

Erstellung einer GUI für die  
Pythonic mmWave Toolbox

# Agenda

Zu Beginn wird die Aufgabenstellung beschrieben, um anschließend auf die genutzten Muster einzugehen. Daraufhin werden die einzelnen Sprintergebnisse vorgetragen, damit abschließend das Resultat live demonstriert werden kann.



- Einleitung

- Problembeschreibung
- Genutzte Muster

- Sprint

- Ergebnisse und Ziele der einzelnen Sprints

- Ende

- Live Demonstration



# Agenda

Zu Beginn wird die Aufgabenstellung beschrieben, um anschließend auf die genutzten Muster einzugehen. Daraufhin werden die einzelnen Sprintergebnisse vorgetragen, damit abschließend das Resultat live demonstriert werden kann.



- **Einleitung**
  - Problembeschreibung
  - Genutzte Muster
- Sprint
  - Ergebnisse und Ziele der einzelnen Sprints
- Ende
  - Live Demonstration



# Problembeschreibung

Nicht jeder ist in der Lage die Kommandozeile zu verwenden. Hierfür ist eine graphische Benutzeroberfläche Vorteilhaft, da Menschen gewohnt sind über graphische Benutzeroberflächen mit Rechnern zu interagieren.

- Entwicklung einer GUI für Pythonic mmWave Toolbox
- Programmierung in Python mit PyQt5
- Ziele:
  - Nutzung der Pythonic mmWave Toolbox über GUI
  - Verbindung mit IWR6843AOP EVM
- Einfachheit der Benutzung im Vordergrund
- SCRUM basierte Entwicklung



# Agenda

Zu Beginn wird die Aufgabenstellung beschrieben, um anschließend auf die genutzten Muster einzugehen. Daraufhin werden die einzelnen Sprintergebnisse vorgetragen, damit abschließend das Resultat live demonstriert werden kann.

- **Einleitung**

- Problembeschreibung

- **Genutzte Muster**

- **Sprint**

- Ergebnisse und Ziele der einzelnen Sprints

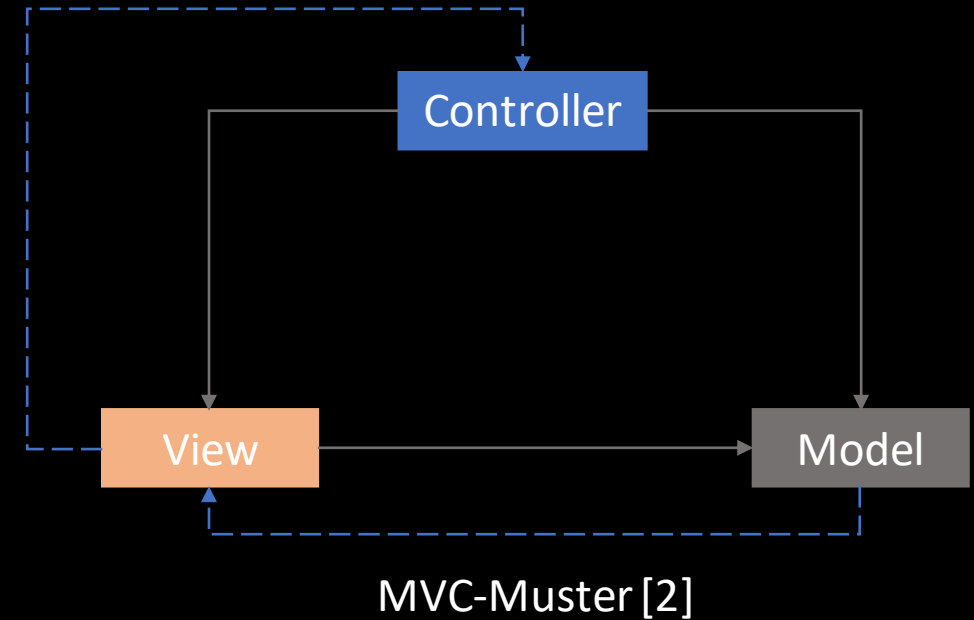
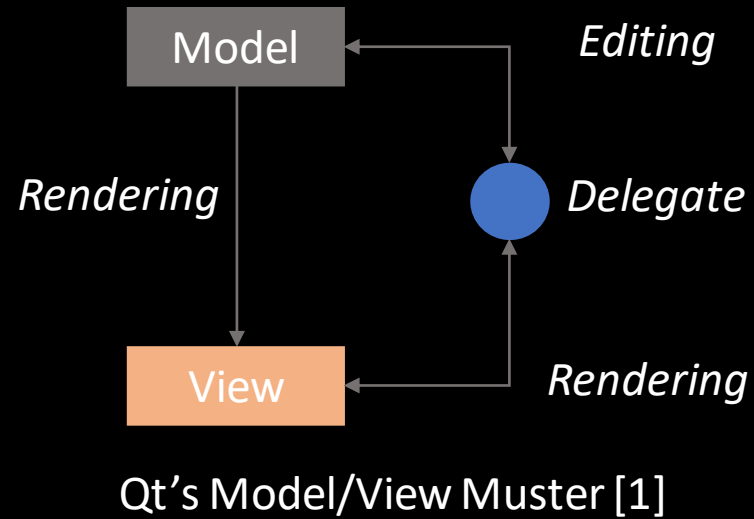
- **Ende**

