



# Process Mining bei hybriden Vorgehensmodellen zur Umsetzung von Unternehmenssoftware

Beitrag für das Informatik Spektrum, 2021, Sonderheft „Innovation in der Software Entwicklung“

Sascha Alpers<sup>1</sup> · Thomas Karle<sup>2</sup> · Clemens Schreiber<sup>3</sup> · Frank Schönthaler<sup>3</sup> · Andreas Oberweis<sup>1,3</sup>

Angenommen: 13. April 2021 / Online publiziert: 12. Mai 2021  
© Der/die Autor(en) 2021

## Zusammenfassung

Process Mining hat sich in den vergangenen Jahren zur Analyse von Prozessdaten etabliert und wird in verschiedenen Kontexten, wie beispielsweise Industrie 4.0, eingesetzt. Die Potenziale dieser Technologie liegen jedoch nicht nur in der Analyse von Wertschöpfungsprozessen im Kontext von Produktion und Verwaltung. Die Technologie kann darüber hinaus auch für die Verbesserung des Vorgehens bei den dazugehörigen – meist großen und komplexen – Softwareprojekten genutzt werden. Für die Umsetzung der für Industrie-4.0-Prozesse erforderlichen Unternehmenssoftware werden häufig agile oder hybride Vorgehensmodelle eingesetzt. Software Innovations unterstützen die Umsetzung in zweierlei Hinsicht. Software Innovations bezeichnen zum einen Innovationen für das Software Engineering durch neue Vorgehensmodelle, Methoden und Werkzeuge. Zum anderen umfasst der Begriff Innovationen, welche durch (neuartige) Software ermöglicht werden. Ausgehend von diesen beiden Aspekten von Software Innovations beschreibt der Beitrag, wie Process Mining zur Analyse und Verbesserung des Vorgehens bei Unternehmenssoftwareprojekten verwendet werden kann. Der Schwerpunkt liegt auf der Anwendung und Verbesserung von hybrid durchgeführten Entwicklungs- und Konfigurationsprojekten. Das vorgestellte Verfahren kann jedoch auch für klassisch oder agil durchgeführte Projekte angewendet werden. Es dient generell dazu, den Prozess zur Softwareerstellung kontinuierlich, anhand von Erkenntnissen aus laufenden und abgeschlossenen Projekten, automatisiert zu analysieren und zu verbessern. Hierzu wird exemplarisch der hybride Referenzprozess aus dem Vorgehensmodell eines mittelständischen Software- und Beratungsunternehmens als Anwendungsfall betrachtet.

---

Sascha Alpers  
alpers@fzi.de

Thomas Karle  
thomas.karle@promatis.de

✉ Clemens Schreiber  
clemens.schreiber@kit.edu

Frank Schönthaler  
frank.schoenthaler@promatis.de

Andreas Oberweis  
andreas.oberweis@kit.edu

<sup>1</sup> FZI Forschungszentrum Informatik, Haid-und-Neu-Straße 10–14, 76131 Karlsruhe, Deutschland

<sup>2</sup> PROMATIS software GmbH, Pforzheimer Straße 160, 76275 Ettlingen, Deutschland

<sup>3</sup> Karlsruher Institut für Technologie, Kaiserstraße 12, 76131 Karlsruhe, Deutschland

## Einleitung

Bei der Softwareerstellung als Teil des Software Engineering hat in den letzten Jahrzehnten die Methodenvielfalt deutlich zugenommen. Das in den 1980er-Jahren entwickelte Cleanroom-Software-Engineering [1], welches formale Methoden wie die Verifikation auf Basis von Modellen mit iterativen Entwicklungszyklen und statistischen Testverfahren verbindet, findet aufgrund von Kosten- und Zeitrestriktionen kaum noch Anwendung. Stattdessen sind agile Methoden gegenwärtig. Auch im Bereich des Projektmanagements sind agilere Methoden entstanden. Die Vielzahl an Methoden hat jedoch auch dazu geführt, dass es schwieriger wurde zu identifizieren, welche Faktoren für erfolgreiche Softwareprojekte entscheidend sind. Das Ermitteln von Faktoren, die zu einem Projekterfolg oder -misserfolg führen, ist besonders wichtig für Projekte zur Implementie-

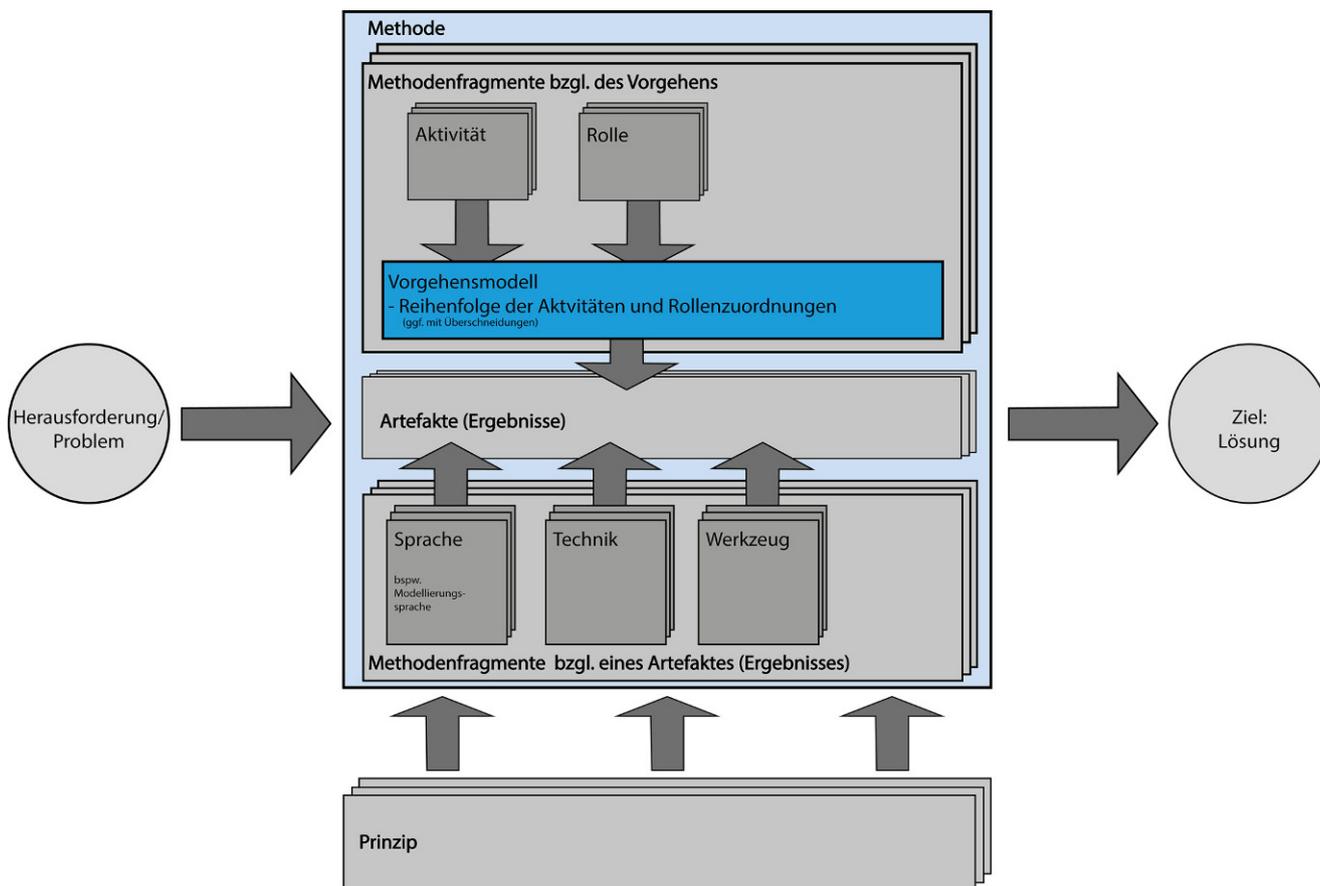


Abb. 1 Zusammenhang Vorgehensmodell, Methode und Prozess

nung von Unternehmenssoftwarelösungen, da es sich hier in der Regel um sehr komplexe und unternehmenskritische Anwendungssysteme handelt, deren Umsetzungen nahezu immer große Risiken mit sich bringen.

