

Agile Softwareentwicklung in regulierten Organisationen

Radebeul, 26 September 2025

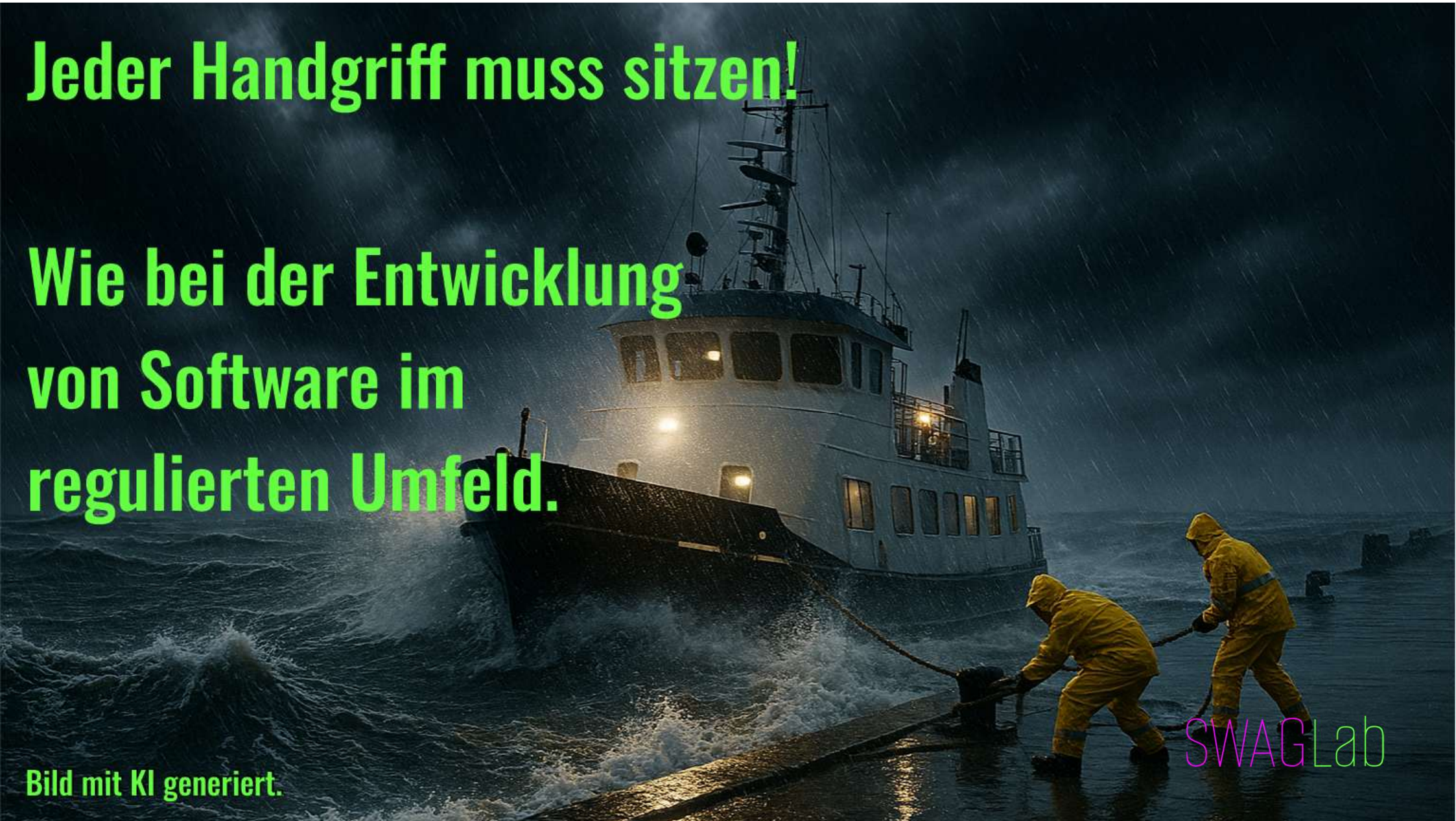
Uwe Vigneschow / Principal Consultant / SWAGLab

Jeder Handgriff muss sitzen!

