

Allgemeine Informationen

Quick Load



1. Projektplanung

Die Grundlage des Projekts *Quick Load* war die Idee, im Rahmen der schulischen **Projektarbeit** ein Auftragssystem für die bestehende Industrieanlage 4.0 zu entwickeln. Ziel war es, die bisher lokal gebundene Bedienung am Steuerungs-PC durch eine moderne, ortsunabhängige Lösung zu ersetzen, die auch auf **Schul-MacBooks** funktioniert.



Industrieanlage 4.0

Die Anwendung sollte eine **intuitive grafische Oberfläche** bieten, mit der sich Aufträge einfach erstellen, speichern und bearbeiten lassen. Dabei wurde besonderen Wert gelegt auf die **strukturierte Datenspeicherung im XML-Format**, eine **automatisierte Validierung mittels XSD** und eine **Umsetzung in Python**. Die grafische Oberfläche wurde mit **PyQt** realisiert.

1.1 Zieldefinition und Ausgangssituation

Im bisherigen Zustand konnten Produktionsaufträge ausschließlich direkt an der Industrieanlage eingegeben werden, was umständlich, intransparent und unflexibel war. Die Idee war deshalb, ein separates System zu entwickeln, das die Erstellung und Verwaltung dieser Produkte/Aufträge ermöglicht, ohne dass der Bediener physisch an der Anlage sein muss.

Das Ziel war eine Lösung zu schaffen, die:

- **von jedem Schul-MacBook aus nutzbar ist**
- **eine übersichtliche und leicht bedienbare GUI bietet**
- **die Auftragsdaten in XML speichert**
- **diese Daten mit einer XSD-Datei auf Gültigkeit prüft**

Zudem sollte die Software möglichst wartbar, nachvollziehbar und erweiterbar sein.

2. Implementierung

Die technische Umsetzung von Quick Load erfolgte vollständig in Python. Ziel war es, eine Anwendung zu entwickeln, die Aufträge für eine bestehende Industrieanlage 4.0 nicht nur lokal am Steuerungs-PC, sondern auch ortsunabhängig über Schul-MacBooks erstellen kann.

2.1 XML-Datenhandling

Da die Industrieanlage XML-Dateien als Eingabeformat nutzt, musste eine robuste Lösung zur Verwaltung, Validierung und Erzeugung dieser Dateien geschaffen werden. Dabei wurde eine eigene Python-Klasse namens „ProductXMLManager“ entwickelt. Diese bietet folgende Funktionen:

- **Laden der aktuellen XML-Datei** aus dem Projektverzeichnis
- **Validierung** der XML-Dateien anhand eines zugehörigen **XSD-Schemas**
- **Hinzufügen neuer Produkte** inklusive beschriebener Arbeitsschritte
- **Speichern und Umbenennen** der XML-Dateien mit Zeitstempel
- **Überprüfung auf bereits vorhandene Produkte**, um Duplikate zu vermeiden

Zusätzlich wurden umfassende **Tests zur XML-Verarbeitung** durchgeführt, um die Datenkonsistenz sicherzustellen. Die Validierung erfolgt dabei mithilfe der Bibliothek „xmlschema“, der XML-Aufbau mit „xml.etree.ElementTree“, Formatierung mit „minidom“ sowie Dateiverwaltung über „os“ und „datetime“.

Ein Auszug aus den verwendeten Imports:

```
import xml.etree.ElementTree as ET
import xmlschema
import os
from datetime import datetime
from xml.dom import minidom
```

Die bestehende XML-Datei der Anlage wurde nicht verändert, sondern eingebunden.

2.2 Grafische Benutzeroberfläche (GUI)

Die Entwicklung der grafischen Benutzeroberfläche erfolgte mit **PyQt6** und dem **Qt Designer**, wodurch eine moderne, intuitive und betriebssystemunabhängige Oberfläche geschaffen wurde.

