



BACHELORARBEIT

Herr
Christian Klumpp

**Untersuchung der Kriterien für
Projektmanagementsoftware für kleine
Teams in der Spieleentwicklung**

Mittweida, Februar 2025

BACHELORARBEIT

Untersuchung der Kriterien für Projektmanagementsoftware für kleine Teams in der Spieleentwicklung

Autor:

Christian Klumpp

Studiengang:

Medieninformatik und Interaktives Entertainment

Seminargruppe:

MI21w2-b

Erstprüfer:

Prof. Dr.-Ing. Philipp Klimant

Zweitprüfer:

Dr. Alexander Marbach

Einreichung:

Mittweida, 03.02.2025

Verteidigung/Bewertung:

Mittweida, 2025

BACHELOR THESIS

Investigating the criteria for project management software for small teams in game development

Author:

Christian Klumpp

Course of Study:

Media Informatics and Interactive Entertainment

Seminar Group:

MI21w2-b

First Examiner:

Prof. Dr.-Ing. Philipp Klimant

Second Examiner:

Dr. Alexander Marbach

Submission:

Mittweida, 03.02.2025

Defense/Evaluation:

Mittweida, 2025

Bibliografische Beschreibung

Klumpp, Christian:

Untersuchung der Kriterien für Projektmanagementsoftware für kleine Teams in der Spieleentwicklung. – 2025. – 46 S.

Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2025.

Referat

Das Ziel dieser Arbeit ist es zu untersuchen, welche Anforderungen an Projektmanagementsoftware in der Indie Spieleentwicklung bestehen und welche Besonderheiten und Herausforderungen eine gute Projektmanagementsoftware in diesem Bereich beachten sollte. Um diese Anforderungen zu untersuchen und zu überprüfen, wird sich zunächst mit den Besonderheiten von (Software)- Projektmanagement beschäftigt, und dieses Gebiet dann zunächst auf die Spieleentwicklung und danach auf Indiestudios eingegrenzt. Als Nächstes wird sich über dieselben Eingrenzungen dem Thema über Projektmanagementsoftware genähert. Die Erkenntnisse aus beiden Richtungen werden danach in einem Anforderungsbogen für Projektmanagementsoftware in der Indie Spieleentwicklung kombiniert. Um den Anforderungsbogen dann zu überprüfen wurden Experteninterviews geführt, in denen die jeweiligen Anforderungen diskutiert und priorisiert wurden. Der daraus gewonnene Anforderungsbogen liefert erste Rückschlüsse auf die besonderen Herausforderungen an Projektmanagementsoftware in der Indie Spieleentwicklung sowie auf die Priorisierung der einzelnen Faktoren in Abhängigkeit von der Teamgröße. Außerdem konnten Zielgruppen für Projektmanagementsoftware in diesem Bereich identifiziert werden.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	IV
1 Einleitung	1
1.1 Projektmanagement Allgemein und Geschichte	1
1.2 Überblick über Projektmanagement Techniken	1
1.2.1 Traditionelles (klassisches) Projektmanagement	1
1.2.2 Agiles Projektmanagement	2
1.2.3 Vergleich: Agil vs traditionell	3
1.3 Projektmanagement als Grundlage für Erfolg	3
1.4 Aufgaben von Projektmanagement	3
2 Projektmanagement in der Spieleentwicklung	5
2.1 Genutzte Methoden	5
2.2 Warum agile Methoden genutzt werden	5
2.3 Besondere Herausforderungen an Projektmanagement in der Spieleentwicklung	6
2.4 Zusammenfassung Projektmanagement in der Spieleentwicklung	7
3 Projektmanagement in kleinen Spieleentwicklungs- und Indie-Teams	8
3.1 Definition von Indie-Teams und kleinen Spielentwicklungsteams	8
3.2 Struktur und Arbeitsweise von kleinen Spieleentwicklungsteams	8
3.3 Wichtige Faktoren des Projektmanagements für kleine Spieleentwicklungsteams	9
3.4 Herausforderungen für kleine Spieleentwicklungsteams in Bezug auf das Projektmanagement	10
4 Projektmanagementsoftware (PMS)	11
4.1 Definition von Projektmanagement-Software	11
4.2 Geschichte und Entwicklung der Projektmanagementsoftware	11
4.3 Wann eine Projektmanagement-Software eingesetzt werden sollte	11
4.4 Wichtige Faktoren für PMS	12
4.5 Risiken und Herausforderungen im Hinblick auf Projektmanagementsoftware	17
5 Die Rolle von Projektmanagement-Software in der Indie-Spielentwicklung	18
5.1 Wie ein PMS bei der Spieleentwicklung helfen kann	18
5.2 Unterschiede zu allgemeinen PMS-Anwendungen	19
5.3 Vergleich von Nutzen und Risiken für kleine Spieleentwicklungsteams	19
5.4 Sind PMS für kleine Indie-Teams nützlich?	21
5.4.1 Vorteile für Indie-Teams	21
5.4.2 Herausforderungen für Indie-Teams	22
6 Bewertungsbogen für PMS in der Indie-Spieleentwicklung	23
6.1 Grundlegende Funktionalitäten	23
6.2 Benutzerfreundlichkeit und Benutzeroberfläche	24
6.3 Technische Anforderungen	24

6.4	Kosten und Lizenzierung	25
6.5	Zusätzliche Funktionalitäten	25
6.6	Unterstützung und Schulung	26
6.7	Kulturelle und teambezogene Überlegungen	27
6.8	Schlichtheit und Minimalismus	27
6.9	Motivation und Teammoral	27
7	Überarbeitung des Bewertungsbogens durch Experteninterviews	28
7.1	Notwendigkeit zur Überarbeitung des Bewertungsbogens	28
7.2	Nutzung von Experteninterviews zur Verifizierung	28
7.3	Methodik zur Durchführung von Experteninterviews	29
7.4	Auswahl der Experten	30
7.5	Durchführung des Interviews	31
8	Ergebnisse des Interviews	33
8.1	Allgemeine Ergebnisse	33
8.1.1	Skalierende Prioritäten	33
8.1.2	Unterschiedliche Zielgruppen	33
8.1.3	Unterschiedliche Blickwinkel	34
8.2	Kriterienauswertung	35
8.2.1	Grundlegende Funktionalitäten	35
8.2.2	Benutzerfreundlichkeit und Benutzeroberfläche	37
8.2.3	Technische Anforderungen	38
8.2.4	Kosten und Lizenzierung	39
8.2.5	Zusätzliche Funktionalitäten	39
8.2.6	Unterstützung und Schulung	41
8.2.7	Kulturelle und teambezogene Überlegungen	41
8.2.8	Vereinfachung und Minimalismus	42
8.2.9	Motivation und Teammoral	42
9	Fazit und Einordnung	44
9.1	Ergebnisse für den Anforderungsbogen	44
9.1.1	Diskussion des Anforderungsbogens	44
9.2	Ergebnisse für PMS	45
9.2.1	Unterschiedliche Zielgruppen	45
9.2.2	Die Relevanz von PMS in der Indiespieleentwicklung	45
9.3	Fazit	46
Anhang		47
A	Experteninterview Skript	47
A.1	Einführung und Rahmenbedingungen	47
A.1.1	Einführung durch den Interviewer:	47
A.1.2	Kontext des Interviews:	47
A.1.3	Haftungsausschluss zur Datennutzung und -aufzeichnung:	47
A.2	Allgemeine Fragen zur Person	47
A.3	Fragen zur Überprüfung der Kriterien	48
A.3.1	Grundlegende Funktionalitäten	48

A.3.2 Benutzerfreundlichkeit und Benutzeroberfläche	49
A.3.3 Technische Anforderungen	50
A.3.4 Kosten und Lizenzierung	50
A.3.5 Zusätzliche Funktionalitäten	51
A.3.6 Unterstützung und Schulung	52
A.3.7 Kulturelle und teambezogene Überlegungen	52
A.3.8 Vereinfachung und Minimalismus	52
A.3.9 Motivation und Teammoral	53
A.4 Abschließende Fragen	53
A.5 Abschluss	53
A.5.1 Schlusswort	53
A.5.2 Erinnerung an die Datenverarbeitung:	54
B Experteninterview Transkript	55
B.1 Grundlegende Funktionalitäten	55
B.2 Benutzerfreundlichkeit und Benutzeroberfläche	62
B.3 Technische Anforderungen	66
B.4 Kosten und Lizenzierung	70
B.5 Zusätzliche Funktionalitäten	72
B.6 Unterstützung und Schulung	77
B.7 Kulturelle und teambezogene Überlegungen	79
B.8 Vereinfachung und Minimalismus	80
B.9 Motivation und Teammoral	81
B.10 Allgemeine Zitate	83
Literaturverzeichnis	85
Eidesstattliche Erklärung	90

Abbildungsverzeichnis

1.1 Wasserfallmodell <i>Quelle:</i> [Pau09]	2
1.2 Aufbau eines Kanban Boards <i>Quelle:</i> [Jen23]	2
2.1 Die 6 Entwicklungsphasen der Spieleentwicklung nach dem GDLC Modell <i>Quelle:</i> [RW13]	6
4.1 Jira: Kalender <i>Quelle:</i> [Atl24b]	12
4.2 Jira: Gantt-Diagramm <i>Quelle:</i> [Atl24a]	13
4.3 Asana: Kommunikation in Tasks <i>Quelle:</i> [Asa25d]	13
4.4 Asana: Ticketsystem <i>Quelle:</i> [Asa25c]	14
4.5 Codecks: Burndown-Diagramm <i>Quelle:</i> [Cod25a]	14
4.6 Asana: Budgetverwaltung <i>Quelle:</i> [Asa25a]	15
4.7 Codecks: Handübersicht <i>Quelle:</i> [Cod25b]	16
4.8 Asana: Dashboard <i>Quelle:</i> [Asa25b]	16
5.1 Kombination der Erkenntnisse aus mehreren Kapiteln	19
9.1 Veränderung des Anforderungsbogens durch das Experteninterview	44

1 Einleitung

1.1 Projektmanagement Allgemein und Geschichte

Projektmanagement bezeichnet eine Vielzahl von Methoden, die Wissen, Fähigkeiten, Werkzeuge und Techniken kombinieren um ein Projekt zum Erfolg zu führen. Das "Project Management Institute" (PMI) definiert Projektmanagement als :

The application of knowledge, skills, tools, and techniques to project activities to meet project requirements. Project management refers to guiding the project work to deliver the intended outcomes. [Pro21, S.4]

Dieses Vorgehen stellt sicher, dass Projekte effizient und effektiv durchgeführt werden und mit den Unternehmenszielen und den Erwartungen der Stakeholder und Zielgruppen übereinstimmen. Die Geschichte des Projektmanagements lässt sich bis ins frühe 20. Jahrhundert zurückverfolgen, vor allem während der industriellen Revolution, als groß angelegte technische Projekte strukturiertes und standardisiertes Management erforderte.[WB20] Frühe Pioniere auf dem Gebiet, wie Henry Gantt, der für das bis heute genutzte Gantt-Diagramm bekannt ist oder Frederick Taylor, der wissenschaftliche Managementprinzipien eingeführt, legten den Grundstein für die modernen Projektmanagementpraktiken. Durch größere Projekte in Branchen wie der Luft- und Raumfahrt, dem Bauwesen und dem Militär wurde der Bereich des Projektmanagements Mitte des 20. Jahrhunderts erheblich weiter entwickelt, da diese Projekte es erforderten bessere und anspruchsvollere Managementtechniken zu entwickeln.[WB20]

1.2 Überblick über Projektmanagement Techniken

Die Methoden und Techniken des Projektmanagements haben sich im Laufe der Zeit diversifiziert, um den unterschiedlichen Anforderungen von Projekten in verschiedenen Branchen gerecht zu werden. Die beiden Hauptkategorien sind das traditionelle (oder klassische) Projektmanagement und das agile Projektmanagement.

1.2.1 Traditionelles (klassisches) Projektmanagement

Das traditionelle Projektmanagement, das häufig durch das Wasserfallmodell veranschaulicht wird, folgt einem linearen und sequenziellen Ansatz[Ker13]. Bei dieser Methode wird das Projekt in verschiedene Phasen unterteilt: Initiierung, Planung, Ausführung, Überwachung und Steuerung sowie Abschluss.[Pro21] Jede Phase muss abgeschlossen sein, bevor die nächste beginnt, wobei es nur minimale Überschneidungen gibt. Die Anforderungen werden zu Beginn festgelegt, und während der Ausführungsphase sind Änderungen in der Regel nicht erwünscht. [Ker13] Ein bekanntes Beispiel für klassisches Projektmanagement ist das Wasserfallmodell aus 1.1. Bei diesem wird der Softwarezyklus als ganzes betrachtet und jeder Schritt muss erledigt werden, bevor zum nächsten vorgegangen wird. Während das Wasserfallmodell selbst zwar nur noch wenig Anwendung findet, existieren jedoch eine Vielzahl komplexer Modelle, die aus diesem hervorgegangen sind. Die Modelle des klassischen

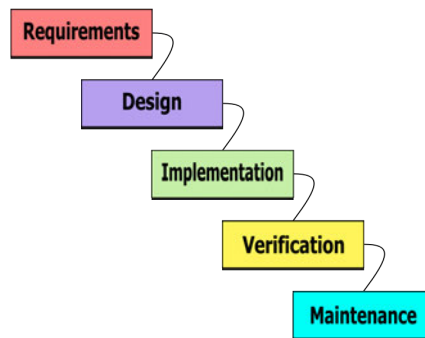


Abbildung 1.1: Wasserfallmodell *Quelle:* [Pau09]

Projektmanagements legen den Schwerpunkt auf eine gründliche Dokumentation und eine frühzeitige Planung und eignet sich daher für Projekte mit klar definierten Zielen und stabilen Anforderungen, wie z. B. Bau- oder Fertigungsprojekte.[Tur14] Es sind klare Ergebnisse und Meilensteine vorgegeben, wodurch eine unkomplizierte Verwaltung und Kontrolle ermöglicht wird.[Ker13]

1.2.2 Agiles Projektmanagement

Im Gegensatz zum traditionellen Projektmanagement verfolgt das agile Projektmanagement einen inkrementellen und iterativen Ansatz[Hig10]. Agile Methoden wie Scrum und Kanban legen den Schwerpunkt auf Flexibilität, Zusammenarbeit mit dem Kunden und die schnelle Lieferung von funktionalen Produktionsschritten[Ken20] Projekte werden in kleine Iterationen oder Sprints unterteilt, die in der Regel zwei bis vier Wochen dauern und es den Teams ermöglichen, sich schnell an Änderungen anzupassen und das Produkt kontinuierlich zu verbessern.[Hig10] Ein häufig genutztes

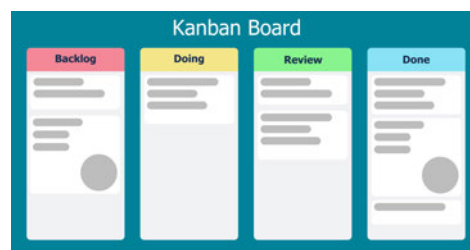


Abbildung 1.2: Aufbau eines Kanban Boards *Quelle:*[Jen23]

Mittel beim Arbeiten mit Kanban ist das sog. "Kanban Board" welches nach dem Schema aus 1.2 aufgebaut sein kann. Auf diesem Board werden die persönlichen Aufgaben des Entwicklers verwaltet und können in unterschiedliche Kategorien eingeteilt werden. Auf diese Weise sind auch Zwischenabnahmen und Prüfungen möglich. Agiles Vorgehen entstand in der Softwareentwicklungsbranche als Reaktion auf die Grenzen traditioneller Modelle bei der Bewältigung sich ändernder Anforderungen und der Förderung von Innovationen.[Lar04] Das 2001 veröffentlichte Agile Manifest betont Individuen und Interaktionen gegenüber Prozessen und Tools, funktionierende Software gegenüber umfassender Dokumentation, Zusammenarbeit mit dem Kunden gegenüber Vertragsverhandlungen und Reagieren auf Veränderungen gegenüber dem Befolgen eines Plans.[Ken13]

1.2.3 Vergleich: Agil vs traditionell

Der grundlegende Unterschied zwischen traditionellen und agilen Methoden liegt in ihrem Umgang mit Veränderungen und Unsicherheit. Das traditionelle Projektmanagement legt den Schwerpunkt auf Vorhersehbarkeit und Kontrolle, was es für Projekte mit klaren, unveränderlichen Anforderungen geeignet macht.[Tur14] Agile Methoden sind offen für Veränderungen und fördern eine regelmäßige Neubewertung der Projektziele und Anpassungen an neue Informationen oder Kundenfeedback.[Hig10] In der Softwareentwicklung beispielsweise ermöglichen agile Praktiken eine häufige Neubewertung der Nutzerbedürfnisse und technologischen Möglichkeiten, sodass die Teams relevantere und hochwertigere Produkte liefern können.[Ken20] Traditionelle Modelle können in solchen Umgebungen aufgrund ihrer Starrheit und ihres Widerstands gegen Veränderungen Schwierigkeiten bereiten.[Tur14]

1.3 Projektmanagement als Grundlage für Erfolg

Ein effektives Projektmanagement ist von entscheidender Bedeutung für die Erreichung der Projektziele innerhalb der Grenzen von Umfang, Zeit, Kosten und Qualität.[Pro21] Aus diesen Gründen nutzen nahezu alle größeren und kleineren Unternehmen heutzutage Methoden des Projektmanagements.[Tur14] Projektmanagement bietet einen strukturierten Rahmen, der es Organisationen ermöglicht, ihre Projekte effizient zu planen, durchzuführen und zu kontrollieren. Unternehmen setzen Projektmanagement ein, um die Produktivität zu steigern, Risiken zu bewältigen und sicherzustellen, dass die Projekte mit den strategischen Unternehmenszielen übereinstimmen.[MP14] Projektmanagement erleichtert die klare Kommunikation zwischen den Beteiligten, fördert die Teamarbeit und hilft bei der Ressourcenoptimierung.[JM15] Durch die Festlegung von Zielen und Meilensteinen können die Teams ihre Bemühungen konzentrieren und den Fortschritt effektiv messen.[Pro21] Darüber hinaus ermöglicht es ein proaktives Risikomanagement, indem es potenzielle Probleme frühzeitig identifiziert und Strategien zur Risikominderung implementiert.[HS20] Ein unzureichendes Projektmanagement kann zu Kostenüberschreitungen, Terminüberschreitungen und dem Nichterreichen von Projektzielen führen, was sich negativ auf den Ruf und das finanzielle Ansehen einer Organisation auswirkt.[Fly14] So ergab eine Studie der Standish Group, dass nur 31% der Projekte erfolgreich und innerhalb des Zeit- und Kostenrahmens abgeschlossen wurden, was häufig auf unzureichende Projektmanagementpraktiken zurückzuführen ist.[sta15] Im heutigen wettbewerbsorientierten und schnelllebigen Geschäftsumfeld ist die Fähigkeit, Projekte effektiv zu managen, ein wichtiges Kriterium zur Verbesserung der Wettbewerbsfähigkeit.[Pro20] Sie ermöglicht es den Unternehmen, innovativ zu sein, sich an Marktveränderungen anzupassen und den Kunden effizient einen Mehrwert zu bieten. Da die Projekte immer komplexer werden, steigt die Nachfrage nach qualifizierten Projektmanagement-Fachleuten weiter an, was die Bedeutung von Projektmanagement unterstreicht.

1.4 Aufgaben von Projektmanagement

Die Untersuchung der Aufgaben von allgemeinem Projektmanagement lässt bereits Schlüsse auf mögliche Kategorien für Anforderungen zu, bei denen eine Projektmanagementsoftware den Entwicklungsprozess unterstützen sollte:

Unterstützung bekannter Prozesse und Verfahren Im Projektmanagement gibt es bereits genormte und getestete agile und klassische Konzepte, die das Vorgehen vereinfachen.

Überwachung von Ressourcen Aufgabe des Projektmanagements ist es Ressourcen wie Zeit, Umfang oder Kosten zu verwalten und zuzuweisen.

Vereinfachen der Kommunikation Projektmanagement soll die Kommunikation zwischen allen beteiligten Akteuren vereinfachen und optimieren.

Förderung und Verbesserung der Teamarbeit Dadurch, dass Softwareprojekte meist von mehreren Personen im Team entwickelt werden, sollte das Projektmanagement diese Art der Arbeit fördern und vereinfachen.

Risikomanagement Projektmanagement sollte Risiken für und während des Entwicklungsprozesses abschätzen können und bei etwaigen Problemen unterstützen

Unterstützung der Wettbewerbsfähigkeit Projektmanagement sollte Unternehmen dabei unterstützen, sich an Marktveränderungen anzupassen und die Wettbewerbsfähigkeit des Unternehmens fördern.

2 Projektmanagement in der Spieleentwicklung

Projektmanagement spielt eine entscheidende Rolle bei der erfolgreichen Entwicklung von Videospielen, einer Branche, die durch rasante technologische Fortschritte und sich verändernde Verbrauchererwartungen geprägt ist. Die Einzigartigkeit der Spieleentwicklung erfordert spezielle Projektmanagementansätze, die dem dynamischen, interdisziplinären und kreativen Umfeld gerecht werden.[\[Mar+19, S.1-2\]](#) Dieses Kapitel befasst sich mit den Methoden, die üblicherweise im Projektmanagement der Spieleentwicklung eingesetzt werden, sowie mit den besonderen Herausforderungen, die sich in diesem Bereich stellen.

2.1 Genutzte Methoden

In der Spieleentwicklung werden vor allem agile Projektmanagement Methoden genutzt, die jedoch je nach Entwickler und Spiel angepasst werden. Während durch dieses Vorgehen viele unterschiedliche Praktiken vorhanden sind, basieren diese jedoch meist auf agilen Konzepten.[\[SN08, S.145-146\]](#) Agile Methoden sind aufgrund ihrer Flexibilität und Anpassungsfähigkeit zur Basis des Projektmanagements in der Spieleentwicklung geworden.[\[KL13, S.1\]](#) Im Gegensatz zu traditionellen Projektmanagement-Ansätzen sind agile Methoden offen für Veränderungen und fördern die iterative Entwicklung, was sich gut mit den kreativen Prozessen der Spieleentwicklung vereinbaren lässt.

Scrum Scrum ist eine der am weitesten verbreiteten agilen Methoden in der Spieleindustrie.[\[Cli10\]](#) Dabei wird die Arbeit in Sprints unterteilt, die, in der Regel, zwei bis vier Wochen lange Zeiträume sind, in denen bestimmte Ziele erreicht werden.[\[Rob04\]](#) Dieser Ansatz ermöglicht es den Teams, stufenweise Fortschritte zu erzielen, Feedback zu erhalten und notwendige Anpassungen zeitnah vorzunehmen.

Kanban Kanban ist eine weitere agile Methode, die in der Spieleentwicklung eingesetzt wird und sich auf die Visualisierung der Arbeit, die Begrenzung der laufenden Arbeit und die Verbesserung des Arbeitsflusses konzentriert.[\[Rob04, S.1\]](#) Es ist besonders nützlich für Teams, die Flexibilität bei der Aufgabenverwaltung ohne die festen Zeitrahmen von Sprints benötigen.

2.2 Warum agile Methoden genutzt werden

Die Bevorzugung agiler Methoden in der Spieleentwicklung ist auf mehrere wesentliche Faktoren zurückzuführen:

Umgang mit Veränderungen Bei Spieleentwicklungsprojekten kommt es häufig zu Änderungen bei Design, Funktionen und Technologien. Agile Methoden sind darauf ausgelegt, solche Änderungen zu ermöglichen, ohne den Zeitplan oder die Ziele des Projekts zu gefährden.[\[KL13\]](#)

Verbesserung der Zusammenarbeit Agile Methoden fördern die Zusammenarbeit zwischen fachübergreifenden Teams[\[KFH13, S.124\]](#), was bei der Spieleentwicklung, bei der Künstler, Designer, Programmierer und andere Spezialisten zusammenarbeiten müssen, von entscheidender Bedeutung ist.

Schrittweise Entwicklung von Projekten Durch die schrittweise Entwicklung von Teilen des Spiels können die Teams Konzepte frühzeitig testen und validieren, wodurch das Risiko späterer umfangreicher Nacharbeiten verringert wird.[Mit+16] In der Grafik 2.1 ist der typische Entwicklungs-

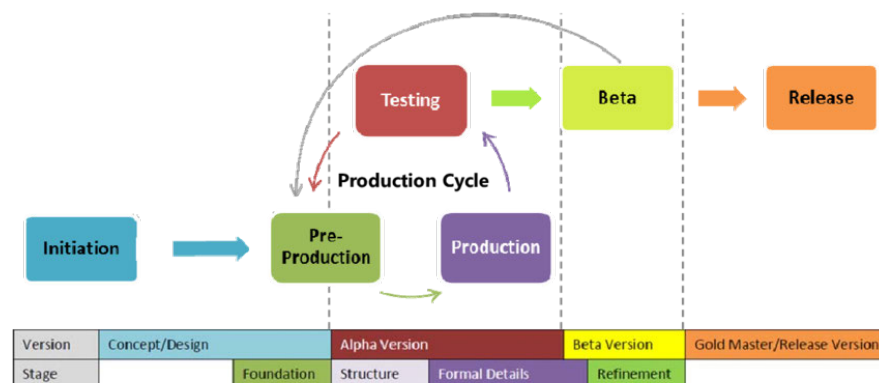


Abbildung 2.1: Die 6 Entwicklungsphasen der Spieleentwicklung nach dem GDLC Modell *Quelle:* [RW13]

zyklus für die Spieleentwicklung dargestellt. Aufgrund der vielen kreativen Prozesse kann das Ziel zu Beginn des Projekts nicht klar definiert werden, wodurch klassische Projektmanagement Modelle auf Probleme stoßen. Andererseits können iterative Modelle die häufige Tests, Anpassung der Ziele und des Vorgehens ermöglichen, wie es bei der agilen Entwicklung der Fall ist, den Entwicklungsprozess maßgeblich unterstützen.

2.3 Besondere Herausforderungen an Projektmanagement in der Spieleentwicklung

Hohe Ungewissheit und wechselnde Anforderungen Spieleentwicklungsprojekte sind durch ein hohes Maß an Unsicherheit gekennzeichnet. Designelemente, technologische Möglichkeiten und Markttrends können sich schnell ändern und sich auf den Projektumfang und die Ziele auswirken.[TSC05, S.103] Die Bewältigung dieser Veränderungen erfordert ein flexibles Projektmanagement, das sich an neue Informationen anpassen und die Pläne entsprechend anpassen kann.

Interdisziplinäre Teams und Zusammenarbeit Die Entwicklung von Spielen erfordert die Zusammenarbeit verschiedener Fachrichtungen, darunter Programmierung, Kunst, Design, Tontechnik und Schreiben von Geschichten.[Des20, K.7] Die Koordination dieser verschiedenen Bereiche erfordert effektive Kommunikation und Teamarbeit, um sicherzustellen, dass alle Komponenten nahtlos ineinandergreifen. Projektmanager müssen die Zusammenarbeit zwischen diesen Disziplinen erleichtern und oft die Kluft zwischen technischen und kreativen Teammitgliedern überbrücken.[Mus+10] Diese Interdisziplinarität erhöht die Komplexität des Projektmanagements und erfordert Strategien, die den Zusammenhalt und das gemeinsame Verständnis fördern.

Die Bedeutung von Kreativität und Innovation Kreativität ist das Herzstück der Spieleentwicklung. Projekte müssen ein Umfeld schaffen, in dem innovative Ideen gedeihen können, während gleichzeitig die Projektfristen und -budgets eingehalten werden.[CS07] Das Gleichgewicht zwischen kreativer Freiheit und Projektbeschränkungen ist eine große Herausforderung für Projektmanager.

Die Förderung von Tests und iterativen Entwicklungsprozessen kann zu ansprechenderen und originelleren Spielen führen, birgt aber auch Risiken in Bezug auf die Ausweitung des Projektumfangs und das Zeitmanagement.[SFT14] Projektmanager müssen das richtige Gleichgewicht finden, um die Kreativität zu nutzen, ohne die Projektziele zu gefährden.

Zeitliche Beschränkungen und Fristen Der Wettbewerb in der Spieleindustrie führt oft zu strengen Fristen für die Produktveröffentlichung. Eine rechtzeitige Veröffentlichung ist von entscheidender Bedeutung, um Marktchancen zu nutzen und die Erwartungen von Herausgebern und Spielern zu erfüllen.[EW13] Verzögerungen können zu erhöhten Kosten und entgangenen Einnahmen führen, was das Zeitmanagement zu einem kritischen Aspekt des Projektmanagements in der Spieleentwicklung macht. Projektmanager müssen den Fortschritt effektiv planen und überwachen, um sicherzustellen, dass die Meilensteine eingehalten werden.

Technische Komplexität Moderne Spiele sind technologisch komplex und stoßen oft an die Grenzen dessen, was aktuelle Hardware und Software leisten können.[Jus10] Der Umgang mit technischen Risiken wie Leistungsproblemen, Kompatibilität und neuen Technologien ist von entscheidender Bedeutung. Die Projektmanager müssen eng mit den technischen Teams zusammenarbeiten, um potenzielle Herausforderungen frühzeitig zu erkennen und die Ressourcen entsprechend zuzuweisen. Dazu gehört auch, dass die Teammitglieder über die notwendigen Fähigkeiten und Werkzeuge verfügen, um den technischen Anforderungen gerecht zu werden.

