



BOSCH

Whitepaper

Digitale Transformation: Optimierung des Software- Freigabeprozesses in der Automobil- und Finanz- industrie



Einleitung

In hochregulierten Domänen wie der Finanz- und Automobilbranche ist die Freigabe von Software ein komplexer und zeitaufwendiger Prozess. Die Einhaltung strenger Vorschriften und die Gewährleistung der Sicherheit und Zuverlässigkeit von Software-Lösungen sind von entscheidender Bedeutung, um das Vertrauen der Kunden und die Einhaltung gesetzlicher Bestimmungen zu gewährleisten. Zudem hat die Komplexität von Software-Lösungen in den vergangenen Jahren deutlich zugenommen, während die Produktivität in der Software-Entwicklung zu stagnieren scheint.

Um diese Komplexität zu bewältigen und die Produktivität in der Software-Entwicklung zu steigern, setzen Unternehmen auf verschiedene Ansätze und Technologien. Dazu zählt beispielsweise die Automatisierung von Entwicklungs- und Bereitstellungsprozessen.

Dieses Whitepaper beleuchtet die Herausforderungen und Optimierungspotenziale im Zusammenhang mit der Software-Freigabe in hochregulierten Domänen. Vorgestellte Lösungsansätze sollen Unternehmen dabei unterstützen, effizientere und sicherere Freigabeprozesse zu implementieren. Darüber hinaus wird aufgezeigt, wie automatisierte Freigabemechanismen dazu beitragen können, die Compliance zu verbessern, die Time-to-Market zu verkürzen und die Produktivität in der Software-Entwicklung sowie die Qualität von Software-Lösungen in hochregulierten Branchen zu steigern.

Was sind Herausforderungen und zu bewältigende Aufgaben im Software-Freigabeprozess?

Software-Entwickler sehen sich derzeit gewaltigen Herausforderungen ausgesetzt. Im Spannungsfeld aus steigender Komplexität von Software-Lösungen und einer geforderten Effizienz im Entwicklungsprozess müssen sie einerseits die gestiegenen Anforderungen an Funktionalität, Sicherheit, Skalierbarkeit, Interoperabilität und Benutzerfreundlichkeit meistern. Andererseits gestalten die zunehmende Vernetzung von Systemen sowie die Integration neuer Technologien wie beispielsweise der künstlichen Intelligenz, dem Internet der Dinge und Cloud-Computing das Aufgabenfeld noch vielschichtiger. Die Gesamtheit der Systeme muss verstanden beziehungsweise transparent dargestellt werden. Zudem müssen Fehler identifiziert und behoben werden, um die Sicherheit und Zuverlässigkeit der Lösungen gewährleisten zu können.

Gleichzeitig sind die Ansprüche an Produktivität und Effizienz in der Software-Entwicklung gewachsen, um den steigenden Bedarf an innovativen Lösungen erfüllen und die Time-to-Market verkürzen zu können. Die Fähigkeit, komplexe Software-Lösungen schnell und effektiv zu entwickeln, zu testen und bereitzustellen, ist entscheidend für den Erfolg von Unternehmen im Zeitalter der digitalen Transformation.

Mit diesem Spagat beschäftigt sich ebenfalls eine Studie von McKinsey & Company. Sie verdeutlicht, wie sich die steigende Komplexität von Software-Lösungen und die stagnierende Software-Entwicklungsproduktivität auseinander bewegen (siehe Abbildung 1).

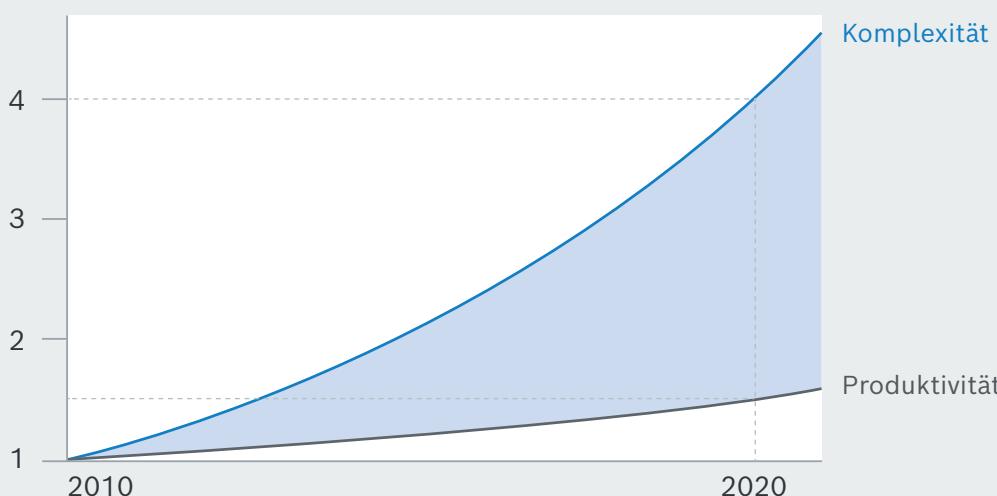


Abbildung 1: Spannungsfeld zwischen stagnierender Produktivität und steigender Komplexität bei der Software-Entwicklung¹

¹Quelle: In Anlehnung an / Modifiziert nach „When code is king: Mastering automotive software excellence“ | McKinsey
<https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/when-code-is-king-mastering-automotive-software-excellence>

2.1. Manuelle Software-Freigabe

Der Software-Freigabeprozess in hochregulierten Bereichen wie der Finanz- oder Automobilbranche **läuft noch nicht automatisiert ab**. Da Fehler oder Sicherheitslücken schwerwiegende Konsequenzen haben können, unterliegen in diesen Branchen Software-Produkte strengen regulatorischen Anforderungen und

Standards. Aus diesem Grund gestalten sich Freigabeprozesse für Software-Entwickler und -hersteller noch einmal deutlich komplexer. Der Software-Freigabeprozess umfasst in diesen Bereichen in der Regel mehrere Schritte (siehe auch Abbildung 2):

Software-Freigabe in 5 Schritten

1

Anforderungsanalyse:

Die Software-Anforderungen werden sorgfältig analysiert und dokumentiert, um sicherzustellen, dass sie den regulatorischen Anforderungen entsprechen.