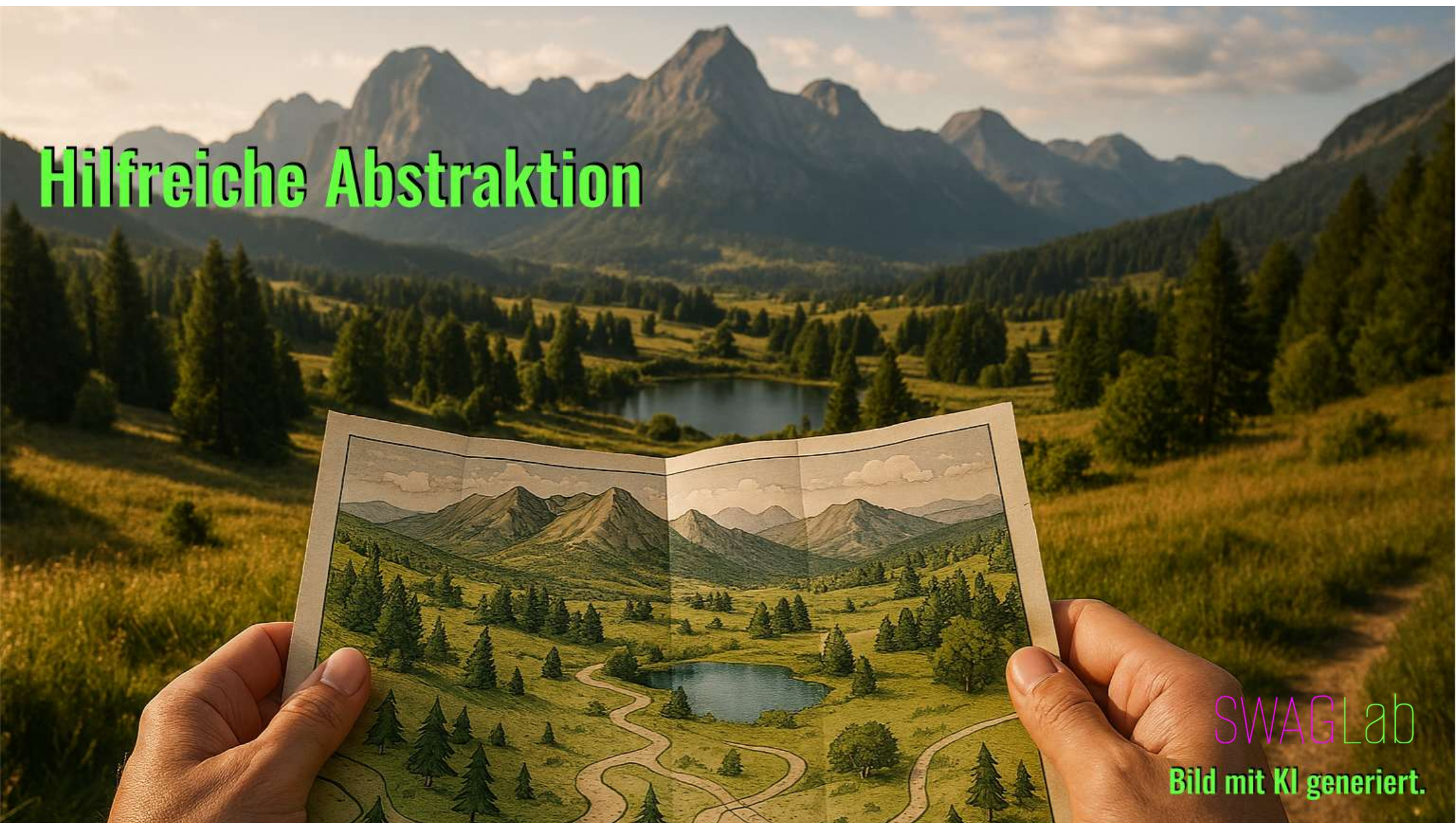
**Wie bei der Entwicklung
von Software im
regulierten Umfeld.**

Bild mit KI generiert.

SWAGLab



Hilfreiche Abstraktion



SWAGLab
Bild mit KI generiert.

Grundlagen

Reguliertes Umfeld

- > z.B.: Entwicklung von Medizingeräten oder von Software für die pharmazeutische Industrie
- > Vorgaben durch Gesetze und Standards wie IEC 62304, 21 CFR Part 820, GAMP 5, 21 CFR Part 11, etc. zu Entwicklungsprozessen, Datenintegrität, elektronischen Signaturen, Validierung usw.*

Agilität

GAMP: Good Automated Manufacturing Practice, CFR: Code of Federal Regulations

*fordert implizit ein (agiles) QMS

SWAGLab

Grundlagen

Reguliertes Umfeld

Arbeite nach definierten Prozessen und folge etablierten Methoden!
Wir erstellen kritische Software und benötigen hohe Qualität,
um die Risiken im Griff zu haben!

Agilität – Konzept zur Bewältigung komplexer Vorhaben

- > Etablierte Entwicklungsframeworks mit Fokus auf Kommunikation, Zusammenarbeit, Selbstorganisation, kontinuierlicher Verbesserung und schneller Reaktionsfähigkeit auf Veränderungen
- > Basiert auf Werten, dazu passenden Prinzipien (des Agilen Manifests) und konkreten Praktiken

SWAGLab



Was ist der Kern
regulierter Softwareentwicklung?

SWAGLab

Was macht regulierte Entwicklung und formales Testen aus?

> 4-Augen-Prinzip

Reviews oder **Pair Working** für jeden Schritt im Entwicklungsprozess

> Nachverfolgbarkeit

Welche Anforderung durch wen ist wo von wem umgesetzt, durch wen wann im Review gewesen und wann, wie und mit welchem Ergebnis getestet worden?

