

Freie Universität Berlin

Bachelorarbeit am Institut für Informatik der Freien Universität Berlin

Arbeitsgruppe Software Engineering

Erstellung eines Messplans zur Ermittlung des Erfolgs des NeMo-Prozesses

Ailis Oßwald

Matrikelnummer: 4997184

ailis@zedat.fu-berlin.de

Eingereicht bei: Prof. Dr. Lutz Prechelt

Zweitgutachter: Prof. Dr. Margarita Esponda-Argüero

Berlin, 23. Oktober 2019

Zusammenfassung

Motivation: Softwareprozesse sind ein wichtiger Teil der Softwareentwicklung. Allerdings gibt es kaum Forschung, die ermitteln lässt, wie sehr ein Softwareprozess bestimmten Softwareprozess-Güte-Kriterien entspricht.

Ziel: Das Ziel dieser Bachelorarbeit war es, den Erfolg eines konkreten Entwicklungsprozesses messbar zu machen. Der Erfolg setzt sich zusammen aus Softwareprozess-Güte-Kriterien.

Methoden: Durch Literaturrecherche und qualitative Interviews sind Softwareprozess-Güte-Kriterien ermittelt worden. Um für jedes Softwareprozess-Güte-Kriterium Maße zu ermitteln, wurde die Goal Question Metric Methode verwendet.

Ergebnisse: Durch die Interviews ist zusätzlich ein repräsentatives Meinungsbild für den Neue Modellseiten Softwareprozess (NeMo-Prozess) entstanden. Von den ermittelten Kriterien sind folgende für den Erfolg relevant: Transparenz, Effizienz, Planbarkeit, Softwarequalität, Verbesserungsfähigkeit, Plausibilität und Skalierbarkeit. Diese Kriterien sind alle, bis auf Skalierbarkeit, für den NeMo-Prozess messbar gemacht worden. Testmessungen lieferten Hinweise zur Messungserhebung.

Fazit: Es sind Maße für Kriterien erarbeitet worden, anhand derer der Erfolg des NeMo-Prozesses festgestellt werden kann. Um den Erfolg des NeMo-Prozesses zu ermitteln, müssen die hier erarbeiteten Messungen durchgeführt werden.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

23. Oktober 2019

Ailis Oßwald

Danksagung

Hier möchte ich mich bei all denjenigen bedanken, die mich während meiner Bachelorarbeit unterstützt haben und beratend zur Seite standen.

Zu Beginn möchte ich mich daher bei der "Das Büro am Draht GmbH" und vor allem dem NeMo-Projektmanager-Team bedanken, welches mir das Schreiben dieser Arbeit ermöglicht hat.

Ich bedanke mich bei bei Herrn Professor Doktor Lutz Prechelt, der meine Bachelorarbeit betreut und begutachtet hat, für die Vorschläge an passender Stelle und die hilfreichen Anregungen, mit denen neue Möglichkeiten eröffnet wurden.

In diesem Zusammenhang möchte ich mich auch besonders herzlich bei Nadine Brinkmann bedanken, die mir die ganze Arbeit über beratend und kritisierend zur Seite stand. Ich möchte mich bedanken für das stetige und hilfreiche Feedback mit konstruktiver Kritik und das gelegentliche Lösen meiner Gedankenknoten.

Genauso gilt mein Dank Sora Lee, Janosch Schilling und Randolph Nawrath, die mich mit Nadine Brinkmann als Team tatkräftig unterstützt haben.

Besonderer Dank gilt all denen, die an meinem Interview teilgenommen haben und mir durch ihre Informationsbereitschaft ermöglicht haben, ein besseres Bild des NeMo-Prozesses zu erlangen.

Ebenfalls möchte ich mich bei meinem Kommilitonen Robert Selack herzlich bedanken, der mir die Bachelorarbeit über mit viel Geduld, Interesse und Hilfsbereitschaft zur Seite stand. Dabei möchte ich mich für die Diskussionen und Ideen bedanken, die mich zu immer neuen Erkenntnissen angeregt haben.

Abschließend bedanke ich mich bei Hanna Pfannschmidt, Marek Meyer und Ernst Friedrich Oßwald, die meine Arbeit Korrektur gelesen haben.

Inhaltsverzeichnis

1	Einführung	9
2	Ermittlung von Softwareprozess-Güte-Kriterien	9
2.1	Literaturrecherche	9
2.2	Interviews zum NeMo-Prozess	10
2.2.1	Vorbereitung und Durchführung der Interviews	10
2.2.2	Interviewauswertung: Kriterien	11
2.2.3	Interviewauswertung: Meinungsbild	13
3	Entwicklung von Maßen für Erfolgskriterien des NeMo-Prozesses	14
3.1	Auswahl der für den Erfolg relevanten Softwareprozess-Güte-Kriterien	14
3.2	Herleitung von Maßen mithilfe der Goal Question Metric Methode . . .	16
3.3	Maße der einzelnen Kriterien	17
3.3.1	Skalierbarkeit	17
3.3.2	Transparenz	18
3.3.3	Effizienz	19
3.3.4	Planbarkeit	21
3.3.5	Softwarequalität	22
3.3.6	Verbesserungsfähigkeit	23
3.3.7	Plausibilität	24
4	Übersicht aller Maße im Prozessablauf	25
5	Fazit	28
	Literaturverzeichnis	30
	Anhang	
A	Literaturergebnisse	32
B	Interview Fragen	33
C	Messpläne der Softwareprozess-Güte-Kriterien	34
C.1	Transparenz	34
C.1.1	Beteiligte nach Prozessphasen	35
C.1.2	Liste der notwendigen Informationen je Prozessphase	35
C.2	Effizienz	37
C.3	Planbarkeit	38
C.4	Softwarequalität	39
C.5	Verbesserungsfähigkeit	40
C.6	Plausibilität	41

1 Einführung

Die Idee für meine Arbeit entstand während meiner Werksstudentenzeit in der "Das Büro am Draht GmbH" (DBaD GmbH) im Team "Neue Modellseiten" (NeMo). Die Softwareentwicklungs-Agentur DBaD GmbH, hat sich auf die Entwicklung von Webapplikationen spezialisiert.

Als NeMo-Prozess wird der Softwareentwicklungsprozess (folgend Prozess) dieses Teams bezeichnet. Zwar wird dieser NeMo-Prozess seit einiger Zeit angewendet, es können aber keine Aussagen darüber getroffen werden, wie erfolgreich der Prozess ist. Der Prozess läuft, Sprints werden in der vorgesehenen Zeit abgeschlossen, der Kunde arbeitet im Prozess mit und das Team hat sich an die sich durch den Prozess veränderten Arbeitsweisen gewöhnt. Aber es wurden bisher noch keine fundierten Überlegungen angestellt, was der Prozess verbessert hat. Damit ist es schwierig den Prozess selbst zu verbessern. Man kann nur vermuten, was der Prozess verbessert haben könnte und wie sehr: Ist der NeMo-Prozess effizient? Begünstigt er Transparenz? Ist das NeMo-Team durch ihn eher planbar und organisiert oder flexibel und schnell? - All das sind Fragen, die im Moment nicht fundiert beantwortet werden können.

Hier setzt meine Arbeit an. Ich möchte den NeMo-Prozess so messbar zu machen, dass es möglich ist dessen Erfolg zu beurteilen.

2 Ermittlung von Softwareprozess-Güte-Kriterien

Um eine Aussage über den Erfolg treffen zu können, mache ich die dafür wichtigsten Softwareprozess-Güte-Kriterien messbar (Abschnitt 3.3). Softwareprozess-Güte-Kriterien (kurz Kriterien) sind Eigenschaften, anhand derer Softwareprozesse bewertet werden können. Bevor die Kriterien ausgewählt werden konnten (3.1), sind mögliche Kriterien durch eine Literaturrecherche (2.1), gefolgt von Interviews (2.2.2) ermittelt worden. Trotz umfassender Suche über mehrere Wochen fiel die Literaturrecherche wenig ergiebig aus. Die Interviews wurden in einem Zeitfenster von 2 Wochen mit 15 Personen durchgeführt (2.2.1) und haben, inklusive Vor- und Nachbereitung, insgesamt 4 Wochen beansprucht.

2.1 Literaturrecherche

Auf der Suche nach Kriterien habe ich nach passender Literatur gesucht, auch um generell mehr über Prozess -Evaluationen und -Messungen zu erfahren und eventuelle Ansätze darüber zu finden. Dabei habe ich passende Ansätze gefunden, wie die später verwendete Goal Question Metric Methode (3.2).

Um zum Thema "Erfolgsmessung eines Prozesses" passende Literatur zu finden, habe ich über Google-Scholar mithilfe von Kombinationen und Variationen aus folgenden Schlagwörtern gesucht: "software process success", "validation", "quality" und "metric". Da der NeMo-Prozess, der zu passender Zeit näher erläutert wird (4), stark an Scrum orientiert ist, habe ich auch Scrum in Verbindung mit den zuvor genannten Wörtern als Schlagwort genutzt. Scrum ist ein agiler Softwareentwicklungsprozess in welchem die einzelnen Iterationen Sprint genannt werden [16].

2. Ermittlung von Softwareprozess-Güte-Kriterien

Die Recherche nach Quellen mit dem Schwerpunkt "Erfolgsmessung eines Prozesses" gestaltete sich schwierig. Ich habe überwiegend Artikel gefunden, die sich mit dem Projekterfolg auseinandersetzen [10], [18], [5]. Die meisten dieser Artikel betrachten Prozesse meist nur im Zusammenhang mit dem Projekt. Weitere Artikel diskutieren die Korrektheit theoretischer Prozessmodelle [7]. Jüngere Artikel befassen sich mit den Erfolgsfaktoren, die für die Nutzung von Scrum bedeutend sind [12], [18], [11], [6].

Da ein Großteil der Artikel keine konkreten Kriterien lieferte, habe ich aus Erfolgsfaktoren, Aussagen und Projekt-Erfolgs-Kriterien Prozess-Güte-Kriterien abgeleitet. Ich habe versucht für jeden dieser Begriffe ein Güte-Kriterium zu ermitteln, welches einer Eigenschaft entspricht, die einen Prozess beschreibt oder maßgeblich von diesem beeinflusst werden kann.

Einige Artikel beschäftigen sich mit "Critical Success Factors" [4], [18], [11], [8]. Das Ziel dieser Artikel ist es, die für den Erfolg eines Projektes wichtigsten Faktoren zu finden. Oft sind die dort genannten Faktoren von Prozessen beeinflussbar, wie beispielsweise Planbarkeit [4], [11] und Effizienz [8].

Ein weiterer Artikel beschäftigt sich mit Scrum und dessen Vorteilen. Hier wird beispielsweise eine verbesserte Produktqualität, sowohl in der Nutzbarkeit, als auch in der Zuverlässigkeit genannt [3]. Da dies einer der gewünschten Effekte bei der Einführung von Scrum sei, ist dies eine vom Prozess beeinflussbare Eigenschaft und damit für mich ein Kriterium.

Aus den agilen Prinzipien lassen sich weitere Eigenschaften ableiten [1]. Etwa aus dem 2. Prinzip: "Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen." Hier ist von Flexibilität die Rede: Das Prinzip besagt, dass Veränderungen in den Anforderungen auch spät in der Entwicklung noch beachtet werden sollten.

Die vollständige Liste der durch die Literaturrecherche gefundenen Kriterien ist im Anhang A zu finden.

2.2 Interviews zum NeMo-Prozess

Um zusätzlich zu den in der Literaturrecherche (Abschnitt 2.1) gefundenen Kriterien weitere zu finden, habe ich qualitative Interviews geführt (2.2). Für diese werden Durchführung und Ziele erklärt (2.2.1), die dadurch gefundenen Kriterien erläutert (2.2.2) und schließlich das dabei entstandene Meinungsbild aufgezeichnet (2.2.3).

2.2.1 Vorbereitung und Durchführung der Interviews

Hauptziel der Interviews war es, weitere Kriterien zu ermitteln. Dafür sollten hauptsächlich Prozess-Beteiligte befragt werden. Ich bin davon ausgegangen, dass es vielen Teilnehmern schwer fallen würde direkt Kriterien zu nennen. Daher beziehen sich die Fragen auf den NeMo-Prozess und sind damit greifbarer. Dies eröffnete die Möglichkeit, zusätzlich nach Meinungen zum NeMo-Prozess zu fragen. Allerdings sind Aussagen, die auf einen konkreten Prozess bezogen sind, kleiner gefasst als auf Prozesse allgemein bezogene Aussagen. Daher habe ich zwei Versionen jeder Frage formuliert, eine allgemein gefasste und eine auf den NeMo-Prozess bezogene. Die komplette Liste mit allen Fragen des Interviews ist im Anhang B zu finden.

Jedes Interview begann mit einer thematischen Einführung und mit folgender Frage: "Als wie erfolgreich beurteilst du den NeMo-Prozess?". Durch die für die Antwort stattfindende Bewertung sollte der Einstieg in das Thema beschleunigt werden. Jede Antwort sollte begründet werden, wodurch meist schon erste Kriterien beschrieben wurden. Im Verlauf jedes Interviews habe ich versucht, die NeMo-Fragen als Einstieg zu nutzen und zu den allgemeinen Fragen zu wechseln.