- Live Demonstration



# Genutzte Muster

Zwei unterschiedliche Muster sind genutzt worden. Zu Beginn das Qt-eigene Model/View Muster. Anschließend auf Bitte der Stakeholder, Prof. Dr.-Ing. Igel sowie Dr. Pietrek, in das Model-View-Controller-Muster umgeschrieben.



# Agenda

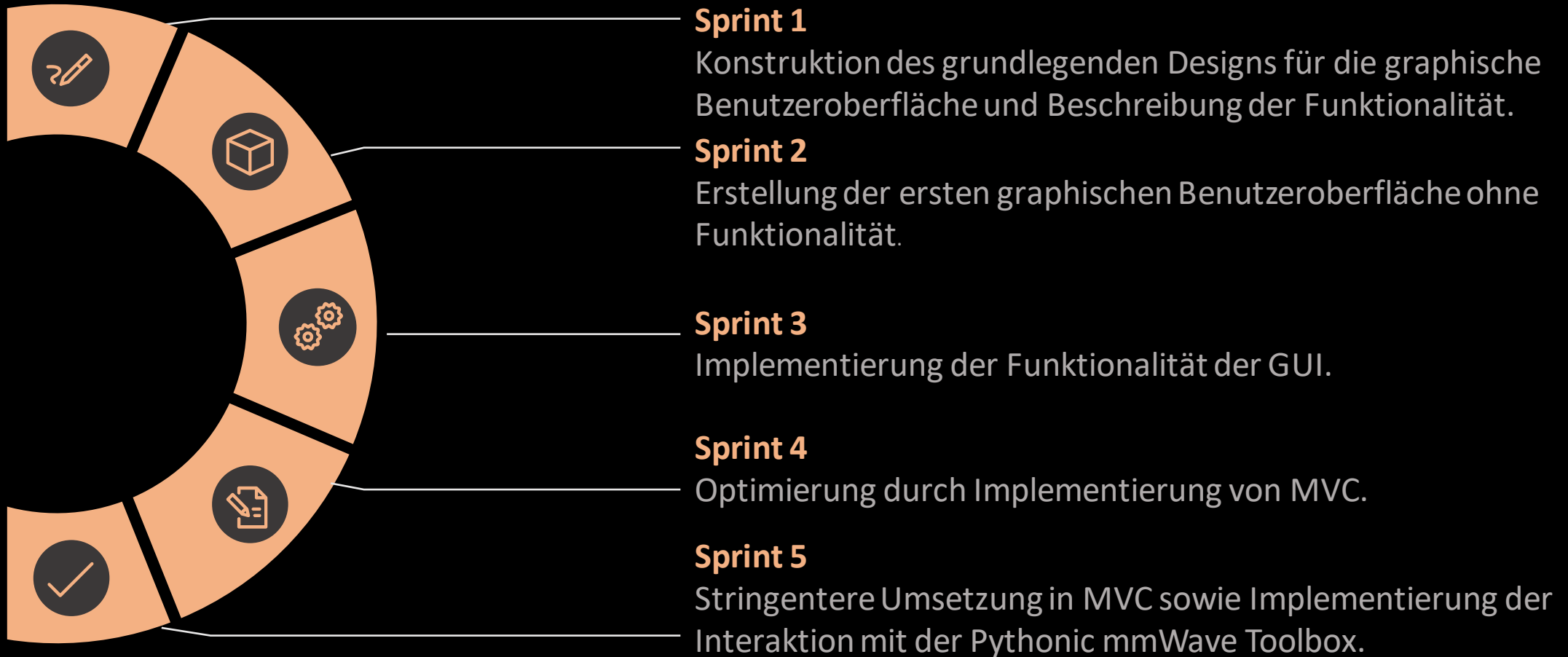
Zu Beginn wird die Aufgabenstellung beschrieben, um anschließend auf die genutzten Muster einzugehen. Daraufhin werden die einzelnen Sprintergebnisse vorgetragen, damit abschließend das Resultat live demonstriert werden kann.