> Reproduzierbarkeit

Definierte Anforderungen, die auf einem definierten System definiert getestet wurden

Setze die Konzepte von RE, DDD, Scrum, Kanban, XP, TDD, BDD, etc. **vollständig** um!

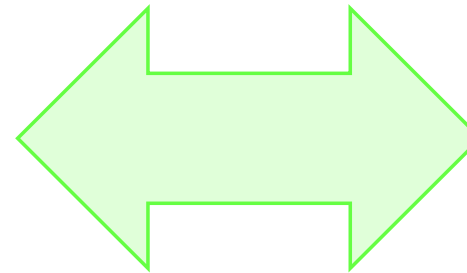
SWAGLab

Nachvollziehbarkeit ist eine Voraussetzung für Revisionssicherheit

Revisionssicherheit ist die Fähigkeit eines Systems

- > Sämtliche relevanten Daten, Dokumente und Änderungen
- > Vollständig
- > Nachvollziehbar
- > Unveränderbar (Datenintegrität)
- > Jederzeit überprüfbar

zu speichern.



Zugriffskontrolle und
Rollenmodelle

Audit-Trails und
Änderungsprotokolle

SWAGLab



**Warum hat Softwareentwicklung
im regulierten Umfeld so einen schlechten Ruf?**

Die Sicht der Kunden-QM im Audit...

... und die Lösung der Entwicklung

Wenn es nicht dokumentiert wurde,
ist es auch nicht gemacht worden!

- > Integriere die Erstellung bzw. Generierung von Dokumenten in den Kern des Entwicklungsprozesses $\Rightarrow$ **Model-Driven**
- > Implementiere eine tool-gestützte Traceability
- > Setze die Dokumentation und die Traceability so ein, dass sie direkt die Entwicklung, Wartung und Erweiterbarkeit unterstützt

Mit KI erstellt



Mit KI erstellt



Die Sicht der Kunden-QM im Audit...

... und die Lösung der Entwicklung

Wenn es nicht dokumentiert wurde,
ist es auch nicht gemacht worden!

Ziel:
Qualitativ hochwertige, sichere, wartbare
und jederzeit lieferbare Software!

Mit KI erstellt



Mit KI erstellt





**Die Ziele von
Regularien zu kennen, legt den Grundstein
für ein gemeinsames Verständnis
und eine konstruktive Zusammenarbeit.**

Bild mit KI generiert.

Alte Vorurteile...

- > Änderungen gehen nicht
Volle Kontrolle und Nachvollziehbarkeit über Änderungen und ihre Konsequenzen
- > Testen geht erst am Ende
TDD, BDD und alle aktuellen, parallelen, hierarchisch aufeinander aufbauenden Testverfahren wie auch explorative Tests sind sinnvoll und erlaubt
- > Agil geht erst recht nicht
Alle etablierten, dokumentierten und nachvollziehbaren Verfahren sind erlaubt

Alte Vorurteile...

> Dokumentations-lastig

Fokussierte, im Entwicklungsprozess genutzte Dokumentation + generierte Dokumente
Trenne stabile Teile von den Teilen, die sich häufig ändern ⇒ Zweck von Dokumentation

In der SOP nur den allgemeinen Prozess bzw. das Management-Framework analog zum Scrum Guide beschreiben.
Notwendige Details kommen in ein Wiki und werden in der SOP referenziert.

Alte Vorurteile und das V-Modell

> Dokumentations-lastig

Fokussierte, im Entwicklungsprozess genutzte Dokumentation + generierte Dokumente
Trenne stabile Teile von den Teilen, die sich häufig ändern ⇒ Zweck von Dokumentation

