

Bachelorarbeit

im Bachelorstudiengang

Wirtschaftsinformatik (Fakultät IM)

an der Hochschule für angewandte Wissenschaften Neu-Ulm / Technische Hochschule Ulm

Entwicklung eines Reifegradmodells zur Bewertung von Systemen und Software

Erstkorrektor/-in: Prof. Dr. Oliver Griebel
Zweitkorrektor/-in: Prof. Dr. Jörg-Oliver Vogt
Betreuer/-in: Alexander Bunk

Verfasser/-in: Jan Schrade (Matrikel-Nr.: 3137931)
Hochschulsemester: 11

Arbeit angemeldet: 01.05.2025
Arbeit abgegeben: 20.08.2025

Abstract

In der folgenden wissenschaftlichen Arbeit wird das Thema einer Entwicklung eines Reifegradmodells zur Bewertung von Systemen und Software behandelt. Durch die zunehmende Anzahl von Systemen und Software besteht in Zukunft die Notwendigkeit einer Methode zur Bewertung, um festzustellen, ob Systeme oder Software auf einem guten technologischen Stand sind. Mit Hilfe des zu entwickelnden Reifegradmodells soll solch eine Bewertung und Einschätzung möglich sein. Zudem wird eine Checkliste erstellt, die dieses Reifegradmodell unterstützt und bei der Bewertung von Benutzerfreundlichkeit, Kosten und Skalierbarkeit eine große Rolle spielt.

Daraus ergibt sich die Forschungsfrage für die folgende Analyse: Wie kann ein praxisorientiertes Reifegradmodell zur Optimierung der IT-Infrastruktur entwickelt werden, welche Maßnahmen zu den Automatisierungen und Systemanpassungen beachtet, integriert und anhand von bestimmten Kriterien wie Benutzerfreundlichkeit, Effizienzsteigerung, Kosten und Skalierbarkeit bewertet werden?

Um diese Punkte zu beantworten, wurde zuerst eine Recherche zu verschiedenen bereits existierenden Reifegradmodellen angewendet. Mithilfe eines Ablaufplans zur Erstellung eines Reifegradmodells wurde ein neues Reifegradmodell entwickelt, das auf der Grundlage von anderen Reifegradmodellen basiert und durch Erweiterungen angepasst wurde. Durch die iterative Entwicklung wurde das Modell und die Checkliste erstellt und in mehreren Schritten verbessert. Nach einer Evaluation mit einem Unternehmen wurde das Reifegradmodell mit seiner Checkliste in der Praxis erprobt und bestätigte dabei die zuvor zu erwarteten Ergebnisse.

Daraus resultiert, dass das Reifegradmodell zur Bewertung von Systemen und Software eine geeignete Einschätzung treffen kann. Gleichzeitig bietet das Reifegradmodell durch bestimmte Weiterentwicklungen mehr Potenzial für noch präzisere Bewertungen.

Inhalt

Abstract	2
1 Abbildungsverzeichnis	5
2 Tabellenverzeichnis	5
3 Einleitung	6
3.1 Forschungsdesign.....	6
4 Theoretische Grundlagen.....	7
4.1 Informationssysteme	7
4.2 Software	7
4.3 Systemsoftware.....	7
4.3.1 Betriebssysteme	8
4.4 Anwendungssoftware	8
4.4.1 Individualsoftware	8
4.4.2 Standardsoftware	9
4.4.3 Qualitätsmerkmale von Software.....	9
4.5 Reifegradmodell.....	10
4.5.1 Übersicht verschiedener Reifegradmodelle	10
4.5.1.1 Capability Maturity Model (CMM).....	10
4.5.1.2 Control Objectives for Information and Related Technology	11
4.5.1.3 ISO/IEC 15504	11
4.5.1.4 Business Process Maturity Model.....	12
4.5.1.5 Capability Maturity Model Integration	12
5 Vergleich und Analyse von Reifegradmodellen.....	13
5.1 CMMI.....	13
5.2 SPICE.....	16
5.3 Vergleich der zwei Reifegradmodelle	20
6 Vorgehen bei der Entwicklung des Reifegradmodells	22
7 Entwicklung des Reifegradmodells	23
7.1 Unternehmensvorstellung	23
7.2 Problemdefinition	24
7.3 Modellvergleich.....	25
7.4 Festlegung der Entwicklungsstrategie	25
7.5 Iterative Reifegradmodellentwicklung.....	25
7.6 Konzeption von Transfer und Evaluation	28
7.7 Implementierung der Transfermittel	28
7.8 Durchführung der Evaluation.....	31
7.9 Modellweiterentwicklung und Lebenszyklus.....	31

8	Fazit	36
9	Anhang	37
10	Literaturverzeichnis	50

1 Abbildungsverzeichnis

Abbildung 1: Reifegradstufen des Modells CMMI	14
Abbildung 2: CMMI-Modell mit Prozessbereichen und Stufen.....	15
Abbildung 3: Beispiel für Equivalent Staging.....	16
Abbildung 4: SPICE Dimensionsmodell.....	17
Abbildung 5: Beispiele für Primary Life Cycle Processes	18
Abbildung 6: Beispiele für Organizational Life Cycle Processes	18
Abbildung 7: Beispiel Support Processes.....	18
Abbildung 8: Reifegraddimension.....	19
Abbildung 9: Vorgehensmodell Reifegradmodellentwicklung	22
Abbildung 10: Neu entwickeltes Reifegradmodell.....	27
Abbildung 11: Fragen zur Checkliste	29
Abbildung 12: Checkliste zur Bewertung von Systemen/Software Teil 1	29
Abbildung 13: Checkliste zur Bewertung von Systemen/Software Teil 2	30
Abbildung 14: Checkliste zur Bewertung von Systemen/Software Teil 3	30
Abbildung 15: Fragen zur Checkliste	32
Abbildung 16: Checkliste IBM Cognos Teil 1	33
Abbildung 17: Checkliste IBM Cognos Teil 2.....	33
Abbildung 18: Checkliste Microsoft Power Bi Teil 1	34
Abbildung 19: Checkliste Microsoft Power BI Teil 2.....	34
Abbildung 20: Checkliste ChemGes Teil 1	35
Abbildung 21: Checkliste ChemGes Teil 2	35
Abbildung 22: Übersicht der Systeme und Software der Herrmann Bantleon GmbH.....	37
Abbildung 23: Benutzung der verschiedenen Systeme und Anwendungen der verschiedenen Abteilungen .	38
Abbildung 24: Neu entwickeltes Reifegradmodell.....	39
Abbildung 25: Checkliste zur Bewertung von Systemen/Software Teil 1	40
Abbildung 26: Checkliste zur Bewertung von Systemen/Software Teil 2	41
Abbildung 27: Checkliste zur Bewertung von Systemen/Software Teil 3	42
Abbildung 28: Checkliste IBM Cognos Teil 1	43
Abbildung 29: Checkliste IBM Cognos Teil 2.....	44
Abbildung 30: Checkliste Microsoft Power BI Teil 1.....	45
Abbildung 31: Checkliste Microsoft Power BI Teil 2.....	46
Abbildung 32: Checkliste ChemGes Teil 1	47
Abbildung 33: Checkliste ChemGes Teil 2	48
Abbildung 34: Fragen zur Checkliste.....	49

2 Tabellenverzeichnis

Tabelle 1: Zusammenfassung Vergleich der Reifegradmodelle CMMI und SPICE.....	21
---	----

3 Einleitung

Durch die sich immer weiterbildende Digitalisierung kommt es in vielen Unternehmen zu weitreichenden Veränderungen in deren Strukturen und Anforderungen. Dabei sollen nicht nur Dokumente und Abläufe mithilfe digitaler Medien verbessert und aufbereitet werden, sondern auch die bereits verwendeten IT-Systeme und Anwendungen. In zahlreichen Unternehmen findet man dabei veraltete Systeme und Software mit unzureichenden und ineffizienten Prozessen, die zu redundanten und überflüssigen Arbeitsschritten führen. Dies erhöht die Komplexität und mindert die Produktivität der allgemeinen Arbeitsschritte.

Die Arbeit verfolgt das Ziel, eine Lösung zu finden, die den Anforderungen der Mitarbeitenden und der Systeme entgegenzukommen und aus beiden die bestmöglichen Ergebnisse herauszuholen. Das allgemeine Ziel ist dabei die IT-Landschaft effizienter zu gestalten. Ein großer Anteil spielt dabei die richtige und effiziente Nutzung der bereits vorhandenen IT-Systeme.

Aus diesem Zusammenhang ergibt sich folgende Forschungsfrage: Wie kann ein praxisorientiertes Reifegradmodell zur Optimierung der IT-Infrastruktur entwickelt werden, das Maßnahmen zu den Automatisierungen und Systemanpassungen berücksichtigt, integriert und anhand von bestimmten Kriterien wie Benutzerfreundlichkeit, Effizienzsteigerung, Kosten und Skalierbarkeit bewertet werden?

3.1 Forschungsdesign

Bereits bestehenden Konzepte wie Cloud Computing, Lean IT, IT Infrastructure Management, IT Lean Management haben bereits große Anteile zum Thema Verbesserung der Systemlandschaft geleistet. Trotz dieser unterschiedlichen Ansätze zur IT-Optimierung besteht weiterhin Forschungsbedarf bei der Ermittlung von bewährten und vertrauenswürdigen Vorgehensweisen für die Implementierung und Verbesserung von Systemen und Software. Diese sollen in Zukunft nicht nur benutzerfreundlich, sondern auch effizienzsteigernd sein.

Als Grundlage dafür soll die Forschungsmethode Design Science Research angewendet werden, um ein Artefakt, in diesem Fall ein Reifegradmodell zur Bewertung von Systemen, zu erstellen. Dieses Modell soll es ermöglichen, dass die bereits bekannten Systeme und Software evaluiert werden, indem die IT-Umgebung analysiert wird. Dabei sollen die Systeme und Software auf Benutzerfreundlichkeit, Effizienz, Automatisierungsgrad und viele weitere Eigenschaften bewertet werden, um dadurch eine richtungsweisende Entscheidung treffen zu können.

Das Artefakt wird aufgrund der Eigenschaften von Systemen und Software erstellt und abschließend in einer realen IT-Umgebung überprüft und getestet. Ziel ist es, durch Effizienzsteigerung, Zeiterparnis, Skalierbarkeit oder Benutzerfreundlichkeit eine Optimierung der IT-Infrastruktur zu erreichen.

4 Theoretische Grundlagen

Für die Entwicklung des Artefaktes ist eine ausreichende Grundlage zu verschiedenen Themen erforderlich. Dabei wird der Schwerpunkt auf den Themen System und Software liegen, sowie auf der Grundlage wie ein Reifegradmodell erarbeitet und erstellt werden kann, um ein Modell zur Bewertung bereitzustellen.

4.1 Informationssysteme

Informationssystemen sind „soziotechnische Systeme, die aus menschlichen und maschinellen Komponenten bestehen“ (Kaufmann & Müller,2022 zitiert nach WKWI11, S.1). Diese Systeme erfüllen Aufgaben und unterstützen somit die Benutzer bei ihrer alltäglichen Arbeit. Der Fokus liegt darauf, die richtigen Informationen, Formen, Zeiten und Orte bereitzustellen. Grundsätzlich muss man unterscheiden zwischen automatisierbaren und nicht automatisierbaren Aufgaben. Der automatisierbare Teil entspricht dabei dem Teil des Anwendungssystems. Anwendungssysteme bestehen meist aus verschiedenen Komponenten, wie Hardware, Software, Benutzer und Daten.¹

4.2 Software

Den Begriff Software muss man aus zwei unterschiedlichen Definitionsbereichen betrachten. Der Bereich der technischen Definition umfasst Software aller Programme, die dem Rechner bereitgestellt sind. Nach der betriebswirtschaftlichen Definition handelt es sich bei Software um alle Ware und Leistung, die für eine erfolgreiche Handlung mit der zu benutzende Hardware benötigt wird. Somit umfasst Software alle Produkte und Leistungen, die benötigt werden. Des Weiteren wird Software immer wieder als „immaterielles Gut“ (Kaufmann & Müller,2022) beschrieben, da sie den Anwender bei seinen Aufgaben und dessen Durchführung unterstützt. Weitere Merkmale von Software sind, dass Software mit gleichbleibender Qualität kopiert werden kann, nicht abgenutzt und mehrfach verwendet werden kann. Gleichzeitig muss die Software immer wieder auf den neusten Stand gebracht werden, um weiterhin nützlich zu bleiben und nicht unbrauchbar zu werden.²

4.3 Systemsoftware

Bei einer Systemsoftware handelt es sich um ein Steuerungsinstrument der Hardware. Diese beinhalten die Betriebssysteme, Datenbanksysteme und andere Softwareentwicklungstools. Heutzutage wird

¹ Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg. S.28

² Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg. S.271 f.

solche Software in der Regel von Softwareunternehmen bereitgestellt.³ Des Weiteren ist in der Systemsoftware oft ein weiterer Bestandteil enthalten, welcher sich Middleware nennt und für die Kommunikation mit Systemen verantwortlich ist (Marwedel, 2021).

4.3.1 Betriebssysteme

Betriebssysteme steuern die zu benutzenden Daten zwischen Anwendungssoftware und Hardwarekomponenten. Es sorgt somit dafür, dass verschiedene Anwendungen ausgeführt werden können und übernimmt dabei die Aufgaben wie die Überwachung der Bereiche der Benutzerschnittstelle, die Auftragsverwaltung, die Arbeitsspeicherverwaltung, Dateiverwaltung und der generellen Benutzerverwaltung. Beispiele für solche Betriebssysteme sind Windows, MacOS oder Linux. Diese Systeme unterscheiden sich auf ihre eigenen Arten und Weisen und bieten je nach Anfrage andere Darstellungen und Anwendermöglichkeiten.⁴

4.4 Anwendungssoftware

Die Anwendungssoftware wird zur Benutzung und Bearbeitung von verschiedenen Aufgaben verwendet. Dabei lässt sich Software in verschiedene Kategorien unterteilen. Die sogenannte betriebswirtschaftliche Software unterstützt, wie der Name bereits aussagt, den Betrieb in ihren geschäftlichen Tätigkeiten. Ein weiterer Teilbereich ist die technische Software, die bei mathematischen oder anderen strukturellen Problemen unterstützend wirkt. Zu guter Letzt gibt es den Bereich der Collaboration-Software, welche den Teil der Kommunikation übernimmt. Auch Zusammenarbeit an bestimmten Projekten und ihre jeweilige Dokumentation fallen in diese Kategorie. Diese verschiedenen Teilbereiche sind unterschiedlich je nachdem, in welchem Tätigkeitsbereich und Geschäftszweig das Unternehmen tätig ist.⁵

4.4.1 Individualsoftware

Bei einer individuellen Software handelt es sich um eine Software, die explizit für ein Unternehmen erstellt wird. Dabei wird oft eine Lücke von fehlender Software geschlossen. Diese Software ist dabei nicht im Standardsoftwarepaket enthalten. Ein weiterer Grund der Erschaffung ist die Generierung eines Wettbewerbsvorteiles gegenüber anderen Firmen. Dabei muss die Entscheidung getroffen

³ Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg. S.273

⁴ Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg. S.274 ff.

⁵ Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg. S.273 f.

werden, ob die eigenständig erstellte Software von den eigenen IT-Mitarbeitern erstellt oder von einer externen Firma beauftragt wird. Diese Entscheidung wird auf Grundlage des Schwierigkeitsgrads und der Anforderungen der Software getroffen.⁶

4.4.2 Standardsoftware

Im Gegensatz zur Individualsoftware, wird bei der Standardsoftware ein bereits existierendes Produkt gekauft oder gebucht. Dabei wird spezifisch eine Anwendung für einen bestimmten Arbeitsbereich gesucht. Dabei profitiert die Software von Erfahrungen und Problemen anderer Unternehmen und entwickelt sich so stets weiter. Die Einführung von Standardsoftware ist dabei meist schneller, einfacher und planbarer. Auch die Service- und Beratungskosten sind meist bereits im Voraus abzusehen. Standardsoftware ist aber auch in verschiedenen Geschäftsbereichen kombinierbar und sorgt im Unternehmen dafür, dass ein gemeinsamer Datenpool benutzt wird. Wenn dies der Fall ist, wird diese Software als integrierte Standardsoftware bezeichnet.⁷

4.4.3 Qualitätsmerkmale von Software

Damit eine Software die gewünschte Qualität erreicht, gibt es verschiedene Kriterien die einen Einfluss auf diese haben. Die Merkmale sind verschieden und sorgen insgesamt für die allgemeine Qualität, obwohl diese Merkmale sich eventuell widersprechen könnten. Das erste zentrale Merkmal ist die Funktionalität. Erfüllt das System die Anforderungen, die die Person an die Software stellt oder ergeben sich bestimmte Fehler in der Software. Ein weiteres Merkmal ist die Laufzeit, denn es ist wichtig, wie schnell und flüssig die Software arbeitet. Vor allem liegt der Fokus bei Echtzeitsystemen, welche erhebliche Zeitüberschreitungen beinhalten sollten. Der dritte wichtige Punkt ist die Zuverlässigkeit. Dabei soll die Software immer eine stabile Verbindung aufweisen und auch fehlerfrei funktionieren. Dieses Kriterium ist aber aufgrund seiner Eigenschaften oft abhängig von anderen Qualitätsmerkmalen. Des Weiteren wird die Softwarequalität an der Benutzbarkeit gemessen. Hierbei soll eine einfache Bedienung für den Nutzer zur Verfügung stehen, damit der Arbeitsablauf so einfach wie möglich ist. Software sollte zudem auch wartbar sein, denn Fehler und andere neue Funktionen sollten nachträglich anpassbar sein. Bei Fehlern oder Problemen sollte es zudem möglich sein, die Software im schlimmsten Fall zu reparieren. Ein Problem, welches immer wieder auftritt, ist, dass Software oft mit der Zeit unübersichtlich wird und große Mengen an Daten beinhaltet. Bei guter Software ist es wichtig, dass die Transparenz hoch ist. Das bedeutet, dass der Quellcode leicht einsehbar und übersichtlich ist. Das vorletzte Qualitätsmerkmal ist, die Übertragbarkeit der Software.

