

Zugänglichkeit des Quellcodes:

- Open Source: Bei Open-Source-Software ist der Quellcode für jedermann frei zugänglich. Jeder kann den Quellcode einsehen, ändern und verteilen, solange er sich an die Lizenzbedingungen hält. Dies fördert die Transparenz und Zusammenarbeit in der Entwicklergemeinschaft.
- Proprietär: Bei proprietärer Software ist der Quellcode nicht öffentlich zugänglich. Die Entwickler behalten die Kontrolle über den Code und geben normalerweise nur kompilierte Anwendungen heraus.

Kosten:

- Open Source: Die meisten Open-Source-Softwarelösungen sind kostenlos verfügbar, was bedeutet, dass man sie ohne Kosten herunterladen und verwenden kann. Es gibt jedoch keine Garantie, dass sie immer kostenlos bleiben, da einige Projekte möglicherweise in kostenpflichtige Varianten übergehen.
- Proprietär: Proprietäre Software ist oft kostenpflichtig. Sie müssen in der Regel eine Lizenzgebühr zahlen, um sie zu nutzen, und möglicherweise auch laufende Gebühren für Support und Updates.

Anpassungsfähigkeit und Flexibilität:

- Open Source: Da der Quellcode verfügbar ist, können Entwickler Open-Source-Software an ihre spezifischen Anforderungen anpassen. Dies ermöglicht eine hohe Flexibilität und Anpassungsfähigkeit.
- Proprietär: Bei proprietärer Software haben Sie in der Regel weniger Flexibilität bei der Anpassung, da Sie keinen Zugriff auf den Quellcode haben. Sie müssen sich auf die vom Hersteller bereitgestellten Anpassungsmöglichkeiten beschränken.

Community-Unterstützung:

- Open Source: Open-Source-Projekte profitieren oft von einer aktiven Entwicklergemeinschaft, die Fehler behebt, neue Funktionen entwickelt und Unterstützung bietet. Die Community kann ein wertvoller Ressourcenpool sein.
- Proprietär: Proprietäre Software wird normalerweise vom Hersteller unterstützt. Sie erhalten Support und Updates direkt vom Unternehmen, das die Software entwickelt hat.

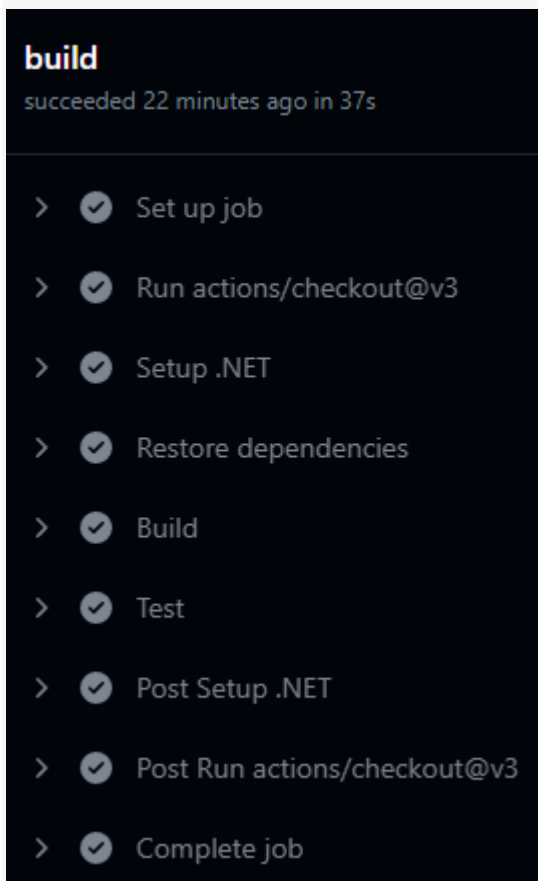
Lizenzbedingungen:

- Open Source: Open-Source-Software unterliegt verschiedenen Lizenzbedingungen wie der GNU General Public License (GPL) oder der MIT-Lizenz. Diese Lizenzen legen fest, wie der Quellcode verwendet, modifiziert und verteilt werden darf.