2

Design und Entwicklung:

Die Software wird unter Berücksichtigung der regulatorischen Anforderungen entwickelt, wobei besonderes Augenmerk auf Sicherheit, Zuverlässigkeit und Compliance gelegt wird.

3

Validierung und Verifizierung:

Die Software wird gründlich getestet und ggf. angepasst, um sicherzustellen, dass sie den regulatorischen Anforderungen entspricht und korrekt funktioniert.

4

Dokumentation:

Alle relevanten Dokumente, einschließlich technischer Spezifikationen, Testberichten und Compliance-Zertifikaten, werden sorgfältig erstellt und aufbewahrt.

5

Freigabe:

Nach der erfolgreichen Validierung und Verifizierung laut Spezifikation wird die Software freigegeben.

Die für die Software-Freigabe erforderlichen Daten müssen häufig **aus verschiedenen Quellen abgerufen werden** – aus Software-Entwicklungstools (z.B. Azure DevOps), Issue-Tracker (z.B. JIRA) und Datenspeicher

(z.B. Sharepoint). Die daran anschließende Überprüfung der Anforderungserfüllung erfolgt in der Regel **manuell** und ist aufgrund der hohen Anforderungen an die Software sehr **zeitaufwendig**.



Abbildung 2: Software-Freigabeprozess mit manuellen Prüfungen

2.2. Zukünftige Herausforderungen bei der Software-Freigabe in hochregulierten Domänen

Vor diesem Hintergrund ergeben sich diverse Herausforderungen im Zusammenhang mit der Software-Freigabe,

welche die Wettbewerbsfähigkeit, Produktivität und Compliance von Unternehmen einschränken.

Herausforderungen	Auswirkungen
Wiederholte, zeitaufwendige Anforderungsprüfungen, die manuell durchgeführt werden (Risiko menschlicher Fehler)	<ul style="list-style-type: none">• Begrenzte Anzahl an Projekten• Lange Entwicklungszyklen (langsame Reaktionszeit auf die Reifegradbewertung von Software) → verlängerte Time-to-Market• Hohe Entwicklungskosten (Verschwendungen von Engineering-Ressourcen durch repetitive, manuelle, zeitintensive Aufgaben wie zum Beispiel Anforderungsprüfungen) → Einschränkung der Wettbewerbsfähigkeit
Häufig fehlende Kommunikation zwischen Entwicklern und Release Managern	Geringe Transparenz und Abstimmung über Anforderungen und Compliance-Kriterien
Mangelnde Klarheit bezüglich der Erfüllung von Anforderungen	Der Compliance-/Freigabestatus ist bis kurz vor der Freigabe/Prüfung ungewiss → Verzögerung der Software-Freigabe und keine Zeit, um Risiken zu mitigieren → Gefahr, eine fehlerhafte Software freizugeben und einen Rückruf zu riskieren
Häufig nicht vorhandene Transparenz über die Qualität der eigenen Software	Qualitätsrisiken sind bis kurz vor der Freigabe/Prüfung ungewiss
Anforderungen/Artefakte von verschiedenen Plattformen (DOORS, EA, BitBucket) und statischen Dateien (HTML) sind schwer zu verknüpfen	Fehlende Transparenz über die Artefakte und Anforderungen

Tabelle 1: Herausforderungen und ihre Auswirkungen bei der Software-Freigabe in hochregulierten Domänen

Wo liegen die Ursachen?

Wie bereits beschrieben, führt die erhöhte Komplexität speziell in hochregulierten Bereichen zu erhöhtem Anforderungsumfang und damit auch zu aufwendigeren Freigabe-Prozessen. In diesen Prozessen wird der Weg von der Entwicklung der Software bis hin zur finalen Freigabe als „Release Pipeline“ bezeichnet. Die häufige Unterbrechung der Release Pipeline führt dabei zu unnötigen Verzögerungen und einem erhöhten Aufwand in der Produktfreigabe. Brüche treten in diesem Zusammenhang in zwei Dimensionen auf: horizontal und vertikal.

3.1. Horizontale Durchgängigkeit: Unterbrechung der Release Pipeline

Eine horizontale Durchgängigkeit erlaubt es einem neu entwickelten Software-Feature, möglichst rasch und problemlos durch die verschiedenen Schritte der Freigabe zu gelangen. Dabei spielt beispielsweise die Automatisierung der einzelnen Schritte eine sehr große Rolle.

Da viele regulatorische Anforderungen in hochregulierten Bereichen manuell geprüft werden müssen, bevor die finale Freigabe des Software-Features erfolgen kann, entsteht jedoch oft ein horizontaler Bruch (siehe Abbildung 3).

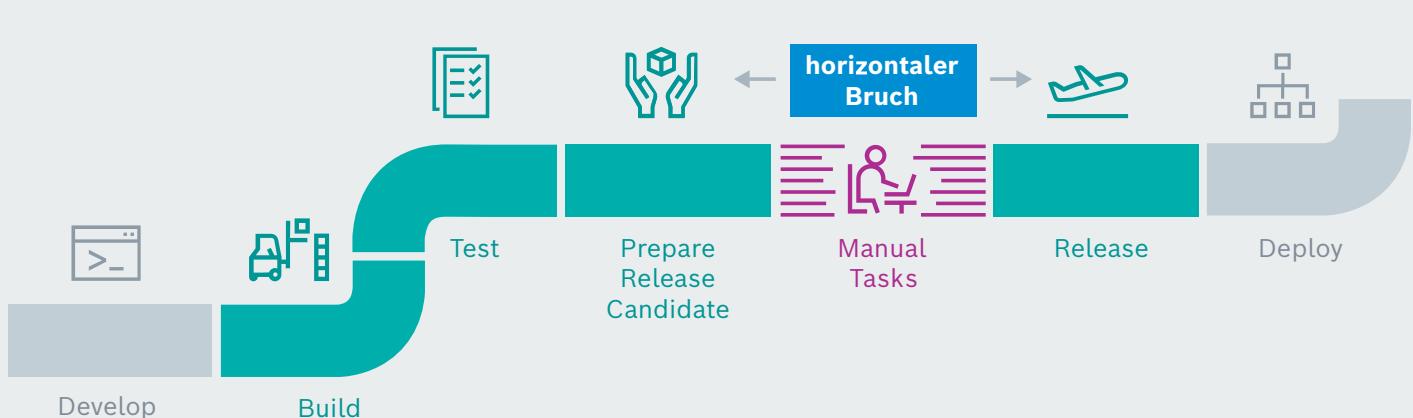


Abbildung 3: Darstellung des horizontalen Bruchs durch manuelle Prüfungen im Software-Freigabeprozess

