

# AGILE ENTWICKLUNG VON KI-AGENTEN

Warum generative Agenten zentrale Annahmen agiler Entwicklung infrage stellen – und vier Handlungsempfehlungen für die Praxis.

# WANN AGILE METHODEN AN IHRE GRENZEN STOßEN

**Zum Weiterlesen:** Dieses Knowledge Nugget fasst den Beitrag *Agile Entwicklung generativer KI-Agenten: Herausforderungen und praxisnahe Handlungsempfehlungen* (Autor\*innen: Fetzer, Gimpel, Jung, Schoch, Strasser) zusammen, der in der **HMD Praxis der Wirtschaftsinformatik** im September 2026 erschienen ist.

Generative KI-Agenten gewinnen zunehmend an Bedeutung: autonome Softwareeinheiten, die ihre Umgebung wahrnehmen, Entscheidungen treffen und zielgerichtet handeln [12]. Sie übernehmen nicht mehr nur einfache Assistenzfunktionen [15], sondern zunehmend komplexe Aufgaben vollständig autonom. Arbeiten mehrere Agenten zusammen, entstehen Multi-Agenten-Systeme, die komplexe operative und strategische Prozesse automatisieren können [1].

Viele Unternehmen haben die Entwicklung von Multi-Agenten-Systemen initiiert, doch ein Großteil gelangt nicht über die Pilotphase hinaus oder scheitert bereits in dieser Phase [6]. Ein wesentlicher Grund liegt in den Besonderheiten solcher Agenten, die etablierte agile Entwicklungs- und Abnahmeprozesse herausfordern. Anders als klassische Software wird das Verhalten eines Agenten nicht durch explizit programmierte Logik bestimmt, sondern durch einen generativen KI-Kern, der **probabilistisch, teiltransparent und kontextabhängig** agiert. Gesteuert wird weniger über Code als über Kontext und Prompts in natürlicher Sprache [3, 10].

Einige Forschungsbeiträge untersuchen den Einsatz von KI zur Verbesserung agiler Entwicklungsprozesse, etwa für Aufwandsschätzungen, Risikoerkennung oder die Priorisierung von Backlogs [2, 5, 7, 11]. Weit weniger untersucht ist hingegen die umgekehrte und für die Praxis zentrale Frage, inwieweit agile Steuerungslogiken tragfähig bleiben, wenn das zu entwickelnde System selbst auf KI basiert.

**Wie tragfähig bleibt die agile Steuerungslogik, wenn das entwickelte System selbst KI-basiert ist?**

**i** **Agile Methoden werden im Kontext von KI-Agenten nicht obsolet, sie müssen aber anders operationalisiert werden.** Weg von der einmaligen Sprint-Abnahme, hin zu einem kontinuierlichen, adaptiven Prozess, der Entwicklung und Betrieb integriert.

Dieser Beitrag identifiziert vier zentrale Stellen im Scrum-Zyklus, an denen bei der agilen Entwicklung von Multi-Agenten-Systemen spezifische Herausforderungen auftreten, und leitet **konkrete Handlungsempfehlungen** zu deren Bewältigung ab.

# KLASSISCHE SOFTWARE VS. GENERATIVER KI-AGENT

Agile Methoden stützen sich auf Artefakte wie das Product Backlog und die Definition of Done [4, 13, 14]. Damit setzen sie voraus, dass sich erwartetes Systemverhalten hinreichend spezifizieren, überprüfen und über die Zeit stabil reproduzieren lässt. Für deterministische Komponenten, etwa Schnittstellen oder Datenbankverbindungen, bleibt diese Annahme weitgehend bestehen. Wo jedoch ein probabilistischer GenKI-Kern und Prompts das Systemverhalten maßgeblich prägen, geraten diese Grundannahmen ins Wanken – und mit ihnen etablierte Logiken der Spezifikation, Abnahme und Steuerung [8].