2.4 Zusammenfassung Projektmanagement in der Spieleentwicklung

Projektmanagement in der Spieleentwicklung erfordert spezielle Ansätze, um die einzigartigen Herausforderungen zu bewältigen. Aus vorangegangenen Untersuchungen lassen sich folgende Anforderungen ableiten:

Nutzung agiler Methoden Agile Methoden werden aufgrund ihrer Flexibilität und der Unterstützung einer iterativen Entwicklung bevorzugt, was der kreativen und dynamischen Natur der Spieleentwicklung entgegenkommt.

Iterative Entwicklung Spiele benötigen einen iterativen Entwicklungsprozess, bei dem Zwischenergebnisse getestet und evaluiert werden können. Das Projektmanagement sollte diese Iterationen ermöglichen und sich an nötige Änderungen oder neue Zieldefinitionen anpassen.

Interdisziplinäre Teamarbeit In der Spieleentwicklung arbeiten Personen aus unterschiedlichen kreativen und informatischen Bereichen zusammen. Das Projektmanagement muss diese Zusammenarbeit ermöglichen und verbessern.

Zeitverwaltung Durch Wettbewerb und zeitliche Vorgaben von Publishern existieren Fristen und Termine, die das Entwicklerstudio oder die einzelnen Entwickler einhalten müssen. Das Projektmanagement sollte die jeweilige Person bei der Einhaltung der Termine unterstützen.

Risikomanagement Durch die Leistungs- und Qualitätsanforderungen an moderne Spiele müssen Risiken im Entwicklungsprozess schnell durch das Projektmanagement erkannt und gelöst werden.

3 Projektmanagement in kleinen Spieleentwicklungs- und Indie-Teams

Kleine Spieleentwicklungsteams, die häufig auch als Indie-Teams bezeichnet werden, spielen eine wichtige Rolle in der Spieleindustrie, da sie innovative und einzigartige Spiele außerhalb des kommerziellen Mainstream-Marktes produzieren.[Joh16] Um die Erfolgsbedingungen und Herausforderungen dieser Teams zu identifizieren, ist es wichtig zu verstehen, wie das Projektmanagement in diesen Teams funktioniert.

3.1 Definition von Indie-Teams und kleinen Spielentwicklungsteams

Ein Indie-Team, kurz für Independent Team, ist in der Regel ein Spieleentwicklungsteam, das ohne die finanzielle Unterstützung eines großen Verlags arbeitet und die kreative Kontrolle über seine Projekte behält.[FA24, S.2] Nach Angaben der International Game Developers Association (IGDA) zeichnen sich Indie-Teams durch ihre Unabhängigkeit bei der Entscheidungsfindung und bei den Finanzierungsquellen aus.[IGD25] Die Definition, was genau ein kleines Team ist, kann variieren, aber im Allgemeinen besteht ein kleines Spieleentwicklungsteam aus weniger als 20 Mitgliedern. Einige Definitionen betrachten Teams mit bis zu 50 Mitgliedern als klein, aber Indie-Teams haben häufig deutlich weniger Mitarbeiter, manchmal sogar nur 2 bis 10 Personen. Für diese Arbeit wird für die große kleiner Entwicklerteams aus diesem Grund die Definition für Very Small Entities (VSEs)“ der ISO-29110 verwendet:

For the purpose of ISO/IEC 29110, a Very Small Entity (VSE) is an enterprise, an organization, a department or a project having up to 25 people. [Int24]

3.2 Struktur und Arbeitsweise von kleinen Spieleentwicklungsteams

Folgende Faktoren sind nach den vorliegenden Quellen entscheidend für die Entwicklung von Spielen in Indiestudios:

Flache Organisationsstruktur Kleine Spieleentwicklungsteams haben oft eine flache Organisationsstruktur mit nur wenigen Hierarchieebenen.[Kor21] Diese Struktur ermöglicht eine offene Kommunikation und schnelle Entscheidungsfindungen.

Multidisziplinäre Rollen Teammitglieder in kleinen Spieleentwicklungsteams übernehmen aufgrund der begrenzten Ressourcen häufig mehrere Rollen.[KLS13, S.1] So kann beispielsweise ein Programmierer auch Aufgaben des Projektmanagements übernehmen oder ein Designer kann sich um das Marketing kümmern.

Agile und Lean Management Kleine Teams setzen oft agile und lean Managementmethoden ein, um die Effizienz zu maximieren und sich schnell an Änderungen anzupassen.[AAB21]

3.3 Wichtige Faktoren des Projektmanagements für kleine Spieleentwicklungsteams

Die folgende Liste fasst die Erkenntnisse für wichtige Faktoren für Projektmanagement aus 1.4, 2.4 und den Erkenntnissen von Projektmanagement in Indiestudios aus diesem Kapitel zusammen.

- 1. Ressourcenmanagement** Die effektive Zuweisung und Verwaltung der begrenzten Ressourcen ist für kleine Teams entscheidend.
- 2. Kommunikation und Kollaboration** Die Aufrechterhaltung klarer und direkter Kommunikationskanäle ist unerlässlich, um die Bemühungen der Teammitglieder zu koordinieren, die möglicherweise an mehreren Stellen gleichzeitig tätig sind.
- 3. Flexibilität und Anpassungsfähigkeit** Kleine Teams müssen flexibel sein, um sich an veränderte Umstände anpassen zu können, z. B. an wechselnde Markttrends, technologische Neuerungen oder kreative Änderungen.
- 4. Risikomanagement** Die Identifizierung und Abfederung von Risiken ist von entscheidender Bedeutung, da kleine Teams möglicherweise nicht über die Ressourcen verfügen, um größere Rückschläge abzufangen.
- 5. Prioritätensetzung und Umfangsmanagement** Die Konzentration auf die Kernfunktionen und die Verwaltung des Projektumfangs helfen, eine Überforderung zu vermeiden .
- 6. Zeitmanagement** Ein effizientes Zeitmanagement ist notwendig, um Projekttermine und Meilensteine einzuhalten.
- 7. Qualitätssicherung** Die Aufrechterhaltung von Qualitätsstandards ist eine wesentliche Voraussetzung für die Wettbewerbsfähigkeit auf dem Markt, selbst bei begrenzten Ressourcen .
- 8. Marketing und Community Engagement** Das Projektmanagement kann die Bemühungen unterstützen, eine Community um das Spiel herum aufzubauen, was besonders für Indie-Teams wichtig ist.
- 9. Finanzielle Verwaltung** Eine wirksame Finanzplanung und Finanzaufsicht sind entscheidend, um sicherzustellen, dass das Projekt wirtschaftlich tragfähig bleibt .
- 10. Motivation und Teamgeist** Es ist wichtig, die Motivation des Teams aufrechtzuerhalten, insbesondere wenn es mit Herausforderungen konfrontiert wird, die bei der Entwicklung von Indie-Spielen üblich sind.

3.4 Herausforderungen für kleine Spieleentwicklungsteams in Bezug auf das Projektmanagement

Nach der Untersuchung, wie Projektmanagement bei der Entwicklung von Spielen in Indie Studios helfen kann, soll der folgende Abschnitt die besonderen Risiken die Projektmanagement in diesem Bereich mit sich bringt, als Ergänzung der positiven Kriterien aus dem vorherigen Kapitel, beleuchten.

Begrenzte Ressourcen Kleine Teams arbeiten oft mit knappen Budgets und begrenztem Personal.[[Ala14](#)] Diese Einschränkung kann die Umsetzung umfassender Projektmanagementpraktiken erschweren.

Sich überschneidende Rollen Teammitglieder können durch Multitasking überlastet werden, was zu Burnout oder verminderter Effizienz führen kann.[[Bar+22](#), S.46-58]

Fehlende formale Prozesse Informelle oder Ad-hoc-Praktiken des Projektmanagements können zu verpassten Terminen oder übersehenen Aufgaben führen.[[Bor+20](#)]

Aufmerksamkeit auf dem Markt Mit größeren Studios um die Aufmerksamkeit des Marktes zu konkurrieren, ist eine Herausforderung, die effektive Marketingstrategien erfordert.[[Sil17](#)]

4 Projektmanagementsoftware (PMS)

Projektmanagementsoftware, im Folgenden mit PMS abgekürzt, sind Softwareanwendungen zur Unterstützung von Projektmanagern und -teams bei der Planung, Durchführung und Kontrolle von Projektaktivitäten, um bestimmte Ziele zu erreichen und bestimmte Kriterien zu erfüllen. PMS bietet Werkzeuge für die Terminplanung, Ressourcenzuweisung, Zusammenarbeit, Kommunikation, Dokumentation und Budgetverwaltung und verbessern die Effizienz und Effektivität von Projektmanagementprozessen.[Kat20]

4.1 Definition von Projektmanagement-Software

Projektmanagementsoftware kann als Software definiert werden, die die Planung, den Zeitplan, die Ressourcenzuweisung, die Ausführung, die Verfolgung und die Protokollierung von Projekten unterstützt. Sie erleichtert das Management komplexer Projekte, indem sie eine zentralisierte Plattform bereitstellt, auf der alle projektbezogenen Informationen für die Beteiligten zugänglich sind.[Pag89]

4.2 Geschichte und Entwicklung der Projektmanagementsoftware

Die Ursprünge der Projektmanagement-Software gehen auf die 1950er und 1960er Jahre zurück, als Projektmanagementtechniken wie die Critical Path Method (CPM) und die Program Evaluation Review Technique (PERT) entwickelt wurden. Diese Techniken erforderten komplexe Berechnungen, die anfangs manuell oder mithilfe von rudimentären Rechenmaschinen durchgeführt wurden. In den 1960er und 1970er Jahren ermöglichte das Aufkommen von Großrechnern die Entwicklung früher PMS-Lösungen, die aufgrund ihrer hohen Kosten und Komplexität vor allem von großen Organisationen eingesetzt wurden. Diese Systeme wurden individuell entwickelt und erforderten spezialisiertes Personal für den Betrieb. Die 1980er Jahre markierten einen bedeutenden Wendepunkt mit der Einführung von Personal Computern. Software wie Microsoft Project, das 1984 erstmals auf den Markt kam, machte PMS für eine größere Anzahl von Unternehmen zugänglich. Diese Desktop-Anwendungen boten benutzerfreundlichere Schnittstellen und konnten ohne spezielle Schulung bedient werden. In den späten 1990er und frühen 2000er Jahren kamen webbasierte PMS auf, die den Fernzugriff und die Zusammenarbeit über das Internet ermöglichten.[MTL99] Das Cloud Computing revolutionierte die PMS weiter, indem es skalierbare, abonnementbasierte Modelle anbot, die die Anschaffungskosten und die Wartungsanforderungen reduzierten. In den letzten Jahren hat sich die PMS weiterentwickelt, um agile Projektmanagement-Methoden zu unterstützen und Funktionen zu integrieren, die die iterative Entwicklung, Zusammenarbeit und Flexibilität erleichtern.[BR23] Die Tools umfassen nun häufig Funktionen wie Kanban-Boards, Sprint-Planung und Echtzeit-Kommunikationskanäle.

4.3 Wann eine Projektmanagement-Software eingesetzt werden sollte

Komplexe Projektkoordination Eine PMS ist besonders wichtig für die Verwaltung komplexer Projekte, die mit einer Vielzahl von Aufgaben, Ressourcen und Akteuren durchgeführt werden.[Bas07] Es hilft dabei, Projekte in überschaubare Komponenten zu unterteilen, Aufgaben zuzuweisen und den Fortschritt zu verfolgen.

Ressourcenmanagement Die effektive Zuweisung und Nutzung von Ressourcen, einschließlich Personal, Ausrüstung und Material, wird durch eine PMS erleichtert. [Cha13] Es stellt sicher, dass die Ressourcen optimal zugewiesen und potenzielle Probleme frühzeitig erkannt werden.

Zeitmanagement Eine PMS hilft bei der Erstellung von realistischen Zeitplänen und Zeitvorgaben und unterstützt die Projektmanager bei der Festlegung von Meilensteinen und Terminen. Außerdem ermöglicht es die Überwachung des Fortschritts im Hinblick auf den Zeitplan und die Anpassung der Pläne, falls erforderlich. [Pag89]

Kollaboration und Kommunikation Für Teams, die geografisch verteilt sind oder fachübergreifend zusammenarbeiten, bietet PMS Plattformen für Kommunikation, Dateiaustausch und Echtzeit-Updates. [FT14] Dies verbessert die Koordination und verringert Missverständnisse.

Risikomanagement Die Identifizierung, Bewertung und Abfederung von Risiken sind wichtige Bestandteile des Projektmanagements. Ein PMS enthält Tools für das Risikomanagement, die den Teams helfen, potenzielle Probleme vorherzusehen und Notfallpläne zu entwickeln.

Einhaltung von Vorschriften und Dokumentation Eine PMS hilft bei der Pflege der Dokumentation und der Protokolle, die für die Einhaltung von Industriestandards und Vorschriften erforderlich sind. Es gewährleistet Transparenz und Verantwortlichkeit während des gesamten Projektlebenszyklus.

4.4 Wichtige Faktoren für PMS

Im Folgenden werden die gewonnenen Erkenntnisse über PMS in einer Liste an Anforderungen zusammengefasst und Beispiele für die Umsetzung dieser in aktuell genutzter PMS gegeben.

1. Verbesserte Planung und Terminierung PMS bietet zuverlässige Werkzeuge für die Erstellung detaillierter Projektpläne, die Festlegung von Meilensteinen und die Terminierung von Aufgaben. Dies verbessert die Fähigkeit, Zeitpläne genau vorherzusagen und Ressourcen effektiv zuzuweisen.

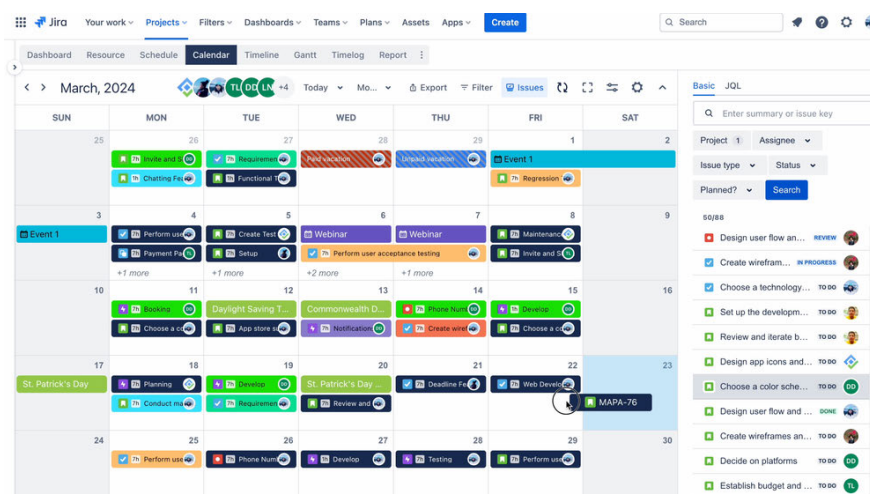


Abbildung 4.1: Jira: Kalender Quelle: [Atl24b]

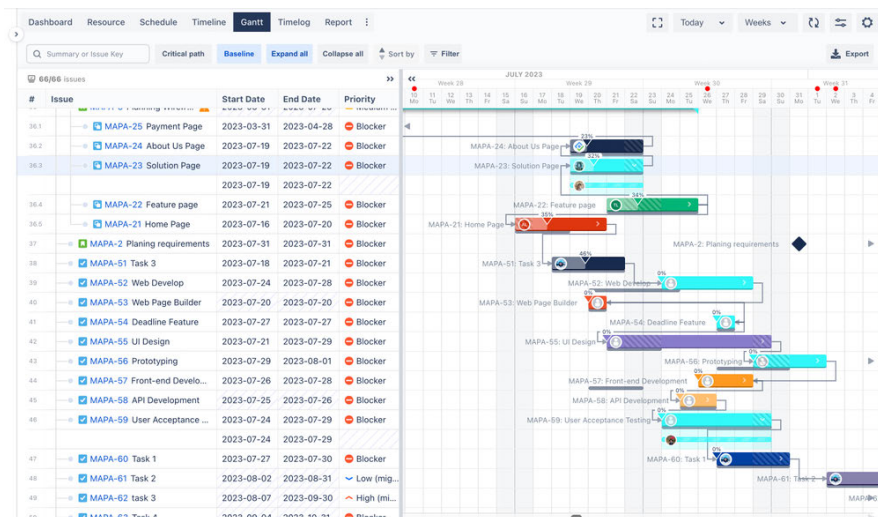


Abbildung 4.2: Jira: Gantt-Diagramm Quelle: [Atl24a]

Eine häufig genutzte PMS Lösung "Jira" bietet neben Features wie der automatischen Termin- und Abgabeverwaltung für Aufgaben auch Möglichkeiten sich die aktuelle Planung grafisch zu verdeutlichen. Hierfür werden einstellbare Kalenderansichten, die sich an Projekte/Aufgabe/Personen anpassen lassen, zur Verfügung gestellt. 4.1 Zudem können auch im Projektmanagement standardisierte Ansichten, wie Gantt-Diagramme 4.2 erstellt werden, welche eine Überwachung aller laufenden Aufgaben und Tasks ermöglichen und es vereinfachen den gesamten Projektverlauf zu überblicken.

2. Verbesserte Zusammenarbeit Funktionen wie gemeinsam genutzte Arbeitsbereiche, Messaging und die gemeinsame Bearbeitung von Dokumenten fördern die Teamarbeit und die Kommunikation zwischen den Teammitgliedern. Dies ist besonders für verteilte Teams von Vorteil.

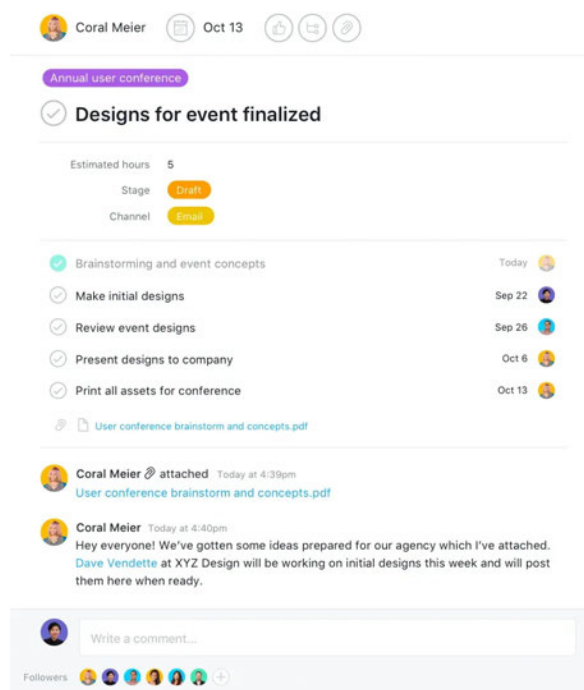


Abbildung 4.3: Asana: Kommunikation in Tasks Quelle: [Asa25d]

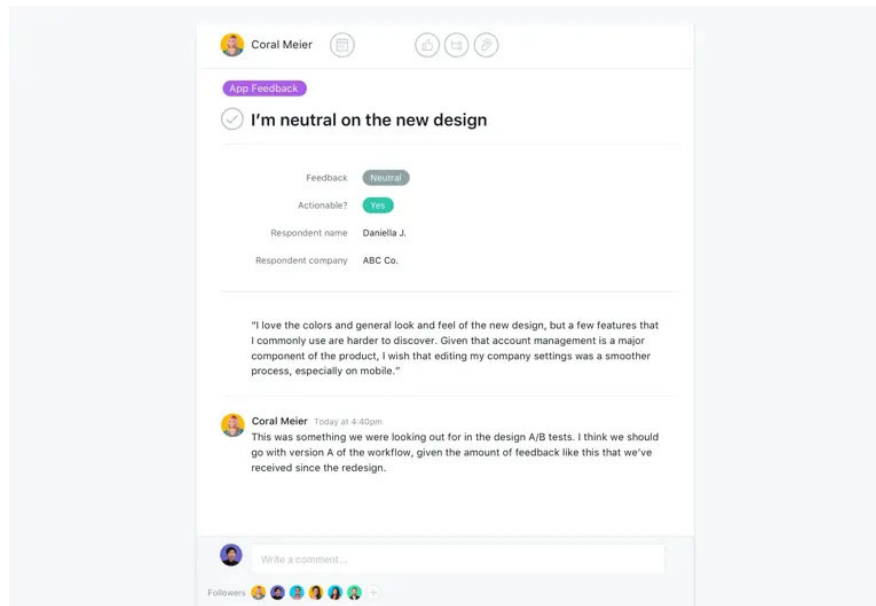


Abbildung 4.4: Asana: Ticketsystem *Quelle:* [Asa25c]

Eine der führenden Lösungen im Bereich PMS ist "Asana". Um die Zusammenarbeit zu fördern, bietet Asana Kommunikation zwischen den Entwicklern innerhalb der PMS an, ermöglicht das Hochladen von Dokumenten und Bildern entsprechend zu den geplanten Aufgaben oder Meetings 4.3, sodass direkte themenbezogene Kommunikation gefördert wird. Zudem ermöglicht Asana auch Kommunikation mit Stakeholdern, Auftraggebern oder einer Community über eine Ticketverwaltung 4.4. Auch hier findet die Kommunikation immer nah am Problem statt und ein Entwickler, der eine Aufgabe oder ein Ticket neu zugeteilt bekommt, kann so die bisherige Kommunikation nachvollziehen.

3. Ressourcen-Optimierung Durch die Übersicht über die Ressourcenzuweisung und -verfügbarkeit hilft PMS, eine Über- oder Unterbelegung von Ressourcen zu vermeiden.

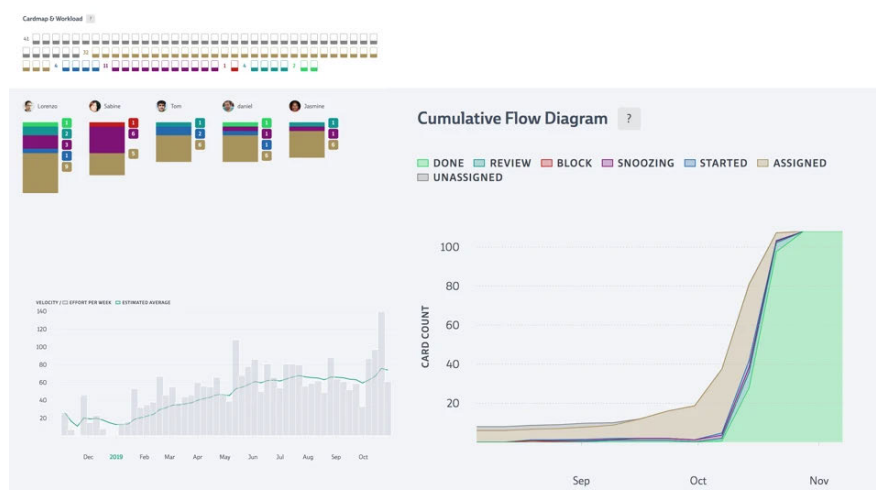


Abbildung 4.5: Codecs: Burndown-Diagramm *Quelle:* [Cod25a]

”Codecks” ist eine PMS Lösung, die sich auf die Verwaltung von Spieleentwicklungsprojekten fokussiert und einen starken Gamifikation-Ansatz hat. Für die Überwachung der Ressourcenverteilung bietet Codecks unter anderem ”Burndown-Diagramme” an 4.5, die vor allem in agilem Projektmanagement Anwendung finden. Auf das Projekt bezogen geben diese Diagramme Aufschluss darüber, ob Sprint- oder Projektziele erreicht werden, auf die Mitarbeiter bezogen kann so kontrolliert werden, wo Über- oder Unterbelastungen zu erwarten sind.

4. Budget-Verwaltung Mit einem PMS lassen sich die Projektkosten und -ausgaben im Vergleich zum Budget verfolgen, was zur Kontrolle der Ausgaben und zur Verbesserung des Finanzmanagements beiträgt.

Task name	Assignee	Due date	Estimated Spend	Actual Spend	Department	Approval stage
Budget tracking						
Update checkout page design 5 h	Daniela Var...	Oct 31				
Contract UX writer: 10 hours 1 h	Avery Lomax	Sep 15 – Oct 15	\$750		Design	Approved
In-house designer: 40 hours	Kabir Madan	Sep 15 – Oct 15	\$4,000		Design	Approved
Contract front-end engineer: 40 hours	Kat Mooney	Oct 1 – 31	\$4,000		Engineering	Approved
UX research study	Avery Lomax	Wednesday	\$500		Design	Ready for Review
Legal review	Blake Pham	Oct 27	\$500		Legal	Ready for Review
Update saved payment functionality 5 h	Daniela Var...	Oct 31				
Contract back-end engineer: 40 hours	Kat Mooney	Oct 1 – 31	\$4,000		Engineering	Approved
Privacy training for engineer	Kat Mooney	Wednesday	\$300		Legal	Changes Needed
Legal review	Blake Pham	Oct 27	\$500		Legal	Approved
Privacy review	Blake Pham	Oct 27	\$500		Legal	Approved
Desk space for Kat	Daniela Var...	Oct 1 – 31	\$2,000		Engineering	Ready for Review
Contingency fund	Daniela Var...	Sep 1 – Oct 31	\$1,750		Design	Approved
SUM			\$18,750			

Abbildung 4.6: Asana: Budgetverwaltung Quelle: [Asa25a]

Asana bietet für die Budgetkontrolle vorformatierte Listen an, in denen alle Kostenpunkte aufgelistet und mit Status versehen werden können 4.6, auf diese Weise können alle Kostenpunkte an einem Ort überblickt und kalkuliert werden. Zudem ermöglicht Asana den Export der Daten in weitere Programme wie zum Beispiel ”Excel”, sodass mit den Daten, falls nötig, auch extern weiter gearbeitet werden kann.

5. Risikominderung Werkzeuge zur Risikoidentifizierung und -analyse ermöglichen es den Projektmanagern, potenzielle Herausforderungen vorbeugend anzugehen. Das Risikomanagement greift hierbei an allen Stellen im Projekt an und gibt Warnungen an autorisierte Personen heraus, wenn zum Beispiel ein Mitarbeiter überlastet wird, die Budgetverwaltung überschritten oder Meilensteine nicht eingehalten werden können. Solche Benachrichtigungen für interne Analysen bieten alle modernen PMS Lösungen.

6. Erhöhte Transparenz und Verantwortlichkeit Da alle Projektaktivitäten dokumentiert und zugänglich sind, erhöht ein PMS die Transparenz und macht es einfacher, den Fortschritt zu überwachen und die Teammitglieder zur Verantwortung zu ziehen.

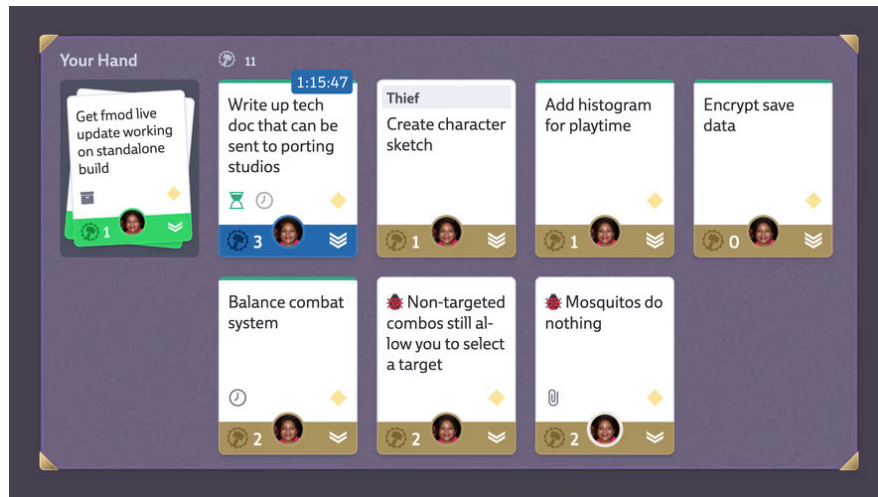


Abbildung 4.7: Codecs: Handübersicht *Quelle:* [Cod25b]

Beim Gamifikation-Ansatz von Codecs werden alle Aufgaben zu Karten, Aufgaben an denen der Entwickler aktuell arbeitet oder Aufgaben, die bis zum nächsten Sprint erfüllt werden müssen, hat der Entwickler auf seiner Hand, erledigte im Ablagestapel 4.7. Auf diese Weise hat der Entwickler eine klare Übersicht über alle aktuell für ihn relevanten Aufgaben, kann schnell überblicken, welche Aufgaben welchen Status haben und wann diese jeweils fällig sind. Dadurch, dass eine Aufgabe immer genau einer Person zugeordnet ist, ist transparent nachvollziehbar, wo die jeweilige Zuständigkeit liegt.

7. Berichte und Analysen PMS erstellt Berichte und Analysen, die Einblicke in die Projektleistung geben und eine datengestützte Entscheidungsfindung unterstützen.