Da bei der Dokumentation von Vorgehensmodellen die Beschreibung der entsprechenden Prozesse ein zentraler Aspekt ist (vergleiche Abb. 1), bietet es sich an, das Vorgehen durch Prozessmodelle zu beschreiben. Aus diesem Grund hat die PROMATIS software GmbH ihr hauseigenes Vorgehensmodell IQPM (Integrated Quality Procedure Model) mit einer Petri-Netz-basierten Notation dokumentiert [2]. PROMATIS ist ein Software- und Beratungshaus, das seit mehr als 20 Jahren Produkte und Dienstleistungen im Geschäftsprozess- und Unternehmenssoftwareumfeld anbietet. Das konkrete gelebte Vorgehen bei durchgeführten Projekten, welches sich an den Prozessen des Vorgehensmodells orientiert, kann mithilfe von Process Mining untersucht werden, um Erkenntnisse zu gewinnen, wie dieses Vorgehen ggf. verbessert werden kann. Beispielsweise können unterschiedliche Schwerpunkte und Intensitäten in verschiedenen Projekten bei den einzelnen Aktivitätstypen wie dem Requirements Engineering, dem Design, der Konfiguration und Entwicklung, der Testfallerstellung und der

Dokumentation dazu führen, dass je nach Konstellation die Projektergebnisse besser oder schlechter ausfallen. Für die Identifikation solcher Konstellationen und die Ableitung der richtigen Verbesserungsmaßnahmen für das Vorgehensmodell selbst oder seine Umsetzung kann Process Mining als Unterstützung eingesetzt werden.

Die aktuell im Rahmen von vielen Softwareerstellungprojekten eingesetzten Werkzeuge, wie die Versionsverwaltung Git und die Projekt- und Aufgabenverwaltung Jira, können eine Datenbasis zur Rekonstruktion des Erstellungsprozesses liefern. Dazu werden aus den Werkzeugen detaillierte Ereignisprotokolle (Event-Logs) extrahiert und anschließend mithilfe von Process-Mining-Verfahren analysiert. Dabei kann in einem ersten Schritt betrachtet werden, ob der tatsächlich gelebte Prozess dem definierten Prozess des Vorgehensmodells entspricht. Weiterhin können neben dem Prozessablauf auch die Dauer der einzelnen Aktivitäten und die Zusammenarbeit der Projektmitglieder untersucht werden. Liegen diese Daten von mehreren Entwicklungsprojekten vor und gibt es zusätzlich Informationen bezüglich der Qualität der Ergebnisse der verschiedenen Projekte, können daraus Schlüsse für die Weiterentwicklung von Best Practises abgeleitet werden. Beispielsweise lassen

**Tab. 1** Übersicht über veröffentlichte Anwendungsfälle für Process Mining in Softwareprojekten

Artikel	Anwendungsfall	PM Verfahren	Werkzeug	Datenquelle
[6]	Fünf Teilprojekte zur Weiterentwicklung der Open-Source-Software ArgoUML	Discovery, Conformance, Enhancement	ProM	Konfigurationsmanagementsystem (Subversion)
[7]	Fehlerbearbeitungsprozess bei der Entwicklung einer kommerziellen Software	Discovery	ProM	Konfigurationsmanagementsystem
[8]	2000 kommerzielle Softwareentwicklungsprojekte über 5 Jahre	Discovery, Conformance	ProM	Internes Projektüberwachungssystem
[9]	Fehlerbearbeitungsprozesse und Aufgabenmanagement in der Open-Source-Softwareentwicklung	Discovery, Enhancement	ProM	Softwarerepositorium (SourceForge), Bugtracker, Mailarchiv
[10]	Fehlerbearbeitungsprozesse und Aufgabenmanagement bei Softwareprojekten in der Lehre	Discovery, Conformance, Enhancement	Disco	Bugtracker, Kommunikationsplattform, Versionskontrollsystem
[11, 12]	Weiterentwicklung und Pflege eines Computerservierungssystems in der Tourismusindustrie	Discovery, Enhancement	ProM, Disco	Client/Server Kommunikation
[13]	Fehlerbehebungsprozess bei mehreren kleinen Softwareentwicklungsprojekten	Discovery, Enhancement	Disco	Bugtracker
[14]	Fehlerbearbeitungsprozesse und Aufgabenmanagement bei 9 Sprints innerhalb von 4 Wochen	Discovery, Enhancement	ProM, Disco	Softwareprojektmanagementplattform (Microsoft TFS), Bugtracker
[15]	Aufgabenmanagement und Codeanpassungen bei 2 agilen Softwareentwicklungsprojekten	Discovery, Enhancement	ProM, Disco	Softwareprojektmanagementplattform (Jira)
[16]	Aufgabenmanagement bei Softwareprojekten in der Lehre	Discovery, Conformance, Enhancement	ProM	Integrated Development Environment (Eclipse)
[17]	Aufgabenmanagement und Codeanpassungen bei Softwareprojekten in der Lehre	Discovery, Enhancement	Disco, Celonis	Softwareprojektmanagementplattform (GitLab)

sich Muster ableiten, die eine Projektdurchführung positiv beeinflussen können, oder Abläufe identifizieren, die vermehrt zu problematischen Entwicklungen führen.

Im nachfolgenden Abschnitt wird der Stand der Forschung und Praxis beschrieben. Dabei werden verschiedene Anwendungsfälle betrachtet, bei denen unterschiedliche Process-Mining-Verfahren und Datenquellen zum Einsatz kommen. Im dritten und vierten Abschnitt werden ein hybrides Vorgehensmodell aus der Praxis zur Implementierung einer Unternehmenssoftware betrachtet und Möglichkeiten zur Einbindung von Process-Mining-Verfahren erläutert. Im fünften Abschnitt wird der vorgeschlagene Ansatz an einem Anwendungsbeispiel verdeutlicht. Es folgt ein Fazit mit einem Ausblick.

## State of the Art

Der Einsatz von Prozessdaten zur Analyse von Abläufen zur Softwareerstellung wurde zum ersten Mal 1995 von Cook und Wolf vorgeschlagen [3]. Cook und Wolf schlugen zunächst Verfahren aus dem Bereich des maschinellen Lernens vor, um mithilfe von Prozessdaten den Ablauf der Softwareentwicklung zu modellieren. Drei Jahre später erschien die erste Fallstudie zur Anwendung der vorgeschlagenen Verfahren [4]. Aber erst durch die Weiterentwicklung von Process-Mining-Algorithmen und -Werkzeugen wurde die Möglichkeit der Analyse von Prozessen zur Softwareer-

stellung einfacher und vielseitiger. Neben der Modellierung des ausgeführten Ablaufs im Rahmen der Softwareerstellung (Process Discovery), kann der durchgeführte Prozess in Bezug auf den geplanten Ablauf überprüft (Conformance Checking) und die Prozessperspektive, z. B. durch die Betrachtung des zeitlichen Ablaufs und dem Einsatz von Ressourcen erweitert werden (Process Enhancement) [5]. In Tab. 1 werden verschiedene Anwendungsbeispiele aufgeführt, die sich jeweils in Bezug auf die gewählten Process-Mining-Verfahren, die verwendeten Werkzeuge und die Datenquellen unterscheiden. Da es sich bei den Beispielen um Forschungsprojekte handelt, wird für die Analyse der Prozessdaten am häufigsten auf die Open-Source-Software ProM<sup>1</sup> zurückgegriffen. Ein weiteres häufig verwendetes Werkzeug ist Disco,<sup>2</sup> bei dem es sich um eine kommerzielle Software handelt, die vor allem Vorteile durch eine einfach zu bedienende Nutzeroberfläche aufweist. Jedoch ist die Funktionalität von Disco gegenüber ProM deutlich eingeschränkter und ermöglicht beispielsweise kein Conformance Checking. Allerdings stellen Process Discovery und Process Enhancement auch die am häufigsten angewandten Verfahren bei den bisherigen Forschungsarbeiten dar.

Damit Process-Mining-Verfahren überhaupt eingesetzt werden können, ist die hinreichende Verfügbarkeit von Pro-

<sup>1</sup> [www.promtools.org](http://www.promtools.org).