| Dimension     | Klassische Software                                      | GenKI-Agent                                                          |
|---------------|----------------------------------------------------------|----------------------------------------------------------------------|
| Steuerung     | über Code und explizit programmierte Entscheidungslogik  | über Kontext und Prompts in natürlicher Sprache                      |
| Anforderungen | formal spezifizierbar, über Schnittstellen begrenzt      | großer Interpretationsspielraum, oft bewusst oder unbewusst unscharf |
| Ausgaben      | reproduzierbar, stabile Eingabe-Ausgabe-Beziehung        | variabel und semantisch offen, je nach Situation                     |
| Abnahme       | Soll-Ist-Vergleich, klares „bestanden / nicht bestanden“ | multidimensional und nur interpretativ bewertbar                     |
| Im Betrieb    | stabil bis zur bewussten Code-Anpassung                  | Drift durch Daten- und Modellupdates externer Anbieter               |



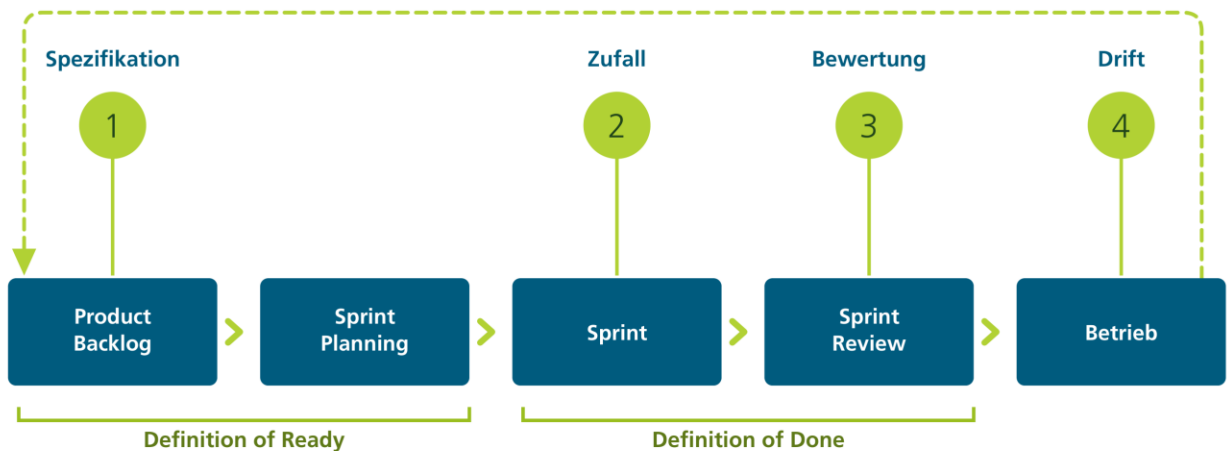
**Bei Multi-Agenten-Systemen verstärkt sich jede dieser Verschiebungen:** Das Systemverhalten entsteht aus dem dynamischen Zusammenspiel mehrerer, teils autonom agierender Agenten und lässt sich einzelnen Komponenten kaum noch zuordnen [1, 9].

**i** Die rechte Spalte bildet den Regelfall generativer Agentensysteme ab. Aus den damit verbundenen Besonderheiten ergeben sich vier zentrale Herausforderungen entlang des Scrum-Zyklus und vier darauf abgestimmte Handlungsempfehlungen.

# VIER HERAUSFORDERUNGEN ENTLANG DES SCRUM-ZYKLUS

Entlang des Scrum-Zyklus, von der Spezifikation fachlicher Anforderungen in Product Backlog Items bis zur Steuerung nach der Abnahme, zeigen sich vier zentrale Herausforderungen, die sich bei Multi-Agenten-Systemen zusätzlich verschärfen.

Kontinuierlicher Kreislauf von Entwicklung und Betrieb



## Natürliche Sprache ersetzt die Spezifikation

Ausgaben in natürlicher Sprache weiten den Interpretationsspielraum erheblich. Spezifikationslücken werden nicht nur von Menschen unterschiedlich gedeutet, sondern vom Agenten zur Laufzeit selbst gefüllt; Backlog Items verlieren ihre Funktion als eindeutige gemeinsame Referenz.