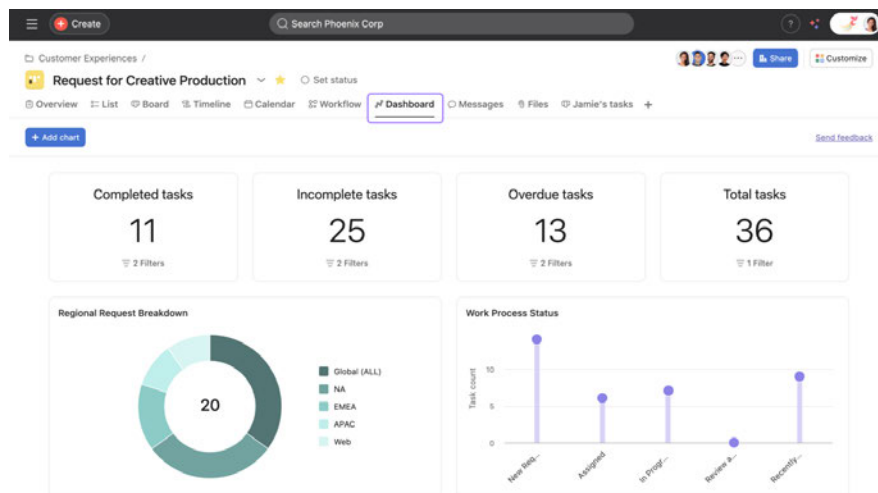


Abbildung 4.8: Asana: Dashboard *Quelle:* [Asa25b]

In Asana finden sich auf der Hauptseite des "Dashboards" eine Vielzahl von standardisierten Diagrammen und Übersichtsgrafiken, die einen schnellen Gesamteindruck über das Projekt ermöglichen. Außerdem kann dieses Dashboard entsprechend der Bedürfnisse für das jeweilige Projekt und den jeweiligen Benutzer angepasst werden. Auf diese Weise können unterschiedliche Übersichtsseiten für die jeweiligen Bereiche des Projekts entstehen.

4.5 Risiken und Herausforderungen im Hinblick auf Projektmanagementsoftware

Nach der Untersuchung, wie PMS das Projektmanagement unterstützen kann, soll der folgende Abschnitt die besonderen Risiken, die PMS mit sich bringt, beleuchten.

1. Komplexität und Lernkurve Einige PMS-Lösungen können sehr komplex sein und erfordern unter Umständen einen erheblichen Zeit- und Schulungsaufwand für die Teammitglieder, um sich mit der Software vertraut zu machen.[BAM08] Dies kann zu Widerstand oder unsachgemäßer Nutzung der Software führen.

2. Kostenüberlegungen Die für PMS erforderlichen finanziellen Investitionen, einschließlich Lizenzgebühren und Schulungskosten, können ein Hindernis darstellen, insbesondere für kleine Organisationen oder Projekte mit begrenzten Budgets.[Fon+17]

3. Übermäßige Abhängigkeit von Software Eine übermäßige Abhängigkeit von PMS kann dazu führen, dass wesentliche Projektmanagementfähigkeiten wie Führung, Kommunikation und Problemlösung vernachlässigt werden. Die Software sollte diese menschlichen Aspekte ergänzen, nicht ersetzen.[Iri01]

4. Datensicherheit und Datenschutz Die Speicherung sensibler Projektinformationen auf digitalen Plattformen, insbesondere auf Cloud-basierten Lösungen, wirft Probleme hinsichtlich der Datensicherheit und des Datenschutzes auf. Unternehmen müssen sicherstellen, dass geeignete Schutzmaßnahmen vorhanden sind. [WMM21]

5. Fragen der Integration Die Kompatibilität von einem PMS mit bestehenden Systemen und Werkzeugen, die innerhalb der Organisation verwendet werden, kann ein Problem darstellen und zu Ineffizienz führen. [BR23]

6. Widerstände gegen Veränderungen Die Einführung einer neuen PMS kann auf den Widerstand von Teammitgliedern stoßen, die an die bestehenden Prozesse gewöhnt sind. [Leg95] Um die Einführung zu erleichtern, sind wirksame Strategien für das Änderungsmanagement erforderlich.

5 Die Rolle von Projektmanagement-Software in der Indie-Spielentwicklung

Das Folgende Kapitel soll die Erkenntnisse für wichtige Faktoren im Projektmanagement in der Spieleentwicklung aus 3.3 und 3.4 mit den Erkenntnissen für die Anwendung von PMS aus 4.4 und 4.5 zusammenführen. Dazu werden Unterstützungsmöglichkeiten und wichtige Einschränkungen untersucht. Danach werden potenzielle Vorteile und negative Konsequenzen durch die Verwendung der Software beleuchtet. In 5.3 wird aus den diskutierten Faktoren eine Liste mit Aufgabenbereichen für PMS erstellt.

5.1 Wie ein PMS bei der Spieleentwicklung helfen kann

Verbesserte Zusammenarbeit und Kommunikation In der Spieleentwicklung sind Projekte von Natur aus interdisziplinär und erfordern eine nahtlose Zusammenarbeit zwischen Programmierern, Künstlern, Designern und Tontechnikern. PMS erleichtert diese Zusammenarbeit durch die Bereitstellung zentraler Plattformen, auf denen die Teammitglieder kommunizieren, Dateien austauschen und Aufgaben in Echtzeit koordinieren können. Funktionen wie gemeinsame Dashboards, Aufgabenzuweisungen und Nachrichtensysteme helfen dabei, Grenzen zu überwinden und Transparenz zu schaffen.

Verwaltung komplexer Arbeitsabläufe Die Entwicklung von Spielen umfasst komplizierte Arbeitsabläufe mit zahlreichen voneinander abhängigen Aufgaben. Ein PMS hilft dabei, diese Arbeitsabläufe zu beschreiben, Abhängigkeiten zu erkennen und Aufgaben entsprechend zu planen. Dadurch wird sichergestellt, dass kritische Prozesse identifiziert und Ressourcen effizient zugewiesen werden, um Engpässe zu vermeiden.

Unterstützung von agilen Methoden Angesichts des iterativen Charakters der Spieleentwicklung sind PMS, die agile Methoden wie Scrum und Kanban unterstützen, besonders vorteilhaft. Sie ermöglichen es den Teams, Sprints zu planen, Backlogs zu verwalten und sich schnell an Änderungen anzupassen, was dem kreativen Prozess der Spieleentwicklung entgegenkommt.

Erleichterung für entfernte und dezentrale Teams Indie-Spielentwickler arbeiten oft an unterschiedlichen Orten oder haben Teammitglieder, die über verschiedene Länder verteilt sind. PMS bietet Cloud-basierte Lösungen, die den Fernzugriff ermöglichen, so dass die Teams unabhängig von geografischen Barrieren effektiv zusammenarbeiten können.

Verbessertes Ressourcenmanagement Ein PMS hilft bei der Überwachung von Ressourcen, einschließlich Zeit, Budget und Personal, was bei der Spieleentwicklung, bei der Projekte schnell ressourcenintensiv werden können, entscheidend ist. Es hilft Indie-Teams bei der Überwachung der Aufgaben und der Einhaltung von Budgetvorgaben.

5.2 Unterschiede zu allgemeinen PMS-Anwendungen

Während ein PMS allgemeine Vorteile für das Projektmanagements bietet, unterscheidet sich die Anwendung in der Spieleentwicklung aufgrund der besonderen Anforderungen der Branche:

Kreativität und Iteration: Die Spieleentwicklung erfordert Werkzeuge, die kreative Iterationen und schnelles Prototyping unterstützen. PMS muss in diesem Zusammenhang häufige Änderungen zulassen und die Versionskontrolle für Assets unterstützen.

Asset-Verwaltung: Die Verwaltung einer großen Anzahl digitaler Assets wie Grafiken, Audiodateien und Codeschnipsel ist bei der Spieleentwicklung besonders wichtig. PMS ist häufig mit Asset-Management-Systemen integriert, um diese Anforderungen zu erfüllen.

Testen und Qualitätssicherung: PMS für die Spieleentwicklung umfasst häufig Funktionen zur Verwaltung von Testzyklen, zur Fehlerverfolgung und zu spielspezifischen Qualitätssicherungsprozessen.

5.3 Vergleich von Nutzen und Risiken für kleine Spieleentwicklungsteams

Für dieses Kapitel wurden die Faktoren für Projektmanagement in Indie-Studios aus Kapitel 3.3 mit den Faktoren für PMS in der Indie-Spieleentwicklung 4.4 und mit den Risiken für beide aus Kapitel 3.4 und 4.5 wie folgt gegenübergestellt:

	Kapitel 4.4						Kapitel 3.4				Kapitel 4.5			
	Planung & Terminierung	Zusammenarbeit	Ressourcen-Optimierung	Budget-Verwaltung	Risikominderung	Transparenz & Verantwortlichkeit	Begrenzte Ressourcen	Überschneidende Rollen	Fehlende Formale Prozesse	Aufmerksamkeit auf dem Markt	Komplexität und Lernkurve	Kostenüberlegungen	Abhängigkeit von Software	Datensicherheit und Datenschutz
Kapitel 3.3	Ressourcenmanagement	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Kommunikation & Kollaboration	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Flexibilität & Anpassungsfähigkeit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Risikomanagement	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Prioritäts- & Umfangsmanagement	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Zeitmanagement	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Qualitätssicherung	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Marketing & Communitymanageme	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Finanzielle Verwaltung	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Motivation und Teamgeist	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Abbildung 5.1: Kombination der Erkenntnisse aus mehreren Kapiteln

Die Grafik 5.1 zeigt in der linken horizontalen Spalte alle Faktoren aus Kapitel 3.4 und zeigt durch die Haken, welche Faktoren aus Kapitel 4.4 in den linken vertikalen Spalten dort zur Anwendung kommen. Der Mittlere und Rechte Block in der Tabelle zeigt, welche der Risikofaktoren aus Kapitel 3.5 und 4.5 jeweils eine Rolle in dem Bereich spielen.

1. Ressourcenmanagement

Vorteil: Ein PMS bietet Werkzeuge für die Budgetierung und Ressourcenzuweisung und hilft kleinen Teams, ihre begrenzten Ressourcen zu optimieren.

Risiko: Die Komplexität von PMS kann kleine Teams überfordern und zu einer ineffizienten Nutzung führen, wenn es nicht richtig verwaltet wird.

2. Kommunikation und Kollaboration

Vorteil: Verbesserte Kommunikation durch integrierte Messaging- und Kollaborationsfunktionen, wichtig für Teams, deren Mitglieder mehrere Aufgaben wahrnehmen.

Risiko: Eine übermäßige Abhängigkeit von PMS für die Kommunikation kann die persönliche Interaktion reduzieren, was den Zusammenhalt des Teams beeinträchtigen kann.

3. Flexibilität und Anpassungsfähigkeit

Vorteil: Ein PMS, welches agile Projektmanagementmethoden unterstützt, ermöglicht es kleinen Teams, schnell auf Veränderungen zu reagieren, was den Anforderungen an die nötige Flexibilität gerecht wird.

Risiko: Die Starrheit einiger PMS-Tools kann die Anpassungsfähigkeit behindern, wenn sie nicht an die Arbeitsabläufe des Teams angepasst werden können.

4. Risikomanagement

Vorteil: Ein PMS umfasst Risikomanagementmodule, die kleinen Teams helfen, Risiken proaktiv zu erkennen und zu mindern.

Risiko: Ein unzureichendes Verständnis dieser Instrumente kann zu übersehenen Risiken oder falscher Sicherheit führen.

5. Prioritätensetzung und Umfangsmanagement

Vorteil: Das PMS hilft bei der Priorisierung von Aufgaben und der Verwaltung des Umfangs und verhindert eine Überdehnung.

Risiko: Der Missbrauch oder die Fehlinterpretation von PMS-Daten kann zu schlechten Entscheidungen bei der Priorisierung führen.

6. Zeitmanagement

Vorteil: Zeitplanungsfunktionen helfen Teams, Termine und Meilensteine einzuhalten.

Risiko: Eine ungenaue Zeiterfassung durch unsachgemäße Verwendung kann ein falsches Gefühl für den Fortschritt vermitteln.

7. Qualitätssicherung

Vorteil: Ein PMS integriert die Fehlerverfolgung und Testpläne, die für die Aufrechterhaltung der Qualität unerlässlich sind.

Risiko: Die Abhängigkeit von PMS ohne angemessene Testverfahren kann dazu führen, dass Qualitätsprobleme übersehen werden .

8. Marketing und Community-Engagement

Vorteil: Einige PMS-Tools bieten eine Integration mit Marketingplattformen und unterstützen so das Community-Engagement.

Risiko: Begrenzte Funktionen in PMS unterstützen die Marketinganforderungen von Indie-Teams möglicherweise nicht vollständig und erfordern zusätzliche Tools.

9. Finanzverwaltung

Vorteil: Funktionen zur Budgetverfolgung helfen bei der Überwachung der Ausgaben und der Prognose des Finanzbedarfs.

Risiko: Ungenaue Finanzdaten aufgrund von Benutzerfehlern können zu Problemen bei der Budgetierung führen.

10. Motivation und Team-Moral

Vorteil: Transparenz bei der Aufgabenzuteilung und dem Fortschritt kann die Verantwortlichkeit und Motivation steigern.

Risiko: Die Überbetonung von Kennzahlen kann Druck erzeugen und sich negativ auf die Moral auswirken.

5.4 Sind PMS für kleine Indie-Teams nützlich?

Der Nutzen von Projektmanagement-Software für kleine Indie-Teams hängt von verschiedenen Faktoren ab, darunter Teamgröße, Projektkomplexität und verfügbare Ressourcen.

5.4.1 Vorteile für Indie-Teams

Zentralisierung von Informationen PMS bietet eine zentrale Quelle für alle projektbezogenen Daten, was für kleine Teams, in denen die Mitglieder mit mehreren Rollen konfrontiert sind, sehr wertvoll ist.

Verbesserte Organisation Hilft bei der Organisation von Aufgaben, Terminen und Verantwortlichkeiten und verringert die Wahrscheinlichkeit, dass Aufgaben in Vergessenheit geraten.

Erleichtert die Arbeit aus der Ferne Da viele Indie-Entwickler aus der Ferne arbeiten, ermöglicht ein PMS eine effektive Zusammenarbeit und Kommunikation.

5.4.2 Herausforderungen für Indie-Teams

Kostenbeschränkungen Ein PMS kann teuer sein, und die Abonnementgebühren können die begrenzten Budgets von Indie-Teams überfordern. Es gibt jedoch kostenlose oder kostengünstige Optionen, die den Anforderungen entsprechen können.

Komplexität und Lernkurve Der Zeit- und Arbeitsaufwand, der für das Erlernen und Implementieren von PMS erforderlich ist, kann die Aufmerksamkeit von Entwicklungsaufgaben einschränken.

Overhead Die Implementierung von PMS führt zu administrativem Overhead, der für sehr kleine Teams sehr belastend sein kann.

6 Bewertungsbogen für PMS in der Indie-Spieleentwicklung

Der folgende Bewertungsbogen fasst die in den vorherigen Kapiteln gewonnenen Erkenntnisse in Anforderungen an PMS für die Indie Spieleentwicklung zusammen und nimmt eine geschätzte Vorpriorisierung der einzelnen Anforderungen vor. Die einzelnen Punkte des Bewertungsbogens stützten sich auf die Erkenntnisse von Kapitel 5, sowie die Erfahrung und Features durch das Testen bekannter PMS, mögliche Risiken und deren Umgehung aus den Kapiteln 3.4 und 4.5 sowie logische Rückschlüsse aus der Kombination dieser Anforderungen. Die Art, wie diese mehrdimensionalen Faktoren zusammenhängen, erschwert eine grafische Darstellung, weswegen die einzelnen Anforderungen kurz erklärt werden. Ziel dieses Kapitels ist es, einen vorläufigen Bewertungsbogen für PMS in der Indie-Spieleentwicklung zu erstellen, der möglichst vollständig ist und diesen danach zu überprüfen und anzupassen.

6.1 Grundlegende Funktionalitäten

1.1 Unterstützung für agile Methoden

Priorität: Hohe Priorität

Beschreibung: Das PMS muss agile Projektmanagement-Praktiken wie Scrum und Kanban unterstützen, um der iterativen Entwicklung und den häufigen Änderungen, die mit der Entwicklung von Spielen einhergehen, gerecht zu werden.

Erläuterung: Agile Methoden ermöglichen es den Teams, sich schnell an Änderungen anzupassen, was für die kreative und dynamische Natur von Indie-Game-Projekten entscheidend ist.

1.2 Aufgabenmanagement und Zeitplanung

Priorität: Hohe Priorität

Beschreibung: Die Software sollte zuverlässige Werkzeuge für die Erstellung, Zuweisung, Priorisierung und Planung von Aufgaben, einschließlich Abhängigkeiten und Meilensteinen, bieten.

Erläuterung: Eine effektive Aufgabenverwaltung stellt sicher, dass sich die Teammitglieder über ihre Zuständigkeiten und Fristen im Klaren sind, was für das Zeitmanagement und das Erreichen der Projektziele unerlässlich ist.

1.3 Kommunikations- und Kollaborationswerkzeuge

Priorität: Hohe Priorität

Beschreibung: Integrierte Kommunikationsfunktionen wie Messaging, Kommentare und Echtzeit-Updates sind notwendig, um die Zusammenarbeit zwischen den Teammitgliedern zu erleichtern.

Erläuterung: Eine nahtlose Kommunikation hilft bei der Koordinierung der Bemühungen, insbesondere in Teams, in denen die Mitglieder möglicherweise an mehreren Aufgaben oder aus der Ferne arbeiten.

1.4 Gemeinsame Nutzung von Dateien und Dokumentenverwaltung

Priorität: Hohe Priorität

Beschreibung: Das PMS sollte eine einfache gemeinsame Nutzung, Versionierung und Verwaltung

von Dokumenten und Assets ermöglichen und verschiedene in der Spieleentwicklung übliche Dateitypen unterstützen (z.B. Grafiken, Audiodateien).

Erläuterung: Ein effizienter Umgang mit digitalen Assets verhindert Fehlkommunikation und Arbeitsverluste, was angesichts der multidisziplinären Natur der Spieleentwicklung entscheidend ist.

1.5 Integration mit Entwicklungswerkzeugen

Priorität: Hohe Priorität

Beschreibung: Die Software sollte sich in gängige Entwicklungstools und -plattformen integrieren lassen, z. B. Git für die Versionskontrolle, Unity oder Unreal Engine für die Spieleentwicklung und andere Produktivitätstools.

Erläuterung: Die Integration strafft die Arbeitsabläufe und verringert die Notwendigkeit, zwischen mehreren Anwendungen zu wechseln, was die Effizienz steigert.

6.2 Benutzerfreundlichkeit und Benutzeroberfläche

2.1 Benutzerfreundliches Interface

Priorität: Hohe Priorität

Beschreibung: Das PMS muss über eine intuitive und leicht zu navigierende Oberfläche verfügen, die die Lernkurve der Teammitglieder möglichst einfach gestaltet.

Erläuterung: Eine benutzerfreundliche Oberfläche stellt sicher, dass alle Teammitglieder, unabhängig von ihren technischen Kenntnissen, die Software ohne umfangreiche Schulungen effektiv nutzen können.

2.2 Anpassungsfähigkeit

Priorität: Mittlere Priorität

Beschreibung: Die Möglichkeit, Dashboards, Workflows und Projektansichten an die spezifischen Bedürfnisse und Präferenzen des Teams anzupassen.

Erläuterung: Die Anpassungsfähigkeit ermöglicht es den Teams, das PMS an ihre individuellen Prozesse anzupassen, was die Flexibilität und Anpassungsfähigkeit erhöht.

2.3 Mobile Zugänglichkeit

Priorität: Niedrige Priorität

Beschreibung: Zugriff auf das PMS über mobile Geräte durch responsives Design oder spezielle Apps.

Erläuterung: Der mobile Zugang bietet Komfort für Teammitglieder, die Aktualisierungen überprüfen oder kommunizieren müssen, während sie sich nicht an ihrem regulären Arbeitsplatz befinden.

6.3 Technische Anforderungen

3.1 Cloud-basierte Anwendung

Priorität: Hohe Priorität

Beschreibung: Das PMS sollte online zugänglich sein und die standortübergreifende Arbeit und Zusammenarbeit unterstützen.

Erläuterung: Indie-Teams haben oft verteilte Mitglieder; ein Cloud-basiertes PMS stellt sicher, dass jeder in Echtzeit auf Projektinformationen zugreifen kann.

3.2 Skalierbarkeit

Priorität: Mittlere Priorität

Beschreibung: Die Software sollte das Wachstum des Teams und die zunehmende Komplexität der Projekte ohne nennenswerte Leistungsprobleme bewältigen können.

Erläuterung: Die Skalierbarkeit stellt sicher, dass das PMS auch bei einer Ausweitung des Projektumfangs oder einer Vergrößerung des Teams leistungsfähig bleibt.

3.3 Datensicherheit und Datenschutz

Priorität: Hohe Priorität

Beschreibung: Robuste Sicherheitsmaßnahmen zum Schutz sensibler Projektdaten, einschließlich Verschlüsselung, regelmäßiger Backups und Einhaltung der Datenschutzbestimmungen.

Erläuterung: Der Schutz von geistigem Eigentum und vertraulichen Informationen ist für Indie-Entwickler, die sich auf ihre einzigartigen Kreationen verlassen, von entscheidender Bedeutung.

3.4 Niedrige Systemanforderungen

Priorität: Mittlere Priorität

Beschreibung: Das PMS sollte keine High-End-Hardware erfordern oder übermäßig viele Systemressourcen verbrauchen.

Erläuterung: Indie-Teams können eine Vielzahl von Hardware-Konfigurationen verwenden; niedrige Systemanforderungen stellen sicher, dass alle Teammitglieder die Software effektiv nutzen können.

6.4 Kosten und Lizenzierung

4.1 Günstiges Tarifsysteem

Priorität: Hohe Priorität

Beschreibung: Das PMS sollte Preismodelle anbieten, die für kleine Teams geeignet sind, einschließlich kostenloser Stufen oder preiswerter Abonnements, ohne dass wesentliche Funktionen beeinträchtigt werden.

Erläuterung: Budgetbeschränkungen sind ein wichtiges Anliegen für kleine Teams; eine kostengünstige Software ermöglicht es ihnen, Ressourcen für andere wichtige Bereiche zu verwenden.

4.2 Flexible Lizenzierung

Priorität: Mittlere Priorität

Beschreibung: Optionen für monatliche oder jährliche Abonnements und die Möglichkeit, die Anzahl der Benutzer leicht anzupassen.

Erläuterung: Die flexible Lizenzierung passt sich an Änderungen der Teamgröße und der Projektdauer an, so dass sich die Kosten an der tatsächlichen Nutzung orientieren.

6.5 Zusätzliche Funktionalitäten

5.1 Zeiterfassung

Priorität: Mittlere Priorität

Beschreibung: Funktionen, die es den Teammitgliedern ermöglichen, die für Aufgaben aufgewendeten Stunden zu protokollieren, was das Zeitmanagement und die Produktivitätsanalyse erleichtert.

Erläuterung: Die Zeiterfassung hilft bei der Überwachung des Fortschritts und kann Informationen für künftige Projektschätzungen liefern, obwohl sie einen zusätzlichen Verwaltungsaufwand verursachen kann.

5.2 Budgetplanung und finanzielle Kontrolle

Priorität: Mittlere Priorität

Beschreibung: Werkzeuge zur Verfolgung von Ausgaben, Budgets und Finanzprognosen innerhalb des Projekts.

Erläuterung: Das Finanzmanagement ist von entscheidender Bedeutung, um sicherzustellen, dass die Projekte realisierbar bleiben und der Kostenrahmen nicht überschritten wird.

5.3 Risikomanagement-Tools

Priorität: Niedrige Priorität

Beschreibung: Module zur Identifizierung, Bewertung und Abschwächung von Projektrisiken.

Erläuterung: Das Risikomanagement ist zwar wichtig, kann aber oft mit einfacheren Mitteln in kleinen Teams bewältigt werden, sodass dies eine niedrigere Priorität hat.

5.4 Protokollierung und Analyse

Priorität: Mittlere Priorität

Beschreibung: Erstellung von Berichten und Analysen zur Projektleistung, Teamproduktivität und anderen wichtigen Kennzahlen.

Erläuterung: Datengestützte Erkenntnisse können bei der Entscheidungsfindung helfen, sind aber für sehr kleine Teams möglicherweise weniger wichtig.

5.5 Integration mit Marketingplattformen

Priorität: Niedrige Priorität

Beschreibung: Fähigkeit zur Verbindung mit sozialen Medien, E-Mail-Marketing oder anderen Plattformen zur Unterstützung von Marketingmaßnahmen.

Erläuterung: Nützlich für Teams, die sich auf das Community-Engagement konzentrieren, aber innerhalb des PMS selbst nicht unbedingt erforderlich.

6.6 Unterstützung und Schulung

6.1 Kundensupport

Priorität: Hohe Priorität

Beschreibung: Verfügbarkeit eines leicht zugänglichen Kundendienstes zur Unterstützung bei technischen Problemen oder Fragen.

Erläuterung: Ein reibungsloser Support minimiert die Ausfallzeiten und hilft den Teams, Probleme schnell zu lösen.

6.2 Dokumentation und Tutorials

Priorität: Mittlere Priorität

Beschreibung: Umfassende Anleitungen, Tutorials und Wissensdatenbanken, die den Benutzern helfen, die Software zu erlernen und effektiv zu nutzen.

Erläuterung: Eine gute Dokumentation unterstützt die Einführung des PMS und hilft den Teammitgliedern, kompetente Benutzer zu werden.

6.7 Kulturelle und teambezogene Überlegungen

7.1 Sprachliche Unterstützung

Priorität: Niedrige Priorität

Beschreibung: Verfügbarkeit des PMS in mehreren Sprachen, wenn das Team mehrsprachig ist.

Erläuterung: Verbessert die Benutzerfreundlichkeit für Teams, deren Mitglieder verschiedene Sprachen bevorzugen, ist jedoch nicht für alle Teams von Bedeutung.

7.2 Anpassung an den Team-Workflow

Priorität: Hohe Priorität

Beschreibung: Das PMS sollte mit den bestehenden Arbeitsabläufen und Prozessen des Teams übereinstimmen oder an diese angepasst werden können.

Erläuterung: Eine Software, die sich auf natürliche Weise in die Arbeitsabläufe des Teams einfügt, verringert Reibungsverluste und erhöht die Effizienz.

6.8 Schlichtheit und Minimalismus

8.1 Vermeidung einer Überfrachtung mit Funktionen

Priorität: Hohe Priorität

Beschreibung: Das PMS sollte sich auf wesentliche Funktionen konzentrieren, ohne unnötige Komplexität, die die Benutzer überfordern könnte.

Erläuterung: Die Einfachheit stellt sicher, dass sich die Teammitglieder auf die Entwicklungsaufgaben konzentrieren können, anstatt die Software selbst zu verwalten.

6.9 Motivation und Teammoral

9.1 Transparenz und Nachverfolgung

Priorität: Mittlere Priorität

Beschreibung: Die Software sollte den Projektfortschritt sichtbar machen und so das Gefühl von Leistung und Verantwortlichkeit fördern.

Erläuterung: Transparenz kann die Arbeitsmoral steigern, indem sie zeigt, wie die individuellen Bemühungen zum Gesamtprojekt beitragen.

9.2 Anerkennungs- und Feedback-Mechanismen

Priorität: Niedrige Priorität

Beschreibung: Funktionen, die es ermöglichen, Leistungen von Teammitgliedern anzuerkennen oder Feedback zu Aufgaben zu geben.

Erläuterung: Diese Funktionen sind zwar für die Moral des Teams von Vorteil, werden aber eher als Extras denn als unverzichtbar angesehen.

7 Überarbeitung des Bewertungsbogens durch Experteninterviews

Der im letzten Kapitel erstellte Bewertungsbogen, der die Anforderungen und Regeln für eine optimale Projektmanagement-Software für die Entwicklung von Indie-Spielen umschreibt, dient als grundlegender Rahmen, der aus Literaturrecherchen, vorherigen Analysen und eigener Erfahrung durch Testung bestehender PMS-Systeme abgeleitet wurde. Um jedoch die praktische Anwendbarkeit und Relevanz für aktuelle Branchenpraktiken sicherzustellen, ist es unerlässlich, diesen Bewertungsbogen durch empirische Methoden zu überarbeiten und zu validieren. Experteninterviews sind eine qualitative Forschungsmethode, die tiefgehende Einblicke von Experten auf dem Gebiet geben können und wertvolle Perspektiven zur Verfeinerung des Bewertungsbogens liefern.

7.1 Notwendigkeit zur Überarbeitung des Bewertungsbogens

Während der Bewertungsbogen kritische Anforderungen basierend auf theoretischem Verständnis und allgemeinem Branchenwissen erfasst, erfordert die dynamische Natur der Indie-Spielentwicklung eine kontinuierliche Validierung. Die Herausforderungen und Bedürfnisse von Indie-Entwicklern können erheblich variieren, abhängig von Faktoren wie Teamgröße, Projektumfang und verfügbaren Ressourcen. Darüber hinaus bedeuten Fortschritte in der PMS-Technologie und sich entwickelnde Projektmanagementmethoden, dass sich die Anforderungen im Laufe der Zeit ändern können. Durch die Überarbeitung des Bewertungsformulars wird die:

Relevanz sichergestellt: Die Anforderungen mit dem aktuellen Vorgehen und Herausforderungen von Indie-Entwicklern in Einklang bringen.

