

Automatisierung gestern – KI heute: Ein Blick zurück nach vorn

Martin Polak

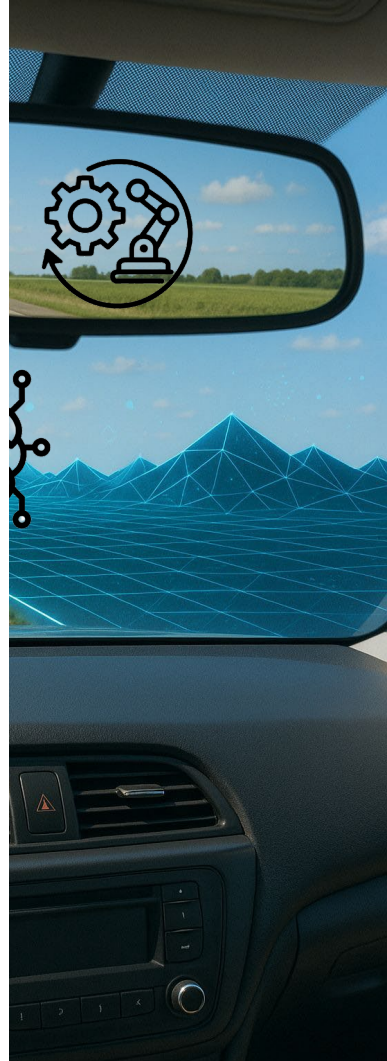
2025

Vorstellung

- Ex-UC4/Automic Consulting (Professional Services)
- 15 Automic/CDA Implementierungen, 14 Jahre, 5 Länder, 4 Kontinente
 - Produkt-Ablösungen
 - “Green field”
 - Misch-Szenarien
 - Verschiedenste Branchen: Banken/Finance, Behörden, Automotive, Handel, IT, ...
- Seit 2023: Solution Engineer @ Broadcom für Automation (alle Produkte)
 - Verantwortlich für strategische Kunden
 - “Link” zwischen (ausgewählten) Kunden & Engineering/Support/Sales/Product Mgmt

Automatisierung im Rückspiegel

- Review von 3 Implementierungen
- Hätte AI geholfen?
- Was wäre heute anders? Was bringt die Zukunft?



Fallbeispiele



Beispiel 1

Situation (Pilot)

- Große deutsche Behörde
- Ziel: Pilot für Automatisierung (3 Applikationen)
- Zeit: 6 Wochen, Team: 2 Personen (P. & ich)
- Erwartung: Technische Umsetzung + Überzeugung Management



Beispiel 1

Situation – das macht's nicht einfacher...

- Applikationen: DB + Web + Backend + Konfig
- Viele Schnittstellen (intern & extern)
- Dokumentation: 60–80 Seiten je App (200+ Seiten)
 - Qualität: ungleichmäßig, teils lückenhaft, chaotisch



Beispiel 1

Herausforderungen

- Kaum Ansprechpartner verfügbar
- Mehrere Schnittstellen zu (uns) unbekannten Systemen
- Komplexe Deployment-Reihenfolge & Logik
- Wenig Tool-Erfahrung
- Doku & Präsentation zusätzlich gefordert



Beispiel 1

Realität damals

- (Hotel-) Nächte mit Dokumenten-Studium (60-70h / Woche)
- Trial & Error bei Schnittstellen
- Skripte mühsam per Hand (Powershell – uns davor unbekannt)
- Dokumentation „händisch“ aus Logs
- Präsentation auf den letzten Drücker

Beispiel 1

Wie/Wo hätte KI geholfen?

- **Dokumentenanalyse:** Zusammenfassungen, Checklisten, Extraktion wichtiger Punkte
- **Schnittstellenanalyse:** Mapping, Validierung
- **Script-Generierung:** Deployments modular & fehlerarm
- **Doku & Präsentation:** automatisch aus Ergebnissen
- **Analyse von Fehlerfällen mit unbekannten Systemen:** Rechte? Schnittstelle? Anderes?