Auf der Suche nach Gründen für den Bruch lassen sich eine Reihe von Unterschieden zwischen den allgemeinen, automatisierten Pipeline-Schritten (Türkis) und den speziellen, für hochregulierte Bereiche wichtigen, Prüfungsschritten (Purpur) finden:

	Automatisierte Schritte der Release Pipeline	Manuelle Schritte der Release Pipeline
Art der Aktivität	Hochautomatisierte Pipelines	Manuelle Prüfung von Checklisten und Prozessanforderungen
Art der Artefakte/ Dokumente	Maschinenlesbare Dokumente, Textformate, leicht zu verarbeiten	Office-Dokumente, PDFs, Scans, nur schwer automatisiert zu verarbeiten
Fokus der Tests	Technische Prüfungen, z.B. Integrierbarkeit, Vollständigkeit, Testabdeckung	Compliance, Safety, Security, (regulatorische Policies, sowie Unternehmenspolicies)
Involvierte Personen/Gruppen	SW-Entwickler, DevOps-Entwickler, SW-Tester, mit starkem Verständnis für technische Feinheiten, aber ohne Detailwissen zu Compliance Politik und Freigabeprozessen	Release Manager & Quality Manager, mit starkem Prozess- und Compliancewissen, in der Regel aber nur oberflächlichem technischen Verständnis für die Freigabe-Prozesse
Speicherorte für Artefakte/ Dokumente	Issue Tracker, Source-Code-Management (SCM) und Continuous Integration/Continuous Deployment (CI/CD) Systeme. In der Regel gute Interoperabilität mit APIs etc.	Office-Plattformen, SharePoint, Netzlaufwerke. In der Regel sehr eingeschränkte Interoperabilität über APIs, etc.
Zugriffs- und Bearbeitungsrechte für Artefakte, Berichte, Dokumente	Entwickler haben Zugriff auf alle SCM & CI/CD Systeme, bevorzugen Textformate, für die sie Unterstützung durch integrierte Entwicklungsumgebungen (IDE) erhalten.	Häufig kein direkter Zugriff auf Build-Artefakte in SCM oder CI/CD Systemen, Office-Formate und -Plattformen werden bevorzugt. Schwierigkeiten im Umgang mit Textformaten.
Art der Kommunikation	Kommunikation innerhalb der Entwicklungsplattform, z.B. Issues/Pull Requests in GitHub, GitLab etc.	Kommunikation per E-Mail, Telefon oder in Meetings

Tabelle 2: Unterschiede zwischen allgemeinen, automatisierten Pipeline- und speziellen, hochregulierten Prüfungsschritten

Zentrale Herausforderungen

Bei der Gegenüberstellung der relevanten Personengruppen Software-Entwickler und Release Manager finden sich Differenzen, aus denen viele der Unterschiede resultieren. So sind zentrale Herausforderungen, die

zu einem Bruch in der horizontalen Durchgängigkeit führen, uneinheitliche Systeme, Formate, Ablagen sowie eine uneinheitliche Sprache beziehungsweise Kommunikation.

3.2. Vertikale Durchgängigkeit: Bruch zwischen kleinen/ häufigen Freigaben und großen Freigaben

Die fehlende horizontale Durchgängigkeit kann zusätzlich einen Bruch in der vertikalen Durchgängigkeit zur Folge haben.

Kleine, aber häufige Inkremente bestimmen heutzutage die Software-Entwicklung. Oft sind sie an das Ende eines Sprint-Zyklus gekoppelt, in dem Code-Änderungen, wie beispielsweise neue Features oder Bugfixes, mithilfe von Pull Requests (manuelle Code-Reviews, unterstützt durch automatisierte Prüfungen) geprüft und dann integriert werden. Um die Software-Entwicklung nicht auszubremsen, wird dort der Fokus häufig auf die leicht automatisierbaren Prüfungen sowie technisch sinnvollen bzw. notwendigen Prüfungen gelegt.

Hierfür existieren mächtige und umfangreiche Automatisierungen auf Basis komplexer Continuous Integration Tools, die dem Entwickler zeitnah und detailliert Rückmeldung zur Qualität des neuen Software-Features geben.

Parallel zu diesen kleinen Iterationen läuft die langwierige Vorbereitung der eigentlichen Produktfreigabe. Diese beinhaltet aufwendige und oft manuelle Prüfungen der Compliance und anderer regulatorischer Anforderungen und dauert in der Regel deutlich länger als der kleine Zyklus eines Pull Requests oder einer Entwicklungsfreigabe (siehe Abbildung 4).