⁶ Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg. S.279

⁷ Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg. S.279 f.

Dies soll dazu führen, dass bei einem Wechsel der Plattform oder anderen Veränderungen die Software immer noch genauso funktioniert wie davor. Als letztes gibt es die Testbarkeit, denn um Fehler zu entdecken oder gewisse Funktionen zu testen, muss eine Software variabel und einfach genug testbar sein.⁸

4.5 Reifegradmodell

Ein Reifegradmodell dient dazu, den Entwicklungsstand eines Prozesses, einer Prozessgruppe oder einer Organisation systematisch zu bewerten. Dabei wird die Einheit, die bewertet wird in verschiedene Reifegradstufen eingeordnet. Diese Stufen starten bei einem niedrigem Startniveau, bei welchem die Einheit noch nicht richtig integriert ist und nur nebenher existiert. Ziel ist es die höchste Stufe zu erreichen, um den optimalen Reifegrad zu erreichen. Dies ist nur durch kontinuierliche Verbesserung möglich. Diese verschiedenen Reifegradstufen sind dabei klar definiert und zeigen in jeder Stufe an, wie weit der Prozess oder die Organisation entwickelt ist. Um den derzeitigen Reifegrad zu ermitteln, ist eine Bewertung des Systems, Prozesses oder Organisation benötigt. Dies erfolgt durch ein Bewertungsverfahren, welches meist aus einer Checkliste oder Fragebögen besteht. Aufgrund dieser Bewertungen kann dann der aktuelle Stand abgerufen werden und weitere Verbesserungsmöglichkeiten geplant werden.⁹

4.5.1 Übersicht verschiedener Reifegradmodelle

Im Folgenden werden verschiedene bereits existierende Reifegradmodell kurz vorgestellt und beleuchtet. Der Schwerpunkt liegt auf den Modellen Capability Maturity Model, Control Objectives for Information and Related Technology, ISO/IEC 15504, Business Process Maturity Model und zum Schluss das Capability Maturity Model Integration.

4.5.1.1 Capability Maturity Model (CMM)

Das sogenannte Capability Maturity Model (CMM) besteht darin, bewerten zu können, wie gut Unternehmen Software entwickeln. Es bietet die Grundlage für weitere Reifegradmodelle. Dabei beschreibt das Modell insgesamt fünf Entwicklungsstufen. In Stufe 1 befindet man sich im Anfangszustand, dabei arbeitet jeder auf seine Weise. Es gibt somit keine Regeln oder festen Abläufe. In der nächsten Stufe sind aber bereits erste Standards festgelegt und werden wiederholt in den Abläufen verwendet. In Stufe 3 findet man klar definierte Prozesse, die klar strukturiert und bewertbar sind. In

⁸ Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag. S. 6 ff.

⁹ Mangiapane, M & Büchler, R. P. (2024). Modernes IT-Management: Methodische Kombination von IT-Strategie und IT-Reifegradmodell. Springer-Verlag. S.13 f.

der vorletzten Stufe werden Vergleiche zu anderen herausgearbeitet und verbessert. In dieser Stufe werden Best Practice Ansätze zur Optimierung verwendet. Zu guter Letzt ist in der fünften Stufe der Bereich der ständigen Entwicklung und Optimierung. Das Wichtigste an diesem Reifegradmodell ist nicht die Bewertung an sich, sondern die individuelle Weiterentwicklung der Organisation. Dabei ist es wichtig aufzuzeigen, dass die Ziele durch die mithilfe des Reifegradmodells besser erreicht werden können.¹⁰

4.5.1.2 Control Objectives for Information and Related Technology

Das Modell Control Objectives for Information and Related Technology (COBIT) unterstützt Unternehmen dabei, ihre IT an den Unternehmenszielen auszurichten und dabei eine Verbesserung für das Unternehmen zu schaffen. Dabei wird der Top-Down-Ansatz verwendet. Das bedeutet, dass die Ziele der IT so erstellt werden, dass sie den Zielen des Unternehmens entsprechen. Dabei wird die IT-Architektur so angepasst, dass die Ziele vereint werden. COBIT beschreibt insgesamt was vorhanden sein muss und nicht genau wie das Unternehmen die Ziele umsetzen soll. Es dient somit als Strukturierung der ganzen IT-Architektur und deren IT-Prozesse. Zur Bewertung dieser Prozesse erfolgt eine weitreichende Analyse der Kennzahlen und Größen. Dabei wird mit einem Bottom-Up-Prinzip gearbeitet, um eine ständige Verbesserung herbeizuführen. Diese Veränderungen werden im Modell in verschiedene Ziele untergliedert und in den vier Bereichen Planung und Organisation, Beschaffung und Implementierung, Betrieb und Support sowie Überwachung und Kontrolle eingeteilt und bearbeitet.¹¹

4.5.1.3 ISO/IEC 15504

ISO/IEC 15504, auch durch den Namen SPICE bekannt, ist ein Modell zur Bewertung von Prozessen. Dabei liegt der Fokus hauptsächlich bei der Software- und Systementwicklung. Das Ziel des Modells ist es wie bei den anderen Modellen eine Verbesserung zu schaffen. Dies erfolgt durch die Bewertung der Fähigkeiten und Reife der jeweiligen Prozesse stattfinden. Hierbei entsteht die Eingliederung in verschiedene Reifegrade, welche den bereits genannten Reifegraden anderes Modell ähneln. Beim ersten Grad handelt es sich um den Performed Process, also dem durchgeführten Prozess. Darunter versteht man einen Prozess, der ausgeführt werden kann und auch sein Ziel erreicht, jedoch nicht weiter gesteuert und erfasst werden. Im nächsten Grad dem Managed Process, also dem gesteuerten Prozess, wird der Prozess geplant, überwacht und gelenkt. Dabei erhält man ein nachvollziehbares

¹⁰ Haghsheno, S., Lennerts, K. & Gentes, S. (2019). 30.BBB-Assistententreffen in Karlsruhe – Fachkongress der wissenschaftlichen Mitarbeiter Bauwirtschaft | Baubetrieb | Bauverfahrenstechnik: 10. -12. Juli 2019, Institut für technologie und Management im Baubetrieb (TMB), Karlsruher Institut für Technologie (KIT). KIT Scientific Publishing. S. 73 ff.

¹¹ Mangiapane, M & Büchler, R. P. (2024). Modernes IT-Management: Methodische Kombination von IT-Strategie und IT-Reifegradmodell. Springer-Verlag. S.16 f.

Ergebnis. Im dritten Level dem Established Process ist die Rede von einem standardisierten Prozess, der immer wieder gleich ablaufend ausgeführt wird und somit immer wieder regelmäßig verwendet wird. Bei der vorletzten Stufe Predictable Process, kann mit Hilfe von gemessenen und überwachten Daten ein Prozess vorhergesagt werden. Somit sind Details zur Leistung und Qualität stets steuerbar und anpassbar. Im letzten Grad des Modells dem Optimizing Process findet man eine kontinuierliche Verbesserung der Prozesse, um eine stetige Weiterbildung und Verbesserung zu erreichen.¹²

4.5.1.4 Business Process Maturity Model

Das Business Process Maturity Model (BPMM) ist ein Reifegradmodell, welches bestimmt, wie gut ein Unternehmen seine Geschäftsprozesse unter Kontrolle hat und inwieweit das Unternehmen sich ständig verbessert. Es dient als Leitfaden, um angestrebte Verbesserungen zu identifizieren und den aktuellen Entwicklungsstand zu bewerten. Anders als die anderen Modelle bezieht sich das BPMM auch auf die Strategie, Kundenbeziehungen und Organisation im Unternehmen. Dabei wird sich vor allem an vier verschiedenen Punkten orientiert. Diese vier Punkte geben das sogenannte IMCO-Modell. Der erste Punkt an dem sich orientiert wird ist der Input, also welche Ressourcen und Informationen in den Prozess verwendet werden. Des Weiteren wird geschaut, wie die Informationen verarbeitet werden. Genauer gesagt welche Geräte, Abläufe oder Personal daran beteiligt sind. Zudem wird geprüft, wie man den Prozess verfolgt oder die Qualität sicherstellt. Zum Schluss wird genau betrachtet, was das Endprodukt ergibt, das heißt ob zum Beispiel eine Dienstleistung oder ein Produkt erstellt wird. Wie die anderen Modelle verfolgt das BPMM auch die fünf Reifegrade, um diese gewissen Beobachtungen und Abläufe zu bewerten. Die fünf Reifegrade sind aufgeteilt in Grad eins mit dem unorganisierten Prozess, Grad zwei mit dem gesteuerten Prozess, Grad drei mit dem definierten Prozess, Grad vier mit dem quantitativ gesteuerten Prozess und zu guter Letzt dem Grad des optimierten Prozesses.¹³

4.5.1.5 Capability Maturity Model Integration

Das Capability Maturity Model Integration (CMMI) ist ein Reifegradmodell, welches die verschiedenen Reifegradmodelle zu einem einheitlichen Standard vereinen will. Zuvor führten die verschiedenen Modelle zu Redundanz und unterschiedlichen Anforderungen im Unternehmen. Die Lösung ist ein flexibles Modell, welches alle Strukturen vereint. Das CMMI ist dabei anders als andere Schemas und ist nicht nur für Software geeignet, sondern auch für andere Gebiete wie zum Beispiel Systemintegrationen oder Lieferantenmanagement. Das Modell enthält dabei nicht nur das Stufenmodell, sondern ergänzt das Ganze mit einem kontinuierlichen Modell. Dies ermöglicht eine gezieltere

¹² Rout, T.P. (2003), ISO/IEC 15504—Evolution to an international standard. *Softw. Process: Improve. Pract.*, 8: 27-40.

¹³ Chang, K., Wang, W., Chen, L., Ellis, C., Hsu, C., Tsoi, A., Wang, H. (2007). *Advances in Web and Network Technologies and Information Management: APWeb/WAIM 2007 International Workshops: DBMAN 2007, WebETrends 2007, PAIS 2007 and ASWAN 2007*. Springer-Verlag. S.384 ff.

Verbesserung der einzelnen Abläufe. Die Reifegrade also das Stufenschema entspricht dabei den anderen Reifegradmodellen. Nun kommt zusätzlich ein kontinuierliches Schema hinzu. Dieses Schema besteht aus einer Skala von 0-5. Null steht für Incomplete, eins für Performed, zwei für Managed, drei für Defined, vier für Quantitatively Managed und fünf für Optimizing. Um den Aufbau zu erweitern, wird das CMMI in 22 Bereiche eingeteilt mit den Kategorien Process Management, Project Management, Engineering und Support. Um die endgültige Bewertung zu vollziehen, wird das sogenannte Standard CMMI Assessment Method for Process Improvement (SCAMPI) benötigt.¹⁴

5 Vergleich und Analyse von Reifegradmodellen

Im Folgenden werden zwei verschiedene Reifegradmodelle genauer analysiert und vorgestellt. Anschließend erfolgt ein Vergleich der zwei Reifegradmodelle, bevor weitere Details zur Erstellung des eigenen Reifegradsystems erarbeitet werden.

5.1 CMMI

Das Capability Maturity Model Integration (CMMI) ist ein Reifegradmodell zur Verbesserung von Prozessen und Organisationen. Es wird vor allem in der Software- und Systementwicklung verwendet. Entwickelt wurde es vom Software Engineering Institute (SEI) an der Carnegie Mellon University. Der ursprüngliche Auftrag kam dabei vom US-Verteidigungsministerium. CMMI wurde entwickelt, weil es bereits eine Vielzahl an Reifegradmodellen gibt, die ungefähr dasselbe Ziel verfolgen. Um diese Modelle zu vereinen, entstand ein einheitliches und übergreifendes Reifegradmodell. Das Ziel dieses Modells ist es, Redundanzen zu vermeiden und den Unternehmen ein strukturierter Rahmen zur Prozessverbesserung bereitzustellen.¹⁵

CMMI soll dabei kein konkreter Prozess oder eine Vorschrift sein, sondern eine strukturierte Orientierung zur Bewertung und Verbesserung der organisatorischen Möglichkeiten bieten. Im Grunde dient es der Entdeckung von Schwächen vor allem im Bereich der Prozesse. Nach dem Identifizieren der Schwächen können als Resultat bestimmte Maßnahmen gestartet werden, um die Qualität im Unternehmen oder im Prozess zu verbessern. Die allgemeinen Vorteile dieses Modells sind zum einen die einheitliche Sprache und Struktur der Prozessumgebung und zum anderen eine verbesserte Kommunikation und Entscheidungsfindung in bestimmten Bereichen im Unternehmen. Des Weiteren können einfacher Schwächen und Unsicherheiten entdeckt werden und durch bessere klar strukturiertere Standards ersetzt werden. Mehr oder weniger ist ein zentraler Aspekt auch das Risikomanagement. Dabei hilft das Modell CMMI der Organisation, Risiken vorzeitig zu erkennen, zu

¹⁴ Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag. S. 530 ff.

¹⁵ Chcomley. (o. D.). Integration des Capability Maturity Model (CMMI), Hintergrundnotizen - Azure Boards. Microsoft Learn. <https://learn.microsoft.com/de-de/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops> Stand: 04.08.2025

bewerten und auch zu handeln. Dies sorgt dafür, dass auch bei kurzfristigen Problemen Systeme oder Prozesse weiterhin stabil agieren könnten.¹⁶

Wie bereits im Bereich zur Definition erwähnt, gibt es zwei Bewertungsansätze im Bereich des CMMI-Modells. Der erste Bereich ist das Stufenmodell, auch Staged Representation genannt (siehe Abbildung 2). Dabei wird bei der Analyse ein von fünf Reifegraden, auch Maturity Levels genannt, zugeordnet (siehe Abbildung 1 Reifegrade des Modells CMMI).¹⁷

Characteristics of the Maturity levels

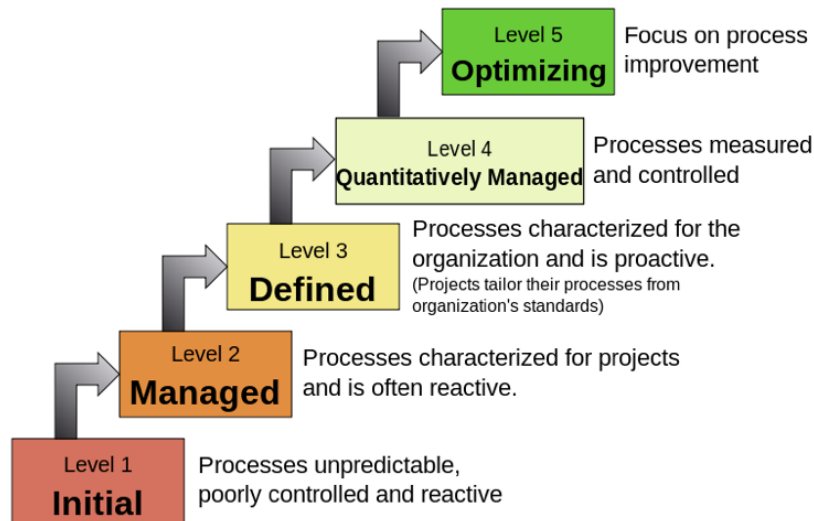


Abbildung 1: Reifegradstufen des Modells CMMI

Die fünf Reifegrade sind zum einen Initial, was chaotisch bedeutet. Die Stufe Managed, dabei wird der Begriff gesteuert oft verwendet. Als nächstes gibt es die Stufen Defined und darauffolgend Quantitatively Managed. Zu guter Letzt gibt es die Stufe Optimizing, was aussagen soll, dass eine kontinuierliche Verbesserung stattfindet. Neben dem Stufenmodell gibt es das sogenannte kontinuierliche Modell auch Continuous Representation genannt (siehe Abbildung 2). Dieses Modell bewertet dabei jeden einzelnen Prozessbereich auf einer Skala von null bis fünf. Es eignet sich vor allem für einzelne Prozesse, da diese individuell verbessert und priorisiert werden können. Die Bewertung dieser Bereiche ähnelt denen des Stufenmodells und werden Capability Levels genannt. Die Stufen erhalten neben den schon bekannten Stufen Optimizing, Quantitatively Managed, Defined und Managed auch noch zwei weitere Stufen darunter. Dabei ersetzt die Stufe Performed den Reifegrad Initial und bedeutet, dass der Prozess läuft, jedoch noch nicht ausgereift ist. Die zusätzliche Stufe, unter den anderen Stufen, ist die sogenannte Incomplete Stufe, dabei handelt es sich um nicht funktionierende Prozesse, die nicht vollständig sind. Insgesamt lässt sich sagen, dass das Stufenmodell für Führungskräfte leicht

¹⁶ Chcomley. (o. D.). Integration des Capability Maturity Model (CMMI), Hintergrundnotizen - Azure Boards. Microsoft Learn. <https://learn.microsoft.com/de-de/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops> Stand: 04.08.2025

¹⁷ Chcomley. (o. D.). Integration des Capability Maturity Model (CMMI), Hintergrundnotizen - Azure Boards. Microsoft Learn. <https://learn.microsoft.com/de-de/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops> Stand: 04.08.2025

verständlich ist und somit sehr gut geeignet für die allgemeine Einordnung ist. Im Gegensatz dazu ist das kontinuierliche Modell eher für individuellere und genauere Prozesse geeignet, sodass dort weitestgehend immer ein Verbesserungsprozess stattfindet.¹⁸