Die Teilnehmer der Interviews habe ich so ausgewählt, dass jede Prozess-Rolle mindestens zweimal vertreten war. Repräsentativ für das NeMo-Projekt war die Mehrheit der Teilnehmer Entwickler.

Die Interviews liefen insgesamt erfolgreich ab, Aussagen wurden meist ohne Aufforderung begründet. Kriterien wurden, anders als erwartet, seltener benannt als beschrieben. In vielen Interviews konnten kaum allgemeine Fragen genutzt werden, weil sich diese Fragen als zu abstrakt für die Befragten herausstellten. Das Interview wurde oft als Feedback-Gespräch zum NeMo-Prozess wahrgenommen, was wahrscheinlich durch die auf den NeMo-Prozess bezogenen Fragen zurückzuführen ist. Dies hatte zur Folge, dass neben der Beschreibung von Kriterien ein umfangreicheres Meinungsbild als erwartet entstand.

2.2.2 Interviewauswertung: Kriterien

Unter Prozess-Güte-Kriterien sind Eigenschaften zu verstehen, die einem Prozess zugeschrieben werden können, und diesen oder durch diesen maßgeblich beeinflusste Arbeitsweisen beschreiben.

Selten wurden Kriterien so konkret wie in folgendem Beispiel genannt: "Der NeMo-Prozess ist komplex und groß, aber man kann ihn trotzdem verstehen, wenn man das System dahinter erkennt." Die hier konkret genannten Kriterien sind **Komplexität** und **Größe**. Häufiger wurden Effekte bestimmter Kriterien beschrieben. Wie auch in vorheriger Aussage, in der **Plausibilität** beschrieben wird. Das geht daraus hervor, dass der Prozess als verständlich bezeichnet wird. Prozess-Beteiligte befolgen den Prozess, weil sie verstehen können, warum die vom Prozess vorgesehenen Arbeitsabläufe sinnvoll sind.

Ähnlich wie ich Plausibilität abgeleitet habe, kann **Planbarkeit** aus folgender Aussage abgeleitet werden: "Zum Beispiel haben sich früher während ich gecoded habe, manchmal die Anforderungen (...) verändert. Durch den jetzigen Prozess passiert so etwas nicht mehr. Ich weiß zu Sprintstart immer, welche Tickets im Sprint erledigt werden sollen, und dass die Anforderungen sich nicht mehr ändern." Der Unterschied zwischen den beiden Prozessen scheint das Vorhandensein von Planungssicherheit in den Anforderungen zu sein. Diese Eigenschaft beschreibt einen Aspekt von Planbarkeit. Planbarkeit beschreibt die Möglichkeit, sich der Realität annähernde Vorhersagen zu treffen, um vorausplanen zu können, aber auch bestimmte Sicherheiten garantieren zu können, in deren Rahmen zum Beispiel Voraussetzungen für Schätzungen nicht weiter verändert werden.

"Für mich ist ein guter Prozess ein Prozess, der (...) die Umsetzung von Tickets beschleunigt." Auch hier wird nicht konkret eine Eigenschaft benannt, sondern eine Auswirkung beschrieben, die dieser zugeordnet wird: **Effizienz**. Effizienz lässt sich so erklären, dass gegebene Ressourcen bestmöglich ausgenutzt werden sollen. Ein

2. Ermittlung von Softwareprozess-Güte-Kriterien

Äquivalent zu Effizienz ist Produktivität.

Eine weitere Art der Aussage ist die, in der Maßnahmen beschrieben werden, die zum Erreichen eines gewissen Ziels durchgeführt werden, das einem bestimmten Kriterium zugeordnet werden kann. "Ein Prozess sollte optimalerweise immer versuchen, sich weiter zu entwickeln, damit er auch langfristig nutzbar ist, indem er sich den sich verändernden Bedürfnissen anpasst". Hier wird beschrieben, warum ein langfristiger Prozess sich weiterentwickeln sollte. Dies beschreibt Eigenschaften, die ein Prozess, dem das Kriterium Verbesserung zugeschrieben wird, erfüllen sollte. **Verbesserungsfähigkeit** beschreibt die Fähigkeit eines Prozesses sich zu verändern, und damit weiter nutzbar zu bleiben. Das Kriterium **Langfristigkeit** wird ebenfalls in diesem Satz erwähnt. Dieses Kriterium beschreibt, worauf ein Prozess ausgelegt ist: Für einen längeren Zeitraum, in dem Anpassungen unumgänglich sind. Gegenätzlich dazu ist ein **kurzfristiger** Prozess, der durch die kurze Einsatzzeit solche Anpassungen vermeiden kann.

Eine weitere der vorherigen Art von Aussagen ähnelnde Aussage ist das Nennen eines bestimmten Teils des Prozesses, wie "Ich finde die Refinements wichtig, weil so jeder weiß, was zu tun ist, weil alle im Team es einmal gesehen haben". Das genannte Meeting ist wichtig, weil Informationen im Team verteilt werden. Dies beschreibt **Transparenz**, diese Eigenschaft lässt sich auf den Prozess beziehen, aber auch auf Informationen für und über den Fortschritt des Projektes. Transparenz bedeutet, unabhängig auf welche Art von Informationen bezogen, dass diese entweder bekannt oder frei verfügbar sind.

Aus den Interviews gingen insgesamt 18 Kriterien hervor, die alle in der Literatur gefundenen Kriterien bestätigen. Davon sind Komplexität, Größe, Plausibilität, Planbarkeit, Effizienz, Verbesserungsfähigkeit, Kurz- und Langfristigkeit und Transparenz bereits beschrieben worden. Weiterhin wurden durch die Interviews folgende Kriterien erschlossen:

- Bei **dokumentationsgetriebener** Entwicklung wird jedes Feature dokumentiert.
- **Flexibilität** als Gegenteil von Planbarkeit; Änderungen können den ganzen Prozess über auch sehr kurzfristig noch angewendet werden.
- In einem **fortbildenden** Prozess haben Prozess-Beteiligte die Möglichkeit ihre Fähigkeiten und ihr Wissen zu erweitern, durch dafür im Prozess integrierte Möglichkeiten.
- **Selbstständigkeit** bezieht sich auf die Arbeitsweisen im Prozess und das Fördern dieser.
- Die **Skalierbarkeit** eines Prozesses beschreibt seine Flexibilität hinsichtlich der Anzahl der Prozess-Beteiligten
- **Stressfreiheit** für alle Beteiligten in dem Sinne, dass der Prozess jeder Prozessrolle ermöglichen sollte, sich nur auf deren Aufgaben fokussieren zu können.
- **Softwarequalität** beschreibt unter anderem die Zuverlässigkeit und Nutzbarkeit des Softwareproduktes, welches das Ziel jedes Softwareprozesses ist.

- Bei **testgetriebener** Entwicklung wird kein Code ohne Test herausgegeben.
- **Verantwortungsbildend** ist ein Prozess, wenn Verantwortungen innerhalb des Prozesses klar definiert sind und das Übernehmen dieser Verantwortungen vom Prozess motiviert wird.

2.2.3 Interviewauswertung: Meinungsbild

Mit der ersten Frage des Interviews habe ich explizit erfragt wie erfolgreich der Prozess eingeschätzt wird. Insgesamt wurde der Prozess von allen befragten Personen als erfolgreich eingeschätzt, die Antworten variieren von sehr erfolgreich bis zu mäßig erfolgreich. Die Befragten empfinden das Arbeiten im Prozess als angenehm, denn der Prozess ist trotz seiner Größe und hohen Komplexität nachvollziehbar und verständlich. Ein weiterer positiver Effekt des Prozesses ist seine Planbarkeit, durch die Stressfaktoren, wie das Ändern von Anforderungen durch den Kunden während der Entwicklung, vermieden werden können. Im Zuge dieses Effektes wurde oft die dadurch verloren gegangene Flexibilität als Nachteil genannt. Dies wird gegenüber den gewonnen Vorteilen allerdings kaum als nachteilig empfunden.

Durch den NeMo-Prozess sei es den Entwicklern möglich sich auf ihre Hauptaufgabe, das Programmieren, zu konzentrieren. Die Zeit, die ihnen zum Entwickeln zur Verfügung stehe, könne ungestört genutzt werden. Die Zeit, die in Meetings verbracht wird, scheint, für den NeMo-Prozess allein betrachtet, effektiv auf einem Minimum gehalten zu werden.

Allerdings überschneiden sich in der DBaD GmbH durch horizontale Teams mehrere Projekte, wodurch einige Entwickler nicht nur im NeMo-Prozess arbeiten, sondern auch in weiteren Prozessen involviert sind. In diesem Fall erscheint die neben den Prozessmeetings übrig bleibende Zeit zum Entwickeln als zu gering. Um dies zu verbessern, könnte an den Schnittstellen des NeMo-Prozesses zu anderen Prozessen gearbeitet werden. Eine weitere Motivation, um an den Schnittstellen zu arbeiten, könnten die Aussagen sein, in denen die Zusammenarbeit mit anderen Teams als aufwendiger und komplizierter eingeschätzt wird als übergreifende Teamarbeit innerhalb des NeMo-Projektes.

Aus einer positiven Meinung kommt der Verbesserungswunsch, eine firmenweite Review zu halten, da man teilweise nicht mitbekomme was in den anderen Abteilungen überhaupt gemacht wird. Durch diese Review bekommt man innerhalb des NeMo-Projektes mit, woran die anderen Teams arbeiten. Die Review des NeMo-Prozesses findet ohne den Kunden statt und soll den Entwicklern eine Möglichkeit geben Geschaffenes vorzustellen.

Diese und weitere hier nicht erwähnte Schwachstellen zu verbessern, wäre eine Möglichkeit für diese Bachelorarbeit gewesen. In dem Zeitraum, in dem diese Bachelorarbeit entstanden ist, hat unangekündigt eine Umstrukturierung der Abteilungen begonnen. Durch diese Umstrukturierung verändern sich Teamzusammenstellungen. Es entstehen neue Schnittstellen zwischen Teams und bisher vorhandene fallen weg. An diese Veränderungen muss sich der NeMo-Prozess stark anpassen. Während der Umstrukturierung sind keine Kapazitäten mehr frei, um einzelne Prozesse gezielt zu verbessern. Die Erfolgsmessung des NeMo-Prozesses könnte dazu beitragen, dass dieser weiterhin verwendet und von anderen Abteilungen adaptiert wird. Bei einer

3. Entwicklung von Maßen für Erfolgskriterien des NeMo-Prozesses

positiven Aussage über den Erfolg des NeMo-Prozesses, die durch noch zu erhebende Messungen belegt werden könnte, scheint eine weitere Verwendung des Prozesses einleuchtend.

3 Entwicklung von Maßen für Erfolgskriterien des NeMo-Prozesses

Aus der Menge der ermittelten Kriterien (Abschnitt 2.2.2) konnten die sieben für den Erfolg des NeMo-Prozesses relevantesten Kriterien innerhalb weniger Tage ausgewählt werden (3.1). Unter Anwendung der Goal Question Metric Methode (3.2) habe ich sechs dieser Kriterien über einen Gesamtzeitraum von über einem Monat messbar gemacht und erprobt (3.3).

3.1 Auswahl der für den Erfolg relevanten Softwareprozess-Güte-Kriterien

Die Auswahl der Kriterien erfolgte aus dem zuvor durch Literatur und Interviews erstellten Kriterien-Pool. Zuerst wählte jeder NeMo-PM die für den Erfolg wichtigen Kriterien bewusst aus einer selbst festgelegten Perspektive aus. Bei einem gemeinsamen Termin wurden diese Kriterien zusammengetragen. Dabei stellte sich heraus, dass jede vom NeMo-PM ausgeübte Prozessrolle als Perspektive vertreten wurde. Im NeMo-Prozess übt ein Projektmanager mehrere Prozessrollen aus. Zum einen die Rolle eines PMs; in dieser Rolle betreut der NeMo-PM sein Entwickler-Team, kommuniziert mit dem Kunden und achtet auf das Budget. Zusätzlich werden die Rollen Scrum Master und Product Owner (PO) ausgeübt. Als Scrum Master wird auf die Einhaltung des Prozesses geachtet, als PO steht der Projektmanager bei Fragen zum Produkt zur Verfügung und hat eine Vision des Produktes [16]. Dabei vertritt der NeMo-PM den PO des Auftraggebers. Jeder der NeMo-PMs übt als eine Person alle diese Rollen aus. Daher ist die Auswahl der Kriterien aus der Gesamtheit jeder dieser Sichten repräsentativ für einen NeMo-PM. In der folgenden Tabelle wird dargestellt, aus welcher Sicht welche Kriterien ausgewählt worden sind.