<sup>2</sup> [www.fluxicon.com/disco](http://www.fluxicon.com/disco).

zessdaten eine grundlegende Voraussetzung. Bei den aufgeführten Anwendungsbeispielen in Tab. 1 werden eine Vielzahl an Datenquellen vorgeschlagen und getestet. Projektmanagementwerkzeuge und Software Repositorien dienen häufig als wichtigste Datenquelle, dazu zählen beispielsweise Subversion [6], SourceForge [9], Microsoft TFS [14], Jira Software [15] und GitLab [17]. Generell ist aber auch die Verwendung mehrerer verschiedener Datenquellen möglich (siehe Tab. 1). Die Wahl der Datenquelle bestimmt auch, ob ein Softwareprojekt als kompletter Prozess betrachtet werden kann oder nur einzelne Teilprozesse. Einige Anwendungsfälle beziehen sich nur auf Teilprozesse der Softwareentwicklung, wie das Fehlermanagement [7, 13] und die Softwarenutzung [11, 12]. Häufig ist auch eine manuelle Anpassung der Beschriftung der Aktivitäten notwendig, um die Bezeichnungen zu vereinheitlichen und wiederkehrende Abläufe erkennen zu können. In diesem Zusammenhang, bietet beispielsweise das Framework for Analysing Software Repositories (FRASER) [9] Unterstützung bei der Auswahl von Datenquellen und der Zusammenführung von Daten.<sup>3</sup>

Neben der klassischen linearen Softwareerstellung, bei der die Prozesse stark strukturiert sind, gibt es bereits viele Fallstudien, die Process Mining auch bei einem agilen Vorgehen einsetzen [10, 11, 14–16]. Eine Übersicht zu Forschungsartikeln zum Thema Process Mining in agilen Softwareprojekten gibt [18]. In Bezug auf die agile Softwareerstellung können zum Beispiel mithilfe von Process Mining die Scrum-Rollen identifiziert, die Priorisierung und Zuordnung der ausstehenden Aufgaben vorgenommen und die Zusammenarbeit der Teammitglieder analysiert werden [15]. Eine weitere Möglichkeit besteht darin, den Grad der Agilität eines Projektes zu bewerten [14]. Dennoch existiert auch eine Vielzahl an Herausforderungen, die insbesondere auf die Analyse agiler Softwareprojekte zutreffen. (1) Eine wesentliche Herausforderung ist die Datenverfügbarkeit und Nachverfolgung der Implementierungsschritte. Dies liegt vor allem daran, dass die Teammitglieder kontinuierlich ihr Vorgehen beschreiben und dokumentieren müssen. Eine Alternative stellt die automatische Nachverfolgung von Codeänderungen oder Zwischenstandsänderungen dar. (2) Doch auch wenn es für die Dokumentation von Softwareprojekten eine große Anzahl an Unterstützungswerkzeugen gibt, bleibt weiterhin das Problem, dass agile Prozesse zur Softwareerstellung wenig strukturiert sind und sich die Ausführungen der Entwicklungsprozesse sehr stark voneinander unterscheiden können. Da-

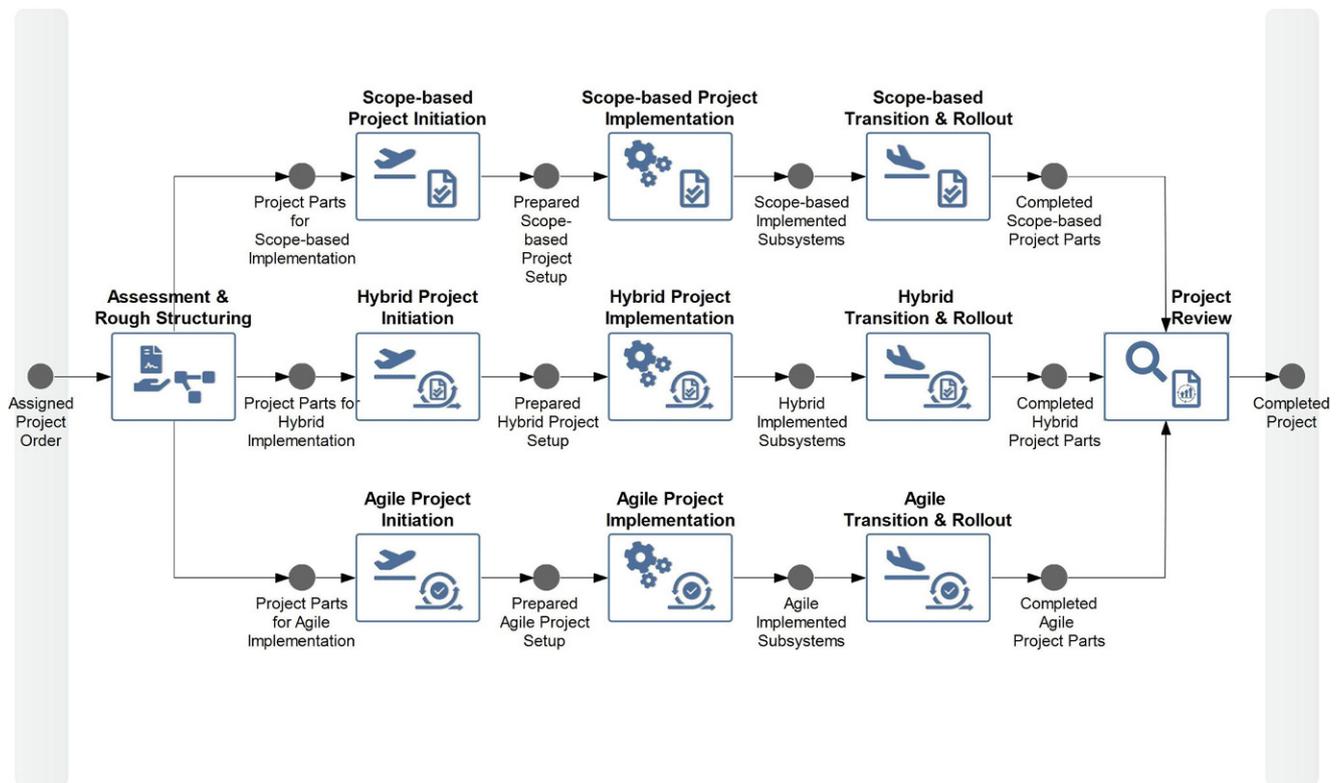
<sup>3</sup> Die Extraktion von Daten aus Software Repositorien stellt wiederum einen eigenen Forschungsbereich dar, bei dem es allgemein um die Analyse von Softwareentwicklungsprozessen geht. Hier sei auf die International Conference on Mining Software Repositories (<https://www.msconf.org/>) verwiesen, die als Teilkonferenz der ICSE regelmäßig stattfindet.

durch wird vor allem ein Vergleich der Prozesse erschwert. (3) Eine weitere Herausforderung ist die Zuordnung von Aktivitäten, wie beispielsweise Codeänderungen, zu den übergeordneten Teilprozessen. Die hybride Softwareerstellung, bei der lineare und agile Vorgehensweisen miteinander verknüpft werden, bietet für die genannten Herausforderungen mögliche Lösungsansätze. Im folgenden Abschnitt wird die hybride Softwareentwicklung aus Praxis-sicht betrachtet. Insbesondere wird beschrieben, wie Unternehmenssoftwareprojekte mithilfe eines hybriden Ansatzes umgesetzt werden können und wie Process Mining dabei zur Qualitätssicherung beitragen kann.

## Vorgehensmodelle für Unternehmenssoftware

Für die Durchführung von Softwareentwicklungsprojekten existiert eine Vielzahl von Methoden, Vorgehensmodellen und Prozessen (Prozess im Sinne von Anordnung der Aktivitäten in einer bestimmten für eine Vielzahl von Projekten empfohlenen Reihenfolge). Der Zusammenhang wird in Abb. 1 beschrieben, besonders hervorgehoben ist das Vorgehensmodell, dessen kontinuierliche Verbesserung im Fokus des Beitrags steht. Bei der Implementierung von Unternehmenssoftware steht man derzeit vor vielen neuen Herausforderungen und Konflikten, die es in solchen Projekten zu meistern und aufzulösen gilt [19]. Einerseits geht auch hier der Trend beim Vorgehen in der Umsetzung hin zum Einsatz agiler Methoden. Andererseits wird jedoch auch eine finanzielle Planungssicherheit gefordert und – zumindest bei einzelnen Teilkomponenten einer Unternehmenssoftware – ein gegebener Scope an Funktionalität, der umgesetzt sein muss, bevor ein System dann in Betrieb genommen werden kann. Dies ist vor allem in Bereichen wie Finanzwesen oder Supply-Chain-Management der Fall. Hier müssen vor einer Inbetriebnahme viele entsprechende Detailfunktionalitäten konform zu gegebenen gesetzlichen Anforderungen implementiert und getestet sein. Bei diesen Teilbereichen eines Unternehmenssoftwareprojektes ist in vielen Fällen nach wie vor ein Scope-basierter Ansatz erforderlich, da der Scope hier in weiten Teilen nicht flexibel anpassbar ist. Hier werden oft weiterhin klassische Vorgehensmodelle eingesetzt, d. h. eine phasenorientierte und vorwiegend lineare Vorgehensweise gewählt, wie beispielsweise das Wasserfallmodell oder das V-Modell. Bei Unternehmenssoftwareprojekten in den zuvor genannten Bereichen kommen häufiger auch nichtsequentielle Weiterentwicklungen zum Einsatz, wie beispielsweise das Spiralmodell, das zu den iterativen Modellen gehört.