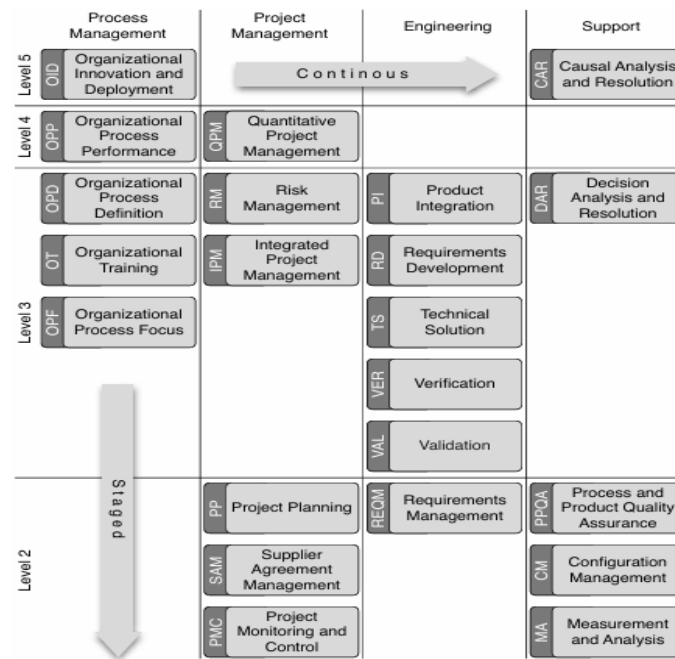


Abbildung 2: CMMI-Modell mit Prozessbereichen und Stufen

Das CMMI-Modell unterstützt eine Liste bestimmter Prozessbereiche (siehe Abbildung 2: einzelne Bereiche wie z.B.: RM = Risk Management). Diese Bereiche sind in vier verschiedenen Kategorien eingeteilt: Projektmanagement, Prozessmanagement, Engineering und Supportfunktionen (siehe Abbildung 2: obere Beschriftungen). Die einzelnen Prozessbereiche bestehen wiederum aus drei verschiedenen Arten von Bestandteilen. Der erste Bestandteil sind Erforderliche Komponenten. Dabei handelt es sich um die generischen und spezifischen Ziele eines Bereiches, diese müssen erfolgreich abgeschlossen werden um als erreicht anzusehen. Sie sind der Hauptkern des Modells. Die zweite Art ist erwartete Komponenten, diese sind dabei nicht verpflichtend, sondern es wird empfohlen diese zu verwenden. Im Allgemeinen handelt es sich hierbei um Methoden oder Ansätze, die genutzt werden, um das Ziel zu erreichen. Sie spiegelt mehr oder weniger den Leitfaden für die Bewertung und Umsetzung. Den Abschluss macht die informative Komponente. Diese sorgt, wie der Name schon sagt für Informationen, Beispiele und Ergebnisse. Sie helfen dabei Unternehmen bei der Umsetzung der Prozessverbesserung.¹⁹

¹⁸ Chcomley. (o. D.). Integration des Capability Maturity Model (CMMI), Hintergrundnotizen - Azure Boards. Microsoft Learn. <https://learn.microsoft.com/de-de/azure/devops/boards/work-items/guidance/cmmi/guidance-back-ground-to-cmmi?view=azure-devops> Stand: 04.08.2025

¹⁹ Chcomley. (o. D.). Integration des Capability Maturity Model (CMMI), Hintergrundnotizen - Azure Boards. Microsoft Learn. <https://learn.microsoft.com/de-de/azure/devops/boards/work-items/guidance/cmmi/guidance-back-ground-to-cmmi?view=azure-devops> Stand: 04.08.2025

Da die Bewertung des kontinuierlichen Modells bereichsweise erfolgt, kann sie auf das klassische Stufenmodell übertragen werden. Diesen Ablauf nennt man auch Equivalent Staging. Um dies besser veranschaulichen, folgt ein Beispiel. Um die Reifegradstufe drei zu erreichen, müssen alle Prozessbereiche der Stufen zwei und drei mindestens eine Bewertung von Stufe drei der Capability Levels erreichen. Für die Bewertung werden somit alle anderen Stufen nicht benötigt und erhalten das Stufenlevel null. Ist beispielsweise der Prozessbereich Risk Management, der sich in Level drei befindet, nicht mit dem Capability Level drei ausgestattet, fällt die ganze Organisation in die Reifestufe zwei zurück (siehe Abbildung 3).²⁰

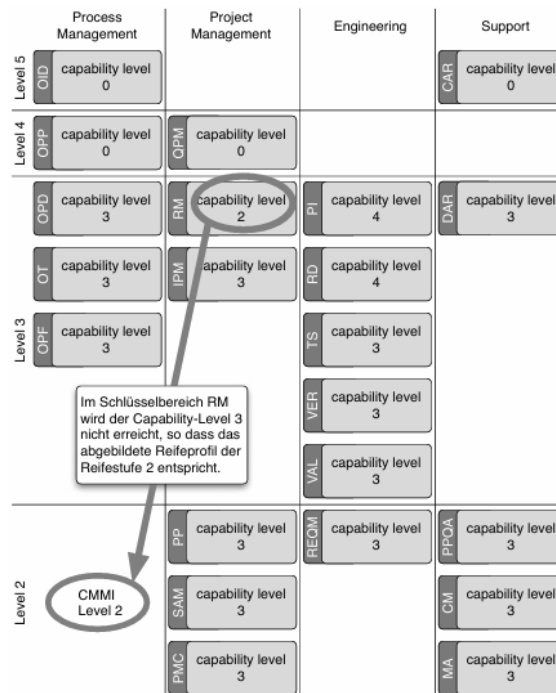


Abbildung 3: Beispiel für Equivalent Staging

5.2 SPICE

Das internationale Projekt SPICE, kurz für Software Process Improvement and Capability dEtermination, wurde in den 1990er-Jahren gestartet. Ziel war es, mithilfe der zwei Organisationen ISO (International Organization for Standardization) und IEC (International Electrotechnical Commission) einen internationalen Standard für die Beurteilung und Fortentwicklung von Softwareentwicklungsprozessen zu schaffen. Der erste Teil der Version entstand bereits Ende der 90er-Jahre und war unter dem Titel ISO/IEC 15504 bekannt. In den folgenden Jahren wurde diese Version immer wieder überarbeitet, wobei immer wieder neue Veröffentlichungen präsentiert werden. SPICE definiert generell keine festen Inhalte, sondern legt gängige Leistungsanforderungen fest, die jedes Bewertungsmodell oder Arbeitsweise erfüllen muss, um den Ansprüchen gerecht zu werden. Beispiele für so ein Modell ist die ISO/IEC 12207, dabei wird ein Software-Lebenszyklusprozess beschrieben. Ein weiteres

²⁰ Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag. S. 530 ff.

Modell ist die ISO/IEC 15288, welches System-Lebenszyklusprozesse beschreibt. Der Fokus liegt stets auf der Bewertung der eigenen Prozesse und weniger auf die Zertifizierung. Ziel ist wie von Anfang an definiert, eine strukturierte Beurteilung und Verbesserung der einzelnen Prozesse in einer Organisation.²¹

SPICE kann in zwei verschiedene Dimensionen unterteilt werden. Dabei wird zwischen den zwei verschiedenen Dimensionen Prozessdimension und Reifegraddimension unterschieden (siehe Abbildung 4).

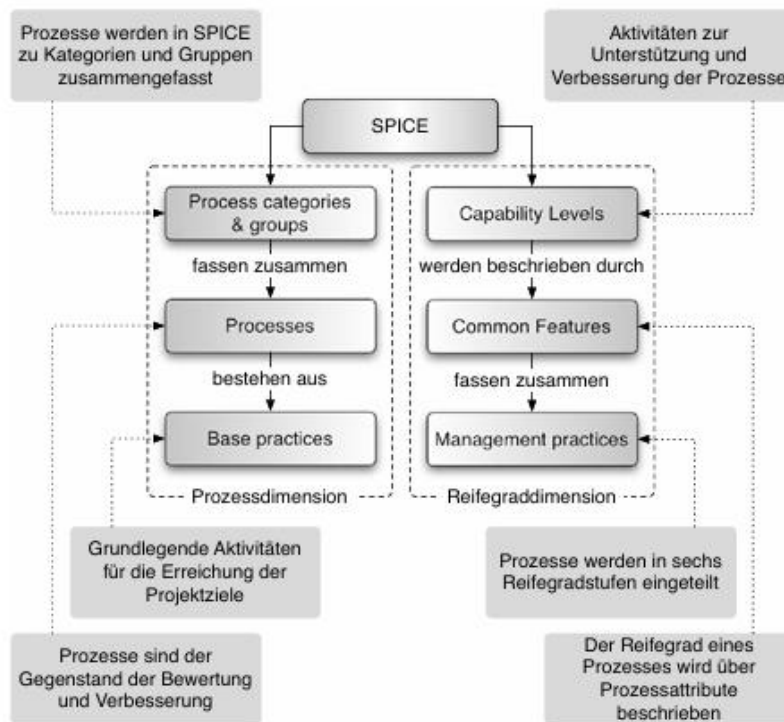


Abbildung 4: SPICE Dimensionsmodell

In der Abbildung ist ein Ablauf der einzelnen Dimensionen dargestellt. In der Prozessdimension wird zwischen den Schritten Process Categories & Groups, Processes und Base Practices unterschieden. Im Allgemeinen kann man sagen, dass die Prozessdimension von Base Practices handelt. Diese beinhalten alle Geschäftigkeit, die für das Erreichen der Ziele benötigt werden. Diese Base Practices setzen sich aus Prozessen zusammen, welche die Grundlage der Bewertung und Verbesserungen bilden. Anschließend werden diese Prozesse in verschiedenen Kategorien und Gruppen eingeteilt und zusammengefasst. Die zweite Dimension, die Reifegraddimension, beinhaltet die Management Practices, bei welchem die Prozesse in sechs Reifestufen eingeteilt werden. Diese werden von Common Features zusammengestellt, die alle benötigten Attribute für den Prozess beinhalten. Die Eigenschaften werden weiter von den Capability Levels umschrieben, welche die Hauptaufgabe haben Prozesse zu unterstützen und voranzutreiben.²²

²¹ Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag. S. 535 f.

²² Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag. S. 536 f.

In der Prozessdimension werden die verschiedenen Abläufe in verschiedene Kategorien unterteilt, welche erneut in unterschiedliche Gruppen und Einzelteile zerteilt werden. Die erste Kategorie bilden die Primary Life Cycle Processes, in welcher sich Prozesse befinden, die direkt bei der Projektumsetzung beteiligt sind. Beispiele hierfür sind zum Beispiel Einzelheiten von der Lieferung oder Beschaffung von Systemen (siehe Abbildung 5). Die nächste Kategorie ist Organizational Life Cycle Processes. Hierbei handelt es sich um unterstützende Managementprozesse. Beispiele hierfür sind die Prozessverbesserung oder die allgemeine Projektplanung (siehe Abbildung 6). Abschließend gibt es die Kategorie der Support Processes. In dieser Kategorie stehen Unterstützungsfunktionen im Vordergrund. Beispiele hierfür sind Maßnahmen zur Qualität oder die Erstellung von Dokumenten (siehe Abbildung 7).²³

Acquisition Process Group (ACQ)			
ACQ.1	Acquisition preparation	ACQ.4	Supplier monitoring
ACQ.2	Supplier selection	ACQ.5	Customer acceptance
ACQ.3	Contract agreement		
Supply Process Group (SPL)			
SPL.1	Supplier tendering	SPL.3	Product acceptance support
SPL.2	Product release		
Engineering Process Group (ENG)			
ENG.1	Requirements elicitation	ENG.7	Software integration
ENG.2	System requirements analysis	ENG.8	Software testing
ENG.3	System architectural design	ENG.9	System integration
ENG.4	Software requirements analysis	ENG.10	System testing
ENG.5	Software design	ENG.11	Software installation
ENG.6	Software construction	ENG.12	Software and system maintenance
Operation Process Group (OPE)			
OPE.1	Operational use	OPE.2	Customer support

Abbildung 5: Beispiele für Primary Life Cycle Processes

Management Process Group (MAN)			
MAN.1	Organizational alignment	MAN.4	Quality management
MAN.2	Organizational management	MAN.5	Risk management
MAN.3	Project management	MAN.6	Measurement
Process Improvement Process Group (PIM)			
PIM.1	Process establishment	PIM.3	Process improvement
PIM.2	Process assessment		
Resource and Infrastructure Process Group (RIN)			
RIN.1	Human resource management	RIN.3	Knowledge management
RIN.2	Training	RIN.4	Infrastructure
Reuse Process Group (REU)			
REU.1	Asset management	REU.2	Reuse program management
REU.3	Domain engineering		

Abbildung 6: Beispiele für Organizational Life Cycle Processes

Support Process Group (SUP)			
SUP.1	Quality assurance	SUP.6	Product evaluation
SUP.2	Verification	SUP.7	Documentation
SUP.3	Validation	SUP.8	Configuration management
SUP.4	Joint review	SUP.9	Problem resolution management
SUP.5	Audit	SUP.10	Change request management

Abbildung 7: Beispiel Support Processes

²³ Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag. S. 536 ff.

In der zweiten Dimension, der Reifegraddimension, wird ein sechsstufiges Reifegradmodell verwendet. Dieses Modell betrachtet nicht das ganze Unternehmen, sondern fokussiert sich auf die einzelnen Prozesse im Unternehmen.

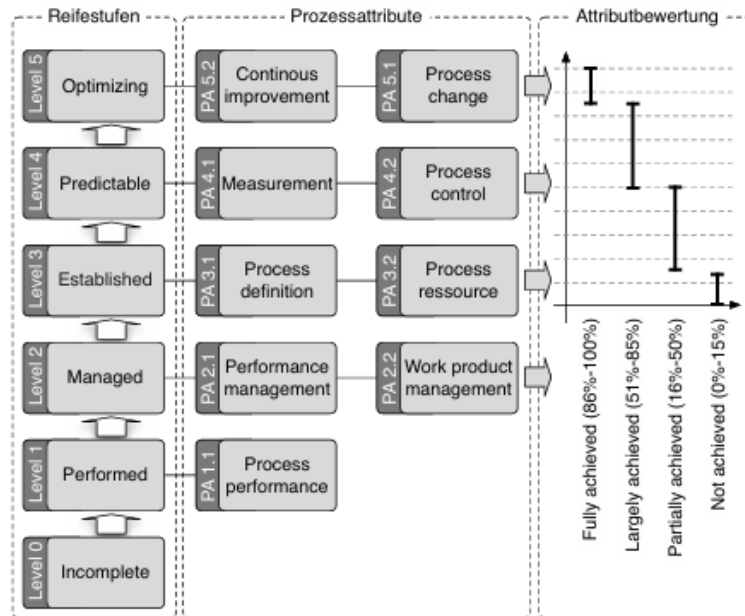


Abbildung 8: Reifegraddimension

Die Reifegraddimension des SPICE-Modells definiert sechs verschiedene Reifestufen (siehe Abbildung 8). Die unterste Stufe Incomplete gibt wieder, dass der Prozess nicht vorhanden oder noch nicht kontrolliert und eingeordnet ist. Auf Level 1, Performed, ist bereits ein Prozess vorhanden aber noch nicht wirklich gesteuert und festgehalten. Im nächsten Level, Managed, handelt es von geplanten, dokumentierten und gelenkten Prozessen. Im Level Established entsprechen die Prozesse bereits einem gewissen Standard und werden im ganzen Unternehmen verwendet. Im vierten Level, Predictable, sind Prozesse messbar, stabil und in gewisser Weise vorhersehbar. Auf der höchsten Stufe Optimizing werden die Prozesse immer wieder verbessert, um einen sehr guten Standard zu etablieren. Die zu bewerteten Prozessattribute werden nun bei einem Assessment beurteilt und ein Erfüllungsgrad in Prozent bestimmt. Dabei wird in vier verschiedene Kategorien unterteilt (siehe Abbildung 8). Bei der Erfüllung von 86%-100% ist ein Attribut vollständig erfüllt. Bei 51%-85% ist das Attribut weitestgehend erfüllt. Teilweise erfüllt ist ein Prozessattribut zwischen 16%-50%. Wenn der Prozentsatz unter 16% fällt, ist das Attribut nicht erfüllt. Diese Prozentwerte bilden die Grundlage für die spätere Einordnung in die verschiedenen Stufen.²⁴

²⁴ Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag. S. 539 ff.

5.3 Vergleich der zwei Reifegradmodelle

Im Vergleich der bereitgestellten Ergebnisse ergibt sich folgender Vergleich. Das CMMI bezieht sich auf die Vereinheitlichung verschiedener Reifegradmodelle, um eine generelle Verbesserung von Organisationsprozessen zu schaffen. Der Schwerpunkt liegt auf der Prozessverbesserung innerhalb des Unternehmens. Der Ansatz des CMMI kann somit als praxisorientiert und vereinheitlichend definiert werden. SPICE verfolgt das Ziel, einen internationalen Standard für die Bewertung und Verbesserung von einzelnen Softwareprozessen zu schaffen. Der Fokus liegt dabei auf der strukturellen Beurteilung und Weiterentwicklung von Prozessen, somit ergibt sich hierbei ein Framework, welches bewertungsorientiert und standardisierbar ist.

Des Weiteren unterscheiden sich die zwei Modelle in ihrer Struktur. CMMI nutzt zwei verschiedenen Modellvarianten, die Staged Representation und die Continuous Representation. Dabei haben die zwei Varianten verschiedenen Eignungsgebiete. Die Staged Representation ist eher für eine klare Gesamtbewertung und somit einfacher und für Führungskräfte geeignet. Die Continuous Representation ist eher für einzelnen Prozessanalysen geeignet und je nachdem individuell anpassbar. Während das CMMI mit zwei Modellvarianten arbeitet, enthält das Modell von SPICE zwei Dimensionen. Einmal die Prozessdimension, bei welcher die Prozesse in Kategorien und Gruppen eingeteilt werden. Die andere Dimension ist die Reifegraddimension mit ihren verschiedenen Reifegraden pro Prozess. Die endgültige Bewertung wird dann mit einem Erfüllungsgrad in Prozent angegeben und in verschiedenen Bewertungsstufen zugeordnet. Wenn man die beiden miteinander vergleicht, erkennt man, dass CMMI insgesamt flexibler wirkt durch seine zwei verschiedenen Modelle. Jedoch bietet das SPICE Modell einen standardisierten Ablaufplan zu genau seinem erdachten Zweck. Die Reifegradmodelle von CMMI und SPICE sind inhaltlich sehr ähnlich, unterscheiden sich jedoch in der Anwendung. Während CMMI sowohl einzelne Prozesse als auch die gesamte Organisation bewertet, steht bei SPICE der Fokus auf die Bewertung einzelner Prozesse.