Kriterium	Projektmanager	Product Owner	Scrum Master
Transparenz	x	x	x
Effizienz	x	x	x
Planbarkeit	x	x	
Verbesserungsfähigkeit	x		x
Softwarequalität	x	x	
Plausibilität	x		x
Skalierbarkeit			x
Kontrolle		x	
Stressfreiheit		x	

Tabelle 1: durch die NeMo-PMs ausgewählte Kriterien, Quelle: Eigene Tabelle

Transparenz ist ein für alle vom NeMo-PM ausgeübten Prozessrollen relevantes Kriterium. Ein transparenter Prozess vereinfacht die Kommunikation erheblich und

3.1 Auswahl der für den Erfolg relevanten Softwareprozess-Güte-Kriterien

macht diese so effektiv, dass sie optimalerweise kein hindernder Faktor ist. Hier ist der PO hauptsächlich am Informationsfluss interessiert, der das Projekt vorantreibt, während der Scrum Master und PM auch daran interessiert sind, dass der Prozessablauf allen bekannt ist.

An der **Effizienz** des Prozesses ist besonders der PO interessiert, da dieser der Geldgeber ist und daher möglichst viel aus seiner Investition erhalten möchte. Der PM muss den Aufwand gegenüber dem Kunden verantworten und versucht die gegebenen Ressourcen effektiv zu nutzen. Grundsätzlich ist Effizienz, zusammen mit Transparenz, einer der Hauptgründe, warum Prozesse eingesetzt werden, weshalb auch der Scrum Master ein Interesse daran hat.

Durch **Planbarkeit** lassen sich Aufwände planen und Aussagen dazu treffen, wann Features umgesetzt sein werden. Dies sind Informationen, aus denen hervorgeht welche Menge an Ressourcen benötigt wird. Diese Informationen sind für den PO und den PM aus dem zuvor genannten Grund relevant.

Die **Verbesserungsfähigkeit** des Prozesses ist vor allem für einen Scrum Master von Interesse. Denn jeder Prozess, der von seinen Beteiligten angenommen werden soll, muss sich ihren Bedürfnissen anpassen. Die Möglichkeit, den Prozess verbessern zu können, stellt einen effizienteren Prozess in Aussicht, was für den PM von Vorteil ist.

Eine möglichst hohe **Softwarequalität** ist das Hauptziel jedes Prozesses und vor allem für den PO von Interesse. Dieser möchte das am Ende des Softwareprojekt stehende Produkt weiterverwenden.

Die **Plausibilität** des Prozesses vereinfacht es den Beteiligten im Prozess zu arbeiten und dessen Vorgehensweisen zu adaptieren. Dies erleichtert die Einführung des Prozesses und später einem PM das Koordinieren der Beteiligten, da ein Großteil des eigentlich benötigten Koordinationsaufwandes vom Prozess übernommen wird.

Die **Skalierbarkeit** des Prozesses wurde ausgewählt, weil ein Prozess optimalerweise nicht nur für ein Projekt mit fester Teamgröße verwendbar sein sollte, sondern auch für andere Projekt mit anderen Teamzusammenstellungen und -größen verwendbar sein könnte.

Das neu hinzugekommene Kriterium **Kontrolle** soll beschreiben wie viel Einfluss der PO während des Prozesses hat, um das Endprodukt gegebenenfalls beeinflussen zu können. Dies entspricht Transparenz und Softwarequalität, weshalb es nicht als separates Kriterium betrachtet wird. In einem transparenten Prozess kann zu jedem Zeitpunkt vom PO überprüft werden, wie sehr die Entwicklung seinen Anforderungen entspricht. Falls aus seiner Sicht eine Abweichung erkannt wird, kann er darauf folgend im Rahmen des Prozesses handeln. Wie sehr das Endprodukt mit den zu Beginn gestellten Anforderungen übereinstimmt, wird mit dem Kriterium Softwarequalität betrachtet.

Das Kriterium **Stressfreiheit** wurde explizit so ausgewählt, dass es sich nur auf den PO bezieht. Die Gruppe der Prozessbeteiligten, die von diesem Kriterium angesprochen wird, ist minimal. Mögliche Verbesserungen dieses Kriteriums würden dem gesamten Prozess kaum Vorteile bringen. Daher wird dieses Kriterium nicht weiter in die Erfolgsmessung mit einbezogen.

3.2 Herleitung von Maßen mithilfe der Goal Question Metric Methode

Die Goal-Question-Metric (GQM) Methode (Ziel-Frage-Maß) ist ein zielorientierter Ansatz zum Definieren von Software-Maßen. Bei diesem Ansatz werden Software-Maße so konstruiert, dass mit den Erhebungen dieser Maße das zuvor gesetzte Ziel erreicht werden kann. Die Motivation für die Durchführung der Messungen ist als Ziel festgehalten. Dieses Ziel beschreibt, was warum gemessen werden soll, aus welcher Perspektive dies geschehen soll und in welchem Kontext die Messungen durchgeführt werden. Die Ermittlung der Maße erfolgt top-down, dabei werden die Ebenen nach unten hin immer konkreter und detaillierter. Auf der Ebene unter dem Ziel werden Fragen gesammelt. Deren Beantwortung ermöglicht es Schlüsse in Richtung des Ziels zu ziehen. Um die Fragen beantworten zu können werden Maße ermittelt, deren Erhebung die Fragen beantworten soll [17].

Das Ziel, die Erfolgsmessung des NeMo-Prozesses, habe ich als GQM-Ziel wie folgt festgehalten: **Analysiere den NeMo-Prozess, um ihn bezüglich seines Erfolges zu evaluieren, aus der NeMo-PM-Perspektive im Kontext des NeMo-Projektes.** Normalerweise folgen nach dem Ziel die Fragen und anschließend die Maße. Aber um den Erfolg des NeMo-Prozesses zu bewerten, sollen einzelne Kriterien gemessen werden. Aus den gesammelten Bewertungen der einzelnen Kriterien lässt sich der gesamte Prozess evaluieren. Dadurch ergibt sich eine Struktur mit einem Hauptziel "Evaluieren des Prozesses" und mehreren Unterzielen, wie "Analysiere den NeMo-Prozess, um seine Effizienz zu evaluieren, aus Sicht der NeMo-PMs im Kontext des NeMo-Projektes.". Analog ist jedes der ausgewählten Kriterien ein eigenes Unterziel.

Obwohl der Prozess für die Erfolgsmessung nur evaluiert werden soll, ist die Motivation für die Erfolgsmessung den Prozess nach einer Evaluierung zu verbessern, weshalb die Maße den Prozess hauptsächlich evaluierbar machen, aber auch Ansatzpunkte aufzeigen, an denen der Prozess verbessert werden könnte.

Um jedes Kriterium messbar zu machen habe ich in Zusammenarbeit mit den NeMo-PMs als ersten Schritt ein Brainstorming durchgeführt. In Anlehnung an die GQM Methode habe ich dieses in Abstraction Sheets festgehalten. Dies ist eine inhaltliche Formvorgabe mithilfe derer Ideen in 4 Sektionen aufgeteilt werden können: Es werden Messideen und Faktoren, die die Messungen beeinflussen könnten festgehalten. Zusätzlich werden Hypothesen dokumentiert, die eine Schätzung der Messergebnisse darstellen [17]. Die Erhebung der Messungen und das Auswerten derselben ist nicht Teil dieser Bachelorarbeit, da die Messungen über mehrere Sprints hinweg erhoben werden sollen und dies zeitlich nicht möglich war. Ein Sprint des NeMo-Prozesses geht über 3 Wochen, bei optimaler Lage könnte über 3 Sprints während der Bachelorarbeit gemessen werden. Dies würde aber die Zeit, die benötigt wird um die Kriterien zu finden und die Maße zu erarbeiten, außer acht lassen.

Die Erkenntnisse des Brainstormings habe ich als Ansatzpunkte zum Entwickeln der Maße genutzt. Diese weiterentwickelten Maße wurden in weiteren Meetings durch die NeMo-PMs auf deren Nutzbarkeit hin validiert. Weiterhin ist für jedes Maß sichergestellt worden, dass es vom Prozess beeinflusst wird, wenn auch nicht immer vollständig. Es haben sich verschiedene Kategorien von Maßen durchgesetzt. Maße, die nach Prozessbestandteilen suchen stehen oft mit solchen Maßen im Zusammenhang, die den Effekt eines Kriteriums messen. Solche Prozess-Maße müssen im Verlauf der

Messungen nur einmal erhoben werden, während die anderen Maße meist in größeren Mengen und in mehreren Zeiträumen wiederholt erhoben werden sollten, um Ergebnisse liefern zu können, die sich der Realität genug annähern. Durch die Erhebung dieser beider Maße soll zum einen der Zusammenhang zwischen Prozess und dem was der Prozess bewirken kann sichtbar werden. Außerdem sollen Anhaltspunkte für mögliche Prozessverbesserungen aufgezeigt werden.

Bei der Auswertung aller Messungen ist zu beachten, dass der Prozess meist die Möglichkeit schaffen kann, etwas besser umzusetzen, die optimale Umsetzung dieses Vorschlags aber nicht garantieren kann. Es ist der Mensch im Prozess, der diesen umsetzen kann, ebenso wie der Mensch im Prozess es ist, der die Möglichkeit hat diesen zu verbessern. Der Prozess erleichtert viele Arbeiten, kann sie dem Menschen jedoch nicht vollständig abnehmen.

3.3 Maße der einzelnen Kriterien

In folgenden Unterabschnitten werden die ausgewählten Maße (3.1) je Kriterium nach QM Fragen strukturiert beschrieben. Pro Kriterium ist eine Zusammenfassung aller Maße vorhanden (C).

3.3.1 Skalierbarkeit

Skalierbarkeit bezeichnet die Fähigkeit, dass der Prozess nicht nur auf ein Projekt angewendet werden kann sondern auch auf andere ähnliche Projekte, welche sich sowohl in der Anzahl der Beteiligten als auch in der Projektgröße unterscheiden können. Die Skalierbarkeit eines Prozesses sagt aus, für wie viele Beteiligte und was für Projektgrößen der Prozess eingesetzt werden kann.

Um die Skalierbarkeit des NeMo-Prozesses zu ermitteln, könnte die Eigenschaft zunutze gemacht werden, dass im NeMo-Projekt Sprints zu auf den Tickets basierenden Festpreisen abgerechnet werden. Jedes Ticket, was immer einer einzelnen Anforderung entspricht, wird geschätzt und basierend auf dieser Schätzung wird der Preis des Tickets berechnet. Hier könnte der Punkt errechnet werden, ab dem die Einnahme durch die im Sprint erledigten Tickets nicht mehr den gesamten Aufwand deckt, was Prozessoverhead und Entwicklungszeit entspricht. Diese Berechnung der Skalierbarkeit funktioniert allerdings nur für die minimale Sprintgröße, nicht für die maximale, da Meetings ab einer bestimmten Anzahl an Teilnehmern nicht mehr sinnvoll sind.

Eine weitere Möglichkeit, die Skalierbarkeit des NeMo-Prozesses zu messen, wäre es, den NeMo-Prozess mit verschiedenen vielen Beteiligten laufen zu lassen und dabei zu überwachen. So könnte herausgefunden werden mit was für Teamgrößen der NeMo-Prozess funktioniert. Dies wäre aber ungemein aufwendig und der Nutzen wäre für das NeMo-Projekt nicht sonderlich hoch. Analog dazu könnte der NeMo-Prozess mit verschiedenen Projektgrößen getestet werden.

Wie sehr ein Prozess skalierbar ist und in welchem Rahmen ist eine nützliche Information, sobald feststeht, dass der Prozess für andere Projekte verwendet werden soll. Welche Stärken und Schwächen der Prozess mit sich bringt, wird durch die Skalierbarkeit des Prozesses nicht sichtbar, weshalb Skalierbarkeit eher weniger Aussagekraft über den Erfolg eines Prozesses hat und deshalb nicht weiter betrachtet wird.

3.3.2 Transparenz

Transparenz bedeutet, dass Informationen so verteilt sind, dass sie für alle zugänglich sind. Aus der Sicht der NeMo-PMs sollte dabei sowohl der Prozess transparent sein, als auch der Fortschritt des Sprints, Projekts oder Tickets.

Damit die Transparenz des Prozesses beurteilt werden kann, sollte daher durch "F1: Welches Prozesswissen ist verfügbar und wie ist es verteilt?" und "F3: Wie sind Fortschritte und Abhängigkeiten dokumentiert?" ermittelt werden, wie verbreitet Prozess- und Projektwissen unter den Beteiligten des NeMo-Prozesses ist. Um den Zusammenhang zum Prozess sichtbar zu machen wird Frage "F2. Mit welchen Mitteln schafft der Prozess Transparenz?" gestellt. Mit "F4. Welche Informationen fehlen während eines Sprints?" hoffe ich Intransparentes zu finden, was sonst möglicherweise übersehen würde. Eine zusammenfassende Darstellung aller Maße ist im Anhang C.1 zu finden.