- Einleitung
  - Problembeschreibung
  - Genutzte Muster
- Sprint
  - Ergebnisse und Ziele der einzelnen Sprints
- Ende
  - Live Demonstration



# Ergebnisse und Ziele der einzelnen Sprints

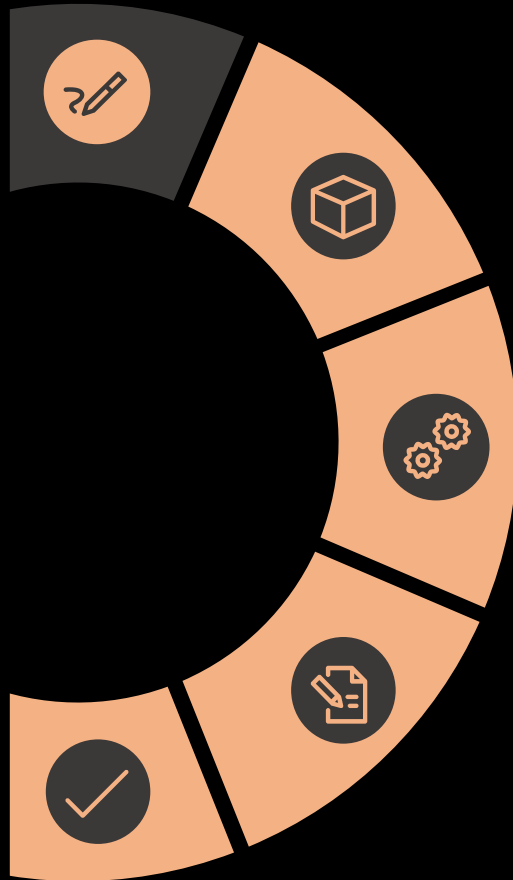
Ein Überblick über die fünf Sprintziele des Entwicklungsprozesses der graphischen Benutzeroberfläche. Im folgenden wird nochmal genauer auf die fünf Sprints und deren Sprintziele eingegangen, sodass der Entwicklungsprozess verstanden wird.



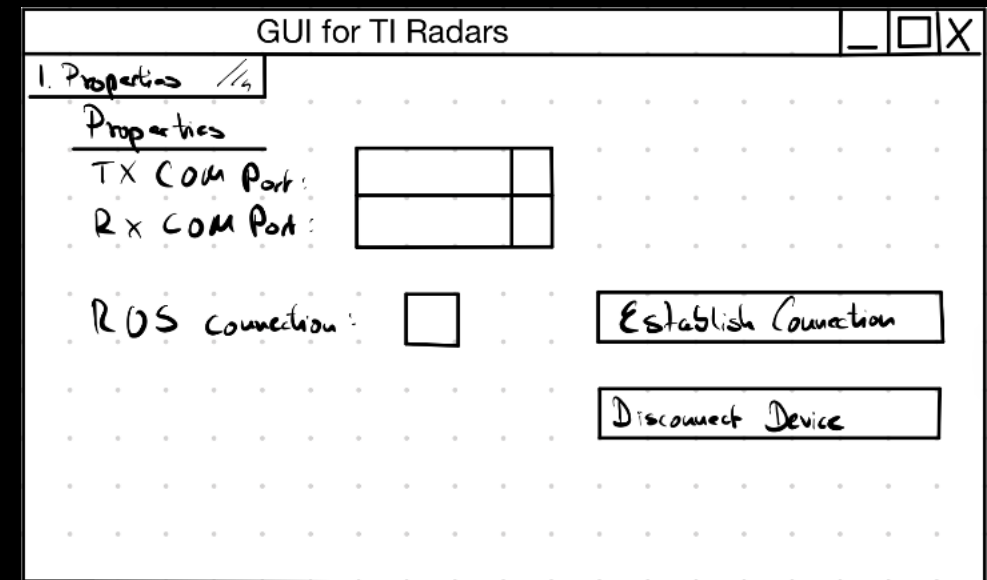


# Ergebnisse und Ziele der einzelnen Sprints – Sprint 1

Konstruktion des grundlegenden Designs für die graphische Benutzeroberfläche und Beschreibung der Funktionalität.

