweisen ist, dass man ein gesamtheitliches Bild von dem Produkt hat, alle Funktionen kennt und man somit bereits am Projektbeginn das Budget und den entsprechenden Zeitaufwand festlegen kann. Zudem ist es dem Unternehmen bekannt, dass die etablierten Prozesse von den Mitarbeitern verstanden werden und entsprechend verwendet werden können. Gegenüber Behörden sind die Prozesse bereits mehrfach geprüft und als adäquat beurteilt worden.

Der Nachteil klassischer Herangehensweisen ist, dass Probleme in den Anforderungen erst sehr spät entdeckt werden und dass das Unternehmen sehr träge ist, falls sich der Markt oder die Anforderungen der Stakeholder ändern. Die damit verbundenen Änderungen sind meist sehr aufwändig und somit auch kostspielig. Obwohl viele Projekte in der Vergangenheit mittels klassischer Methoden durchgeführt wurden, müssen oftmals Time-lines verschoben und das Budget erhöht werden.

Der bisherige Ansatz hatte eine sequenzielle Abfolge der Anforderungsdefinition. Nachdem beispielsweise alle Product Requirements definiert und unterschrieben sind, kann mit der Definition der davon abgeleiteten Technical und Software Requirements gestartet werden. Nachdem alle Requirements definiert sind, kann die Implementierungs- und Testphase starten. Die Auswirkung dessen ist, dass der Zeitraum der Definition sich über einen langen Zeitraum erschließt und das oftmals erst kritische Rückfragen bei der Implementierung und dem Test der Hardware und Software erfolgt.

Im Vergleich dazu wird in SAFe nur eine Teilmenge der Anforderungen während eines Program Increment behandelt, die Entwicklungsteams der Agile Release Trains liefern das neue Program Increments immer nach einem festgelegten Intervall von zwölf Wochen. Das bedeutet, dass bereits bei der Planung darauf Rücksicht genommen werden muss, dass die Features so definiert sind, dass sie innerhalb eines Program Increments von den Teams erledigt werden können.

Der Vorteil der Anwendung von SAFe ist, dass Anforderungen nach dessen Wichtigkeit priorisiert werden und alle 12 Wochen entschieden werden kann, welche Themen zur Implementierung freigegeben werden. Verändert sich der Markt oder die Anforderungen der Stakeholder, so kann leicht darauf reagiert werden, da noch nicht alle Detailfragen geklärt sind, sondern immer nur jene Themen und Aspekte, die für die nächste Implementierung benötigt werden.

Vereinfacht gesagt bestrebt SAFe, dass eine Anforderung vom Solution Backlog bis hin ins Team Backlog heruntergebrochen werden kann und anschließend innerhalb von zwölf Wochen implementiert und getestet wird.

Der Nachteil der Anwendung von SAFe ist, dass es nur vollständig die Vorteile inkrementeller Definition, Implementierung und Testen entfalten kann, wenn auch entsprechend der Entwicklungsprozess angepasst wird. Folglich muss das Unternehmen sein Quality Ma-

nagement System (QMS) von einem Wasserfall Ansatz zu einem Lean QMS transformieren. Bei einem Lean QMS werden ebenfalls bekannte Ansätze, unter Anderem mittels Plan-Do-Check-Adjust Methodik, zur Qualitätssicherung durchgeführt, allerdings werden die Zyklen an die Inkremente angepasst.

Nr.	Problemstellung	Lösungsansätze mit SAFe
1	Sequenzielle Definition der Anforderungen	<ul style="list-style-type: none"> • Priorisiertes Backlog nach Wert für den Kunden • Iteratives Abarbeiten anhand Program Increments
2	Unklare technische Umsetzung während Definition der Anforderungen	<ul style="list-style-type: none"> • Iteratives Abarbeiten anhand Program Increments • Technische Klärungen im Architectural Runway vor Umsetzung
3	Lange Zeitspanne zwischen Definition und Implementierung	<ul style="list-style-type: none"> • Priorisiertes Backlog nach Wert für den Kunden • Iteratives Abarbeiten anhand Program Increments
4	Frühes Definieren benötigt nachträgliches Anpassen der Anforderungen	<ul style="list-style-type: none"> • Flexibilität durch priorisiertes Backlog • Neuer Kundennutzen kann in jedem neuen Program Increment erzeugt werden
5	Fehlende Anforderungen für Systeme, die miteinander interagieren müssen	<ul style="list-style-type: none"> • Definition des Solution Backlogs • Verschiedene Agile Release Trains arbeiten parallel an entsprechenden Features

Abbildung 34 - Lösungsansätze mit der Anwendung von SAFe

Abbildung 34 fasst die bereits beschriebenen Lösungsansätze zusammen.