Um die Prozesstransparenz zu beurteilen, soll die Qualität der verfügbaren Informationen, sowie die Verbreitung dieser gemessen werden. Die Qualität der aufgelisteten Informationen soll hinsichtlich ihrer Zugänglichkeit, Lesbarkeit, Vollständigkeit und Korrektheit bewertet werden (M1.1.), letzteres durch Prozessexperten. Sind Informationen nicht zugänglich und lesbar, können sie niemandem helfen, Wissenslücken zu füllen. Durch unvollständige und fehlerhafte Informationen werden irreführende Informationen verbreitet. Die Informiertheit der Prozessbeteiligten (M1.2.) kann durch eine automatisierte Umfrage ermittelt werden, die zur Sicherstellung ehrlicher Antworten Zeitlimits und Freitextfelder einsetzt. Durch Zeitlimits soll das nachträgliche Suchen nach nicht bekannten Informationen verhindert werden, in dem Freitextfeld soll jeweils die abgefragte Information oder deren Lagerort angegeben werden. Ich habe den NeMo-Prozess in Phasen unterteilt, für die ich gemeinsam mit den NeMo-PMs festgelegt habe, was bekannt sein sollte (C.1.2). Zusätzlich wurde je Phase bestimmt, welche Prozessrollen zwingend informiert sein sollten (siehe Anhang C.1.1). Dadurch sollen nicht ausreichend transparente Prozessphasen erkannt werden. Beispielsweise sollte in der Planungsphase, in der ein Sprint geplant wird, bekannt sein, welche Tickets im Moment eingeplant sind. Dies ist im Sprintboard dokumentiert. Es sollte außerdem klar sein wann der Sprint beginnt, endet und welche Meetings in der Phase stattfinden. Teilnehmern von Meetings sollte immer der Zeitpunkt, der Ort, die Agenda und der Grund für ihre Anwesenheit bekannt sein. Können alle Umfrageteilnehmer aufzeigen wo diese Informationen zu finden sind, ist der Prozess zumindest in der Planungsphase transparent.

Es sollen Meetings (M2.1.) und Artefakte (M2.2.) gefunden werden, die Transparenz schaffen. Dafür bietet es sich an die Prozessdokumentation durchzusehen. Dies sollte in Zusammenarbeit eines Prozessexperten mit einer mit den Messungen betrauten Person passieren, da es keine zentrale Prozessdokumentation des NeMo-Prozesses gibt. Die Person des Messteams soll während der Erhebung dieser Maße darauf achten, dass nur relevantes festgehalten wird.

Das NeMo-Projekt ist ein fortlaufendes Projekt, welches kein komplett neues Softwareprodukt entwickelt, sondern an einem großen Softwareprodukt einzelne Features implementiert oder Bestehendes verändert. Aus diesem Grund wird kein Projekt-

fortschritt benötigt. Wichtig ist der Fortschritt einzelner Tickets (M3.1.), aus welchem sich der Fortschritt des Sprints (M3.2.), so wie der Features (M3.3.) konstruieren lässt. In der DBaD GmbH wird das Ticketsystem JIRA verwendet [2], die vorgesehene Art des Umgangs mit diesem System und seinen Funktionen ist im NeMo-Prozess festgelegt. Dieser besagt, dass über ein Ticket der Fortschritt davon abzulesen ist. Im Sprintboard soll der Sprintstatus abzulesen sein, der sich aus einzelnen Tickets zusammensetzt. Dies gilt auch für Features, die als Epics modelliert sein sollten. Daher reicht es, den Fortschritt des Tickets in diesem zusammen mit der Vollständigkeit der Abhängigkeiten zu überprüfen. Diese Beurteilung geschieht am besten durch diejenigen, die an der Umsetzung des Tickets interessiert waren. Das trifft für jedes Ticket mindestens auf einen NeMo-PM zu, der dafür der zuständige PO war. Abhängigkeiten sollten zusätzlich durch den am Ticket beteiligten Entwickler beurteilt werden, nicht aber der Fortschritt während der Entwicklung, da er für die Dokumentation des Fortschritts verantwortlich war. Diese Bewertungen sollen mit Beendigung jedes Tickets erhoben werden. Zur einfacheren Auswertung kann jedes Qualitätskriterium mit der im Anhang unter M3.1 - M3.3. vorgeschlagener Skala bewertet werden. Analog zu M3.1. können M3.2. und M3.3. erhoben werden, wobei hier darauf zu achten ist, nicht die einzelnen Tickets zu bewerten.

Den gesamten Messzeitraum über soll dokumentiert werden, wenn Informationen fehlen (M4.1.). Bei Probemessungen ist aufgefallen, dass anscheinend kaum Informationen fehlen. Dies könnte daran liegen, dass Informationen nicht als fehlend wahrgenommen werden, so lange eine Person über diese verfügt. Aber ein Prozess mit einer einzigen Person die alles weiß, ist dadurch nicht transparent. Solche scheinbar vorhandenen Informationen sollten in die Messung miteinbezogen werden, weshalb Prozessbeteiligte nicht nur angehalten werden sollten ihnen fehlende Informationen zu dokumentieren, sondern auch Informationen, nach denen sie gefragt werden. Für jede Information soll deren Wichtigkeit analog zur Ticketpriorisierung in JIRA dokumentiert werden.

3.3.3 Effizienz

Effizienz bedeutet, dass mit den gegebenen Ressourcen mehr oder schneller Output generiert werden kann. Um festzustellen wie schnell der NeMo-Prozess ist, muss beantwortet werden, wie schnell eine Anforderung umgesetzt werden kann (F1.). Für den NeMo-Prozess sehe ich diese Messungsmethode als passender an, als die Menge an umgesetzten Anforderungen zu messen. Diese Menge hängt direkt von den stark variierenden Kapazitäten der Teams je Sprint und der vom Kunden gewünschten Sprintgröße ab. Das Softwareprodukt ist das Ziel jeder Entwicklung, weshalb die Zeit der Entwickler die wichtigste Ressource in einem Softwareentwicklungsprozess ist. Diese Zeit soll so gut wie möglich ausgenutzt werden. Es stellt sich die Frage "F2: Wie viel Zeit steht einem Entwickler effektiv zum Programmieren zur Verfügung?". Zur effektiven Zeitausnutzung zählt auch, dass der Aufwand für Kommunikation im Prozess minimal gehalten wird, während gleichzeitig die minimal notwendige Kommunikation vom Prozess organisiert wird. Um "F3. Wie effizient sind Meetings

3. Entwicklung von Maßen für Erfolgskriterien des NeMo-Prozesses

und Calls?“ zu beantworten, wird genau dies betrachtet. Eine zusammenfassende Darstellung aller Maße ist im Anhang C.2 zu finden.

Um zu ermitteln, in welchem Zeitraum eine Anforderung umgesetzt werden kann, werden einzelne Tickets als eine Anforderung betrachtet und deren Erstellungs- und Deploymentzeitpunkt ermittelt (M1.1.). Werden die Ticketstatus in die Messung mit einbezogen, kann ermittelt werden, wie lange ein Ticket in welcher Prozessphase verbringt (M1.2.) und wie oft es diese wiederholt (M1.3.). Um zu ermitteln, wie lange ein Ticket in einer bestimmten Prozessphase liegt, müssen diesen Ticketstatus zugeordnet werden. Durch diese Messergebnisse könnten ineffiziente Prozessphasen identifiziert werden. Beim Erheben der Maße M1.2. und M1.3. ist zu beachten, dass im NeMo-Projekt mit zwei JIRAs gearbeitet wird und die Ticketstatus nicht übereinstimmen. Die Planung, das Testen und die Abnahme werden im JIRA des Kunden festgehalten, während die Entwicklung im JIRA der DBaD GmbH geschieht. Weiterhin ist zu beachten, dass der Status 'blocked' keiner Prozessphase eindeutig zuzuordnen ist. Die Erhebung der Daten kann durch das JIRA-Addon 'Time in Status' [13] erleichtert werden. Dieses zeigt je Ticket die in den Status verbrachte Zeit an. Vor Start der Messungen sollte bei den Beteiligten daran erinnert werden, Status genauer zu setzen, da bei Testmessungen nicht korrekt gesetzte Status erkannt worden sind.

Zur Ermittlung wie viel Zeit einem Entwickler zur Verfügung steht und wie viel vom Prozess beansprucht wird (M2.1.), können die Büro-Kalender der Entwickler ausgewertet werden. Dabei sollte der Zeitraum eines Sprints betrachtet werden, da es Meetings gibt, die nur einmal pro Sprint stattfinden. Außerdem ist auf die Pflege der Kalender zu achten. An welchen Meetings die Entwickler beteiligt sind, lässt sich aus Tabelle 5 ableiten. Wird nur die Gesamtheit der Entwicklungszeit betrachtet, wird die Zeit außer Acht gelassen, die nach jeder Unterbrechung zur erneuten Einarbeitung aufgewandt werden muss. Daher werden die Zeitabstände zwischen Meetings gemessen (M2.2.). Ein weiterer Aspekt, der die Entwicklungszeit der Entwickler beeinträchtigt, sind Unterbrechungen. Unterbrechungen während des Entwickelns sollen fortlaufend protokolliert werden (M2.3.). Optimalerweise sollte es kaum Unterbrechungen geben. Falls Entwickler häufig unterbrochen werden, können anhand der Messergebnisse Rückschlüsse auf die Ursache gezogen werden und mögliche Verbesserungen eingeleitet werden.

Um festzustellen wie effizient die den Entwicklern durch den Prozess genommene Zeit genutzt wird, wird für alle Meetings gemessen, wie die angesetzten Meetingzeiten mit den reellen übereinstimmen (M3.1.), ob Meetings durch fehlende Teilnehmer behindert werden (M3.2.) und wie sehr der Inhalt des Meetings der geplanten Agenda entspricht (M3.3.). Alle Maße sollten von der Person dokumentiert werden, die das Meeting führt.

Eine weitere Möglichkeit die Effizienz des Prozesses zu messen, wäre zu ermitteln wie wirtschaftlich der Prozess ist, indem die Aufwände mit dem Verkaufspreis in Beziehung gesetzt werden. Dadurch würde allerdings nur gemessen werden, wie wirtschaftlich das Angebot gestellt wurde. Aus diesem Maß könnten keine Schlüsse über die Effizienz des Prozesses gezogen werden oder darüber wie der Prozess effizienter werden könnte, sondern nur, wie das Angebot verändert werden sollte.

3.3.4 Planbarkeit

Planbarkeit schlüsselt sich in Planungssicherheit und Genauigkeit von Schätzungen auf. Mit den Schätzungen werden Vorhersagen gemacht. Durch die Planungssicherheit kann garantiert werden, dass die in Schätzungen einbezogene Informationen nach der Schätzung unverändert bleiben. Daher lässt sich die Planbarkeit des Prozesses evaluieren, indem folgende Fragen beantwortet werden: "F1. Welche Planungssicherheiten werden erwartet und werden sie eingehalten?" und "F2. Wie akkurat sind Vorhersagen?" Die Akkuratheit der Messungen wird zu einem großen Teil durch die Fähigkeit der schätzenden Person bestimmt; der Prozess kann diese Schätzungen durch Vor- und Nachbereitungen verbessern (F3.). Eine zusammenfassende Darstellung aller Maße ist im Anhang C.3 zu finden.

Für Planbarkeit wird überprüft, ob erwartete Planungssicherheiten zutreffen und wie sehr von ihnen abgewichen wird. Um dies zu messen wird ermittelt, wofür und wann Planungssicherheiten erwartet werden (M1.1.), die Erhebung kann analog zu Transparenz M1.1. - M1.2. erfolgen und könnte vereinfacht werden durch zuvor erarbeitete Vorschläge von einer in die Messungen eingearbeitete Person. In M1.2. werden die erwarteten Planungssicherheiten gemessen, indem ab dem Punkt, ab dem z.B. die Anforderungen nicht mehr geändert werden dürfen, jede Änderung in den Anforderungen dokumentiert wird.

Damit wird die Akkuratheit, der aus meiner Sicht wichtigsten Schätzungen im Prozess, gemessen. Jedes der Maße vergleicht den geschätzten Wert mit dem dazu passenden realen Wert. Stimmen beide überein ist die Schätzung akkurat. Es wird gemessen, wie genau die Schätzungen der Anforderungen sind, indem die Story Points betrachtet werden. Mit Story Points wird in Scrum die Komplexität einer Anforderung eingeschätzt [16]. Die vor Beginn des Sprints geschätzten Story Points eines Tickets werden nach Ende des Sprints mit den auf dieses Ticket verbuchten Zeiten aus dem Buchungssystem verglichen (M2.1.). Bei den Ergebnissen dieser Messungen ist zu beachten, dass Komplexität und Aufwand verschiedene Werte sind. Folgend wird daher gemessen wie gleichmäßig die Schätzungen sich verhalten. Dies wird ausgewertet, indem für jedes Ticket errechnet wird, wie groß ein Story Point wäre und die Ergebnisse dieser Berechnungen verglichen werden. Sind die durchschnittlichen Story Points in etwa gleichgroß, sind die Schätzungen ausreichend akkurat. Bei ersten Messungen ist aufgefallen, dass Zeiten nicht korrekt erfasst sind und dadurch Aufwände von Tickets verloren gegangen sind. Bei Messungen ist darauf zu achten, dass alle am Ticket Beteiligten bereits gebucht haben. Neben genannter Messung wird überprüft, wie gut der vorhergesagte Sprintumfang für das gesamte NeMo-Team eingehalten wird (M2.2), wie genau die Kapazitäten je Team eingeschätzt werden (M2.3.) und wie gut das Deploymentdatum eingehalten wird (M2.4.).