Die beiden Modelle unterscheiden sich sowohl in den Komponenten als auch in den Bewertungskriterien. Während beim CMMI die Prozessbereiche in vier verschiedenen Kategorien unterteilt sind, wird beim SPICE Modell mit drei Hauptkategorien gearbeitet. Die einzige Kategorie, die bei beiden übereinstimmend ist, ist die Kategorie Support. Die weiteren drei Kategorien Projektmanagement, Prozessmanagement und Engineering des CMMI, werden im SPICE Modell in den zwei Kategorien Primary Life Cycle und Organisational Life Cycle zusammengefasst. Auch in der Bewertungsart gibt es Unterschiede. CMMI nutzt Capability Level und Reifegradstufen, während SPICE eine prozentbasierende Bewertung der Prozessattribute vorweist. Insgesamt verfolgt CMMI strukturierte Ziele und Methoden, während SPICE sich auf seine durchdachten Standards verlässt.

SPICE und CMMI unterscheiden sich auch hinsichtlich ihrer Zielgruppen. SPICE richtet sich an Unternehmen, die ISO-konforme Bewertungen und Verbesserungen in Softwareprozesse und Projekten durchführen möchten. Durch den ISO-Standard ergibt sich auch direkt eine Möglichkeit der internationalen Vergleichbarkeit, sowie eine strukturierte Prozessbewertung mit ISO-Konform. CMMI

bietet dagegen eine einheitliche Sprache, Risikomanagement und eine Schwächenanalyse mit eventuellen Verbesserungsvorschlägen.

Tabelle 1: Zusammenfassung Vergleich der Reifegradmodelle CMMI und SPICE

Aspekt	CMMI	SPICE
Ziel	Vereinheitlichung der Reifegradmodelle zur Prozessverbesserung in Organisationen	Entwicklung internationalen Standards zur Bewertung und Verbesserung einzelner Softwareprozesse
Fokus	Prozessverbesserung im Unternehmen	Standardisierte Beurteilung und Verbesserung einzelner Prozesse
Modellstruktur	Zwei Modelle: Staged Representation Continuous Representation	Zwei Stufen mit Dimensionen: Prozessdimension Reifegraddimension
Bewertungsansatz	Stufenmodell Capability Levels	Prozentbasierte Bewertung Erfüllungsstufen
Bewertungsfokus	Einzelne Prozesse und gesamte Organisation	Einzelne Prozesse
Kategorien	Projektmanagement Prozessmanagement Engineering Support	Primary Life Cycle Organizational Life Cycle Support
Zielgruppen	Unternehmen unterschiedlicherer Bereiche, welche den Fokus auf Prozessverbesserungen legen	Unternehmen die ISO-konforme Bewertungen und Verbesserungen in Softwareprozesse und Projekten erreichen wollen
Nutzen	einheitliche Sprache Risikomanagement Schwächenanalyse	internationalen Vergleichbarkeit strukturierte Prozessbewertung mit ISO-Konform

Insgesamt kann man sagen, dass CMMI eher geeignet ist für die Unterstützung einer gesamten Organisation. Dabei liegt der Fokus auf der Verbesserung der allgemeinen Prozesslandschaft mit dem Fokus auf Risikominimierung und Management der Prozesse. SPICE hingegen ist geeignet für detaillierte Prozesse. Durch den ISO-Standard erhalten die einzelnen Prozesse auch international einen Standard, der anerkannt wird.

6 Vorgehen bei der Entwicklung des Reifegradmodells

Um den ersten Schritt zur Entwicklung eines Reifemodells zu gehen, wird das Prinzip von einem Vorgehensmodell inspiriert. Dieses Vorgehensmodell gliedert sich im Allgemeinen in acht verschiedene Phasen, die auf die Anforderungen sowie bewährte Vorgehensweisen aus der Praxis zurückgreifen (siehe Abbildung 9). Die zentrale Orientierung bei der Entwicklung wurde dabei von anderen Reifegradmodellen wie CMMI übernommen.²⁵

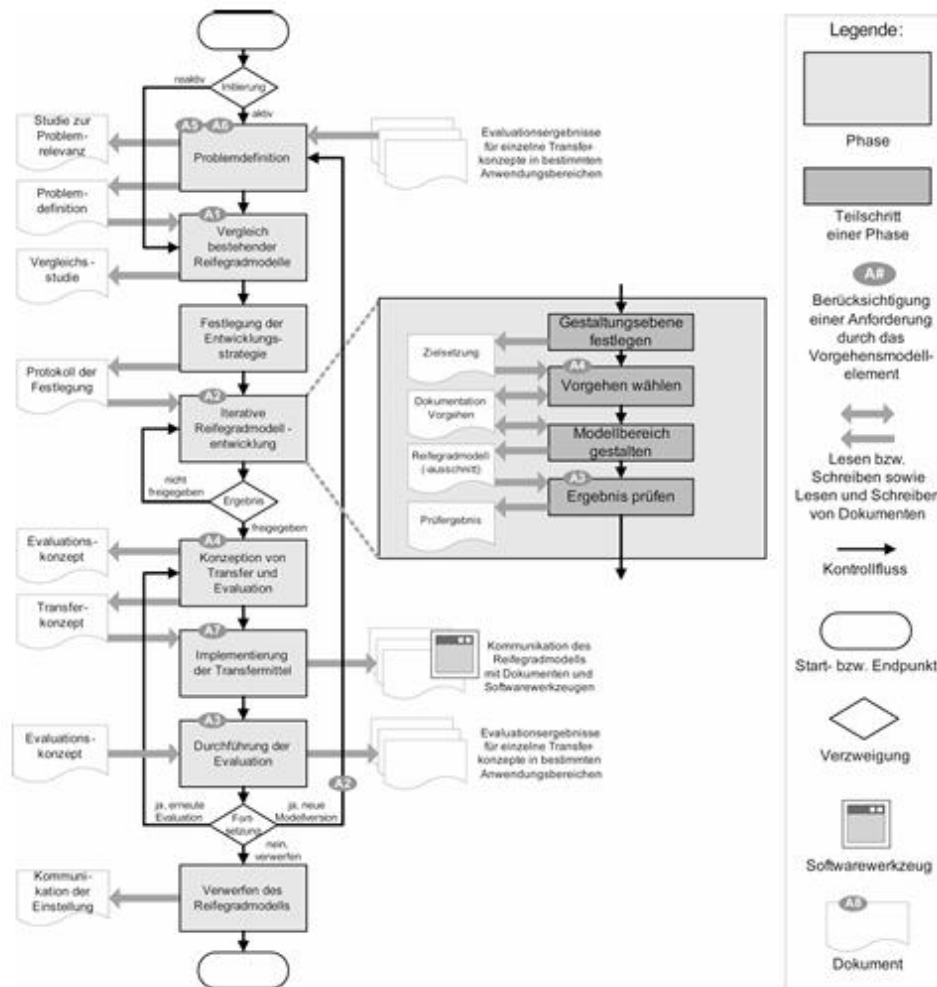


Abbildung 9: Vorgehensmodell Reifegradmodellentwicklung

Bei der ersten Phase Problemdefinition handelt es sich um die Definition des zu lösenden Problems. Dabei soll der Anwendungsbereich, sowie die Zielgruppen des Modells festgelegt und definiert werden. Danach folgt der Modellvergleich, bei welchem bereits existierende Reifegradmodelle analysiert und verglichen werden. Ziel ist es, Stärken und Schwächen der Modelle herauszuarbeiten, um eine Grundlage für die Auswahl der Entwicklungsstrategie zu schaffen. Basierend auf der Analyse wird entschieden, ob eine vollständige Neuentwicklung, eine Weiterentwicklung eines bestehenden

²⁵ Becker, J., Knackstedt, R. & Pöppelbuß, J. Entwicklung von Reifegradmodellen für das IT-Management. *Wirtsch. Inform.* 51, 249–260 (2009). <https://doi.org/10.1007/s11576-009-0167-9>

Modells oder eine Kombination von Modellen durchgeführt wird. Dabei ist festzuhalten, welches der verschiedenen Ansätze man wählt. In der vierten Phase Iterative Reifegradmodellentwicklung werden die einzelnen kleineren Phasen Gestaltungsebene festlegen, Vorgehen wählen, Modellbereich gestalten und Ergebnis prüfen durchlaufen. Dabei kann die Architektur des Modells eindimensional oder auch multidimensional sein. Am Ende jeder iterativen Phase wird überprüft, ob das Modell vollständig, geeignet und anpassbar ist. Nach erfolgreichem Test wird der Entwicklungsprozess weitergeführt. Anschließend steht dann eine Entwicklung zur Bewertung an. Dabei handelt es sich um Checklisten, Handbücher oder andere Tools. In dieser Phase sollten bereits Gedanken zu möglichen Bewertungsverfahren stattfinden. Danach werden im Schritt der Implementierung der Transfermittel das Modell und seine weiteren Dateien weitergegeben und bereitgestellt. Wichtig ist hierbei, dass die zugestellten Gegenstände auf die jeweiligen Gruppen abgestimmt sind. Kurz vor dem Schluss werden in der Phase der Durchführung der Evaluation das Modell auf die Probe gestellt und geprüft, ob es den Anforderungen entspricht. Dabei kann die Evaluation dafür sorgen, dass das Modell weiterentwickelt oder dass das Modell verworfen werden muss. In der letzten Phase Modellweiterentwicklung und Lebenszyklus soll das Reifegradmodell in naher Zukunft weiter überprüft und weiterentwickelt werden. Dabei sind neue Einflüsse von Mitarbeitenden und anderen Modellen eine große Priorität, um langfristig das Modell auf einem gewissen Standard zu halten.²⁶

7 Entwicklung des Reifegradmodells

Im Folgenden wird die Entwicklung eines neuen Reifegradmodells nach der zuvor erklärten Vorgehensweise schrittweise dargestellt. Anschließend wird eine Evaluation stattfinden, welche zusammen mit einem Unternehmen ausgeführt wird. Die Problemstellung und andere Informationen basieren auf dem Wissen und Erfahrungen, die bei der Zusammenarbeit mit dem Unternehmen gewonnen wurden.

7.1 Unternehmensvorstellung

Vor der Entwicklung des Reifegradmodells wird das Unternehmen, die Hermann Bantleon GmbH, mit welchem ich die Umsetzung der Reifegradmodelle entwickle, näher vorgestellt. Die Hermann Bantleon GmbH wurde im Jahr 1918 gegründet und ist seitdem in Ulm ansässig. Das Unternehmen beschäftigt derzeit etwa 277 Mitarbeitende, die sich auf 17 verschiedenen Fachbereiche verteilen. Mit dem Leitspruch „Qualität fängt bei mir an!“ verfolgt die Hermann Bantleon GmbH einen hohen Anspruch an die Qualität ihrer Produkte und Dienstleistungen. Das Unternehmen handelt als Komplettanbieter in der metallverarbeitenden Industrie und deckt dort Großteile der Prozessketten ab. Als

²⁶ Becker, J., Knackstedt, R. & Pöppelbuß, J. Entwicklung von Reifegradmodellen für das IT-Management. *Wirtsch. Inform.* 51, 249–260 (2009). <https://doi.org/10.1007/s11576-009-0167-9>

Schmierstoffspezialist bietet das Unternehmen sowohl standardisierte Produkte als auch individuell auf Kundenanforderungen zugeschnittene Produkte. Damit stellt das Unternehmen sicher, dass qualitativ hochwertige Lösungen für jeden Anwender gefunden werden.²⁷

In der Anfangszeit des Unternehmens konzentrierte Hermann Bantleon sich auf die Produktion von Leder-, Huf- und Wagenfette sowie Speiseöl und Waschpulver. Im Laufe der Zeit veränderten sich die Schwerpunkte des Unternehmens. Heute liegt der Schwerpunkt auf Schmierstoffe, Reinigung und Korrosionsschutz. Seit 1950 ist die Hermann Bantleon GmbH Teil der AVIA-Gruppe und zählt damit zu den 35 Gesellschaftern dieser Organisation. Aktuell ist Bantleon an drei Standorten vertreten: Blaubeurer Straße in Ulm, Ulm Nord und Crimmitschau. Diese Standorte beliefern Kunden in über 40 Ländern weltweit.²⁸

Die Produkte der Hermann Bantleon GmbH finden in einer Vielzahl von Branchen Anwendung. In der Metallbearbeitung tragen die eingesetzten Kühlschmierstoffe wesentlich zur Reduzierung von Reibung, Verschleiß und Temperatur bei, was zur Prozessoptimierung führt.²⁹ Auch in der Maschinen- und Anlagentechnik sowie in der Luftfahrtindustrie sind Schmierstoffe in ständiger Benutzung, um Reibung, Verschleiß und Korrosion zu minimieren, wodurch die Effizienz und Langlebigkeit der Maschinen gegeben werden.³⁰ ³¹Ein weiterer wichtiger Anwendungsbereich ist die Medizintechnik, wo bestimmte Anforderungen an das Material gegeben sein müssen. Hier müssen Schmierstoffe rückstandsfreie Reinigung ermöglichen, sowie auch humanverträglich und biokompatibel sein, um den hohen Standards gerecht zu werden.³² Im Bereich der Schwerindustrie (Heavy Machinery) spielen Schmierstoffe ebenfalls eine zentrale Rolle. Neben Motoren- und Hydraulikölen sind auch Schmierstoffe für die Fahrzeuge von großer Bedeutung. Besonders im Fokus stehen hierbei die Bau- und Recyclingwirtschaft, Land- und Forstwirtschaft sowie der alpine Bereich. Dort kommen Schmierfett, Motorenöle sowie Forstschutzmittel zum Einsatz.³³

Auf diese Weise unterstützt die Hermann Bantleon GmbH verschiedenste Branchen durch ihre Schmierstofflösungen, die zur Optimierung zahlreicher Prozesse beitragen. Dabei werden in der IT und anderen Bereichen verschiedene Lösungsansätze gebraucht, um schnell alles bereitzustellen. Dabei kommen viele verschiedene Software und Systeme zum Einsatz. Durch die Überflutung an verschiedenen Systemen entstand der Grundgedanke für das Thema der Bachelorarbeit.

7.2 Problemdefinition

Durch die zunehmende Anzahl von Systemen und Software in den Unternehmen kann es dazu führen, dass Software oder Systeme unzureichend funktionieren und nicht ersetzt werden. Zudem besteht die

²⁷ BANTLEON – Über uns. <https://bantleon.de/de/unternehmen/ueber-uns/>

²⁸ BANTLEON – Geschichte. <https://bantleon.de/de/unternehmen/geschichte/>

²⁹ BANTLEON – Metallbearbeitung. <https://bantleon.de/de/branchen/metallbearbeitung/>

³⁰ BANTLEON – Maschinen – und Anlagentechnik. <https://bantleon.de/de/branchen/maschinen-anlagentechnik/>

³¹ BANTLEON – Luftfahrt. <https://bantleon.de/de/branchen/luftfahrt/>

³² BANTLEON – Medizintechnik. <https://bantleon.de/de/branchen/medizintechnik/>

³³ BANTLEON – Branchen im Heavy Machinery Bereich. <https://bantleon.de/de/branchen/heavy-machinery/>

Möglichkeit, dass bei der Benutzung von Systemen oder Software nicht alle Funktionen benutzt werden. Um die Entscheidung zu erleichtern, ob ein System unzureichend ist oder ob es sich auf einem guten Weg befindet, soll ein neues Reifegradmodell entwickelt werden.

Für das kooperierende Unternehmen kann anhand eines Überblickes über alle Systeme dargestellt werden, welche Systeme verwendet werden und mit wie vielen anderen Systemen und Anwendungen diese kommunizieren und zusammenarbeiten (siehe Anhang 1). Zur weiteren Darstellung aller Systeme und Anwendungen wurde auch eine Tabelle erstellt, um die Benutzung der verschiedenen Systeme darzustellen, um eventuelle Entscheidungen bezüglich der zur Bewertung ausgewählten Systemen zu treffen (siehe Anhang 2).

7.3 Modellvergleich

Die allgemeine Darstellung und kurze Einführung von verschiedenen Reifegradmodelle erfolgten bereits im zweiten Kapitel dieser Arbeit. Eine detaillierte Vorstellung, Analyse und Vergleich verschiedener Reifegradmodelle wurden im Kapitel drei vorgenommen. Dabei wurden die zwei Reifegradmodelle Capability Maturity Model Integration (CMMI) und Software Process Improvement and Capability dEtermination (SPICE) weiter erläutert und beleuchtet. Dabei wurde das Ergebnis festgestellt, dass CMMI eher für die Unterstützung eines gesamten Unternehmens dient. Im Gegensatz dazu liegt der Fokus von SPICE auf der Verbesserung der Prozesslandschaft. Die Vorstellung und Analyse der beiden Modelle sollen in der Entwicklung des Reifegradmodells eine zentrale Rolle spielen.