Vergleicht man den bisher verwendeten Ansatz mit der Anwendung von SAFe, so überwiegen die Vorteile bei der Anwendung von SAFe. Anforderungen werden nach Kundennutzen priorisiert und nur dann weiter verfeinert, wenn sie zeitnahe implementiert werden sollen. Der Planungshorizont der Agile Release Trains umfasst zwölf Wochen und beinhaltet die Backlog Items mit dem höchsten Kundennutzen. Durch das Erstellen von Enablern und der Adressierung derer im Architectural Runway wird sichergestellt, dass technische Herausforderungen und Unklarheiten in der Architektur vor deren Implementierung geklärt werden. Nachdem Backlog Items und die dazugehörige Architektur verfeinert wurden, werden deren Inhalte im nächsten Program Increment behandelt, sodass kein Wissensverlust zwischen der Erstellung der Anforderungen und der Implementierung entsteht.

5.3 Testablauf und Freigabeprozess

Die Verifizierung und Validierung eines Produkts in der Medizintechnik ist elementarer Bestandteil der zulassungsrelevanten Dokumentationen. Dabei muss der Hersteller Nachweisen, dass er alle Anforderungen entsprechend umgesetzt hat und Risiken für den Patienten und den Anwender unter Kontrolle hat. Alle verbleibenden Probleme und Risiken müssen dem Anwender ausgewiesen werden, sodass dieser sich mit diesen vor Kauf auseinandersetzen kann und Produkte miteinander vergleichen kann.

Abbildung 35 zeigt eine Sammlung von Problemstellungen die sich mit den momentanen Testabläufen und dem Freigabeprozess ergeben.

Nr.	Problemstellung	Auswirkung
1	Terminierte Lieferungen entsprechen nicht den Erwartungen	<ul style="list-style-type: none"> • Verspätete Lieferungen verzögern die System Tests • Lieferungen in mangelnder Qualität sind nicht integrierbar
2	Finale Testkampagnen am Ende des Projekts	<ul style="list-style-type: none"> • Hoher Aufwand, um Qualität formal zu Bestätigen • Funktionen, die vorher nicht getestet werden konnten, haben das Risiko das Projekt zu verzögern
3	Starrer Freigabeprozess	<ul style="list-style-type: none"> • Kunde erhält das Produkt, sobald alle Anforderungen implementiert und getestet wurden

Abbildung 35 - Problemstellungen des Testablaufs und Freigabeprozesses

5.3.1 Bisheriger Ansatz - unter Anwendung klassischer Methoden

In der Vergangenheit wurden Hardware, Software und andere benötigte Komponenten in einem Big Bang Ansatz integriert und getestet. Das bedeutet, dass man alle Anforderungen vollständig implementiert hat und erst dann begonnen hat, alle Komponenten zusammen, zu integrieren und zu testen. Dadurch gab es in späten Projektphasen einen sehr hohen Aufwand für Anpassungen an Hardware und Software.

Dieser Ansatz wurde die letzten Jahre immer weiter adaptiert, in Projekt Titan hat man sich dafür entschieden, dass alle Komponenten Hardware und Software entwickeln können, bis Funktionsmodelle zur Verfügung stehen. Ab diesem Zeitpunkt gibt es alle drei Monate eine Integration auf System Level, die immer Software und von Zeit zu Zeit auch Hardware beinhaltet. Dieser Ansatz hilft den Komponenten Teams frühzeitig Probleme an den Schnittstellen zu erkennen und ermöglicht den Teams, dass ihre Komponenten in einem kundennahen Laborumfeld getestet werden.

