

Pflichtenheft und technische Spezifikation im Programmierprojekt

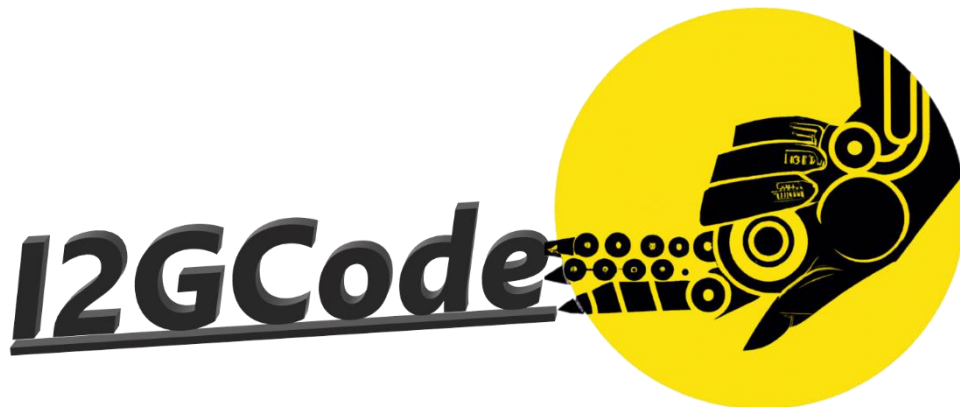
Image2GCode

Mitarbeiter: Leonardo Mateljan

Max Jacob Damm

Paul Seelig

Pierre M. Wiele



Inhaltsverzeichnis

1	Visionen und Ziele.....	1
2	Anforderungsanalyse	2
2.1	Funktionale Anforderungen / Use-Cases.....	2
2.2	Nicht-funktionale Anforderungen.....	2
2.3	Risiken	2
2.4	GUI	3
3	Realisierung	4
4	Entwicklungs- und Teststrategie.....	5
5	Kooperationen und Verwertungsplan	5

1 Visionen und Ziele

Dieses Pflichtenheft dokumentiert die Anforderungen und Spezifikationen für das Projekt Image2G-code. Es dient als Grundlage für die Planung, Entwicklung und Umsetzung dieses Projekts. Image2G-code ist eine Desktopanwendung, welche stark mit der Bildverarbeitung zusammenarbeitet, und hat als Ziel, aus einem hochgeladenen Bild die Laufwege einer CNC-/ Laser-/ Fräsmaschine zu berechnen und in Form von G-code auszugeben.

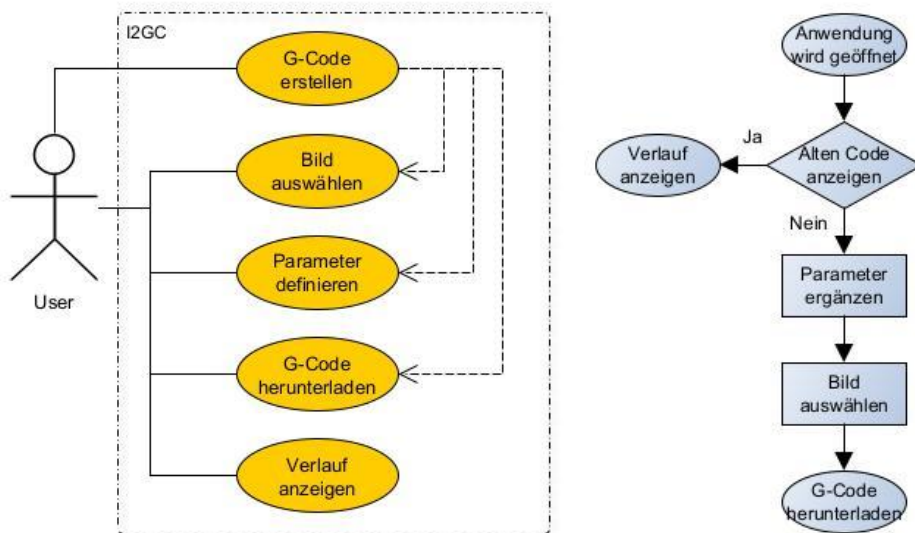
Die Vision hinter diesem Projekt ist es, Bilder und Namen fräsen/ lasern so einfach wie möglich zu machen. Wir streben danach, eine verständlich und einfach zu bedienende Nutzeroberfläche zu erreichen.