7.4 Festlegung der Entwicklungsstrategie

Nach dem Vergleich und der genaueren Darstellung der beiden Modelle gilt es nun zu entscheiden, welche Art des neu selbst erstellten Reifegradmodell entsprechen soll. Dabei muss entschieden werden, ob eine vollständige Neuentwicklung, eine Weiterentwicklung oder eine Kombination der beiden Modelle stattfinden soll. Nach einiger Überlegung kam es zu dem Entschluss, dass eine Kombination der beiden Reifegradmodelle mit einer Weiterentwicklung die beste Entscheidung sein wird. Hierbei werden die Maturity Levels beider Modelle verwendet und in anderer Art wiederverwendet. Auch die Erfüllungsstufen von SPICE sollen in modifizierter Art verwendet werden.

7.5 Iterative Reifegradmodellentwicklung

Zu Beginn der ersten Iterationsstufe wurde ein grundlegendes Modell entwickelt, welches aus den Bewertungskriterien, Reifestufen und einer Bewertungsskala besteht. Die Bewertungskriterien basierten dabei auf der davor ausgeführten Literaturrecherche. Aus dem Kapitel 2.4.3

Qualitätsmerkmale von Software wurden die verschiedenen Bereiche Funktionalität, Laufzeit, Zuverlässigkeit, Benutzbarkeit, Transparenz, Überarbeitung und Testbarkeit herausgesucht und daraufhin optisch dargestellt. Des Weiteren basierend auf der Recherche ist die Inspiration der Bewertungsskala entstanden. Dabei wurde als Vorbild die sechs verschiedenen Capability Levels ausgewählt. Diese sechs Stufen wurden aus den Modellen CMMI und SPICE übernommen, um die Bewertung der einzelnen Bewertungskriterien vorzunehmen. Auch die Reifestufen wurden im Wesentlichen vom CMMI-Modell übernommen. Die fünf Stufen Optimizing, Quantitatively Managed, Defined, Managed und Initial sollten die Endbewertung des zu Bewerteten Systems, Software oder Anwendung sein. Die Endbewertung sollte aus der durchschnittlichen einzelnen Bewertung der verschiedenen Bewertungskriterien erfolgen. Dieser erste Entwurf wurde nach erster Rücksprache mit dem Ansprechpartner meiner Firma in dieser Form abgelehnt und musste weiter überarbeitet werden. Nach kurzer Absprache und verschiedenen Brainstormings wurde das Modell erweitert. Dabei wurden die Bewertungskriterien in drei Kategorien eingeteilt und mit verschiedenen Unterkategorien erweitert oder anders umschrieben. Unter der ersten Kategorie werden alle technischen Aspekte zusammengefasst. Darunter werden die Funktionalität, Zuverlässigkeit, Verfügbarkeit, Wartbarkeit und Skalierbarkeit einzeln bewertet. Dasselbe gilt für die Kategorie Nicht funktionale Attribute, dabei werden die vier verschiedenen Kriterien Performance, Sicherheit, Zusammenarbeit und Flexibilität bewertet. Schlussendlich kommen wir zur größten Kategorie die Benutzerorientiert heißt. Dabei wird der Fokus auf die neun verschiedenen Kriterien Benutzerfreundlichkeit, Zugänglichkeit, Zufriedenheit, Kosten, Nachhaltigkeit, Komplexität, Transparenz, Innovation und Innovation im Unternehmen gelegt. Alle Kategorien sollen einzeln bewertet werden und zum Gesamtergebnis dazuzählen. Jedoch war nach Rücksprache das Modell auf einem ordentlichen Weg mit Verbesserungspotential. In der dritten Iterationsstufe wurde somit nach einer präziseren Bewertungsmöglichkeit gesucht. Da eine Abhebung von den anderen Modellen wie CMMI und SPICE stattfinden sollte, wurde die Entscheidung getroffen die Bewertungsgrundlage zu überdenken und abzuändern. Dabei sollte eine Wertigkeit eingeführt werden, welche die Gewichtung der einzelnen Bewertungskriterien verstärken soll. Der Ansatz dieser separaten Gewichtung entspricht dabei wieder den Capability Levels. Das heißt die Gewichtung entspricht wie auch bei den Capability Levels sechs Stufen, welche sich aber namentlich unterscheiden. Die erste Stufe ist dabei die Stufe „Nicht relevant“. Sie soll ein Kriterium beschreiben, welches keine Anforderung an das System hat oder, beziehungsweise nicht relevant für die Auswertung ist. Diese Stufe wird mit der Wertigkeit null versehen, welche später bei der Berechnung der Endstufe von Nöten ist. Die Wertigkeit von eins erhält die Stufe „Sehr gering“. Dabei stuft das Unternehmen die Anforderung als kaum bedeutend ein. Bei der dritten Stufe „Gering“, wird die Wertigkeit zwei verteilt. Dabei hat das Kriterium eine kleine Wichtigkeit auf das System oder Software. Die Wertigkeit drei wird in der Stufe „Mittel“ verteilt. Diese soll den normalen Einfluss beschreiben. Bei der fünften Stufe „Hoch“ wird das bestimmte Kriterium als sehr wichtig eingestuft. Als letztes mit einer Wertigkeit von fünf wird die Stufe „Kritisch“ vorgestellt. Dabei hat das Kriterium eine hohe Priorität und ist essenziell für das Unternehmen. Diese Einstufung der bestimmten Kriterien kann sowohl von Unternehmen zu Unternehmen als auch von System zu System variieren. Bei der Erstellung der Bewertungscheckliste soll jeder dabei die Gewichtung selbst bestimmen können. Mit Hilfe dieser Wertigkeit

kann nun eine gesamte und eine tatsächliche Punktzahl berechnet werden. Je nachdem welcher Prozentsatz der Punktzahl erreicht wurde, kann dann die bestimmte Reifestufe des Systems oder Software stattfinden. Die maximale Punktzahl errechnet sich dabei aus der Summe aller Produkte der verschiedenen Bewertungskriterien. Das Produkt wird dabei aus deren Wertigkeit und der maximalen Bewertung fünf berechnet. Mathematisch ergibt sich dann folgende Summe $\sum_{i=1}^m W_i * 5$. Das m steht dabei für die Anzahl der Bewertungskriterien und W_i für die Gewichtung des Bewertungskriterium i. Die zweite benötigte Summe für die Endberechnung ist die Berechnung der tatsächlichen Punktzahl. Dabei ändert sich von der vorherigen Summe nur der Aspekt der Bewertung, da dieser nun jeden Wert zwischen null und fünf annehmen kann. Der Wert der Bewertung wird in der Summe mit B_i beschrieben. Daraus ergibt sich die Punktzahl wie folgt: $\sum_{i=1}^m W_i * B_i$. Um nun das zu bewertendes System oder Software in die fünf Reifestufen einzusortieren, wurde die Entscheidung getroffen die einzelnen Stufen in Prozente einzuteilen, da somit das Reifegradmodell dynamisch funktionieren kann. Dabei wurde nach Rücksprache sich auf folgende Einteilung geeinigt. Ein System oder Software ist in der Reifestufe Initial, wenn es unter oder gleich 29% der maximalen Punktzahl entspricht. Bei den Stufen Managed (30%-49%) und Defined (50%-69%) erkennt man weiter einen durchschnittlichen Prozentwert von 19%, dieser wird dann aber in den nächsten Stufen weniger, da die zunehmende Qualität, um einiges schwieriger zu erhalten ist. Die vierte Stufe Quantitatively Managed erreicht man, wenn man sich zwischen 70%-84% befindet. Jedes System, Software oder Anwendung das über 85% erhält gilt als Optimizing. Die Prozentzahl wird im Allgemeinen durch das Produkt von 100 und dem Quotienten zwischen der tatsächlichen Punktzahl und der maximalen Punktzahl berechnet. Daraus erhält man folgende Formel: $\frac{\text{Punktzahl}}{\text{maximale Punktzahl}} * 100$. Um das beschriebene besser darzustellen, steht in Abbildung 10 das Reifemodell im Kleinen zur Verfügung. Das Ganze kann auch im Anhang 3 genauer und größer betrachtet werden.

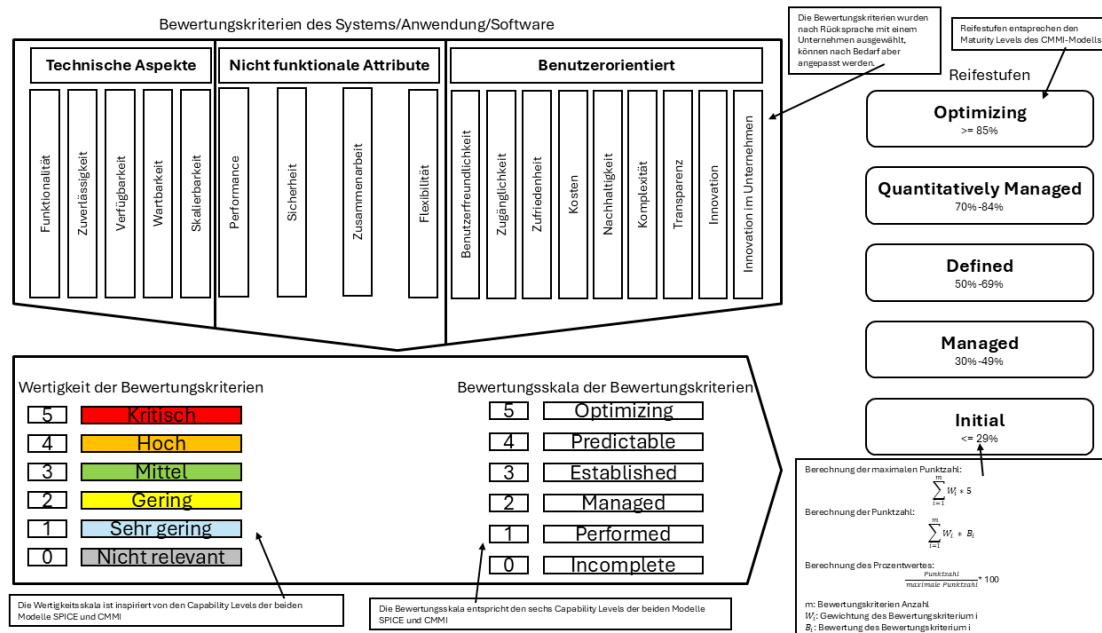


Abbildung 10: Neu entwickeltes Reifegradmodell

7.6 Konzeption von Transfer und Evaluation

Nach der Erstellung des Reifegradmodells erfolgt die Umsetzung und Weitergabe an die Anwender. Im Allgemeinen wird die Evaluation anhand einer Art Bewertungscheckliste stattfinden, die es ermöglicht, jedes System oder Software individuell zu beurteilen. Durch die individuelle Festlegung der Gewichtung und Bewertung kann die Checkliste flexibel eingesetzt werden und liefert variable Ergebnisse. Eine Checkliste bietet den Vorteil, dass sie einfach durchzuführen und in den meisten Fällen leicht zu verstehen ist. Damit wird für eine benutzerfreundliche Alternative gesorgt. Des Weiteren bietet eine Checkliste eine transparente Bewertungsgrundlage, die es ermöglicht die Herangehensweise zu verstehen. Zudem kann durch eine Checkliste etwas schneller ohne weiteres benötigtes Vorwissen die Analyse des Systems oder Software stattfinden.

Um diese Punkte erreichen zu können, soll eine Checkliste bearbeitet werden, die unterschiedlich definierte Kriterien enthält. Zudem soll eine kurze Anleitung, wie genau die Checkliste auszufüllen ist, beigelegt werden. Die Auszufüllende Liste mit Beschreibungen und anderen Anmerkungsfeldern soll so einfach wie möglich dargestellt werden und genügend Platz zur Bewertung bieten. Kategorien oder gewisse Wörter werden näher erklärt, falls Bedarf vorliegen sollte. Des Weiteren sollen Felder für das Endresultat zu dem leichteren Verständnis dargestellt werden.

Nach der Anwendung der Checkliste soll eine Art Interview mit dem Benutzer durchgeführt werden, in welcher Feedback eingeholt werden soll. Was kann verbessert werden? War die Hilfe der Checkliste erfolgreich? Kann mithilfe der Checkliste eine Tendenz zum System oder Software gemacht werden? Dabei soll festgestellt werden, ob und wie das System verbessert werden kann und in welchen Teilen die Änderungen stattfinden sollen. Wie die Informationen der Checkliste bewertet und genutzt werden liegt aber auch hier individuell beim Benutzer der Checkliste. Die Liste soll nur unterstützen und zu einer sinnvollen Entscheidung beitragen.

7.7 Implementierung der Transfermittel

Die Dokumente der Checkliste (Abbildung 12-14) und eventuelle Fragen (siehe Abbildung 11), die der ausführenden Person gestellt werden könnten, sind hier in verkleinerter Form dargestellt. Im Anhang sind des Weiteren die Dokumente einsehbar (Abbildung 18-20 und Anhang 4). Die Datei kann per Word- oder PDF-Datei weitergegeben werden, sodass eventuelle Änderungen vorgenommen werden können.

1. Waren die Fragen klar und eindeutig formuliert?
2. Hatten Sie den Eindruck, dass alle relevanten Themenbereiche abgedeckt wurden?
3. Haben die Fragen geholfen, den tatsächlichen Reifegrad realistisch einzuschätzen?
4. Gab es inhaltliche Lücken oder fehlende Aspekte, die ergänzt werden sollten?
5. War der Aufbau der Checkliste übersichtlich und leicht nachvollziehbar?
6. Wie verständlich waren die Anweisungen zur Bearbeitung?
7. War die Länge der Checkliste angemessen?
8. War der zeitliche Aufwand für das Ausfüllen angemessen?
9. Was würden Sie an der Checkliste verbessern?
10. Würden Sie die Checkliste auch anderen empfehlen? Warum oder warum nicht?
11. Hat sich Ihre Sicht auf den Reifegrad durch die Bearbeitung verändert?
12. Welche zusätzlichen Informationen oder Erläuterungen hätten Sie sich gewünscht?

Abbildung 11: Fragen zur Checkliste

Checkliste zur Bewertung von Systemen/Software

Diese Checkliste soll zu einer systematischen und einheitlichen Bewertung von Systemen und Software dienen. Dabei soll grundlegend eine Anregung zur Beurteilung von den jeweiligen Systemen führen, um eine Grundlage für weitere Entscheidungen zu schaffen.

Anleitung: Füllen sie Schritt für Schritt die leeren Felder aus. Dabei wird zuerst mit der Gewichtung der Kategorien gestartet. Jedes Bewertungskriterium soll dabei in eine von sechs verschiedenen Kategorien (siehe Gewichtung) eingeordnet werden. Dabei spielt es eine Rolle, für wie wichtig ihr Unternehmen das jeweilige Kriterium einordnet. Nachdem für jedes Kriterium eine Gewichtung erstellt wurde, erfolgt nun die Bewertung des Kriteriums. Dabei wird das System anhand des bestimmten Kriteriums bewertet. Diese Bewertung findet auch in einer Bewertung mit sechs Stufen statt. Indem zum Beispiel ein System mit fünf bewertet wird, ist das System im jeweiligen Kriterium in herausragender Verfassung. Zu näheren Erläuterung der Entscheidung kann, wenn benötigt ein Bemerkungsfeld ausgefüllt werden. Dies dient zur Darstellung der Begründung für andere Personen, damit der Entscheidungsweg nachvollziehbar ist. Nachdem alle Kriterien mit ihrer Gewichtung und Bewertung versehen sind kann eine Berechnung der maximalen und tatsächlichen Punkte stattfinden. Dabei werden für die tatsächliche Punktzahl eines Kriteriums die Wertigkeit mit der Bewertung multipliziert. Für die maximale Punktzahl wird die Gewichtung mit der am höchsten zu erreichende Punktzahl (in dem Fall fünf) multipliziert. Um das Endgültige Ergebnis zu erhalten, werden alle Punkte zusammengezählt und mithilfe von Reifestufen eingeordnet. Nach dieser Einordnung ist es möglich weitere Schritte bezüglich des Systems oder Software zu treffen.

Gewichtung:

5	Kritisch
4	Hoch
3	Mittel
2	Gering
1	Sehr gering
0	Nicht relevant

Bewertungsskala:

5	Optimizing
4	Predictable
3	Established
2	Managed
1	Performed
0	Incomplete

Reifestufen

Optimizing (>= 85%)
Quantitatively Managed (70%-84%)
Defined (50%-69%)
Managed (30%-49%)
Initial (<= 29%)

Bedeutung Begriffe Bewertungsskala: Incomplete = System/Software erfüllt wesentliche Anforderungen nicht; Performed = erfüllt Aufgaben, reagiert auf Probleme; Managed = aktive Überwachung und feste Verantwortlichkeiten, Established = stabil, dokumentiert und mit festen Standards; Predictable = stabil und berechenbar, gezielte Steuerung; Optimizing = technisch ausgereift, aktive Überwachung und optimiert

Reifegradstufen: Initial = unstrukturierter, instabil, abhängig → Ersetzen; Managed = kontrolliert, reaktiv → Ändern; Defined = dokumentiert, solide → Ändern; Quantitatively Managed = gesteuert, datenbasiert → gezielt verbessern; Optimizing = wenige Verbesserungen

Abbildung 12: Checkliste zur Bewertung von Systemen/Software Teil 1

Name des Systems/Software:				Endpunktzahl:	
Ausführende Person:			Reifegradstufe:		
			Datum:		
Bewertungskriterium (Die Fragen sollen bei der Beantwortung helfen, müssen aber nicht genutzt werden)	Anmerkungen/Begründung	Bewertung (0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)	Gewichtung (0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)	Tatsächliche Punktzahl (Bewertung x Wertigkeit)	Maximale Punktzahl (Wertigkeit x 5)
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?					
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignen sich viele Ausfälle oder andere Hindernisse?					
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?					
4. Wartbarkeit Wie einfach finden Softwareaktualisierungen statt? Wie einfach sind Fehler zu beheben?					
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?					
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?					
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?					
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?					
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?					

