

buerokratie-automatisieren.de – ein Informationsportal der J.v.G. technology GmbH

Mission: Bürokratische Prozesse durch intelligente Automatisierung auflösen.

Mein Maintenance-Budget: Das Arbeitsblatt für die wahren Kosten Ihres Open-Source-Automatisierungs-Stacks

Für wen ist dieses Arbeitsblatt – und warum es Ihre wichtigste Planungsgrundlage wird

Dieses Arbeitsblatt richtet sich an IT-Leiter, Heads of Automation und Lead Developer, die den Wechsel von teuren kommerziellen RPA-Lizenzen zu einem souveränen Open-Source-Stack mit Camunda (BPMN-basierte Prozessorchestrierung – eine Engine, die definiert, welcher Schritt wann und unter welchen Bedingungen ausgeführt wird) und Robocorp (code-basierte RPA-Ausführungsschicht – Software-Roboter, die konkrete Aufgaben wie Dateneingaben oder Abfragen automatisch erledigen) planen.

Es löst ein konkretes Problem: Die abstrakte Angst vor „Wartungschaos“ wird in einen planbaren, verteidigungsfähigen Budgetposten übersetzt. Gehen Sie die vier Bereiche Schritt für Schritt durch, schätzen Sie Ihren monatlichen Zeitaufwand pro Aufgabe und rechnen Sie am Ende alles in eine FTE-Planung (Full-Time Equivalent – die Anzahl an Vollzeitstellen, die Sie für den Betrieb brauchen) um.

Das Ergebnis: Eine realistische Kalkulation, die Sie Ihrem Management vorlegen können – und die den Kern jeder Automatisierungsentscheidung trifft: die Fähigkeit, bürokratische Prozesse kontrolliert, sicher und wirtschaftlich zu automatisieren, ohne von einem einzelnen Anbieter abhängig zu sein.

So nutzen Sie dieses Arbeitsblatt: Drucken Sie es aus oder öffnen Sie es digital. Füllen Sie die Felder ehrlich und eher konservativ aus. Planen Sie ca. 30–45 Minuten ein.

Quick Check: Die 3 größten Kostenfallen bei Open-Source-Automatisierung

Bevor Sie mit der Detailplanung beginnen, prüfen Sie diese drei häufig unterschätzten Risiken. Haken Sie ab, was Sie bereits adressiert haben. Jedes leere Kästchen ist ein Warnsignal.

- ☐ **Falle 1: Nur Implementierungsaufwand geplant.** Der größte Aufwand entsteht nicht beim Einrichten, sondern im laufenden Betrieb – Überwachung, Patching, Anpassung. Viele Teams unterschätzen diesen Faktor erheblich, typischerweise um den Faktor 2–4 gegenüber der reinen Ersteinrichtung (Richtwert aus Praxisberichten in DevOps-Communities; der genaue Faktor hängt stark von Ihrer Stack-Komplexität ab).
- ☐ **Falle 2: Kein dedizierter Verantwortlicher.** Open-Source-Pflege „nebenbei“ funktioniert nicht. Es braucht mindestens eine klar benannte Person mit definiertem Zeitkontingent. Ohne diese Zuständigkeit bleiben Updates liegen, Wissen geht verloren und Probleme eskalieren schneller.
- ☐ **Falle 3: Sicherheits-Updates als optional betrachtet.** Das regelmäßige Einspielen von Security-Patches (Sicherheitsaktualisierungen, die bekannte Schwachstellen schließen) ist nicht verhandelbar. Jede Open-Source-Komponente veröffentlicht Security Advisories (öffentliche Warnmeldungen über Sicherheitslücken). Diese müssen aktiv überwacht und zeitnah umgesetzt werden.

Auswertung: Wenn Sie weniger als 2 von 3 Punkten abhaken können, behandeln Sie dieses Arbeitsblatt mit besonderer Sorgfalt – Ihr Stack hat ein erhöhtes Betriebsrisiko.

Teil 1: Updates und Security Patches – der proaktive Kern

Hier planen Sie den wiederkehrenden Aufwand, um Ihren Stack sicher und aktuell zu halten. Die voreingetragenen Stundenwerte in der ersten Zeile dienen als Orientierungswert für ein typisches mittelgroßes Setup. Ihre tatsächlichen Werte hängen ab von: Anzahl der Umgebungen (Staging, Produktion), Komplexität Ihrer Prozesse, Regulierungsanforderungen und Teamgröße.