Um in die Beurteilung wie planbar der Prozess ist, miteinzubeziehen was der Prozess tut, um die Planbarkeit zu erhöhen, soll ermittelt werden wie der Prozess Schätzungen verbessert (M3.2.). Dieses Maß unterscheidet sich in den zu ermittelten Prozessaufgaben von Transparenz M1.1. - M1.2. und kann daher analog erhoben werden.

Wie viele Sprints im Vorraus mithilfe des Prozesses geplant werden können, ist

3. Entwicklung von Maßen für Erfolgskriterien des NeMo-Prozesses

eine weitere Frage, die man sich im Bezug auf Planbarkeit stellen sollte. Dies kann für den NeMo-Prozess auch ohne Messungen beantwortet werden. Ähnlich wie im Scrum-Prozess plant der NeMo-Prozess immer nur den aktuellen und den nachfolgenden Sprint. Aus Erfahrungen ist bekannt, dass es mit dem NeMo-Prozess nicht möglich ist, weiter zu planen. Diesen Anspruch muss der Prozess nach aktuellem Stand nicht erfüllen, weshalb dies nicht in die Erfolgsmessungen einbezogen werden muss.

3.3.5 Softwarequalität

Das Kriterium Softwarequalität beschreibt mehrere Eigenschaften des Endproduktes. Für die Erfolgsmessung des NeMo Prozesses reicht es für eine erste Messung aus, die Anforderungsgleichheit (F1.), die Code Qualität (F2.) sowie die Fehlerfreiheit (F3.) zu betrachten. Dies soll mit der Beantwortung der Fragen "F1. Wie weit entspricht die Umsetzung den Anforderungen?" und analog "F2. Wie weit kann der Prozess die Fehlerfreiheit des Codes beeinflussen?" und "F3. Wie gut strukturierten Code bringt der Prozess hervor?" messbar gemacht werden. Dies resultiert daraus, dass das NeMo-Projekt kein Projekt ist mit dem ein komplett neues Softwareprodukt programmiert wird, sondern ein fortlaufendes Projekt, welches ein nun bestehendes System wartet, verändert und ausbaut. Der Auftraggeber ist im NeMo-Projekt ein externer PO, der einzelne Anforderungen gesammelt in Sprints in Auftrag gibt. Die Konkretisierung dieser Tickets, sowie deren Umsetzung, aus den vorgegebenen Anforderungen und dem Konzept heraus, ist Aufgabe des NeMo-Teams und der Projektmanager, die als eine Art Proxy-PO fungieren. Der Proxy-PO vertritt einen zu weit entfernten PO, indem er seine Rolle übernimmt und das Produkt betreut. Die gesammelten Maße für Softwarequalität sind im Anhang C.4 zu finden.

Um F1. zu beantworten, werden Werte erhoben, aus denen die Anforderungsgleichheit von Spezifikationen zur Umsetzung je Ticket erkennbar ist (M1.2.). Bei der Erhebung dieses Maßes wird davon ausgegangen, dass die im Ticket dokumentierten Anforderungen vollständig sind. Es werden nur Tickets in die Messung miteinbezogen, deren Anforderungen konkret nachzuprüfen sind. Dies lässt sich über die Umsetzungstickets ermitteln, in welchen entsprechend gekennzeichnete Untertickets für nicht erfüllte Anforderungen liegen. Solche Untertickets werden von Entwicklern und Testern erstellt und können nach Beendigung des Sprints ausgewertet werden. Um dieses Maß besser auf den Prozess beziehen zu können, soll analog zu M2.1. - M2.2. von Transparenz ermittelt werden, mit welchen Mitteln der Prozess die Anforderungsgleichheit beeinflussen kann (M1.1.).

Indem Fehler pro Ticket gemessen werden (M2.2), soll eingeschätzt werden, wie fehlerfrei das Produkt am Ende des Prozesses ist. Der Begriff Fehler bezeichnet hier Fehlverhalten der Software. Es muss beachtet werden, dass derselbe Fehler verschieden wichtig sein kann. Ein nicht sichtbares Komma eines Textes ist ein unschön zu lesender Rechtschreibfehler, ein nicht sichtbares Komma in einer Preisangabe dagegen wäre erheblich schlimmer. Daher muss bei der Erhebung solcher Messungen nicht nur der Fehler an sich betrachtet werden, sondern auch der Kontext. Um diese Prio-

risierung der Fehler in der Messung zu beachten, sollen zur Vereinfachung der Messung nur solche Fehler beachtet werden, die von der Qualitätssicherung im JIRA am höchsten priorisiert wurden. Solche Fehler sind als Subtasks je Ticket dokumentiert. Wie beim vorherigen Aspekt der Softwarequalität, wird auch hier ermittelt, womit der Prozess Fehler reduziert (M2.2.).

Die Qualität des Codes wird an seiner Strukturiertheit gemessen (M3.3.). Jedes umgesetzte Ticket wird von mindestens einem weiteren Entwickler gereviewed, welcher damit praktisch schon das Maß M3.3. erhebt und seine Bewertung nur noch dokumentieren muss. Der Einfachheit halber soll dies in Form einer Schulnote getan werden, welche ein stark vereinfachtes Urteil über die Strukturiertheit dieses Codes bildet. Analog zu den vorherigen Aspekten sollen auch hier alle vom Prozess vorgesehenen Regeln und Aufgaben, die die Code Qualität begünstigen, ermittelt werden (M3.2.). Zusätzlich dazu soll zu Ende jedes Sprints von Entwicklern und NeMo-PMs je Ticket festgehalten werden, welche Methoden bei der Umsetzung des Tickets angewendet wurden um den Code zu verbessern (M3.1.). Dies könnte in Verbindung mit M3.3. mögliche Methoden aufzeigen, welche die Code Qualität verbessern und daher fest in den Prozess integriert werden sollten.

3.3.6 Verbesserungsfähigkeit

Unter Verbesserungsfähigkeit verstehe ich die Fähigkeit des Prozesses sich zum positiven zu verändern. Dabei sind vor allem die Intentionen der Prozessbeteiligten ausschlaggebend und die vom Prozess gegebenen Möglichkeiten diese zur Sprache zu bringen und anzuwenden. Daher ist die wichtigste zu beantwortende Frage aus meiner Sicht "F1. Wie intensiv wird sich im Prozess mit dem Prozess auseinandergesetzt?" Denn erst durch diese bewusste Auseinandersetzung mit dem Prozess entsteht die Möglichkeit diesen zu verbessern. Wie erfolgreich das getan wird, wird durch die Fragen "F2. Wie erfolgreich werden Probleme gelöst?" und "F3. Wie verbessert sich der Prozess langfristig?" betrachtet. Alle Maße, um die Verbesserungsfähigkeit zu messen sind im Anhang C.5 zu finden.

Um die Frage F1. zu beantworten, wird der Prozess explizit nach sich mit Prozessverbesserung beschäftigenden Meetings (M1.1.) und Aufgaben (M1.2.) durchsucht. Dieses Maß kann ähnlich wie Transparenz M2.1. - M2.2. erhoben werden. Wie bereits erwähnt, halte ich es für wichtig, dass der Prozess regelmäßig Möglichkeiten gibt diesen zu kritisieren. Aus Protokollen soll hervorgehen, wer sich wie oft und wie lange mit Prozessverbesserungen beschäftigt (M1.3.) und daraus wie bewusst und in welcher Regelmäßigkeit dies geschieht. Die Maße M1.1. und M1.2. können durch eine mit M1.3. ergänzende Umfrage erhoben werden, in der Beteiligte gefragt werden, wann sie sich aktiv mit dem Prozess beschäftigen und ob ihnen Meetings oder Prozessaufgaben mit solch einer Agenda bekannt sind.

Eine kurzfristige Prozessverbesserung entspricht meist der Lösung eines Problems. Wie erfolgreich der Prozess solche kurzfristigen Verbesserungen umsetzen lässt, soll über eine Dokumentation aller Prozessveränderungen pro Sprint (M2.2.) und aller angegangenen Probleme gemessen werden (M2.1.). Wie veränderbar der Prozess kurzfristig ist, soll durch M2.2. gemessen werden. Dieses Maß soll periodisch für jeden

Sprint erneut erhoben werden. Zur Auswertung sollten allerdings mehrere Sprints betrachtet werden. Die Erhebung dieser Maße kann von den NeMo-PMs durchgeführt werden, da diese zu Teilen Scrum Master sind und daher an den meisten Prozessveränderungen selbst beteiligt sind oder über diese informiert werden.

Die langfristigen Ziele und deren Erfolgsquote sollen zu einem Teil durch die Betrachtung der gesetzten Ziele (M3.1.), der davon erreichten Ziele (M3.2.) und den Aktivitäten, die unternommen wurden um ein Ziel zu erreichen (M3.3.), betrachtet werden und zum anderen über langfristige Trends anderer Kriterien (M3.4.). Die Auswertung dieser Maße ähnelt einem Quartalsrückblick. Da NeMo-PMs auch die Scrum Master Rolle ausüben, bietet es sich an, dass diese bei der Erhebung der Maße unterstützen werden. Bei der Erhebung dieser Maße ist darauf zu achten, dass falls es keine Ziele gibt, alle weiteren Maße unter F3. nicht erhoben werden müssen. Da keine gezielten Anstrengungen unternommen werden den Prozess langfristig zu verbessern. Verbesserungen in Trends der anderen Kriterien wären Zufälle, die weder dem Prozess noch den Prozessbeteiligten zuzuschreiben wären.

3.3.7 Plausibilität

Das Kriterium Plausibilität ist ein gänzlich subjektives Kriterium. Es beschreibt die Nachvollziehbarkeit des Prozesses aus Sicht der Prozess-Beteiligten. Zur Einschätzung der Plausibilität, bietet sich folgende Frage "F1. Als wie plausibel wird der Prozess wahrgenommen?" an. Ein erweiternder Aspekt an dem die Plausibilität des Prozesses festgemacht werden kann, lässt sich durch die Frage "F2. Wie genau wird sich an den Prozess gehalten?" darstellen. Eine Zusammenfassung aller Maße für Plausibilität ist im Anhang C.6 zu finden.

Um F1. zu beantworten gibt es zwei Herangehensweisen, die zusammen ausgewertet werden sollten. Zum einen wird ermittelt wie nachvollziehbar der Prozess den Beteiligten erscheint, aufgeteilt in die Gesamtbetrachtung des Prozesses (M1.1.) und Prozessphasen (M1.2.). Zur Erhebung dieser Maße soll die Frage "Als wie einleuchtend empfindest du den NeMo-Prozess?" gestellt werden und mit einer Skala von 'einleuchtend/intuitiv' bis 'erscheint sinnlos' beantwortet werden. Zum Gegenprüfen der Ergebnisse aus M1.1. und M1.2. wird die Unzufriedenheit in Form von Beschwerden gemessen (M1.3.). Ein plausibler Prozess sollte möglichst wenige Beschwerden hervorrufen. Es bietet sich an, dieses Maß nach Ende jedes Sprints im Rahmen der Retrospektive zu erheben, einem auch in Scrum vorgesehenen Meeting in dem rückblickend betrachtet wird, wie der Sprint empfunden wurde [16]. Beschwerden sollen aber hauptsächlich von den NeMo-PMs den Sprint über dokumentiert werden, da Beschwerden oft an sie heran getragen werden. Die Nutzung beider genannter Messvorschläge führt zu einem vollständigeren Bild. Das Erheben der Maße sollte über mehrere Sprints wiederholt werden. Wird sich wiederholt über eine bestimmte Prozessaufgabe oder -phase beschwert, scheint diese nicht einleuchtend zu sein und sollte näher betrachtet werden. Bei ihrer Erhebung ist zu beachten, dass dies nicht als eine Gruppenumfrage stattfinden sollte, um gegenseitige Beeinflussungen auszuschließen.

Ein weiteres Indiz für die Plausibilität eines Prozesses ist wie sehr sich an den Prozess gehalten wird. Das kann mit der Häufigkeit und Regelmäßigkeit von Pro-

zessabweichungen ermittelt werden (M2.1). Prozessabweichungen könnten allerdings unbemerkt ausgeführt werden, weil die Abweichung beispielsweise schon so oft geschehen ist, dass sie zur Routine gehört. Solches Verhalten ist nicht einfach zu erkennen. Bewusste Prozessabweichungen können allerdings durch eine Umfrage ermittelt werden und Erkenntnisse liefern. Unbewusste Prozessabweichungen könnten durch Außenstehende bemerkt werden. Weiterhin könnte versucht werden, diese zu ermitteln, indem ein Prozessexperte sich von einem Prozessbeteiligten dessen Prozessablauf schildern lässt und alle nicht dem Prozess entsprechenden Aussagen notiert. Dies ist allerdings extrem aufwändig, für eine erste Erfolgsmessung kann auf diese Art der Maßerhebung verzichtet werden.