Abbildung 13: Checkliste zur Bewertung von Systemen/Software Teil 2

Benutzerorientiert					
1. Benutzerfreundlichkeit Wie einfach ist das System/Software zu bedienen? Ist das System/Software ansprechend für den Benutzer? Wie einfach ist es für Neueinsteiger zu bedienen?					
2. Zugänglichkeit Ist das System/Software für alle nutzbar? Braucht man Lizenzierungen für eine gewisse Anzahl an Personen?					
3. Zufriedenheit Wie zufrieden sind die Mitarbeitenden?					
4. Kosten Wie hoch sind die Beschaffungskosten? Wie hoch sind Betriebs- und Wartungskosten?					
5. Nachhaltigkeit Ist das System umweltverträglich? Wie sieht der Energieverbrauch aus?					
6. Komplexität Ist das System/Software im Aufbau und Verhalten komplex? Ist das System/Software nachvollziehbar?					
7. Transparenz Können Entscheidungen und Abläufe nachvollzogen werden? Kann man auf ältere Abläufe zugreifen und einsehen?					
8. Innovation Ist das System/Software technologisch fortschrittlich? Kann es noch neue Veränderungen am System/Software geben?					
9. Innovation im Unternehmen Bringt das System/Software noch weitere Vorteile, die noch nicht genutzt werden? Kann das System/Software eventuell noch andere Projekte oder Aufgaben übernehmen?					
Reifegradstufen:				Summe aller Punkte	Summe max. Punkte
<input type="checkbox"/> Optimizing (>= 85%)	<input type="checkbox"/> Defined (50%-69%)	<input type="checkbox"/> Initial (<= 29%)			
<input type="checkbox"/> Quantitatively Managed (70%-84%)	<input type="checkbox"/> Managed (30%-49%)				
				Reifegradstufe:	Prozent Rechnung: $\frac{\text{Punktzahl}}{\text{maximale Punktzahl}} \cdot 100$

Abbildung 14: Checkliste zur Bewertung von Systemen/Software Teil 3

7.8 Durchführung der Evaluation

Die Durchführung der Evaluation erfolgte gemeinsam mit dem Mitarbeiter Alexander Bunk der Hermann Bantleon GmbH statt. Dabei wurde schrittweise die Checkliste durchgearbeitet. Zunächst wurde die Gewichtung der Bewertungskriterien bestimmt. Anschließend die Bewertung mit ihren Anmerkungen ausgefüllt, um schließlich die endgültige Bewertung zu erreichen. Dabei wurden verschiedene betrachtet, sowohl als einzelne Software als auch im ganzen System.

Den Start machte dabei IBM Cognos. Aus der Checkliste geht dabei hervor, dass IBM Cognos sich in der Reifegradstufe Managed aufhält (siehe Abbildung 16, 17 und Anhang 5). Dies signalisiert, dass die Software zwar kontrolliert abläuft. Jedoch sehr reaktive ist und erst reagiert, wenn Probleme auftreten. Des Weiteren verdeutlicht das Endergebnis, dass das System technologisch nicht gut genug ist. Durch die Checkliste wurde die vorherige Einschätzung bestätigt.

Ein Nachfolger von IBM Cognos wird Microsoft Power BI, welches als nächstes geprüft werden soll. Bei der Bearbeitung der Checkliste von Microsoft Power BI fällt direkt die bessere Bewertung ins Auge. Die Reifestufe beträgt dabei Quantitatively Managed (siehe Abbildung 18, 19 und Anhang 6). Es bestätigt, dass der Wechsel von IBM Cognos auf Microsoft Power BI eine gute Entscheidung sein wird.

Auch die Software ChemGes wurde mit der Checkliste geprüft. Auch hier bestätigte die Checkliste die bereits vermutete Einschätzung. Die Software erreichte die Reifegradstufe Defined (siehe Abbildung 20, 21 und Anhang 7) und stellt somit eine solide Software dar. Besonders die Kriterien der Kosten und Funktionalität spielen hier eine entscheidende Rolle, während andere Bereiche nicht mit guten Werten glänzen. Jedoch ergibt sich aus den zwei gut bewerteten Kategorien der Entschluss, dass die Software gut genug für ihren Zweck ist.

Durch die Analyse der Systeme und Software war eine ordentliche Einschätzung das Ergebnis. Das zuvor erwartete Ergebnis wurde in den Fällen mehr oder weniger bestätigt. Somit schafft die Checkliste eine gute Bewertungsgrundlage für andere Systeme und Software.

Nach Abschluss der drei Checklisten erfolgte eine Befragung zur Handhabung Checkliste statt (siehe Abbildung 15 und Anhang 8). Das Feedback war insgesamt sehr positiv. Kleinere Verbesserungsvorschläge wurden bereits getätigt, wie beispielsweise der Wechsel von der festen Checkliste in eine dynamische Liste in Excel oder die Aufnahme von mehreren Fragen zu den einzelnen Themengebieten. Zusammenfassend lässt sich feststellen, dass die Einschätzung vor der Checkliste weitestgehend mit den Ergebnissen der Checkliste entspricht.

7.9 Modellweiterentwicklung und Lebenszyklus

Wie bereits erwähnt, besteht die Möglichkeit, das Modell weiterzuentwickeln. Geplant ist ein dynamisches Modell in Excel. Dabei können aus den gegebenen Punktzahlen die Berechnungen und

Reifegrade automatisch generiert werden. Zudem steht eine Ergänzung weiterer Fragen, auf der Aufgabenliste.

Eine weitere Form der Weiterentwicklung kann durch die Inspiration von neuen Reifegradmodellen entstehen. Dabei gilt es dem neusten Stand der Forschung zu folgen und sein Reifegradmodell kontinuierlich auf Grundlage von Feedback und anderen Erfahrungen stetig zu optimieren.

1. Waren die Fragen klar und eindeutig formuliert?

Ja, die Fragen waren klar und eindeutig formuliert.

2. Hatten Sie den Eindruck, dass alle relevanten Themenbereiche abgedeckt wurden?

Ja, alle relevanten Themenbereiche wurden abgedeckt.

3. Haben die Fragen geholfen, den tatsächlichen Reifegrad realistisch einzuschätzen?

Ja, die Fragen haben geholfen, den tatsächlichen Reifegrad realistisch einzuschätzen.

4. Gab es inhaltliche Lücken oder fehlende Aspekte, die ergänzt werden sollten?

Nein, es gab keine inhaltlichen Lücken oder fehlenden Aspekte.

5. War der Aufbau der Checkliste übersichtlich und leicht nachvollziehbar?

Ja, der Aufbau der Checkliste war übersichtlich und leicht nachvollziehbar.

6. Wie verständlich waren die Anweisungen zur Bearbeitung?

Die Anweisungen waren verständlich, allerdings gab es Überschneidungen der Themen, da verschiedene Themen zusammenhängen.

7. War die Länge der Checkliste angemessen?

Ja, die Länge der Checkliste war angemessen.

8. War der zeitliche Aufwand für das Ausfüllen angemessen?

Ja, der zeitliche Aufwand für das Ausfüllen war angemessen.

9. Was würden Sie an der Checkliste verbessern?

Eventuell die Checkliste in Excel übertragen, um eine automatisierte Auswertung zu ermöglichen.

10. Würden Sie die Checkliste auch anderen empfehlen? Warum oder warum nicht?

Ja, die Checkliste bietet mit wenig Aufwand eine gute Einschätzung für ein System oder eine Software.

11. Hat sich Ihre Sicht auf den Reifegrad durch die Bearbeitung verändert?

Die Bearbeitung hat meine Einschätzung bestätigt.

12. Welche zusätzlichen Informationen oder Erläuterungen hätten Sie sich gewünscht?

Eine stärkere Eingrenzung durch gezieltere Fragen wäre wünschenswert, auch wenn dies den Bearbeitungsaufwand erhöhen würde.

Abbildung 15: Fragen zur Checkliste

Name des Systems/Software: IBM Cognos			Endpunktzahl: 149/335		
Ausführende Person: Alexander Bunk/ Jan Schrade		Reifegradstufe: Managed			
Datum: 13.08.2025					
Bewertungskriterium (Die Fragen sollen bei der Beantwortung helfen, müssen aber nicht genutzt werden)	Anmerkungen/Begründung	Bewertung (0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)	Gewichtung (0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)	Tatsächliche Punktzahl (Bewertung x Wertigkeit)	Maximale Punktzahl (Wertigkeit x 5)
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?	Erfüllt nicht alle erwarteten Aufgaben	2	4	8	20
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignen sich viele Ausfälle oder andere Hindernisse?	Zwei On-Premise Systeme die gepflegt werden müssen	3	1	3	5
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?	Gibt nach Updates regelmäßig Probleme Nicht direkt von außerhalb erreichbar	2	3	6	15
4. Wartbarkeit Wie einfach finden Softwareaktualisierungen statt? Wie einfach sind Fehler zu beheben?	Externer Dienstleister für Softwareaktualisierungen notwendig	1	3	3	15
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?	Benötigt zusätzliche Serverressourcen für zusätzliche Daten	4	5	20	25
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?	Eventuell längere Wartezeiten	3	4	12	20
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?	Geschützt vor unbefugtem Zugriff	5	5	25	25
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?	Großer Aufwand für zusätzliche Schnittstellen	1	4	4	20
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?	Größere Anpassungen nur durch externen Dienstleister	1	4	4	20

Abbildung 16: Checkliste IBM Cognos Teil 1

Benutzerorientiert					
1. Benutzerfreundlichkeit Wie einfach ist das System/Software zu bedienen? Ist das System/Software ansprechend für den Benutzer? Wie einfach ist es für Neueinsteiger zu bedienen?	Nicht intuitive Oberfläche	1	4	4	20
2. Zugänglichkeit Ist das System/Software für alle nutzbar? Braucht man Lizenzierungen für eine gewisse Anzahl an Personen?	Lizenzen nur für bestimmten Benutzerkreis verfügbar	2	3	6	15
3. Zufriedenheit Wie zufrieden sind die Mitarbeitenden?		2	5	10	25
4. Kosten Wie hoch sind die Beschaffungskosten? Wie hoch sind Betriebs- und Wartungskosten?	Externer Dienstleistungen Hohe jährliche Wartungskosten	2	4	8	20
5. Nachhaltigkeit Ist das System umweltverträglich? Wie sieht der Energieverbrauch aus?		3	4	12	20
6. Komplexität Ist das System/Software im Aufbau und Verhalten komplex? Ist das System/Software nachvollziehbar?	Komplexes und nicht nachvollziehbar	1	3	3	15
7. Transparenz Können Entscheidungen und Abläufe nachvollzogen werden? Kann man auf ältere Abläufe zugreifen und einsehen?	Details zum Teil einsehbar	3	3	9	15
8. Innovation Ist das System/Software technologisch fortschrittlich? Kann es noch neue Veränderungen am System/Software geben?	Neue Funktionen/Veränderungen sind mit Kosten verbunden	1	4	4	20
9. Innovation im Unternehmen Bringt das System/Software noch weitere Vorteile, die noch nicht genutzt werden? Kann das System/Software eventuell noch andere Projekte oder Aufgaben übernehmen?	Im Prinzip können mehr Projekte übernommen werden, jedoch zusätzliche Kosten und komplexe Änderungen	2	4	8	20
Reifegradstufen:				Summe aller Punkte	Summe max. Punkte
<input type="checkbox"/> Optimizing (>= 85%)	<input type="checkbox"/> Defined (50%-69%)	<input type="checkbox"/> Initial (<= 29%)		149	335
<input type="checkbox"/> Quantitatively Managed (70%-84%)	<input type="checkbox"/> Managed (30%-49%)			Reifegradstufe: Managed	Prozent Rechnung: $\frac{\text{Punktzahl}}{\text{maximale Punktzahl}} \cdot 100$ 44,5%

Abbildung 17: Checkliste IBM Cognos Teil 2

Name des Systems/Software: Microsoft Power BI

Endpunktzahl: 264/335

Ausführende Person: Alexander Bunk/ Jan Schrade

Reifegradstufe: Quantitatively Managed

Datum: 13.08.2025

Bewertungskriterium (Die Fragen sollen bei der Beantwortung helfen, müssen aber nicht genutzt werden)	Anmerkungen/Begründung	Bewertung (0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)	Gewichtung (0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)	Tatsächliche Punktzahl (Bewertung x Wertigkeit)	Maximale Punktzahl (Wertigkeit x 5)
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?	Erwartete Funktionen sind gegeben	5	4	20	20
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignen sich viele Ausfälle oder andere Hindernisse?	Bisher keine Ausfälle oder andere Probleme mit der Erreichbarkeit	5	1	5	5
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?	Kann von außerhalb bedient werden Jederzeit erreichbar	5	3	15	15
4. Wartbarkeit Wie einfach finden Softwareaktualisierungen statt? Wie einfach sind Fehler zu beheben?	Kleinere Fehler bei der Datenaktualisierung sind möglich, jedoch wieder schnell behoben	4	3	12	15
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?	System ist auf große Datenmengen ausgelegt Läuft in der Cloud, somit keine Einschränkungen durch Hardware oder andere Hindernisse	5	5	25	25
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?	Solide Performance Zum Teil nicht beeinflussbar	3	4	12	20
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?	Daten werden aus externem System übertragen. Sicherheitsstufen können erteilt werden Unbefugter Zugriff wird durch Sicherheitsmaßnahmen verhindert	5	5	25	25
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?	Gateways/Konnektoren zu einer Vielzahl von Systemen stehen zur Verfügung	4	4	16	20
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?	System benötigt technisches Vorwissen zur Anpassung Mit nötigem Knowhow ist das System flexibel anpassbar	2	4	8	20

Abbildung 18: Checkliste Microsoft Power Bi Teil 1

Benutzerorientiert					
1. Benutzerfreundlichkeit Wie einfach ist das System/Software zu bedienen? Ist das System/Software ansprechend für den Benutzer? Wie einfach ist es für Neueinsteiger zu bedienen?	System benötigt kleinere Einarbeitungszeit Evtl. kleinere Schulung	3	4	12	20
2. Zugänglichkeit Ist das System/Software für alle nutzbar? Braucht man Lizenzierungen für eine gewisse Anzahl an Personen?	Notwendige Lizenz steht für alle PC-Benutzer zur Verfügung	4	3	12	15
3. Zufriedenheit Wie zufrieden sind die Mitarbeitenden?	Feedback unterschiedlich, Benutzer sehen nach einer gewissen Zeit einen Mehrwert	4	5	20	25
4. Kosten Wie hoch sind die Beschaffungskosten? Wie hoch sind Betriebs- und Wartungskosten?	Ist kombiniert mit anderem System/Anwendungen	3	4	12	20
5. Nachhaltigkeit Ist das System umweltverträglich? Wie sieht der Energieverbrauch aus?	Funktioniert und läuft über die Cloud	3	4	12	20
6. Komplexität Ist das System/Software im Aufbau und Verhalten komplex? Ist das System/Software nachvollziehbar?	Zusammenarbeit von verschiedenen Komponenten sorgt für Komplexität Zusammenhang ist jedoch nachvollziehbar	3	3	9	15
7. Transparenz Können Entscheidungen und Abläufe nachvollzogen werden? Kann man auf ältere Abläufe zugreifen und einsehen?	Kleinere Änderungen werden angezeigt und archiviert	3	3	9	15
8. Innovation Ist das System/Software technologisch fortschrittlich? Kann es noch neue Veränderungen am System/Software geben?	Neue Funktionen werden regelmäßig hinzugefügt	5	4	20	20
9. Innovation im Unternehmen Bringt das System/Software noch weitere Vorteile, die noch nicht genutzt werden? Kann das System/Software eventuell noch andere Projekte oder Aufgaben übernehmen?	Kann für Auswertungen in weiteren Bereichen genutzt werden	5	4	20	20
Reifegradstufen:				Summe aller Punkte	Summe max. Punkte
<input type="checkbox"/> Optimizing (>= 85%)	<input type="checkbox"/> Defined (50%-69%)	<input type="checkbox"/> Initial (<= 29%)		264	335
<input type="checkbox"/> Quantitatively Managed (70%-84%)	<input type="checkbox"/> Managed (30%-49%)			Reifegradstufe: Quantitatively Managed	Prozent Rechnung: $\frac{\text{Punktzahl}}{\text{maximale Punktzahl}} \cdot 100$ 78,8%

Abbildung 19: Checkliste Microsoft Power BI Teil 2

Bewertungskriterium <small>(Die Fragen sollen bei der Beantwortung helfen, müssen aber nicht genutzt werden)</small>	Anmerkungen/Begründung	Bewertung <small>(0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)</small>	Gewichtung <small>(0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)</small>	Tatsächliche Punktzahl <small>(Bewertung x Wertigkeit)</small>	Maximale Punktzahl <small>(Wertigkeit x 5)</small>
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?	Erfüllt den Zweck, der Weg dahin ist aber komplex Bestimmte Anforderungen nicht enthalten	4	4	16	20
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignen sich viele Ausfälle oder andere Hindernisse?	Nach Updates sind Ausfälle möglich	2	4	8	20
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?	On-Premise Server Über remote Server erreichbar	4	3	12	15
4. Wartbarkeit Wie einfach finden Softwareaktualisierungen statt? Wie einfach sind Fehler zu beheben?	Fehler nach Updates nur durch den Hersteller zu beheben	2	3	6	15
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?	Systemarchitektur nicht für große Datenmengen ausgelegt	2	1	2	5
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?		4	4	16	20
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?	Läuft auf On-Premise Server Nur bestimmte Personen haben Zugriff	5	5	25	25
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?	Nur eingeschränkte Schnittstellen verfügbar	1	4	4	20
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?	Änderungen müssen individuell programmiert werden	2	4	8	20

Abbildung 20: Checkliste ChemGes Teil 1

Benutzerorientiert					
1. Benutzerfreundlichkeit Wie einfach ist das System/Software zu bedienen? Ist das System/Software ansprechend für den Benutzer? Wie einfach ist es für Neueinsteiger zu bedienen?	Nicht intuitive Oberfläche Viele Shortcuts Bedienkonzept stark veraltet	0	4	0	20
2. Zugänglichkeit Ist das System/Software für alle nutzbar? Braucht man Lizensierungen für eine gewisse Anzahl an Personen?	Nicht jeder braucht einen Zugang Lizenz in der Anwendung enthalten	5	1	5	5
3. Zufriedenheit Wie zufrieden sind die Mitarbeitenden?	Veraltetes Bedienkonzept, unflexibel	1	5	5	25
4. Kosten Wie hoch sind die Beschaffungskosten? Wie hoch sind Betriebs- und Wartungskosten?	Kosten sind im Vergleich zur Konkurrenz niedrig	5	4	20	20
5. Nachhaltigkeit Ist das System umweltverträglich? Wie sieht der Energieverbrauch aus?		3	4	12	20
6. Komplexität Ist das System/Software im Aufbau und Verhalten komplex? Ist das System/Software nachvollziehbar?	Informationen werden im Sever mitgeführt	3	3	9	15
7. Transparenz Können Entscheidungen und Abläufe nachvollzogen werden? Kann man auf ältere Abläufe zugreifen und einsehen?	Historie der Daten wird im Programm mitgeführt	4	3	12	15
8. Innovation Ist das System/Software technologisch fortschrittlich? Kann es noch neue Veränderungen am System/Software geben?	Kleinere Updates Updates enthalten nur rechtliche Anpassungen keine funktionellen	1	4	4	20
9. Innovation im Unternehmen Bringt das System/Software noch weitere Vorteile, die noch nicht genutzt werden? Kann das System/Software eventuell noch andere Projekte oder Aufgaben übernehmen?		0	4	0	20
Reifegradstufen:				Summe aller Punkte	Summe max. Punkte
Optimizing (>= 85%)	Defined (50%-69%)	Initial (<= 29%)		164	320
Quantitatively Managed (70%-84%)	Managed (30%-49%)				
Reifegradstufe: Defined				Prozent Rechnung: $\frac{\text{Punktzahl}}{\text{maximale Punktzahl}} * 100$	51,25%

Abbildung 21: Checkliste ChemGes Teil 2

8 Fazit

Das Ziel dieser Arbeit war die Entwicklung eines Reifegradmodells zur Bewertung von Systemen und Software. Die zugrundeliegende Forschungsfrage lautete: Wie kann ein praxisorientiertes Reifegradmodell zur Optimierung der IT-Infrastruktur entwickelt werden, damit Maßnahmen zu den Automatisierungen und Systemanpassungen beachtet, integriert und anhand von bestimmten Kriterien wie Benutzerfreundlichkeit, Effizienzsteigerung, Kosten und Skalierbarkeit bewertet werden?

