**Factsheet | Enterprise Unified Process (EUP)**

# **Kurze Geschichte zum EUP und die Namensherkunft**

# Der Enterprise Unified Process (EUP) wurde im Jahr 2000 von Scott W. Ambler und Larry Constantine als Erweiterung des Unified Process (UP) entwickelt. Später wurde er durch das Disciplined Agile Delivery (DAD) Toolkit ersetzt und 2019 vom Project Management Institute (PMI) übernommen.

# **Funktionsweise und Einteilung ob klassisch oder agil**

# Der Enterprise Unified Process (EUP) ist eine Weiterentwicklung des Unified Process (UP) und basiert auf einem schrittweisen und wiederholenden Ansatz.

# Im Gegensatz zum UP, der sich nur auf die Softwareentwicklung konzentriert, betrachtet der EUP den gesamten Lebenszyklus eines IT-Systems – von der Planung über die Entwicklung bis hin zur Wartung und Außerbetriebnahme.

# **Wofür geeignet auch in Bezug auf Projektumfang, Verwendungszweck der Software, Teamgröße, Erfahrung, ...**

Der Enterprise Unified Process (EUP) eignet sich besonders für langfristige und komplexe Projekte, da er die Wartung und Erweiterung von Software im Vergleich zum klassischen Unified Process (UP) strukturiert und nachhaltig gestaltet.

Vor allem große erfahrene Entwicklergruppen bevorzugen diese Methode, da sie gegenüber einer monolithischen, sequenziellen Programmierung (bei der der gesamte Code in einer einzigen Datei liegt) besser wartbar und erweiterbar ist.

Durch Modularisierung und objektorientierte Programmierung (OOP) lassen sich Fehler schneller finden und gezielt beheben.

# **Vor- und Nachteile**

**🎯 Fazit: Was macht EUP-freundlichen Code aus?**

✅ Modularisierung → Code in kleine, unabhängige Teile aufteilen

✅ OOP statt rein sequenzieller Programmierung → Klassen für bessere Wartbarkeit

✅ Design Patterns nutzen → Bewährte Lösungen für Wartbarkeit einsetzen

✅ Microservices statt Monolithen → Flexibilität erhöhen

# **Möglichkeit der Umsetzung (z.B. auf/mit welcher Plattform, Webseite, Software kann das Modell umgesetzt werden)**

Der Enterprise Unified Process (EUP) kann mit verschiedenen Tools und Plattformen umgesetzt werden, die Projektmanagement, Softwareentwicklung, Architekturplanung und langfristige Wartung unterstützen (z. B. Jira, GitHub, Enterprise Architect, Jenkins).