4 Übersicht aller Maße im Prozessablauf

In der folgenden Grafik wurden alle zu erhebenden Messungen in den Prozessablauf eingeordnet dargestellt. Dabei bilden die Spalten die Prozessphasen ab, während in den Zeilen die Kriterien ihrer Relevanz nach aufgelistet sind.

In der obersten Zeile ist der Ablauf des NeMo-Prozesses dargestellt. Der NeMo-Prozess ist ein an das Nexus-Framework angelehnter Prozess. Das Nexus-Framework ist ein Large Scale Scrum Prozess [14]. Der Prozess kann gradlinig dargestellt werden, da die Phasen wasserfallartig sind. Zu Beginn des Prozesses steht die Anforderung, welche in der Refinement-Phase so weit analysiert wird, dass alle Informationen, die zur Umsetzung benötigt werden, dokumentiert sind. In der Sprintplanungsphase wird konkret geplant, welche Tickets im nächsten Sprint umgesetzt werden sollen. Diese werden dann in der Umsetzungsphase umgesetzt. Wie der Name sagt, wird in der Testphase getestet und in der letzten Phase, dem Rückblick, wird auf den vergangenen Sprint geschaut. Bei diesem Rückblick wird überprüft, was im Sprint erreicht wurde, und es wird versucht aus dem vergangenen Sprint zu lernen und Verbesserungen für den nächsten Sprint mitzunehmen. Eine Besonderheit des NeMo-Prozesses ist die, dass während ein Sprint X läuft, immer die im vorherigen Sprint X-1 umgesetzten Anforderungen getestet und deployt werden und schon der nächste Sprint X+1 mit Refinements vorbereitet wird.

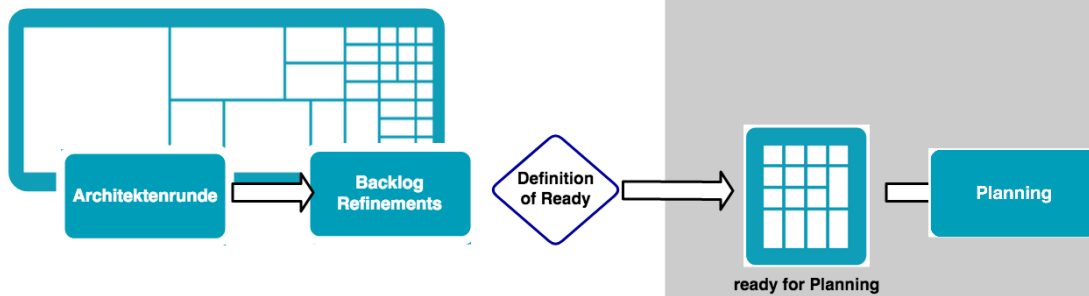
Jedes in der Grafik erwähnte Maß ist durch eine Kennung zu identifizieren, die zum jeweiligen Kriterium gehört. Die Liste aller Maße ist nach Kriterium geordnet im Anhang C zu finden.

Refinement-Phase

Sprintplanungsphase

Ablauf des NeMo-Prozesses nach Phasen

vom Prozess zeitlich unabhängige Maße



Transparenz

- M1.1. Qualität der Prozessinformationen
- M1.2. Informiertheit der Prozessbeteiligten
- M2.1. Transparenz schaffende Meetings
- M2.2. Transparenz schaffende Artefakte

M4

Effizienz

- M2.1. Prozesszeit und Entwicklungszeit
- M2.2. Zeitabstände zwischen Meetings
- M3.1.-M3.3. geplante Agenda, Start- und Endzeit je Meeting

M2.3. dokumentiere

- M3.1. • je Meeting: fehlende Teilnehmer, Start- und Endzeit, besprochene Themen

- M3.1. • je Meeting: fehlende Teilnehmer, Start- und Endzeit, besprochene Themen

- M3.1. • je Meeting: fehlende Teilnehmer, Start- und Endzeit, besprochene Themen

Planbarkeit

- M1.1. erwartete Planungssicherheiten
- M3.2. Prozessmaßnahmen mit Ziel, Schätzungen zu verbessern

- M2.1. • Ticket-ID
- DevPoint

M1.2. dokumentiere Änderungen in Artefakten (Story Points, Sprintumfang, ...)

M3.2. d

Softwarequalität

- M1.1. Anforderungsgleichheit begünstigende Prozessmaßnahmen
- M2.1. Fehlerquote reduzierende Prozessmaßnahmen
- M3.2. Codequalität verbessernde Prozessmaßnahmen

M3.1. dokumentiere je Ticket alle angewandten Methoden u

Verbessernd

- M1.1. Prozessmeetings, die sich mit Prozessverbesserung beschäftigen
- M1.2. Prozessmaßnahmen, die sich mit Prozessverbesserung beschäftigen
- M1.3. Aufwand um den Prozess zu verbessern je Beteiligten
- M3.2. • langfristige Prozessziele
- M3.4. Trends anderer Prozess-Güte-Kriterien

- M2.1. • In diesen Verbesser (Ziele)

M2.1.-M2.

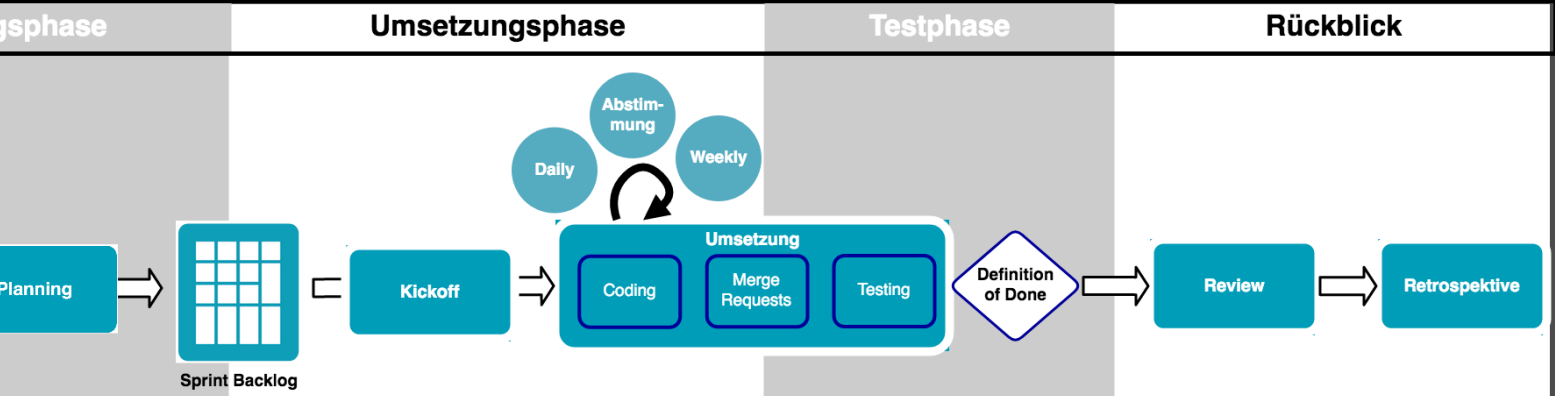
M3.2.-M3.3. dokumentiere alle

Plausibilität

- M1.1. Nachvollziehbarkeit des Prozesses
- M1.2. Nachvollziehbarkeit der einzelnen Prozessphasen
- M1.3. Beschwerden über den Prozess

M2.1. dokumentiere Pr

Abbildung 1: Darstellung aller Maße je Kriterium im NeMo-Prozessablauf - Teil 1, Quelle: Eigene Darstellung, Gesamtansicht des Messplans nach Prozess (<https://t1p.de/xd1d>)



- M3.1.**
 - Im Ticket ablesbarer Ticketfortschritt
- M3.2.**
 - Im Epic ablesbarer Featurefortschritt
- M3.3.**
 - Im Sprintboard ablesbarer Sprintfortschritt

M4.1. dokumentiere alle fehlenden Informationen

- M1.1. & M1.2.**
 - Umsetzungsdauer je Ticket nach Prozessphasen und gesamt

M3. dokumentiere alle Unterbrechungen in der vorgesehenen Entwicklungszeit

- M3.1.**
 - je Meeting: fehlende Teilnehmer, Start- und Endzeit, besprochene Themen

- M3.1.**
 - je Meeting: fehlende Teilnehmer, Start- und Endzeit, besprochene Themen

- M3.1.**
 - je Meeting: fehlende Teilnehmer, Start- und Endzeit, besprochene Themen

- M3.1.**
 - je Meeting: fehlende Teilnehmer, Start- und Endzeit, besprochene Themen

- M2.1.**
 - auf ein Ticket verbuchter Aufwand

M2.1. dokumentiere alle Abweichungen (z.B. Sprintumfang, ...), die gegen erwartete Planungssicherheiten verstoßen

- M2.2. & M2.3.**
 - Liste aller committeten Tickets
 - je Team committete Story Points

- M2.4.**
 - geplanter Deploymenttermin

- M2.2. & M2.3.**
 - Liste aller im Sprint erledigten Tickets

- M2.4.**
 - Deploymenttermin dokumentieren

M3.2. dokumentiere schätzungsverbessernde Maßnahmen

- M1.2.**
 - Bewertung der umgesetzten Anforderungen

- M3.3.**
 - Code Qualität durch Reviewer bewertet

- M2.2.**
 - Anzahl pro Sprint gefundene Bugs

M2.2. dokumentiere alle Änderungen an Testmethoden um den Code qualitativ zu verbessern

- M2.1.**
 - In diesem Sprint zu Verbessertes (Ziele)

M2.1.- M2.2. dokumentiere alle Prozessveränderungen je Sprint

M2.1.- M2.2. dokumentiere alle Anstrengungen und Prozessveränderungen je langfristigem Ziel

M2.1.- M2.2. dokumentiere Prozessabweichungen

5 Fazit

Das Ziel meiner Bachelorarbeit war es, den NeMo-Prozess so messbar zu machen, dass es möglich ist, dessen Erfolg zu beurteilen. Dazu wurden für die relevanten Kriterien Maße entwickelt, die in dieser Bachelorarbeit dokumentiert sind.

Nach Erhebung der Maße sollte es möglich sein, den Erfolg des NeMo-Prozesses aus Sicht der NeMo-PMs zu beurteilen. Weiterhin wird für jedes Kriterium sichtbar, an welchen Stellen angesetzt werden kann, um den Prozess zu verbessern. Ein Großteil der Messungsergebnisse ist nur in Textform darstellbar und nutzt keine Skala. Dadurch können die Messergebnisse präziser beschrieben werden. Es ist anzumerken, dass in dieser Arbeit Einflüsse der Kriterien aufeinander außer Acht gelassen wurden. Diese Einflüsse sollte bei jeder Prozessverbesserung beachtet werden, ist für die Evaluierung aber zunächst nicht notwendig.

In der Bachelorarbeit habe ich mich konkret mit der Evaluierung bestimmter Software-Prozess-Güte-Kriterien auseinander gesetzt und diese für den NeMo-Prozess evaluierbar gemacht, damit kann der NeMo-Prozess mit anderen Prozessen verglichen werden oder Veränderungen im NeMo-Prozesses durch Trends von Kriterien sichtbar gemacht werden. Auch können diese Maße in abgewandelter Form bei weiteren Softwareentwicklungsprozessen angewendet werden.

Nach den Messungen ist für jedes Kriterium einzuschätzen, in welchen Aspekten der Prozess diesem Kriterium entspricht.

In den Bereichen Transparenz, Plausibilität und einem Aspekt der Effizienz (F1), liefern die Maße aus meiner Sicht sehr genaue Ergebnisse, die nah an der Realität liegen. Die Plausibilität kann durch eine Umfrage umfassend ermittelt werden. Die Genauigkeit der Ergebnisse hängt hier von der Bewertungsskala ab.