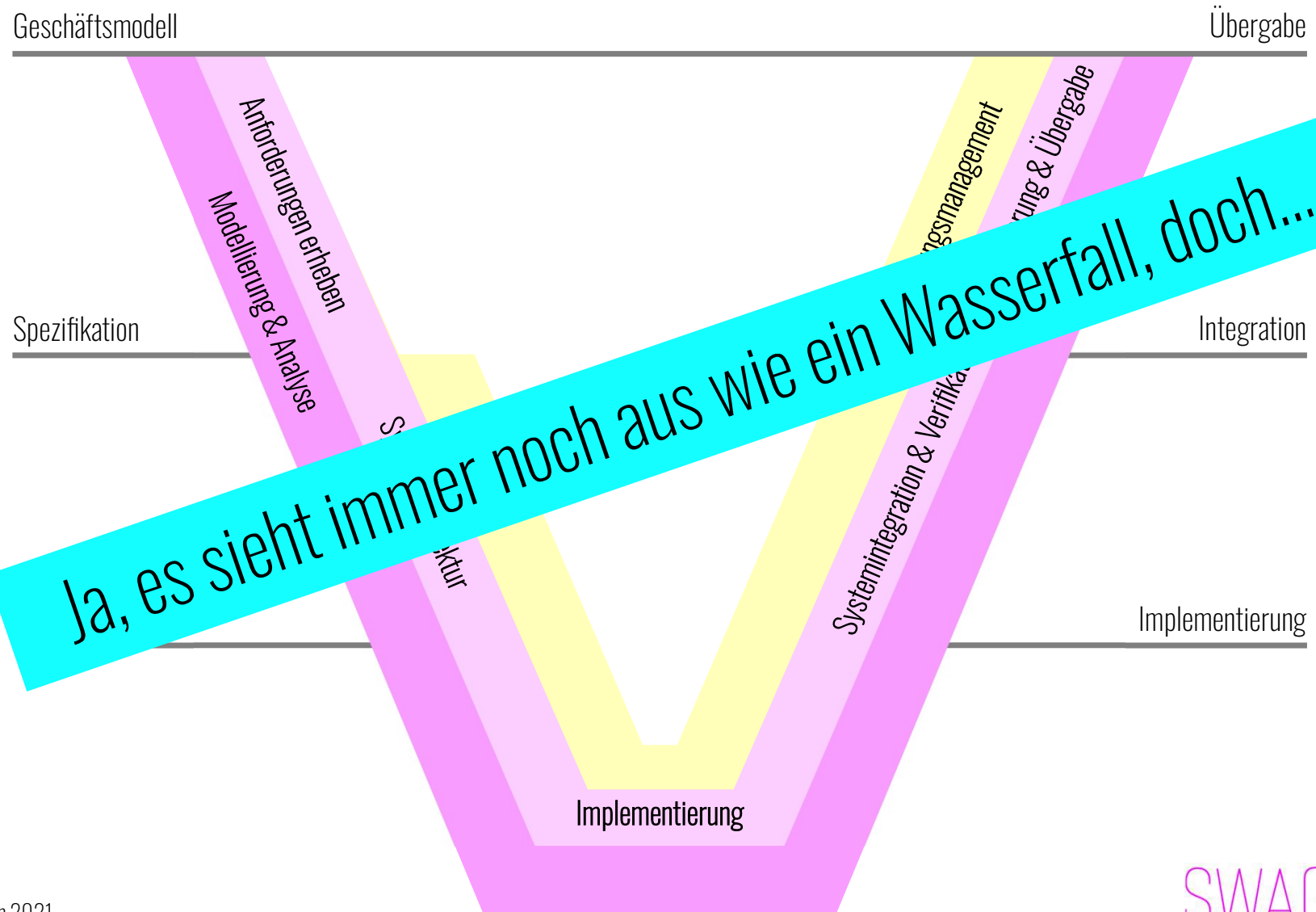
> Träge

Methodische, initiale Entwicklung, doch deutlich geringerer Aufwand für Wartung und Bugfixing
konstantere Entwicklungsgeschwindigkeit und jederzeit handlungsfähig

> V-Modell bzw. Wasserfall ...

