

W-JAX 2026, München



Business Case für die Teilnahme einer Entwicklerin oder eines Entwicklers

2. - 6. November 2026 | The Westin Grand Munich | Java, Software-Architektur & Enterprise AI

Zusammenfassung

Die W-JAX 2026 fällt in ein Umsetzungsfenster, in dem KI-gestützte Entwicklung vom persönlichen Coding-Assistenten zur **agentenbasierten Team-Praxis** wird. Die Keynote "AI Coding war gestern. Jetzt kommen die Agenten!" trifft den Kern: Wenn Agenten Entwicklungsaufgaben übernehmen, werden Spezifikationen, Architektur, Tests und Nachvollziehbarkeit zum Engpass - nicht mehr die Code-Erzeugung.

Das Programm der W-JAX konzentriert sich auf Enterprise AI, Software-Architektur und AI-gestützte Softwareentwicklung - genau die Themen, die gerade vom Experiment in den produktiven Einsatz übergehen. Parallel werden AI Act und Cyber Resilience Act zu konkreten Anforderungen. Gegenüber reinem Selbststudium bietet ein Konferenzbesuch vor allem Verdichtung: konkrete Patterns, Erfahrungswerte und Gegenpositionen aus Architektur, AI Engineering, Security und Modernisierung lassen sich in wenigen Tagen vergleichen - statt sie über Einzelquellen und eigene Versuche zusammenzutragen.

Angefragt wird: Konferenzteilnahme inklusive Enterprise AI Day bzw. relevanter Workshop-Tage, Reise und Unterkunft. Im Gegenzug bringe ich die relevanten Erkenntnisse in einem internen Recap ins Team zurück. Die Investition in Ticket, Reise und Arbeitszeit soll Lern- und Entscheidungszyklen verkürzen, die sonst im laufenden Projekt stattfinden. **Der Nutzen:** weniger Trial-and-Error bei Agenten, geringere Review- und Rework-Kosten, bessere Compliance-Fähigkeit und risikoärmere Modernisierung.

1. Timing: aus AI Coding wird Agentic Engineering

Der Enterprise AI Day und die Programmschwerpunkte zeigen: 2026 geht es nicht mehr primär darum, ob Entwicklerinnen und Entwickler KI nutzen, sondern **wie daraus ein reproduzierbarer Teamprozess wird**. Jetzt ist der Zeitpunkt, gemeinsame Standards zu setzen, bevor sich pro Person eigene Tools, Prompts und Qualitätsmaßstäbe verfestigen.

- **Wiederholbarer Workflow:** Der Workshop "Mit KI und Agenten entwickeln" führt von Spezifikation und Planung über Implementierung und Tests bis zur Dokumentation - mit LLMs, MCP, Skills und autonomen Workflows.
- **Quality Gates & Produktionsreife:** "Harness Engineering" koppelt Agenten an technische und fachliche Regeln; Observability und Evaluation ersetzen "Vibe Checks" durch messbare Kriterien.

Einordnung: Wenn wir KI-gestützte Entwicklung produktiv einsetzen oder evaluieren, ist die teure Komponente nicht das Tool-Abo, sondern die Lernkurve im laufenden Projekt. Die W-JAX verlagert einen Teil dieser Lernkurve aus dem Projekt in ein kompaktes Weiterbildungsformat.

2. Risiko & Compliance: Pflichten in technische Kontrollen übersetzen

Der AI Act ist seit 2. August 2026 grundsätzlich anwendbar; High-Risk-Regeln folgen nach aktueller EU-Zeitleiste 2027/2028. Beim Cyber Resilience Act beginnen die Meldepflichten am 11. September 2026, die vollständige Anwendung folgt 2027. Das ist ein **Implementierungsfenster**: Auditierbarkeit, Vulnerability Management und Sicherheitsgrenzen sollten technisch vorbereitet werden, bevor sie nachträglich teuer eingebaut werden.

Thema	Konkretes Risiko	Was die Session liefert
Secure Java Software Supply Chain	Unklare Dependencies, CVEs und fehlende SBOM-/Remediation-Prozesse	Automatisierte CVE-Erkennung und -Behebung, SBOM, Redeployment; direkte Verbindung zum CRA
Threat Modeling für KI-Agenten	RAG-Poisoning, indirekte Prompt Injection und MCP-Tool-Chain-Missbrauch	Fünf-Zonen-Modell, Trust Boundaries und Architekturkontrollen für agentische Systeme
Architecting for the EU AI Act	Auditierbarkeit, Human Oversight und Daten-Governance werden zum Retrofit	Muster für Logging, Isolation, Modellversionierung und menschliche Eingriffe

Einordnung: Ergänzt um die OWASP Top Ten 2025 übersetzt das W-JAX-Programm Rechts- und Sicherheitsanforderungen in konkrete Architektur- und Pipeline-Maßnahmen. Rechtsberatung ersetzt es nicht.

