**Warum haben wir was benutzt? (Programmiersprache/ Umgebung, Github)**Bevor wir angefangen haben das Projekt durchzuführen und zu implementieren haben wir uns überlegt, welche Programmiersprache, welche Programmierumgebung und welche Plattform wir zur Versionsverwaltung benutzen wollen.  
**Warum Python:**MQTT kann mit verschiedensten Programmiersprachen, wie Java, C++ oder Python programmiert werden.  
Die Sprachen haben ihre eigenen Vor und Nachteile, wodurch sie besser oder nicht so gut geeignet sind.  
Java und C++ sind statisch typisiert und kompilierte Sprachen, was diese zur Laufzeit schneller macht. Der Code muss aber auf unterschiedlichen Betriebssystemen neu Kompiliert werden, da das Programm sonst nicht funktioniert.  
Auf der anderen Seite gilt bei Python „**write once, run anywhere**“ was bedeutet, dass der geschriebene Code auf allen Betriebssystemen funktioniert.   
Außerdem ist Python einfacher in der Verwendung und einfacher zu verstehen. Also im Vergleich zu Java oder C++ sind die Codezeilen kürzer.  
Wir haben uns für Python endschieden, weil der Code auf allen Betriebssysteme einfach ausgeführt werden kann. Ein weiterer Grund ist, das Python einfacher in der Verwendung ist.  
https://www.bmc.com/blogs/python-vs-java/  
https://www.bitdegree.org/tutorials/python-vs-c-plus-plus/  
  
**Atom:**  
Um in Python zu programmieren haben wir das Programm Atom benutzt. Dieses ist einfach zu bedienen und der Code kann über F5 direkt getestet/ausgeführt werden.  
  
Es wird eine Datei mit der Endung .py erstellt und schon kann mit der Programmierung begonnen werden.  
  
  
  
Es gibt verschiedenste Python-MQTT-Open-Source-Clientbibliotheken, wie paho-mqtt, HBMQTT und gmqtt. Die HBMQTT und gmqtt Bibliothek unterstützen nicht die neuste MQTT version.  
Deswegen ist die beste Bibliothek, um MQTT in Python zu programmieren die paho-mqtt Clientbibliothek.   
  
Zur Versionsverwaltung haben wir Github verwendet.  
Github ist einer der am meisten verwendeten Plattformen zur Versionsverwaltung. Wir haben mit Github gearbeitet, weil es viele Funktionen bietet um in einer Gruppe das Projekt immer aktuell zu halten.  
So kann z.B. über die Desktop App ganz einfach der Programmcode hochgeladen werden und jeder aus der Gruppe kann sich diesen über den Github Order auf seinem PC anschauen.