**Beispiel:** Ein Item verlangt eine Antwort „im Sinne des Unternehmens“: was das heißt, füllt der Agent selbst aus.



## Zufall beeinflusst die Ausgaben

Der generative Kern ist inhärent nichtdeterministisch. Gleiche Eingaben führen zu unterschiedlichen, nur eingeschränkt reproduzierbaren Ergebnissen; Qualität und Relevanz variieren je nach Situation. Eine Sprint-Abnahme kann sich nicht auf stabil reproduzierbares Verhalten stützen.

**Beispiel:** „Fachlich korrekte“ Antworten variieren in Länge, Struktur und Tonalität.



## Ausgaben sind nicht eindeutig bewertbar

Sprachliche Ergebnisse sind semantisch reichhaltig, aber strukturell offen und entziehen sich einer eindeutigen formalen Bewertung. Ein klares „bestanden / nicht bestanden“ lässt sich oft nicht festlegen; Kriterien wie Verständlichkeit bleiben interpretationsabhängig.

**Beispiel:** Ob eine Antwort „empathisch“ ist, lässt sich nicht objektiv bewerten.



## Verhalten bleibt im Betrieb nicht stabil

Datenformate ändern sich, Nutzungsmuster verschieben sich, und externe Anbieter aktualisieren Modelle oder Safety-Policies. Das Verhalten kann sich auch nach der Abnahme zeitlich versetzt verändern, sodass die Grenze zwischen Entwicklung und Betrieb verschwimmt.

**Beispiel:** Ein Modellupdate verändert Wochen später Tonalität und Länge der Ausgaben.

# VIER HANDLUNGSEMPFEHLUNGEN FÜR DIE PRAXIS

Die Antwort ist nicht, agile Methoden aufzugeben, sondern sie gezielt anzupassen. Die vier Empfehlungen folgen der Reihenfolge der vier Herausforderungen, von der Definition of Ready bis in den laufenden Betrieb.

## 1 Definition of Ready mit belastbaren Guardrails

Ein Backlog Item gilt erst als „ready“, wenn Fachseite und Entwicklung ein gemeinsames mentales Modell der Aufgabe teilen. Da ein vollständig identisches Erwartungsbild kaum erreichbar ist, helfen Kriterien mit Guardrail-Charakter: ein kurzes, KI-spezifisches Template hält fest, welche Ausgaben der Agent keinesfalls produzieren darf, etwa bei einem Agenten für Mandanten-E-Mails keine verbindliche steuerliche Einzelfallberatung ohne menschliche Freigabe.

## 2 Akzeptanzkriterien als erlaubter Verhaltensraum

Statt eines einzigen Soll-Ergebnisses beschreiben die Akzeptanzkriterien einen zulässigen Verhaltensraum: welche Mindestanforderungen gelten, welche Variationen in Länge, Struktur und Tonalität zulässig sind und welche Ausgaben kategorisch ausgeschlossen werden. Positiv- und Negativbeispiele machen diesen Raum greifbar und dienen als Bewertungsgrundlage im Sprint Review.