Die in diesem Pflichtenheft festgelegten Ziele sind darauf ausgerichtet, diese Vision in die Realität umzusetzen. Zu den wichtigsten Zielen gehören eine einfache und verständliche GUI, eine schnelle Analyse des Bildes und die Ausgabe des G-codes als Upload oder Download. Diese Ziele werden dazu beitragen, persönliche Projekte schneller fertigzustellen und oder die Möglichkeit erstmal darzustellen.

Dieses Pflichtenheft ist als Leitfaden für alle Beteiligten an Image2G-code gedacht und soll sicherstellen, dass alle Anforderungen und Spezifikationen klar definiert sind. Es dient als Referenzdokument, um sicherzustellen, dass das Projekt erfolgreich abgeschlossen wird und die gesetzten Ziele erreicht werden.

2 Anforderungsanalyse

2.1 Funktionale Anforderungen / Use-Cases



* Idealerweise sollte nach der Konturbearbeitung des Bildes verschiedene Versionen des Konturweges angezeigt werden.

* Möglicherweise wird zu einem späteren Zeitpunkt die Funktion "Gravieren" hinzugefügt.

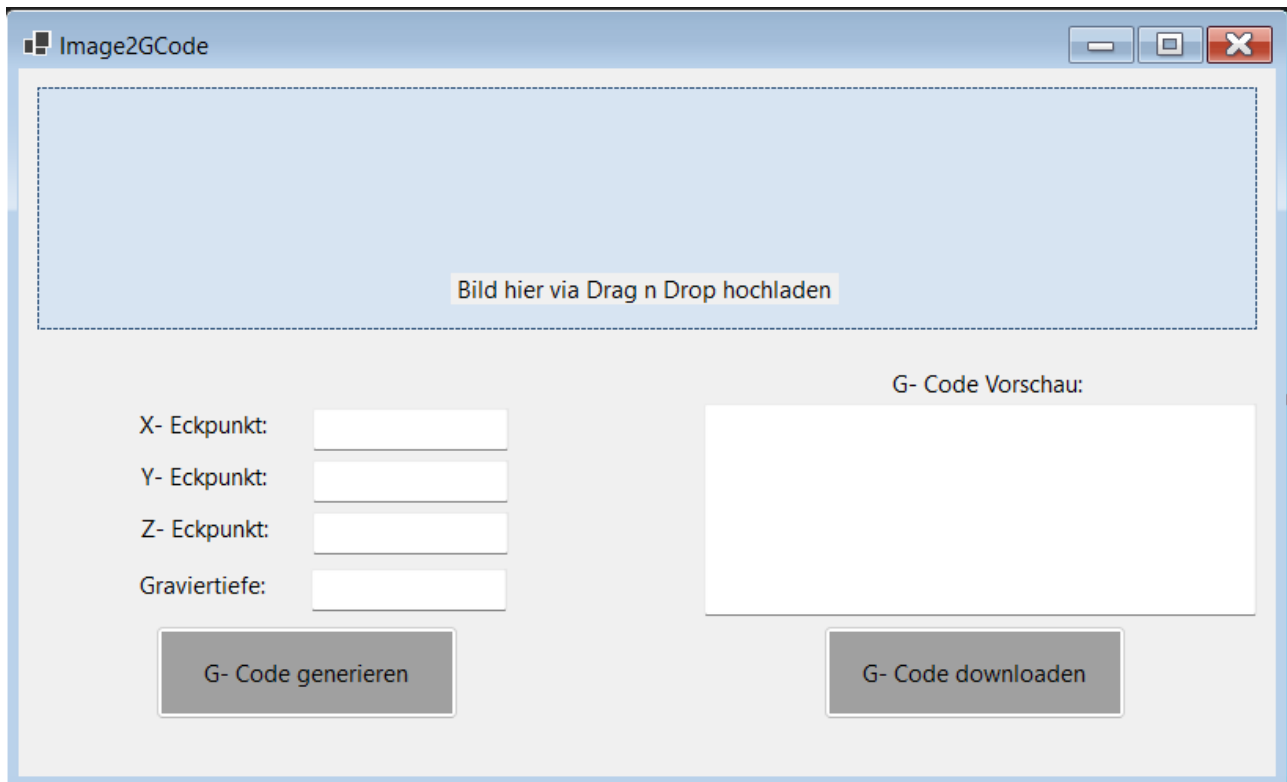
2.2 Nicht-funktionale Anforderungen

Die Software soll unter Windows 10 und 11 lauffähig sein, im .Net Standard2.0 Framework laufen und einen G-Code nach Eingabe der Parameter und Auswahl des (240 x 540 Pixel) Bildes generieren und soll dafür nicht länger als 30 Sekunden benötigen (die Zeitangabe bezieht sich auf ein System mit folgender Ausstattung: AMD Ryzen 5 5625U mit Radeon Graphics (2.30 GHz) und 16,0 GB Ram).

2.3 Risiken

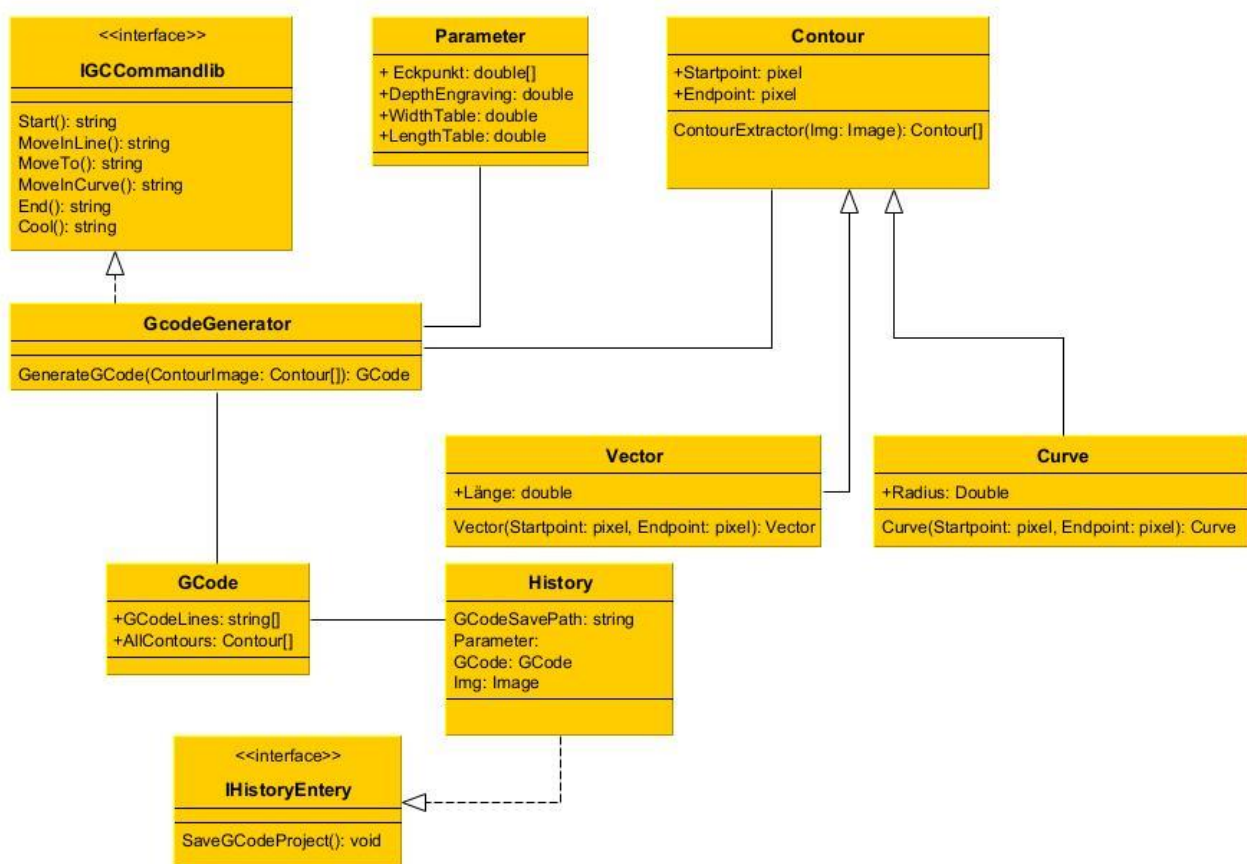
Risiken	Maßnahmen
Misskommunikation	Klare Zielsetzung und wöchentliche Meetings
Teile aus dem G-Code folgen nicht den Normen	Absprache mit Herr Schauer
G-Code NuGet unverständlich oder nicht für uns konzipiert	Selbst programmieren
Konturen sind nicht vollständig bzw. genau	Experimentieren mit verschiedenen Schwellenwerten (Grayscaleing verändern)