Die Maße für Softwarequalität, Verbesserungsfähigkeit und Planbarkeit sind ungenauer oder weniger umfassend und sind als grobe Einschätzung anzusehen, die nachfolgend genauer betrachtet werden sollte. Die Maße des Kriteriums Softwarequalität konzentrieren sich zwar nur auf die für das NeMo-Projekt wichtigsten Aspekte, können diese aber aufgrund der Komplexität nicht in ihrer Gänze erfassen. Alle drei Maße von Softwarequalität liefern daher eine Annäherung an die Realität auf Basis derer bereits eine grobe Einschätzung stattfinden kann. Das Kriterium Verbesserungsfähigkeit ist schwer zu bewerten. Viele Änderungen innerhalb kurzer Zeit sind nicht unbedingt besser als wenige Änderungen oder keine. Zu viele Veränderungen könnten den Prozessablauf stark stören und einen negativen Einfluss auf etwa die Effizienz haben. Ich denke, bei der Evaluierung der Verbesserungsfähigkeit ist darauf zu achten, dass bei allen Prozessbeteiligten ein gewisses Bewusstsein für den Prozess da ist und die Bereitschaft etwas auch selber zu verändern. Weiterhin sollten nach der Einführungsphase des Prozesses zumindest eine geringe Anzahl von Zielen definiert sein, auf die zugearbeitet wird. Die Maße des Kriteriums Planbarkeit sind sehr prozessbezogen und geben keine Möglichkeit darüber hinaus zu sehen. Dies ist für den NeMo-Prozess akzeptabel. Eine Verbesserungsmöglichkeit für den Prozess wäre hier auch langfristig planbar zu werden. In diesem Fall müssten weitere Maße ermittelt werden, die dies erfassen könnten.

Durch qualitative Interviews wurden nicht nur weitere Kriterien gefunden, allen

Prozessbeteiligten wurde die Möglichkeit gegeben, Feedback zum NeMo-Prozess zu geben. Dabei ergab sich ein überwiegend positives Meinungsbild, in dem Stärken und Schwächen aufgezeigt wurden. Von den dabei genannten Verbesserungswünschen sind erste umgesetzt worden.

Zur schlussendlichen Bewertung des Erfolges des NeMo-Prozesses, anhand der Messungen, erscheint es mir sinnvoll die Relevanz der Kriterien vor Messungsauswertung gegeneinander zu gewichten und Zusammenhänge und Einflüsse so zu notieren, dass diese in die Auswertung miteinbezogen werden. Einige der Maße, wie beispielsweise Effizienz, sind so konzipiert, dass sie Ansätze liefern sollen für mögliche Verbesserungsintentionen. Trotzdem liefern diese Maße auch Informationen, die in die Bewertung des Kriteriums miteinbezogen werden sollten.

Um den Erfolg des NeMo-Prozesses zu ermitteln, wäre der nächste Schritt, die Messungen durchzuführen. Soll dabei die GQM Methode verfolgt werden, sollte sich zuvor mit dieser vertraut gemacht werden und ein Analyseplan erstellt werden. Mit den Ergebnissen kann der Erfolg bewertet werden und der Prozess gegebenenfalls gezielt verbessert werden. Zu Beginn dieser Messung sollte für alle an den Messungen beteiligten eine Einführung stattfinden und Messungen sollten erklärt werden und probeweise ausgeführt werden.

Literaturverzeichnis

- [1] 12 principles behind the agile manifesto. <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>, Aug 2019.
- [2] Atlassian. Übersicht über jira: Produkte, projekte und hosting. <https://www.atlassian.com/de/software/jira/guides/getting-started/overview>.
- [3] Juyun Cho, Yongseog Kim, and David Olsen. A case study on the applicability and effectiveness of scrum software development in mission-critical and large-scale projects. *AMCIS 2006 Proceedings*, page 445, 2006.
- [4] Tsun Chow and Dac-Buu Cao. A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81(6):961–971, 2008.
- [5] Anton De Wit. Measurement of project success. *International journal of project management*, 6(3):164–170, 1988.
- [6] A César C França, Fabio QB da Silva, and Leila MR de Sousa Mariz. An empirical study on the relationship between the use of agile practices and the success of scrum projects. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, page 37. ACM, 2010.
- [7] Volker Gruhn. Validation and verification of software process models. In *European Symposium on Software Development Environments*, pages 271–286. Springer, 1991.
- [8] Yasir Hafeez, Asma Batool, Sohail Asghar, and Muhammad Jamal. Role of software requirements to improve the quality of scrum framework. *Science International-Lahore*, 26(1):165–168, 2014.
- [9] Watts S Humphrey. Characterizing the software process: a maturity framework. *IEEE software*, 5(2):73–79, 1988.
- [10] Lavagnon A Ika. Project success as a topic in project management journals. *Project Management Journal*, 40(4):6–19, 2009.
- [11] Rich C Lee. The success factors of running scrum: A qualitative perspective. *Journal of Software Engineering and Applications*, 5(06):367, 2012.
- [12] Subhas C Misra, Vinod Kumar, and Uma Kumar. Success factors of agile software development. *Software engineering research and practice*, 1:233–239, 2006.
- [13] Obs. Time in status. <https://marketplace.atlassian.com/apps/1211756/time-in-status?hosting=cloud&tab=overview>.
- [14] Ken Schwaber. Nexus guide. the definitive guide to scaling scrum with nexus: The rules of the game. 2018. verfügbar unter https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-01/2018-Nexus-Guide-English_0.pdf?nexus-file=https%3A%2F%2Fscrumorg-website-prod.s3.amazonaws.com%2Fdrupal%2F2018-01%2F2018-Nexus-Guide-English_0.pdf.

- [15] Syed Muhammad Ali Shah, Efi Papatheocharous, and Jaana Nyfjord. Measuring productivity in agile software development process: a scoping study. In *Proceedings of the 2015 International Conference on Software and System Process*, pages 102–106. ACM, 2015.
- [16] Jeff Sutherland and Ken Schwaber. Der scrum guide. *The definitive guide to scrum: The rules of the game*. Scrum.org, 268, 2013. verfügbar unter <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf>.
- [17] Rini van Solingen and Egon W Berghout. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. McGraw-Hill, 1999.
- [18] Jiangping Wan, Yahui Zhu, and Ming Zeng. Case study on critical success factors of running scrum. *Journal of Software Engineering and Applications*, 6(02):59, 2013.

A Literaturergebnisse

Kriterium	Ursprung	Quelle
Effizienz (Entwicklung)	Faktor für Projekterfolg (CSF)	[11], [8]
	Effekt von Scrum	[3]
	Prozess-Kriterium	[15]
Effizienz (Kosten)	Faktor für Projekterfolg (CSF)	[8]
	Effekt von Scrum	[3]
Flexibilität	Faktor für Projekterfolg (CSF)	[8]
	Prinzip / agiles Manifest	[1]
Fortbildend	Faktor für Projekterfolg (CSF)	[12]
	Prinzip / agiles Manifest	[1]
Planbarkeit	Faktor für Projekterfolg (CSF)	[12]
Planbarkeit (Kosten)	Kriterium zum Projekterfolg	[5], [10], [9]
	Faktor für Projekterfolg (CSF)	[11], [4]
Planbarkeit (Zeit)	Kriterium zum Projekterfolg	[10]
	Prozess-Kriterium	[9]
	Faktor für Projekterfolg (CSF)	[11], [4]
Produktqualität	Kriterium zum Projekterfolg	[10]
	Prinzip / agiles Manifest	[1]
	Prozess-Kriterium	[9]
	Faktor für Projekterfolg (CSF)	[11], [4]
	Effekt von Scrum	[3]
Transparenz	Faktor für Projekterfolg (CSF)	[8], [12], [4]
	Prinzip / agiles Manifest	[1]
	Faktor für Scrumerfolg	[3]
	Kriterium zum Projekterfolg	[11]

Tabelle 2: Darstellung aller während der Literaturrecherche ermittelten Kriterien, Quelle: Eigene Tabelle

B Interview Fragen

Art	Frage mit verschiedenen Formulierungen und Folgefragen
NeMo	1. Als wie erfolgreich beurteilst du den NeMo-Prozess? Was waren deine Gedanken dabei? An welche Eigenschaften hast du dabei gedacht?
NeMo	2. Was schätzt du an unserem NeMo-Prozess, was eher nicht? Was findest du an unserem Nemo-Prozess besser als bei anderen? Was fehlt dir an unserem Nemo-Prozess?
allg.	2. Was macht einen guten Prozess aus? Was bedeutet 'gut' für dich? Was für Eigenschaften fallen dir ein?
NeMo	3. Was für Prozess-Kriterien würdest du unserem Nemo-Prozess zuordnen? Was für Prozesskriterien fallen die sonst noch ein? Was kann der Nemo-Prozess? Wie kann der Nemo-Prozess sein?
allg.	3. Was für Eigenschaften/Charakteristiken würdest du einem Prozess zuordnen? Was kann ein Prozess sein?
NeMo	4. Was hat der NeMo-Prozess verändert? (zur vorherigen Arbeitsweise)
allg.	4. Was kann man mit einem Prozess erreichen/bewirken? Wozu führt man Prozesse ein? Warum nutzt man Prozesse?
allg.	5. Was für Unterschiede gibt es zwischen der Entwicklung mit und ohne Prozess? Worin unterscheidet sich beides? Worin nicht? Was für Vor- und, oder Nachteile bringt ein Prozess mit sich

Tabelle 3: Liste der Interviewfragen, aufgeteilt in auf den NeMo-Prozess bezogene Fragen und auf Prozess allgemein bezogene Fragen, Quelle: Eigene Tabelle

C Messpläne der Softwareprozess-Güte-Kriterien

C.1 Transparenz

Auflistung aller Maße des Softwareprozess-Güte-Kriteriums TRANSPARENZ	
F1.	Welches Prozesswissen ist verfügbar und wie ist es verteilt? M1.1. Qualität der Information Qualitätskriterien: Korrektheit, Vollständigkeit, Zugänglichkeit, Lesbarkeit Prozessexperten/einmalig/Skala: zufriedenstellend, ausreichend, lückenhaft, nicht nutzbar M1.2. Informiertheit der Prozess-Beteiligten Prozessbeteiligte/einmalig/Umfrage basierend auf C.1.1 und C.1.2
F2.	Mit welchen Mitteln schafft der Prozess Transparenz? M2.1. Transparenz schaffenden Meetings Prozessdokumentation/einmalig/dokumentiere als Liste: Meetingname, Agenda M2.2. Prozessartefakte, die Transparenz schaffen Prozessdokumentation/einmalig/dokumentiere als Liste: Artefaktbezeichnung, durch Artefakt verteilte Information
F3.	Wie sind Fortschritte und Abhängigkeiten dokumentiert? Wie umständlich ist es Fortschritte abzulesen? M3.1. Aktualität und Vollständigkeit des Ticketfortschritts im Ticket Kriterien: Status, Abhängigkeiten, Probleme, Entscheidungen NeMo-PM, Kunde, Entwickler(am Ticket beteiligt)/alle umgesetzten Tickets im Messzeitraum/Skala: zufriedenstellend, ausreichend, lückenhaft, unbrauchbar M3.2. Aktualität und Vollständigkeit des Fortschritts eines Features im Epic Kriterien: Status, Abhängigkeiten, Probleme, Entscheidungen am Feature beteiligte: NeMo-PM, Kunde, Entwickler/alle fertiggestellten Features im Messzeitraum/Skala: zufriedenstellend, ausreichend, lückenhaft, unbrauchbar M3.3. Aktualität und Vollständigkeit des Sprintfortschritts im Sprintboard Kriterien: Status, Abhängigkeiten, Probleme, Entscheidungen am Sprint beteiligte: NeMo-PM, Kunde, Entwickler/alle Sprints im Messzeitraum/Skala: zufriedenstellend, ausreichend, lückenhaft, unbrauchbar
F4.	Welche Informationen fehlen während eines Sprints? Fehlen regelmäßig Informationen während der Sprints? M4.1. fehlende Informationen Prozessbeteiligte/gesamter Messzeitraum/Beschreibung, Auswirkung, Workaround

Tabelle 4: Messplan des Softwareprozess-Güte-Kriteriums Transparenz, Quelle: Eigene Tabelle

C.1.1 Beteiligte nach Prozessphasen

	Priorisierung	Konzept	Refinement	Planung	Umsetzung	Testen	Deployment
Architekt		x	x	x	x		
Entwickler			x	x	x	x	x
Konzepter	x	x	x	x			
Kunde	x	x		x			x
NeMo-PM	x	x	x	x	x	x	x
Teamlead			x	x	x		
QS		x	x	x	x	x	x

Tabelle 5: Darstellung der Beteiligungen nach einzelnen Prozessphasen (oberste Zeile: Prozessphasen, linkeste Spalte: Prozessrollen), , Quelle: Eigene Tabelle

C.1.2 Liste der notwendigen Informationen je Prozessphase

Die folgende Liste enthält alle Informationen die den in der Phase Beteiligten Bekannt sein sollten. Jeder der im NeMo-Prozess arbeitet soll wissen wo das Tickethandling dokumentiert ist oder es gemäß dieser Dokumentation umsetzen können. Weiterhin ist für jedes Meeting und jeden Call für alle Teilnehmer zu wissen, wer die anderen Teilnehmer sind, wann und wo das Meeting stattfindet, was auf der Agenda steht und auch warum die Person selber im Meeting anwesend sein soll.