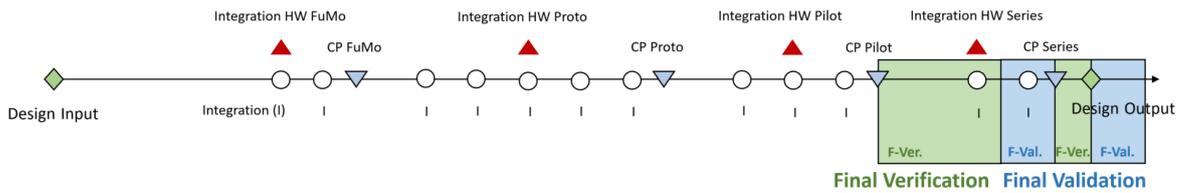


Abbildung 36 - Integrationspunkte und Zeitpunkte finaler Verifikation und Validation

Formal müssen alle Anforderungen mittels Testfällen abgedeckt sein und in einer finalen Testkampagne durchgeführt werden. Finale Tests müssen auf allen Ebenen des V-Modells durchgeführt werden, Abweichungen und aufgetretene Fehler müssen dokumentiert werden. Das Projektteam muss zusammen mit den Stakeholdern bewerten, ob diese Fehler noch innerhalb des Projekts gelöst werden müssen, oder ob das Produkt mit ihnen in den Markt eingeführt werden kann.

Generell kann man zwischen einer finalen Verifikationskampagne und einer finalen Validierungskampagne unterscheiden. Bei der finalen Verifikation wird geprüft, ob alle Anforderungen korrekt in Hardware und Software implementiert wurden. Die finale Validation prüft, ob Anforderungen der Stakeholdern korrekt in einem kundenähnlichen Umfeld durchgeführt werden können und weist somit die Zweckbestimmung des Medizinprodukts nach.

Finale Kampagnen können prinzipiell durchgeführt werden, sobald Software und Hardware einen Reifegrad haben, dass sie potenziell dem Kunden ausgeliefert werden können. Diese Hard- und Softwarestände werden danach eingefroren und alle künftigen Änderungen müssen auf potenzielle negative Auswirkungen geprüft werden. Zu diesem Zweck hat man sich im Projekt Titan entschieden, dass alle Tests bis Checkpoint Pilot nur als vorläufige Tests durchgeführt werden. Ab Checkpoint Pilot startet die finale Verifikation, gefolgt von einer finalen Validation. Wie in Abbildung 36 zu sehen, findet eine zweite finale Verifikation und Validierung statt, damit allfällige Fehler nochmals mit neuer Software getestet werden können. Ebenfalls fällt während dieser Zeitspanne die Integration von Serienhardware an, typischerweise ist aber davon auszugehen, dass der Reifegrad der Piloten bereits so gut ist, dass es sich nur noch um minimale Modifikationen handelt, die ebenfalls nochmals den Verifikations- und Validierungsprozess durchlaufen müssen. Die Freigabe des Produkts findet am Ende des Projekts zu Meilenstein Launch Decision statt.

Eine Integration auf System Level beinhaltet alle Lieferungen zu einem terminierten Zeitpunkt. Das Objekt der Integration können Lieferungen von Hardware und Software sein. Abbildung 37 zeigt, dass alle Lieferungen der Komponenten (in blau), der externen Zulieferer (in orange) und der anderen System Produkte (in grün) gemeinsam integriert und getestet werden.