Praktikabilität verbessert: Praktische Erkenntnisse einbeziehen, die aus der Literatur allein möglicherweise nicht ersichtlich sind.

Lücken identifiziert: Fehlende Anforderungen oder über- oder unter priorisierte Aspekte erkennen, die Anpassungen benötigen.

7.2 Nutzung von Experteninterviews zur Verifizierung

Experteninterviews sind eine qualitative Forschungsmethode, die strukturierte oder halbstrukturierte Gespräche mit Personen umfasst, die über spezialisiertes Wissen oder Erfahrung in einem bestimmten Bereich verfügen.[TH08] Im Kontext dieser Studie bietet das Interviewen von Indie-Spielentwicklern und PMS-Erstellern direkten Zugang zu Expertenmeinungen und persönlichen Erfahrungen. Ein Experteninterview zur Überprüfung des Anforderungsbogens bietet die folgenden Vorteile:

Tiefgehende Einblicke: Experteninterviews ermöglichen es, komplexe Themen im Detail zu erkunden und nuancierte Verständnisse der Anforderungen und Herausforderungen im Zusammenhang mit PMS in der Indie-Spielentwicklung aufzudecken.

Validierung der Ergebnisse: Interviews können das ursprüngliche Bewertungsformular validieren, indem sie bestätigen, welche Anforderungen wirklich wesentlich sind und welche möglicherweise weniger relevant sind.

Entdeckung neuer Informationen: Experten können neue Überlegungen oder aufkommende Trends einführen, die in der Literaturübersicht nicht erfasst wurden.

Kontextuelles Verständnis: Das Verständnis des Kontexts, in dem Indie-Entwickler tätig sind, hilft, das Bewertungsformular an reale Szenarien anzupassen.

7.3 Methodik zur Durchführung von Experteninterviews

Um den Bewertungsbogen effektiv zu überarbeiten, müssen die Experteninterviews sorgfältig vorbereitet und durchgeführt werden. Die Methodik umfasst mehrere Hauptbestandteile:

Zieldefinition Das Ziel des Interviews sollte klar definiert sein.

Der Zweck des Interviews besteht darin, die PMS-Anforderungen für die Entwicklung von Indie-Spielen zu bewerten und zu verfeinern. Die zu behandelnden Themen sind:

- Validierung bestehender Anforderungen.
- Identifikation fehlender Anforderungen.
- Priorisierung der Anforderungen basierend auf ihren Erfahrungen.
- Herausforderungen bei der Nutzung von PMS.

Auswahl der Teilnehmer Die Experten sollten langfristige Erfahrung auf dem zu untersuchenden Gebiet haben.

Dazu wurden Experten ausgewählt, die Indie-Spielentwickler mit oder ohne Erfahrung im Projektmanagement oder Schöpfer von PMS-Tools sind, die speziell für die Spieleentwicklung entwickelt wurden. Da PMS von allen am Projekt beteiligten Personen verwendet wird, ist es wichtig neben der Meinung von Projektmanagern, auch Erfahrungen von zum Beispiel Stakeholdern oder Entwicklern einzuholen, da diese die PMS ebenfalls verwenden müssen, auch ohne das Verständnis von Projektmanagement.

Stichprobengröße Es sollte eine vielfältige Gruppe von Experten angestrebt werden, um eine Vielzahl von Perspektiven zu erfassen, typischerweise zwischen 5 und 15 Teilnehmern für qualitative Studien.

A sample size of 5 to 15 experts is often sufficient in qualitative research to reach saturation and obtain meaningful insights. [Mas10, S.25]

Gestaltung des Interviewleitfadens Als Format für das Interview wurde eine semistrukturierte Interviewform gewählt. Bei einem semistrukturierten Interviewleitfaden werden offenen Fragen gestellt, die detaillierte Antworten fördern und gleichzeitig Flexibilität bieten, um aufkommende Themen zu erforschen.[HW18] Dieses Interviewformat wurde gewählt, um sicherzustellen, dass durch Erfragen auch potenziell bisher nicht erfasste Anforderungen identifiziert werden können, indem Nachfragen außerhalb des Interviewleitfadens möglich sind.

Durchführung der Interviews

Ethische Überlegungen: Es sollte die schriftliche Zustimmung der Teilnehmer für die erhobenen Daten und deren Verarbeitung eingeholt werden. Vertraulichkeit und Datenschutz müssen gewährleistet sein.

Aufzeichnung: Mit Erlaubnis des Interviewpartners können die Interviews aufgenommen werden, um eine genaue Transkription und Analyse zu ermöglichen.

Datenanalyse Aufgrund der unterschiedlichen Blickwinkel auf Projektmanagement der Experten bietet sich eine interpretative Auswertung der Ergebnisse an, dazu werden sowohl die allgemeinen Erkenntnisse für Projektmanagement in der Indie Spieleentwicklung, als auch der Anforderungsbogen betrachtet.

Auswertung Aufgrund der unterschiedlichen Blickwinkel durch die Vorkenntnisse der Experten und dem semistrukturierten Format des Interviews können nur bedingt spezifische Aussagen der Experten miteinander verglichen werden. Aus diesem Grund bietet sich eine interpretative Auswertung des Interviews an, bei der die Aussagen der Experten jeweils im Kontext der Fragestellung untersucht werden.

7.4 Auswahl der Experten

Wie im vorherigen Kapitel beschrieben, sollte die Stichprobengröße der Experten 5 bis 15 Personen betragen. Um sicherzustellen, dass unterschiedliche Ansichten auf PMS berücksichtigt werden, ist es wichtig, dass die Experten aus unterschiedlichen Aufgabenbereichen kommen, um einen möglichst diversen Blick auf das Thema zu werfen. Aus diesem Grund wurden Experten im Bereich Entwicklung, Projektmanagement und Projektgründer/Stakeholder ausgewählt. Folgende Experten wurden interviewt:

Tessa Abraham i.F. Interviewpartner: A

Aktuelle Tätigkeit: Producer/Projektmanager

Erfahrung: Seit 6 Jahren in der professionellen Spieleentwicklung bei KingArt

Benutzte PMS: Internes Tool, Jira, Trello, Asana, Hansoft, Excel, MS Project, GDrive, Sharepoint, Confluence

René Habermann i.F. Interviewpartner: B

Aktuelle Tätigkeit: Product Owner, Projektmanager, Director, Gamedesigner

Erfahrung: Mehrere Projekte mit 1-15 Mitarbeitern

Benutze PMS: Jira, Excel

Kai Mergener i.F. Interviewpartner: C

Aktuelle Tätigkeit: Senior Material Artist bei Gunzilla Games

Erfahrung: Über 6 Jahre in Indie Studio mit am Ende 20 Mitarbeitern

Benutzte PMS: Excel, GDrive, Discord, Trello, Asana (mit eigenen Tools), Wiki-Software

Riad Djemili i.F. Interviewpartner: D

Aktuelle Tätigkeit: Codecks CEO, Maschinen Mensch Gründer

Erfahrung: Seit 2005 Spieleentwickler, Projektplanung in AAA-Bereich, seit 10 Jahren mit Maschinen Mensch (von 2 zu 23 Personen)

Benutzte PMS: Codecks, Jira, Trello, Hansoft, Mantis

Julien Schillinger i.F. Interviewpartner: E

Aktuelle Tätigkeit: Geschäftsführer von Pandabee, Produkt-/Projektmanagement

Erfahrung: Vor 3 Jahren Gründung von Pandabee, Projekte mit 2-7 Entwicklern

Benutze PMS: Jira, Discord, Trello, Asana, Codecks, GDrive

7.5 Durchführung des Interviews

Aus dem Bewertungsbogen aus Kapitel 6 und den Kriterien aus Kapitel 7.3 wurde der Fragebogen in A für das semistrukturierte Experteninterview erstellt. Alle Interviews wurden über online Kommunikationstools (Discord, Google Meet) durchgeführt und Ton und Video des Interviews aufgezeichnet. Zu Beginn des Interviews wurde zunächst kurz das Thema der Arbeit erläutert und danach um Zustimmung zur Aufzeichnung und Datenverarbeitung gefragt. Zu Beginn der Befragung wurden einige Fragen zur persönlichen Erfahrung der Experten und welche PMS Lösungen bereits verwendet wurden, gestellt. Danach folgen 26 Kriterien und Features für PMS, eingeteilt in 9 Kategorien. Die Kriterien decken sich mit denen aus Kapitel 6. Zu jedem der Kriterien wurde dann nach der Relevanz, Priorität und Messbarkeit gefragt.

Relevanz: Erfragt, ob dieses Kriterium in der Indiespieleentwicklung genutzt wird, ob Anwendungsfälle gesehen werden und dieses Kriterium dem Entwicklungsprozess helfen würde

Priorität: Erfragt, wie wichtig ein Feature dem Befragten in einer PMS wäre. Dabei soll nur eine grobe Priorisierung zwischen gar nicht, niedrig, mittel und hoch vorgenommen werden

Messbarkeit: Erfragt, wann der Interviewpartner das jeweilige Kriterium als erfüllt ansehen würde oder welche Voraussetzungen erfüllt sein müssen, damit das Feature als "gut" angesehen wird.

Bei der Beantwortung der Fragen wurden jeweils Nachfragen gestellt, Beschreibungen gegeben und ein Dialog geführt, um Unklarheiten zu beseitigen. Die Befragten hatten zudem jederzeit die Möglichkeit eine Frage oder Einschätzung zu überspringen, falls das Kriterium außerhalb ihres persönlichen Erfahrungsbereiches liegt, da aufgrund der unterschiedlichen Blickwinkel nicht alle Experten mit allen Bereichen einer PMS zu tun haben.

Nach Abschluss der Kriterien und Feature Fragen wurden die Experten nach Korrekturen, die sie an ihren bisherigen Aussagen vornehmen wollen, gefragt, sowie ob ihnen noch Features oder Kriterien bekannt sind, die in der Auflistung gefehlt haben. Die letzten zwei Fragen beziehen sich auf das wichtigste aller Features oder Kriterien für PMS und wie die Experten beurteilen, ob sie eine PMS, die sie benutzen, weiternutzen würden oder wann sie über einen Wechsel nachdenken. Die abschließenden Fragen dienen der Kontrolle der bisherigen Antworten und zur Fokussierung des

jeweiligen Experten.

Nach der Durchführung der Interviews wurden diese transkribiert und die jeweiligen Kernaussagen extrahiert, mit Zitaten versehen und in dem Dokument unter [B](#) zusammengefasst.

8 Ergebnisse des Interviews

8.1 Allgemeine Ergebnisse

Neben der Bewertung der einzelnen Kriterien konnten durch die Interviews auch allgemeine Erkenntnisse über Projektmanagement und PMS in der Indiespieleentwicklung gewonnen werden, die Rückschlüsse auf den Aufbau einer PMS in dem Bereich zulassen.

8.1.1 Skalierende Prioritäten

Für die Prioritäten der einzelnen Kriterien wurde vor dem Interview eine Skala von niedrig bis hoch festgelegt. Während der Interviews ist jedoch schnell klar geworden, dass diese beschränkte Skala nicht ausreichend ist. Viele der Prioritäten, wie zum Beispiel bei der Anpassungsfähigkeit der Oberfläche oder der Terminplanung sind für kleine Teams eher unwichtig, ab einer gewissen Größe aber nahezu unverzichtbar. Interviewpartner D sagte zu Aufgabenmanagement und Zeitplanung, dass:

Sobald man eine gewisse Größe hat und ein größeres Budget hat [...] müssen termingerecht komplexe Dinge fertig werden [...] ab dann hat man auch zeitliche Verpflichtungen.

Somit gibt es Anforderungen, deren Priorität mit steigender Teamgröße, Projekt Budgetierung und Projektkomplexität steigen. Da diese Änderung bereits im ersten Interview schnell deutlich wurde, wurden für das gesamte Interview skalare Angaben für die Priorisierung von allen Experten zugelassen.

8.1.2 Unterschiedliche Zielgruppen

Ableitend von den skalierenden Prioritäten stellt sich natürlich die Frage, ab wann ein Kriterium für ein Team von zum Beispiel wenig zu mittlerer Priorität voranschreitet. Aufgrund der Antworten auf Nachfragen bei den Experten wurden drei Zielgruppen identifiziert, die es in der Indiespieleentwicklung gibt. Im Folgenden werden diese Zielgruppen genauer definiert. Die Wechsel zwischen diesen Gruppen sind jedoch flüssig und nicht als harte Grenzen anzusehen.

Freunde Teams

Teamgröße: bis 10 Personen oder bis ein dedizierter Projektmanager benötigt wird

Charakteristika: Diese kleinen Gruppen bestehen meistens aus wenigen, vielleicht sogar nur 1-2 Personen. In diesem Team kennen sich alle beteiligten gut, sind vermutlich befreundet und wenn sie nicht sogar in einem Büro zusammenarbeiten, dann findet zumindest ein regelmäßiger Austausch statt. Projekte sind meistens eher klein, manchmal auch nur Hobbyprojekte, weswegen kaum finanzielle Mittel vorhanden sind.

Besondere Anforderungen: Für diese kleinen Teams ist die finanzielle Seite der PMS das entscheidende. Freundes Teams würden auch schlechtere Alternativen benutzen, wenn diese dafür kostenlos sind. Die Tatsache, dass jedoch eine direkte Kommunikation zwischen allen Entwicklern

möglich ist, reduziert auch stark die Anforderungen an eine PMS. So stellte Interviewpartner E bei Kommunikations- und Kollaborationswerkzeugen fest, dass diese, sofern das Team klein genug für direkte Kommunikation ist, nicht wirklich benötigt werden.

Semiprofessionelle Teams

Teamgröße: 10 bis 20

Charakteristika: Semiprofessionelle Teams benutzen standardisierte Verfahren und haben dedizierte Projektmanager. Diese Teams haben immer noch den Vorteil, dass sich alle Personen untereinander kennen, was eine gewisse Freiheit im Vorgehen ermöglicht. Jedoch haben diese Teams durch den höheren Budgetrahmen auch stärkere Verpflichtungen gegenüber Publishern und Stakeholdern.

Besondere Anforderungen: Teams dieser Kategorie verfügen über die nötigen Mittel, um sich auch eine bezahlte PMS Lösung zu kaufen, können dafür jedoch keine riesigen Beträge aufbringen. Das Vorhandensein eines Projektmanagers ändert jedoch auch die Anforderungen an das PMS. Ein dedizierter Projektmanager bedeutet, dass nun Dinge wie Auswertungen und Zeitplanung einen höheren Stellenwert haben, da diese ausgewertet und überprüft werden. Durch die stärkeren Verpflichtungen zum Nachweis von verwendeter Arbeitszeit oder die Einbindung von Freelancern muss das Projekt Strukturen folgen, um übersichtlich zu bleiben und kurze Einarbeitungszeiten zu ermöglichen. Experte C hat diese Veränderung durch die Erweiterung des original Teams von einem "Freundes Team" in ein "Semiprofessionelles Team" bereits erlebt:

Wir haben extra einen Praktikanten gehabt für Projektmanagement in dem kleinen Team wäre es auch bis zum Ende ohne PMS möglich gewesen, [...] als wir mit 10 Leuten ins 2. Projekt gegangen sind [...] war es alleine aufgrund der Komplexität nötig mit Tickets und Tasks zu arbeiten [...] denn nur so war es möglich neue Leute einzuarbeiten [...] und da haben wir festgestellt, dass es auch gar nicht ohne geht.

Vollwertiges Studio

Teamgröße: 25+

Charakteristika: Ab einer gewissen Größe ist es nicht mehr möglich, dass alle am Projekt beteiligten Personen jederzeit den vollen Überblick über das Projekt haben, geschweige denn alle am Projekt beteiligten Personen persönlich kennen. Ab diesem Zeitpunkt werden in nahezu allen Bereichen des Projektmanagements Industriestandards benutzt und die einzelnen Entwickler und Projektmanager spezialisieren sich auf ihre jeweilige Richtung.

Besondere Anforderungen: Ab diesem Punkt gibt es nur noch wenige Unterschiede zwischen den Anforderungen einer Indie Produktion und AA oder AAA Produktionen im Bereich des Projektmanagements. Mit steigender Größe werden die einzelnen Abläufe komplexer und eine PMS muss hier vor allem bei der Skalierbarkeit anpassungsfähig sein. Jedoch soll diese Gruppe aufgrund ihrer Größe und veränderten Anforderungen in dieser Arbeit nicht weiter betrachtet werden.

8.1.3 Unterschiedliche Blickwinkel

Bei den Interviews ist vor allem auch der Unterschied zwischen den verschiedenen Rollen in einem Studio im Hinblick auf PMS aufgefallen. Als Beispiel hat Experte A die PMS eher aus der Sicht eines Projektmanagers geschildert, während Experte C Erfahrungen als Entwickler gesammelt hat. Auch

als Projekt Owner wie bei Experte E verändern sich die Anforderungen und Prioritäten für PMS. Für Entwickler ist vor allem die Einfachheit der Bedienung entscheidend und dass ihnen wenig Overhead entsteht. Diese Anforderung ist vor allem in der Zielgruppe der Freunde Teams entscheidend, da in dieser meist alle beteiligten Entwickler sind. Diese haben nicht die Zeit große Planungen zu machen oder Statistiken auszuwerten, was durch die schnelleren Kommunikationswege aber auch eine geringere Relevanz hat. Ab dem Moment in dem sich das Team in mehr spezialisierte Rollen aufteilt, es also auch einen Projektmanager gibt und das Team in die Zielgruppe der semiprofessionellen Teams wechselt, kommen die Anforderungen dieser speziellen Rollen noch zum Projektmanagement dazu. Die Anforderungen von den Entwicklern bleiben jedoch auch bei einem größeren Team gleich. Auch in semiprofessionellen Teams oder vollwertigen Studios möchten die Entwickler die PMS möglichst einfach bedienen, den Entwicklern sind Projektmanagement Grundsätze und Abläufe weniger wichtig, denn für die Planung und Implementierung von denen ist der Projektmanager zuständig. Ein Entwickler möchte möglichst wenig Zeit in der PMS und möglichst viel Zeit an den Aufgaben verbringen. Die PMS muss es schaffen, diese Anforderung von Entwicklern und die zusätzlichen Anforderungen des Projektmanagers in semiprofessionellen Teams und vollwertigen Studios in Einklang zu bringen. Ein Projektmanager hat die Zeit sich auch in komplexere Vorgänge einzuarbeiten, sofern er einen Nutzen aus den gewonnenen Daten ziehen kann, ein Entwickler hat diese Zeit nicht. Natürlich sind die Anforderungen des Teams an die PMS für den Projektmanager genauso wichtig wie die Verfügbarkeit aller Funktionen, die der Projektmanager benötigt. Experte A hat das auf die Frage zu der Bewertung einer PMS im Team zusammengefasst:

Ich glaube das wichtigste ist, [...] benutzt es dein Team.

8.2 Kriterienauswertung

Im Folgenden werden die Kriterien, die im Interview abgefragt wurden, nacheinander interpretativ ausgewertet. Direkte Zitate der Experten werden benutzt, sofern auf diese direkt Bezug genommen wird. Eine Auflistung der Kernaussagen und Zitate aller Experten zu jedem Kriterium findet sich in Anhang B.

8.2.1 Grundlegende Funktionalitäten

1.1 Unterstützung für agile Methoden

Relevanz: Agile Methoden sind der Standard in der Spieleentwicklung und finden an vielen Stellen Anwendung, jedoch wird eher selten eine agile Methode in der Ausprägung genutzt, wie zum Beispiel in der Softwareentwicklung, sondern unterschiedliche Methoden teils auch aus dem klassischen Projektmanagement genutzt und so an die Abläufe im jeweiligen Studio angepasst.

Priorität: Die Priorität wurde von allen Experten als hoch bewertet, bis auf Experte C, der die Priorität zunächst als niedrig ansah, jedoch mit steigender Größe des Teams ebenfalls als wichtiger ansieht. Die Diskrepanz erklärt sich hierbei daraus, dass für besonders kleine Teams keine formalen Prozesse genutzt werden und man vor allem als Entwickler eher weniger mit dem Prozessablauf und mehr mit dem Prozess selbst zu tun hat. Abschließend ist die Priorität dieses Kriteriums als hoch zu bewerten, jedoch sinkt die Priorität für besonders kleine Teams aus der Freunde Team Zielgruppe.

Messbarkeit: Die Mehrheit der Experten hat Kanban-Boards und Sprintplanung als wichtigen Punkt hervorgehoben. Alle Experten waren sich jedoch auch einig, dass es das Wichtigste ist, dass man

sich modular aus verschiedenen agilen Konzepten die Teile heraussuchen und nutzen kann, die am besten zu den eigenen Abläufen passen.

1.2 Aufgabenmanagement und Zeitplanung

Relevanz: Dieses Kriterium wurde von allen Kriterien als wichtig angesehen, zwei Experten haben es sogar als einen der entscheidenden Faktoren für PMS bezeichnet.

Priorität: Die Priorität wurde von den meisten Experten als hoch angesehen, jedoch haben Experte C und E die Priorität bei besonders kleinen Teams als eher Mittel angesehen. Das deckt sich mit den Erfahrungen von Expertin A, für die die Arbeit als Projektmanagerin kaum möglich ist, ohne die Zeitdaten zu haben. Aus diesem Grund liegt hier eine skalare Priorisierung vor, bei der die Priorität nach Größe des Teams von Mittel bis Hoch steigt.

Messbarkeit: Als wichtige Features wurden hier vor allem Gantt-Diagramme, Meilensteine oder Story-Point bzw. eine Gegenüberstellung von benötigter zu veranschlagter Zeit genannt. Wichtig war den Experten vor allem, dass es eine Möglichkeit gibt zu identifizieren, welche Aufgaben länger gedauert haben als vorgesehen und wie das Projekt im allgemeinen Zeitrahmen liegt.

1.3 Kommunikations- und Kollaborationswerkzeuge

Relevanz: Bei diesem Kriterium haben die Experten zwischen globaler, direkter Kommunikation und aufgabenspezifischer Kommunikation unterschieden. Während eine Einbindung einer direkten Kommunikationssoftware als Ersatz für zum Beispiel Anwendungen wie Discord keinerlei Relevanz hat, da es genügend auch kostenlose Alternativen gibt, ist die Einbindung von Kommunikation und Ergebnissen zu bestimmten Tasks ein für alle Experten wichtiger Bestandteil von PMS.

Priorität: Die Priorität für dieses Kriterium wird von allen Experten als hoch angesehen, mit einem leichten Übergang zu mittlerer Priorität für besonders kleine Teams.

Messbarkeit: Für Kommunikation zu Aufgaben sollten, laut den Experten, Kommentare sowie das Verlinken anderer Entwickler möglich sein. Ein Feature, das auch von fast allen Experten angesprochen wurde, war ein integrierter Dateiupload für Bilder oder Videos, um einzelnen Tasks mehr Kontext zu geben.

1.4 Gemeinsame Nutzung von Dateien und Dokumentenmanagement

Relevanz: Laut der Experten hat dieses Feature für kleinere Teams gar keine Relevanz und selbst für größere Teams nur in geringem Maße. Als Hauptgrund wurde hier genannt, dass es genügend Alternativen, wie zum Beispiel Google Drive gibt, mit denen ein PMS kaum mit Preis und Komfort konkurrieren könnte. Größere Teams setzten hierbei dann auf Cloudspeicher, Enterprise Lösungen. Als einziges Feature, was eine PMS in dieser Richtung haben könnte, wäre eine vereinfachte Integration der bestehenden Cloudlösung. Aus diesen Gründen wird diese Anforderung aus dem Bogen entfernt.

1.5 Integration mit Entwicklungswerkzeugen

Relevanz: Direkte, feste und vorgeschriebene Integrationen sind eher unerwünscht, was jedoch die Mehrheit der Experten begrüßen würde wäre eine API über die Dinge wie zum Beispiel automatisierte Bugreports ermöglicht werden.

Priorität: Experte E hat angemerkt, dass das Pflegen dieser Funktionen kleine Teams überfordern

könnte, da das Ansprechen der API einen Mehraufwand in der Entwicklung bedeuten könnte. Je größer das Team ist, desto eher sind Kapazitäten für diese Funktionen frei und automatisiertes Bug Reporting, mit den Details zu jedem Bug direkt in der PMS kann für QA (Quality Assurance) eine große Hilfe sein. Die Priorität steigt somit abhängig von der Teamgröße von Mittel bis hoch.

Messbarkeit: Nach Experte C ist eine allumfassende vorab Integration der PMS in die Systeme kaum möglich, da die Projekte sich zu viel voneinander unterscheiden. Um diesen Punkt zu erfüllen wäre somit nicht eine Integration, sondern eine gut dokumentierte API, die es einfach möglich macht, Werkzeuge, die dem Team bei der Entwicklung helfen ins Projekt einzubauen. Aus diesem Grund sollte das Kriterium von "Integration mit Entwicklerwerkzeugen" zu "Erweiterbare API" umbenannt werden.

8.2.2 Benutzerfreundlichkeit und Benutzeroberfläche

2.1 Benutzerfreundliche Oberfläche

Relevanz: Eine leicht verständlich aufgebaute Benutzeroberfläche ist vor allem für die Entwickler relevant. Projektmanager und Owner achten eher darauf, ob alle benötigten Funktionen verfügbar sind, während für einen Entwickler das entscheidendste Kriterium ist, möglichst schnell und möglichst einfach mit diesem Werkzeug klarzukommen. Expertin A hat dabei hervorgehoben, dass viele aktuelle PMS auf die Softwareentwicklung, also Programmierer ausgelegt sind, eine PMS für die Spieleentwicklung jedoch auch von Künstlern, Autoren und vielen kreativen Menschen genutzt wird und damit die Oberfläche auch für diese Personen ansprechend sein muss.

Priorität: Alle Experten haben dem Kriterium hohe Priorität gegeben.

Messbarkeit: Ein wichtiger Punkt, der mehrfach genannt wurde, ist die Einarbeitungszeit, diese sollte möglichst kurz sein und nach Expertin A am besten nicht länger als eine Stunde dauern. Vor allem auf Entwicklerseite würden sich Experte C und E kurze Klickstrecken und eine einstellbare Hauptseite wünschen.

2.2 Anpassungsfähigkeit

Relevanz: Für kleine Entwicklerstudios ist es nach der Meinung der Mehrheit der Experten wesentlich wichtiger, dass die Standardlösung gut funktioniert, da dort keine Zeit für komplizierte Anpassungen vorhanden ist. Experte C möchte als Entwickler keine Anpassungen vornehmen können, jedoch steigt die Relevanz dieses Kriteriums mit der Größe des Teams und es kann vor allem für den Projektmanager eine Hilfe sein, die Oberfläche nach Aufgabenbereichen für die Entwickler voreinzustellen.

Priorität: Für Entwicklerstudios im Bereich der Freunde Teams hat dieses Kriterium wenig bis keine Relevanz, diese steigt über semiprofessionelle Teams mit Mittel bis zu hoch, sobald das Team zur Kategorie der vollständigen Entwicklerstudios gehört.

Messbarkeit: Die Experten sind sich einig, dass Anpassbarkeit an Abläufe und Projekte wichtig ist, allerdings wurde von den meisten Experten auch hervorgehoben, dass es vor allem für kleine Studios die keine Zeit für langwierige Anpassungen haben, wichtig ist, dass auch gut durchdachte Vorlagen existieren.

2.3 Mobile Zugänglichkeit

Relevanz: Für größere Projekte kann es für den Projektmanager nützlich sein, die PMS auch mobil aufrufen zu können, jedoch sind sich alle Experten einig, dass fast alle Arbeit am PC stattfindet und dieses Kriterium keine Relevanz hat. Das Kriterium wird aus diesem Grund entfernt.

8.2.3 Technische Anforderungen

3.1 Cloud-basierte Lösung

Relevanz: Alle Experten sind sich einig, dass die PMS über eine Cloudlösung laufen sollte, um den Anforderungen an einen modernen Arbeitsplatz gerecht zu werden.

Priorität: Alle Experten sehen die Priorität als hoch an.

Messbarkeit: Kleinere und mittlere Teams wollen sich nicht mit dem eigenen Bereitstellen der Cloudlösung beschäftigen. Für Studios, die eine Größe außerhalb des Fokus dieser Arbeit erreichen, kann es jedoch interessant sein, die Server der PMS selber zu verwalten.

3.2 Skalierbarkeit

Relevanz: Alle Experten sind sich einig, dass während eines Projektes die Anzahl der Entwickler nur wenig variiert, allerdings existieren durch Freelancer oder den Publisher immer wieder leichte Fluktuationen im Entwicklungsteam. Es ist wichtig, dass die PMS diese kleinen Anpassungen problemlos vornehmen kann. Größere Teamänderungen passieren nur zwischen Projekten und Teams wachsen nur selten sprunghaft.

Priorität: Je nachdem, ob die Experten den Punkt für hohe oder niedrige Skalierungen gewichtet haben, wurden unterschiedliche Prioritäten der Experten gegeben. Jedoch sind sich alle Experten einig, dass das Kriterium eine hohe Priorität für kleine Teamänderungen hat, große Skalierung aber keine Rolle spielen.

Messbarkeit: Für die meisten Experten ist es wichtig, dass eine Skalierung im kleinen Rahmen möglich ist, Expertin A merkte noch an, dass es einfach möglich sein sollte, auch externe Experten hinzuzufügen.