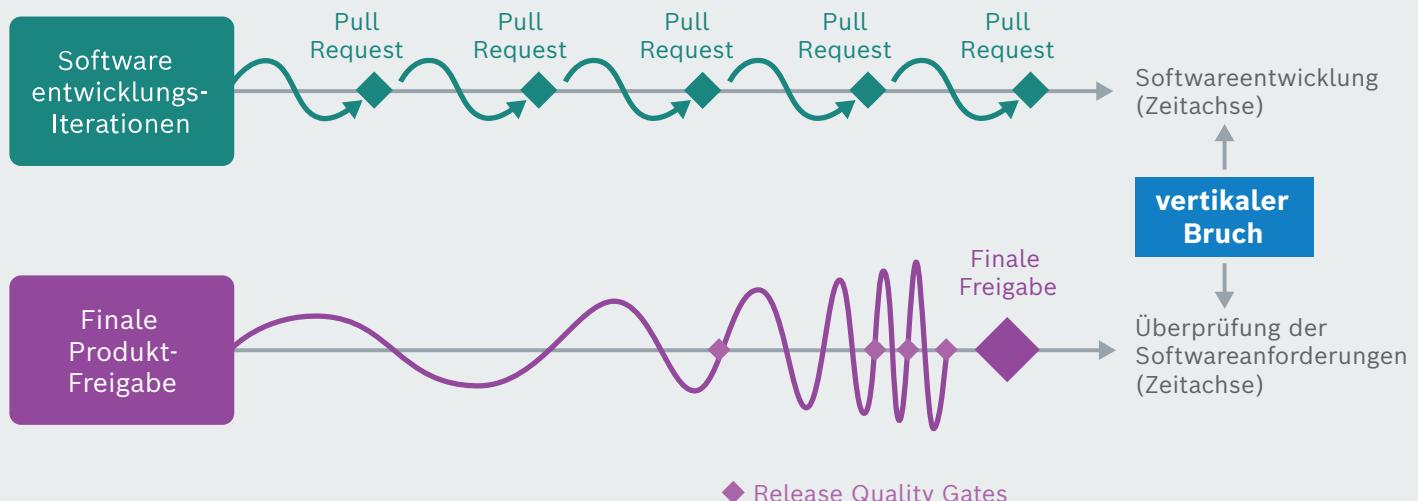


Abbildung 4: Gegenüberstellung der Prozesse bei der Software-Entwicklung und der Produktfreigabe

Mit jedem Schritt hin zur finalen Produktfreigabe sind dabei mehr regulatorische Anforderungen zu erfüllen und Prüfungen durchzuführen, was gleichzeitig die Anforderungen an Dokumentation und Rückverfolgbarkeit erhöht. Die Prüfungen sind teilweise nur manuell durchführbar oder basieren auf Verarbeitung von Dokumentenformaten, die sich nicht oder nur sehr aufwendig automatisieren lassen. Daraus resultieren lange Vorlaufzeiten für Freigaben.

Die Diskrepanz in der Anzahl und Art der ausgeführten Prüfungen führt zum Bruch der vertikalen Durchgängigkeit.

Der Fokus bei den entwicklungsbegleitenden Prüfungen (z.B. im Rahmen von Pull Requests) liegt auf einfacher Automatisierung und Geschwindigkeit. So wird die bestmögliche Nutzbarkeit der Ergebnisse für die finale Software-Freigabe nicht berücksichtigt. Bereits durchgeführte Prüfungen auf Pull-Request-Ebene müssen deshalb oftmals erneut neu durchgeführt werden. Zudem ist es nötig, Dokumente und Artefakte in die korrekten Formate für die finale Freigabe zu konvertieren beziehungsweise an einem geeigneten Ort abzulegen, der für alle relevanten Personen zugänglich ist. Dies kann beispielsweise dazu führen, dass Dokumente, die bereits automatisiert erstellt wurden, nochmals händisch in ein anderes Format konvertiert werden müssen. Nur so können sie hinterher einen Dokumenten-Freigabeprozess in einem speziellen Workflow-System durchlaufen.

Ein weiteres Beispiel sind Signaturen in PDF-Dokumenten. So werden beispielsweise händisch Webseiten mit Ergebnissen aus CI/CD Systemen als PDF exportiert. Und das lediglich mit dem Ziel, sie im PDF-Format signieren zu können. Damit sie von allen Beteiligten

geöffnet und bearbeitet werden können, müssen einige freigaberelevante Informationen wie beispielsweise Open Source Scans in verschiedenen Formaten vorliegen und oft händisch aufbereitet werden. Das kann im Beispiel der Open Source Scans dazu führen, dass Probleme aufgrund der Nutzung von Open Source Software (OSS) erst spät im Rahmen der Prüfungen für die finale Freigabe entdeckt werden. Wären die OSS-Prüfungen in den Pull Requests und den Prüfungen für die finale Freigabe identisch, würden Probleme frühzeitig entdeckt und Verzögerungen der Produktfreigabe vermieden werden.

Für den vertikalen Bruch ergeben sich somit folgende Probleme:

- Prüfungen werden auf oberen Ebenen implementiert ohne eine Kontrolle, ob diese Prüfungen und deren Ergebnisse auch für die finalen Freigaben relevant, hilfreich beziehungsweise geeignet sind. Dies führt zu ähnlichen Prüfungen auf verschiedenen Ebenen und damit zu einer vermeidbaren Dopplung der Implementierungen.
- Andererseits sollten bereits auf unteren Ebenen kritische Situationen sichtbar werden, die später auf höheren Ebenen zu Verzögerungen führen können.

Werden also wichtige oder kritische Fragen erst auf der oberen Ebene gestellt und nicht regelmäßig im allgemeinen Software-Entwicklungsprozess geprüft, kann dies unnötige Verzögerungen nach sich ziehen.

Zentrale Herausforderungen

Die fehlende Durchgängigkeit der Qualitätsprüfungen zwischen den verschiedenen Freigabe-Ebenen behindert den Entwicklungsprozess. Gleichzeitig erfolgt

die Prüfung von wichtigen und kritischen Compliance-Aspekten zu spät, da diese oft nur auf höheren Ebenen geprüft werden.

4

Wie lassen sich die Herausforderungen meistern?

Mit folgenden Lösungsansätzen kann den häufigen Unterbrechungen der Release Pipeline und den damit verbundenen unnötigen Verzögerungen und erhöhten Aufwänden in der Produktfreigabe begegnet werden.

4.1. Lösungsansätze für eine bessere horizontale Durchgängigkeit

Eine gemeinsame Plattform ermöglicht es den Nutzern, die aufgezeigten Herausforderungen der horizontalen Durchgängigkeit zielgerichtet anzugehen. So wären folgende zentrale Punkte gewährleistet:

Einheitliches System

Während Entwickler klassischerweise CI/CD Tools und Entwicklungsplattformen nutzen, arbeiten Manager in der Regel mit Office-Plattformen. Die gemeinsame Nutzung einer Plattform, die sowohl technisch anspruchsvolle Realisierungen automatisierter Prüfungen erlaubt, als auch eine einfache, konsistente und aktuelle Darstellung und Auswertung der Prüfungen bietet, ist deshalb notwendig.

Einheitliche Formate

Auch wenn Textformate für Software-Entwickler einen großen Charme besitzen, benötigen Manager in der Regel eher visuell aussagekräftige Formate. Eine Plattform, die Prüfungen, Reports etc. sowohl in maschinlesbaren Formaten als auch grafisch hochwertig aufbereitet darstellen kann, bietet die Möglichkeit einer Single-Source-of-Truth. Diese kommt ohne zusätzliche Format-Konvertierungen aus.

