

HERZLICH WILLKOMMEN

PROJEKTPHASE I CAMP2CODE



Batterien laden!

**Beginnen Sie, falls noch nicht geschehen, mit
dem Laden der Akkus!**

Überblick Projektphase

- *In jeder Gruppe/Team soll jeweils gemeinsam eine Software für das Auto erstellt werden.*
 - Ausführung verschiedener Fahrmodi nach Beschreibung im Lastenhaft.
 - Finales Ziel ist es das das Auto einer Linie auf dem Boden folgen kann.
 - Aufzeichnung und Darstellung der Fahrdaten.
 - Der Quellcode soll angemessen dokumentiert werden/sein.
 - Anwenderfreundliches Nutzerinterface mit kurzer Dokumentation.
- *Finale Präsentation* der Ergebnisse je Gruppe.
- *Regelmäßige tägliche Treffen* aller Teilnehmer (Progress-Meeting)
- *Die Teams arbeiten in eigenen Besprechungsräumen.*
- *Optionale Sourcecodeverwaltung im Team mit Git und GitHub.*

Agenda Projektphase 1 - Woche 1

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
Vormittag	<p>08:30 Uhr</p> <ul style="list-style-type: none"> ▪ Begrüßung / Agenda ▪ Vorstellung des Lastenhefts und der Arbeitsmaterialien (Repository) <p>09:30 Uhr</p> <ul style="list-style-type: none"> • Einrichten des Raspberry Pi • Remoteverbindung • (Installation von Software)* • (Kontrolle des Modellautos)* 	<p>08:30 Uhr</p> <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen <p>09:00 Uhr</p> <ul style="list-style-type: none"> ▪ Aufsetzen einer Versionierungsverwaltung (Git?) 	<p>08:30 Uhr</p> <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen <p>09:00 Uhr</p> <ul style="list-style-type: none"> • Progress-Meeting <p>09:30 Uhr</p> <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben im Lastenheft 	<p>08:30 Uhr</p> <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen <p>09:00 Uhr</p> <ul style="list-style-type: none"> • Progress-Meeting <p>09:30 Uhr</p> <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben im Lastenheft 	<p>08:30 Uhr</p> <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen <p>09:00 Uhr</p> <ul style="list-style-type: none"> • Progress-Meeting <p>09:30 Uhr</p> <ul style="list-style-type: none"> • Einführung Plotly Dash
	12:00 – 13:00 Uhr Mittagspause	12:00 – 13:00 Uhr Mittagspause	12:00 – 13:00 Uhr Mittagspause	12:00 – 13:00 Uhr Mittagspause	12:00 – 13:00 Uhr Mittagspause
Nachmittag	<p>13:00 Uhr</p> <ul style="list-style-type: none"> ▪ Rollenverteilung im Team ▪ Projektzieldefinition ▪ Erstellung Projektplan ▪ Test der Basisklassen 	<p>13:00 Uhr</p> <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben im Lastenheft <p>14:00 Uhr</p> <ul style="list-style-type: none"> • Progress-Meeting 	<p>13:00 Uhr</p> <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben im Lastenheft <p>14:00 Uhr</p> <ul style="list-style-type: none"> ▪ Progress-Meeting 	<p>13:00 Uhr</p> <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben im Lastenheft <p>14:00 Uhr</p> <ul style="list-style-type: none"> • Progress-Meeting 	<p>13:00 Uhr</p> <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben im Lastenheft <p>14:00 Uhr</p> <ul style="list-style-type: none"> ▪ Zusammenfassung der Woche ▪ Feedbackrunde