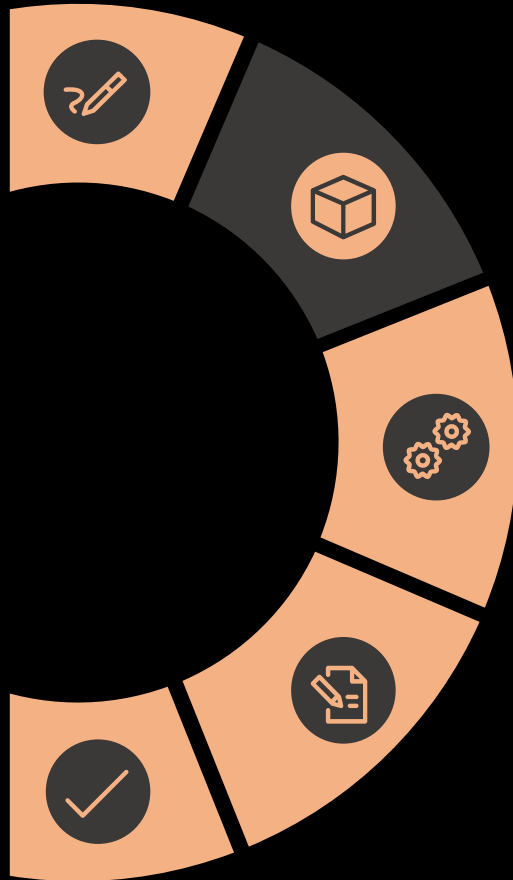


Sprintplanung	Backlog
<ul style="list-style-type: none"> <li>Definition der GUI.</li> <li>Definition der Funktionalität.</li> </ul>	<ul style="list-style-type: none"> <li>Warten auf Informationen aus dem ersten Sprint, um Backlog zu füllen.</li> </ul>



# Ergebnisse und Ziele der einzelnen Sprints – Sprint 2

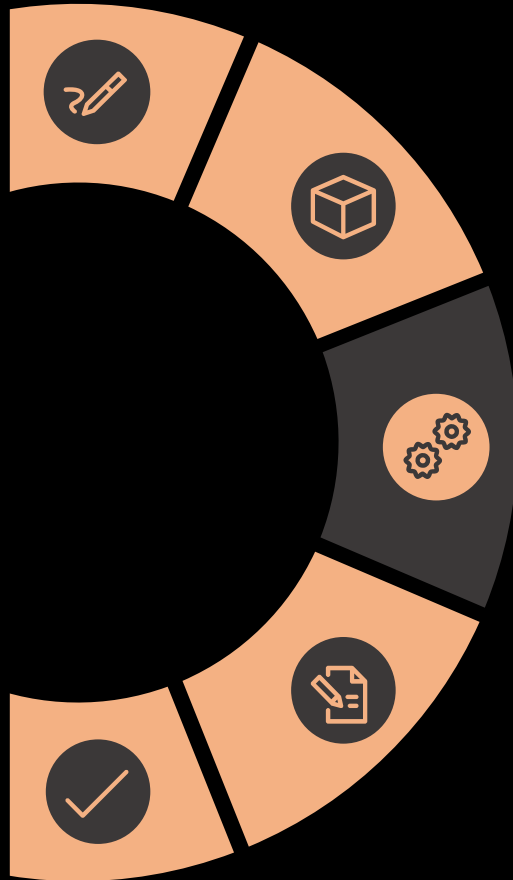
Erstellung der ersten graphischen Benutzeroberfläche ohne Funktionalität.