## 3 Qualität multidimensional abnehmen

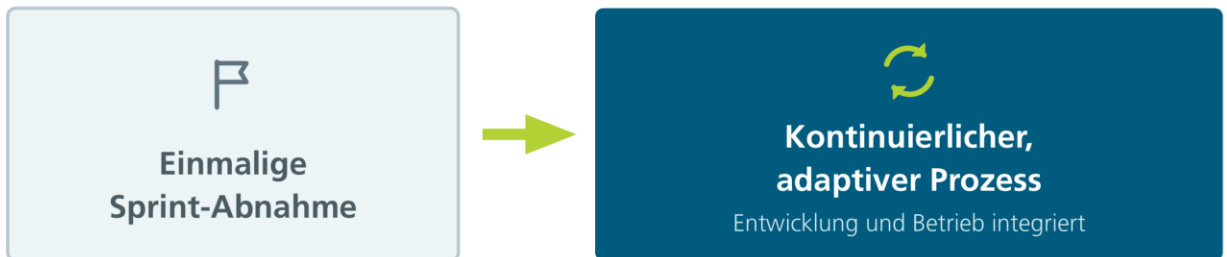
Wenige klare Dimensionen, etwa Korrektheit, Compliance, Nachvollziehbarkeit und Adressatengerechtigkeit, werden mit Stufen und einem Mindestniveau beurteilt. Menschliche Prüfung und die automatisierte Bewertung durch ein Sprachmodell „LLM-as-a-Judge“ greifen ineinander und werden über ein versioniertes Golden Dataset kalibriert. Die konkrete Ausgestaltung ist im zugrunde liegenden Beitrag ausführlich beschrieben.

## 4 Im Betrieb kontinuierlich evaluieren

Die Definition of Done wird um Betriebskriterien erweitert: Monitoring und Logging werden eingerichtet, ein „Stability Budget“ quantifiziert die tolerierbare Drift pro Release, und automatisierte Testläufe nach Anpassungen an Prompts, Tools oder Modellversionen werden in die Definition of Done aufgenommen. So bleibt die Sprint-Abnahme vor dem Go-Live möglich, ohne den iterativen Prozess durch Laufzeitdaten zu blockieren.

# FAZIT UND AUSBLICK

Agile Methoden werden im Kontext generativer Agenten **nicht obsolet**. Sie müssen aber anders operationalisiert werden. Die Rolle agiler Steuerung verschiebt sich dabei grundlegend: von einer einmaligen Sprint-Abnahme hin zu einem kontinuierlichen, adaptiven Prozess, der Entwicklung und Betrieb integriert.



Konkret werden vier Steuerungselemente neu ausgerichtet:

- Die **Definition of Ready** verankert ein gemeinsames mentales Modell zwischen Fachabteilung und Entwicklung.
- **Akzeptanzkriterien** beschreiben einen erlaubten Verhaltensraum statt eines punktuellen Zielergebnisses.
- **Qualitätsabnahmen** kombinieren multidimensionale menschliche und automatisierte Bewertungen.
- Die **Definition of Done** wächst um betriebsbezogene Stabilitätskriterien und kontinuierliche, Trace-basierte Evaluationen.

So wird die Abnahme generativer Agenten verlässlich, ohne den iterativen Scrum-Prozess auszubremsen.

**i** **Agile Methoden bleiben gültig, sie werden für KI-Agenten gezielt neu ausgerichtet.** Weiterer Forschungsbedarf besteht in der Operationalisierung von Verhaltensräumen, der Messung der Stabilität von Multi-Agenten-Systemen sowie der agilen Governance im Kontext externer generativer KI-Anbieter.

# QUELLEN

Dieses Knowledge Nugget fasst den Beitrag **„Agile Entwicklung generativer KI-Agenten“** (HMD Praxis der Wirtschaftsinformatik) in kompakter Form zusammen.

[1] Allmendinger, S., Bonenberger, L., Endres, K., Fetzer, D., Gimpel, H. & Kühl, N. (2026). Multi-agent AI. Electronic Markets, 36(1). <https://doi.org/10.1007/s12525-025-00862-z>

[2] Almalki, S. S. (2025). AI-Driven Decision Support Systems in Agile Software Project Management: Enhancing Risk Mitigation and Resource Allocation. Systems, 13(3), 208. <https://doi.org/10.3390/systems13030208>

[3] Banh, L. & Strobel, G. (2023). Generative artificial intelligence. Electronic Markets, 33(1). <https://doi.org/10.1007/s12525-023-00680-1>

[4] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2001). Manifesto for agile software development. <https://agilemanifesto.org/>

