

Behörde / Institution
byte - Bayerische Agentur für Digitales GmbH
Projektname
Code Inspector
Verwaltungsebene
Land
Website / URL
https://byte.bayern
Einreichungskategorie
Künstliche Intelligenz
Möchten Sie das Projekt noch in einer zweiten Kategorie einreichen?
Nein
Projektbeschreibung
Die byte hat mit dem AppDev-Standard eine Checkliste entwickelt, die Anforderungen aus dem BSI IT-Grundschutz und weiteren Vorgaben für Softwareentwicklungen in der öffentlichen Verwaltung bündelt. Darauf aufbauend entstand der Code Inspector, ein Open-Source-Werkzeug, das Code-Repositories automatisiert auf die Einhaltung dieser Anforderungen prüft. Der Code Inspector analysiert den Quellcode, bewertet die Erfüllung der Anforderungen, liefert nachvollziehbare Begründungen und generiert konkrete Verbesserungsvorschläge. Dadurch lassen sich Sicherheitslücken und Qualitätsmängel frühzeitig erkennen und der Aufwand für manuelle Prüfungen deutlich reduzieren. Die Ergebnisse werden anschließend durch qualifizierte Softwareentwickler validiert. Das aktuelle System prüft automatisiert bis zu 40 Kernanforderungen des AppDev-Standards und erreicht dabei eine verifizierte Trefferquote von mindestens 80 Prozent. Es lässt sich direkt in CI/CD-Pipelines, beispielsweise GitLab, integrieren und ist sowohl für Cloud- als auch On-Premises-Umgebungen ausgelegt. Die modulare Architektur ermöglicht zudem die Erweiterung um weitere Anforderungskataloge. Perspektivisch soll der Code Inspector alle automatisierbaren Anforderungen des AppDev-Standards abdecken, um eine Benutzeroberfläche ergänzt und für weitere Entwicklungsstandards geöffnet werden. Als Open-Source-Projekt kann die Lösung von anderen Behörden nachgenutzt und gemeinsam weiterentwickelt werden.
Beschreiben Sie den technischen Innovationsgrad des Projektes und den Beitrag zur Verwaltungsmodernisierung.
Der Code Inspector verfolgt einen innovativen Ansatz zur Compliance-Prüfung von Software-Projekten. Kern der Lösung ist der Einsatz von KI-Agenten, was aktuellen technologischen Entwicklungen entspricht. Im Unterschied zu statischen Analysewerkzeugen passen sich KI-Agenten dynamisch an Projektstrukturen an und erschließen relevante Zusammenhänge eigenständig. Der KI-Agent entscheidet selbstständig über geeignete Prüfschritte, statt ausschließlich vordefinierten Regeln zu folgen. Dadurch kann er tiefgreifende Analysen durchführen, Anforderungen kontextbezogen bewerten und individuelle Besonderheiten eines Software-Projekts berücksichtigen. Gleichzeitig reduziert sich der Aufwand zur Integration neuer Anforderungskataloge, da nicht für jede Anforderung individuelle statische Prüfschritte implementiert werden müssen. Die Lösung wurde zudem so konzipiert, dass Sicherheit und Datenschutz („security by design“ und „privacy by default“) von Beginn an berücksichtigt werden. Der Zugriff auf Repositories erfolgt über bestehende Rollen- und Berechtigungskonzepte der GitLab-Umgebung, so dass ausschließlich autorisierte Zugriffe möglich sind. Für die Analyse wird das Repository lediglich geklont; Änderungen am eigentlichen Repository erfolgen nicht. Anschließend werden die Commit-Historien entfernt und der Klon nach Abschluss der Analyse verworfen, sodass keine unnötigen personenbezogenen Daten verarbeitet oder dauerhaft gespeichert werden. Der Code Inspector leistet einen direkten Beitrag zur Verwaltungsmodernisierung. Er

unterstützt die frühzeitige Identifikation sicherheitsrelevanter Schwachstellen und verbessert so die Softwarequalität über den gesamten Entwicklungsprozess hinweg. Die automatisiert erzeugten Prüfberichte schaffen erstmals die Möglichkeit, die Umsetzung beauftragter Sicherheits- und Qualitätsanforderungen KI-gestützt und nachvollziehbar nachzuweisen. Der Code Inspector sorgt damit für effiziente, standardisierte und nachvollziehbare Software-Abnahmeprozesse.

Welchen ökonomischen Nutzen hat das Projekt?

Der Code Inspector bietet einen erheblichen ökonomischen Nutzen, der weit über einzelne Software-Projekte hinausgeht. Der größte wirtschaftliche Hebel liegt in der Open-Source-Bereitstellung für den Gesamtstaat: Wenn die Prüfinfrastruktur in Bayern als zentrales Werkzeug entwickelt wird, entfallen die initialen Entwicklungs- und Konzeptionskosten für andere Bundesländer und den Bund komplett. Dieser volkswirtschaftliche Skaleneffekt vermeidet teure Doppelentwicklungen, bündelt öffentliche Ressourcen und ermöglicht eine flächendeckende Nachnutzung.

Auf Ebene der einzelnen Projekte werden Prüf- und Abnahmeprozesse effizient, standardisiert und nachvollziehbar gestaltet. Durch die automatisierte Analyse von Code-Repositories können Sicherheits- und Qualitätsanforderungen deutlich schneller überprüft werden als in rein manuellen Verfahren. Dies reduziert den zeitlichen und personellen Aufwand bei Abnahmen signifikant. Fachpersonal wird entlastet und kann gezielt für Validierungen sowie komplexe Bewertungen eingesetzt werden.