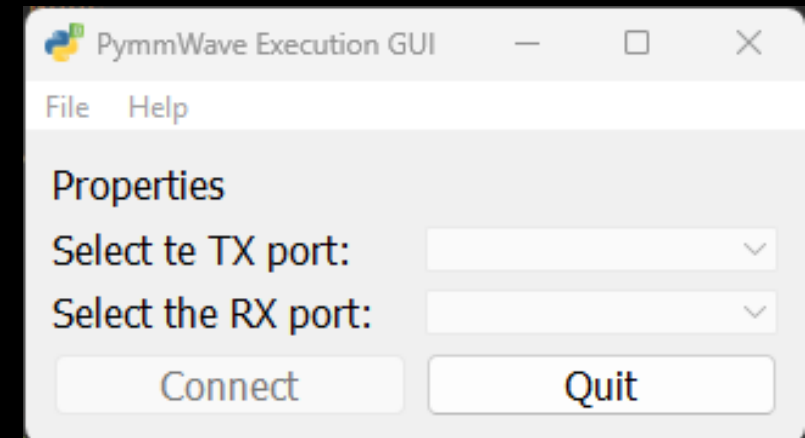
Sprintplanung	Backlog
<ul style="list-style-type: none"> <li>Erstellung der GUI: <ul style="list-style-type: none"> <li>Knöpfe</li> <li>Label</li> <li>Dropdown-Menü</li> <li>Titel</li> <li>Tabs</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>GUI-Funktionalität: <ul style="list-style-type: none"> <li>Knöpfe</li> <li>Checkbutton</li> <li>Dropdown-Menü</li> <li>Tabs</li> </ul> </li> <li>Funktionalität mit Pythonic mmWave Toolbox: <ul style="list-style-type: none"> <li>GUI interagiert mit der Toolbox.</li> </ul> </li> <li>Funktionalität mit ROS: <ul style="list-style-type: none"> <li>Verbindung mit ROS herstellen.</li> </ul> </li> </ul>

# Ergebnisse und Ziele der einzelnen Sprints – Sprint 3

Implementierung der Funktionalität der GUI.

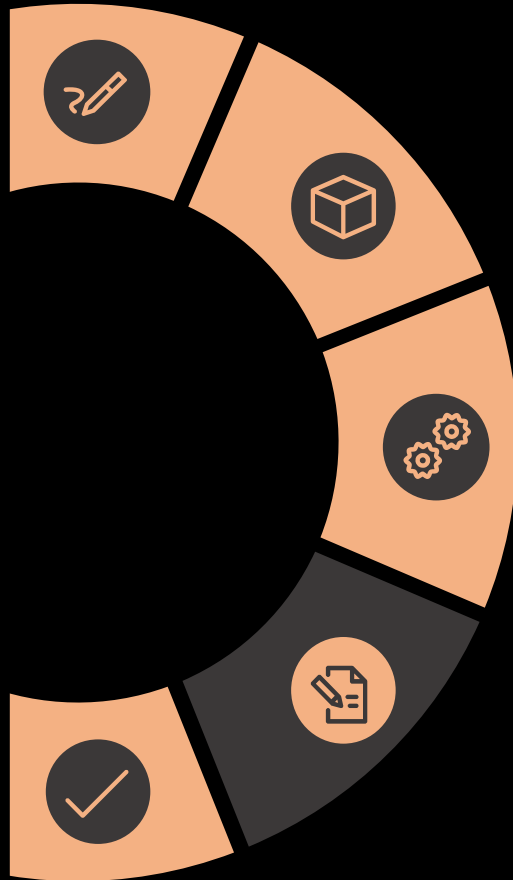


Sprintplanung	Backlog
<ul style="list-style-type: none"> <li>GUI-Funktionalität implementieren: <ul style="list-style-type: none"> <li>Knöpfe</li> <li>Checkbox</li> <li>Dropdown-Menü</li> <li>Tabs</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Funktionalität mit Pythonic mmWave Toolbox: <ul style="list-style-type: none"> <li>GUI interagiert mit der Toolbox.</li> </ul> </li> <li>Funktionalität mit ROS: <ul style="list-style-type: none"> <li>Verbindung mit ROS herstellen.</li> </ul> </li> </ul>