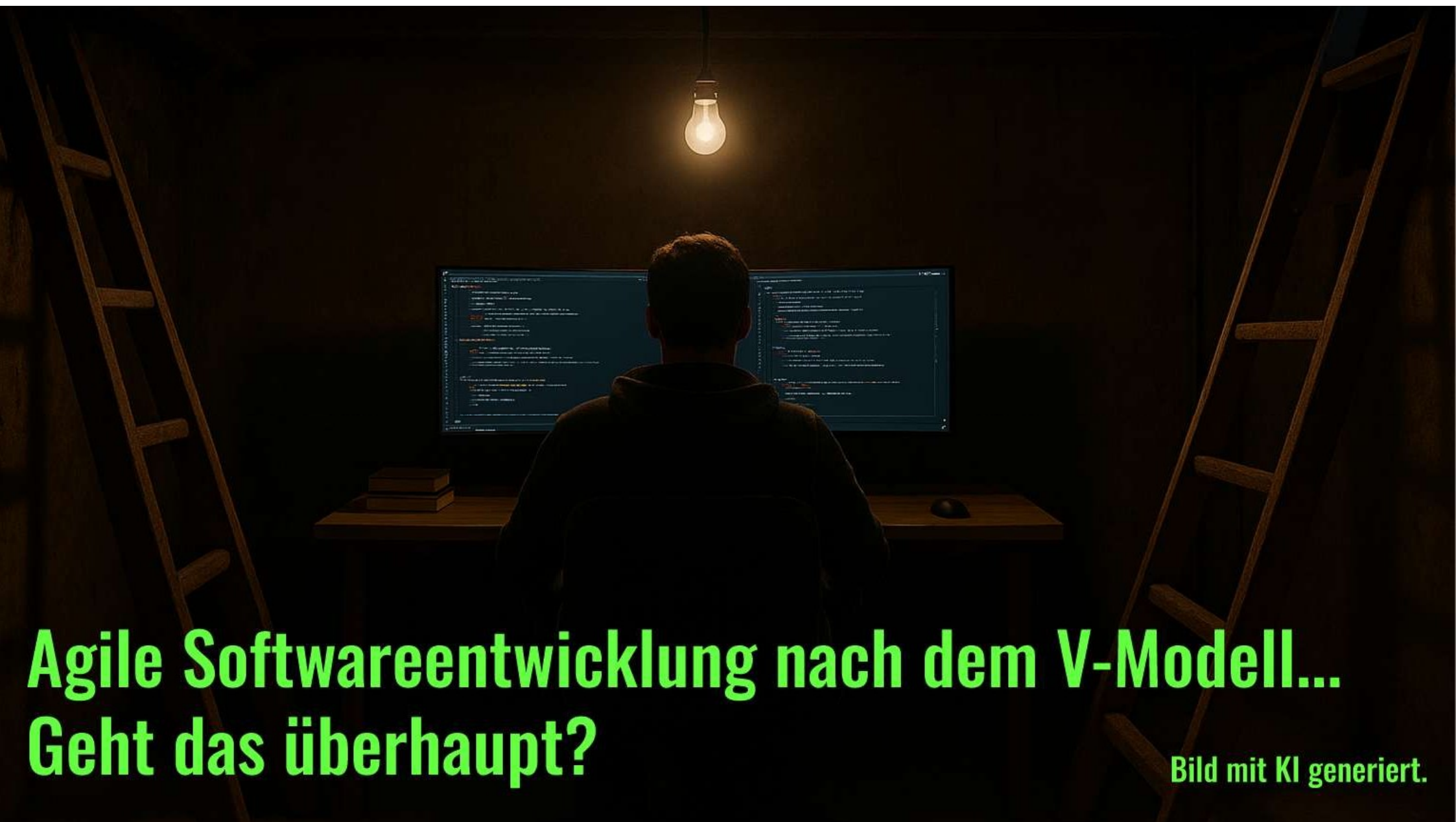
SWAGLab

V-Modell



Nach VDI/VDE 2206 von 2021

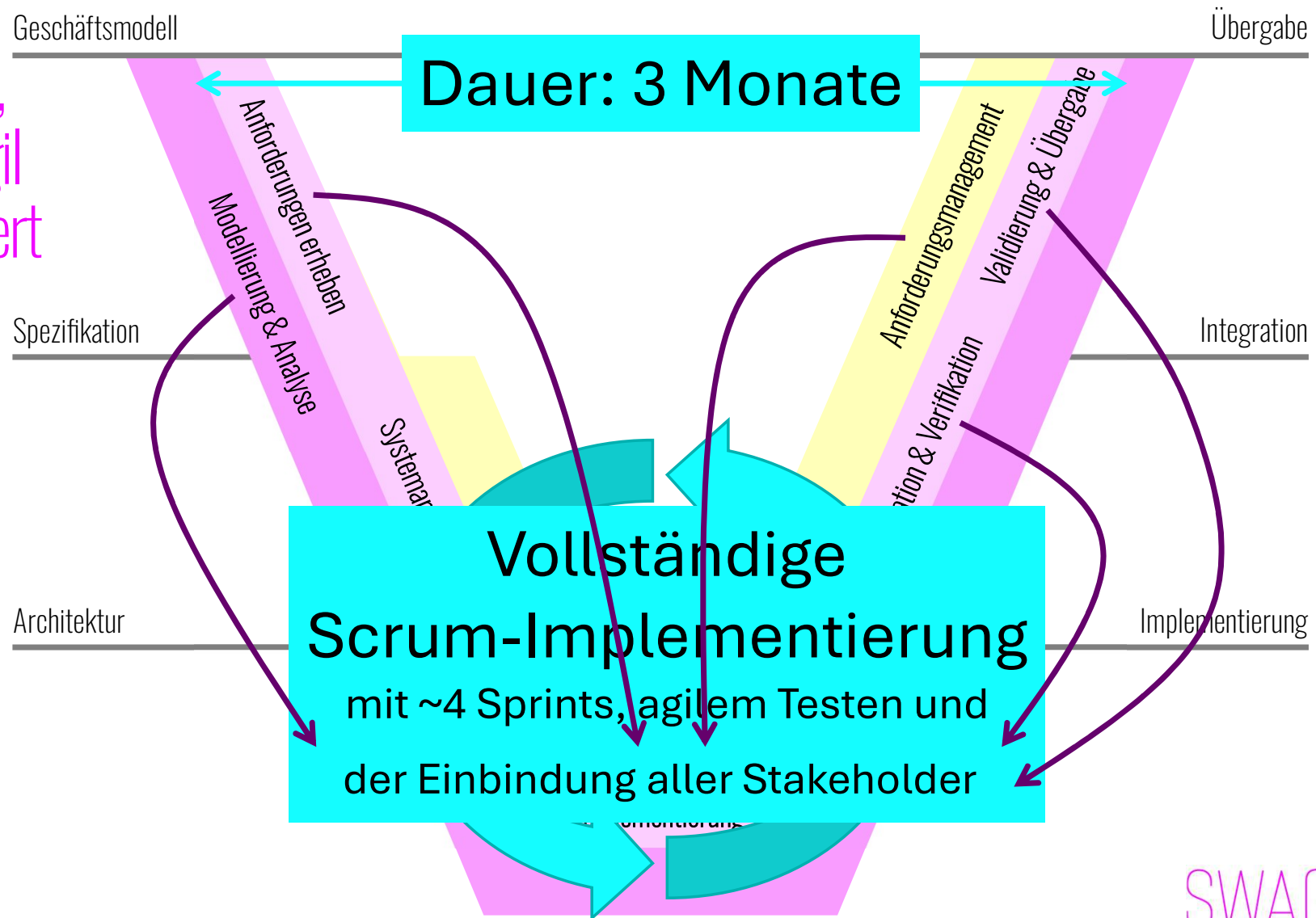
SWAGLab



Agile Softwareentwicklung nach dem V-Modell...
Geht das überhaupt?

Bild mit KI generiert.

V-Modell,
wie es agil
funktioniert



SWAGLab

Minimal notwendige Voraussetzungen

Minimiere einen Durchlauf im V-Modell auf wenige Monate – inkrementelles Vorgehen

- > Anforderungsmanagement bzw. Requirements Engineering –
vorbereitet auf die Releasezyklen der Entwicklung und die Zerlegung in User Stories