Einheitliche Datenablagen

Software-Entwicklern und Managern gleichzeitig einen komfortablen Zugriff auf die entwicklungsrelevanten Dokumente und Berichte zu ermöglichen, ist nur schwer zu realisieren.

Office-Plattformen bieten oft nicht die notwendigen Mittel für Automatisierungen, Versionierungen und Anbindungen an andere externe Systeme etc. Eine gemeinsame unabhängige Plattform und Ablage kann die notwendigen Mittel für umfangreiche Automatisierungen bieten. Gleichzeitig schafft sie einen einfachen Zugriff für alle Personengruppen.

Einheitliche Kommunikation

Die Sprache innerhalb von Issue-Trackern, Source-Code-Management (SCM) oder CI/CD Systemen ist immer sehr technisch geprägt. Compliance-Dokumente auf der anderen Seite beinhalten oft Formulierungen, die sich aus Gesetzestexten und Normen herleiten. Dieser Unterschied erschwert eine gemeinsame Nutzung der Dokumente für beide Personengruppen. Auch die Kommunikation in den technischen Systemen gestaltet sich anders als Manager aus E-Mail-Austausch, Telefon- und Videokonferenzen gewohnt sind.

So sind Informationen in der Regel dezentral abgelegt. Zum Beispiel erfolgt die Abstimmung von Produkt-Features in Issue-Trackern, während ein anderer Teil der Diskussion parallel dazu im zugehörigen Pull Request im Source-Code-Management-System abläuft. Eine

gemeinsame Plattform kann helfen, hier eine einheitliche Sprache zu finden. Zusätzlich kann sie den Herausforderungen angepasste Kommunikationsmittel bereitstellen, die von allen Beteiligten ohne Hindernisse genutzt werden können.

4.2. Lösungsansätze für eine bessere vertikale Durchgängigkeit

Auch die zentralen Herausforderungen der vertikalen Durchgängigkeit können mittels zielgerichteter Maßnahmen gemeistert werden. So lassen sich folgende zentrale Punkte gewährleisten:

Einheitliche Qualitätsprüfungen

Eine gemeinsame Plattform, auf der sowohl die Prüfungen der unteren Ebenen als auch die Qualitätsprüfung der finalen Freigabe implementiert und deren Ergebnisse sichtbar sind, hilft bei der Wiederverwendbarkeit von Prüfungen und deren Ergebnissen. Eine gemeinsame Sicht auf den Status der Freigabevorbereitung führt zu einer besseren Abstimmung zwischen Entwicklern sowie Release und Quality Managern.

Frühere Prüfung wichtiger Compliance Policies

Auch wenn viele Compliance-Dokumente erst zur finalen Freigabe notwendig sind, sollte deren Vorbereitung schon in früheren Quality Gates erfolgen. Idealerweise verwenden sämtliche Freigabestellen somit dieselben Tools zum Sammeln und Aufbereiten der Daten.

Hierfür ist es essenziell, dass beide Personengruppen transparenten Zugang zu den Freigabeprüfungen der jeweils anderen Gruppe erhalten: Software-Entwickler sollten beispielsweise einsehen können, welche Prüfungen für die finale Freigabe erforderlich sind und wie sie implementiert sind. So können Synergien besser genutzt und das gegenseitige Bewusstsein über notwendige und wichtige Prüfungen gesteigert werden. Entwickler haben so die Möglichkeit, den Stand der Produktfreigabe einzusehen und auch einen Einblick in die dortigen Anforderungen zu bekommen. Im Idealfall dringen diese dann zukünftig erst gar nicht bis in die Produktfreigabeprüfungen vor.

Darüber hinaus wäre es sinnvoll, wenn sich die verschiedenen Ebenen für die Freigabe-Automatisierung idealerweise auch Fragenkataloge und zugehörige Automatisierungslösungen teilen.

5

Automatisierung mit der Software-Lösung Yaku

Vor dem Hintergrund dieser Ansätze erscheint eine Software-Lösung mit gemeinsamer Plattform, die sowohl die horizontale wie auch die vertikale Durchgängigkeit sicherstellt, angebracht. Die Software-Lösung Yaku gewährleistet die angesprochenen zentralen Punkte. Sie soll deshalb im Folgenden vorgestellt werden.

5.1. Einführung: Was ist Yaku?

Yaku ist eine Software-Lösung zur automatisierten Generierung von Freigabeberichten in hochregulierten Bereichen. Um die Möglichkeiten der Plattform übersichtlich aufzeigen zu können, soll zunächst der mög-

liche zeitliche Ablauf einer Software-Freigabe mit Yaku einem Ablauf ohne die Nutzung von Yaku gegenübergestellt werden (siehe Tabelle 3).

Ohne Yaku	Mit Yaku
Manuelles Sammeln der benötigten Informationen aus den Freigabe-Dokumenten.	1 Automatisierung durch Data-Fechter-Autopiloten und Adapter zu Software-Entwicklungs-tools, Issue-Trackern und Datenspeichern, die die benötigten Artefakte laden.
Manuelle Auswertung von Freigabe-Dokumenten zur Überprüfung der Anforderungen für die Software-Freigabe.	2 Automatisierung durch Evaluator-Autopiloten, die die Erfüllung der Anforderungen überprüfen.
Manuelle Erstellung von Freigabeberichten.	3 Automatisierte Erstellung von Berichten, die den Status der Freigabe für alle Anforderungen transparent darstellen.

Tabelle 3: Zeitlicher Ablauf einer Software-Freigabe ohne und mit Yaku

Abbildung 5 zeigt, wie Yaku den Software-Freigabeprozess automatisiert.