Funktionen der GUI:

- Eingabe von Produktnamen, Beschreibungen und Arbeitsschritten
- Auswahl typischer Produktionsfunktionen über Dropdowns
- Übergabe der eingegebenen Daten an das XML-Modul

Zum Einsatz kam die Bibliothek PyQt6:

```
from PyQt6 import QtCore, QtGui, QtWidgets
```

2.3 Zusammenführung & Integration

Nachdem GUI und XML-Modul unabhängig entwickelt wurden, wurden beide Teile zusammengeführt, um eine durchgängige Funktionalität zu ermöglichen: Die Nutzereingaben aus der Oberfläche werden direkt als valide XML-Datei gespeichert. Dabei wurde sichergestellt, dass:

- alle Eingaben korrekt übergeben werden.
- die erzeugte XML-Datei vollständig dem XSD-Schema entspricht.
- die Oberfläche logisch und benutzerfreundlich arbeitet.

Es fanden **Tests** des vollständigen Systems mit Fokus auf die Datenverarbeitung statt und die korrekte Anbindung aller GUI-Elemente wurde sichergestellt. Damit wurde der Übergang von Benutzereingaben zur fertigen XML-Datei technisch sauber umgesetzt.

2.4 XML-Dateiaufbau

Die erzeugte Datei entspricht dem Format der Industrieanlage. Die XML-Datei beginnt mit dem Root-Element <Products> und enthält allgemeine Attribute wie Version und Creator.

Jeder Auftrag wird in einem <Product>-Element gespeichert. Dieses enthält:

- <ProductName>: Name des Produkts
- <ProductDescription>: Beschreibung
- <Workplan>: Enthält die Arbeitsschritte als <Step>-Elemente

Jeder Schritt hat:

- <Number>: Schritt-Nummer
- <Function>: Funktionskürzel
- <Parameter>: zugehöriger Parameter
- <FunctionDescription>: Beschreibung der Funktion

Beispiel:

```

1  <Products Version="1.0" Creator="Festo Didactic">
2    <Product>
3      <ProductName>Rote Kappe</ProductName>
4      <ProductDescription>Rote Kappe montiert für International</ProductDescription>
5      <Workplan>
6        <Step>
7          <Number>10</Number>
8          <Function>RR</Function>
9          <Parameter>1</Parameter>
10         <FunctionDescription>release red workpiece</FunctionDescription>
11        </Step>
12        <Step> ... </Step>
13      </Workplan>
14    </Product>
15
16    <Product> ... </Product>
17  </Products>

```

Die Struktur ermöglicht eine klare und maschinenlesbare Definition von Produkten mit allen notwendigen Bearbeitungsschritten. Dadurch können Aufträge direkt an die Industrieanlage übergeben werden.

2.5 XSD-Validierung

Zur Sicherstellung der korrekten Struktur wird die XML-Datei mit einer XSD-Datei validiert. Die XSD legt den Aufbau der Elemente, deren Datentypen und Wiederholungen fest.

Beispielsweise:

- <Product> darf mehrfach vorkommen
- <ProductName>, <ProductDescription> sind Pflichtfelder vom Typ xs:string
- <Step> enthält fest definierte Felder wie Number (xs:integer) und FunctionDescription (xs:string)
- Die Attribute Version und Creator im Root-Element sind erforderlich

So wird garantiert, dass die XML-Datei fehlerfrei und für die Anlage verwertbar bleibt.

3. Testphase

Nach der Implementierung wurde *Quick Load* ausführlich getestet, um die **Funktionalität, Stabilität und Benutzerfreundlichkeit** sicherzustellen.

3.1 Funktionale Tests

Funktionale Tests wurden durchgeführt und konzentrierten sich auf das **XML-Datenhandling**:

- Validierung erzeugter XML-Dateien gegen das XSD-Schema
- Test der Produkt-Duplikat-Erkennung
- Prüfung der Schrittstruktur (korrekte Nummerierung, Reihenfolge, Inhalt)
- Verhalten bei fehlerhafter XML-Datei

3.2 Integrationstests

Nach der Zusammenführung der Komponenten erfolgten gemeinsame **Integrationstests**:

- Durchgängige Testfälle vom Start der Anwendung bis zur XML-Erstellung
- Szenarien mit mehreren Produkten und komplexen Arbeitsschritten
- Validierung des erzeugten Outputs über das XSD-Schema

Die Tests bestätigten die korrekte Zusammenarbeit aller Module und eine durchgängig funktionierende Anwendung.