- > Planung auf zwei Ebenen als Rolling-Wave-Planning –
anstehenden Sprint ausreichend mit User Stories versorgen und Folgesprint bereits etwas grober mit User Stories vorplanen.
- > Infrastruktur, auf der entwickelt, getestet und geliefert wird, läuft stabil und automatisiert –
DevOps: erfolgt in jedem Team durch dedizierte Teammitglieder und einem DevOps-Enabling Team oder durch ein Plattform-Team.

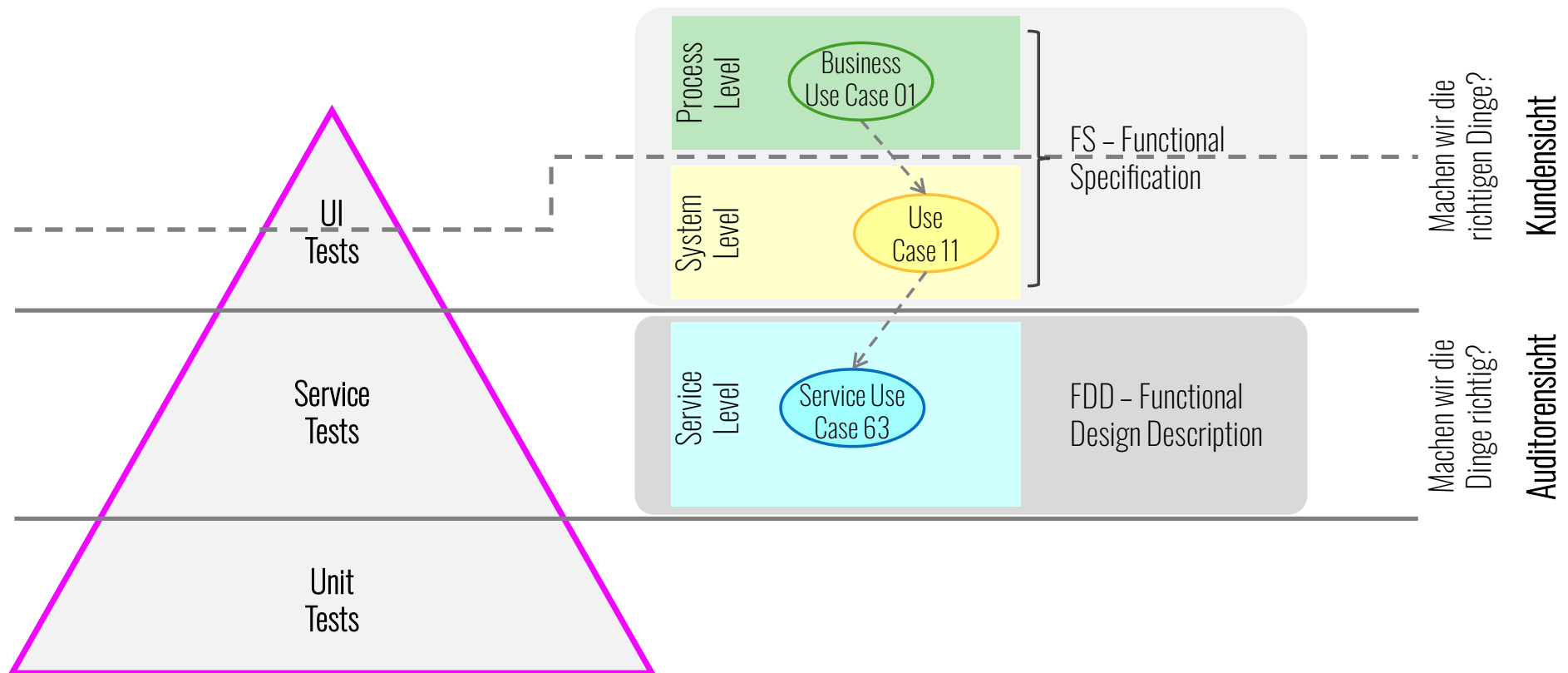
SWAGLab

Anforderungen strukturieren!