* ... bereits im Vorfeld geschehen

Agenda Projektphase 1 - Woche 2

	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
Vormittag	<p>08:30 Uhr <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen </p> <p>09:00 Uhr <ul style="list-style-type: none"> ▪ Stand-up-Meeting </p> <p>09:30 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p>	<p>08:30 Uhr <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen </p> <p>09:00 Uhr <ul style="list-style-type: none"> • Progress-Meeting </p> <p>09:30 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p>	<p>08:30 Uhr <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen </p> <p>09:00 Uhr <ul style="list-style-type: none"> • Progress-Meeting </p> <p>09:30 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p>	<p>08:30 Uhr <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen </p> <p>09:00 Uhr <ul style="list-style-type: none"> • Progress-Meeting </p> <p>09:30 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p>	<p>08:30 Uhr <ul style="list-style-type: none"> ▪ Stand-up-Meeting der Gruppen </p> <p>09:00 Uhr <ul style="list-style-type: none"> ▪ Präsentation der Ergebnisse der Gruppen ▪ Feedbackrunde </p>
	12:00 – 13:00 Uhr Mittagspause	12:00 – 13:00 Uhr Mittagspause			
Nachmittag	<p>13:00 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p> <p>14:00 Uhr <ul style="list-style-type: none"> ▪ Zusammenfassung </p>	<p>13:00 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p> <p>14:00 Uhr <ul style="list-style-type: none"> • Progress-Meeting </p>	<p>13:00 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p> <p>14:00 Uhr <ul style="list-style-type: none"> • Progress-Meeting </p>	<p>13:00 Uhr <ul style="list-style-type: none"> ▪ Eigenständige Bearbeitung der Aufgaben </p> <p>14:00 Uhr <ul style="list-style-type: none"> • Progress-Meeting </p>	<p>13:00 Uhr <ul style="list-style-type: none"> ▪ Zeit für weitere Fragen </p>

Präsentation der Ergebnisse (Freitag Woche 2)

Ein paar Anhaltspunkte für den Inhalt:

Eventuell eine kurze Vorstellung des Teams

Funktionsdemo / Produktvorstellung

Das ist das fertige Produkt! Eventuell ein kurzes Video und ein Demo der Software.

Special Feature

Falls Euch ein bestimmtes Features besonders am Herzen liegt!

Grundidee für Steuerung und Aufzeichnung der Daten

Wie werden Steueranweisungen organisiert?

Wann wird warum geloggt?

Strukturierung des Quellcodes

z.B. Unterteilung in Funktionen/Klassen

Was passiert beginnend mit dem Programmstart?

Dokumentation

Kurzer Verweis auf die Dokumentation

Zeitlicher Umfang

maximal 30 min inklusive Fragen

Angestrebter Zeitplan

Gruppe	Uhrzeit
1	9:00 Uhr
2	9:30 Uhr
3	10:00 Uhr
4	10:30 Uhr

Erste Schritte in der Projektphase 1

- 1) **Vorstellen des Repositorys:** https://gitlab.com/u4i/camp2code/-/tree/project_phase_1
 - 1) Anleitung_Projektphase1.pdf
 - 2) Anforderungen_Projektphase1.pdf (basisklassen.py, config.json, greet.py, start_after_boot.py)
- 2) **Vorstellung der Anforderungen an die Software** (Anforderungen_Projektphase1.pdf)
- 3) **Finden einer individuellen Arbeitsumgebung (IDE?, Remoteverbindung?)** (Anleitung_Projektphase_1.pdf)
- 4) **Download des Repositorys auf den RPi**
- 5) **Softwareinstallation und Kontrolle des Modellautos*** (Anleitung_Projektphase_1.pdf)
- 6) **Projektplanung in den Gruppen**
 - 1) Besprechen des Lastenhefts
 - 2) Eventuelle Aufgabenteilung in den Teams
 - 3) Eventuelle Verwendung eines Kanbanboard
 - 4) Versionierungsverwaltung mit Git? Wie soll Quellecode im Team ausgetauscht werden?
- 7) **Testen der Basisklassen** (basisklassen.py)

* ... bereits im Vorfeld geschehen

Git und GitHub in den Projektphasen

Git ist ein weit verbreitetes Werkzeug zur Versionsverwaltung. Es erlaubt einzelne Änderungen am Quellcode zu verwalten.

GitHub ist ein Onlinedienst für Softwareprojekte, welche ein zentrales Git-Repository zur Verfügung stellt.