3.3 Datensicherheit und Datenschutz

Relevanz: Die Experten haben diesen Punkt zwischen Datensicherheit und Datenschutz aufgetrennt, beides wird von allen Experten als wichtig angesehen, jedoch mit leicht unterschiedlichen Erwartungen. Auf der einen Seite sind da Backups und auf der anderen Seite Datenschutzrichtlinien, die eingehalten werden müssen

Priorität: Die Priorität für die Datensicherheit wird von allen Experten für alle Teams als hoch angesehen. Der Datenschutz spielt für kleine Studios eher eine untergeordnete Rolle, nimmt aber durch Vorgaben und Verpflichtungen an Relevanz zu, wenn das Team größer wird.

Messbarkeit: Wichtig ist allen Experten, dass keine Daten verloren gehen, und automatisch regelmäßig Backups erstellt werden. Expertin A hat noch darauf verwiesen, dass größere Studios mitunter auch lokale Backups wünschen. Bei der Datensicherheit waren sich die Experten einig, dass erwartet wird, dass die aktuellen Standards eingehalten werden und Zugangskontrollen stattfinden. Experte E hat zudem Features wie 2-Faktor-Authentifizierung für bestimmte Accounts gefordert.

3.4 Geringe Systemanforderungen

Relevanz: Alle Experten erwarten, dass die marktüblichen Geschwindigkeiten eingehalten werden. Da alle Experten eine Cloudlösung bevorzugen, bezieht sich dieser Punkt auf die Ladezeiten und den Ressourcenverbrauch des Browsers.

Priorität: Kurze und schnelle Ladezeiten bei wenig Ressourcennutzung wurde von allen Experten mit hoher Priorität bewertet.

Messbarkeit: Die Ladezeiten sollten unter einer Sekunde lang sein, sodass beim Arbeiten keine Wartezeiten entstehen.

8.2.4 Kosten und Lizenzierung

4.1 Bezahlbares Preismodell

Relevanz: Die Experten sind sich einig, dass vor allem für kleine Studios der Kostenpunkt ausschlaggebend ist und hier vor allem bei Studios aus der Freunde Team Zielgruppe, wenn sie PMS benutzen eine kostenlose Version bevorzugen. Mit steigender Teamgröße steigen die Anforderungen an eine PMS, jedoch sind auch mehr finanzielle Mittel verfügbar, um eine zu erwerben. Je größer das Studio wird, desto mehr wird neben dem Preis auch auf andere Faktoren geachtet.

Priorität: Hohe Priorität für kleine Entwicklerstudios, die Priorität sinkt jedoch auf mittel mit steigender Größe des Teams.

Messbarkeit: Experte B würde 20\$ pro Monat für angemessen betrachten, Experte E um die 2% der monatlichen Kosten, jedoch nur, wenn die PMS einen Mehrwert liefert, der die Ausgaben rechtfertigt. Besonders kleine Teams sind mitunter auf kostenlose PMS angewiesen, weswegen allein, dass die Software Kosten verursacht, für diese schon ein Ausschlusskriterium sein kann.

4.2 Flexible Lizenzierung

Relevanz: Keiner der Experten sah dieses Kriterium als relevant an, da die Art des Bezahlmodells keine große Relevanz hat, solange es transparent und einsichtig ist. Aus diesem Grund wird dieses Kriterium entfernt, jedoch sollte Transparenz des Bezahlmodells zur Messbarkeit von 4.1 hinzugefügt werden.

8.2.5 Zusätzliche Funktionalitäten

5.1 Zeiterfassung

Relevanz: Daten zur Zeiterfassung zu sammeln kann für alle Teams interessant sein, jedoch unterscheiden sich die Ausprägungen. Expertin A verweist darauf, dass Daten zu Zeiterfassung zu sammeln nur dann relevant ist, wenn man danach mit diesen Daten auch etwas macht. Experte B verweist darauf, dass auch Publisher oder Finanzierungen bestimmte Eckdaten der Zeitznutzung benötigen. Es sollten auf jeden Fall Daten zur Zeiterfassung gesammelt werden, jedoch ist der nötige Detailgrad vom jeweiligen Team abhängig.

Priorität: Bei kleineren Teams wird die Priorität von allen Experten als Mittel angesehen, jedoch steigt die Priorität für semiprofessionelle Teams die im Übergang zu vollwertigen Studios sind bis auf hoch an.

Messbarkeit: Für kleinere Teams ist der grobe Überblick besser, da die Daten für detaillierte Zeiterfassungen nicht ausgewertet werden können. Die Experten sind sich jedoch auch einig, dass eine Minuten genaue Zeiterfassung nicht benötigt wird und eher grobe Zeiträume für bearbeitete Tasks oder noch gröber ganze Projekte wichtig sind. Für Experte E ist vor allem die Vergleichbarkeit von benötigter Zeit über zum Beispiel Story-Points wichtig, um Prognosen genauer treffen zu können.

5.2 Budgetierung und finanzielle Kontrolle *Relevanz:* Expertin A und Experte B sehen diesen Punkt als ein mögliches Feature an, jedoch nicht als zwingend benötigt. Experte C und E sehen für dieses Kriterium keine Relevanz, da Tabellenkalkulationen wie Excel da ausreichen und bereits Abläufe existieren. Die Diskrepanz erklärt sich hier aus persönlichen Gewohnheiten und der Zeit, die dafür benötigt wird, das Tool zu nutzen.

Priorität: Für Entwicklerstudios die sich im semiprofessionellen Bereich oder darüber befinden kann die Funktion mit niedriger Priorität eine "nice to have"-Funktion sein, für kleine Entwicklerstudios ist dieses Feature irrelevant.

Messbarkeit: Um einen Mehrwert zu bieten, müsste das Tool ähnlich wie ein Tabellenkalkulationsprogramm funktionieren und sollte in der Lage sein, auch zu gängigen Tabellenformaten zu exportieren.

5.3 Risikomanagement-Tools

Relevanz: Experte B sieht das Risikomanagement als einen informellen Prozess an, der über Meetings gelöst wird. Auch Experte A und E sehen formales Risikomanagement nicht als nötig an. Was die Experten jedoch von einer PMS erwarten, ist die Priorisierung von Aufgaben oder das Markieren dieser als kritisch. Beide Experten sehen auch die Möglichkeit, dass die PMS über das Erkennen von Abgabeterminen oder Überschreitungen gewisser Kennwerte den Projektmanager benachrichtigt. Die Daten sollten jedoch dann menschlich ausgewertet werden und nicht automatisiert verarbeitet werden.

Priorität: Je kleiner das Team ist, desto eher kann das Risikomanagement nur über Meetings funktionieren und die Priorität für solche Funktionen ist niedrig, sobald das Team wächst und Meetings nicht mehr mit allen stattfinden und es einen dedizierten Projektmanager gibt, steigt die Priorität auf Mittel an.

Messbarkeit: Die PMS sollte in der Lage sein, Erinnerungen oder Warnungen für vom Projektmanager festgelegte Aufgaben oder Kennwerte zurückzugeben. Zudem sollte es eine Möglichkeit geben, Aufgaben bestimmte Markierungen entsprechend ihres Risikos zu geben, sodass der Entwickler diese Information auch hat.

5.4 Dokumentation und Analyse

Relevanz: Expertin A sieht die Dokumentation und Analyse der Kenndaten des Projekts als der zentrale Aufgabenbereich eines Projektmanagers an, aus diesem Grund hat das Kriterium einen hohen Stellenwert, sobald ein Team einen dedizierten Projektmanager hat. Experte B, C und E sind der Meinung das es einige bestimmte Diagramme gibt, die auch für kleine Teams eine Relevanz haben um Prognosen und Ressourcen richtig zu verplanen, jedoch kann dort vieles auch durch Meetings gelöst werden.

Priorität: Die Priorität für Freundes Teams ist noch gering bis mittel, da hier nur wenige Diagramme sinnvoll verwendet werden können. Die Priorität steigt jedoch, sobald ein dedizierter Projektmanager im Team dabei ist und kann für große Teams sogar als hoch angesehen werden.

Messbarkeit: Bun-Down-Charts wurden hier von vielen der Experten genannt, genauso wie Möglichkeiten zur Feststellung von Ressourcennutzung wie zum Beispiel Ressourcenmatrizen. Nach Expertin A wäre auch ein Export in das .csv Format wünschenswert, falls der Projektmanager die Daten der PMS selber weiter verarbeiten will.

5.5 Integration mit Marketingplattformen

Relevanz: Die Experten sind sich einig, dass diese Funktion etwas ist, was sie nicht von einer PMS erwarten würden. Zum einen aus Sicherheitsbedenken, dass etwas durch die PMS an die Öffentlichkeit kommt, was nicht gewollt war oder da genügend gut geeignete externe Tools dafür existieren. Dieses Kriterium wird entsprechend entfernt.

8.2.6 Unterstützung und Schulung

6.1 Kundensupport

Relevanz: Alle Experten sind sich einig, dass, sollte es ein Problem mit der PMS geben, der Kundensupport möglichst schnell antworten und das Problem lösen sollte, denn da eine PMS das Herzstück der Organisation ist, würde ansonsten kaum Arbeit stattfinden.

Priorität: Die Priorität für einen Kundensupport wurde von allen Experten als hoch angesehen, Expertin A hat die Priorität für kleine Entwicklerstudios als gering eingestuft, jedoch haben die anderen Experten dieses Kriterium auch für kleine Studios als hoch priorisiert eingestuft.

Messbarkeit: Der Kundensupport solle nach Ansicht der Experten noch am selben Tag antworten. Experte C war es dabei vor allem wichtig, dass auch technischer Export für zum Beispiel Fragen zur API schnell zur Verfügung steht.

6.2 Dokumentation und Tutorials

Relevanz: Alle Experten sahen dieses Kriterium als relevant an, um Einarbeitungszeiten zu verkürzen und Arbeitsabläufe nachzuvollziehen.

Priorität: Alle Experten haben dieses Kriterium als hoch priorisiert eingestuft.

Messbarkeit: Hervorgehoben wurde hier von allen Experten eine gut strukturierte und leicht zu durchsuchende Wissensdatenbank. Klicktutorials oder bebilderte Tutorials können nach Expertin A und Experte B vor allem die Einarbeitungszeit verkürzen.

8.2.7 Kulturelle und teambezogene Überlegungen

7.1 Sprachliche Unterstützung

Relevanz: Alle Experten waren sich einig, dass Englisch als Sprache für die PMS ausreichend ist und Übersetzungen für die Wahl der PMS keine Rolle spielen, sofern Englisch angeboten wird. Aus diesem Grund wird dieses Kriterium entfernt.

7.2 Anpassung an den Arbeitsablauf des Teams

Relevanz: Die Experten sind bei diesem Kriterium alle der Meinung, dass Anpassungen möglich sein müssen, um existierende und funktionierende Abläufe beizubehalten, jedoch kann es auch helfen, wenn Arbeitsabläufe vorgegeben werden, falls die bisherigen Lösungen des Teams nicht funktionieren. Auch sind sich die Experten einig, dass die Art und Weise dieser Anpassungen stark vom Projekt und vom jeweiligen Team abhängig ist.

Priorität: Die Priorität wird von den Experten als Mittel bis Hoch eingestuft.

Messbarkeit: Den Experten fiel es schwer hier eine genaue Messbarkeit vorzugeben, da die Art und Weise, wie sich die PMS anpassen muss und wo Dinge vorgegeben sein sollten, je nach Team und Projekt unterschiedlich ist. Eine PMS sollte dementsprechend im besten Fall vordefinierte Vorgänge in allen Bereichen vorgeben, sich jedoch auch anpassen können, falls es vom Team gewünscht ist.

8.2.8 Vereinfachung und Minimalismus

8.1 Vermeidung einer Überfrachtung mit Funktionen

Relevanz: Alle Experten sind sich einig, dass eine Überforderung der Nutzer durch zu viele Funktionen durch die Oberfläche der PMS vermieden werden sollte. Experte C verweist hier vor allem auf die Sicht der Entwickler, die im besten Fall intuitiv mit der PMS umgehen können sollten.

Priorität: Aus Sicht des Projektmanagers ist, laut Experte A und B, die Priorität für dieses Kriterium als Mittel anzusehen, da der Projektmanager die Zeit hat sich in das Programm, auch wenn es komplexer ist, einzuarbeiten. Aus Sicht des Entwicklers ist das Kriterium jedoch so wie von Experte C und E als hoch einzuschätzen, da die Entwickler nicht die Zeit für große Einarbeitung oder Anpassungen haben. Die Priorität ist deswegen insgesamt als hoch zu bewerten.

Messbarkeit: Vor allem für Entwickler sollte die Oberfläche nach Experte C besonders aufgeräumt und leicht überschaubar sein. Experte A und E sind zudem der Meinung, dass Möglichkeiten zur Anpassung oder zum Ausblenden von nicht benötigten Funktionen wichtig sind.

8.2.9 Motivation und Teammoral

9.1 Transparenz und Nachverfolgung

Relevanz: Alle Experten sind sich einig, dass die PMS unterstützen kann, um Feedback zu liefern und Fortschritte anzuzeigen, jedoch das eher eine Hilfe für den Projektmanager ist und persönliche Interaktion dadurch nicht ersetzt werden kann.

Priorität: Die Experten haben dieses Kriterium sehr unterschiedlich gewichtet, je nach Interpretation der Funktionen, die die PMS leisten kann. Da jedoch die persönliche Interaktion bei allen Experten im Vordergrund stand und die PMS eher über Feedback an den Projektmanager unterstützen kann, ist die Priorität als niedrig für kleine Teams anzusehen und ab semiprofessionellen Teams mit Projektmanager als Mittel.

Messbarkeit: Nach Experte E kann die PMS durch kleine Bestärkungen und eine grobe Projektübersicht zumindest ein wenig bei der Motivation helfen. Experte B sieht hier eher einige Metriken im Vordergrund und die Hauptmotivation im Fortschritt, den das zu entwickelnde Spiel macht.

9.2 Anerkennungs- und Feedback-Mechanismen

Relevanz: Expertin A meint, dass die Anerkennung und Feedback Mechanismen zwar eine gute Ergänzung sein können, aber im Vergleich zu persönlichem Feedback keine Relevanz haben. Die Experten B, C und E schließen sich der Meinung, dass Feedback persönlich gemacht werden sollte an. Experte C merkt zudem an, dass durch die schlechte Vergleichbarkeit von Aufgaben das automatische System auch demotivierend wirken kann. Aus diesen Gründen wurde dieses Kriterium entfernt.

9 Fazit und Einordnung

9.1 Ergebnisse für den Anforderungsbogen

Die folgende Grafik stellt die in Kapitel 8.2 diskutierten Ergebnisse und die damit verbundenen Änderungen an den Anforderungsbogen grafisch dar:



Abbildung 9.1: Veränderung des Anforderungsbogens durch das Experteninterview

Durch die Aufteilung des Anforderungsbogens in die unterschiedlichen Zielgruppen wird deutlich, dass mit steigender Größe des Teams auch die Anforderungen an eine PMS steigen. Zudem ist es wichtig zu bedenken, dass PMS nicht nur für einen Projektmanager gut funktionieren muss, sondern sich auch an die Entwickler anpassen muss. In Freundes Teams, in denen es keinen dedizierten Projektmanager gibt, fällt diese Zielgruppe dadurch vollständig weg. In den Interviews ist deutlich geworden, dass in diesen kleinen Teams neben finanziellen Fragen vor allem der durch die PMS entstehende Overhead eine große Relevanz hat. Ohne Projektmanager gibt es in diesen Teams keine Person, die neben der Arbeit am Projekt die Zeit hat sich in eine Komplexe PMS Lösung einzuarbeiten und selbst wenn, würden viele der zusätzlichen Daten einem so kleinen Team nicht helfen. Was eine PMS jedoch schon leisten kann ist es Teams an grundsätzliche Funktionen und Arbeitsabläufe heranzuführen. Kleine Teams benötigen viel Freiheit, um sich kreativ zu entfalten, da die Projekte häufig sehr individuelle Abläufe haben eine grundsätzliche Struktur ermöglicht es außerdem einen besseren Überblick über das Projekt zu haben und kann kleinen Teams dabei helfen diese Projekte zu einem Abschluss zu bringen. Semiprofessionelle Projekte sind bereits einen Schritt weiter und benutzen und erwarten auch mehr Funktionalitäten von einer PMS.

9.1.1 Diskussion des Anforderungsbogens

Abschließend ist der überarbeitete Bewertungsbogen eine erste Einordnung für den Bedarf und die Ausprägung von PMS in der Indiespieleentwicklung. Während der Bogen zwar einen guten Überblick über grundsätzliche Funktionen und die Erwartungen der Spieleentwickler an eine PMS widerspiegelt, müssten viele der Unterpunkte noch einmal, auch mit quantitativen Umfragen, überprüft werden. Durch die Experteninterviews konnte jedoch die Grundrichtung festgestellt werden, auf dessen Basis die Anforderungen genauer beleuchtet werden können.

9.2 Ergebnisse für PMS

9.2.1 Unterschiedliche Zielgruppen

Eine der großen Erkenntnisse durch die Experteninterviews war die Einteilung der vielen verschiedenen Indiestudios aus der Sicht von PMS in drei Gruppen. Die Anforderungen dieser Gruppen variieren und in jedem Bereich ist auch die Konkurrenz durch andere Software anders verteilt, da die Priorisierungen, wie durch den Bewertungsbogen gezeigt wurde, nicht konstant sind.

PMS für "Freunde Teams"

Der Name, der dieser Gruppe gegeben wurde, lässt auf die Zusammensetzung schließen, diese kleinen Teams arbeiten meist nur mit wenig, manchmal sogar zunächst ohne finanzielle Unterstützung. Kommunikation findet dabei meistens mit dem ganzen Team statt, wodurch die Kommunikationswege sehr kurz sind und jeder Entwickler den anderen auch persönlich kennt. Bei dieser Zielgruppe zählen hauptsächlich Kosten und Einarbeitungszeit. Die Software, mit der sich eine PMS hier vergleichen muss, sind kostenlose Dienste wie Google Cloud, Discord oder Miro Boards. Gegenüber diesen kostenlosen Werkzeugen muss die PMS einen Mehrwert bieten und zudem auch kostenlos sein. Aus Sicht eines Unternehmens, welches PMS zur Verfügung stellt, ist diese Zielgruppe jedoch trotzdem interessant, da wie Experte C anmerkte:

Ich denke, mit der PMS, mit der man anfängt, bei der bleibt man auch.

Wenn also eine PMS Lösung kostenlos Vorteile gegenüber anderen kostenlosen Alternativen bietet und sich die Teams einmal an die Arbeitsabläufe und Oberfläche des PMS gewöhnt haben, ist die Wahrscheinlichkeit, dass diese, wenn sie wachsen, zur Bezahlversion der PMS übergehen, sehr hoch. Sofern die Software neben den Anforderungen der kleinen Teams auch mit den gestiegenen Anforderungen durch die semiprofessionellen Teams mithalten kann.

9.2.2 Die Relevanz von PMS in der Indiespieleentwicklung

Projektmanagementsoftware ist auch in der Indiespieleentwicklung ab einer gewissen Größe eine Notwendigkeit und ein Projekt ohne für diese Teams kaum machbar. Jedoch haben sich auch Besonderheiten herauskristallisiert, auf die eine PMS besonders in diesem Bereich achten muss, um benutzt zu werden:

Interdisziplinäre Teams

Noch unterschiedlicher als in den meisten anderen Projekten ist in der Spieleentwicklung die Interdisziplinarität der einzelnen Entwickler. Je größer die Teams werden, desto spezialisierter sind die jeweiligen Entwickler und auch bei kleinen Teams gibt es meist eine Aufteilung zwischen den Künstlern und den Programmierern. Mit der unterschiedlichen Herangehensweise an Aufgaben von diesen Personen muss eine PMS umgehen können. Neben der Funktionalitäten, die eine PMS mit sich bringt, muss gewährleistet sein, dass diese PMS auch von allen Entwicklern verwendet werden kann und die Entwickler sich dabei wohlfühlen. Dabei können zum Beispiel unterschiedliche Ansichten, je nach Aufgabenbereich, helfen.

Dynamische Prozesse

In der Spieleentwicklung gibt es keine Festen Prozesse, die von jedem Team genutzt werden. Das Experteninterview hat verdeutlicht, wie unterschiedlich die Herangehensweisen innerhalb von Teams sein können und dass auch die Priorisierung von Projekt zu Projekt abweichen kann. Aus diesem Grund muss eine PMS in der Lage sein, nicht benötigte Funktionen zu verstecken und keine festgelegten Prozesse erzwingen. So nutzen viele Entwicklerteams unterschiedliche Methoden aus agilem oder sogar aus dem klassischen Projektmanagement und die PMS sollte sich an diese Prozesse anpassen können. Jedoch kann die PMS vor allem auch neuen Teams durch vorgeschlagene Prozesse unter die Arme greifen, um nachzuhelfen, falls der bestehende Prozess im Team noch nicht so wie gewünscht funktioniert.

Komplexität und Übersicht in Einklang bringen

Eine Balance zwischen den in den beiden vorangegangenen Abschnitten beschriebenen Einstellungs- und Anpassungsmöglichkeiten zu finden ist vermutlich die schwerste Aufgabe für eine PMS in der Spieleentwicklung - denn all diese Anpassungen erhöhen die Komplexität und den Overhead bis die Software effizient genutzt werden kann enorm. Für semiprofessionelle Teams und vollwertige Studios wäre eine Projektmanagementansicht und eine Entwickleransicht, die der Projektmanager über Templates konfigurieren kann, eine Lösung, die bei den Experten gut angekommen ist. Auf diese Weise müsste sich nur der Projektmanager, dessen Aufgaben am meisten mit der PMS zu tun haben, sich in diese einarbeiten und könnte dann die Arbeitsbereiche der Entwickler optimal einrichten. Entwicklerstudios der Kategorie Freunde Teams haben nicht die Zeit sich so viel mit der Anpassung einer Software zu beschäftigen und würden dann eher eine schlechtere Software benutzen, die aber schneller einsatzfähig ist. Besonders für diese Studios ist es wichtig, dass die Software neben den Einstellungsöglichkeiten auch vorgefertigte direkt nutzbare Konfigurationen oder Templates anbietet, mit denen ein kleines Team direkt in die Produktion starten kann.

9.3 Fazit

Abschließend liefert diese Arbeit einen ersten Eindruck der Herausforderungen, Anforderungen und Zielgruppen, die für PMS in der Indie Spieleentwicklung wichtig sind. Der Anforderungsbogen konnte überprüft und überarbeitet werden, jedoch sind aufgrund der Natur von Experteninterviews weitere Untersuchungen der einzelnen Aspekte nötig. Die Einordnung der Zielgruppen kann Entwicklern von PMS dabei helfen, ihre Kunden besser zu verstehen und ihre Software so zu optimieren, sodass diese Zielgruppen besonders angesprochen werden.

Projektmanagement Software ist bereits jetzt ein fester Bestandteil der Indie Spieleentwicklung, werden jedoch meistens in Kombination mit anderen Werkzeugen genutzt, sodass es auch wichtig ist aktuellen Trends zu folgen und die PMS so zu optimieren, dass sie sich mit bereits genutzten Werkzeugen ergänzt. Sowie für die meiste andere Anwendungssoftware gilt aber auch für PMS, dass sie nur dann effektiv ist, wenn sie auch von allen am Projekt Beteiligten genutzt wird.

Anhang A: Experteninterview Skript

A.1 Einführung und Rahmenbedingungen

A.1.1 Einführung durch den Interviewer:

Hallo, und vielen Dank, dass Sie sich bereit erklärt haben, an diesem Interview teilzunehmen. Mein Name ist Christian Klumpp, und ich führe eine Forschungsarbeit im Rahmen einer Bachelorarbeit durch, die sich mit der Identifizierung der wichtigsten Anforderungen an eine Projektmanagement-Software für Indie-Spiele-Entwicklerteams beschäftigt. Ihre Einblicke als Indie-Entwickler/PMS-Entwickler sind von unschätzbarem Wert, wenn es darum geht, diese Kriterien zu verfeinern und sicherzustellen, dass sie mit realen Bedürfnissen und Branchenpraktiken übereinstimmen.

A.1.2 Kontext des Interviews:

Das Ziel dieses Interviews ist es, eine Reihe von möglichen Anforderungen für PMS im Kontext der Entwicklung von Indie-Spielen zu überprüfen. Ich werde nach der Relevanz der einzelnen Anforderungen, ihrer Priorität für kleine Indie-Teams und, wie eine „gute Lösung“ oder ein messbarer Standard für diese Anforderungen aussehen könnte, fragen. Ihr Feedback wird uns helfen, den Bewertungsbogen zur Beurteilung verschiedener PMS-Tools zu überarbeiten und zu verbessern.

A.1.3 Haftungsausschluss zur Datennutzung und -aufzeichnung:

Ich möchte Sie darauf hinweisen, dass ich mit Ihrer Erlaubnis dieses Interview zu Transkriptionszwecken aufzeichnen werde. Das aufgezeichnete Material wird vertraulich behandelt, und nur ich und mein akademischer Betreuer werden Zugang zu den Rohdaten haben. Die Ergebnisse werden in der Abschlussarbeit in zusammengefasster und anonymisierter Form dargestellt, sodass kein einzelner Befragter identifiziert werden kann. Ihre Teilnahme ist freiwillig, und Sie können jederzeit die Beantwortung einer Frage verweigern oder sich aus dem Interview zurückziehen, ohne dass dies Konsequenzen hat.

Einverständniserklärung: Habe ich Ihr Einverständnis, dieses Interview zu den genannten Zwecken aufzuzeichnen und sind Sie damit einverstanden, unter diesen Bedingungen fortzufahren?
«Abwarten auf Antwort»

A.2 Allgemeine Fragen zur Person

- Könnten Sie kurz Ihre Rolle und Erfahrung in der Indie-Game-Entwicklung/PMS-Branche beschreiben?
- An welcher Art von Projekten haben Sie gearbeitet, und wie vertraut sind Sie mit der Verwendung von PMS-Tools in diesem Zusammenhang? Welche PMS Tools haben Sie verwendet?

A.3 Fragen zur Überprüfung der Kriterien

«Themeneinleitung»

Für jedes der Kriterien werde ich zunächst eine kurze Erläuterung geben und dann die folgenden drei Aspekte abfragen:

Relevanz: Ist dieses Kriterium für ein Indie-Studio relevant?

Priorität: Wie wichtig oder dringend ist dieses Kriterium im Vergleich zu anderen?

Messbarkeit: Wie können wir feststellen, ob ein PMS dieses Kriterium tatsächlich erfüllt?

A.3.1 Grundlegende Funktionalitäten

1.1 Unterstützung für agile Methoden (hohe Priorität)

Kriterium: Das PMS sollte agile Praktiken (z. B. Scrum, Kanban) unterstützen, um iterative Entwicklung und häufige Änderungen zu ermöglichen.

Relevanz: Wie relevant ist Ihrer Erfahrung nach die Fähigkeit, agile Methoden zu unterstützen, für das PMS eines Indie-Studios?

Priorität: Würden Sie dies als eine Anforderung mit hoher Priorität für Ihr eigenes oder ähnliche Teams betrachten? Warum oder warum nicht?

Messbarkeit: Welche Funktionen oder Indikatoren würden zeigen, dass ein PMS agile Arbeitsabläufe effektiv unterstützt? Welche spezifischen Boards, Sprint-Planungstools oder Anpassungsfunktionen würden Sie zum Beispiel als wesentlich erachten?

1.2 Aufgabenmanagement und Zeitplanung (Hohe Priorität)

Kriterium: Das PMS sollte zuverlässige Werkzeuge für die Erstellung, Zuweisung, Priorisierung und Planung von Aufgaben mit Meilensteinen und Abhängigkeiten bieten.

Relevanz: Wie wichtig ist eine umfassende Aufgabenverwaltung und Terminplanung für Indie-Teams?

Priorität: Auf einer Skala von niedrig bis hoch, wie wichtig ist diese Funktionalität für Ihre Projekte?

Messbarkeit: Was macht eine Aufgabenverwaltungsfunktion „gut“? Wäre es die einfache Erstellung von Aufgaben, visuelle Zeitleisten, Gantt-Diagramme oder spezifische Metriken wie der Prozentsatz der termingerecht gelieferten Aufgaben?

1.3 Kommunikations- und Kollaborationswerkzeuge (hohe Priorität)

Kriterium: Integrierte Kommunikationsfunktionen, um eine reibungslose Zusammenarbeit zu ermöglichen, insbesondere für entfernte oder verteilte Indie-Teams.

Relevanz: Wie relevant ist die integrierte Kommunikation innerhalb des PMS selbst, da viele Teams separate Kommunikationstools verwenden (z. B. Discord)?

Priorität: Würden Sie ein PMS, das integrierte Kommunikation bietet, einem anderen vorziehen? Und warum?

Messbarkeit: Welche Kommunikationsfunktionen (z. B. Diskussionen mit Threads, @Mentions, Dateikommentare) würden Sie als wesentlich oder als Indikator für ein gutes Kommunikationsmodul betrachten?