Wichtiger Hinweis: Die Beispielwerte unten sind vereinfachte Richtwerte für einen Stack mit einer Staging- und einer Produktionsumgebung. Sie berücksichtigen keine Rollback-Szenarien (Rücksetzen auf eine vorherige Version bei Fehlern), keine Kompatibilitätstests zwischen Komponenten und keine regulatorischen Dokumentationspflichten. In der Praxis können diese Faktoren den Aufwand um geschätzt 20–50 % erhöhen.

Komponente	Überwachung (Advisories lesen, bewerten)	Testaufwand (Staging- Umgebung)	Deployment (Produktivschaltung)	Ihre Schätzung: Stunden / Monat
Camunda Engine (Orientierung)	~1 h	~4 h	~2 h	_____ h
Robocorp Agent/Worker (Orientierung)	~1 h	~2 h	~1 h	_____ h
Datenbank (z. B. PostgreSQL)	~1 h	~2 h	~1 h	_____ h
Message Queue (z. B. RabbitMQ)	~1 h	~2 h	~1 h	_____ h
Weitere Komponente: _____	_____ h	_____ h	_____ h	_____ h
Weitere Komponente: _____	_____ h	_____ h	_____ h	_____ h

Komponente	Überwachung (Advisories lesen, bewerten)	Testaufwand (Staging- Umgebung)	Deployment (Produktivschaltung)	Ihre Schätzung: Stunden / Monat
------------	--	---------------------------------------	------------------------------------	--

Summe Teil 1: _____ h

Was dieses Ergebnis bedeutet: Sie haben den wiederkehrenden Grundaufwand für die technische Instandhaltung Ihres Stacks quantifiziert. Dieser Aufwand fällt jeden Monat an – auch wenn gerade alles „ruhig“ läuft.

Teil 2: Community und Wissensmanagement – Ihre Wissensversicherung

Aktives Engagement in Open-Source-Communities (Foren, Slack-Channels, GitHub-Repositories) ist keine Kür. Es ist Ihre günstigste Versicherung gegen unbekannte Probleme. Hier sichern Sie sich schnellen Zugang zu Lösungen, erfahren frühzeitig von Breaking Changes (Änderungen in einer neuen Version, die bestehende Funktionen beeinträchtigen können) und bauen internes Know-how auf.

Aufgabe	Was genau zu tun ist	Ihre Schätzung: Stunden / Monat
Aktive Überwachung	Regelmäßiges Lesen von: Camunda-Forum, Robot Framework-Forum, relevante GitHub-Repositories, Slack-Channels, Fach-Blogs. Ziel: Frühwarnung bei Problemen und neuen Versionen.	_____ h
Interne Dokumentation	Gefundene Lösungen, Workarounds und Entscheidungen für das eigene Team aufbereiten (Wiki, Confluence, oder ein einfaches Markdown-Repository).	_____ h
Troubleshooting-Recherche	Gezielte Suche in der Community bei konkreten Problemen. Erfahrungsgemäß intensiver in den ersten 6–12 Monaten nach Einführung.	_____ h
Wissenstransfer im Team	Kurze interne Sessions oder Code-Reviews, damit Know-how nicht bei einer Person hängen bleibt.	_____ h
Summe Teil 2:		_____ h

Was dieses Ergebnis bedeutet: Sie haben den Zeitaufwand für den Aufbau und Erhalt von überlebenswichtigem Know-how eingeplant. Diesen Posten zu streichen, spart kurzfristig Stunden – und kostet langfristig Tage.

Teil 3: Interner Support und Incident Management – Ihr reaktiver Puffer

Hier planen Sie den Aufwand, der bei Problemen im laufenden Betrieb entsteht. Dieser Teil ist naturgemäß schwer vorherzusagen. Planen Sie deshalb einen expliziten Puffer ein. Ein typischer Richtwert: Addieren Sie 20–30 % auf Ihre Schätzung, um Monate mit gehäuften Incidents (Störungen, die den normalen Betrieb beeinträchtigen) aufzufangen.

Aufgabe	Was genau zu tun ist	Ihre Schätzung: Stunden / Monat
Anwendersupport	Unterstützung der Fachbereiche bei Fragen zu automatisierten Prozessen. Umfasst auch die Kommunikation, warum ein Prozess sich anders verhält als erwartet.	_____ h
Fehleranalyse	Analyse von Log-Dateien und Prozessinstanzen (einzelne Durchläufe eines automatisierten Prozesses) bei Incidents. Erfordert tiefes Verständnis von Camunda-Prozessmodellen und Robocorp-Task-Logs.	_____ h
Hotfixes und Ad-hoc-Anpassungen	Kurzfristige Korrekturen an Prozessen oder Robotern, z. B. wenn sich eine Ziel-Webseite geändert hat oder eine API-Schnittstelle (Programmierschnittstelle – der Punkt, an dem zwei Systeme Daten austauschen) ein neues Format liefert.	_____ h
Incident-Dokumentation	Jedes gelöste Problem dokumentieren, damit es beim nächsten Mal schneller geht. Klingt selbstverständlich, wird in der Praxis fast immer vergessen.	_____ h
Summe Teil 3:		_____ h
+ Puffer (20–30 %):		+ _____ h
Summe Teil 3 mit Puffer:		_____ h

