BEST-PRACTICE-ANLEITUNG · TRANSFORM

Phase 05 — Build

Von der Spezifikation zum produktionsreifen Inkrement

In der Build-Phase implementieren Multi-Agenten-Systeme die Ziellösung auf Basis der qualitätsgesicherten Spezifikationen aus Analyse und Design. Beraterinnen und Berater steuern die Agenten, sichern die Qualität über Reviews und Policies und halten die DORA-Metriken der Lieferstrecke auf Elite-Niveau.

Ziel der Phase

Ergebnis sind produktionsreife, getestete Inkremente mit vollständiger Traceability von der Anforderung bis zum Code — geliefert in kurzen Zyklen über eine trunk-basierte, vollautomatisierte Integrationsstrecke.

Vorgehensmodell

01 **Delivery-Setup & Definition of Done**
Aufsetzen der agentischen Lieferstrecke (Repositories, CI, Policies, Umgebungen); Vereinbarung von Definition of Done, Codierungs- und Review-Standards mit dem Kundenteam.

02 **Spezifikationsgetriebene Agenten-Implementierung**
Zerlegung der Epics in agentenfähige Arbeitspakete mit präzisen Akzeptanzkriterien; Multi-Agenten-Codegenerierung gegen die Verträge und ADRs der Design-Phase.

03 **Kontinuierliche Integration & Review**
Trunk-based Development mit kleinen Batches; jede Änderung durchläuft automatisierte Prüfungen und ein menschliches Review der fachlich kritischen Pfade.

04 **Qualitäts- & Sicherheits-Gates**
Statische Analyse, Dependency- und Secret-Scanning, Architektur-Konformanzprüfung gegen die Zielarchitektur; Policy-Gates blockieren nicht-konforme Änderungen.

05 **Mess- & Lernschleife**
Kontinuierliche Erhebung der DORA-Metriken und der Agenten-Effektivität; Retrospektiven zur Verbesserung von Prompts, Kontexten und Arbeitspaket-Schnitt.

Methodik-Mix

METHODE	EINSATZZWECK	REFERENZ
Trunk-Based Development	Kleine, kurzlebige Branches und tägliche Integration; Voraussetzung für agentische Liefergeschwindigkeit ohne Merge-Konflikte.	trunkbaseddevelopment.com
Continuous Delivery	Deployment-Pipeline mit automatisierten Qualitätsstufen; jede Änderung ist potenziell releasefähig.	Humble & Farley (2010)
Test-Driven Development	Testgetriebene Spezifikation der fachlich kritischen Logik; Akzeptanztests als ausführbare Anforderungen für die Agenten.	Beck (2002)

METHODE	EINSATZZWECK	REFERENZ
DORA-Metriken	Deployment Frequency, Lead Time, Change Failure Rate und MTTR als empirisch validierte Steuerungsgrößen der Lieferstrecke.	Forsgren, Humble & Kim (2018)
Clean Code & SWEBOK	Verbindliche Codierungs- und Wartbarkeitsstandards; SWEBOK als Referenzrahmen der Software-Engineering-Disziplinen.	IEEE CS (SWEBOK v4)
ReqPOOL Agentic Engineering Playbook	Hausinterner Standard für Arbeitspaket-Schnitt, Kontext-Engineering, Agenten-Reviews und Mensch-Agent-Arbeitsteilung.	ReqPOOL (intern); vgl. Anthropic (2024)

Plattform-Unterstützung

ReqPOOL Suite: **reqCoder** (KI-Softwareentwicklung) · **Estimation Manager** (Schätzen · Scopen · Steuern)

reqCoder — agentische Implementierung (Coming soon)	EU-basierte KI entwickelt die Software direkt aus dem AI Coding Intent — kein Training mit Kundendaten, lückenloser Audit-Trail; menschliche Engineers prüfen, härten und integrieren den generierten Code.
Specification as Code als Arbeitsgrundlage	Der KI-generierte Implementierungsauftrag aus dem Estimation Manager (Masken, Use Cases mit Akzeptanzkriterien, Geschäftsobjekte, Schnittstellen, Rechte-Matrix) dient als direkte Arbeitsgrundlage der agentischen Entwicklung.
Estimation Manager — Scope-Steuerung im Build	Scope-Änderungen während der Implementierung werden als versionierte Change Requests erfasst; Scope-Drift wird laufend gegen die Baseline gemessen.
Traceability über IDs	Anforderungs-IDs aus Spezifikation und Coding Intent verknüpfen Arbeitspakete, Code und Testfälle — lückenlose Nachweisführung für Audits und Quality Gates.

Best-Practice-Checkliste

- [ ] Arbeitspakete entlang der Use Cases des AI Coding Intent schneiden — klarer Kontext, ausführbare Akzeptanzkriterien, begrenzter Blast-Radius.
- [ ] Akzeptanzkriterien vor der Generierung als Tests formulieren; die KI implementiert gegen ausführbare Anforderungen.
- [ ] Menschliche Reviews auf fachlich kritische Pfade und Architekturgrenzen fokussieren; generierten Code nie ungeprüft integrieren.
- [ ] Jede Scope-Änderung sofort als Change Request im Estimation Manager erfassen — Scope-Drift wird gegen die Baseline gemessen (Gate: Code complete).
- [ ] DORA-Metriken wöchentlich im Team reviewen; Verschlechterungen als Impediment behandeln, nicht als Berichtsgröße.
- [ ] Prompts und Kontextpakete versionieren und in Retrospektiven verbessern — Agenten-Effektivität ist ein Engineering-Artefakt.

Artefakte & Ergebnisse

— Produktionsreife, getestete Inkremente je Release

— Vollständige Traceability Anforderung → Code → Test
— Automatisierte Integrations- und Qualitätsstrecke
— Migrierte und modernisierte Legacy-Komponenten
— DORA-Metrik-Baseline und Delivery-Report

Quality-Gate-Kriterien

☐	Alle Inkremente erfüllen die Definition of Done
☐	Policy-Gates ohne offene Ausnahmen bestanden
☐	Traceability-Kette lückenlos nachgewiesen
☐	DORA-Metriken im vereinbarten Zielkorridor

Wissenschaftliche Vertiefung

Forsgren, Humble & Kim: Accelerate (2018) — Empirische Grundlage der DORA-Metriken und Hochleistungs-Delivery.
https://itrevolution.com/product/accelerate/

DORA Research Program — Laufende Forschung zu Software-Delivery-Performance inkl. KI-Einfluss.
https://dora.dev/

Humble & Farley: Continuous Delivery (2010) — Standardwerk der Deployment-Pipeline und Release-Automatisierung.
https://continuousdelivery.com/

Beck: Test-Driven Development by Example (2002) — Grundlagenwerk des testgetriebenen Vorgehens.
https://www.informit.com/store/test-driven-development-by-example-9780321146533

IEEE CS: SWEBOK Guide v4 — Body of Knowledge des Software Engineering.
https://www.computer.org/education/bodies-of-knowledge/software-engineering

Anthropic: Building Effective Agents (2024) — Entwurfsmuster für zuverlässige agentische Systeme.
https://www.anthropic.com/research/building-effective-agents