1.4 Gemeinsame Nutzung von Dateien und Dokumentenmanagement (hohe Priorität)

Kriterium: Einfache gemeinsame Nutzung, Versionierung und Verwaltung von Dokumenten und digitalen Assets (Grafiken, Audio, Codeschnipsel).

Relevanz: Wie wichtig ist eine integrierte Dateifreigabe für den Arbeitsablauf eines Indie-Teams?

Priorität: Ist diese Funktion im Vergleich zu anderen Kriterien ein Muss oder ein Nice-to-have?

Messbarkeit: Wonach würden Sie in Bezug auf Speicherkapazität, Versionskontrolle oder einfachen Zugriff suchen, um zu entscheiden, ob ein PMS die Dateiverwaltung effektiv handhabt?

1.5 Integration mit Entwicklungswerkzeugen (hohe Priorität)

Kriterium: Fähigkeit zur Integration mit gängigen Entwicklungswerkzeugen (z. B. Git, Unity, Unreal Engine) und Produktivitätsplattformen. *Relevanz:* Wie relevant ist die Integration in bestehende Toolchains für Indie-Entwickler?

Priorität: Würden Sie die Integrationsfähigkeiten ganz oben auf Ihre Anforderungsliste setzen, oder könnten Sie die Prioritäten zurückstellen?

Messbarkeit: Welche Art von Integrationen (z. B. automatische Build-Trigger, Pull-Request-Links, Asset-Synchronisation) bedeuten, dass ein PMS eine „gute“ Integrationslösung bietet? Welche Tools sollten integrierbar sein?

A.3.2 Benutzerfreundlichkeit und Benutzeroberfläche

2.1 Benutzerfreundliche Oberfläche (hohe Priorität)

Relevanz: Wie wichtig ist eine intuitive und einfach zu bedienende Schnittstelle für kleinere Teams, die möglicherweise keine eigenen Projektmanager haben?

Priorität: Würden Sie UI/UX als einen kritischen Faktor bei der Wahl Ihres PMS einstufen?

Messbarkeit: Wie würden Sie die Benutzerfreundlichkeit bewerten? Zum Beispiel minimale Einarbeitungszeit, einfache Navigation oder Bewertungen der Benutzerzufriedenheit? Welche konkreten Features haben Sie bisher als besonders hilfreich empfunden?

2.2 Anpassungsfähigkeit (Mittlere Priorität)

Relevanz: Ist die Möglichkeit der Anpassung von Dashboards und Workflows für Indie-Teams, die möglicherweise einzigartige Prozesse haben, relevant?

Priorität: Würden Sie dies als mittlere Priorität betrachten, oder würde es für Ihr Team einen höheren oder niedrigeren Stellenwert haben?

Messbarkeit: Wie würden Sie feststellen, ob die Anpassungsoptionen angemessen sind? Zum Beispiel die Anzahl der anpassbaren Felder, Dashboard-Layouts oder die Unterstützung für benutzerdefinierte Workflows?

2.3 Mobile Zugänglichkeit (niedrige Priorität)

Relevanz: Ist der mobile Zugriff für Ihr Team notwendig?

Priorität: Würden Sie angesichts begrenzter Ressourcen in ein PMS investieren, das eine mobile

App oder responsive Design bietet, oder ist dies weniger wichtig?

Messbarkeit: Ja/Nein

A.3.3 Technische Anforderungen

3.1 Cloud-basierte Lösung (hohe Priorität)

Relevanz: Wie relevant ist ein cloudbasiertes PMS für Remote-Teams oder die Zusammenarbeit?

Priorität: Würden Sie sagen, dass dies in modernen Entwicklungsumgebungen nicht verhandelbar ist?

Messbarkeit: Welche Maßstäbe oder Indikatoren (z. B. garantierte Betriebszeit, einfacher Fernzugriff) definieren eine „gute“ cloudbasierte Lösung?

3.2 Skalierbarkeit (Mittlere Priorität)

Relevanz: Wie wichtig ist die Skalierbarkeit für kleinere Indie-Teams, die zwar wachsen können, aber dennoch relativ klein bleiben?

Priorität: Ist die Skalierbarkeit etwas, das Sie von Anfang an in Betracht ziehen würden, oder erst, wenn das Team expandiert?

Messbarkeit: Woran lässt sich eine gute Skalierbarkeit erkennen? Zum Beispiel eine stabile Leistung bei mehr Benutzern oder die Bewältigung größerer Projektteile ohne Verzögerung?

3.3 Datensicherheit und Datenschutz (hohe Priorität)

Relevanz: Wie wichtig sind den Indie-Entwicklern die Datensicherheit und der Schutz des geistigen Eigentums?

Priorität: Ist dies eine hohe Priorität angesichts des Risikos des Verlusts einzigartiger kreativer Ressourcen?

Messbarkeit: Welche Sicherheitsmerkmale (Verschlüsselung, Backups, Zugangskontrolle) würden darauf hinweisen, dass ein PMS in dieser Hinsicht „gut“ ist?

3.4 Geringe Systemanforderungen (mittlere Priorität)

Relevanz: Wie wichtig ist ein geringer Ressourcenbedarf für Indie-Teams, die mit verschiedenen Hardwarekonfigurationen arbeiten?

Priorität: Würden Sie die Systemeffizienz als kritisch oder nur als angenehm einstufen?

Messbarkeit: [y/n]

A.3.4 Kosten und Lizenzierung

4.1 Bezahlbares Preismodell (hohe Priorität)

Relevanz: Wie empfindlich sind Indie-Teams in Bezug auf die Preisgestaltung von PMS-Lösungen?

Priorität: Wären zu hohe Kosten ein Hindernis für den Abschluss?

Messbarkeit: Welcher Prozentsatz des Gesamtbudgets oder der monatlichen Einnahmen des Projekts wäre akzeptabel, um ihn für PMS aufzuwenden?

4.2 Flexible Lizenzierung (Mittlere Priorität)

Relevanz: Ist eine flexible Lizenzierung (Pay-as-you-go, einfache Skalierung der Benutzer) für kleinere Teams wichtig?

Priorität: Erachten Sie dies im Vergleich zu den Kernfunktionalitäten als wichtig?

Messbarkeit: Welche Preisstrukturen oder Lizenzmodelle würden Sie als fair und überschaubar ansehen?

A.3.5 Zusätzliche Funktionalitäten

5.1 Zeiterfassung (Mittlere Priorität)

Relevanz: Halten Indie-Teams die Zeiterfassung für unverzichtbar oder eher für ein „nice-to-have“ für Planung und Retrospektiven?

Priorität: Ist sie von mittlerer Priorität, oder würden Sie sie höher oder niedriger einstufen?

Messbarkeit: Wie würde eine gute Zeiterfassungsfunktion aussehen - genaue Protokolle, automatische Erfassungsoptionen oder Berichte, die als Grundlage für Projektschätzungen dienen?

5.2 Budgetierung und finanzielle Kontrolle (mittlere Priorität)

Relevanz: Wie relevant ist die integrierte Budgetierung für kleine Teams mit begrenzter finanzieller Komplexität?

Priorität: Hätte dies eine mittlere Priorität im Vergleich zu den Kernfunktionen des Projektmanagements?

Messbarkeit: Welche Art von Finanzübersichten oder Kostenwarnungen würden ein „gutes“ Budgetierungsinstrument darstellen?

5.3 Risikomanagement-Tools (niedrige Priorität)

Relevanz: Benötigen Indie-Teams formale Risikomanagement-Tools oder können sie Risiken informell managen?

Priorität: Würden Sie zustimmen, dass dies eine geringe Priorität hat, oder könnte es wichtiger sein?

Messbarkeit: Welche Merkmale - Risikoregister, Pläne zur Risikominderung - würden eine wirksame Unterstützung des Risikomanagements bedeuten?

5.4 Dokumentation und Analyse (mittlere Priorität)

Relevanz: Sind Berichte und Analysen wertvoll für Entwicklungsentscheidungen in kleinen Teams?

Priorität: Wie wichtig ist dies im Vergleich zu anderen Funktionen?

Messbarkeit: Welche Berichte (z. B. Geschwindigkeitsdiagramme, Burn-Down-Diagramme, Ressourcennutzung) würden Sie benötigen, um die Dokumentation eines PMS als gut zu bewerten?

5.5 Integration mit Marketingplattformen (niedrige Priorität)

Relevanz: Ist die Integration von Marketing-Tools in das PMS relevant, oder bevorzugen Indie-Teams separate Lösungen für das Marketing?

Priorität: Würden Sie dies als eine niedrige Priorität betrachten, oder wird es relevanter, wenn das Spiel kurz vor der Veröffentlichung steht?

Messbarkeit: Woran würde sich eine gute Integration erkennen lassen - an automatisierten sozialen Beiträgen, einfachen Analysen oder der Verfolgung des Feedbacks der Community?

A.3.6 Unterstützung und Schulung

6.1 Kundensupport (hohe Priorität)

Relevanz: Wie wichtig ist ein reaktionsschneller und hilfreicher Kundensupport?

Priorität: Ist der Support so wichtig, dass er eine hohe Priorität hat?

Messbarkeit: Würden schnelle Reaktionszeiten, engagierte Support-Mitarbeiter oder umfangreiche FAQs einen guten Support definieren?

6.2 Dokumentation und Tutorials (Mittlere Priorität)

Relevanz: Sind umfassende Anleitungen und Dokumentationen für kleine Teams notwendig?

Priorität: Mittlere Priorität scheint angemessen - würden Sie zustimmen?

Messbarkeit: Welche Dokumentationsmerkmale (Video-Tutorials, durchsuchbare Wissensdatenbank) machen eine gute Schulungsressource aus?

A.3.7 Kulturelle und teambezogene Überlegungen

7.1 Sprachliche Unterstützung (niedrige Priorität)

Relevanz: Ist die mehrsprachige Unterstützung für Ihr Team relevant oder reicht Englisch aus?

Priorität: Hat dies eine niedrige Priorität, oder könnte es in bestimmten Kontexten wichtig sein?

Messbarkeit: Wie würden Sie entscheiden, ob die Sprachunterstützung angemessen ist?

7.2 Anpassung an den Arbeitsablauf des Teams (hohe Priorität)

Relevanz: Wie wichtig ist es, dass sich das PMS nahtlos in Ihre bestehenden Arbeitsabläufe einfügt?

Priorität: Hat dies hohe Priorität, oder ist es etwas, das Sie anpassen können?

Messbarkeit: Wäre eine gute Lösung eine, bei der nur minimale Änderungen an Ihren bestehenden Prozessen erforderlich sind, oder eine, bei der das PMS leicht konfiguriert werden kann, um Ihren Arbeitsabläufen zu entsprechen?

A.3.8 Vereinfachung und Minimalismus

8.1 Vermeidung einer Überfrachtung mit Funktionen (hohe Priorität)

Relevanz: Wie wichtig ist es, dass das PMS einfach bleibt und Sie nicht mit unnötigen Funktionen überschüttet?

Priorität: Würden Sie die Einfachheit als eine der wichtigsten Anforderungen betrachten?

Messbarkeit: Wie würden Sie eine Überfrachtung mit Funktionen erkennen - vielleicht zu viele Registerkarten, eine komplexe Navigation oder unnötige Add-ons?

A.3.9 Motivation und Teammoral

9.1 Transparenz und Nachverfolgung (mittlere Priorität)

Relevanz: Trägt eine transparente Projektverfolgung dazu bei, das Team zu motivieren, indem sie Fortschritte aufzeigt?

Priorität: Ist dies eine mittlere Priorität, oder würden Sie es anders einstufen?

Messbarkeit: Welche Indikatoren - wie Fortschrittsbalken, Prozentsätze der Fertigstellung oder öffentliches Lob - würden eine gute Transparenz zeigen?

9.2 Anerkennungs- und Feedback-Mechanismen (niedrige Priorität)

Relevanz: Ist es wichtig, dass das PMS über integrierte Anerkennungsinstrumente oder Feedbackschleifen für die Arbeitsmoral verfügt?

Priorität: Würden Sie diesem Punkt eine niedrige Priorität einräumen?

Messbarkeit: Welche Formen der Anerkennung (Abzeichen, Meilensteine, Feedback von Kollegen) wären von Vorteil?

A.4 Abschließende Fragen

- Gibt es Kriterien, die wir besprochen haben, die Ihrer Meinung nach neu priorisiert werden sollten (entweder höher oder niedriger)?
- Gibt es zusätzliche Anforderungen oder Funktionalitäten, die Ihrer Meinung nach übersehen wurden?
- Wie messen Indie-Teams Ihrer Meinung nach am besten den Gesamterfolg oder die Eignung eines PMS?
- Welches der besprochenen Kriterien würden sie als wichtigstes ansehen?

A.5 Abschluss

A.5.1 Schlusswort

Vielen Dank für Ihre Zeit und Ihre Erkenntnisse. Ihr Beitrag wird zur Überarbeitung und Verbesserung des Bewertungsformulars verwendet, damit es die tatsächlichen Bedürfnisse und Prioritäten von Indie-Spielentwicklern besser widerspiegelt. Wenn Sie Fragen haben oder die Ergebnisse nach Abschluss der Analyse überprüfen möchten, lassen Sie es mich bitte wissen.

A.5.2 Erinnerung an die Datenverarbeitung:

Zur Erinnerung: Die Aufnahmen und Transkripte werden vertraulich behandelt und ausschließlich für dieses Forschungsprojekt verwendet. Sollten Sie Bedenken haben oder Ihre Daten zurückziehen wollen, können Sie mich gerne über cklumpp@hs-mittweida.de kontaktieren.

«Interview Ende»

Anhang B: Experteninterview Transkript

B.1 Grundlegende Funktionalitäten

1.1 Unterstützung für agile Methoden (hohe Priorität)

A:

Relevanz: Agile Prozesse werden auf jeden Fall genutzt

Priorität: hoch

Messbarkeit: Kanban-Boards, Meetingverwaltung, Stand-ups, besprochenes aufnehmen, um Listen abzuarbeiten, Gantt-Charts - Es muss eine Darstellungsmöglichkeit der agilen Abläufe geben.

Zitate:

Rein agile Software funktioniert auf jeden Fall nicht [...], es ist aber schon wichtig, das [die PMS] sollte das auf jeden Fall können, vor allem weil ganz viele Leute nicht so gerne mit Listen arbeiten, sondern wirklich eher mit Kanban-Boards, das ist ein Standard in den sich einfach viele Leute eingearbeitet haben, in der Spieleindustrie[. . .]

Ich glaube vor allem im agilen Kontext wäre es wichtig, dass das [Darstellungsmöglichkeiten für agile Abläufe] da hereingespielt, und bekannte Arbeitsabläufe [...] unterstützt werden

B:

Relevanz: Ist der Standard in der Industrie

Priorität: hoch

Messbarkeit: Kanban-Boards, Ticketsystem, Zeitschätzungen und Einordnung in unterschiedliche Bereiche

Zitate:

Das ist so der Standard vom Arbeiten, das heißt nicht, dass jemand Scrum oder Kanban machen muss oder eins von beidem, aber eine relativ offene Arbeitsweise mit agilem vorgehen ist Standard

C:

Relevanz: Kann relevant sein, muss für kleine Teams aber nicht existieren *Priorität:* niedrig, steigt mit Größe *Messbarkeit:* / *Zitate:*

Ich selbst würde sagen, nicht, dass ein PMS das unterstützen muss [...] das hängt dann denke ich am Teamlead oder Projektmanager, ob sich das lohnt

Zusammen iterativ an Projekten arbeiten hat in einem kleinen Studio ohne das agile Vorgehen besser geklappt als im großen mit

Für die Entwickler selber spielt es eine geringere Rolle in welchem System sie arbeiten, davon bekommt man als Entwickler nicht viel mit

D:

Relevanz: Entwicklerstudios nutzen hybride Modelle, bei denen PMS, sich je nachdem, was gebraucht wird, anpasst. Priorität: hoch Messbarkeit: Unterstützung für zyklische Arbeitsplanung, Sprints, Möglichkeit sich seine eigene Methode basteln kann Zitate:

Prinzipiell ist die Softwareentwicklung heutzutage agil und in der Spieleentwicklung hat man eine Art hybrides Modell, wo man sich pickt, was man gerade fürs eigene Team braucht. [...] Ich denke, das ist alles so ein bisschen aufgebrochen

Ein großer Unterschied zwischen Computerspielen und anderer Software ist, dass man feste Releasetermine hat, mit denen man planen muss.

Ich habe in den letzten Jahren noch nie ein Team genutzt, dass wirklich alles von Scrum umgesetzt hat

E:

Relevanz: Kein ausschließliches Nutzen von agilen Methoden Priorität: hoch Messbarkeit: Kanban Boards, eine Modularität der Methoden ist relevant Zitate:

Wenn wir Sachen benutzen, machen wir so einen Mix aus Scrum und Kanban. [...], aber wenn dann wichtige Aufgaben reinkommen, dann werden die auch vorgezogen

Wenn man größere Projekte macht, ist, das denke ich sehr relevant

1.2 Aufgabenmanagement und Zeitplanung (Hohe Priorität)

A:

Relevanz: Entscheidendes Feature

Priorität: hoch

Messbarkeit: Zeit muss für Finanzierungsmöglichkeiten auf Tasks eingeteilt werden können, geplante Zeit, benötigte Zeit als Gegenüberstellung, Gantt-Diagramme

Zitate:

Ich würde sagen, das ist eines der höchsten Dinge, weil wenn ich das nicht hätte, würde das für mich als Projektmanager bedeuten, ich müsste das die ganze Zeit abhaken, [...] ich brauche ja die Daten, um sie zu verarbeiten und die sollen von alleine entstehen und sich pflegen.

B:

Relevanz: Vor allem relevant ist die Zeitplanung und das Erkennen von Problemen

Priorität: hoch

Messbarkeit: Gantt-Charts, Unterstützung mehrerer Projekte und geteilten Mitarbeitern, Meilensteine

Zitate:

Ganz wesentlich ist die Ressourcenplanung, man muss einstellen wie viel jemand im Projekt da ist und dann feststellen können, ob sich Zeiträume überschneiden

C:

Relevanz: Abhängig von der Teamgröße steigend, aber einer gewissen Größe unabdingbar

Priorität: mittel bis hoch (steigend mit Teamgröße)

Messbarkeit: Meilensteine, Priorisierung, Deadlines

Zitate:

Das ist für mich eine der sinnvolleren Funktionen [...] in einem größeren Team, indem ist das wichtiger, für ein kleineres Team könnte auch ein wiki ausreichen

Ich denke in kleinen Teams ist es nicht notwendig, es ist cool, wenn die Zeit dafür da ist, aber in kleinen Teams weiß eigentlich eh jeder, was gerade gemacht wird.

Wenn man sich dazu entschließt eine PMS zu benutzen, was ab einer gewissen Teamgröße oder Budget halt einfach nötig ist, dann sollte diese Funktionalität auf jeden Fall verfügbar sein

D:

Relevanz: Zeitplanung ist auch in kleinen Studios wichtig, Aufgabenplanung kann auch anders gelöst werden.

Priorität: mittel steigt mit Teamgröße

Messbarkeit:

Zitate:

Es gibt auch viele Indieteam, die Erfolgreiche sind ohne Aufgabenplanung, die arbeiten eher mit einer Knowledgebase [...] die haben wenig Abhängigkeit zwischen den Arbeitsschritten

Ich denke, wenn man jetzt sagt wir haben hier 10 Waffen und eine dauert so und so lange, das ist schon wichtig, dass man dann abschätzen kann, wie lange man für alle braucht.

Sobald man eine gewisse Größe hat und ein größeres Budget hat [...] müssen termingerecht komplexe Dinge fertig werden [...] ab dann hat man auch zeitliche Verpflichtungen

E:

Relevanz: Unabdingbar bei der Arbeit

Priorität: hoch (hängt auch vom Projekt ab)

Messbarkeit: Meilensteine, Story-Points

Zitate:

Hohe Priorität, alleine schon weil, sobald wenn man mit Stakeholdern arbeitet, es eine Notwendigkeit ist

Da wir an mehreren Projekten arbeiten, müssen wir Woche für Woche entscheiden, was gerade wichtig ist und da ist die Zeitplanung unabdingbar

1.3 Kommunikations- und Kollaborationswerkzeuge (hohe Priorität)

A:

Relevanz: Wäre kein Ausschlusskriterium, wenn es fehlt, ist aber gewünscht

Priorität: mittel bis hoch

Messbarkeit: Ergebnis Dokumentation und Zusammenfassung von Beschlüssen sollten festgehalten werden, direkte Kommunikation eher ausgelagert, ein Tagging-System ist relevant

Zitate:

Ich bin der festen Überzeugung, wenn es in der Software ein Kommunikationstool gibt, dann sollte dort alles gesammelt werden, was beschlossen wurde, ich brauche da nicht die komplette Konversation [...], was wünschenswert wäre, wenn ich die Conclusion im Projektmanagementtool habe.

B:

Relevanz: Ein wichtiges Feature, das auf jeden Fall verfügbar sein sollte

Priorität: hoch

Messbarkeit: Kommunikation in Tasks, mit Kommentaren, mit Datenupload (Bilder, Videos, Mentions), globale Kommunikation über externe Tools

Zitate:

Die Möglichkeit sich an den Tickets austauschen zu können ist essentiell

Direkte Kommunikation über die Projektmanagementsoftware würde ich nicht machen [...] wir machen alles über Discord.

C:

Relevanz: Stellt sicher, dass keine Dinge vergessen werden.

Priorität: hoch

Messbarkeit: Kommentare zu Tasks mit Ergebnissen, direkte Kommunikation extern

Zitate:

Ich denke Tasks kommentieren oder flaggen zu können [...] ist definitiv ein wichtiger Punkt, weil dann nichts untergeht

D:

Relevanz: Kommunikation sollte bei den Aufgaben stattfinden und falls nötig direkt zu neuen Tasks werden.

Priorität: hoch

Messbarkeit: Themen sollten abgeschlossen werden, damit Dinge nicht verloren gehen

Zitate:

[...] Ich denke es ist wichtig, dass man zu einer Aufgabe diskutieren kann und dann auch den Diskussionsverlauf sehen kann [...] das halte ich schon für wichtig da auch den Kontext dabei zu haben.

Ich glaube wichtig ist, dass man das niederschwellig machen kann und das dann auch schnell zuordnen kannst.

E:

Relevanz: Kann praktisch sein, wenn es vorhanden ist, kleinere Teams können das auch über Meetings lösen.

Priorität: mittel, bei größeren Teams eher hoch

Messbarkeit: Beschreibung ist wichtig

Zitate:

Es würde hauptsächlich der Klarheit helfen, um einen Task genauer zu beschreiben. Ich denke schon, das ist wichtig, würde aber vor allem von mir genutzt werden, um Kontext für Aufgaben zu geben.

Kommentare sehe ich eher wichtig bei größeren Teams, wo nicht alle miteinander reden können.

1.4 Gemeinsame Nutzung von Dateien und Dokumentenmanagement (hohe Priorität)

A:

Relevanz: Es macht Sachen leichter und ist praktisch, aber kein Must-have

Priorität: Mittel (Vollständige Datenverwaltung gar nicht)

Messbarkeit: Funktionen zum Hochladen von Zwischenständen oder Bildern ja, aber nicht zur Dateiverwaltung des Projektes

Zitate:

Ich würde mir eine Möglichkeit zur Verlinkung wünschen [...], aber das [Speicherung von Daten im PMS] sollte absolut nicht darinnen sein. [...] das ist extern besser aufgehoben

Eine Möglichkeit zum Hochladen von Zwischenständen, zum Beispiel Bildern wäre wünschenswert [...]

B:

Relevanz: Im Vergleich zu Google Drive ist es kaum möglich zu konkurrieren

Priorität: niedrig bis gar nicht

Messbarkeit: /

Zitate:

Könnte nützlich sein, aber wir nutzen Google Drive und ich würde das Feature nicht nutzen. [...] Am Ende muss man sich fragen, ist dieses Feature genauso stark wie Google Drive[...]

C:

Relevanz: Integration ja, aber kein Ersetzen

Priorität: keine

Messbarkeit: /

Zitate:

Ich finde das einfacher über externe Tools zu arbeiten [...] einen weiteren Kanal zu schaffen, wo es schon Tools gibt, die man ohnehin nutzt, ist nicht nötig

D:

Relevanz: Es ist wichtig, dass dieses Tool existiert, sollte aber kein Teil der Projektmanagementsoftware sein.

Priorität: niedrig

Messbarkeit: /

Zitate:

Man braucht auf jeden Fall eine Versionierungslösung, früher haben wir das über GoogleDrive gemacht, aber das hat nicht mehr funktioniert später

Ob die Daten dann verlinkt werden oder anderweitig referenziert werden ist glaube ich egal, wichtig ist, dass es einfach verfügbar ist und eine Versionierung hat

E:

Relevanz: Eine extra Cloud wird nicht benötigt, es reicht, wenn es verlinkt ist

Priorität: niedrig

Messbarkeit: /

Zitate:

Direkt im Projektmanagementtool würde ich für uns eher als unwichtig ansehen [...] wir haben unser Drive und würden das eher nicht im PMS machen

1.5 Integration mit Entwicklungswerkzeugen (hohe Priorität)

A:

Relevanz: Hohe Relevanz, da es vieles erleichtern kann, wenn es in der PMS existiert

Priorität: Mittel, da es auch andere Tools dafür gibt

Messbarkeit: Weniger direkte Funktionen, aber auf Bugs oder Feedback bezogen hilfreich

Zitate:

Wir haben für Jira ein internes Tool geschrieben, um Bugs zu tracken [...], da hat man dann automatisch ein Ticket mit allen Infos, die die Coder brauchen, um den Bug zu fixen

Ich weiß nicht, ob das extra zur Projektmanagementsoftware dazu gehören muss, dafür gibt es ja externe Tools, aber es wäre natürlich wünschenswert

B:

Relevanz: Kann praktisch sein, aber nicht wirklich nötig

Priorität: niedrig

Messbarkeit: /

Zitate:

So wie wir es jetzt machen schreiben wir die Branches dahin und so viele unterschiedliche gibt es da nicht

C:

Relevanz: Allumfassende Integration nicht möglich, aber eine API über die automatisiert Bugreports übermittelt werden können, ist eine große Hilfe

Priorität: hoch

Messbarkeit: Es sollte erweiterbar sein oder eine API bieten, die man dann ansprechen kann, um automatisierte Tickets oder Bugreports zu erstellen.

Zitate:

Das sehe ich als sehr hohe Prio und ich würde jedem Studio raten davon Nutzen zu machen

Das ist ein sehr wichtiges Tool, wenn man QA inhaus effektiv machen möchte

[...] Das Tracken ist vom Spiel abhängig, bei uns wurde zum Beispiel ein Screenshot und was der Spieler zuletzt gemacht hat und eine große Menge an Daten als Momentaufnahme des Problems [...] in Asana eingetragen.

D:

Relevanz: Kein Must-have, aber wenn integriert, dann für QA eine große Hilfe

Priorität: mittel bis hoch (Bugreporting)

Messbarkeit: API zur Integration von Automatisierung für Bugreports

Zitate:

Wir haben uns eine Lösung geschrieben, bei der man Bugs direkt aus dem Spiel in das Tool integrieren kann [...] und das ist sehr, sehr praktisch. [...] Das wir sofort die Sachen, die wir brauchen, an einem Ort haben, die wir brauchen.

Generell denke ich, dass solche Integrationen schon Sinn machen und einem auch Informationen geben, auf die man sonst keinen Zugriff hätte

Diese Community Integration [...] dass die an der Entwicklung teilnehmen können ist schon sehr wichtig

E:

Relevanz: Der Overhead zum Pflegen der Funktionen könnte größer sein, als der tatsächliche Nutzen

Priorität: niedrig

Messbarkeit: /

Zitate:

Klingt in der Theorie toll und praktisch, aber ich denke nicht, dass es dann genutzt würde und mehr Overhead als Nutzen gibt. [...]

Vielleicht ein System, das automatisch erkennt, wenn ein Asset in der Cloud abgelegt wurde und dann den Task beendet, aber für uns keine Relevanz

B.2 Benutzerfreundlichkeit und Benutzeroberfläche

2.1 Benutzerfreundliche Oberfläche (hohe Priorität)

A:

Relevanz: Sehr relevant wird aber von Producer/PM Seite weniger darauf geachtet

Priorität: hoch

Messbarkeit: Es sollte mit einem Onboarding / Einarbeitung in einer Stunde verständlich sein.

Zitate:

Grundsätzlich ist es da relevanteste, da wird aber weniger drauf geachtet, da gucken wir dann eher von Producer/Projektmanager Seite, ob für uns alles drinnen ist, anstatt für das Team. Das macht uns die Arbeit aber am Ende 200-mal schwerer

Wenn wir ein Tool hätten [...], wo jeder mit klarkommt, ist aber auch ganz schwierig zu sagen, weil du hast ja coder, die logisch und textbasiert arbeiten und dann hast du Animatoren, Künstler [...] die sehr, sehr kreativ und mit chaos arbeiten [..]