Für andere Bereiche, die ebenfalls durch die Implementierung entsprechender Unternehmenssoftwarekomponenten abgedeckt werden sollen, können die Vorteile



**Abb. 2** High-Level-Übersicht des Vorgehensmodells IQPM™ von PROMATIS

durch den Einsatz agiler Methoden deutlich besser genutzt werden. Dies sind beispielsweise Komponenten wie ein zu implementierender Webshop. Dieser muss zwar auch Grundfunktionalitäten bereitstellen und entsprechenden gesetzlichen Anforderungen genügen, jedoch gibt es bei der Umsetzung der Detailfunktionen viele Freiheitsgrade, die im Laufe eines Projekts durch eine agile Vorgehensweise zu einem idealen Ergebnis geführt werden können. Auch beispielsweise bei der Umsetzung von Enterprise-Performance-Management-Lösungen oder sonstigen eher Reporting-orientierten Unternehmenssoftwarekomponenten kommen die Vorteile einer agilen Vorgehensweise aus den gleichen Gründen besser zum Tragen.

Zwischen diesen beiden Ansätzen – phasenorientiert auf der einen und agil auf der anderen Seite – entwickeln sich zunehmend hybride Vorgehensmodelle, welche Konzepte aus beiden Ansätzen kombinieren. Beispielsweise werden Konzepte wie Sprints und Daily-Standup-Meetings aus dem Scrum-Umfeld in ein phasenorientiertes Gesamtverfahren eingebettet. Auch für diesen hybriden Ansatz gibt es Anwendungsfälle, bei dem dieser besonders geeignet ist. Dies sind in der Regel Anwendungsfälle, bei denen einerseits eine vorgegebene Menge an Mindestanforderungen meist rechtlich oder datenschutztechnisch vorgegeben ist, es aber darüber hinaus einen großen flexibel umsetzbaren Teil gibt. Dies ist beispielsweise bei der Umsetzung

von Geschäftsprozessen im Bereich Human-Capital-Management der Fall. Für die Verwaltung von Personaldaten und den dazugehörigen Prozessen sind entsprechende gesetzliche und datenschutztechnische Anforderungen vor einer Inbetriebnahme einer neuen Unternehmenssoftware ohne Einschränkungen umzusetzen. Bei Prozessen, welche beispielsweise die Organisation der Weiterbildung von Mitarbeitern im Unternehmen betreffen, kann die Umsetzung deutlich flexibler erfolgen.

Um mit diesen aktuellen Rahmenbedingungen in Projekten umgehen zu können, hat PROMATIS sein bisheriges Vorgehensmodell IQPM™ (PROMATIS software GmbH, Ettlingen, Deutschland) von einem Spiralmodell auf ein Vorgehen umgestellt, in dem für einzelne Teilbereiche Scope-basierte, hybride und agile Umsetzungen erfolgen. Abb. 2 zeigt die grobe Struktur des Vorgehensmodells. Bei einem Unternehmenssoftwareprojekt erfolgt zunächst eine grobe Bewertung und Einteilung in Teilprojekte, die entweder klassisch Scope-basiert, hybrid oder agil umgesetzt werden sollen. Im Anschluss werden die Teilprojekte unterschiedlich, entsprechend dem gewählten Typ, initiiert, umgesetzt und am Ende in Betrieb genommen. Abschließend erfolgt ein Review des Gesamtprojekts.

Zur Modellierung des Vorgehensmodells eignen sich verschiedene Modellierungssprachen. Wir verwenden in diesem Beitrag eine Modellierung mithilfe des Horus

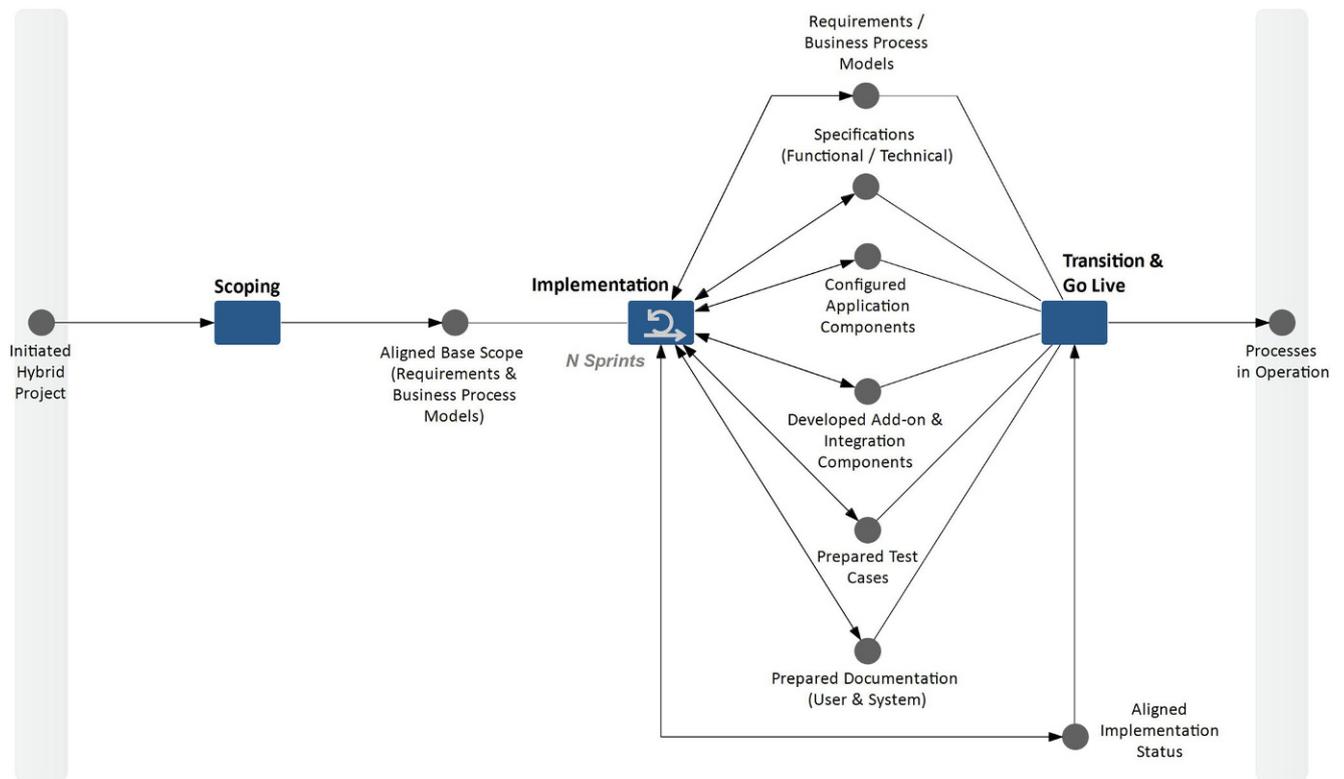


Abb. 3 Beispiel für ein hybrides Vorgehen. (aus dem Vorgehensmodell IQPM™ von PROMATIS)

Business Modeler ([www.horus.biz](http://www.horus.biz)). Sie basiert auf Petri-Netzen und verwendet verschiedene Erweiterungen wie die Updateverbindung (doppelt gerichtete Kante) und die Leseverbindung (nicht gerichtete Kante) [2].

Im Rahmen des Vorgehensmodells ist eine individuelle Anpassung an spezifische Rahmenbedingungen des jeweiligen Projekts vorgesehen. Darüber hinaus müssen bei einem Projekt nicht zwangsläufig alle 3 Grundarten vertreten sein, d. h. im einfachsten Fall werden auch Projekte durchgeführt, bei denen gegebenenfalls nur eine der 3 dargestellten Arten für die Umsetzung gewählt wird. Da die Umsetzung mit einem hybriden Vorgehen im Bereich Unternehmenssoftware aus den oben genannten Gründen für die meisten Praxisfälle anwendbar ist, wird dieses für die weitere Betrachtung im Rahmen dieses Beitrags genutzt.

In Abb. 3 wird beispielhaft ein angepasstes Vorgehen für ein rein hybrides Projekt dargestellt. Das entsprechende Unternehmenssoftwareprojekt ist grob in die 3 Phasen Scoping, Implementation und Transition & Go Live strukturiert. Die Implementierung ist im dargestellten Beispiel agil vorgesehen, jedoch in die generelle Phasenstruktur eingebettet und soll in Form von Sprints durchgeführt werden. Ein Rollout – beispielsweise für verschiedene Länder – ist in dem Beispielvorgehen nicht vorgesehen.