Ein weiterer wirtschaftlicher Vorteil liegt in der frühzeitigen Identifikation von Sicherheitslücken, Qualitätsmängeln und Abweichungen von definierten Anforderungen. Probleme können bereits während der Entwicklung erkannt und behoben werden, bevor sie in späteren Projektphasen zu erhöhtem Nachbesserungsaufwand führen. Dadurch lassen sich Fehlerkosten reduzieren und Verzögerungen im Projektverlauf minimieren.

Durch die Konsolidierung relevanter Sicherheits- und Qualitätsanforderungen im AppDev-Standard sowie deren automatisierte Überprüfung schafft der Code Inspector eine einheitliche und wiederverwendbare Prüfgrundlage für unterschiedliche Projekte. Die Standardisierung und Nachvollziehbarkeit der Prüfprozesse verbessert die Planbarkeit von Entwicklungs- und Abnahmephase und reduziert Abstimmungsaufwände zwischen Auftraggebern und Dienstleistern.

Beschreiben Sie die Nachhaltigkeit der Lösung.

Der Code Inspector verfolgt einen nachhaltigen Ansatz, ausgelegt auf langfristige Nutzbarkeit, Erweiterbarkeit und maximale Interoperabilität. Statt starrer Prüf-Pipelines nutzen wir autonome KI-Agenten, die sich dynamisch an unterschiedliche Projektstrukturen anpassen. Dies macht die Lösung resilient gegenüber technologischen Paradigmenwechseln.

Die skalierbare Architektur erlaubt eine nahtlose Ergänzung neuer Prüfkriterien. Regulatorische Vorgaben lassen sich als strukturierte Prompts integrieren, ohne den Python-Quellcode tiefgreifend anpassen zu müssen. Das System wächst somit kontinuierlich mit den fachlichen Anforderungen.

Hinsichtlich Wiederverwendbarkeit und digitaler Souveränität setzt die Anwendung auf eine strikte Technologieunabhängigkeit. Die konsequente API-First-Architektur und vollständige Containerisierung (inklusive bereitgestelltem Helm Chart) gewährleisten einen plattformunabhängigen Betrieb – sowohl in modernen Cloud-Umgebungen als auch streng abgeschottet On Prem. Um Vendor-Lock-in zu vermeiden, verzichtet die Architektur bewusst auf proprietäre Managed Services. Stattdessen setzt das System auf quelloffene Standardprotokolle wie S3 für den Objektspeicher oder das Redis-Protokoll (implementiert via Valkey) für die asynchrone Task-Verteilung.

Diese Interoperabilität setzt sich bei allen Kernschnittstellen fort: Die Anwendung ist agnostisch gegenüber der zugrundeliegenden Git-Plattform und akzeptiert jedes standardkonforme Token, das einen Git-Clone ermöglicht. Auch die Anbindung der KI-Modelle ist vollständig abstrahiert. Durch diesen provider-agnostischen Ansatz kann nahtlos zwischen Cloud-Modellen

und lokal gehosteten LLMs gewechselt werden – ein kompletter Provider-Wechsel erfordert lediglich die Anpassung von Umgebungsvariablen. Einmal definierte Prüfanforderungen können so von unterschiedlichen Behörden, unabhängig von deren spezifischer IT-Infrastruktur, als Grundlage für Sicherheits- und Qualitätsprüfungen nachgenutzt werden.

Wie trägt das Projekt zur Stärkung der Digitalen Souveränität bei?

Der Code Inspector stärkt die digitale Souveränität der öffentlichen Verwaltung, indem er technische Kontrolle und eine unabhängige Weiterentwicklung konsequent unterstützt. Der Code Inspector kann sowohl in Rechenzentren der öffentlichen Verwaltung als auch in Private-Cloud- oder On-Premises-Umgebungen gehostet werden. Es können bestehende Repository- und Entwicklungsplattformen sowie eigene Large-Language-Models angebunden werden. Dadurch behalten Behörden und öffentliche IT-Dienstleister die Kontrolle über ihre Daten, Prüfprozesse und eingesetzten Technologien. Sensibler Quellcode muss nicht an externe Plattformen oder Drittdienstleister übertragen werden, wodurch die Datenhoheit gewahrt bleibt.

Der Code Inspector kann außerdem unabhängig innerhalb der öffentlichen Verwaltung weiterentwickelt werden. Grundlage dafür ist die geplante Bereitstellung der Anwendung als Open-Source-Software auf OpenCoDE, wodurch die Funktionsweise des Systems transparent eingesehen, verstanden und überprüft werden kann. Mittels eines Contributor License Agreement (CLA) wird sichergestellt, dass Contributions rechtlich gesichert ins Hauptprojekt einfließen.

Der Open-Source-Ansatz ermöglicht es Behörden, öffentlichen IT-Dienstleistern und weiteren Beteiligten, die Lösung kollaborativ weiterzuentwickeln oder individuell an eigene Bedürfnisse anzupassen. Abhängigkeiten von Herstellern und proprietären Lösungen werden drastisch reduziert. Die automatisiert erzeugten Prüfberichte tragen zum Wissensaufbau innerhalb der öffentlichen Verwaltung bei. Sicherheits- und Qualitätsanforderungen werden strukturiert dokumentiert und nachvollziehbar bewertet. Fachliches Know-how verbleibt dadurch nicht ausschließlich bei externen Dienstleistern, sondern kann innerhalb der Organisationen aufgebaut, genutzt und langfristig weiterentwickelt werden.

Projektwebsite

<https://byte.bayern/was-wir-machen/code-inspector>

Umsetzungspartner

TNG Technology Consulting GmbH

Zusätzliche Dokumente

Keine