2.4 GUI



Das Ziel unserer Desktopanwendung soll es sein, G-Code für eine CNC-Fräse aus einem Bild zu generieren. Dies kann der User in der Dafür vorgesehenen PictureBox tun. Danach sollte er noch die Eckpunktkoordinaten eines Eckpunktes seines Werkstücks eingeben, dadurch kann der Quellcode dann die Position des Werkstücks in der CNC-Maschine, sowie die Höhe des Werkstücks errechnen (hierzu muss noch gesagt werden, dass man das Werkstück nur mittig einspannen kann, daher ist der Mittelpunkt immer gegeben). Optional haben wir hier noch die Graviertiefe Eingabe hinzugefügt, obwohl wir uns noch nicht sicher sind, ob wir die Gravioption final implementieren wollen, weil wir dann zwischen unterschiedlichen Fertigungsprozessen unterscheiden müssten. Wenn alles eingegeben hat und dann auf den Button G-Code generieren drückt, erscheint der generierte G-Code dann rechts in der TextBox als eine Art Vorschau, dort kann man sich den G-Code anschauen. Wenn man dann den Button G-Code downloaden drückt, soll eine .txt Datei angelegt werden mit dem eben generierten G-Code, welche dann in einem Ordner in den Persönlichen Dateien des Users erscheint.

3 Realisierung



Die Implementierung wird weitere Klassen enthalten, insbesondere die CVInvoke Class, Image und Point aus dem NuGet Emgu.CV. Außerdem wird möglicherweise für die GCode-Generierung die Klasse GCodeBuilder aus dem NuGet Sutro.gsGCode verwendet.

4 Entwicklungs- und Teststrategie

Für die Entwicklung der Software haben wir geplant das wir uns wie folgt aufteilen:

- Einer übernimmt das "Frontend", die Entwicklung der GUI und die Abnahme und Eingabe des Bildes und der dazugehörigen Parameter
- Zwei übernehmen die Bildverarbeitung, bzw. einer von beiden ist wahrscheinlich jemand der zwischen Frontend, Bildverarbeitung und G-Code-Generierung hilft
- Der letzte beschäftigt sich mit der G-Code-Generierung, also das von der Bildverarbeitung ausgegebene zu CAM lesbaren Code zu konvertieren

Hierbei ist es unwahrscheinlich, dass wir auf mehrere Branches zugreifen, aber das kann sich im Laufe des Projektes ändern.

Um die verschiedenen Funktionen zu testen, werden wir Unit Tests verwenden. Zusätzlich verwenden wir eine Bilddatenbank, um die Konturerkennung und die GCode-Generierung zu testen, dies machen wir mit bereits generiertem GCode, der zuvor von der CAD-Software generiert wurde.

5 Kooperationen und Verwertungsplan

Kooperationen mit anderen Teams sind derzeit nicht geplant. Die einzige Kooperation, die innerhalb dieses Projektes besteht, ist die zwischen Kai Schauer (Auftraggeber/Dozent) und Erik Rodner (Dozent), v.a. als evaluierende Kraft, aber auch als fachliche Unterstützung. Mit diesen wird ein regelmäßiger Austausch geplant und gepflegt, um jedwede Unklarheiten und Missverständnisse zu vermeiden, möglicherweise bei Problemen fachlichen Rat einzuholen und Transparenz zu schaffen, was die Fortschritte des Projektes angeht.

Aufgrund des geringen Ausmaßes des Projektes und seiner Vorteile gegenüber ähnlichen, schon existenten Lösungen wird es an dieser Stelle keinen ausführlichen Verwertungsplan geben.

Der Projektnutzen besteht vor allem im Erkenntnisgewinn und das Projekt selbst ist eine auf einen bestimmten Maschinentyp zugeschnittene Lösung unter der MIT-Lizenz (Open Source).