Während der Recherche wurde festgestellt, dass bereits viele verschiedene Reifegradmodell existieren, welche alle ihre verschiedenen Ziele verfolgen. Dabei kristallisierte sich die genauere Analyse von den zwei Reifegradmodellen CMMI und SPICE heraus. Diese zwei Modelle schafften auch die Grundlage des eigen entwickelten Reifegradmodells mit ihren Maturity und Capability Levels. Mit Hilfe eines Ablaufplans zu Erstellung eines Reifegradmodells wurde ein iterativer Entwicklungszyklus befolgt und umgesetzt. Teil dieses Ablaufplans war eine passende Evaluation, welche mit einer eigenen erstellten Checkliste erfolgreich ausgeführt wurde.

Die Ergebnisse zeigen, dass die Forschungsfrage durch die Erstellung des Modells und der Checkliste beantwortet werden konnte. Die Ergebnisse der Analyse bestätigten die zuvor getroffenen Einschätzungen der Systeme. Insgesamt ermöglicht das entwickelte Reifegradmodell durch die Bewertung mit seiner Checkliste eine fundierte Einschätzung der bewerteten Systeme und Software. Dabei dienen die Checkliste und der Reifegrad als Unterstützung der Beurteilung des Systems oder Software. Sowohl das Modell als auch die Checkliste sind dabei wie die anderen Reifegradmodell weiterentwickelbar und können durch den Wandel stetig verändert werden.

Auf Grundlage der Ergebnisse lässt sich also sagen, dass das Reifegradmodell eine Hilfestellung zur Bewertung von Systemen ist. Dieses kann jedoch nicht komplett die Entscheidung übernehmen, da eventuell Systeme essenziell für ein Unternehmen sind, ohne dass es einen möglichen Ersatz gibt. Zusammenfassend lässt sich feststellen, dass das entwickelte Reifegradmodell seine Aufgaben erfüllt und einen praktischen Mehrwert für Unternehmen bietet.

9 Anhang

Anhang 1: Whiteboard Systemlandschaft

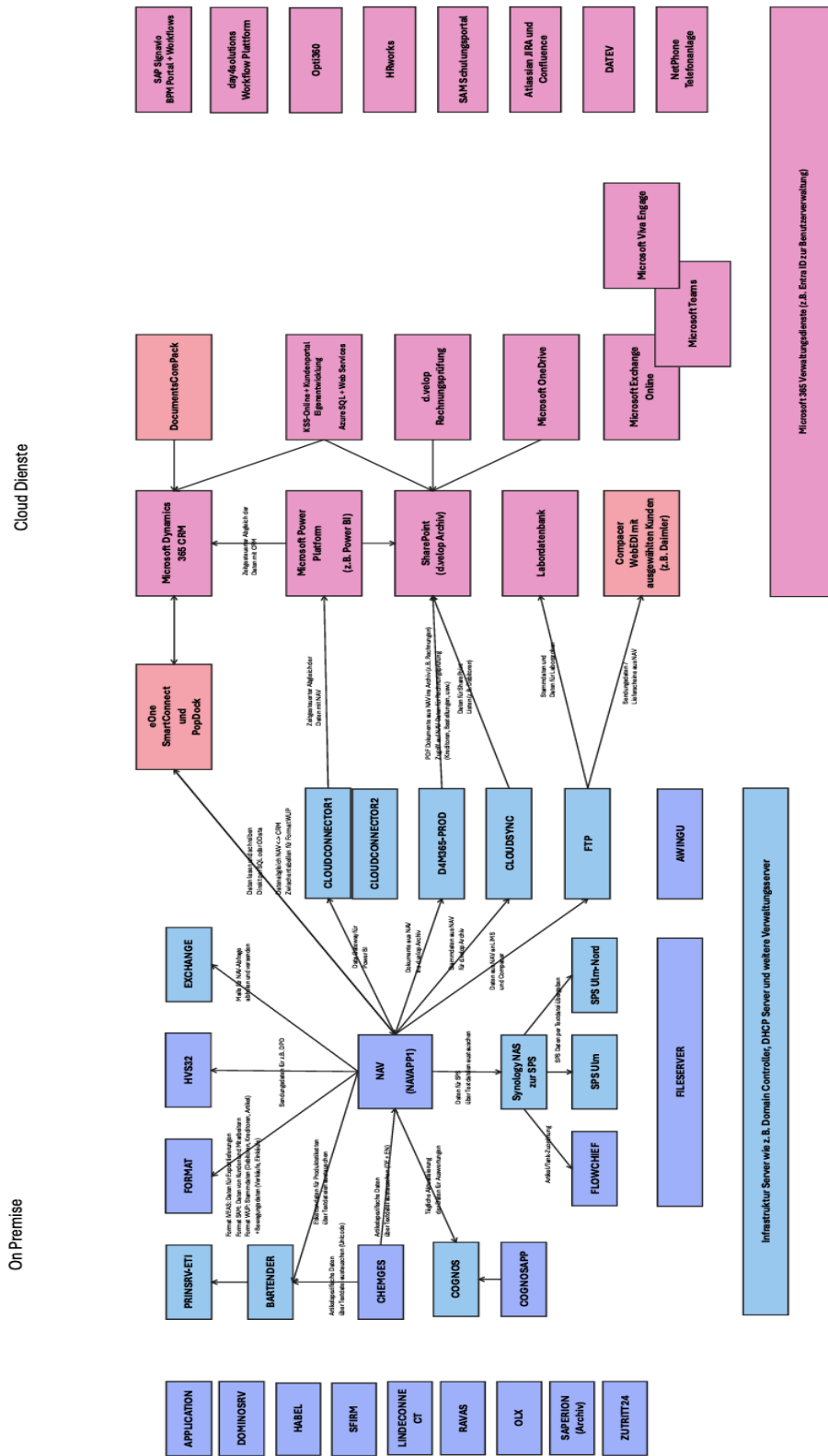


Abbildung 22: Übersicht der Systeme und Software der Herrmann Bantleon GmbH

Anhang 3: Reifegradmodell Darstellung

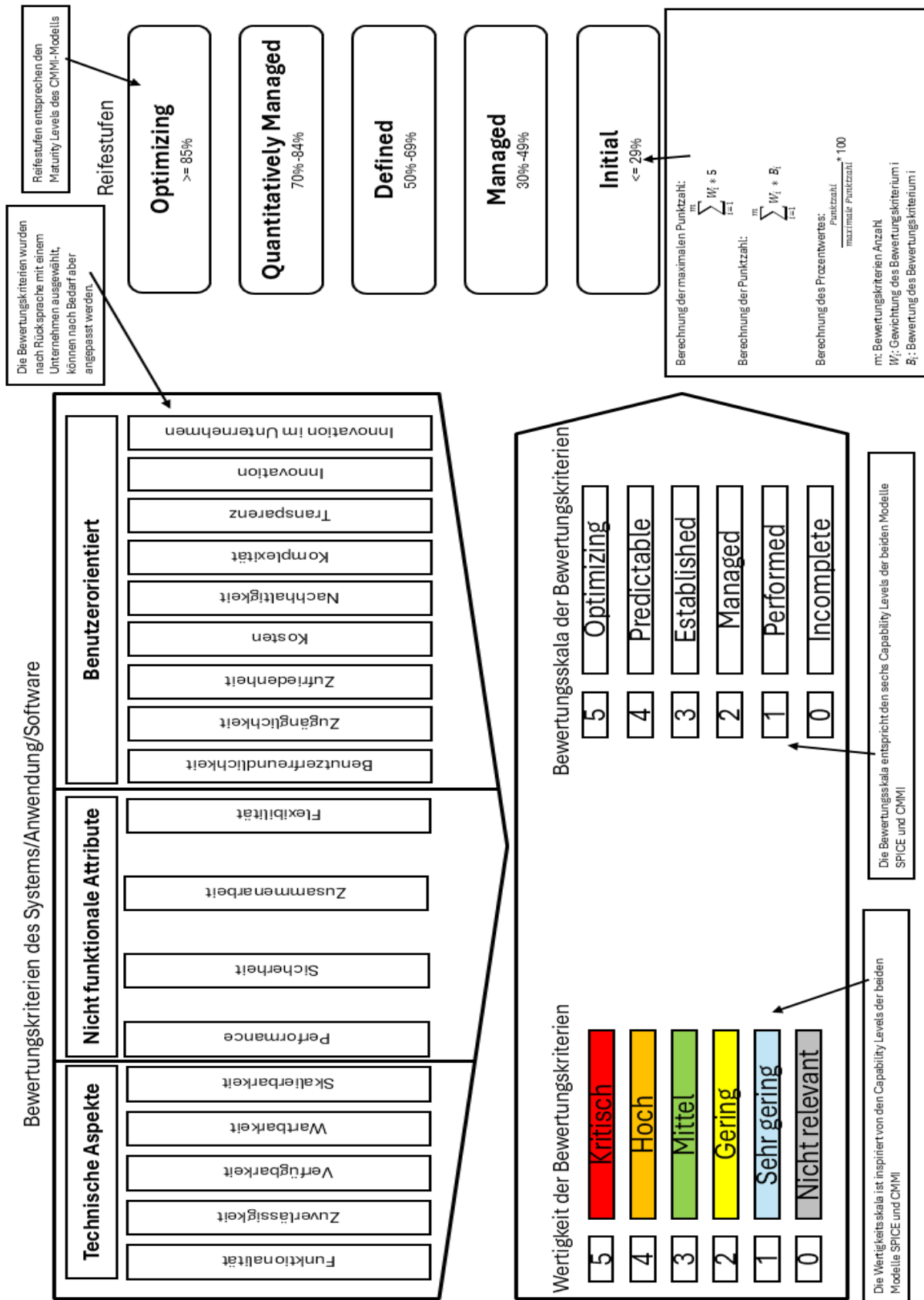


Abbildung 24: Neu entwickeltes Reifegradmodell

Checkliste zur Bewertung von Systemen/Software

Diese Checkliste soll zu einer systematischen und einheitlichen Bewertung von Systemen und Software dienen. Dabei soll grundlegend eine Anregung zur Beurteilung von den jeweiligen Systemen führen, um eine Grundlage für weitere Entscheidungen zu schaffen.

Anleitung: Füllen sie Schritt für Schritt die leeren Felder aus. Dabei wird zuerst mit der Gewichtung der Kategorien gestartet. Jedes Bewertungskriterium soll dabei in eine von sechs verschiedenen Kategorien (siehe Gewichtung) eingeordnet werden. Dabei spielt eine Rolle, für wie wichtig ihr Unternehmen das jeweilige Kriterium einordnet. Nachdem für jedes Kriterium eine Gewichtung erstellt wurde, erfolgt nun die Bewertung des Kriteriums. Dabei wird das System anhand des bestimmten Kriteriums bewertet. Diese Bewertung findet auch in einer Bewertung mit sechs Stufen statt. Indem zum Beispiel ein System mit fünf bewertet wird, ist das System im jeweiligen Kriterium in herausragender Verfassung. Zu näheren Erläuterung der Entscheidung kann, wenn benötigt ein Bemerkungsfeld ausgefüllt werden. Dies dient zur Darstellung der Begründung für andere Personen, damit der Entscheidungsweg nachvollziehbar ist. Nachdem alle Kriterien mit ihrer Gewichtung und Bewertung versehen sind kann eine Berechnung der maximalen und tatsächlichen Punkte stattfinden. Dabei werden für die tatsächliche Punktzahl eines Kriteriums die Wertigkeit mit der Bewertung multipliziert. Für die maximale Punktzahl wird die Gewichtung mit der am höchsten zu erreichende Punktzahl (in dem Fall fünf) multipliziert. Um das Endgültige Ergebnis zu erhalten, werden alle Punkte zusammengezählt und mithilfe von Reifestufen eingeordnet. Nach dieser Einordnung ist es möglich weitere Schritte bezüglich des Systems oder Software zu treffen.

Gewichtung:

5	Kritisch
4	Hoch
3	Mittel
2	Gering
1	Sehr gering
0	Nicht relevant

Bewertungsskala:

5	Optimizing
4	Predictable
3	Established
2	Managed
1	Performed
0	Incomplete

Reifestufen

Optimizing (>= 85%)
Quantitatively Managed (70%-84%)
Defined (50%-69%)
Managed (30%-49%)
Initial (<= 29%)

Bedeutung Begriffe Bewertungsskala: Incomplete = System/Software erfüllt wesentliche Anforderungen nicht; Performed = erfüllt Aufgaben, reagiert auf Probleme; Managed = aktive Überwachung und feste Verantwortlichkeiten, Established = stabil, dokumentiert und mit festen Standards; Predictable = stabil und berechenbar, gezielte Steuerung; Optimizing = technisch ausgereift, aktive Überwachung und optimiert

Reifegradstufen: Initial = unstrukturierter, instabil, abhängig → Ersetzen; Managed = kontrolliert, reaktiv → Ändern; Defined = dokumentiert, solide → Ändern; Quantitatively Managed = gesteuert, datenbasiert → gezielt verbessern; Optimizing = wenige Verbesserungen

Abbildung 25: Checkliste zur Bewertung von Systemen/Software Teil 1

Name des Systems/Software:

Endpunktzahl:

Ausführende Person:

Reifegradstufe:

Datum:

Bewertungskriterium (Die Fragen sollten bei der Beantwortung helfen, müssen aber nicht genutzt werden)	Anmerkungen/Begründung	Bewertung (0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)	Gewichtung (0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)	Tatsächliche Punktzahl (Bewertung x Wertigkeit)	Maximale Punktzahl (Wertigkeit x 5)
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?					
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignen sich viele Ausfälle oder andere Hindernisse?					
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?					
4. Wartbarkeit Wie einfach finden Software-aktualisierungen statt? Wie einfach sind Fehler zu beheben?					
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?					
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?					
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?					
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?					
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?					