Abb. 4 zeigt die Detailaufgaben innerhalb der Sprints der Implementierung. Diese startet mit einer Sprint-Planung. Im Rahmen der hier vorgesehenen agilen Implementierung

wird einerseits eine Konfiguration von betriebswirtschaftlichen Standardmodulen durchgeführt, und es erfolgt andererseits eine Entwicklung von erforderlichen Zusatzkomponenten und Schnittstellen. Darüber hinaus wird jedoch auch die Erstellung und Aktualisierung von Anforderungen, Designspezifikationen, Testfällen und sonstigen Dokumentationsartefakten agil durchgeführt. Dies erfolgt – angelehnt an Scrum – durch eine für das Projekt festgelegte Anzahl von Sprints. Nach jedem Sprint ist eine aktuelle Version eines Prototyps der umzusetzenden Unternehmenssoftwarelösung – zumindest in Teilen – verfügbar. Dieser besteht aus konfigurierten Standardmodulen, entwickelten individuellen Komponenten und Integrationskomponenten. Zusätzlich werden jedoch auch die weiteren Artefakte bereitgestellt, die im Rahmen des Sprints ebenfalls weiterentwickelt wurden. Dies sind Zuordnungen von Anforderungen zu Prozessen, Designspezifikationen und Testfällen. Die Weiterentwicklung dieser Artefakte führt am Ende aller geplanten Sprints zu den finalen Dokumentationsartefakten der Anwender- und Systemdokumentation. Im Sprint-Review wird der aktuelle Stand der Gesamtlösung im Team gemeinsam bewertet. Der Abschluss des Projekts erfolgt nach Erreichen eines adäquaten Zustands der Lösung inklusive aller benötigten Artefakte in der Phase *Transition & Go Live*, in welcher Schulungsmaßnahmen, das Deployment in der Produktionsumgebung und die Inbetriebnahme erfolgen.

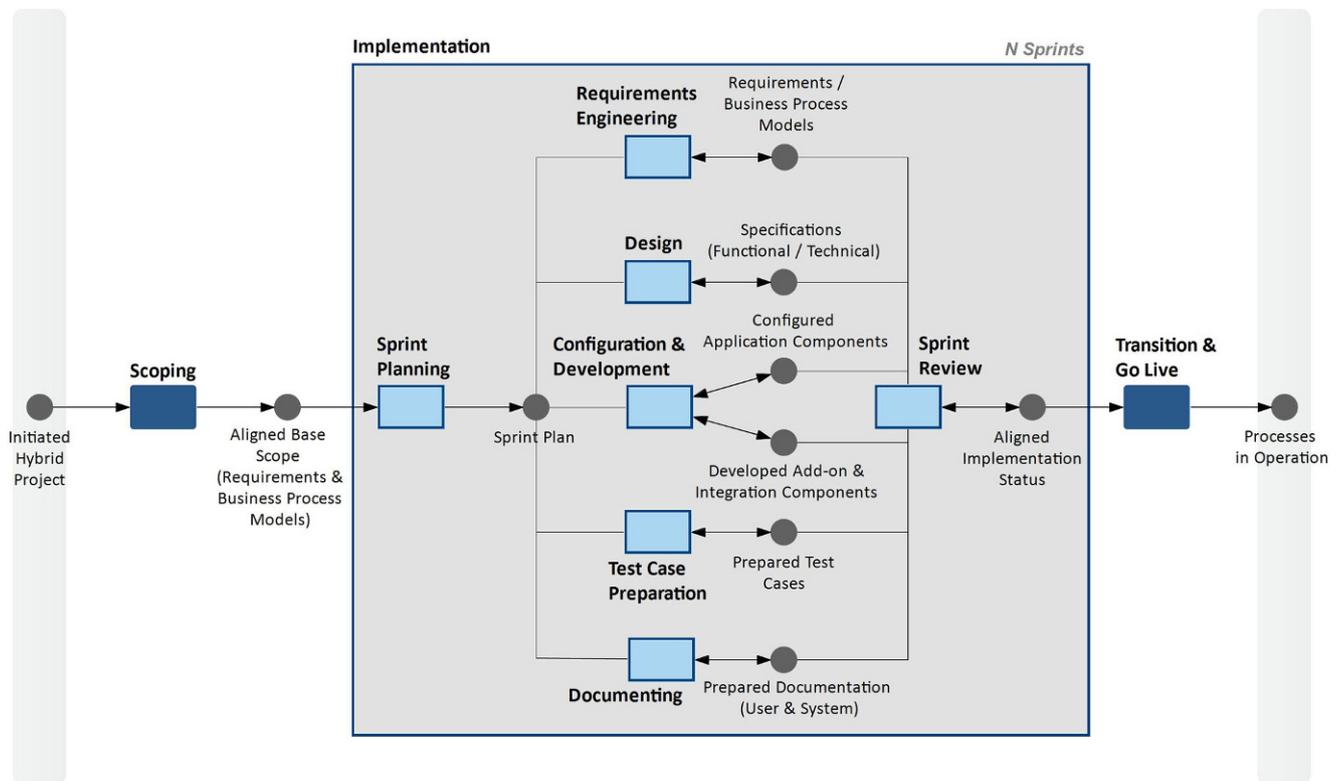


Abb. 4 Weiterentwicklung unterschiedlicher Artefakte im Rahmen der Sprints. (aus IQPM<sup>TM</sup> von PROMATIS)

## Process Mining zur kontinuierlichen Verbesserung des Vorgehensmodells

Für das zuvor grob beschriebene Vorgehen stellen sich bzgl. Bewertung, Verbesserung und einer detaillierten Parametrierung einige Fragen, aus deren Beantwortung sinnvolle Nachjustierungen abgeleitet werden könnten:

- Welche Auswirkungen haben die Länge und der Umfang der Scoping-Phase auf die Dauer und die Qualität der Implementierung? Wie sollte das Aufwands-/Zeitverhältnis zwischen Scoping und Implementation sein?
- Können Regeln abgeleitet werden, um Zeit und Aufwand des Scopings im Verhältnis zu einer optimalen Anzahl von Sprints zu ermitteln? Gibt es hier Unterschiede bei verschiedenen fachlichen Bereichen?
- Wie wirken sich unterschiedliche Schwerpunkte bei den Tätigkeiten innerhalb des Sprints auf den Projekterfolg aus?

Um diese Fragen zuverlässig beantworten zu können, bietet Process Mining die Möglichkeit, bereits abgeschlossene Softwareprojekte zu analysieren und laufende Softwareprojekte an gegebene Umstände anzupassen (siehe Abb. 5).

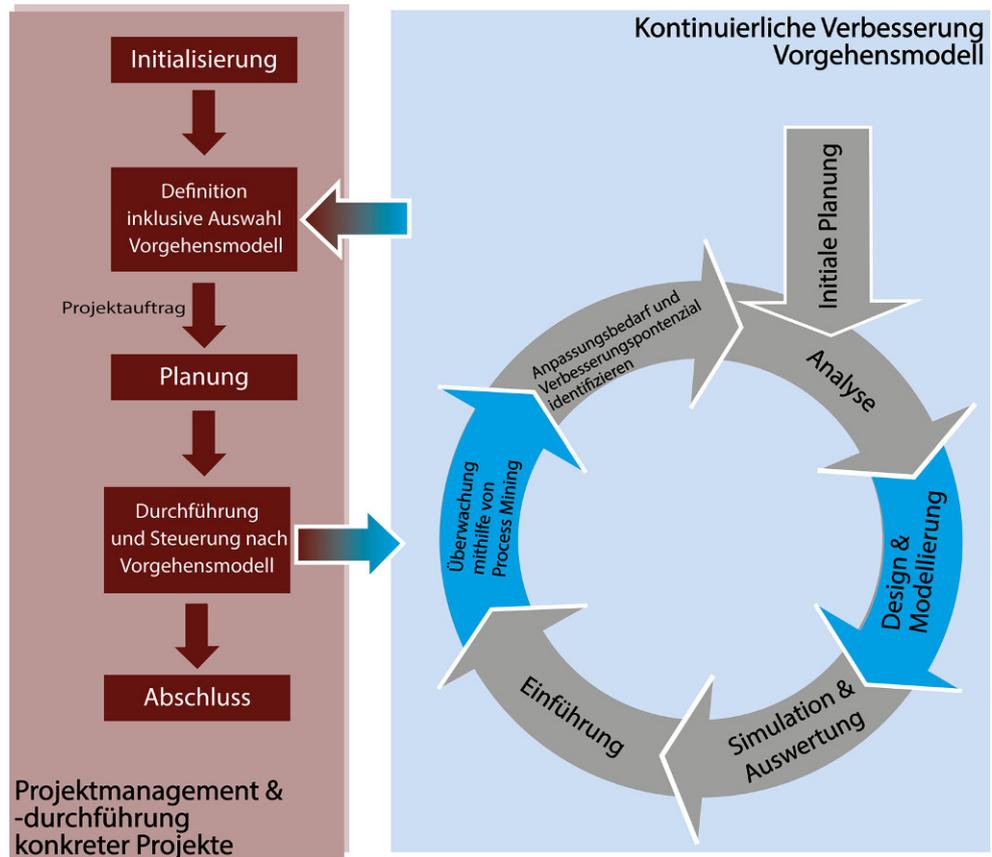
Process Mining umfasst eine Vielzahl an Verfahren für die datenbasierte Prozessanalyse. Die Grundlage für die-