4. Bedienungsanleitung

Die Anwendung *Quick Load* wurde vollständig in Python unter Verwendung von **PyQt6** entwickelt. Nach der Installation aller benötigten Pakete kann das System direkt über das Hauptskript „Quickload.py“ gestartet werden.

Die Bedienung erfolgt über eine intuitive grafische Oberfläche (siehe Abbildung unten):

Produkt Industrie 4.0 Anlage erstellen:

Language selection: Deutsch

Produktname: Roter Zylinder mit Kappe

Produktbeschreibung: Es wird bei einem roten Zylinder eine Kappe montiert und die Farbe kontrolliert. Anschließend kommt es in das Fach für den internationalen

Werkstückauswahl: Werkstück rot

Zusatzauswahl: Kappe montieren, Farbe kontrollieren

Sortierungsauswahl: internationaler Vertrieb

Produkt: Roter Zylinder mit Kappe: Werkstück rot, Kappe montieren, Farbe kontrollieren, internationaler Vertrieb

Es wird bei einem roten Zylinder eine Kappe montiert und die Farbe kontrolliert. Anschließend kommt es in das Fach für den internationalen Vertrieb.

Produkt fertig

Bedienungsschritte:

1. **Produktname** und **Beschreibung** eingeben
2. **Werkstückfarbe**, **Zusatzfunktionen** und **Sortierung** auswählen
3. Der erzeugte Text fasst den Auftrag zusammen
4. Mit Klick auf **“Produkt fertig”** wird gefragt, ob der Auftrag gespeichert und in eine XML-Datei geschrieben werden soll.

The screenshot shows a web interface for creating a product in an Industry 4.0 plant. At the top left are logos for 'QUICK LOAD' and 'SBS'. The main title is 'Produkt Industrie 4.0 Anlage erstellen:'. On the right, there is a 'Language selection' dropdown set to 'Deutsch'. Below the title, there are two input fields: 'Produktname:' with the value 'Roter Zylinder mit Kappe' and 'Produktbeschreibung:' with a text area containing 'Es wird bei einem roten Zylinder eine Kappe montiert und die Farbe kontrolliert. Anschließend kommt es in das Fach für den internationalen'. Below these are three dropdown menus: 'Werkstückauswahl' (set to 'Werkstück rot'), 'Zusatzauswahl' (set to 'Kappe montieren'), and 'Sortierungsauswahl' (set to 'internationaler Vertrieb'). A fourth dropdown, 'Farbe kontrollieren', is located below the 'Zusatzauswahl' menu. A large white box displays the generated text: 'Roter Zylinder mit Kappe: Werkstück rot, Kappe montieren, Farbe kontrollieren, internationaler Vertrieb' and 'Es wird bei einem roten Zylinder eine Kappe montiert und die Farbe kontrolliert. Anschließend kommt es in das Fach für den internationalen Vertrieb.' At the bottom are two buttons: a red one labeled 'Eingaben nochmal verbessern' and a green one labeled 'Produkt ausgeben'.

Produkt Industrie 4.0 Anlage erstellen:

Language selection
Deutsch

Produktname: Roter Zylinder mit Kappe

Produktbeschreibung: Es wird bei einem roten Zylinder eine Kappe montiert und die Farbe kontrolliert. Anschließend kommt es in das Fach für den internationalen