B:

Relevanz: Kleine Indieteam haben wenig Zeit für Overhead, eine Bearbeitung sollte also schnell gehen

Priorität: hoch

Messbarkeit: Bearbeitungszeit für einzelne Aufgaben in der PMS

Zitate:

Entscheidend ist, dass die Vorgänge, die man dort bearbeitet, schnell gehen

C:

Relevanz: Eine Software, die alles kann, aber zu viel Einarbeitungszeit benötigt, würde nicht genutzt werden

Priorität: hoch

Messbarkeit: Es sollte eine allgemeine Übersichtsseite geben und vor allem eine klar strukturierte Navigation

Zitate:

Die Software sollte so easy-access wie möglich sein [...] vor allem in Indie Studios ist Projektmanagement ein Luxus würde ich sagen üblicherweise wird das von allen zusammen gemacht [...] das ist Zeitaufwand, den man nicht mit im Budget hat. Deswegen muss es eine Software sein, die super easy to use ist.

Für gewöhnlich bleibt ein Studio bei der Software, die es zuerst genutzt hat

D:

Relevanz: Für ein kreatives Umfeld mit so vielen verschiedenen Personen ist es wichtig, dass alle das Tool gerne verwenden

Priorität: hoch

Messbarkeit: Anpassbarkeit, Einsteigerfreundlichkeit, ansprechende Optik

Zitate:

Du bist ja in einem kreativen Umfeld [...] wo die Motivation auch wichtig ist, dass sich die Leute wohlfühlen [...] und da ist es auch wichtig, dass die Leute das benutzen.

Benutzerfreundlich heißt für mich einsteigerfreundlich zu sein

E:

Relevanz: Wichtiges Kriterium, vor allem muss das UX für die Entwickler gut funktionieren.

Priorität: hoch

Messbarkeit: Kurze Klickstrecken, Anpassungen für die Entwickler

Zitate:

Es ist extrem wichtig, dass es gutes UX hat [...] weil wenn der einzige, der es versteht, der Projektmanager ist, dann ist das nicht gut

Zu gutem UX gehören kurze und einfach verständliche Arbeitsabläufe, bei denen der Entwickler möglichst kurze Klickstrecken hat

2.2 Anpassungsfähigkeit (Mittlere Priorität)

A:

Relevanz: Vor allem in Preisanpassungen wichtig, um an Unterschiede in der Spielentwicklung angepasst zu werden.

Priorität: hoch

Messbarkeit: Anpassungen an unterschiedliche Pipelines und dynamische Abläufe

Zitate:

Wir haben intern eine [PMS-Lösung] entwickelt, weil uns Prozessmanagement gefehlt hat, [...] vor allem was parallele Tasks angeht, Abhängigkeiten, die prozessübergreifend sind [...] da kommt so viel zusammen, dass wir in unserem Tool geschrieben haben, dass man auf Prozesse warten kann

Ich glaube je näher das Tool an der Spieleentwicklung geschrieben wurde, desto näher passt es sich den Abläufen an

B:

Relevanz: Kann relevant sein, sofern das Anpassen keine lange Zeit in Anspruch nimmt

Priorität: mittel bis hoch

Messbarkeit: Anpassungen möglich, aber vorangepasste Templates je nach Aufgabenbereich sollen existieren

Zitate:

Sinn macht es, wenn ich da Dashboard habe, die über alle Projekte hinweg gebraucht werden, denn ich will das nicht für jeden dann über vier Wochen anpassen müssen.

C:

Relevanz: Visual Fluff, aber für Entwickler nicht relevant

Priorität: niedrig

Messbarkeit: /

Zitate:

Ist mir persönlich egal, ich würde aber davon abraten diese Funktionalität möglich zu machen [...] im Indiebereich muss man davon ausgehen, dass einige wenige Personen, wenn überhaupt das Projektmanagement macht. [...] Es ist wichtig, dass die Entwickler sehen, wo ihre Aufgaben sind und abgesehen davon hat der Aufbau denke ich wenig Relevanz

D:

Relevanz: Es sollte Anpassbarkeit geben, aber es sollte auch eine Standardlösung geben

Priorität: niedrig für kleine Studios, größere Studios hoch

Messbarkeit:

Zitate:

Ich glaube das ist schon wichtig, weil es gibt kein perfektes System, und je nach Größe unterscheidet sich das schon [...] deswegen ist es wichtig, dass man es so weit anpassen kann, dass man es an den aktuellen Bedarf anpassen kann

Klar man muss ein bisschen Limitierung auch geben. [...] Kleine Teams haben zum Beispiel vielleicht noch nie Projektmanagement gemacht, [...] die kommen eher von der Richtung, sag mir was ich machen muss

E:

Relevanz: Keine vielleicht bei größeren Teams nötig

Priorität: niedrig (steigt mit Teamgröße)

Messbarkeit: /

Zitate:

Wir haben den Need noch nicht festgestellt, ich glaube, wir sind zu klein, um diesen Need festzustellen, ich kann mir vorstellen, dass das bei größeren Teams aufkommen könnte.

Für kleine Teams ist eigentlich eine Oberfläche die gut funktioniert relevant

2.3 Mobile Zugänglichkeit (niedrige Priorität)

A:

Relevanz: Es sollte mobil abrufbar sein, aber Arbeit findet am PC statt

Priorität: niedrig bis gar nicht

Messbarkeit: Responsive Design

Zitate:

Ne also vor allem in der Spieleentwicklung ist das [Mobile Zugänglichkeit] nicht relevant, weil du sitzt ja eh den ganzen Tag vor dem Rechner

B:

Relevanz: /

Priorität: gar nicht

Messbarkeit: /

Zitate:

Ich mache ja keine bedeutungsvolle Arbeit am Hand

C:

Relevanz: /

Priorität: gar keine

Messbarkeit: /

Zitate:

Aus Entwicklersicht würde ich sagen nein und will ich auch nicht. [...] Als Projektmanager möglicherweise relevant

D:

Relevanz: Eher als Notizblock oder für kurze Updates, weniger zum Arbeiten

Priorität: mittel

Messbarkeit: /

Zitate:

Ich würde sagen mittel relevant, [...] es ist ein häufiger Featurewunsch, den wir haben [...] aber es ist nicht ultra kritisch

E:

Relevanz: Nur für PM oder Gründer interessant

Priorität: niedrig

Messbarkeit: /

Zitate:

Ich würde sagen es ist nicht unpraktisch, aber es ist nicht nötig schon alleine aus dem Grund, dass seine Mitarbeiter außerhalb der Arbeitszeit arbeiten

Als Gründer oder Projektmanager vielleicht ganz praktisch

B.3 Technische Anforderungen

3.1 Cloud-basierte Lösung (hohe Priorität)

A:

Relevanz: Sehr relevant, das System muss extern verfügbar sein

Priorität: hohe Priorität

Messbarkeit: Verfügbarkeit

Zitate:

Also das muss extern verfügbar sein, vor allem dass du, mal ab von outsourcing, auch Leuten heutzutage Homeoffice anbieten kannst, sonst bist du als Arbeitgeber unglaublich unattraktiv

B:

Relevanz: Keine Cloudlösung wäre Ausschlusskriterium

Priorität: hoch

Messbarkeit: Immer verfügbar, Login mit Rechte und Rollensystem

Zitate:

100% Cloud, ich will keine Software selber betreiben

C:

Relevanz: Jedes Programm, das nicht zusätzlich installiert werden muss, ist gut

Priorität: hoch

Messbarkeit: /

Zitate:

Da bin ich absolut auf der Cloudlösung, vor allem wenn man im Homeoffice arbeitet. [...] Jegliche Funktion, die ich direkt im Browser ausführen kann, ist perfekt

D:

Relevanz: Für die meisten Teams, vor allem kleinere, ist das am einfachsten

Priorität: hoch (Größere Teams haben eher Ansprüche an lokales Hosting)

Messbarkeit: /

Zitate:

Ich weiß, dass es einige Teams gibt, die höhere Datenschutzauflagen haben, für die es wichtig ist, dass es eine lokale Lösung gibt. Aber ich würde sagen, für die meisten Teams ist eine Cloudlösung am einfachsten

E:

Relevanz: Je einfacher die Lösung umzusetzen ist, desto besser

Priorität: hoch

Messbarkeit: /

Zitate:

Ich würde sagen cloudbasiert ist schon ganz gut, weil es weniger umständlich ist.

3.2 Skalierbarkeit (Mittlere Priorität)

A:

Relevanz: Weniger relevant, da der Projektscope eine gute Prognose vorgibt

Priorität: niedrig

Messbarkeit: Es sollte möglich sein, Entwickler oder Externe einfach hinzuzufügen und zu entfernen, aber große Skalierungen sind nicht nötig

Zitate:

Ich glaube das ist nicht so wichtig, du benutzt das Tool ja nur im Scope eines Projektes [...], auf Unternehmensführungsebene, weil das sollte vorab bedacht worden sein.

B:

Relevanz: Anpassungen sind nur in kleinem Rahmen erforderlich, aber genau das wird erwartet

Priorität: hoch für kleine Anpassungen

Messbarkeit: /

Zitate:

Ich würde schon erwarten, dass eine Projektmanagementsoftware skalieren kann. Im Projekt hast du vielleicht deine 10-20 Mitarbeiter und da erwarte ich, dass die sie auch abbilden kann

C:

Relevanz: Indiestudios haben keine großen Änderungen, kleine Skalierung ja, größere eher weniger

Priorität: niedrig

Messbarkeit: /

Zitate:

Das ist unterschiedlich und hängt vom Projekt ab. [...] Für gewöhnlich bleiben Indieteams klein, sodass größere Anpassungen nicht nötig sind. [...] Es ist unwahrscheinlich, dass man so einen blow up hat, dass dieses Kriterium wichtig wäre

D:

Relevanz: Die PMS sollte problemlos erweitert werden können, falls sich je nach Entwicklungsphase die Anzahl der Mitarbeiter ändert

Priorität: hoch

Messbarkeit: /

Zitate:

Ich denke schon, dass das wichtig ist [...] man hat schon eine Fluktuation und das Tool sollte damit umgehen können

E:

Relevanz: Teamgrößen können sich verändern, aber für gewöhnlich nur in kleinem Rahmen

Priorität: mittel

Messbarkeit: /

Zitate:

Skalierbarkeit ist grundsätzlich immer wichtig, aber für uns nicht so wichtig.

Teams holen vielleicht auch Freelancer dazu, dass man gegebenenfalls auch zusätzliche Bereiche benötigt[...]

3.3 Datensicherheit und Datenschutz (hohe Priorität)

A:

Relevanz: Backups und Sicherstellen, dass keine Daten verloren gehen ist wichtig, Datenschutz muss sich an die Standards halten

Priorität: n.A.

Messbarkeit: Backups sollten verfügbar sein, am besten auch lokales Hosting möglich

Zitate:

Für mich als Projektmanager ist das nicht so relevant, .. als Chef gibt es da in Deutschland ja auch Standards die man einhalten muss. Für mich selbst ist nur wichtig, dass es Backups gibt

B:

Relevanz: Hohe Relevanz von Backups, sonstiger Datenschutz ist eher unwichtig

Priorität: Backups hoch, sonstige sollte sich an DSGVO halten, aber kein Fokuspunkt

Messbarkeit: Backups sollten gemacht werden

Zitate:

Wenns Richtung Backup und Verfügbarkeit geht [...] das ist eine absolute Bedingung von Cloudsoftware [...] das erwarte ich, dass es das macht.

[Sicherheit] ist relevant, sollte aber nicht in den Workflow eingreifen.

C:

Relevanz: Datenschutz persönlich keine Erfahrung, regelmäßige Backups sollten immer stattfinden.

Priorität: hoch

Messbarkeit: Regelmäßige Backups

Zitate:

Backups sind ein absolut wichtiger Punkt [...] wenn das auf einmal alles weg wäre das schlecht

D:

Relevanz: Backups und grundsätzlicher Datenschutz sollten immer gegeben sein

Priorität: hoch im Sinne der Grundanforderungen

Messbarkeit: Backups sollte es geben

Zitate:

Für uns als Codecks ist das natürlich sehr wichtig. Aber aus Kundensicht denke ich eher unwichtig. [...] Ich denke nicht das [Datenschutz] eine große Rolle da spielt

Backups werden denke ich erwartet, dass der Anbieter sich darum kümmert

Ich denke der Fokus ist da [beim Kunden] nicht so gegeben

E:

Relevanz: Es sollten Standard-Features für Sicherheit und Backups zur Verfügung stehen

Priorität: hoch

Messbarkeit: Zugangskontrolle, optional 2-Faktor-Authentifizierung

Zitate:

Grundsätzlich natürlich wichtig, [...] man möchte sich ja schon wohlfühlen mit den Tools

3.4 Geringe Systemanforderungen (mittlere Priorität)

A:

Relevanz: Wird eigentlich erwartet

Priorität: Bezogen auf Ladezeiten hoch, ansonsten bei Cloud gar nicht

Messbarkeit: Schnelle Ladezeiten

Zitate:

Ladezeiten sollten unglaublich schnell sein

B:

Relevanz: Wird bei Cloud-Lösungen erwartet

Priorität: hoch

Messbarkeit: Je kürzer die Ladezeiten, desto besser

Zitate:

Die Performance sollte ähnlich sein wie bei einer Applikation auf dem Desktop

C:

Relevanz: Ladezeiten müssen kurz sein, da diese sich durchs ganze Team potenzieren

Priorität: hoch

Messbarkeit: Ladezeiten unter einer Sekunde

Zitate:

Das muss absolut schnell sein, das gilt aber für alle Systeme, wenn das nicht in einer Sekunde lädt, ist das frustrierend

D:

Relevanz: Sollte Grundfunktion sein

Priorität: hoch

Messbarkeit: So schnell wie möglich

Zitate:

Im schlimmsten Fall heißt, dass das die Personen [...] ansonsten hören, die Leute auf, das Tool zu benutzen

E:

Relevanz: Die Software muss nicht besonders schnell sein, aber sollte in Webseiten-üblichen Ladezeiten laden

Priorität: hoch

Messbarkeit: /

Zitate:

Ich sage mal, ich würde Branchenübliche Werte erwarten. [...] Ich würde daran dieselben Anforderungen stellen, die ich auch an andere Webseiten stelle

B.4 Kosten und Lizenzierung

4.1 Bezahlbares Preismodell (hohe Priorität)

A:

Relevanz: Relevanz sinkt mit Größe des Studios und dem höheren Bedarf an Professionalisierung von PM

Priorität: Hoch für kleinere Studios, niedriger bei einer gewissen Größe

Messbarkeit: Einmalzahlung gewünscht, aber Abomodell jedoch Standard

Zitate:

Ich glaube als Indiestudio suchst du erst mal nach einer Free Lösung, da sind dir die anderen Sachen relativ egal

Ich glaube, wenn du noch ganz, ganz klein bist, dann kostenlos und so einfach wie möglich, wenn man aber schon länger besteht [...] dann muss das so gut wie möglich für den günstigsten Preis nehmen

B:

Relevanz: Geld bei Indiestudios ist knapp, deswegen kann der Preis ein Ausschlusskriterium sein.

Priorität: hoch

Messbarkeit: So 20% pro User wäre noch vertretbar, wenn der Nutzen der Software groß genug ist.

Zitate:

Das kommt darauf an, welchen Stellenwert [...] die Leistung der Software hat

Was mir persönlich nicht gefällt ist die Skalierung pro User

Ich würde auch eine schlechtere Software in Betracht ziehen, wenn sie günstiger ist.

C:

Relevanz: Kosten sind immer relevant

Priorität: hoch für kleine Teams, sinkt mit Teamgröße

Messbarkeit: / (Keine Einschätzung möglich)

Zitate:

Definitiv einer der wichtigsten Punkte

Wenn ein Indieteam die Größe erreicht hat sich Gedanken über PMS zu machen, überlegt man sich, was ist das beste für das Team

E:

Relevanz: Sehr wichtig, als kleines Team wird man eher ein kostenloses Tool nutzen

Priorität: hoch für kleine Teams

Messbarkeit: Nicht mehr als 2% der monatlichen Ausgaben

Zitate:

Wenn ich von unseren monatlichen Gehältern ausgehe, würde ich sagen nicht mehr als 2% der monatlichen Kosten

Jede Ersparnis, die man als Indie kriegen kann, ist gut

4.2 Flexible Lizenzierung (Mittlere Priorität)

A:

Relevanz: Kaum relevant für kleine Indiestudios, eher durch Professionalisierung

Priorität: für kleine Entwicklerstudios niedrig, für größere eher mittel

Messbarkeit:

Zitate:

Es gibt ja so Enterprise Lösungen, wo man den Entwickler anschreibt und dann eine Lösung angeboten bekommt

B:

Relevanz: Eher bei mehreren Projekten interessant

Priorität: niedrig bis mittel

Messbarkeit: /

Zitate:

Abomodell finde ich ganz in Ordnung [...] was mir gefallen würde, wäre eine Abstufung, denn eine Firma mit 20 Leuten ist anders organisiert als eine mit 100

C:

Relevanz: Am Ende ist die Frage, was günstiger ist, aber das Preismodell muss transparent sein

Priorität: keine

Messbarkeit: /

Zitate:

Das hängt grundsätzlich davon ab, was günstiger ist [...] wichtig ist, dass die Lizenzierung einfach gehalten und verständlich ist

E:

Relevanz: Einmalzahlungen wären am besten, aber monatlich auch in Ordnung.

Priorität: niedrig

Messbarkeit: Preismodell sollte transparent sein, Einmalzahlung natürlich das beste, aber monatlich zumindest besser kalkulierbar

Zitate:

Wenn dann monatlich, weil Leistungsbasiert finde ich immer gruselig [...] und man bekommt dann plötzlich eine 10k Rechnung, dann weiß ich nicht.

Wir würden eine Software, die etwas schlechter ist, dafür aber gratis nutzen, sofern der Funktionsunterschied nicht zu groß ist.

B.5 Zusätzliche Funktionalitäten

5.1 Zeiterfassung (Mittlere Priorität)

A:

Relevanz: Sollte man Schlüsse aus den Daten ziehen oder diese benötigen, helfen die Daten; ansonsten nicht

Priorität: hoch, weniger bei kleinen Teams

Messbarkeit:

Zitate:

Also wenns eine Automatisierung gibt ist cool, aber wenn der Entwickler am Ende [...] seine Acht Stunden selber darauf [die Aufgaben] verbucht, das reicht auch schon.

Man braucht nicht jede Minute auf alles, aber es ist interessant zu wissen, wie viel Zeit man in Meetings verbringt

Es ist immer die Frage, ob man etwas mit den Daten dann macht, wenn die zur Auswertung oder für Förderung genutzt werden ist gut, ansonsten ist es nur zusätzlicher Aufwand

B:

Relevanz: Grobe Zeiterfassung wird auch von extern gefordert, Eintragung durch Mitarbeiter ausreichend

Priorität: Je nach Projekt mittel bis hoch

Messbarkeit: Grobe Zeiterfassung, im Projekt ist wichtig; granular pro Task ist nicht nötig

Zitate:

Manchmal hat man auch so Verbindlichkeiten nach außen [...] wenn ich mehrere Projekte habe, muss ich sagen können, wie viel ein Mitarbeiter in einem Projekt gearbeitet hat

C:*Relevanz:* Abhängig von Projekt und Größe des Teams*Priorität:* mittel*Messbarkeit:* /*Zitate:*

Ich denke, dass Zeiterfassung schon wichtig ist, um eine Ahnung zu haben, wie lange etwas dauert, aber es sollte nicht im PMS hinterlegt werden, weil das dann schon eine Erwartungshaltung schafft und selten zwei Tasks direkt miteinander verglichen werden können

E:*Relevanz:* Es sollte eine Art geben, eine Prognose anzugeben und diese zu kontrollieren für spätere Einschätzungen*Priorität:* mittel*Messbarkeit:* Eine präzises Zeittracking ist eher nicht erforderlich, aber für Schätzungen relevant*Zitate:*

Ich würde sagen in Sachen Story-Points schon irgendwo zwischen should und Must-have.

Ich würde die Zeit nicht akkurat tracken, sondern über story points [...] wenn ich stark daneben liege, würde ich das dann für zukünftige Schätzungen anpassen

5.2 Budgetierung und finanzielle Kontrolle (mittlere Priorität)

A:*Relevanz:* Keine Kaufrelevanz, aber nice to have*Priorität:* Mittel*Messbarkeit:* Tabellarische Übersicht mit Export für Excel oder ähnliches*Zitate:*

Ich glaube da würde ich mir wünschen, dass das alles in einem Tool ist, da das dann weniger Fehleranfällig ist[...]

budgettechnisch haben wir auch hauptsächlich Excel Listen

B:*Relevanz:* Das Projektmanagement-Tool muss sich hier mit einer Tabellensoftware vergleichen*Priorität:* Mittel*Messbarkeit:* /*Zitate:*

Dafür brauche ich nicht unbedingt das Projektmanagement Tool, [...] Die Berechnung von Projektkosten werde ich nicht im Projektmanagement Tool machen

Ich würde das lieber in einer Tabellenkalkulation machen

C:

Relevanz: Da man je nach Finanzierung andere Voraussetzungen braucht, ist eine Tabellenkalkulation einfacher

Priorität: niedrig

Messbarkeit: /

Zitate:

Würde ich sowieso außerhalb machen, weil man da unterschiedliche Informationen braucht je nachdem wer die haben will, also der Publisher oder Investoren. Und die sollten da auch Zugriff haben. Das würde man eh noch auf so etwas wie Excel transkribieren

E:

Relevanz: Wenn dann in Form einer Tabellenkalkulation, aber das ist Excel dann einfacher verfügbar

Priorität: niedrig bis gar nicht

Messbarkeit: /

Zitate:

Das mache ich lieber extern, meistens in Excel [...] wenn ich ein neues Projekt berechne benutze ich Excel dafür [...] da brauche ich nichts anderes.

Ich würde, das auch gar nicht im PM-Tool haben wollen, weil es ist viel einfacher das in Excel zu machen [...] das würde sich zu stressig anfühlen.

5.3 Risikomanagement-Tools (niedrige Priorität)

A:

Relevanz: Weniger formale Prüfungen als Überwachung zwischen Estimate und tatsächlicher Zeit von Tasks und Warnungen bei Verzögerungen

Priorität: Mittel

Messbarkeit:

Zitate:

Eine Risikoliste ist immer gut, aber die muss nicht Teil der PMS sein. Was ich mir wünschen würde, wäre so ein due Date Reminder [...] oder wenn es ein estimate für die Zeit gibt [...] und das wird überlaufen, dass es da eine Warnung gibt

Oder dass ich Sachen taggen kann als "Vorsicht, das ist eine Risk", sodass der Entwickler das auch sehen kann. Wenn das automatisiert läuft, ist das mega cool

B:

Relevanz: Wird über Meetings oder Miro gelöst, Überblick über die Aufgaben einfach möglich

Priorität: keine

Messbarkeit: /

Zitate:

Was es schon geben muss ist eine Priorisierung der Tickets, [...] aber das [Risikomanagement] ist schon ein informeller Prozess aus meiner Sicht.

C:

Relevanz: Übersprungen, da wenig Erfahrung als Entwickler

Priorität: /

Messbarkeit: /

Zitate: /

E:

Relevanz: Wichtig ist, dass der Overhead nicht zu groß wird.

Priorität: Mittel bis niedrig

Messbarkeit: Kenndaten, die menschlich ausgelesen werden und interpretiert werden

Zitate:

Wir haben das bisher immer tabellarisch gemacht [...] das wäre vielleicht interessant, wenn man Risiken zuordnen und dann einen Puffer dafür geben könnte

5.4 Dokumentation und Analyse (mittlere Priorität)

A:

Relevanz: Für einen Projektmanager sehr relevant, für Teams ohne Projektmanager eher weniger

Priorität: hoch für größere Teams mit PM, für kleinere Teams weniger relevant

Messbarkeit: Burndown, Ressourcen-Matrizen, CSV-Export, Auslastungen, Überbelastung von Mitarbeitern

Zitate:

Alles, was du an Auswertung im Tool machen kannst, ist gut, [...] weil das ist immer ein großer Aufwand das extern zu machen.

B:

Relevanz: Selber noch nicht verwendet

Priorität: /

Messbarkeit: Ressourcenplanung und Prognosen/tatsächliche Zeit Vergleich

Zitate:

Das kann auch nützlich sein, um zu verstehen, wie Prognosen dann in der Wirklichkeit umgesetzt wurden und für andere Projekte anpassen.

Für Dinge wie Ressourcenplanung ist das schon wichtig, aber es ist nicht gleich wichtig für alle Aspekte

C:

Relevanz: Wird in kleinen Teams durch mündlichen Austausch erledigt

Priorität: niedrig

Messbarkeit: /

Zitate:

Kann vielleicht sinnvoll sein, aber da sollte [...] ein mündlicher Austausch in der Hinsicht stattfinden, da weiß ich nicht, wie viel Sinn das da noch machen würde

E:

Relevanz: Bestimmte Diagramme sind nötig, die Wichtigkeit hängt von Größe und Art des Projekts ab

Priorität: mittel

Messbarkeit: Eigene Diagramme wären interessant zur Gegenüberstellung von Daten

Zitate:

Aktuell nutzen wir das nicht wirklich, wenn dann nur Burn-Down-Charts um zu gucken, wie sich die Story-Points entwickelt haben und mehr brauchen wir auch gar nicht.

5.5 Integration mit Marketingplattformen (niedrige Priorität)

A:

Relevanz: Kaum Relevanz, kann für kleine Teams ein nice to have sein

Priorität: niedrig

Messbarkeit: Bugtracker ein nettes Feature, komplette Integration eher gefährlich

Zitates:

Finde ich sogar fast gefährlich, weil was, wenn da etwas drinnen steht, dass man nicht posten wollte und der postet dann die Daten

Ich finde so etwas wie das [Bugtracker, über Discord] cool, für kleine Teams ohne QA-Team, aber absolut nicht benötigt

B:

Relevanz: Verwendet externe Tools dafür, sieht das nicht im Aufgabenbereich einer PMS

Priorität: gar keine

Messbarkeit: /

Zitate:

Wenn eine Software kommt, die sagt, dass sie alles kann [...] dann würde ich nicht davon ausgehen, dass eine Software nichts davon gut kann [...] weil es unrealistisch ist

C:

Relevanz: Keine Aussage, da keine Erfahrung

Priorität:

Messbarkeit:

Zitate:

Transparenz in der Entwicklung ist relevant, aber das sollte nicht über das PMS passieren. Da ist ein gigantischer Risikofaktor, dass Dinge geteilt werden, die nicht geteilt werden sollten

E:

Relevanz: Kein entscheidender Faktor

Priorität: niedrig

Messbarkeit: Ticketsystem kann ein nice to have sein.

Zitate:

Ich würde sagen ein Ticketsystem vielleicht, aber alles andere nicht

B.6 Unterstützung und Schulung

6.1 Kundensupport (hohe Priorität)

A:

Relevanz: Kleinere Studios wäre das weniger relevant, abhängig davon welche Verpflichtungen und Termine eingehalten werden müssen

Priorität: weniger wichtig für kleine Teams, wichtiger je größer das Team

Messbarkeit: Der Support sollte am selben Tag noch antworten

Zitate:

Wenn ich jetzt als kleines Team ein kostenloses Tool ohne Support habe, würde ich da drauf springen, [...] wenn ich jetzt aber noch den Publisher dahinter sitzen habe [...] und jetzt komme ich nicht an mein Zeug dran, dann ist es sehr wichtig

B:

Relevanz: Wenn es ein Problem beim PMS gibt, dann muss das schnellstmöglich behoben werden.

Priorität: hoch

Messbarkeit: Substantielle Antworten innerhalb von einem Tag

Zitate:

Da ist die Geschwindigkeit auch wichtig, [...] unterhalb der Woche sollten die schon am selben Tag noch antworten und das Problem auch in der Zeit lösen

C:

Relevanz: Grunderwartung an Kundensupport

Priorität: hoch

Messbarkeit: Antworten am selben Tag, technische Versiertheit bei Fragen zu API

Zitate:

Jeder Kundensupport sollte in der Lage sein, am selben Tag zu antworten

Gerade, wenn man eine API zur Integration anbietet, ist das wichtig, dass der Support da schnell antwortet

E:

Relevanz: Wird vorausgesetzt

Priorität: hoch

Messbarkeit: Der Kundensupport sollte innerhalb eines Tages antworten und auch Techniker schnell greifbar sein.

Zitate:

Wäre schon gut, wenn der innerhalb eines Tages antwortet

Wichtig wäre auch, dass man nicht nur T1 Support hat [...], sondern auch schnell an jemanden mit technischer Expertise weiter geleitet wird ist wichtig

6.2 Dokumentation und Tutorials (Mittlere Priorität)

A:

Relevanz: Eine gute Doku spart Ressourcen, da weniger externe Hilfe für Onboarding genutzt wird.