sen Ansatz bildet ein Event-Log, in dem die Ereignisse innerhalb einer Prozessausführung festgehalten werden. Für jedes Ereignis müssen eine Bezeichnung und eine zeitliche Zuordnung vorhanden sein, damit ein Prozessmodell abgeleitet werden kann. Dieses Prozessmodell kann dann für weitere Analysen verwendet werden. Im Wesentlichen umfasst das Process Mining die folgenden 3 Bereiche:

- **Process Discovery:** Basierend auf der Reihenfolge der Ereignisse im Event Log wird ein Prozessmodell abgeleitet, das die Prozessausführung widerspiegelt.
- **Conformance Checking:** Nachdem ein Prozess ausgeführt wurde und ein Event Log vorliegt, kann das Event Log mit einem vorgegebenen Prozessmodell verglichen werden. Auf diese Weise können Diskrepanzen zwischen dem Ist-Zustand und dem Soll-Zustand eines Prozesses ermittelt werden.
- **Process Enhancement:** Zusätzlich zu der Ablaufperspektive können weitere Prozessaspekte analysiert werden, wie z. B. der zeitliche Ablauf der Aktivitäten oder die Rollenverteilung und die Zusammenarbeit der einzelnen Akteure innerhalb eines Prozesses.

Immer wichtiger wird auch der Einsatz von maschinellen Lernverfahren (künstliche Intelligenz) beim Process Mining, z. B. um Ereignisvorhersagen oder Ursachenanalysen durchzuführen. Auf diese Weise wird das Process Mining

**Abb. 5** Zusammenhang zwischen der Durchführung konkreter Projekte (Managementphasen in Anlehnung an DIN 69901-2:2009) und der kontinuierlichen Verbesserung des Vorgehensmodells. (Abb. teils basierend auf [20, Sp. 148])

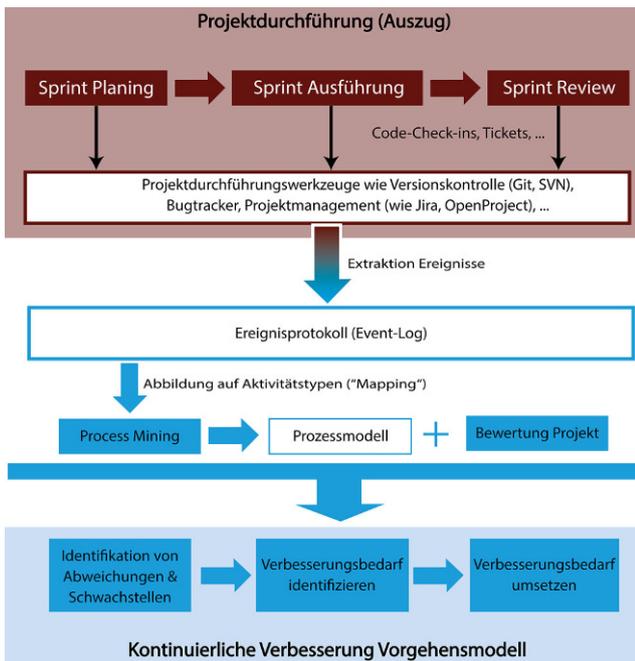


zu einem wichtigen Werkzeug für das operative Management und kann zur Automatisierung von Prozessen beitragen. Allerdings ist auch weiterhin die manuelle Prozessmodellierung notwendig, um den Soll-Zustand eines Prozesses zu definieren. In der Softwareentwicklung geben die beschriebenen Vorgehensmodelle diesen Zustand vor und können zur Vorgehenskontrolle verwendet werden. Im folgenden Abschnitt wird exemplarisch gezeigt, wie Process Mining zur Analyse von hybriden Softwareprojekten verwendet werden kann.

### Anwendungsbeispiel

Bei dem beschriebenen hybriden Vorgehensmodell erfolgt zunächst eine Projektplanung (Scoping), bei der Anforderungen an die zu entwickelnde Software ermittelt und Teilschritte für die Entwicklung festgelegt werden. Danach wird wie bei der rein agilen Softwareentwicklung die Entwicklung mithilfe von Sprints durchgeführt. Die Sprints orientieren sich dabei an der Projektplanung. Als Erweiterung zur agilen Vorgehensweise werden beim Sprint Review jedoch nicht nur das Entwicklungsartefakt, sondern auch weitere Artefakte betrachtet. Aus Sicht des Prozess Mining besteht bei dem hybriden Ansatz der Vorteil, dass ein grundlegen-

der Prozessablauf vorgegeben ist und dieser über den Verlauf des Projekts hinweg dokumentiert werden kann. Dazu werden meistens 3 verschiedene Werkzeuge verwendet: Projektmanagementplattform, Versionskontrollsystem und Bugtracker (siehe Tab. 1). Bei allen größeren Softwareentwicklungsprojekten ist die Einbindung eines Projektmanagementtools (wie beispielsweise Jira, GitHub oder GitLab) wichtig, um eine gemeinsame Arbeitsgrundlage zu schaffen und aktuelle Zwischenstände bei der Softwareentwicklung festzuhalten. Neben dem Projektmanagement ist weiterhin eine laufende Versionskontrolle bei größeren Softwareentwicklungsprojekten notwendig. Hier werden Informationen bezüglich Commits und Fileänderungen dokumentiert. Eines der bekanntesten Werkzeuge zur Versionskontrolle ist Git. Ein weiteres wichtiges Werkzeug für die Dokumentation der Softwareentwicklung ist der Bugtracker. Bei allen Systemen, die das Management von Bugs ermöglichen, werden wesentliche Informationen zur Erstellung einer Fehlermeldung, zur Behandlung eines Fehlers und zu möglichen Teilschritten gesichert. Viele Projektmanagementtools bieten zusätzlich zur allgemeinen Projektplanung auch die Möglichkeit zum Fehlermanagement. Aber es gibt auch zusätzliche Werkzeuge, die speziell für das Management von Bugs verwendet werden (beispielsweise Bugzilla). Abb. 6 (unten, rot hinterlegt) zeigt, dass dadurch im Rahmen der



**Abb. 6** Das Vorgehen der kontinuierlichen Verbesserung des Vorgehensmodells mithilfe von Event Logs

Tätigkeiten der Projektbearbeitung in diesen Werkzeugen entsprechende Ereignisse (z. B. der Check-in von neuem Code, das Mergen eines Branch in den Master Branch, das Erstellen oder Bearbeiten von Tickets) in den Projektdurchführungswerkzeugen ausgelöst werden. Diese werden dort jeweils protokolliert und lassen sich extrahieren. Hierzu gibt es verschiedene Schnittstellen und Frameworks (beispielsweise: GitPython, JGit, PyDriller).

Für die Auswertung des so extrahierten Ereignisprotokolls stehen weiterhin verschiedene kommerzielle sowie nichtkommerzielle Process-Mining-Werkzeuge zur Verfügung (z. B. ProM, Disco [Fluxicon BV, Eindhoven, Niederlande]). Eine Herausforderung dabei ist, die verschiedenen Ereignisse auf die Aktivitätstypen des im Vorgehensmodell beschriebenen Prozesses abzubilden. Dadurch kann ein Prozessmodell (siehe Abb. 6, Mitte „Prozessmodell“) aus dem Ereignisprotokoll gewonnen werden. Gemeinsam mit der Bewertung des Projektergebnisses kann so eine Vielzahl von Projekten genutzt werden, um zunächst Abweichungen vom Vorgehensmodell zu identifizieren. Zusammen mit der Bewertung des Projektergebnisses muss dann ein Experte für die Weiterentwicklung des Vorgehensmodells prüfen, ob daraus ein – für alle Projekte zutreffender – Erfolgsfaktor abgeleitet werden kann. Ist dies der Fall, wird der Verbesserungsbedarf identifiziert und umgesetzt. Misserfolgsk Faktoren führen ebenfalls zu Anpassungen des Vorgehensmodells, insbesondere dann, wenn ein identifizierter Misserfolgskfaktor durch ein Vorgehensmodell vorgeschrieben ist (siehe Abb. 6, blau hinterlegter Bereich). Dieser wird dann

aus dem Modell entfernt und ggf. durch ein alternatives Vorgehen (aus erfolgreichen Projekten) ersetzt.