# Ergebnisse und Ziele der einzelnen Sprints – Sprint 4

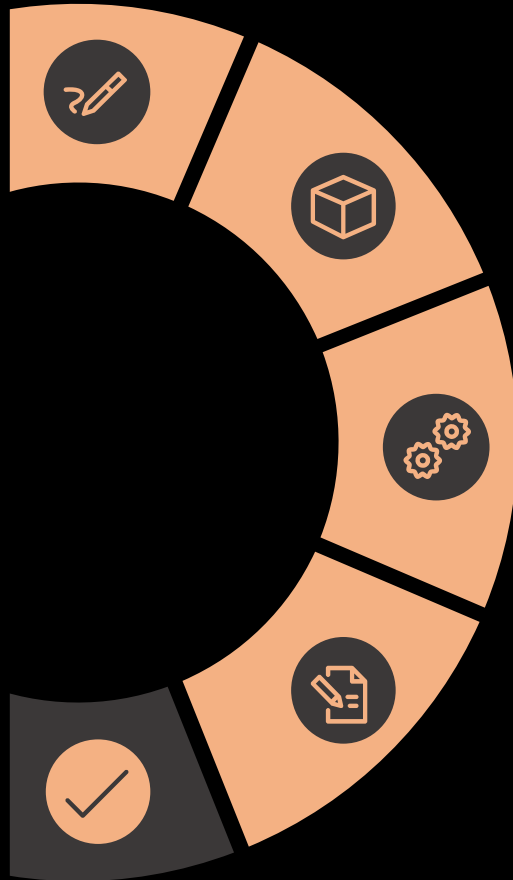
Optimierung durch Implementierung von MVC.



Sprintplanung	Backlog
<ul style="list-style-type: none"> <li>Code in MVC-Muster umschreiben.</li> </ul>	<ul style="list-style-type: none"> <li>Funktionalität mit Pythonic mmWave Toolbox: <ul style="list-style-type: none"> <li>GUI interagiert mit der Toolbox.</li> </ul> </li> </ul>

# Ergebnisse und Ziele der einzelnen Sprints – Sprint 5

Stringentere Umsetzung des MVC-Musters sowie Implementierung der Interaktion mit der Pythonic mmWave Toolbox.



Sprintplanung	Backlog
<ul style="list-style-type: none"> <li>• Stringenter in MVC umschreiben.</li> <li>• Funktionalität mit Pythonic mmWave Toolbox: <ul style="list-style-type: none"> <li>• GUI interagiert mit der Toolbox.</li> </ul> </li> </ul>	

# Agenda

Zu Beginn wird die Aufgabenstellung beschrieben, um anschließend auf die genutzten Muster einzugehen. Daraufhin werden die einzelnen Sprintergebnisse vorgetragen, damit abschließend das Resultat live demonstriert werden kann.

- Einleitung
  - Problembeschreibung
  - Genutzte Muster
- Sprint
  - Ergebnisse und Ziele der einzelnen Sprints
- Ende
  - Live Demonstration



# Live Demonstration

In diesem Abschnitt wird die Präsentation kurz beendet, da die graphische Benutzeroberfläche live präsentiert und der Funktionsumfang kurz erläutert wird.

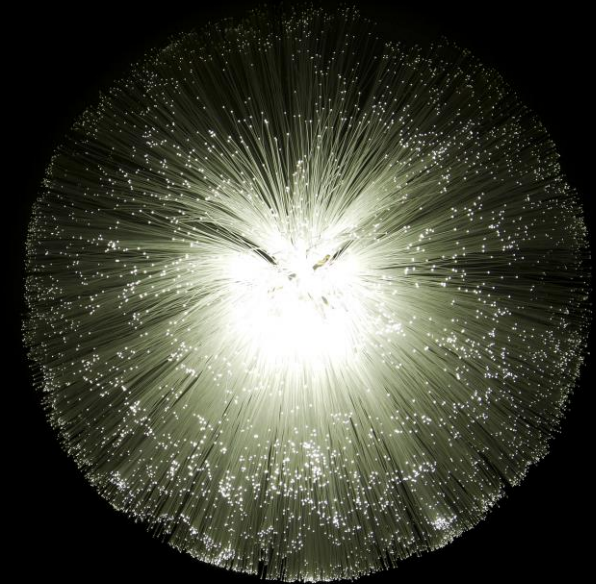
0%

# Quellen

Die genutzten Quellen auf einem Blick.

[1] J. M. Willman, *Beginning PyQt: A Hands-on Approach to GUI Programming with PyQt6*, 2nd ed. Berkeley, CA: Apress; Imprint Apress, 2022.

[2] Wikipedia, *Model View Controller*, Zugriff: 29. Januar 2023. [Online]. Verfügbar: [ModelViewControllerDiagram2 - Model View Controller – Wikipedia](#)





Vielen Dank für eure Zeit und Aufmerksamkeit!