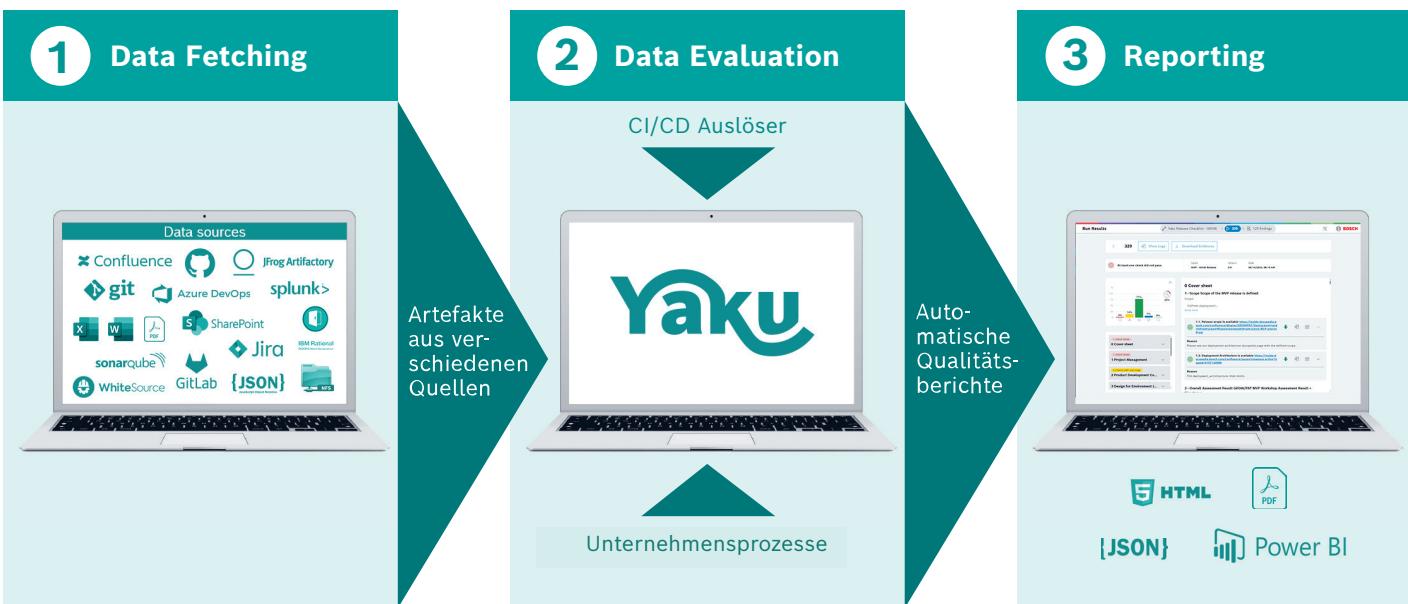


Abbildung 5: Vorgehensweise zur automatisierten Erstellung von Freigabedokumenten mithilfe der Software-Lösung Yaku

Grundsätzlich können folgende Problemstellungen der manuellen Software-Freigabe durch den Einsatz von Yaku gelöst werden (siehe Tabelle 4):

Ohne Yaku	Mit Yaku
Manuelle Dokumentation der Freigabe.	Download der Beweise und Prüfungen, die für die Freigabe durchgeführt wurden.
Es besteht keine bidirektionale Rückverfolgbarkeit zwischen Anforderungen und Tests; sie wird manuell ausgelöst.	Durch die Verwendung von Knowledge-Graphen (MTrace) kann eine automatisierte bidirektionale Rückverfolgbarkeit zwischen Anforderungen und Tests erreicht werden.
Fehlende Integration von Legacy-Systemen wie Doors Classic, CodeBeamer und GitHub mit modernen Software-Entwicklungstools führt zu Inkompatibilitäten und ineffizienten Arbeitsabläufen.	Knowledge-Graph fungiert als Vermittler zwischen den verschiedenen Systemen und ermöglicht es, die Beziehungen und Abhängigkeiten der Daten und Artefakte in den Legacy-Systemen und den modernen Entwicklungstools zu modellieren und zu verknüpfen.

Informationen für die Software-Freigabe sind in verschiedenen Tools gespeichert und sind nicht miteinander verknüpft (keine Transparenz).

Kommunikation während der Software-Freigabe per E-Mail und in Meetings.

Der Compliance-/Freigabestatus ist bis kurz vor der Freigabe/Prüfung ungewiss.

Qualitätsrisiken sind bis kurz vor der Freigabe/Prüfung ungewiss.

Zeitaufwand für eine Software-Freigabe: je nach Komplexität der Software sowie den Kriterien für die Freigabe **mehrere Tage bis mehrere Wochen**.

Informationen für die Software-Freigabe werden durch Yaku als gemeinsame Plattform für alle an der Freigabe beteiligten Personen konsolidiert und im Freigabebericht gespeichert.

Kommunikation innerhalb der Plattform möglich.

Freigabestatus kann kontinuierlich überprüft und die Prüfungen können vorgezogen werden. Maßnahmen können frühzeitig eingeleitet werden.

Fehlende Anforderungen für die Software-Freigabe werden frühzeitig identifiziert, indem Yaku transparent die Anforderungserfüllung darstellt.

Zeitaufwand für eine Software-Freigabe: Abruf des Freigabestatus **innerhalb weniger Minuten**.

Tabelle 4: Problemstellungen der manuellen Software-Freigabe und Lösungen mithilfe von Yaku

Yaku kann in die bestehende Prozess-Methoden-Tool-Landschaft integriert werden und ist somit für Kunden geeignet, die ihre PMT nicht neu aufsetzen können.

So fördert Yaku die Durchgängigkeit und sorgt für schnellere Reaktionszeiten bei der Behebung von kritischen Mängeln. Zusätzlich entstehen Synergie-Effekte bei der gemeinsamen Nutzung von Automatisierungen auf allen Ebenen und für alle Personengruppen.

Durch die Implementierung von Yaku haben

- Entwickler die Möglichkeit, ihre Automatisierungen und Skripte für die Prüfung von freigaberelevanten Kriterien und Anforderungen zu verwalten.
- Release und Quality Manager die Möglichkeit, den Status der Freigabe-Vorbereitungen einzusehen und mitzuwirken beziehungsweise mitzudiskutieren.

5.2. Use-Case:

Realisierung einer Release Pipeline mit Yaku

Abbildung 6 zeigt die Anwendung von Yaku im Rahmen eines Software-Entwicklungs-Workflows, der für einen Kunden im Finanzbereich umgesetzt wurde. Dieser Use Case dient als Blaupause und ist gleichermaßen auf Kunden in der Automobilindustrie übertragbar.