Beispiel 2

Situation (Implementierung)

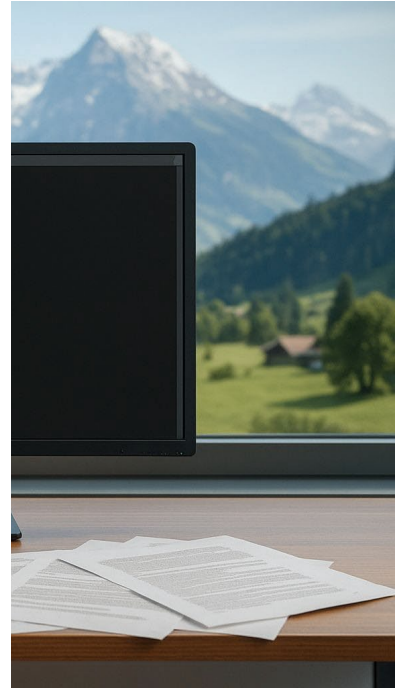
- Schweizer IT-Dienstleister
- Ziel: Ablösung der bestehenden Automatisierung für Deployment von ca. (miteinander verwobenen) 15 Applikationen
- Zeit: 6 Monate, Team: ich
- Erwartung: Technische Umsetzung, gute Wartbarkeit



Beispiel 2

Situation – das macht's nicht einfacher...

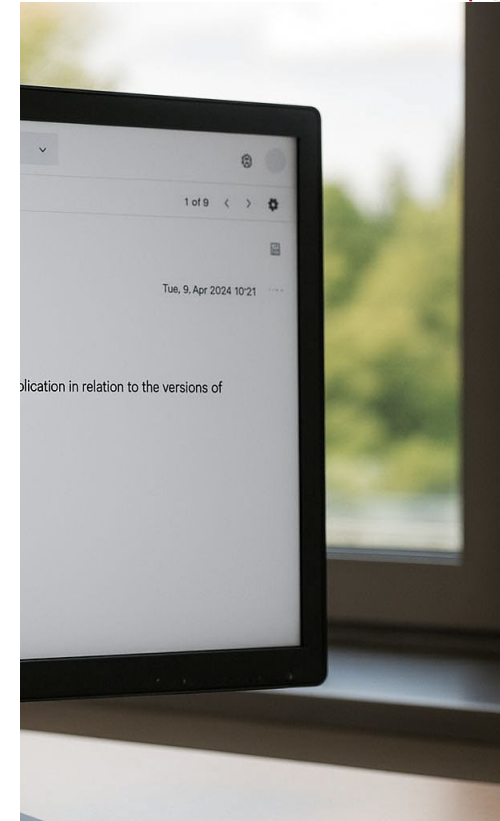
- Bisheriges Deployment in einem einzigen, riesigen bash-Skript
 - 500+ Seiten Doku zum Skript; unbrauchbar
- Ersteller & Warter des Skripts in Rente
- Anpassungen notwendig
- Unklare Quellsysteme bei Fremdherstellern
- Unklare Verarbeitungslogik
- Unklare & unterschiedliche Zielversionen und Versions-Kombinationen bei Kunden
 - Nicht alle Versionen Kompatibel
 - Kunde I bekommt andere Version von App C als Kunde II



Beispiel 2

Herausforderungen

- Monolithisches Skript, keine Modularisierung
- Kommentare im Skript unvollständig, irreführend
- Abhängigkeiten zu Dutzenden Umgebungsvariablen, Schnittstellen
- Kommunikation bez. Versionen
größtenteils via unstrukturierter e-Mails
- Dokumentation nicht nutzbar
- Gefahr: Jeder Eingriff konnte den gesamten Deployment-Prozess lahmlegen.



Beispiel 2

Realität damals

- Wochenlange manuelle Analyse des Skripts (Zeile für Zeile)
- Nicht alles umsetzbar
- Manuelle Eingriffe und Konfiguration nach wie vor notwendig
- Migration in Automatisierungstool unter massivem Zeitdruck & Workarounds
- Transparenz, Flexibilität, Wartbarkeit & Resilienz nicht vollständig gegeben
- Kunde nicht 100% zufrieden

Beispiel 2

Wie/Wo hätte KI geholfen?