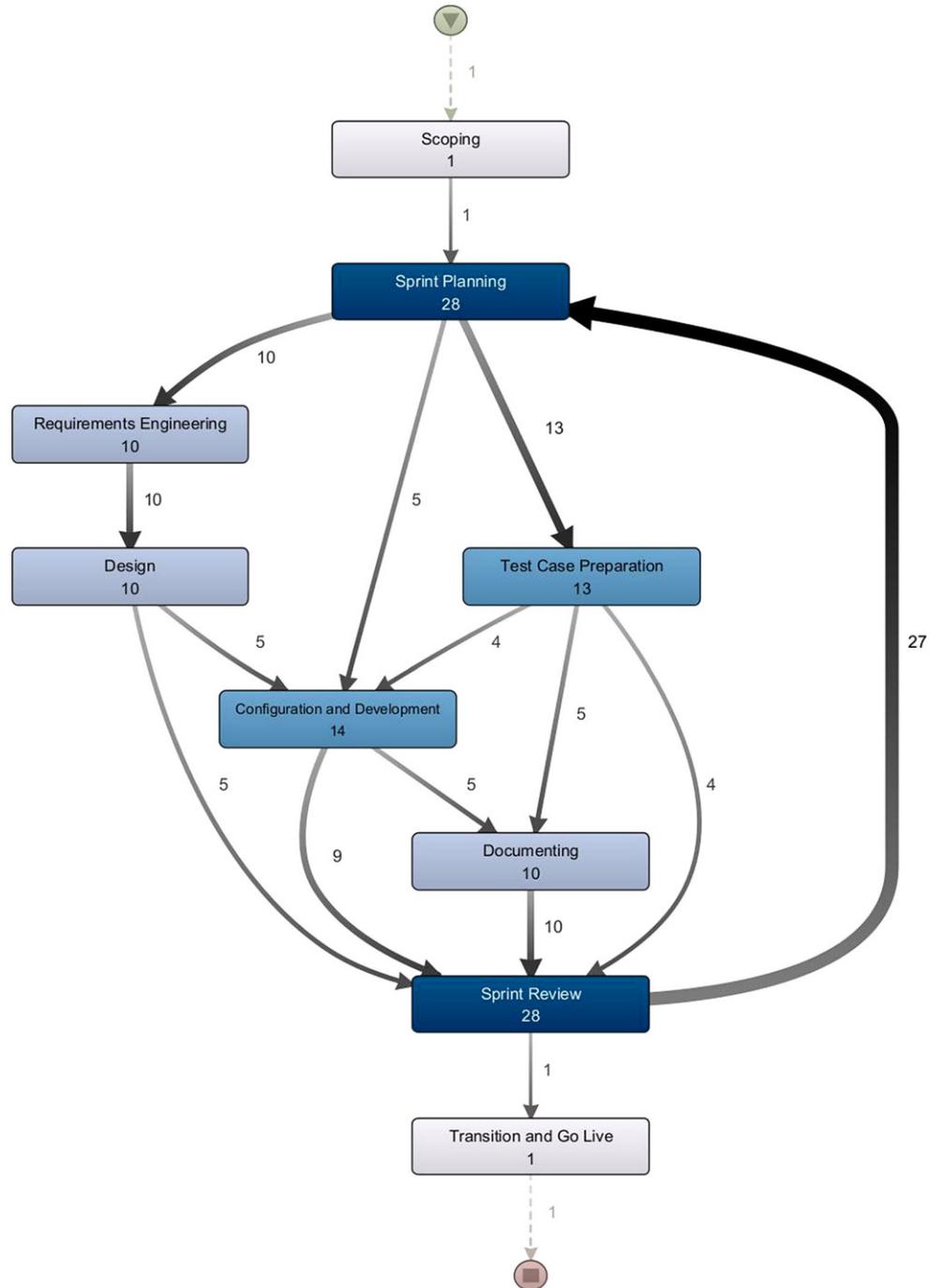
Abb. 7 zeigt beispielhaft ein aus einem konkreten Softwareentwicklungsprojekt gewonnenes Prozessmodell. Das Entwicklungsprojekt basierte auf dem hybriden Vorgehensmodell aus Abb. 4. Das Ablaufdiagramm wurde mithilfe des Process-Mining-Werkzeugs Disco erstellt und basiert auf fiktionalen Daten, wie sie aber auch aus den beschriebenen Datenquellen extrahiert werden könnten. Dabei ist anzumerken, dass die gewonnenen Daten zunächst durch eine Abbildung, wie beispielsweise in [9] beschrieben, den beschriebenen Aktivitäten im hybriden Vorgehensmodell zugeordnet werden müssen. Das präparierte Event Log kann in einem nächsten Schritt in Disco geladen werden, welches automatisch ein Ablaufdiagramm von dem Softwareentwicklungsprojekt erstellt. In dem gegebenen Beispiel wurden insgesamt 28 Sprints durchlaufen, die jeweils mit einem Sprint Planning gestartet und mit einem Sprint Review beendet wurden. Das Projekt wurde zunächst mit dem Scoping begonnen und am Ende mit der *Transition & Go Live*-Phase abgeschlossen. Bei den 28 Sprints wurden alle beschriebenen Aktivitäten aus dem Vorgehensmodell mehrmals ausgeführt, am häufigsten wurden die Aktivitäten „Configuration and Development“ (14-mal) und „Test Case Preparation“ (13-mal) durchgeführt.

In Abb. 8 wird eine detaillierte Auswertung der Zeitdauern aller Aktivitäten gezeigt. Auch diese Werte basieren auf fiktiven Werten und sollen vor allem zur Veranschaulichung der Möglichkeiten des Process Enhancements in der Softwareentwicklung dienen. Abb. 9 stellt ein weiteres Beispiel für ein Softwareentwicklungsprojekt dar, bei dem einige Schritte des hybriden Vorgehensmodells ausgelassen wurden und das Projekt bereits nach 20 Sprints abgebrochen wurde. Insbesondere wurde das Sprint Review 2-mal nach einem Sprint übergangen und die Dokumentation wurde vernachlässigt (nur 2 Ausführungen). Danach wurde das Projekt abgebrochen und die *Transition & Go Live*-Phase nicht mehr ausgeführt. Dieses Beispiel verdeutlicht, dass mithilfe des Process Mining Abweichungen vom Vorgehensmodell leicht erkannt werden können und Anpassungen basierend auf Kennzahlen, wie den Durchlaufzeiten, vorgenommen werden können.

## Fazit & Ausblick

Dieser Beitrag zeigt den Nutzen von Process Mining für die kontinuierliche Verbesserung von Vorgehensmodellen der Softwareentwicklung. Besonders bei relativ jungen Vorgehensmodellen, wie sie aktuell insbesondere im Bereich der hybriden Vorgehensmodelle zu finden sind, entfaltet die neue Möglichkeit großes Potenzial. Aber auch die Anwendung auf etablierte Vorgehensmodelle ist sinnvoll, beispiels-

**Abb. 7** Beispielhafter Projekt-  
ablauf



weise um unternehmenskulturspezifische Anpassungen vornehmen zu können. Erfolgsentscheidend ist es – möglichst ohne Mehraufwand für die Mitarbeiterinnen und Mitarbeiter der Projekte – aus bestehenden Werkzeugen (wie Git und Jira) ein Ereignisprotokoll so zu extrahieren, dass eine automatisierte Zuordnung zu den Aktivitäten des im Vorgehensmodell definierten Prozesses möglich ist.