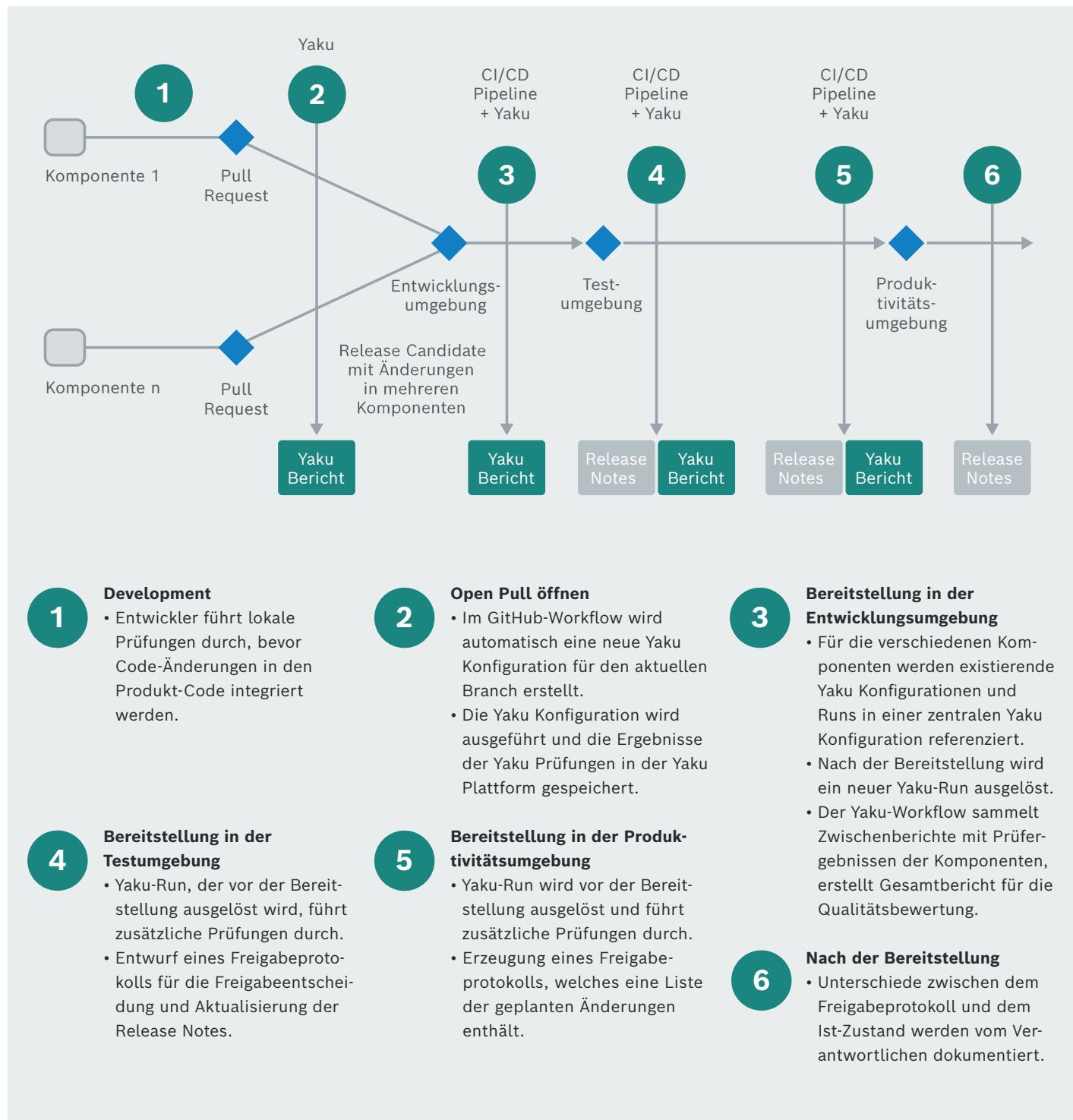


Abbildung 6: Software-Entwicklungs-Workflow mit Yaku

Ausgangspunkt des Workflows sind die Pull Requests der einzelnen Komponenten. Für jede Komponente erfolgt eine Prüfung mit einem eigenen Yaku Workflow, sobald ein Pull Request erstellt wird. Der Fokus bei diesen frühen Prüfungen liegt auf dem Quellcode, so dass die wichtigsten Code-Qualitäts-Metriken erfasst werden. Die Ergebnisse stehen direkt als Yaku Berichte in der Yaku Plattform für alle Personen zur Verfügung.

Nach der Zusammenführung und Integration der verschiedenen Komponenten folgen verschiedene Freigabestufen. Für jede dieser Stufen werden verschiedene Freigabekriterien geprüft und zentral in der Yaku Plattform abgelegt. So ist der Status der Freigabevorbereitung für alle ersichtlich.

Je weiter der Workflow fortgeschritten ist, umso weniger technische Prüfungen sind notwendig. Stattdessen werden produktbezogene, regulatorische Anforderungen (Policies) überprüft, wie beispielsweise Dokumentationspflichten oder Rückverfolgbarkeits-Anforderungen (Release Notes, Release Record etc.).

Dieses Setup stellt die Lösung für die erläuterten zentralen Herausforderungen dar:

1. Die horizontale Durchgängigkeit wird durch einen hohen Automatisierungsgrad der einzelnen Freigabestufen gewährleistet. Sowohl technische Prüfungen als auch Prüfungen der Freigabedokumente sind sichergestellt. So kann die gesamte Release Pipeline deutlich schneller durchlaufen werden.
2. Die vertikale Durchgängigkeit wird durch eine Integration der schnellen, iterativen, entwicklungsbegleitenden Prüfungen auf Pull-Request-Ebene in die Yaku Plattform erreicht: Probleme im Freigabeprozess werden frühzeitig evident und in Yaku integrierte Prüfungen können auch direkt für die finalen Freigabeschritte genutzt werden. Alle Ergebnisse der gesamten Release Pipeline sind für alle sicht- und nutzbar.