Abbildung 26: Checkliste zur Bewertung von Systemen/Software Teil 2

Anhang 5: Checkliste Bewertung Systeme_Software_Cognos

Name des Systems/Software: IBM Cognos		Endpunktzahl: 149/335			
Ausführende Person: Alexander Bunk/ Jan Schrade		Reifegradstufe: Managed			
		Datum: 13.08.2025			
Bewertungskriterium (Die Fragen sollen bei der Beantwortung helfen, müssen aber nicht genutzt werden)	Anmerkungen/Begründung	Bewertung (0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)	Gewichtung (0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)	Tatsächliche Punktzahl (Bewertung x Wertigkeit)	Maximale Punktzahl (Wertigkeit x 5)
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?	Erfüllt nicht alle erwarteten Aufgaben	2	4	8	20
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignissen sich viele Ausfälle oder andere Hindernisse?	Zwei On-Premise Systeme die gepflegt werden müssen	3	1	3	5
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?	Gibt nach Updates regelmäßig Probleme Nicht direkt von außerhalb erreichbar	2	3	6	15
4. Wartbarkeit Wie einfach finden Software-aktualisierungen statt? Wie einfach sind Fehler zu beheben?	Externer Dienstleister für Softwareaktualisierungen notwendig	1	3	3	15
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?	Benötigt zusätzliche Serverressourcen für zusätzliche Daten	4	5	20	25
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?	Eventuell längere Wartezeiten	3	4	12	20
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?	Geschützt vor unbefugtem Zugriff	5	5	25	25
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?	Großer Aufwand für zusätzliche Schnittstellen	1	4	4	20
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?	Größere Anpassungen nur durch externen Dienstleister	1	4	4	20

Abbildung 28: Checkliste IBM Cognos Teil 1

Benutzerorientiert					
1. Benutzerfreundlichkeit Wie einfach ist das System/Software zu bedienen? Ist das System/Software ansprechend für den Benutzer? Wie einfach ist es für Neueinsteiger zu bedienen?	Nicht intuitive Oberfläche	1	4	4	20
2. Zugänglichkeit Ist das System/Software für alle nutzbar? Braucht man Lizenzierungen für eine gewisse Anzahl an Personen?	Lizenzen nur für bestimmten Benutzerkreis verfügbar	2	3	6	15
3. Zufriedenheit Wie zufrieden sind die Mitarbeitenden?		2	5	10	25
4. Kosten Wie hoch sind die Beschaffungskosten? Wie hoch sind Betriebs- und Wartungskosten?	Externer Dienstleistungen Hohe jährliche Wartungskosten	2	4	8	20
5. Nachhaltigkeit Ist das System umweltverträglich? Wie sieht der Energieverbrauch aus?		3	4	12	20
6. Komplexität Ist das System/Software im Aufbau und Verhalten komplex? Ist das System/Software nachvollziehbar?	Komplexes und nicht nachvollziehbar	1	3	3	15
7. Transparenz Können Entscheidungen und Abläufe nachvollzogen werden? Kann man auf ältere Abläufe zugreifen und einsehen?	Details zum Teil einsehbar	3	3	9	15
8. Innovation Ist das System/Software technologisch fortschrittlich? Kann es noch neue Veränderungen am System/Software geben?	Neue Funktionen/Veränderungen sind mit Kosten verbunden	1	4	4	20
9. Innovation im Unternehmen Bringt das System/Software noch weitere Vorteile, die noch nicht genutzt werden? Kann das System/Software eventuell noch andere Projekte oder Aufgaben übernehmen?	Im Prinzip können mehr Projekte übernommen werden, jedoch zusätzliche Kosten und komplexe Änderungen	2	4	8	20
Reifegradstufen:					
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; border-radius: 10px;">Optimizing (>= 85%)</div> <div style="border: 1px solid black; padding: 5px; border-radius: 10px;">Defined (50%-69%)</div> <div style="border: 1px solid black; padding: 5px; border-radius: 10px;">Initial (<= 29%)</div> </div>					
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; border-radius: 10px;">Quantitatively Managed (70%-84%)</div> <div style="border: 1px solid black; padding: 5px; border-radius: 10px;">Managed (30%-49%)</div> </div>					
			Summe aller Punkte	149	Summe max. Punkte
			Reifegradstufe: Managed	Prozent Rechnung: <small>Punktzahl / maximale Punktzahl * 100</small>	44,5%

Abbildung 29: Checkliste IBM Cognos Teil 2

Name des Systems/Software: Microsoft Power BI		Endpunktzahl: 264/335			
Ausführende Person: Alexander Bunk/ Jan Schrade		Reifegradstufe: Quantitatively Managed			
		Datum: 13.08.2025			
Bewertungskriterium (Die Fragen sollen bei der Beantwortung helfen, müssen aber nicht genutzt werden)	Anmerkungen/Begründung	Bewertung (0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)	Gewichtung (0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)	Tatsächliche Punktzahl (Bewertung x Wertigkeit)	Maximale Punktzahl (Wertigkeit x 5)
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?	Erwartete Funktionen sind gegeben	5	4	20	20
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignen sich viele Ausfälle oder andere Hindernisse?	Bisher keine Ausfälle oder andere Probleme mit der Erreichbarkeit	5	1	5	5
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?	Kann von außerhalb bedient werden Jederzeit erreichbar	5	3	15	15
4. Wartbarkeit Wie einfach finden Software-aktualisierungen statt? Wie einfach sind Fehler zu beheben?	Kleinere Fehler bei der Datenaktualisierung sind möglich, jedoch wieder schnell behoben	4	3	12	15
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?	System ist auf große Datenmengen ausgelegt Läuft in der Cloud, somit keine Einschränkungen durch Hardware oder andere Hindernisse	5	5	25	25
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?	Solide Performance Zum Teil nicht beeinflussbar	3	4	12	20
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?	Daten werden aus externem System übertragen. Sicherheitsstufen können erteilt werde Unbefugter Zugriff wird durch Sicherheitsmaßnahmen verhindert	5	5	25	25
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?	Gateways/Konnektoren zu einer Vielzahl von Systemen stehen zur Verfügung	4	4	16	20
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?	System benötigt technisches Vorwissen zur Anpassung Mit nötigem Knowhow ist das System flexibel anpassbar	2	4	8	20

Abbildung 30: Checkliste Microsoft Power BI Teil 1

Benutzerorientiert					
1. Benutzerfreundlichkeit Wie einfach ist das System/Software zu bedienen? Ist das System/Software ansprechend für den Benutzer? Wie einfach ist es für Neueinsteiger zu bedienen?	System benötigt kleinere Einarbeitungszeit Evtl. kleinere Schulung	3	4	12	20
2. Zugänglichkeit Ist das System/Software für alle nutzbar? Braucht man Lizensierungen für eine gewisse Anzahl an Personen?	Notwenige Lizenz steht für alle PC-Benutzer zur Verfügung	4	3	12	15
3. Zufriedenheit Wie zufrieden sind die Mitarbeitenden?	Feedback unterschiedlich, Benutzer sehen nach einer gewissen Zeit einen Mehrwert	4	5	20	25
4. Kosten Wie hoch sind die Beschaffungskosten? Wie hoch sind Betriebs- und Wartungskosten?	Ist kombiniert mit anderem System/Anwendungen	3	4	12	20
5. Nachhaltigkeit Ist das System umweltverträglich? Wie sieht der Energieverbrauch aus?	Funktioniert und läuft über die Cloud	3	4	12	20
6. Komplexität Ist das System/Software im Aufbau und Verhalten komplex? Ist das System/Software nachvollziehbar?	Zusammenarbeit von verschiedenen Komponenten sorgt für Komplexität Zusammenhang ist jedoch nachvollziehbar	3	3	9	15
7. Transparenz Können Entscheidungen und Abläufe nachvollzogen werden? Kann man auf ältere Abläufe zugreifen und einsehen?	Kleinere Änderungen werden angezeigt und archiviert	3	3	9	15
8. Innovation Ist das System/Software technologisch fortschrittlich? Kann es noch neue Veränderungen am System/Software geben?	Neue Funktionen werden regelmäßig hinzugefügt	5	4	20	20
9. Innovation im Unternehmen Bringt das System/Software noch weitere Vorteile, die noch nicht genutzt werden? Kann das System/Software eventuell noch andere Projekte oder Aufgaben übernehmen?	Kann für Auswertungen in weiteren Bereichen genutzt werden	5	4	20	20
Reifegradstufen:					
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Optimizing (≥= 85%)</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Defined (50%-69%)</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Initial (≤= 29%)</div> </div>					
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Quantitatively Managed (70%-84%)</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Managed (30%-49%)</div> </div>					
			Summe aller Punkte	264	Summe max. Punkte
			Reifegradstufe: Quantitatively Managed	Prozent Rechnung: <small>Punktzahl / maximale Punktzahl * 100</small>	78,8%

Abbildung 31: Checkliste Microsoft Power BI Teil 2

Name des Systems/Software: ChemGes		Endpunktzahl: 164/320			
Ausführende Person: Alexander Bunk/ Jan Schrade		Reifegradstufe: Defined			
		Datum: 13.08.2025			
Bewertungskriterium (Die Fragen sollen bei der Beantwortung helfen, müssen aber nicht genutzt werden)	Anmerkungen/Begründung	Bewertung (0-Incomplete, 1-Performed, 2-Managed, 3-Established, 4-Predictable, 5-Optimizing)	Gewichtung (0-Nicht relevant, 1-Sehr gering, 2-Gering, 3-Mittel, 4-Hoch, 5-Kritisch)	Tatsächliche Punktzahl (Bewertung x Wertigkeit)	Maximale Punktzahl (Wertigkeit x 5)
Technische Aspekte					
1. Funktionalität Funktioniert das System/Software vollständig und korrekt? Erfüllt das System/Software alle erwarteten Aufgaben?	Erfüllt den Zweck, der Weg dahin ist aber komplex Bestimmte Anforderungen nicht enthalten	4	4	16	20
2. Zuverlässigkeit Ist das System oder Software fehleranfällig? Ereignen sich viele Ausfälle oder andere Hindernisse?	Nach Updates sind Ausfälle möglich	2	4	8	20
3. Verfügbarkeit Ist das System/Software jederzeit zu erreichen? Ist das System auch von außerhalb erreichbar?	On-Premise Server Über remote Server erreichbar	4	3	12	15
4. Wartbarkeit Wie einfach finden Software-aktualisierungen statt? Wie einfach sind Fehler zu beheben?	Fehler nach Updates nur durch den Hersteller zu beheben	2	3	6	15
5. Skalierbarkeit Kann das System/Software mit großen Mengen an Daten umgehen? Führen Überlastungen zur Reduzierung der Laufzeitgeschwindigkeit oder anderen größeren Problemen?	Systemarchitektur nicht für große Datenmengen ausgelegt	2	1	2	5
Nicht funktionale Attribute					
1. Performance Wie schnell ist das System/Software? Wie genau werden Ressourcen genutzt?		4	4	16	20
2. Sicherheit Wie gut ist das System/Software vor unbefugtem Zugriff geschützt? Können Daten verloren werden?	Läuft auf On-Premise Server Nur bestimmte Personen haben Zugriff	5	5	25	25
3. Zusammenarbeit Wie kompatibel ist das System/Software mit anderen Systemen/Software?	Nur eingeschränkte Schnittstellen verfügbar	1	4	4	20
4. Flexibilität Ist das System/Software anpassbar? Wie einfach oder schwierig es Veränderungen oder neue Anforderungen zu implementieren?	Änderungen müssen individuell programmiert werden	2	4	8	20

Abbildung 32: Checkliste ChemGes Teil 1

Benutzerorientiert						
1. Benutzerfreundlichkeit Wie einfach ist das System/Software zu bedienen? Ist das System/Software ansprechend für den Benutzer? Wie einfach ist es für Neueinsteiger zu bedienen?	Nicht intuitive Oberfläche Viele Shortcuts Bedienkonzept stark veraltet	0	4	0	20	
2. Zugänglichkeit Ist das System/Software für alle nutzbar? Braucht man Lizenzierungen für eine gewisse Anzahl an Personen?	Nicht jeder braucht einen Zugang Lizenz in der Anwendung enthalten	5	1	5	5	
3. Zufriedenheit Wie zufrieden sind die Mitarbeitenden?	Veraltetes Bedienkonzept, unflexibel	1	5	5	25	
4. Kosten Wie hoch sind die Beschaffungskosten? Wie hoch sind Betriebs- und Wartungskosten?	Kosten sind im Vergleich zur Konkurrenz niedrig	5	4	20	20	
5. Nachhaltigkeit Ist das System umweltverträglich? Wie sieht der Energieverbrauch aus?		3	4	12	20	
6. Komplexität Ist das System/Software im Aufbau und Verhalten komplex? Ist das System/Software nachvollziehbar?	Informationen werden im Sever mitgeführt	3	3	9	15	
7. Transparenz Können Entscheidungen und Abläufe nachvollzogen werden? Kann man auf ältere Abläufe zugreifen und einsehen?	Historie der Daten wird im Programm mitgeführt	4	3	12	15	
8. Innovation Ist das System/Software technologisch fortschrittlich? Kann es noch neue Veränderungen am System/Software geben?	Kleinere Updates Updates enthalten nur rechtliche Anpassungen keine funktionellen	1	4	4	20	
9. Innovation im Unternehmen Bringt das System/Software noch weitere Vorteile, die noch nicht genutzt werden? Kann das System/Software eventuell noch andere Projekte oder Aufgaben übernehmen?		0	4	0	20	
Reifegradstufen:						
Optimizing (>= 85%)	Defined (50%-69%)	Initial (<= 29%)				
Quantitatively Managed (70%-84%)	Managed (30%-49%)					
			Reifegradstufe: Defined	Summe aller Punkte 164	Summe max. Punkte 320	Prozent Rechnung: $\frac{\text{Punktzahl}}{\text{maximale Punktzahl}} * 100$ 51,25%

Abbildung 33: Checkliste ChemGes Teil 2

Anhang 8: Fragen zur Checkliste

1. Waren die Fragen klar und eindeutig formuliert?

Ja, die Fragen waren klar und eindeutig formuliert.

2. Hatten Sie den Eindruck, dass alle relevanten Themenbereiche abgedeckt wurden?

Ja, alle relevanten Themenbereiche wurden abgedeckt.

3. Haben die Fragen geholfen, den tatsächlichen Reifegrad realistisch einzuschätzen?

Ja, die Fragen haben geholfen, den tatsächlichen Reifegrad realistisch einzuschätzen.

4. Gab es inhaltliche Lücken oder fehlende Aspekte, die ergänzt werden sollten?

Nein, es gab keine inhaltlichen Lücken oder fehlenden Aspekte.

5. War der Aufbau der Checkliste übersichtlich und leicht nachvollziehbar?

Ja, der Aufbau der Checkliste war übersichtlich und leicht nachvollziehbar.

6. Wie verständlich waren die Anweisungen zur Bearbeitung?

Die Anweisungen waren verständlich, allerdings gab es Überschneidungen der Themen, da verschiedene Themen zusammenhängen.

7. War die Länge der Checkliste angemessen?

Ja, die Länge der Checkliste war angemessen.

8. War der zeitliche Aufwand für das Ausfüllen angemessen?

Ja, der zeitliche Aufwand für das Ausfüllen war angemessen.

9. Was würden Sie an der Checkliste verbessern?

Eventuell die Checkliste in Excel übertragen, um eine automatisierte Auswertung zu ermöglichen.

10. Würden Sie die Checkliste auch anderen empfehlen? Warum oder warum nicht?

Ja, die Checkliste bietet mit wenig Aufwand eine gute Einschätzung für ein System oder eine Software.

11. Hat sich Ihre Sicht auf den Reifegrad durch die Bearbeitung verändert?

Die Bearbeitung hat meine Einschätzung bestätigt.

12. Welche zusätzlichen Informationen oder Erläuterungen hätten Sie sich gewünscht?

Eine stärkere Eingrenzung durch gezieltere Fragen wäre wünschenswert, auch wenn dies den Bearbeitungsaufwand erhöhen würde.

Abbildung 34: Fragen zur Checkliste

10 Literaturverzeichnis

- Kaufmann, J. & Müller, W. (2022). Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung. Springer Vieweg.
- Marwedel, P. (2021). Eingebettete Systeme: Grundlagen Eingebetteter Systeme in Cyber-Physikalischen Systemen. Springer Vieweg.
- Mangiapane, M & Büchler, R. P. (2024). Modernes IT-Management: Methodische Kombination von IT-Strategie und IT-Reifegradmodell. Springer-Verlag.
- Haghsheno, S., Lennerts, K. & Gentes, S. (2019). 30.BBB-Assistententreffen in Karlsruhe – Fachkongress der wissenschaftlichen Mitarbeiter Bauwirtschaft| Baubetrieb | Bauverfahrenstechnik: 10. -12. Juli 2019, Institut für Technologie und Management im Baubetrieb (TMB), Karlsruher Institut für Technologie (KIT). KIT Scientific Publishing.
- Rout, T.P. (2003), ISO/IEC 15504—Evolution to an international standard. Softw. Process: Improve. Pract., 8: 27-40.
- Chang, K., Wang, W., Chen, L., Ellis, C., Hsu, C., Tsoi, A., Wang, H. (2007). Advances in Web and Network Technologies and Information Management: APWeb/WAIM 2007 International Workshops: DBMAN 2007, WebETrends 2007, PAIS 2007 and ASWAN 2007. Springer-Verlag.
- Hoffmann, D. (2008). Software-Qualität. In eXamen.press. Springer-Verlag.
- Becker, J., Knackstedt, R. & Pöppelbuß, J. Entwicklung von Reifegradmodellen für das IT-Management. Wirtsch. Inform. 51, 249–260 (2009). <https://doi.org/10.1007/s11576-009-0167-9>

Internetquellen:

- Rayner, M. (2020, 11. August). Toward a Marketing and Analytics Capability-Maturity Model - Hudson Crossing | Management Consulting for Travel Distributors. Hudson Crossing | Management Consulting For Travel Distributors. <https://www.hudsoncrossing.com/marketing-analytics-capability-maturity-model/> (Stand 17.08.2025)
- Chcomley. (o. D.). Integration des Capability Maturity Model (CMMI), Hintergrundnotizen - Azure Boards. Microsoft Learn. <https://learn.microsoft.com/de-de/azure/devops/boards/work-items/guidance/cmmi/guidance-background-to-cmmi?view=azure-devops> Stand: 04.08.2025
- BANTLEON – Über uns. <https://bantleon.de/de/unternehmen/ueber-uns/> (Stand 17.08.2025)
- BANTLEON – Geschichte. <https://bantleon.de/de/unternehmen/geschichte/> (Stand 17.08.2025)
- BANTLEON – Metallbearbeitung. <https://bantleon.de/de/branchen/metallbearbeitung/> (Stand 17.08.2025)
- BANTLEON – Maschinen- und Anlagentechnik. <https://bantleon.de/de/branchen/maschinen-anlagentechnik/> (Stand 17.08.2025)
- BANTLEON – Luftfahrt. <https://bantleon.de/de/branchen/luftfahrt/> (Stand 17.08.2025)
- BANTLEON – Medizintechnik. <https://bantleon.de/de/branchen/medizintechnik/> (Stand 17.08.2025)
- BANTLEON – Branchen im Heavy Machinery Bereich. <https://bantleon.de/de/branchen/heavy-machinery/> (Stand 17.08.2025)