[5] Cabrero-Daniel, B. (2023). AI for Agile development: a Meta-Analysis. arXiv. <https://doi.org/10.48550/arXiv.2305.08093>

[6] Challapally, A., Pease, C., Raskar, R. & Chari, P. (2025). The GenAI divide: State of AI in business 2025. [https://mlq.ai/media/quarterly\\_decks/v0.1\\_State\\_of\\_AI\\_in\\_Business\\_2025\\_Report.pdf](https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf)

[7] Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T., Ghose, A. & Menzies, T. (2016). A deep learning model for estimating story points. arXiv. <https://doi.org/10.48550/arXiv.1609.00489>

[8] Fetzer, D., Gimpel, H., Meindl, O. & Strickmann, J. (2025). Responsible engineering of information systems based on generative artificial intelligence: An action design research study at a German premium car manufacturer. Business & Information Systems Engineering. <https://doi.org/10.1007/s12599-025-00950-6>

[9] Fetzer, D., Gimpel, H. & Schoch, M. (2026). Orchestrating generative AI-based multi-agent systems for complex knowledge work automation: A design science research approach. Proceedings of the 59th Hawaii International Conference on System Sciences (HICSS).

# QUELLEN

- [10] Feuerriegel, S., Hartmann, J., Janiesch, C. & Zschech, P. (2024). Generative AI. *Business & Information Systems Engineering*, 66(1), 111–126. <https://doi.org/10.1007/s12599-023-00834-7>
- [11] Radwan, A. M., Abdel-Fattah, M. A. & Mohamed, W. (2025). Smart Agile Prioritization and Clustering: An AI-Driven Approach for Requirements Prioritization. *IEEE Access*, 13, 127335–127350. <https://doi.org/10.1109/ACCESS.2025.3589959>
- [12] Sapkota, R., Roumeliotis, K. I. & Karkee, M. (2026). AI agents vs. agentic AI: A conceptual taxonomy, applications and challenges. *Information Fusion*, 126. <https://doi.org/10.1016/j.inffus.2025.103599>
- [13] Schwaber, K. (2004). *Agile project management with scrum*. Microsoft Press.
- [14] Schwaber, K. & Sutherland, J. (2020). *The scrum guide: The definitive guide to scrum: The rules of the game*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- [15] Seufert, S. & Meier, C. (2023). Hybride Intelligenz: Zusammenarbeit mit KI-Assistenzsystemen in wissensintensiven Bereichen. *HMD Praxis der Wirtschaftsinformatik*, 60(6), 1194–1209. <https://doi.org/10.1365/s40702-023-01012-9>

## Fraunhofer-Institut für Angewandte Informationstechnik FIT Generative AI Lab

Alter Postweg 101, 86159 Augsburg

[www.fit.fraunhofer.de](http://www.fit.fraunhofer.de)

<https://www.linkedin.com/company/fraunhofer-fit-generative-ai-lab/>

### Ansprechpersonen:

Prof. Dr. Henner Gimpel

[henner.gimpel@fit.fraunhofer.de](mailto:henner.gimpel@fit.fraunhofer.de)

<https://www.linkedin.com/in/henner-gimpel/>

Prof. Dr. Manfred Schoch

[manfred.schoch@fit.fraunhofer.de](mailto:manfred.schoch@fit.fraunhofer.de)

<https://www.linkedin.com/in/manfred-schoch-42257877/>

**Dieses Knowledge Nugget ist im Rahmen des Forschungsprojekts Agentic Work  
Automation (AWA) entstanden.**

### Finanzielle Förderung:



Bayerische  
Transformations- und  
Forschungsstiftung

### Projektpartner:

Ancud 



 **Fraunhofer**  
FIT



soffico

### Mitwirkende an diesem Knowledge Nugget:

Dominik Fetzter, Henner Gimpel, Carolin Jung, Manfred Schoch, Sidney Straßer, Philipp Szaruga (alle  
Fraunhofer FIT)