Was dieses Ergebnis bedeutet: Sie haben einen realistischen Puffer für unvorhergesehene Störungen und den täglichen Support-Bedarf geschaffen. Dieser Puffer verhindert, dass Ihr Team unter Dauerstress gerät.

Teil 4: Infrastruktur und Betrieb (GitOps/IaC) – das Rückgrat

Die Pflege Ihrer automatisierten Infrastruktur bestimmt die Stabilität des Gesamtsystems. GitOps (ein Ansatz, bei dem der gesamte Infrastruktur- und Deployment-Zustand in einem Git-Repository versioniert wird – jede Änderung ist nachvollziehbar und rücksetzbar) und IaC (Infrastructure as Code – Infrastruktur wird nicht manuell konfiguriert, sondern über Skripte definiert und automatisch aufgebaut) sind keine Luxus-Konzepte. Sie sind die Grundlage für Kontrolle, Auditierbarkeit und Wiederholbarkeit.

Aufgabe	Was genau zu tun ist	Ihre Schätzung: Stunden / Monat
CI/CD-Pipeline-Pflege	Wartung und Optimierung der Build-, Test- und Deployment-Pipelines (automatisierte Abläufe, die Code-Änderungen prüfen, testen und in die Produktionsumgebung überführen). Dazu gehört auch die Aktualisierung von Pipeline-Abhängigkeiten.	_____ h
Infrastruktur-Updates	Anpassung von IaC-Skripten (z. B. Terraform, Ansible) an neue Versionen oder geänderte Anforderungen. Umfasst auch Kapazitätsplanung (brauchen Ihre Worker mehr Ressourcen?).	_____ h
Monitoring und Alerting	Pflege und Anpassung von Dashboards (z. B. Grafana) und Alerting-Regeln (automatische Warnmeldungen bei Auffälligkeiten). Regeln müssen regelmäßig kalibriert werden, sonst produzieren sie entweder zu viele Fehlalarme oder übersehen echte Probleme.	_____ h
Backup und Recovery-Tests	Regelmäßige Prüfung, ob Ihre Backups tatsächlich wiederherstellbar sind. Wird fast immer vergessen – bis es zu spät ist.	_____ h
Summe Teil 4:		_____ h

Was dieses Ergebnis bedeutet: Sie haben den Aufwand für das technische Rückgrat Ihrer Automatisierungsplattform eingeplant. Ohne diese Investition ist Ihr Stack fragil, egal wie gut Ihre Prozesse modelliert sind.

Typisches Szenario aus der Praxis: Was passiert, wenn Sie Teil 1 unterschätzen

Ein Team plant den Umstieg auf Camunda und Robocorp. Die Entwicklung der ersten fünf Prozesse dauert drei Monate. Alles läuft. Sechs Monate später wird eine kritische Sicherheitslücke in einer Java-Bibliothek bekannt, die Camunda als Abhängigkeit nutzt.

Was schiefgeht: Da kein definierter Prozess für Security-Patching existiert, muss das Team am Wochenende ungeplant den Patch einspielen und testen. Der Test schlägt fehl, weil eine Robocorp-Schnittstelle mit der neuen Version inkompatibel ist. Der gesamte automatisierte Prozess steht für 48 Stunden still. Der Fachbereich eskaliert. Das Management stellt die Entscheidung für Open-Source grundsätzlich in Frage.

Was das Team hätte tun können:

- In Teil 1 realistische Stunden für monatliches Patching eingeplant.
- Ein Staging-Environment (Testumgebung, die die Produktionsumgebung nachbildet) permanent verfügbar gehalten.
- Kompatibilitätstests zwischen Camunda-Updates und Robocorp-Tasks als festen Schritt im Patching-Prozess definiert.

Die Lektion: Proaktive Planung von Wartungsfenstern und Testprozessen kostet einen Bruchteil dessen, was reaktive Krisenbewältigung an Arbeitszeit, Reputation und Nerven kostet.