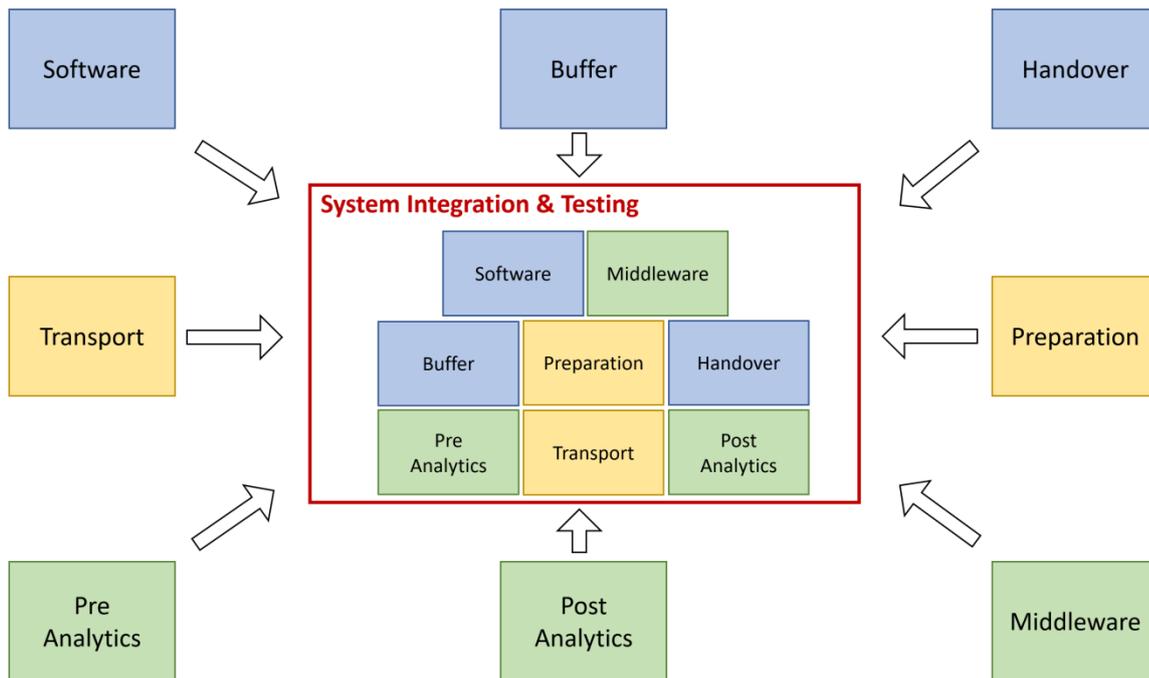


Abbildung 37 - Ansatz System Integration & Testing

Die regelmäßige Integration in eine definierte und kontrollierte Umgebung erlaubt es den Teilnehmern, neue Entwicklungen an den Schnittstellen zu prüfen und somit frühzeitig schwerwiegende Fehler zu identifizieren.

Die Lieferungen erfolgten in der Vergangenheit oftmals unvollständig, ungetestet oder verzögert. Unvollständige Lieferungen beinhalten nicht alle zuvor definierten Inhalte und können deshalb eine negative Auswirkung an den Schnittstellen haben, was eine Integration teilweise unmöglich macht. Ungetestete Lieferungen können einen ähnlich negativen Effekt auf die Schnittstellen haben, da nicht bekannt ist, ob die Schnittstellenanforderungen gewährleistet werden. In diesen Fällen ist es sehr schwierig, die Ursache für die Probleme zu finden. Da alle Lieferungen der Komponenten, der Externen Zulieferer und Systemen eine Abhängigkeit zueinander haben, führt eine verzögerte Lieferung zu einer Verzögerung der Integration.

Zur Minimierung der Qualitätsproblemen wurde damit gestartet, eine virtuelle Testumgebung aufzubauen, in der Software vorab getestet werden kann. Wie Abbildung 38 zeigt, ist der idealtypische Ablauf, dass zuerst die Tests auf Komponenten Ebene durchgeführt werden, gefolgt von einer Integration und Test in der virtuellen Testumgebung und erst dann eine physische Integration auf System Ebene erfolgt. Sobald die Integration auf System Ebene abgeschlossen ist und die Qualitätsziele erreicht wurden, startet das System Testing in kundenähnlicher Umgebung und testet vorab bereits implementierte Funktionalitäten.

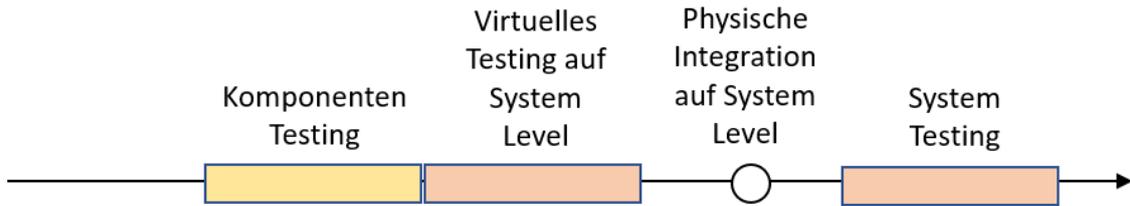


Abbildung 38 – Ablauf mit Einbindung virtueller Tests

Der Aufbau der virtuellen Umgebung hatte bereits positive Auswirkungen auf die Qualität der Lieferungen, allerdings werden häufig kritische Fehler gefunden, sodass die Integration nach wie vor mehrfach verschoben werden muss.