Für die gemeinsame Softwareentwicklung im Team muss Sourcecode zwischen den Teammitgliedern ausgetauscht werden!

- Git löst diese Problem durch das zentrale Repository auf welches jedes Teammitglied zugreifen kann.
- Notwendigkeit eines GitHub-Accounts mit privater E-Mail-Adresse!
- Einführung in Git und Github steht zur Verfügung. (Einführung_Git_und_Github.pdf)
- Ziel ist es nicht Git grundlegend zu verstehen, sondern nur es im Rahmen der Projektphase einzusetzen.
- Alternativ muss der Sourcecode auf andere Weise ausgetauscht werden!

„Hilfe! Ich kann mir nicht immer alles merken“

- **Cheat Sheets** (Spickzettel) fassen häufige Kommandos etc. zusammen!
 - Python: <https://www.pythontcheatsheet.org/>
 - Pandas: <https://www.dataquest.io/blog/pandas-cheat-sheet/>
 - Git: <https://training.github.com/downloads/de/github-git-cheat-sheet/>
 - Linux: <https://cheatography.com/davechild/cheat-sheets/linux-command-line/>
- **Nachschlagen im Web** (Beispiele)
 - Python:
 - Offizielle Docs: <https://docs.python.org/3/reference/>
 - „Python3: Das umfassende Handbuch“ <https://openbook.rheinwerk-verlag.de/python/>
 - W3School: <https://www.w3schools.com/python/default.asp>
 - Pandas:
 - Offizielle Docs: https://pandas.pydata.org/docs/user_guide/index.html
 - Git:
 - Offizielle Seite: <https://git-scm.com>

Agile Softwareentwicklung

„**Agile Softwareentwicklung** (von lat. *agilis* „flink, beweglich“) bezeichnet *Ansätze im Softwareentwicklungs-prozess*, die die Transparenz und Veränderungsgeschwindigkeit erhöhen und zu einem schnelleren Einsatz des entwickelten Systems führen sollen, um so Risiken und Fehlentwicklungen im Entwicklungsprozess zu minimieren. Dazu wird versucht, die Entwurfsphase auf ein Mindestmaß zu reduzieren und im Entwicklungsprozess so früh wie möglich zu ausführbarer Software zu gelangen. Diese wird in regelmäßigen, kurzen Abständen mit dem Kunden abgestimmt. So soll es möglich sein, flexibel auf Kundenwünsche einzugehen, um so die Kundenzufriedenheit insgesamt zu erhöhen.“ (Wikipedia)

- Kleine selbstorganisierte Teams
- Schlanke und flexible Planung
- Iterative und inkrementelle Vorgehensweise
- Enge Kommunikation im Team und mit dem Kunden
- Endprodukt ist oft noch nicht klar definiert

Gegenentwurf zur klassischen Softwareentwicklung mit weitreichender Planung, hierarchischer Organisation und klar definiertem Endprodukt.

Agile Frameworks – Scrum - Begrifflichkeiten

Bekannte Frameworks: Scrum, Kanban, Scrumban, etc...

SCRUM (Modell des Projekt- und Produktmanagements):