- **Proprietär:** Proprietäre Software wird normalerweise von Endbenutzer-Lizenzvereinbarungen (EULAs) geregelt, die die Bedingungen für die Nutzung und Verteilung der Software festlegen.

Insgesamt sind Open-Source-Lösungen offen, anpassbar und oft kostengünstig, während proprietäre Lösungen geschlossener sind, eine höhere Kontrolle durch den Hersteller bieten, aber in der Regel kostenpflichtig sind. Die Wahl zwischen ihnen hängt von den spezifischen Anforderungen, dem Budget und Präferenzen ab.

Unsere Infrastruktur ist Github Actions. Sie wird getriggert, wenn es einen push oder Pull-Request auf dem Mainbranch gibt. Der Workflow wird durch eine `yaml.file` konfiguriert. Dieser enthält mehrere Jobs, in unserem Fall ist dies ein Setup für das "Bauen" vom .NET-Projekt, das Klonen und Ausführen des Quellcodes. Es wird für die Umgebung .NET 6.0 verwendet. Dann werden Unit-Tests durchgeführt. Dann werden für die Postman Tests das Tool Newman benutzt.



```

1  # This workflow will build a .NET project
2  # For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-net
3
4  name: .NET
5
6  on:
7    push:
8      branches: [ "main" ]
9    pull_request:
10     branches: [ "main" ]
11
12  jobs:
13    build:
14
15     runs-on: ubuntu-latest
16
17     steps:
18     - uses: actions/checkout@v3
19     - name: Setup .NET
20       uses: actions/setup-dotnet@v3
21       with:
22         dotnet-version: 6.0.x
23     - name: Restore dependencies
24       run: dotnet restore Itech-Attendance/Itech-Attendance.Tests
25     - name: Build
26       run: dotnet build Itech-Attendance/Itech-Attendance.Tests --no-restore
27     - name: Test
28       run: dotnet test Itech-Attendance/Itech-Attendance.Tests --no-build --verbosity normal
29
30     # Add a step to install Newman (Postman CLI)
31     # - name: Install Newman
32     #   run: npm install -g newman
33
34     # Add a step to run Postman tests
35     # - name: Run Postman Tests
36     #   run: |
37     #     newman run Itech-Attendance/Itech-Attendance.Tests/*.json

```

Ein möglicher Lock-in-Effekt besteht auf jeden Fall zu Github, da bei einem Wechsel des Anbieters das komplette Repository mitsamt der Pipeline und Konfiguration umziehen müsste, was relativ viel Arbeit bedeuten würde. Grundsätzlich besteht ein großer Lock-in-Effekt zu Microsoft, da auch asp.net zu diesem gehört.

Man könnte hiermit am besten umgehen, indem man verschiedene Dienste benutzt. Man könnte zum Beispiel statt Github auch Gitlab benutzen und dort eine andere pipeline aufbauen. Außerdem müsste man die CI/CD Umgebung bestenfalls so konfigurieren, dass sie plattformunabhängig sind. Dies kann zum Beispiel mit Tools wie Jenkins erreicht werden. Somit hätte man eine deutlich geringere Abhängigkeit von Microsoft. Regelmäßiges Backup-Erstellen hilft ebenfalls beim Umzug zu einem anderen Repository. Die Firma sollte auf jeden Fall technisch auf dem neusten Stand sein und sich regelmäßig über neue Produkte und Technologien (open-source) auf dem Markt erkunden, um marktfähig zu bleiben und - falls nötig- Softwarekomponenten austauschen zu können. Zudem schadet es nicht, sich einen Plan oder eine Strategie zurechtzulegen, was für Alternativen es bereits gibt und auf

welche man zu Zeitpunkt X zurückgreifen könnte. Grundsätzlich ist für dieses Projekt die Benutzung von open-source software sinnvoll, da diese häufig funktional völlig ausreichend ist und für jeden Use-Case zutrifft. Außerdem ist open-source generell mehr flexibel und Sicherheitslücken können durch die Community, welche Einblick in den Code hat, rechtzeitig gefunden werden.