5.3.2 Anwendung von SAFe

Eines der elementaren Bausteine beim Testen in SAFe ist, dass alle Teams kontinuierlich ihre neue Software automatisiert erstellen und testen können. Bei der sogenannten Continuous Integration geht es vor allem darum, möglichst früh und möglichst automatisiert auf potenzielle Probleme zu stoßen. Zu diesem Zweck unterstützt das System Team die agilen Teams durch das Erstellen einer Continuous Integration Pipeline und erstellt Testszenarios, die dabei eine Solution Sicht abdeckt. Dadurch, dass Projekt Titan mehrere ARTs beinhaltet, muss das System Team eine entsprechende Infrastruktur zur Verfügung stellen, die auch End to End Szenarien abdeckt. Die agilen Teams selbst, erstellen auch automatisierte Testszenarios, die das Gesamtheitliche Ziel unterstützen.

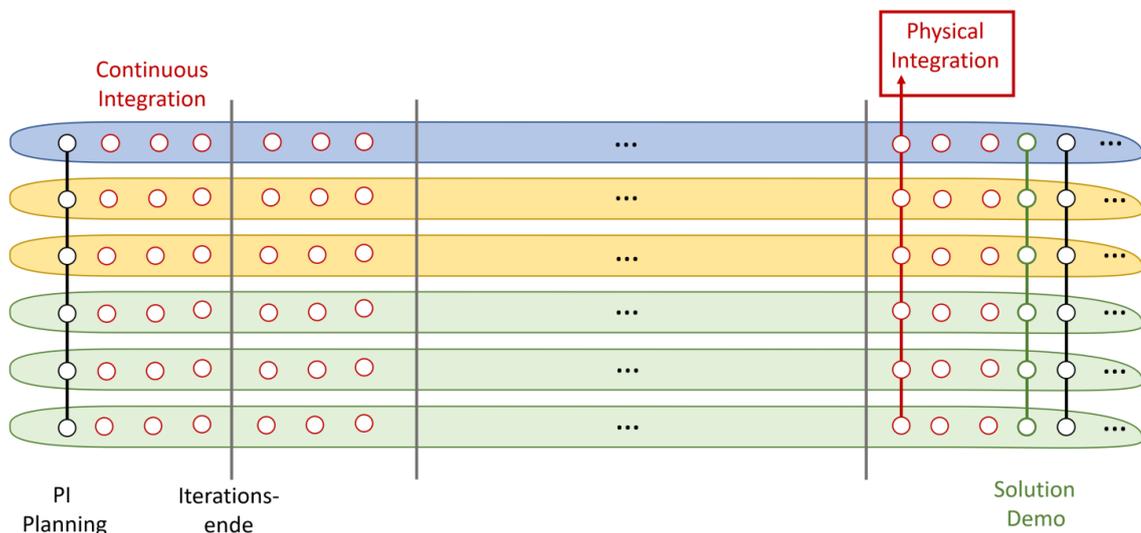


Abbildung 39 - Aufbau des Solution Integration Environments

In Abbildung 39 mit roten Kreisen dargestellt, sieht man Zeitpunkte für Continuous Integration die als Minimum gesehen werden. Generell ist es jedem Team frei überlassen, ob es öfters Integrieren möchte, theoretisch könnte auch mehrmals täglich integriert werden. Im Continuous Integration Test Durchlauf werden alle Szenarien der agilen Teams

selbst und jene End to End Szenarien des System Teams durchgeführt. Stellt man Unstimmigkeiten fest, so kann sofort darauf reagiert werden.