- Analyse des Bash-Skripts
- Automatische Strukturierung (Aufteilung in Module, Funktionsblöcke, Abhängigkeitsdiagramme) & Dokumentation des Skripts
- Hervorhebung redundanter oder toter Code-Pfade
- Analyse der e-Mails und autom. Verarbeitung der Versions-Abhängigkeiten

Beispiel 3

Situation (Implementierung)

- Österr. Handelsunternehmen
- Geplanter Einstieg in Online-Handel
- Getrennt nach Ländern & Produktgruppen (Lebensmittel & Rest)
- Eigenständige Teams, identer Technologie-Stack der Shops



Beispiel 3

Situation – das macht's nicht einfacher...

- Jedes Team hat eigene Prozesse hinter Webshops
- Jedes Team verwendet eigene Tools zur (teilweisen) Automatisierung
- Jedes Team hat andere “sets” von Konfigurationsparametern
- Häufiges manuelles Eingreifen bzw. Nachbessern nötig



Beispiel 3

Herausforderungen

- Tool-Wildwuchs
- Uneinheitliche, undokumentierte Prozesse
- Starke Abhängigkeiten zu gemeinsamen zentralen Komponenten
- Zeitdruck
- Fehler durch manuelle Schritte
→ spät bemerkbar, und durch Logs der Systeme mühsam nachvollziehbar



Beispiel 3

Realität damals

- Manuelles Abstrahieren und Konstruieren eines Prozesses, der für alle Teams passt
- Standards für Developer-Deliverables selbst definieren, dokumentieren und durchsetzen
- Extrahieren der Automatisierungen aus verschiedensten Tools (versch. Skripte, Jenkins, Ansible, ...)
- Händisches Durchforsten sehr vieler Logs, um herauszufinden, wo manuelle Fehler passiert sind & scheinbar Prozesse nicht teamübergreifend passen

Beispiel 3

Wie/Wo hätte KI geholfen?

- Automatisches Erkennen von Prozessen aus Jira-Tickets, E-Mails, Dokumenten
- Konstruieren des “besten” gemeinsamen Prozesses
- Root cause Analysis der manuellen Fehler durch Log-Analyse & Vergleiche

Konklusio



Konklusio 1: Direkte Schlussfolgerungen

- KI als "**Automatisierung der Vor-Automatisierung**"
- Automation Engineer: **Vom Archäologen zum Architekten**
 - Früher: Alte Skripte, Doku, ... durchforsten
 - Heute: Mit KI-Unterstützung Blaupausen entwerfen, ... Fokus verschiebt sich von reaktiver Analyse zu proaktivem Design
- **Qualität durch Beschleunigung**
 - Früher "Garbage In, Garbage Out"
 - Heute: Identifikation von "Müll" mithilfe KI

Konklusio 2: Wohin geht's?

- **Paradoxon der Komplexität:** Vereinfachung durch KI führt zu neuer Form der Komplexität.
 - Statt riesigem Skript (Bsp 2): KI schlägt hunderte von Modulen, Abhängigkeiten etc. vor.
- **Demokratisierung** der Legacy-Transformation: Komplexe Ablösungen bisher nur mit erfahrenen Experten möglich; KI könnte Expertise tw "demokratisieren"
- **Automatisierung wird Konversations-Schnittstelle:** Information liegt in unstrukturierter Form vor (E-Mails, Tickets, Word-Dokumente, ...) → *"Analysiere die E-Mails von Team X der letzten 6 Monate und erstelle einen Workflow-Entwurf für die Versionierung von App C"*

Konklusio 3: Meine Thesen

- **KI bedroht nicht den Job, sondern die Trägheit**
Gewohnheiten: "Das haben wir schon immer so gemacht", "Ist historisch gewachsen", ...
Wer den Mut hat, KI als Erster in der Automatisierung konsequent zu nutzen?
- Ist "perfekte" Dokumentation noch erstrebenswert?
Schlechte Doku sind oft ein Problem. KI ist in der Lage ist, den Zustand eines Systems "live" zu analysieren
→ **Fokus verschiebt sich zu selbst-beschreibenden Systemen**



Danke