Warum dieses Budget über den Erfolg Ihrer Automatisierungsstrategie entscheidet

Der Kern jeder Automatisierungsentscheidung ist nicht die Technologiewahl – es ist die Fähigkeit, bürokratische Prozesse dauerhaft, sicher und kontrolliert zu betreiben. Ein souveräner Stack aus Open-Source-Komponenten gibt Ihnen die Unabhängigkeit von teuren Lizenzmodellen und fremden Roadmaps. Aber diese Souveränität hat einen Preis: den Wartungsaufwand, den Sie gerade kalkuliert haben. Wer diesen Preis kennt und einplant, trifft eine fundierte Entscheidung. Wer ihn ignoriert, tauscht Vendor-Lock-in (die Abhängigkeit von einem einzelnen Software-Anbieter) gegen Betriebschaos.

Ihre Gesamtplanung: Alle Teile zusammengeführt

Übertragen Sie die Summen aus den vier Teilen in diese Abschlussrechnung.

Bereich	Monatliche Stunden
Teil 1 – Updates und Security Patches	_____ h
Teil 2 – Community und Wissensmanagement	_____ h
Teil 3 – Support und Incidents (mit Puffer)	_____ h
Teil 4 – Infrastruktur und Betrieb	_____ h
GESAMTSUMME pro Monat	_____ h

Umrechnung in FTE

Gesamtstunden pro Monat ÷ 160 (typische Arbeitsstunden pro Monat bei einer Vollzeitstelle) = **ca.**
_____ FTE

Wichtige Einschränkung: Diese Berechnung ist ein vereinfachtes Modell. Sie berücksichtigt nicht: Urlaubszeiten und Krankheitstage (planen Sie ca. 15–20 % Aufschlag für realistische Verfügbarkeit), Einarbeitungszeit neuer Teammitglieder, saisonale Schwankungen (z. B. erhöhter Patch-Aufwand nach großen Community-Releases), und die Lernkurve in den ersten 6–12 Monaten, in der der Aufwand typischerweise höher ist als im eingespielten Betrieb.

Entscheidungshilfe: Wie Sie dieses Ergebnis einordnen

- **Unter 0,25 FTE:** Realistisch nur für sehr kleine Stacks mit wenigen Prozessen. Prüfen Sie kritisch, ob Sie nicht zu optimistisch geschätzt haben.
- **0,25–0,5 FTE:** Typischer Bereich für einen mittelgroßen Stack mit 5–15 automatisierten Prozessen. Erfordert eine dedizierte Person mit anteiligem Zeitkontingent.
- **0,5–1,0 FTE:** Erwartbar bei größeren Stacks oder in regulierten Branchen. Rechtfertigt eine eigene Rolle oder ein kleines Team.
- **Über 1,0 FTE:** Prüfen Sie, ob Teile des Stacks sinnvoller als Managed Service (ein extern betriebener Dienst, bei dem der Anbieter Wartung und Betrieb übernimmt) bezogen werden

sollten. Hier greift die „Buy vs. Build“-Entscheidungsmatrix.

Was Sie erreicht haben

Sie haben eine fundierte, nachvollziehbare Grundlage für die Personal- und Budgetplanung Ihres souveränen Automatisierungs-Stacks geschaffen. Diese Kalkulation ist Ihr stärkstes Argument in der internen Diskussion – sie zeigt, dass Sie die Kosten kennen, statt sie zu verdrängen.

Ihr nächster Schritt

Schritt 1 – Vertiefen Sie Ihre Planung:

Nutzen Sie den Architektur-Blueprint für den souveränen Stack auf buerokratie-automatisieren.de, um die technische Grundlage für Ihre Budgetplanung abzusichern. Dort finden Sie die Referenzarchitektur, auf der dieses Maintenance-Budget aufbaut.

→ buerokratie-automatisieren.de/architektur-blueprint

Schritt 2 – Holen Sie sich externe Einschätzung:

Besprechen Sie Ihre Kalkulation mit einem erfahrenen DevOps- oder Automatisierungsberater, der Open-Source-Stacks in vergleichbaren Unternehmensgrößen betrieben hat. Ein externer Blick deckt blinde Flecken in Ihrer Schätzung auf, bevor sie im Betrieb zu Problemen werden.

Speichern oder drucken Sie dieses Dokument. Es ist die Basis für Ihre operative Planung und Ihre Argumentation gegenüber dem Management.

buerokratie-automatisieren.de

Informationsportal der J.v.G. technology GmbH

Unabhängige Inhalte rund um die Automatisierung bürokratischer Prozesse.

JvGLabs

AI visibility architecture

Arbeitsblatt jetzt herunterladen