Durch die Entwicklung von Hard- und Software bleibt es notwendig, dass neue Software auch auf der bestehenden Hardware integriert und getestet wird. Für diese Aktivität ist einerseits jedes Agile Team selbst verantwortlich, andererseits bleibt für die Bereitstellung der physischen Infrastruktur auf Solution Ebene weiterhin das System Team zuständig, dargestellt in Abbildung 39 mittels roter Box. Die physikalische Integration ist mit einmal pro Program Increment festgelegt und findet zu Beginn der letzten Iteration statt. Bei Bedarf kann die Häufigkeit der physischen Integration noch weiter erhöht werden, hier gilt es aber sorgfältig abzuwägen, da physische Integration mit einem erheblichen Mehraufwand zu beziffern ist, verglichen mit Continuous Integration.

Wird neue Hardware zur Verfügung gestellt, so verhält es sich gleich wie bei bestehender Hardware, allerdings wird die Deinstallation und Installation bereits eine Iteration früher durchgeführt.

Formell gesehen, dient die Continuous Integration Pipeline dazu, sofort bestehende Qualität bestätigen zu können und neue Änderungen mit den Erwartungen abzugleichen. Automatisierte Tests dienen unterstützend zur Verifikation. Zusätzlich werden weiterhin manuelle Tests mit Hard- und Software durchgeführt, die zusätzlich die User Stories abtesten. Diese Tätigkeit wird von den agilen Teams im Rahmen ihrer Möglichkeiten durchgeführt und vom System Team durch Schnittstellentests, sowie übergreifenden manuellen End to End Szenarien ergänzt. Das System Team verfügt ebenfalls über ein Team Backlog, das die End to End Aspekte der Solution in User Stories abdeckt und entsprechend testet. Ein weiterer formeller Test der durch das System Team unterstützt wird, ist die Validierung.

Formelle finale Tests werden von den agilen Teams dann durchgeführt, sobald eine Funktion voll implementiert ist und keine weitere Änderungen geplant sind. Diese Tätigkeit kann während einer Iteration, am Ende einer Iteration oder am Ende eines Product Increments ausgeführt werden. Das System Team führt finale Tests nach Abschluss der Integration und erfolgter Durchführung der Solution Demo durch. Nach Durchführung der Solution Demo kann sichergestellt werden, dass die Capability in allen ARTs korrekt implementiert und von allen Stakeholdern als gut befunden wurde.

5.3.3 Vorteile / Nachteile der Anwendung von SAFe

Bei der Anwendung von SAFe im Bereich der Integration und des Testens entstehen keine klaren Vorteile und kann daher als neutral bewertet werden.

Bei der Entwicklung und beim Testen von Hardware empfiehlt SAFe, dass bereits in frühen Phasen Computer-Aided Design (CAD) Integrationen durchgeführt werden, erste Modelle mittels 3D Druck erfolgen und Funktionalitäten nach deren Wichtigkeit für das Ge-

samtsystem mit Priorität behandelt werden. Diese Tätigkeiten sind bereits heute integraler Bestandteil der Hardwareentwicklung und bieten dadurch keinen neuen Mehrwert.

Das Testen von Software wird bereits durch automatisierte Tests durchgeführt. Durch auftauchende Qualitätsprobleme hatte man sich bereits dazu entschieden, dass eine virtuelle Umgebung geschaffen wird, die Schnittstellentests und End to End Tests abbildet und entsprechend schnell den Komponenten Teams Feedback geben kann. Continuous Integration wäre ein Fortschritt, da dort öfters bzw. jederzeit integriert werden kann. Grundsätzlich könnte dieser Fortschritt in beiden Szenarios umgesetzt werden und scheidet momentan noch an der Bereitschaft der Teams auf entsprechende Infrastruktur bereits heute zuzugreifen.

Eine Umstellung der finalen Verifizierung und Validierung auf Increment basierte finale Kampagnen bietet eine Zeitersparnis. Dabei gilt es zu beachten, dass die Dokumentation dennoch erbracht werden muss und entsprechend auf Increment basierte Freigabeprozesse umgestellt werden muss.