- CHECKLISTE
- ☒ Anforderungen klar verstehen & priorisieren
 - ☒ User Stories/Use Cases erfassen
 - ☐ Definition of Ready (DoR) prüfen
 - ☐ Qualitätskriterien (Akzeptanzkriterien, nicht-funktionale, nicht-funktionale) festlegen
- 2 Planung & Verfeinerung (Refinement & Story Planning)
- ☐ Anforderungen herunterbrechen
 - ☐ Aufwand & Risiken bewerten
 - ☐ Testkriterien gemeinsam mit QA definieren („Test-First“-Denken)
- Design mit Qualität im Blick (Clean Code)

Bild mit KI generiert.

Vom Test her denken, um die Anforderungen zu strukturieren



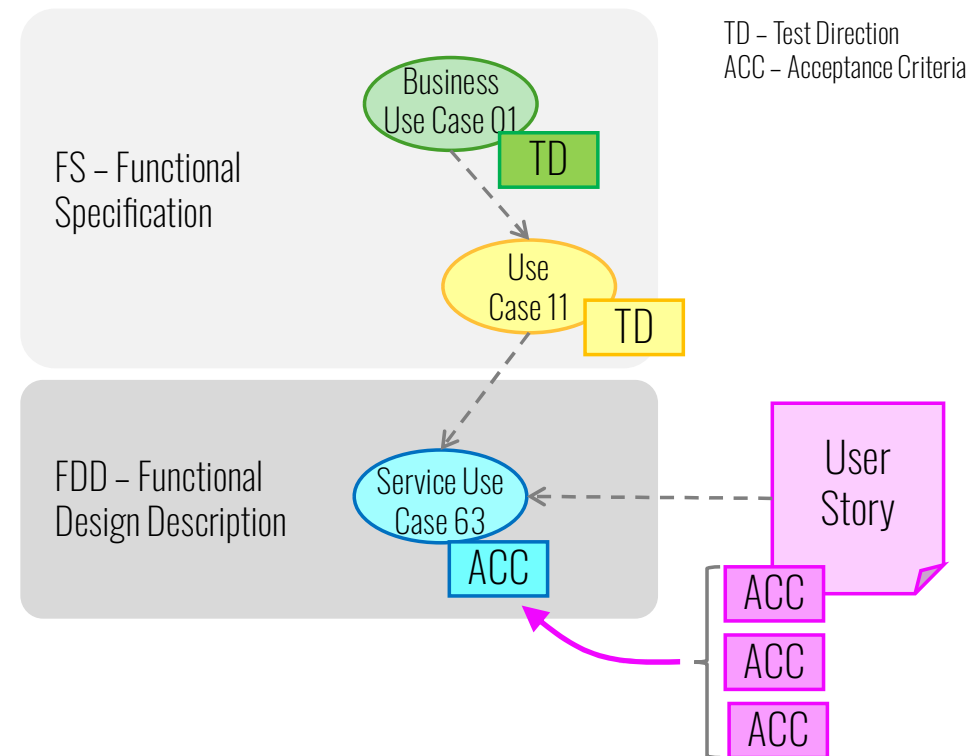
Reference: <https://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid>

SWAGLab

Funktionale Spezifikation als Dreh- und Angelpunkt

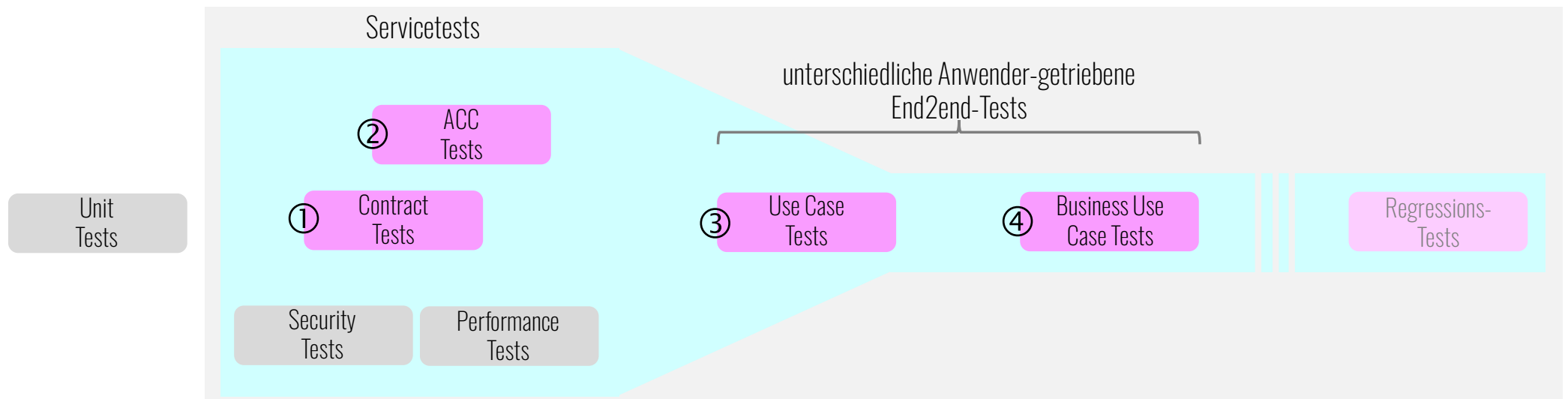
Die FS+FDD ist der Startpunkt aller Aktivitäten!