Priorität: hoch

Messbarkeit: Dokumentation, auch ein Klick-Tutorial könnte hilfreich sein für Onboarding

Zitate:

Also es sollte auf jeden fall eine Doku geben, so groß und detailliert wie möglich, [...] Videos aber nur zur Entscheidung welche Software benutzt wird

Man sollte nur anhand der Demo ohne das Programm gesehen zu haben verstehen können, wie es funktioniert

B:

Relevanz: Video- oder Klicktutorials sind nicht nötig, eine ausführliche Wissensdatenbank aber wichtig

Priorität: mittel

Messbarkeit: Geschriebene Tutorials

Zitate:

Für mich funktioniert etwas gut, wo ich in meinem Tempo durchgehen kann [...] ich hätte gerne geschriebene Tutorials, [...] wo bestimmte Fragen beantwortet werden

C:

Relevanz: Auf jeden Fall wichtig, vor allem für kleine Teams

Priorität: hoch

Messbarkeit: Einfach zu durchsuchen, verringern der Einstiegshürden

Zitate:

Eine umfangreiche Dokumentation ist wichtig, in welchem Format ist mir eigentlich egal, sofern ich schnell das finde, was ich suche

Wenn die Einstiegshürde zu groß ist, würde keiner diese Software nutzen

E:

Relevanz: Grundsätzliche Dokumentation ist gut, für komplexere Dinge können auch Videotutorials praktisch sein

Priorität: hoch

Messbarkeit: Dokumentation für Schnittstellen und Nutzungstutorials

Zitate:

Auf der einen Seite würde ich mir eine gute Dokumentation oder im Zweifelsfall, falls man eingreifen will, auch des Codes dahinter wünschen für die Schnittstellen

Dann aber auch eine gute Nutzungsdokumentation, die durchsuchbar ist

B.7 Kulturelle und teambezogene Überlegungen

7.1 Sprachliche Unterstützung (niedrige Priorität)

A:

Relevanz: Solange EN verfügbar ist, sollte das reichen

Priorität: niedrig (sofern EN verfügbar)

Messbarkeit: /

Zitate:

Es gibt ja meistens eine Firmensprache, die ist bei uns englisch [...] ich denke es reicht, wenn die Tools auf Englisch sind

B:

Relevanz: /

Priorität: keine (sofern EN verfügbar)

Messbarkeit: /

Zitate:

Englisch würde hier ausreichen

C:

Relevanz: /

Priorität: keine (sofern EN verfügbar)

Messbarkeit: /

Zitate:

Englisch reicht aus

E:

Relevanz: /

Priorität: niedrig (sofern EN verfügbar)

Messbarkeit: /

Zitate:

Für uns würde englisch reichen, [...] eigentlich sollte EN in der Spieleindustrie ausreichen

7.2 Anpassung an den Arbeitsablauf des Teams (hohe Priorität)

A:

Relevanz: Es sollte gewisse Arbeitsabläufe enforzen, aber auch Anpassungen erlauben

Priorität: mittel bis hoch (Team abhängig)

Messbarkeit: Sehr Team abhängig, Balance zwischen festen Abläufen und Dynamik des Gamedevelopments

Zitate:

Es ist cool wenns das kann, aber wenn man von keiner Planung umstellt, [...] ich glaube, man kommt nicht drumherum, dass das ein wenig disruptiv ist, aber das ist auch nicht schlimm

B:

Relevanz: Problem ist zu spezifisch und vom Projekt und Team abhängig

Priorität: /

Messbarkeit: /

Zitate:

Manchmal macht es auch Sinn Arbeitsabläufe anzupassen, wenn das Projektmanagement da sinnvollere Lösungen hat, aber es muss vor allem auch bei Spielen Anpassungen geben, falls das vom Projektmanagement nicht passt

C:

Relevanz: Bestehende Abläufe, die funktionieren, sollten beibehalten werden

Priorität: hoch

Messbarkeit: /

Zitate:

Man sollte immer das Projekt an die Arbeitsabläufe anpassen. [...]

E:

Relevanz: Sowohl Anpassbarkeit als auch Leitlinien können nützlich sein

Priorität: mittel

Messbarkeit: Grundsätzliche Bedienbarkeit sollte gegeben sein und Anpassungen machbar, falls wichtig

Zitate:

Ich würde sagen, die grundsätzliche Benutzbarkeit sollte gut genug sein, dass man sich nicht anpassen muss, aber auch eigene Abläufe wie zum Beispiel ein Kanban Board einfügen kann

B.8 Vereinfachung und Minimalismus

8.1 Vermeidung einer Überfrachtung mit Funktionen (hohe Priorität)

A:

Relevanz: Je weniger Überfrachtung, desto besser

Priorität: mittel

Messbarkeit: Projektkontext, Anpassungen sollten eher nur vom PM vorgenommen werden

Zitate:

Ich glaube das Entfernen von Funktionen sollte nicht jeder selber machen, sondern der jeweilige Projektmanager

Es sollte eine Möglichkeit geben, Views zu vereinfachen [...]

Cool wäre es, wenn ich da pro Department oder Entwickler deinen einen One-Pager zusammenstellen könnte, der die für den wichtigen Informationen enthält

B:

Relevanz: Datenklarheit steht im Vorsprung, solange eine Basis an Vereinfachung existiert

Priorität: niedrig bis mittel

Messbarkeit: Anpassungen an Dashboard und Klarheit der Daten

Zitate:

Das ist nicht so wichtig, im Optimalfall sollte die Software nicht 10 Features haben, die irrelevant sind. [...] Wichtiger ist das, wenn ich Daten brauche, nicht auf 4 verschiedene Seiten gucken muss, um mir das zusammenzuspielen

C:

Relevanz: Kleine Teams müssen schnell einen Zugang zur PMS finden

Priorität: hoch

Messbarkeit: Zu den allgemeinen Arbeitsdingen sollte man einfach kommen

Zitate:

Ich würde darauf beharren, dass der Easy-Access und Easy-to-use punkt einer der wichtigsten ist, weil man in einem kleinen Team schnell da rein finden muss. Und da gehört das [Minimalismus] auch dazu

E:

Relevanz: Eine Überfrachtung kann zu Problemen oder Nicht-Benutzung führen.

Priorität: mittel bis hohe Relevanz

Messbarkeit: Möglichkeiten zur Anpassung geben, Unwichtiges verstecken, Wichtiges hervorheben

Zitate:

[...] Das dritte Untermenü ist dann irgendwann zu viel [...] ich finde es ganz gut, wenn man Sachen ausmachen kann, die man nicht braucht

B.9 Motivation und Teammoral

9.1 Transparenz und Nachverfolgung (mittlere Priorität)

A:

Relevanz: Die PMS kann Hinweise liefern, aber Bewertung sollte im Projektkontext eher vom PM

kommen

Priorität: Mittel

Messbarkeit: Es sollte der Projektkontext, der Stand im Projekt sichtbar sein.

Zitate:

Ich glaube, das wichtigste ist, das im Projektkontext zu sehen, dass du siehst, ich arbeite hier an einem Projekt, an einem Produkt, was am Ende fertig ist.

Du sitzt jetzt eine Woche lang an einem Bug und der ist extrem wichtig, in der Zeit macht jemand anderes hundert Bugs, die nur klein waren, da kann man dann ja nicht sehen, wie viel Umfang das jeweils hatte

B:

Relevanz: Unterstützung in dem Bereich möglich, aber in kleinen Teams auch durch Meetings gegeben

Priorität: niedrig

Messbarkeit: Kann unterstützen über Metriken, aber zwischenmenschlich oder durch Meetings ist besser geeignet

Zitate:

Prinzipiell kann die Software da unterstützen, [...] aber sie kann das nicht wirklich leisten, da zeigt man sich seine Arbeit oder redet drüber [...] da sehe ich das Projektmanagement nicht in der Pflicht

Die Leute wollen sehen, dass das Projekt und das Spiel vorangeht und nicht irgendwelche Balken

C:

Relevanz: Keine Relevanz für Entwickler, eher für Projektmanager

Priorität: niedrig bis keine

Messbarkeit: /

Zitate:

Ich denke, dass es für einen Entwickler weder motivierend noch ausschlaggebend ist. [...] Vor allem in Indie Studios sind immer alle auf dem gleichen Stand, sodass so etwas in persönlichen Gesprächen gemacht wird

E:

Relevanz: Kleine Feedback-Einschübe können helfen, ersetzen aber persönliche Interaktion nicht

Priorität: hoch

Messbarkeit: Übersicht über Projekt, kleine Bestärkungen

Zitate:

Ich denke alles, was gute Leistungen betont und Feedback gibt, ist gut [...] es ist wichtig zu sehen, wo man im Projekt ist

9.2 Anerkennungs- und Feedback-Mechanismen (niedrige Priorität)

A:

Relevanz: Kann ein nice to have sein, persönliches Feedback von PM aber nicht ersetzen

Priorität: niedrig

Messbarkeit: /

Zitate:

Es kann witzig sein, [...] aber ich glaube, das kann Leute motivieren, aber man muss dafür sorgen, dass die Motivation auch langfristig bleibt

Man sollte das eher im sozialen Kontext nach Person von PM direkt machen.

B:

Relevanz: Würde nicht genutzt werden

Priorität: niedrig bis keine

Messbarkeit: /

Zitate:

Das ist wichtig, gehört für mich aber nicht ins Projektmanagement. Das wird dann in der Kommunikationsplattform wie Discord gemacht.

C:

Relevanz: Aufgrund der schlechten Vergleichbarkeit sollte das nur persönlich von einem Menschen beurteilt werden

Priorität: niedrig bis keine

Messbarkeit: /

Zitate:

Ich finde das nicht gut das zu haben [...] Die Messbarkeit sollte fürs PM einsehbar sein, aber es sollte nicht an Belohnung oder Bestrafung geknüpft sein. [...] Denn nicht jeder Task hat den gleichen Aufwand [...] Für mich hat es einen demotivierenden Einfluss

E:

Relevanz: Bestärken sollte persönlich gemacht werden, aber das Tool kann helfen dem PM gute Leistung zu erkennen

Priorität: niedrig

Messbarkeit: Unterstützung durch Kennzahlen

Zitate:

Ich denke das beste Feedback gibt sich persönlich und nicht digital [...] Es sollte aber dabei unterstützen gute Leistungen zu erkennen

B.10 Allgemeine Zitate

A:

Ich glaube das wichtigste ist, [...] benutzt es dein Team

B:

Viele der kleinen Studios haben ja keinen Projektmanager, also ist es wichtig, dass die Abläufe schnell sind [...] und die Vorgänge, die man da bearbeitet schnell gehen

Tool Integration wäre wichtig, wir haben unser Projektmanagement Software nach der Miro Integration ausgewählt

C:

Wir haben extra einen Praktikanten gehabt für Projektmanagement in dem kleinen Team wäre es auch bis zum ende ohne PMS möglich gewesen, [...] als wir mit 10 Leuten ins 2. Projekt gegangen sind [...] war es alleine aufgrund der Komplexität nötig mit Tickets und Tasks zu arbeiten [...] denn nur so war es möglich neue Leute einzuarbeiten [...] und da haben wir festgestellt, dass es auch gar nicht ohne geht.

Es würde sicherlich helfen, wenn sich die Software mehr an den Bedürfnissen der Entwickler orientieren würde. [...] Je mehr sich ein PMS in einer Nische bewegt, desto besser wird es für diese

Ich denke, mit der PMS, mit der man anfängt, bei der bleibt man auch.

D:

Ich denke, der Bedarf an Vorlagen [...] wäre auch noch ein Aspekt. Für Teams die noch nicht mit PMS gearbeitet haben zum einfacheren einstieg

E:

Ich glaube ab 10 Leute benötigt man eine professionelle und bezahlte PMS Lösung

Wenn deine PMS es schafft dafür zu sorgen, dass dein PM am Ende nicht den Artist verprügelt, sondern der die von alleine ausgefüllt, dann hat die Software schon viel erreicht

Literaturverzeichnis

- [AAB21] Renae Aurisch, Mohiuddin Ahmed und Abu Barkat. „An outlook at Agile methodologies for the independent games developer“. In: *International Journal of Computers and Applications* 43.8 (2021), S. 812–818. ISSN: 1206-212X. DOI: [10.1080/1206212X.2019.1621463](https://doi.org/10.1080/1206212X.2019.1621463).
- [Ala14] Alan J. W. Thorn. *Practical Game Development with Unity and Blender*. 2014. URL: <https://api.semanticscholar.org/CorpusID:58491222>.
- [Asa25a] Asana. *Asana Budget Planung Grafik*. 2025. URL: <https://asana.com/de/resources/project-budget>.
- [Asa25b] Asana. *Asana Dashboard Grafik*. 2025. URL: <https://help.asana.com/s/article/project-dashboards>.
- [Asa25c] Asana. *Asana Feedback Grafik*. 2025. URL: <https://asana.com/de/uses/team-collaboration>.
- [Asa25d] Asana. *Asana Kommunikations Grafik*. 2025. URL: <https://asana.com/de/uses/team-collaboration>.
- [Atl24a] Atlassian. *Jira Gantt Grafik*. 2024. URL: <https://community.atlassian.com/t5/App-Central-articles/Project-Portfolio-Calendar-for-Jira-TeamBoard-ProScheduler-s-New/ba-p/2596598>.
- [Atl24b] Atlassian. *Jira Kalender Grafik*. 2024. URL: <https://community.atlassian.com/t5/App-Central-articles/Project-Portfolio-Calendar-for-Jira-TeamBoard-ProScheduler-s-New/ba-p/2596598>.
- [BAM08] Abdullah Saeed Bani Ali, Frank T. Anbari und William H. Money. „Impact of Organizational and Project Factors on Acceptance and Usage of Project Management Software and Perceived Project Success“. In: *Project Management Journal* 39.2 (2008), S. 5–33. ISSN: 8756-9728. DOI: [10.1002/pmj.20041](https://doi.org/10.1002/pmj.20041).
- [Bar+22] Inês Barbedo u. a., Hrsg. *Videogame Sciences and Arts*. Communications in Computer and Information Science. Cham: Springer International Publishing, 2022. ISBN: 978-3-030-95304-1. DOI: [10.1007/978-3-030-95305-8](https://doi.org/10.1007/978-3-030-95305-8).
- [Bas07] Basil Mohammed. *Risk and Stakeholder Management in Mega Projects Beyond the Realms of Theory*. 2007. URL: <https://api.semanticscholar.org/CorpusID:196402914>.
- [Bor+20] Markus Borg u. a. „Video Game Development in a Rush: A Survey of the Global Game Jam Participants“. In: *IEEE Transactions on Games* 12.3 (2020), S. 246–259. ISSN: 2475-1502. DOI: [10.1109/TG.2019.2910248](https://doi.org/10.1109/TG.2019.2910248).
- [BR23] Emil BUKŁAHA und Joanna RZEMPAŁA. „PMS TOOLS FOR AGILE PROJECTS“. In: *Scientific Papers of Silesian University of Technology. Organization and Management Series* 2023.185 (2023), S. 39–49. ISSN: 16413466. DOI: [10.29119/1641-3466.2023.185.3](https://doi.org/10.29119/1641-3466.2023.185.3).
- [Cha13] Chantana Chantrapornchai and D. Sawangkokrouk and T. Leadprathom. *Project Management Software : Allocation and Scheduling Aspects*. 2013. URL: <https://api.semanticscholar.org/CorpusID:40476387>.

- [Cli10] Clinton Keith. *Agile Game Development with Scrum*. 2010. URL: <https://api.semanticscholar.org/CorpusID:107721327>.
- [Cod25a] Codecks. *Codecks Burndown Chart*. 2025. URL: <https://www.codecks.io/features/metrics-and-reports/>.
- [Cod25b] Codecks. *Codecks Handansicht Grafik*. 2025. URL: <https://www.codecks.io/features/unique-hand-design/>.
- [CS07] Patrick Cohendet und Laurent Simon. „Playing across the playground: paradoxes of knowledge creation in the videogame firm“. In: *Journal of Organizational Behavior* 28.5 (2007), S. 587–605. ISSN: 0894-3796. DOI: [10.1002/job.460](https://doi.org/10.1002/job.460).
- [Des20] Wendy Despain, Hrsg. *Professional techniques for video game writing*. Second edition. Boca Raton, FL: CRC Press, 2020. ISBN: 9780429196539. URL: <https://www.taylorfrancis.com/books/9780429196539>.
- [EW13] Benjamin Engelsttter und Michael R. Ward. „Strategic Timing of Entry: Evidence from Video Games“. In: *SSRN Electronic Journal* (2013). DOI: [10.2139/ssrn.2379471](https://doi.org/10.2139/ssrn.2379471).
- [FA24] Minhaz-U-Salakeen Fahme und Afia Adiba. „Building Your Dream Team: How Indie Teams Can Form, and Thrive Together“. In: *F1000Research* 13 (2024), S. 42. DOI: [10.12688/f1000research.139274.1](https://doi.org/10.12688/f1000research.139274.1).
- [Fly14] Bent Flyvbjerg. „What you Should Know about Megaprojects and Why: An Overview“. In: *Project Management Journal* 45.2 (2014), S. 6–19. ISSN: 8756-9728. DOI: [10.1002/pmj.21409](https://doi.org/10.1002/pmj.21409).
- [Fon+17] Fagno Fonseca u. a. „Barriers in Information Technology Project Management“. In: *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*. Hrsg. von IEEE. IEEE, 2017, S. 370–375. ISBN: 978-1-5090-1114-8. DOI: [10.1109/TEMSCON.2017.7998403](https://doi.org/10.1109/TEMSCON.2017.7998403).
- [FT14] Maria Ester Ferreira und Anabela Pereira Tereso. „Software Tools for Project Management – Focus on Collaborative Management“. In: *New Perspectives in Information Systems and Technologies, Volume 2*. Hrsg. von Álvaro Rocha u. a. Bd. 276. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2014, S. 73–84. ISBN: 978-3-319-05947-1. DOI: [10.1007/978-3-319-05948-8_8](https://doi.org/10.1007/978-3-319-05948-8_8).
- [Hig10] Jim Highsmith. *Agile project management: Creating innovative products*. 2. ed. The agile software development series. Upper Saddle River, NJ: Addison Wesley, 2010. ISBN: 978-0321658395.
- [HS20] David Hillson und Peter Simon. *Practical project risk management: The atom methodology*. Third edition. Oakland, CA: Berrett-Koehler Publishers, 2020. ISBN: 978-1523089208.
- [HW18] Dean Hammer und Aaron Wildavsky. „The Open-Ended, Semistructured Interview: An (Almost) Operational Guide“. In: *Craftways*. Hrsg. von Aaron Wildavsky. Routledge, 2018, S. 57–101. ISBN: 9780203794517. DOI: [10.4324/9780203794517-5](https://doi.org/10.4324/9780203794517-5).
- [IGD25] IGDA. *What is an indie* ? 2025. URL: <https://igda.org/sigs/indie/>.
- [Int24] International Organization for Standardization. *Systems and software engineering — Lifecycle profiles for very small entities (VSEs)*. 2024.

- [Iri01] Dawn E. Irish. „Putting the horse before the cart“. In: *Proceedings of the 29th annual ACM SIGUCCS conference on User services - SIGUCCS '01*. Hrsg. von Johanna Colgrove, Marv Dunn und Kelly Wainwright. New York, New York, USA: ACM Press, 2001, S. 59. ISBN: 1581133820. DOI: [10.1145/500956.500972](https://doi.org/10.1145/500956.500972).
- [Jen23] Jennifer Falco. *Abstract Kanban Board*. 2023.
- [JM15] Robert Joslin und Ralf Müller. „Relationships between a project management methodology and project success in different project governance contexts“. In: *International Journal of Project Management* 33.6 (2015), S. 1377–1392. ISSN: 02637863. DOI: [10.1016/j.ijproman.2015.03.005](https://doi.org/10.1016/j.ijproman.2015.03.005).
- [Joh16] John Vanderhoef. *An Industry of Indies: The New Cultural Economy of Digital Game Production*. 2016. URL: <https://api.semanticscholar.org/CorpusID:164768708>.
- [Jus10] Jussi Knuuttila. *Modern Game Architectures*. 2010. URL: <https://api.semanticscholar.org/CorpusID:201079895>.
- [Kat20] Branko Katalinic, Hrsg. *Proceedings of the 31st International DAAAM Symposium 2020*. DAAAM Proceedings. DAAAM International Vienna, 2020. ISBN: 9783902734297. DOI: [10.2507/daaam.proceedings](https://doi.org/10.2507/daaam.proceedings).
- [Ken13] Kent L. Beck, Michael A. Beedle, A. V. Bennekum, Alistair Cockburn, Ward Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, Ron Jeffries, J. Kern, Brian Marick, Robert C. Martin, S. Mellor, K. Schwaber, J. Sutherland, Dave A. Thomas. „Manifesto for Agile Software Development“. 2013. URL: <https://api.semanticscholar.org/CorpusID:109006295>.
- [Ken20] Ken Schwaber & Jeff Sutherland. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. 2020. URL: <https://scrumguides.org/index.html>.
- [Ker13] Harold Kerzner. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. 11. Aufl. s.l.: Wiley, 2013. ISBN: 9781118022276. URL: <http://gbv.eblib.com/patron/FullRecord.aspx?p=1113482>.
- [KFH13] Carine Khalil, Valérie Fernandez und Thomas Houy. „Can Agile Collaboration Practices Enhance Knowledge Creation between Cross-Functional Teams?“ In: *Digital Enterprise Design and Management 2013*. Hrsg. von Pierre-Jean Benghozi, Daniel Krob und Frantz Rowe. Bd. 205. Advances in Intelligent Systems and Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 123–133. ISBN: 978-3-642-37316-9. DOI: [10.1007/978-3-642-37317-6_textunderscore11](https://doi.org/10.1007/978-3-642-37317-6_textunderscore11).
- [KL13] Jussi Koutonen und Mauri Leppänen. „How Are Agile Methods and Practices Deployed in Video Game Development? A Survey into Finnish Game Studios“. In: *Agile Processes in Software Engineering and Extreme Programming*. Hrsg. von Wil van der Aalst u. a. Bd. 149. Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 135–149. ISBN: 978-3-642-38313-7. DOI: [10.1007/978-3-642-38314-4_textunderscore10](https://doi.org/10.1007/978-3-642-38314-4_textunderscore10).
- [KLS13] Jussi Kasurinen, Risto Laine und Kari Smolander. „How Applicable Is ISO/IEC 29110 in Game Software Development?“ In: *Product-Focused Software Process Improvement*. Hrsg. von David Hutchison u. a. Bd. 7983. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 5–19. ISBN: 978-3-642-39258-0. DOI: [10.1007/978-3-642-39259-7_textunderscore4](https://doi.org/10.1007/978-3-642-39259-7_textunderscore4).

- [Kor21] Jessica Kordouni. „Besser im Team“. In: *Sozialwirtschaft* 31.1 (2021), S. 26–27. ISSN: 1613-0707. DOI: [10.5771/1613-0707-2021-1-26](https://doi.org/10.5771/1613-0707-2021-1-26).
- [Lar04] Craig Larman. *Agile and iterative development: A manager's guide*. Software engineering. Boston, Mass.: Addison-Wesley, 2004. ISBN: 978-0131111554.
- [Leg95] Thomas L. Legare. „MINIMIZING RESISTANCE TO TECHNOLOGICAL CHANGE A Power and Politics Approach“. In: *Information Systems Management* 12.4 (1995), S. 59–61. ISSN: 1058-0530. DOI: [10.1080/07399019508963006](https://doi.org/10.1080/07399019508963006).
- [Mar+19] Alexander Marbach u. a. „Optimization of Project Management Processes Using the A* Project Management System (AStarPM)“. In: *HCI International 2019 - Posters*. Hrsg. von Constantine Stephanidis. Bd. 1032. Communications in Computer and Information Science. Cham: Springer International Publishing, 2019, S. 78–85. ISBN: 978-3-030-23521-5. DOI: [10.1007/978-3-030-23522-2_textunderscore11](https://doi.org/10.1007/978-3-030-23522-2_textunderscore11).
- [Mas10] Mark Mason. „Sample Size and Saturation in PhD Studies Using Qualitative Interviews: Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, Vol 11, No 3 (2010): Methods for Qualitative Management Research in the Context of Social Systems Thinking“. In: (2010). DOI: [10.17169/fqs-11.3.1428](https://doi.org/10.17169/fqs-11.3.1428).
- [Mit+16] Hugo A. Mitre-Hernández u. a. „Decreasing Rework in Video Games Development from a Software Engineering Perspective“. In: *Trends and Applications in Software Engineering*. Hrsg. von Jezreel Mejia u. a. Bd. 405. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2016, S. 295–304. ISBN: 978-3-319-26283-3. DOI: [10.1007/978-3-319-26285-7_textunderscore25](https://doi.org/10.1007/978-3-319-26285-7_textunderscore25).
- [MP14] Farzana Asad Mir und Ashly H. Pinnington. „Exploring the value of project management: Linking Project Management Performance and Project Success“. In: *International Journal of Project Management* 32.2 (2014), S. 202–217. ISSN: 02637863. DOI: [10.1016/j.ijproman.2013.05.012](https://doi.org/10.1016/j.ijproman.2013.05.012).
- [MTL99] Concepción Maroto, Pilar Tormos und Antonio Lova. „The Evolution of Software Quality in Project Scheduling“. In: *Project Scheduling*. Hrsg. von Frederick S. Hillier und Jan Węglarz. Bd. 14. International Series in Operations Research & Management Science. Boston, MA: Springer US, 1999, S. 239–259. ISBN: 978-1-4613-7529-6. DOI: [10.1007/978-1-4615-5533-9_textunderscore11](https://doi.org/10.1007/978-1-4615-5533-9_textunderscore11).
- [Mus+10] Juergen Musil u. a. „Improving Video Game Development: Facilitating Heterogeneous Team Collaboration through Flexible Software Processes“. In: *Systems, Software and Services Process Improvement*. Hrsg. von Andreas Riel u. a. Bd. 99. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, S. 83–94. ISBN: 978-3-642-15665-6. DOI: [10.1007/978-3-642-15666-3_textunderscore8](https://doi.org/10.1007/978-3-642-15666-3_textunderscore8).
- [Pag89] G. William Page. „Using Project Management Software in Planning“. In: *Journal of the American Planning Association* 55.4 (1989), S. 494–499. ISSN: 0194-4363. DOI: [10.1080/01944368908975439](https://doi.org/10.1080/01944368908975439).
- [Pau09] Paulsmith99. *Bild: Wasserfall Modell*. 2009. URL: https://commons.wikimedia.org/wiki/File:Waterfall_model_%281%29.svg.
- [Pro20] Project Management Institute. *Ahead of the Curve: Forging a Future-Focused Culture. Pulse of the Profession*. 2020. URL: <https://www.pmi.org/learning/library/forging-future-focused-culture-11908>.

- [Pro21] Project Management Institute. *The standard for project management*. Newtown Square, Pennsylvania: Project Management Institute Inc, 2021. ISBN: 978-1-62825-664-2.
- [Rob04] Rob Keefer. *Agile Software Development*. 2004. URL: <https://api.semanticscholar.org/CorpusID:8321691>.
- [RW13] Rido Ramadan und Yani Widayani. „Game development life cycle guidelines“. In: 2013, S. 95–100. ISBN: 978-1-4799-4692-1. DOI: [10.1109/ICACSSIS.2013.6761558](https://doi.org/10.1109/ICACSSIS.2013.6761558).
- [SFT14] Marc Schmalz, Aimee Finn und Hazel Taylor. „Risk Management in Video Game Development Projects“. In: *2014 47th Hawaii International Conference on System Sciences*. IEEE, 2014, S. 4325–4334. ISBN: 978-1-4799-2504-9. DOI: [10.1109/HICSS.2014.534](https://doi.org/10.1109/HICSS.2014.534).
- [Sil17] Silvester Dian Handy Permana. *E-MARKETING STRATEGY IN GAME INDUSTRY WITH SOCIAL MEDIA USING E-BUSINESS MODEL*. 2017. URL: <https://api.semanticscholar.org/CorpusID:168845995>.
- [SN08] Patrick Stacey und Joe Nandhakumar. „Opening up to agile games development“. In: *Communications of the ACM* 51.12 (2008), S. 143–146. ISSN: 0001-0782. DOI: [10.1145/1409360.1409387](https://doi.org/10.1145/1409360.1409387).
- [sta15] standishgroup. „CHAOS REPORT 2015“. 2015. URL: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf.
- [TH08] James Thomas und Angela Harden. „Methods for the thematic synthesis of qualitative research in systematic reviews“. In: *BMC medical research methodology* 8 (2008), S. 45. DOI: [10.1186/1471-2288-8-45](https://doi.org/10.1186/1471-2288-8-45).
- [TSC05] F. TED TSCHANG. „VIDEOGAMES AS INTERACTIVE EXPERIENTIAL PRODUCTS AND THEIR MANNER OF DEVELOPMENT“. In: *International Journal of Innovation Management* 09.01 (2005), S. 103–131. ISSN: 1363-9196. DOI: [10.1142/S1363919605001198](https://doi.org/10.1142/S1363919605001198).
- [Tur14] J. Rodney Turner. *Handbook of project-based management: Leading strategic change in organizations*. Fourth edition. New York und London: McGraw-Hill, 2014. ISBN: 978-0071821780.
- [WB20] Daniel A. Wren und Arthur G. Bedeian. *The evolution of management thought*. Eighth edition. Hoboken, NJ: John Wiley & Sons Inc, 2020. ISBN: 978-1-119-69285-0.
- [WMM21] Kirti Wanjale, Monika Mangla und Paritosh Marathe. „Security of Sensitive Data in Cloud Computing“. In: *Machine Learning Approach for Cloud Data Analytics in IoT*. Hrsg. von Sachi Nandan Mohanty u. a. Wiley, 2021, S. 99–118. ISBN: 9781119785804. DOI: [10.1002/9781119785873.ch5](https://doi.org/10.1002/9781119785873.ch5).

Eidesstattliche Erklärung

Hiermit versichere ich – Christian Klumpp – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

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

Mittweida, 03. Februar 2025

Ort, Datum



Christian Klumpp