

Projekt Automatisierte Datenextraktion und Integration aus Auftragsplanungssoftware

1. Problemstellung

Im Unternehmen wird derzeit ein historisch gewachsenes Auftragsplanungssystem (APS) genutzt, das in den 1990er-Jahren in COBOL entwickelt wurde. Die Anwendung ist monolithisch, bietet keine API und erlaubt keine direkte Anbindung an externe Systeme. Dennoch sind die Planungsdaten für nachgelagerte Prozesse in Produktion und Logistik essenziell. Derzeit können lediglich manuell erstellte CSV-Reports aus dem APS exportiert werden. Diese manuelle Bedienung ist fehleranfällig, zeitraubend und nicht skalierbar.

2. Zielgruppe der Lösung

Die Nutzer der Lösung sind Systemverantwortliche und Entwickler im Unternehmen, die Daten aus dem APS automatisiert extrahieren und für weitere Verarbeitungsschritte bereitstellen wollen, etwa für BI-Auswertungen oder Produktionssteuerung.

3. Ziel der Softwarelösung und Vorteile

Ziel des Projekts ist die Entwicklung einer modernen, automatisierten Schnittstellenlösung, die:

- in regelmäßigen Intervallen (alle 15 Minuten) eine definierte Klickabfolge in der GUI des APS automatisiert ausführt,
- die dabei generierten CSV-Reports abrufen,
- und diese über ein SQL Server Integration Services (SSIS) Package in eine SQL-Datenbank überträgt.

Die Vorteile der Lösung:

- Automatisierung eines bisher vollständig manuellen Prozesses
- Fehlerreduktion durch wiederholbare und getestete Abläufe
- Zeitersparnis durch Hintergrundverarbeitung
- Erweiterbarkeit für zukünftige Schnittstellen oder Formate

4. Technologiewahl und Begründung

- Python wird zur Automatisierung der GUI-Klickabfolge genutzt (pyautogui) und ermöglicht durch Bilderkennung die Interaktion mit der Softwareoberfläche.
- Visual Studio (SSIS) verarbeitet die generierten CSV-Dateien und überträgt die Daten in die SQL-Datenbank.
- Die Lösung läuft automatisiert auf einer Windows-VM mit festgelegten Intervallen (Scheduler).
- Fehlerhandling, Logging und Statusprüfungen werden in Python implementiert.

5. Softwarearchitektur & Testbarkeit

- Die Softwarelösung ist modular aufgebaut:
 - Modul 1: GUI-Automation (Python)
 - Modul 2: CSV-Verarbeitung & ETL (SSIS)
 - Modul 3: Tests (Vergleich von Musterdaten mit DB-Inhalten)
- Die Architektur wird als Komponentendiagramm mit UML oder C4-Model dokumentiert.
- Unit-Tests auf der Python-Seite testen Datei-Erstellung, Pfadverfügbarkeit und Inhalt.
- Integrationstests prüfen die Verarbeitung bis zur Datenbank.