3. Kosten & Lieferfähigkeit: Review, Rework und Incidents begrenzen

Agenten können Code schneller erzeugen, als Menschen ihn prüfen können. Ohne Leitplanken verschiebt sich der Produktivitätsgewinn in Review-Backlogs, technische Schulden und Produktionsfehler. Genau diese **Folgekosten von KI-Code** adressiert das Konferenzprogramm.

Workflow / Problem	Wirkung auf unsere Lieferfähigkeit
Harness Engineering	Backpressure und Outcome Grading prüfen Agentenoutput automatisiert, bevor menschliche Review-Zeit gebunden wird.
Observability & Evaluation	Messbare Kriterien ersetzen "Vibe Checks"; Error Analysis verkürzt Fix-Loops zwischen Prototyp und Produktion.
Ausführbarer Architekturvertrag + lesbare Tests	Rules, Skills, Architekturprüfungen und verständliche Tests verhindern Codebasis-Erosion und machen Verhalten überprüfbar.
FeatureOps / Kill Switches	Feature Flags, progressive Rollouts und Kill Switches begrenzen den Schaden fehlerhafter Releases im Betrieb.

Einordnung: Der wirtschaftliche Nutzen KI-gestützter Entwicklung entsteht nicht durch mehr Codezeilen, sondern wenn Review- und Fehlerkosten nicht proportional mit dem Output wachsen. Weiterbildung ist hier ein Hebel, um diese Folgekosten früh zu begrenzen.

4. Modernisierung: bestehende Systeme ohne Big Bang weiterentwickeln

Ein auffällig großer Teil des Programms beschäftigt sich mit Brownfield- und Legacy-Systemen. Für Unternehmen ist das häufig relevanter als Greenfield-Demos: Der größte Softwarewert steckt in Systemen, die **weiterlaufen müssen, während sie verändert werden**.

Ansatz	Was wir daraus mitnehmen
Strangler Pattern	Schrittweise Ablösung, Parallelbetrieb und Datenmigration ohne Big-Bang-Risiko.
AI Meets Legacy	Screen-by-Screen-Migration mit Build-/Rollback-Schleife, während der Geschäftsbetrieb weiterläuft.
LLM-Assisted Migrations / reale KI-Migrationen	Inkrementelles Refactoring, Testabdeckung, Dokumentation und Wissensaufbau statt einmaligem Rewrite.
Agentic Software Modernization	Seams, Slicing, Anker und Fitness Functions begrenzen den Aktionsraum von Coding Agents im Brownfield.

Einordnung: Diese Muster schützen bestehende Softwareinvestitionen und machen Modernisierung in kleineren, testbaren Schritten budgetierbarer. Der Nutzen liegt damit nicht nur in neuen Technologien, sondern auch im risikoärmeren Umgang mit dem Bestand.

Quellen

Programmgrundlage: [W-JAX-Programmliste](#), Stand 27.08.2026. Für die Argumentation herangezogene Sessions:

- [AI Coding war gestern. Jetzt kommen die Agenten!](#)
 - [Mit KI und Agenten entwickeln: Workshop für Moderne Softwareentwicklung mit OpenCode, MCP und autonomen Workflows](#)
 - [Harness Engineering: Der Druckreaktor des Agentic Engineering](#)
 - [Wenn die Vibes nicht reichen: Observability und Evaluation für verlässlichere GenAI-Anwendungen](#)
 - [Building a Secure Java Software Supply Chain in the Era of the Cyber Resilience Act](#)
 - [Vier grüne Häkchen, trotzdem gehackt: Threat Modeling für KI-Agenten](#)
 - [Architecting for the EU AI Act: Compliance as a Quality Attribute](#)
 - [Web-Security up-to-date: Die OWASP Top Ten 2025](#)
- [Architektur als ausführbarer Vertrag - Wie Coding Agents Ihre Frontend-Architektur respektieren](#)
 - [KI diktiert aber Du haftest: Dein Schutzschild im Code Generierungs-Dschungel](#)
 - [FeatureOps Workshop: From First Flag to Runtime Control](#)
 - [Legacy Software mit Strangler Pattern ablösen](#)
 - [AI Meets Legacy, Desktop-to-Web Modernization Without Interrupting the Business](#)
 - [LLM-Assisted Migrations and Modernizations: from Museum to Mainstream](#)
 - [Agentic Software Modernization: Back to the Roots](#)