- Priorisierung
 - Termin an dem der nächsten kommende Sprint startet
 - Backlog im
- Konzept
 - Backlog im JIRA des Kunden
 - Status und Concept Call
 - Verantwortlicher bei Rückfragen zum Design
- Refinement
 - Konzept-Doku (Production Guide)
 - Architekten Board
 - Meeting: Architektenrunde
 - Architektur des Softwareproduktes (nur Architekt)
 - Refinement Board
 - Meeting: Backlogrefinemnts der Teams
 - Architektur des Softwareproduktes (nur Architekt)
- Planning

C. Messpläne der Softwareprozess-Güte-Kriterien

- Board des zu planenden Sprints
- Sprintplanning Call
- Meeting: Planning und Commitment
- Sprinttermine des Kommenden Sprints: Beginn, Ende, Feature Freeze, Deployment
- Umsetzung
 - Meeting: Kickoff
 - Board des aktuellen Sprints im Firmen JIRA
 - wiederkehrendes Meeting: Daily
 - wiederkehrendes Meeting: Weekly
 - Sprinttermine des aktuellen Sprints: Beginn, Ende, Feature Freeze, Deployment
 - Status und Concept Call
- Testing
 - Qualitätssicherungsboard
 - Sprinttermine des aktuellen Sprints: Beginn, Ende, Feature Freeze, Deployment
 - Board des aktuellen Sprints im JIRA des Kunden
- Deployment und Abnahme
 - Deploymenttermine
 - Sprinttermine der folgenden Sprints: Beginn, Ende, Feature Freeze, Deployment
 - Board des aktuellen Sprints im JIRA des Kunden
 - Review Call
 - Livegangscall

C.2 Effizienz

Auflistung aller Maße des Softwareprozess-Güte-Kriteriums EFFIZIENZ	
F1.	<p>Wie schnell kann eine Anforderung umgesetzt werden?</p> <p>Wie viel Zeit verbringt ein Ticket in welchem Status? Welche Phasen werden öfter als einmal durchlaufen?</p> <p>M1.1. Dauer die benötigt wird um eine Anforderung umzusetzen Priorisierung der Tickets beachten <i>Beliebig/während des Sprintzeitraums umgesetzte Tickets/über JIRA ermitteln</i></p> <p>M1.2. Umsetzungsdauer nach Prozessphasen aufgeteilt <i>Beliebig/während des Sprintzeitraums umgesetzte Tickets/über JIRA ermitteln, Phasen wie in der im Anhang unter C.1.1 zu findenen Tabelle</i></p> <p>M1.3. Häufigkeit in der jede Prozessphase durchlaufen wurde <i>Beliebig/während des Sprintzeitraums umgesetzte Tickets/über JIRA ermitteln, Phasen analog zu Phasen wie in der im Anhang unter C.1.1 zu findenen Tabelle</i></p>
F2.	<p>Wie viel Zeit steht einem Entwickler zum Programmieren zur Verfügung?</p> <p>Ist der Prozessaufwand gegen den Entwicklungsaufwand gerechtfertigt? Wie effizient kann die gegebene Entwicklungszeit genutzt werden?</p> <p>M2.1. für den Prozess aufzuwendende Prozesszeit <i>Beliebig/einmalig/gesamte Dauer aller Prozessmeetings eines Sprints</i></p> <p>M2.2. Zeitabstände zwischen Meetings <i>Beliebig/einmalig/Liste mit durch Prozessmeetings benötigte erneute Einarbeitungen</i></p> <p>M2.3. Unterbrechungen beim Enwtickeln während dafür vorgesehener Zeit <i>Entwickler/Messzeitraum/Protokoll: Dauer, Grund der Unterbrechung</i></p>
F3.	<p>Wie effizient sind Meetings und Calls?</p> <p>Sind Teilnehmer, Dauer und Agenda der Meetings passend angesetzt?</p> <p>M3.1. Abweichungen von angesetzten Meetingzeiten <i>Beliebig/während des Messzeitraums stattfindende Meetings/dokumentiere geplante und reelle Start- und Endzeit</i></p> <p>M3.2. im Meeting fehlende Teilnehmer <i>NeMo-PM/während des Messzeitraums stattfindende Meetings/dokumentiere, wenn durch fehlende Teilnehmer Meetings behindert wird</i></p> <p>M3.3. im Meeting besprochene Themen <i>NeMo-PM/während des Messzeitraums stattfindende Meetings/dokumentiere Anzahl besprochener Agenda-Themen, Anzahl nicht besprochener Agenda-Themen, Anzahl besprochener Themen die nicht zur Agenda zählen</i></p>

Tabelle 6: Messplan des Softwareprozess-Güte-Kriteriums Effizienz, Quelle: Eigene Tabelle

C.3 Planbarkeit

Auflistung aller Maße des Softwareprozess-Güte-Kriteriums PLANBAR
<p>F1. Welche Planungssicherheiten werden erwartet und werden sie eingehalten? In welchen Schritten und bei welchen Artefakten ist der Prozess planungssicher?</p> <p>M1.1. vom Prozess vorgesehene Planungssicherheiten <i>Prozessexperten/einmalig/Liste mit Beschreibung, Zeitpunkt(ab wann) und Artefakt(was)</i></p> <p>M1.2. Anzahl Änderungen die in M1.1. Ermitteltem widersprechen <i>Beliebig/den Messzeitraum über/Protokollierte Veränderungen</i></p>
<p>F2. Wie akkurat sind Vorhersagen?</p> <p>M2.1. Abweichung der geschätzten StoryPoints zu realem Aufwand je Ticket <i>Beliebig/im Messzeitraum umgesetzte Tickets/vergleiche Story Points und gebuchten Aufwand</i></p> <p>M2.2. Abweichungen vom committeten Sprintumfang je Sprint in Story Points <i>Beliebig/Sprints innerhalb des Messzeitraums/vergleiche comittete Story Points mit umgesetzten StoryPoints</i></p> <p>M2.3. Abweichungen von der geschätzten Team-Kapazität je Sprint in StoryPoints <i>Beliebig/Sprints innerhalb des Messzeitraums/vergleiche comittete Story Points mit umgesetzten StoryPoints</i></p> <p>M2.4. Abweichungen vom geplanten Deployment-Datum <i>Beliebig/Sprints innerhalb des Messzeitraums/vergleiche angesetztes Deployment-Datum mit reelem Deployment-Datum</i></p>
<p>F3. Womit wird die Qualität der Schätzungen im Prozess gefördert? Sind alle nötigen Informationen zur gegebenen Zeit für die Schätzung verfügbar? Werden Schätzungen nachbereitet, so dass Beteiligte aus ihnen lernen können?</p> <p>M3.1. vorgesehene Prozessaufgaben die Schätzungen verbessern <i>Prozessexperte/einmalig/Liste inklusive Prozessaufgabe, Auswirkung auf die Schätzung</i></p>

Tabelle 7: Messplan des Softwareprozess-Güte-Kriteriums Planbarkeit, Quelle: Eigene Tabelle

C.4 Softwarequalität

Auflistung aller Maße des Softwareprozess-Güte-Kriteriums SOFTWAREQUALITÄT	
F1.	Wie weit entspricht die Umsetzung den Anforderungen und welche Einfluss hat der Prozess darauf?
M1.1.	Prozessesteile, welche Anforderungsgleichheit begünstigen <i>Prozessexperten/einmalig/Liste mit Beschreibung, Zeitpunkt, Beteiligten</i>
M1.2.	Bewertung der umgesetzten Anforderungen <i>Beliebig/für alle im Messzeitraum umgesetzten Tickets/erfüllte Anforderungen pro Ticket, durch Subtasks ablesen</i>
F2.	Wie weit kann der Prozess die Fehlerfreiheit des Codes beeinflussen?
M2.1.	Prozessregeln und -aufgaben um Fehler im Code zu reduzieren <i>Prozessexperten/einmalig/Liste mit Beschreibung, Zeitpunkt, Beteiligten</i>
M2.2.	Fehler pro Ticket <i>Beliebig/für alle im Messzeitraum umgesetzten Tickets/nur als wichtig eingestufte Fehler beachten, aus den Subtasks ablesbar</i>
F3.	Wie gut strukturierten Code bringt der Prozess hervor?
M3.1.	angewendete Methoden für verbesserte Code Qualität <i>Entwickler(am Ticket beteiligt)/alle umgesetzten Tickets im Messzeitraum/Liste aller angewendeten Methoden mit Bezeichnung, Beschreibung je Ticket</i>
M3.2.	Prozessregeln und -aufgaben um die Codequalität zu verbessern <i>Prozessexperte/einmalig/Liste mit Beschreibung, Zeitpunkt, Beteiligten</i>
M3.3.	Code Qualität <i>Entwickler(nicht am Ticket beteiligt)/alle umgesetzten Tickets im Messzeitraum/Bewertung durch Reviewer, Skala:[1-perfekt, 2-besser als der Standard, 3-dem Standard entsprechend, 4-knapp ausreichend, 5-zu Teilen ungenügend, 6-ungenügend]</i>

Tabelle 8: Messplan des Softwareprozess-Güte-Kriteriums Softwarequalität, Quelle: Eigene Tabelle

C.5 Verbesserungsfähigkeit

Auflistung aller Maße des Softwareprozess-Güte-Kriteriums VERBESSERND	
F1.	<p>Wie viel wird sich im Prozess mit dem Prozess auseinandergesetzt?</p> <p>Wie regelmäßig wird sich mit Prozessverbesserungen beschäftigt? Welcher Anteil der Prozessbeteiligten ist motiviert den Prozess zu verbessern?</p> <p>M1.1. Prozessmeetings die sich mit Prozessverbesserung beschäftigen <i>Prozessexperte/einmalig/Meetingbezeichnung, Teilnehmer, Prozessverbessernde Themen der Agenda</i></p> <p>M1.2. Prozessaufgaben die sich mit Prozessverbesserung beschäftigen <i>Prozessexperte/einmalig/Zeitpunkt, Beteiligte, Prozessverbessernde mögliche Auswirkungen</i></p> <p>M1.3. sich mit Prozessverbesserung beschäftigende Personen <i>Prozessbeteiligte/einmalig/Umfrage: Beschreibung der letzten Prozessverbesserung mit Selbstbeteiligung, Regelmäßigkeit, aufgewendete Zeit</i></p>
F2.	<p>Wie erfolgreich werden Probleme gelöst?</p> <p>Wie oft wird versucht den Prozess kurzfristig zu verbessern?</p> <p>M2.1. Anzahl gelöster Probleme <i>Beliebig/über den gesamten Messzeitraum hinweg/dokumentiere: Veränderung, Problem, Auswirkung der Veränderung</i></p> <p>M2.2. Prozessveränderungen pro Sprint <i>Beliebig/im Messzeitraum stattfindende Sprints/dokumentiere: Veränderung, Auswirkung der Veränderung, Motivation der Veränderung</i></p>
F3.	<p>Wie verbessert sich der Prozess langfristig?</p> <p>Gibt es positive Trends in anderen Prozess-Eigenschaften?</p> <p>M3.1. Anzahl gesetzter Ziele <i>NeMo-PM/einmalig/Liste aller langfristigen Ziele, inklusive Beschreibung</i></p> <p>M3.2. Anzahl erreichter Ziele <i>NeMo-PM/einmalig/Überprüfe Liste aller langfristigen Ziele, inklusive Beschreibung</i></p> <p>M3.3. unternommene Anstrengung um Ziel zu erreichen <i>Beliebig/gesamter Messzeitraum je Ziel/dokumentiere Aktivitäten, Statusbesprechungen zum Ziel</i></p> <p>M3.4. Trends in anderen Güte-Kriterien des Prozesses <i>Beliebig/gesamter Messzeitraum/dokumentiere Trends der anderen Kriterien</i></p>

Tabelle 9: Messplan des Softwareprozess-Güte-Kriteriums Verbesserungsfähigkeit, Quelle: Eigene Tabelle

C.6 Plausibilität

Auflistung aller Maße des Softwareprozess-Güte-Kriteriums PLAUSIBILITÄT	
F1.	Als wie plausibel wird der Prozess wahrgenommen?
M1.1.	Nachvollziehbarkeit des gesamten Prozesses <i>Prozessbeteiligte/einmalig/Umfrage: Skala [einleuchtend/intuitiv, mit Erklärung nachvollziehbar, zu Teilen einleuchtend, zu Teilen nachvollziehbar, nicht nachvollziehbar, erscheint sinnlos]</i>
M1.2.	Nachvollziehbarkeit der einzelnen Prozessphasen <i>Prozessbeteiligte/einmalig/Umfrage: Skala [einleuchtend/intuitiv, mit Erklärung nachvollziehbar, zu Teilen einleuchtend, zu Teilen nachvollziehbar, nicht nachvollziehbar, erscheint sinnlos], Phasen nach C.1.1</i>
M1.3.	Anzahl Beschwerden über den Prozessphasen <i>Prozessbeteiligte/einmalig/Umfrage</i>
F2.	Wie genau wird sich an den Prozess gehalten?
M2.1.	Abweichungen vom Prozess <i>Prozessbeteiligte/einmalig und den Messzeitraum über/Umfrage und Protokoll: Liste mit Beschreibung, Zeitpunkt, Beteiligten, Regelmäßigkeit</i>

Tabelle 10: Messplan des Softwareprozess-Güte-Kriteriums Plausibilität, Quelle: Eigene Tabelle