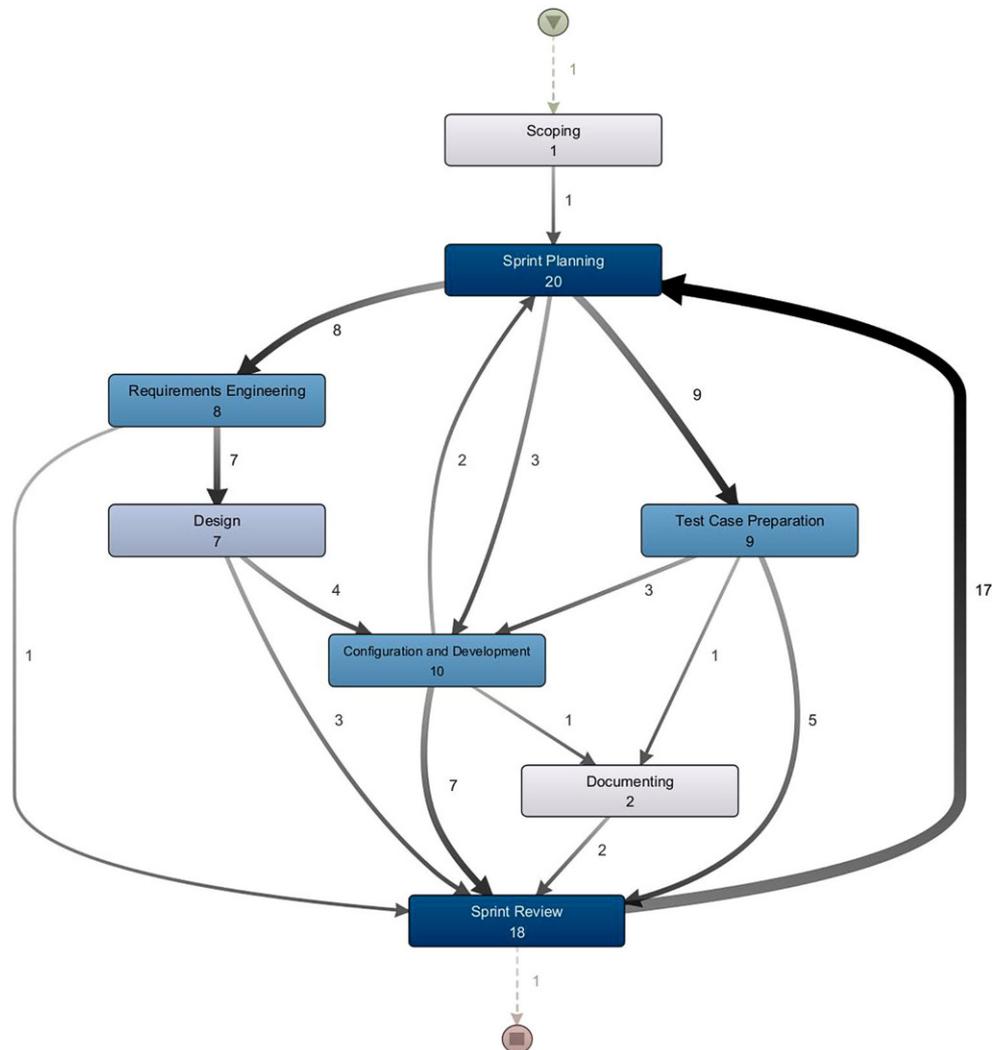
Der vorgeschlagene Ansatz betrachtet nur die kontinuierliche Verbesserung des Vorgehensmodells bzw. präziser

die definierten Aktivitäten und ihre Anordnung. Dabei werden bei Anwendung des Ansatzes in der Regel iterative Verbesserungen entstehen, disruptive Ideen müssen anders geschöpft werden. Methoden für das Software Engineering umfassen weitere Komponenten (vgl. Abb. 1) – auch deren Weiterentwicklung mit anderen Ansätzen ist für die effiziente und effektive Durchführung von Softwareentwicklungsprojekten wichtig.

Activity	▲ Frequency	Relative frequency	Mean duration	Duration range
Sprint Planning	28	24.35 %	54 mins, 25 secs	9 hours, 24 mins
Sprint Review	28	24.35 %	54 mins, 15 secs	7 hours, 11 mins
Configuration and Development	14	12.17 %	1 hour, 1 min	8 hours, 25 mins
Test Case Preparation	13	11.3 %	2 hours, 28 mins	1 day, 1 hour
Requirements Engineering	10	8.7 %	45 mins, 48 secs	5 hours, 49 mins
Design	10	8.7 %	3 hours, 41 mins	1 day, 10 hours
Documenting	10	8.7 %	47 mins	4 hours, 25 mins
Scoping	1	0.87 %	7 hours, 24 mins	0 millis
Transition and Go Live	1	0.87 %	12 hours, 11 mins	0 millis

Abb. 8 Zusätzliche Kennzahlen, die mit Disco ausgewertet werden können

Abb. 9 Beispielhafter Projekt-  
ablauf mit Abweichungen



Künftig ist es vorstellbar, Process Mining nicht nur für die Auswertung von abgeschlossenen Projekten, sondern auch für die Analyse laufender Projekte einzusetzen und dann die Anpassungsmaßnahmen noch für das laufende Projekt zu definieren. In diesem Fall ist jedoch zwischen projektspezifischen Anpassungen (diese sind besonders für das Projekt geeignet, fließen aber nicht in das allgemeine Vorgehensmodell ein, weil sie nicht auf eine Vielzahl von Projekten zutreffen) und allgemeinen Anpassungen (diese fließen in das Vorgehensmodell ein) zu unterscheiden. Die

allgemeine Herausforderung der Änderung des Prozesses und die Auswirkungen auf laufende Instanzen muss dafür jeweils auch betrachtet werden (z.B. kann definiert werden, dass laufende Instanzen die Änderung nicht übernehmen, weil ihr Nutzen gegenüber dem Änderungsaufwand zu gering ist).

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

## Literatur

- Mills HD, Dyer M, Linger RC (1987) Cleanroom software engineering
- Schönthaler F, Vossen G, Oberweis A, Karle T (2012) Business processes for business communities: modeling languages, methods, tools. Springer, Berlin Heidelberg <https://doi.org/10.1007/978-3-642-24791-0>
- Cook JE, Wolf AL (1995) Automating process discovery through event-data analysis. In: 17th International Conference on Software Engineering
- Cook JE, Wolf AL (1998) Discovering models of software processes from event-based data. *ACM Trans. Softw. Eng. Methodol.* 7(3):215–249. <https://doi.org/10.1145/287000.287001>
- van der Aalst W (2016) Process mining. *Data science in action*. Springer, Berlin Heidelberg
- Rubin V, Günther CW, Van Der Aalst WM, Kindler E, Van Dongen BF, Schäfer W (2007) Process mining framework for software processes. In: International conference on software process
- Akman B, Demirörs O (2009) Applicability of process discovery algorithms for software organizations. In: Conference proceedings of the 35th Euromicro conference on software engineering and advanced applications
- Lemos AM, Sabino CC, Lima RMF, Oliveira CAL (2011) Using process mining in software development process management: a case study. In: IEEE international conference on systems, man and cybernetics
- Poncin W, Serebrenik A, van der Brand M (2011) Process mining software repositories. In: 15th European conference on software maintenance and reengineering
- Mittal M, Sureka A (2014) Process mining software repositories from student projects in an undergraduate software engineering course. In: 36th international conference on software engineering
- Rubin V, Lomazova I, van der Aalst WMP (2014) Agile development with software process mining. In: ICSSP
- Rubin V, Mitsyuk AA, Lomazova I, van der Aalst WMP (2014) Process mining can be applied to software too! In: International Symposium on Empirical Software Engineering and Measurement
- Kneuper R (2016) Process Mining bei Softwareprozessen. In: *Lecture Notes in Informatics*, S 121–134
- Erdem S, Demirörs O (2017) An exploratory study on usage of process mining in agile software development. In: *Communications in computer and information science*, S 187–196
- Marques RR, da Silva MM, Ferreira DR (2018) Assessing agile software development processes with process mining: a case study. In: IEEE 20th conference on business Informatics
- Caldeira J, Cardoso J (2019) Assessing software development teams' efficiency using process mining. In: International conference on process mining
- Dumbacher P, Aly A, Zrenner M, Eskofier BM (2020) Exploration of process mining opportunities in educational software engineering—the gitlab analyser. In: 13th international conference on educational data mining
- Erdem S, Demirörs O, Rabhi F (2018) Systematic mapping study on process mining in agile software development. In: *Communications in computer and information science*, S 289–299
- Vossen G, Schönthaler F, Dillon S (2017) The web at graduation and beyond: business impacts and developments. Springer, Berlin Heidelberg
- Alpers S (2019) Modellbasierte Entscheidungsunterstützung für Vertraulichkeit und Datenschutz in Geschäftsprozessen. KIT Scientific Publishing, <https://doi.org/10.5445/KSP/1000094545>