6

Fazit

Durch die Automatisierung des Software-Freigabeprozesses mithilfe von Tools wie Yaku können folgende Mehrwerte erzielt werden:

1. **Sicherstellung der vertikalen Durchgängigkeit:** Automatisierung der technischen Prüfungen und Überprüfung der Freigabedokumente.
2. **Effizienzsteigerung:** Unternehmen können den Freigabeprozess beschleunigen und vereinfachen.
3. **Fehlerreduzierung:** Menschliche Fehler werden minimiert.

4. **Konsistenz:** Eine konsistente Anwendung von Freigabeprozessen führt automatisch zu einer höheren Qualität und Zuverlässigkeit der Software.
5. **Compliance:** Die Einhaltung strenger regulatorischer Anforderungen für Unternehmen im Automobil- oder Finanzbereich kann gewährleistet werden.
6. **Kostensparnis:** Unternehmen können Kosten für manuelle Freigabeprozesse reduzieren, da weniger Arbeitszeit und Ressourcen benötigt werden.

7. Skalierbarkeit: Unternehmen können ihre Freigabe-prozesse bei Bedarf skalieren, um mit dem Wachstum des Unternehmens Schritt zu halten.

8. Risikominderung: Das Risiko von Sicherheitslücken und Fehlfunktionen in der Software kann durch die Standardisierung und Überwachung von Freigabe-prozessen minimiert werden.

9. Kürzere Time-to-Market: Die Gesamtzeit für die Freigabe von Software wird verkürzt, wodurch die Software schneller vermarktet werden kann.

Software-Lösungen wie Yaku tragen dazu bei, die Transparenz hinsichtlich der Software-Qualität und der Erfüllung von Anforderungen zu erhöhen, indem sie alle relevanten Daten und Fakten zur Software-Freigabe in einem einzigen Medium bereitstellen. Dies gewährleistet die horizontale Durchgängigkeit. Zudem können Software-Entwicklungsressourcen effizient genutzt werden, da sie nicht an manuelle Freigabeprozesse gebunden sind. Dies ermöglicht eine signifikante Steigerung der Produktivität und hilft, mit der Herausforderung der steigenden Komplexität umzugehen (siehe Abbildung 7).

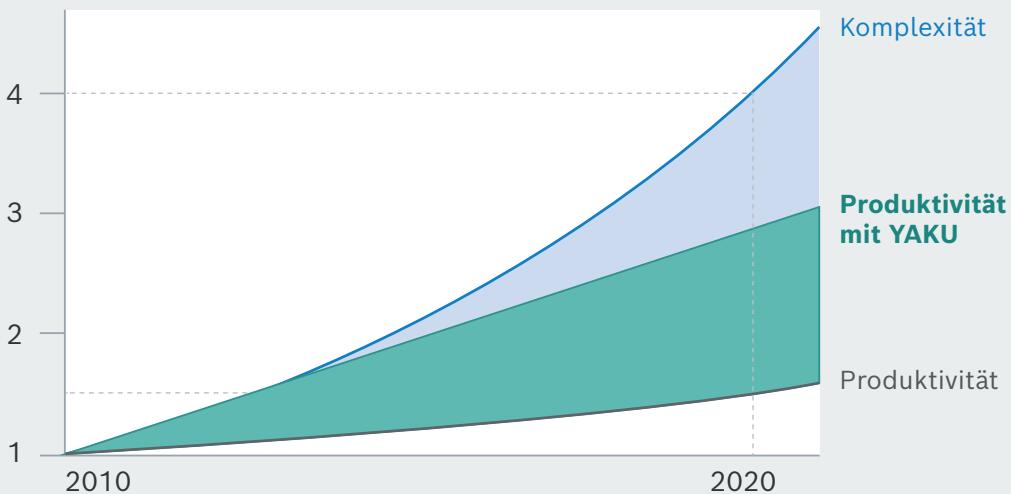


Abbildung 7: Vergleich der Produktivität in der Software-Entwicklung mit und ohne Yaku

Die Automatisierung von Software-Freigaben in hochregulierten Bereichen ist eine hochkomplexe Aufgabe, die sorgfältige Planung und Implementierung erfordert. Es ist notwendig, je nach Einzelfall zu beurteilen, ob und in welchem Maße eine Automatisierung angemessen ist. Einige Aspekte des Software-Freigabeprozesses sind nach wie vor nicht vollständig automatisierbar – denn nur manuelle Tests können wirklich sicherstellen, ob die Software fehlerfrei funktioniert. Ebenso erfordert die Genehmigung der Freigabe durch das Management Input von außen. Darüber hinaus können auch

spezielle Anpassungen oder Konfigurationen erforderlich sein, die nicht automatisiert werden können.

Die vorgestellten Lösungsansätze sollen Unternehmen dabei unterstützen, die gewachsenen Herausforderungen der Software-Entwicklung zu bewältigen und die Vorteile der Automatisierung zu nutzen. Durch die Implementierung von Software-Lösungen wie Yaku sind Unternehmen in der Lage, die Effizienz zu steigern, die Sicherheit zu erhöhen und die Einhaltung gesetzlicher Vorschriften zu gewährleisten.

Kontaktieren Sie uns

Unsere Software-Lösung ermöglicht es Ihnen, den Software-Freigabeprozess zu automatisieren. So können Sie Zeit und Ressourcen sparen und gleichzeitig die Qualität und Sicherheit Ihrer Software verbessern.

Wir arbeiten eng mit Ihnen zusammen, um sicherzustellen, dass unsere Lösung perfekt auf Ihre spezifischen Anforderungen zugeschnitten ist und Ihnen dabei hilft, Ihre Ziele zu erreichen.

Effizient, zuverlässig, automatisiert – für eine reibungslose Software-Freigabe!

Über die Autoren



Isabelle Jahnel

Business Developerin
Bosch Software Flow
✉ isabelle.jahnel@de.bosch.com
📞 +49(173) 6774016

Wolfgang Ulmer

Scrum Master & Software Developer
Bosch Software Flow
✉ wolfgang.ulmer@de.bosch.com
📞 +49(173) 3692359

Kontaktieren Sie das Bosch Software Flow Team für weitere Informationen per Mail

✉ BoschSoftwareFlow.Info@de.bosch.com

Oder besuchen Sie unsere Webseite

🌐 <https://bosch-software-flow.com/>