Durch die Einführung von SAFe entsteht am Ende eines Program Increment, nach Abschluss der anschließenden finalen Verifizierung und Validierung, ein Increment, das potenziell auf den Markt eingeführt werden könnte. Da Medizinprodukte allerdings nicht nur aus Software, sondern aus einer Kombination von Software und Hardware bestehen, muss vor einer Markteinführung Hardware in Serienreife zur Verfügung stehen. Durch die Anwendung von SAFe ist aus momentaner Sicht keine eindeutige Beschleunigung der Hardwareentwicklung erkennbar. Dieser Aspekt führt dazu, dass die Anpassung des Freigabeprozesses zwar möglich und keine klaren Nachteile hat, allerdings ergeben sich auch keine weiteren Vorteile die eine Einführung von SAFe begünstigen würde.

Nr.	Problemstellung	Lösungsansätze mit SAFe
1	Terminierte Lieferungen entsprechen nicht den Erwartungen	<ul style="list-style-type: none"> • Fixe Kadenz mit klaren Zielen für jedes Team • Continuous Integration als integraler Bestandteil der Entwicklung
2	Finale Testkampagnen am Ende des Projekts	<ul style="list-style-type: none"> • Increment basierte Verifikation und Validation
3	Starrer Freigabeprozess	<ul style="list-style-type: none"> • Mit jedem Program Increment entsteht ein Produkt, das freigegeben werden könnte

Abbildung 40 - Lösungsansätze mit der Anwendung von SAFe

6 Zusammenfassende Bemerkungen

Das abschließende Kapitel fasst mit seinen Unterkapiteln den Inhalt der Arbeit und dessen Fazit zusammen und gibt einen Ausblick und erörtert weiteren Forschungsbedarf.

6.1 Zusammenfassung und Fazit

Es gibt eine Vielzahl an Gründen, wieso jährlich eine große Menge an Projekten scheitern. Unzureichende Kommunikation, ständig sich ändernde Anforderungen und schlechte Projektplanung sind nur einige davon. Agile Methoden versprechen generell besser mit sich ändernden Rahmenbedingungen umgehen zu können, allerdings sind diese Methoden oftmals auf reine Softwareentwicklung ausgelegt.

Mit SAFe hat das Portfolio der agilen Entwicklungsmethoden einen weiteren Vertreter, der auch großen Unternehmen durch die Skalierbarkeit des Frameworks einen Mehrwert bieten kann. SAFe greift viele bereits bekannte Methoden auf und vereint sie in einem Framework, das Unternehmen und Teams dabei helfen soll, sich mehr damit zu beschäftigen was Kunden einen echten Mehrwert bietet. Das Framework bietet durch dessen Skalierbarkeit zudem Unternehmen Methodiken zur Durchführung von Großprojekten. Benannte Methodiken beinhalten Prozesse zur Entwicklung von Software, Hardware und Mechatronik und zeigen wie die dafür benötigten Anforderungen, Architekturentscheidungen und Testabläufe getroffen bzw. erstellt werden können.

Im Zuge dieser Arbeit wurden drei Aspekte genauer betrachtet, Projektplanung und Organisation, Anforderungsdefinition und Testablauf und Freigabeprozess.

Bei der Projektplanung und Organisation konnten klare Vorteile identifiziert werden. SAFe unterteilt die Projektplanung in Program Increments von drei Monaten. Die Detailplanung, sowie die dazu benötigten Anforderungen und Architektur werden nur dann erstellt, wenn sie für das kommende Program Increment benötigt werden. Die Teams werden dabei so aufgestellt, dass sie das vorgegebene Ziel optimal und mit möglichst wenig Abstimmung erreichen können. Das Einbeziehen externer Zulieferer, anderer Systeme des Unternehmens, sowie der eigenen Projektteams in die Planung des jeweiligen Program Increments hilft ein einheitliches Verständnis zu schaffen und vermeidet Kommunikationsverluste.

Die Anforderungsdefinition besteht auf Solution Level mittels Solution Backlog, welches von allen Stakeholdern priorisiert wird. Backlog Items mit der höchsten Priorität, bekommen die höchste Aufmerksamkeit und werden zeitnahe zum anstehenden Program Increment weiter verfeinert. Backlog Items mit geringeren Priorität, bleiben im Solution Backlog für künftige Program Increments bestehen, werden aber erst dann verfeinert und

geplant, wenn sie tatsächlich benötigt werden. Für das Unternehmen entstehen hier durch die Einführung von SAFe klare Vorteile, da nur die Arbeiten getätigt werden, die unmittelbar wichtig für den Kunden sind und Aufwand durch stetige Änderungsprozesse minimiert werden kann.