Werkstückauswahl: Werkstück rot

Zusatzauswahl: Kappe montieren

Sortierungsauswahl: internationaler Vertrieb

Farbe kontrollieren

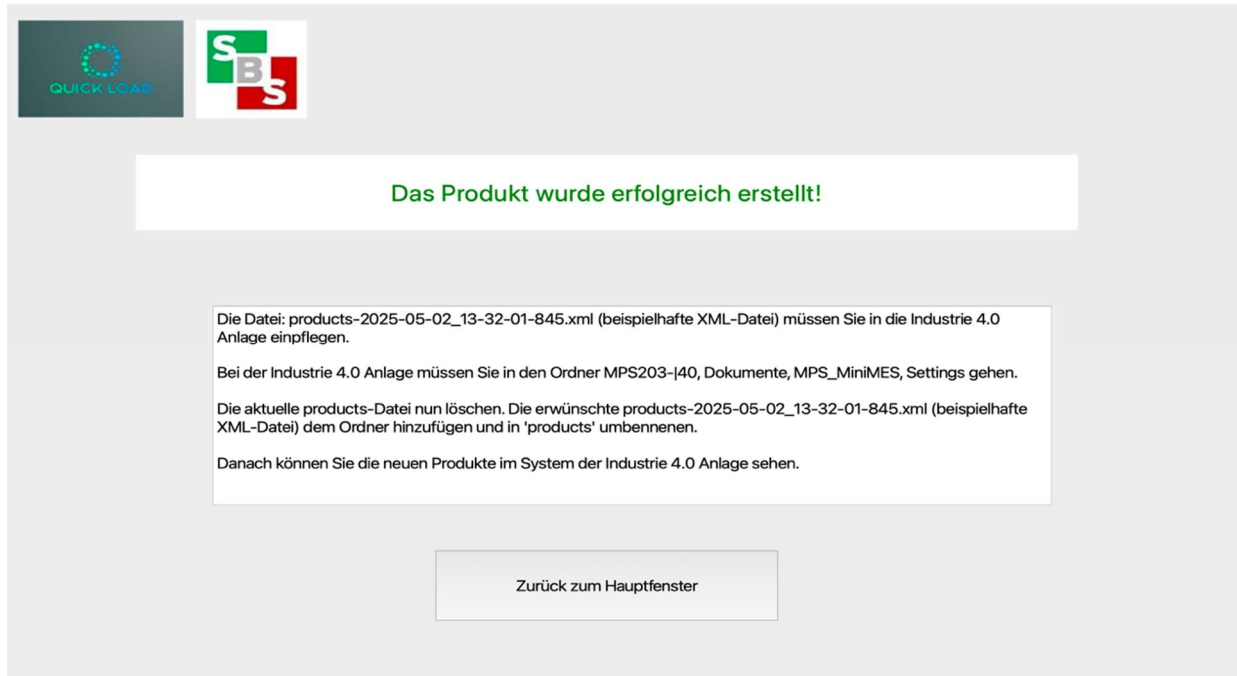
Roter Zylinder mit Kappe: Werkstück rot, Kappe montieren, Farbe kontrollieren, internationaler Vertrieb

Es wird bei einem roten Zylinder eine Kappe montiert und die Farbe kontrolliert. Anschließend kommt es in das Fach für den internationalen Vertrieb.

Eingaben nochmal verbessern

Produkt ausgeben

5. Bestätigt man diese Meldung mit „Produkt ausgeben“ wird das Produkt anschließend im Projektordner unter „products-2025-03-28_08-13-31-184.xml“ (beispielhafter Dateiname) gespeichert. Drückt man jedoch auf „Eingaben nochmal verbessern“ bietet sich die Chance das erstellte Produkt vor der Speicherung abzuändern und man gelangt zur ersten Oberfläche zurück.



6. Nun kann die erstellte products.xml-Datei in auf dem PC der Anlage eingebunden werden. Hierzu muss die Datei umbenannt werden in „products“. Danach muss sie im Ordner ... die alte „products“-Datei ersetzen. (Optionale Einbindung: Speichern der products.xml in einem Netzwerkordner.)

7. Zuletzt die Anwendung „MPS miniMES“ schließen und wieder öffnen. Jetzt lassen sich die neuen Produkte unter dem Reiter „Order Management“ bei den verfügbaren Produkten erkennen.

(8. Es gibt die Möglichkeit Quick Load auf 2 Sprachen (Deutsch & Englisch) zu nutzen. Hierzu kann man oben rechts ein Dropdown-Menü mit „Language selection“ sehen.)

5. Sicherheitshinweise

Die Datei products.xsd darf nicht verändert werden, da sonst die Strukturprüfung der XML-Datei fehlschlägt und keine Aufträge mehr eingelesen werden können.