- > User Stories ableiten
- > Abhängigkeiten prüfen
- > Testfälle ableiten
 - > Exploratives Testen
 - > FS-basierte Freigabetests (FAT)
 - > Service Tests (Contract Tests und Acceptance Tests)
- > Jede Änderung der Software oder an der Umgebung wird zuerst spezifiziert!



SWAGLab

Der automatisierte Test-Workflow



SWAGLab

**Wie kommt die
Qualität in die
Software?**

Bild mit KI generiert.



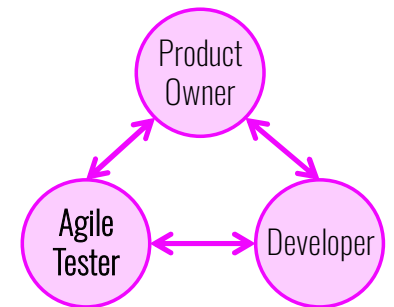
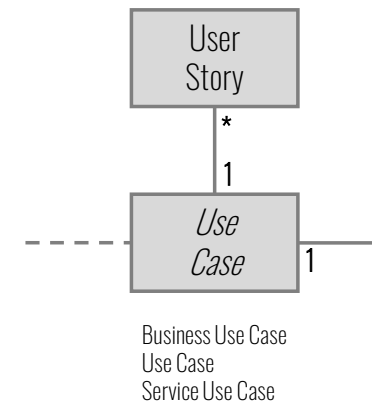
Qualität kann nicht durch formale Tests „hineingetestet“ werden!

> Weitere Rolle im Team – Agile Tester

- > Exploratives Testen
- > auf Basis von Use Case, User Story und Erfahrung
- > sowie lauffähiger Software direkt aus der Entwicklung

> Vorteile

- > Klare Aufgaben- und Rollenverteilung – Support für PO
- > Definierte Kommunikationswege (Vorbereitung für BDD)
- > Abweichungen werden früh gefunden und in der **laufenden** Entwicklung korrigiert



SWAGLab

Ist das noch agil?



Bild mit KI generiert.

Ist das noch agil? Werfen wir einen Blick in das agile Manifest!

We are uncovering better ways of developing software by doing it and helping others do it.
Through this **unterstützt** we come to value:



- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

<http://agilemanifesto.org/>

SWAGLab

Was bedeutet das für die Entwicklung im regulierten Umfeld?

Nichts anderes, als für Softwareentwicklung im nicht-regulierten Umfeld:

- > Leben wir die agilen Werte und folgen den agilen Prinzipien, sind wir agil!
- > Im regulierten Umfeld
 - > Wird ein bewusster Fokus auf die Prozesse gelegt
 - > Haben wir alle Freiheiten, agil zu sein
 - > Haben wir durch Kunden-QM und Regularien weitere Stakeholder und damit komplexere Projekte
- > Die regulatorischen Anforderungen sind Teil der **Produkt-Features!**

SWAGLab

Zusammenfassung



Kernaussagen

- > Agile Softwareentwicklung im regulierten Umfeld ist gewollt, gewünscht und machbar!
 - > Regulatorische Prozesse und dokumentatorische Anforderungen so gestalten, dass
 - > der Entwicklungsprozess sofort und
 - > der spätere Life-Cycle inkl. Wartung und Fehlerkorrekturen davon profitieren.
 - > Vier-Augen-Prinzip
 - > Nachverfolgbarkeit
 - > Reproduzierbarkeit
- Qualität entsteht im Prozess
- regulatorische Grundregeln

SWAGLab

Uwe Vogenschow

Von der Idee...

...zum Ergebnis!



Principal Consultant

Agilität und Resilienz

Qualitätssicherung und Testmanagement

Organisationsentwicklung und Lernen in Organisationen

Interimsmanagement



Uwe Vigerschow – Wir bleiben in Kontakt!



Linked-In

SWAGLab