Der Testablauf und Freigabeprozess wurde in letzter Zeit im Projekt bereits verfeinert, wodurch sich kein klarer Vorteil durch die Einführung von SAFe bemerkbar machen würde.

Die Erstellung und Freigabe von regulatorisch relevanter Dokumentation bleibt auch mit Einführung von SAFe ein wichtiger Aspekt. Bei korrekter Anwendung und entsprechender Etablierung im Quality Management System können Dokumente inkrementell wachsen und freigegeben werden.

6.2 Ausblick und weiterer Forschungsbedarf

In vorliegender Arbeit wurden Teilaspekte von SAFe durchleuchtet und die Vor- und Nachteile dessen Einführung geprüft. Bevor SAFe im Unternehmen eingeführt werden kann, müssen nicht betrachtete Aspekte weiter durchleuchtet werden. Gesamtheitlich müssen vor Einführung alle Aspekte in dem entsprechenden Quality Management System verankert werden, damit das Projekt und das Unternehmen auch weiterhin reibungsfrei neue Produkte am Markt zulassen kann.

Ein wichtiger Aspekt ist, dass durch die Einführung von SAFe sich bisher etablierte Rollenbilder verändern und Individuen eventuell Anpassungen in ihrer Karriereplanung vornehmen müssen. Dieser Aspekt birgt großes Konfliktpotenzial und sollte entsprechend durch das Unternehmen begleitet werden.

Veränderungen wecken einerseits Aufbruchstimmung und andererseits Ängste. Vor Einführung ist deshalb darauf zu achten, dass ausreichend Mitarbeiter in verschiedenen Funktionen und Disziplinen auf SAFe geschult werden, damit die Organisation bereit ist für den Wandel.

Literatur

- agilemanifest.org*. (abgerufen am 03. Oktober 2021). Von <https://agilemanifesto.org/iso/de/manifesto.html> abgerufen
- Amtsblatt der Europäischen Union. (5. April 2017). *WEBSITE DES EUROPÄISCHEN PARLAMENTS UND DER RAT DER EUROPÄISCHEN UNION*. Von <https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:32017R0745&from=IT> abgerufen
- Burghardt, M. (2013). *Einführung in Projektmanagement*. Erlangen: Publicis Erlangen.
- FDA. (11. Mai 2005). *FDA*. Von <https://www.fda.gov/files/medical%20devices/published/Guidance-for-the-Content-of-Premarket-Submissions-for-Software-Contained-in-Medical-Devices---Guidance-for-Industry-and-FDA-Staff.pdf> abgerufen
- Kusay-Merkle, U. (2018). *Agiles Projektmanagement im Berufsalltag*. Berlin: Springer Gabler.
- Kuster, J., Bachmann, C., Huber, E., Hubmann, M., Lippmann, R., Schneider, E., . . . Wüst, R. (2019). *Handbuch Projektmanagement*. Berlin: Springer Gabler.
- Leffingwell, D. (aufgerufen am 16.10.2021). *Scaled Agile Framework*. Von <https://www.scaledagileframework.com/#> abgerufen
- Leffingwell, D., & Knaster, R. (2020). *SAFe distilled; Achieving Business Agility with the Scaled Agile Framework*. USA: Addison-Wesley.
- Rundstedt & Partner. (29. August 2018). *Statista*. Abgerufen am 07. Februar 2021 von Statista: <https://de.statista.com/statistik/daten/studie/911835/umfrage/fachkraeftemangel-in-der-schweiz-nach-berufen/>
- SwissQ - Trends & Benchmarks Report Switzerland. (aufgerufen am 05.01.2021). *SwissQ*. Von SwissQ: <https://report20.swissq.it/> abgerufen
- Timinger, H. (2017). *Modernes Projektmanagement*. Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA.
- Wastian, M., Braumandl, I., & von Rosenstiel, L. (2009). *Angewandte Psychologie für Projektmanager*. Heidelberg: Springer MedizinVerlag.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Kriens, den 16. November 2021

Martin Urragg