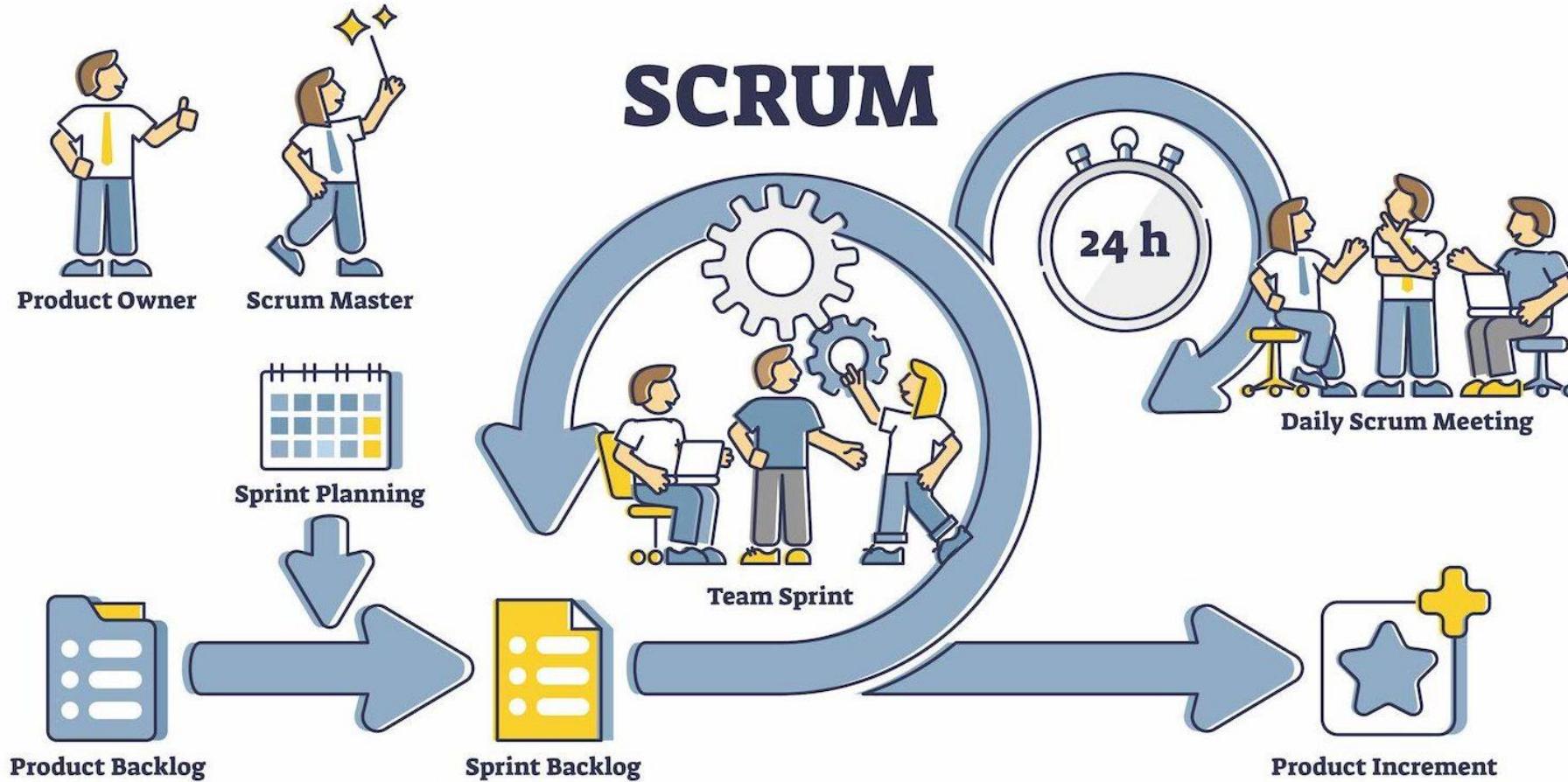
- **Scrum Verantwortlichkeiten/Rollen:**
 - Product Owner - Was? Wann? (Produktgestaltung, Kommunikation mit Stakeholdern, pflegt, erklärt und priorisiert Anforderungen im Product Backlog)
 - Developer - Wie? Oftmals interdisziplinäres Team, dass die Anforderung ohne äußere Hilfe erledigen kann
 - Scrum Master - Betreut Scrum-Prozesse, Coacht Team, behebt Hindernisse und Konflikte im Team
- **Scrum Events**
 - Sprint Planning - Was wird Wie im nächsten Sprint als Product Increment umgesetzt? Sprint Backlog formuliert Plan für Sprint
 - Sprints - Umsetzung des Sprint Plans
 - Daily Scrum - Kurze täglich Meetings mit dem Ziel des Informationsaustauschs
 - Sprint Review - Präsentation und Überprüfung des Product Increments
 - Sprint Retrospective – Review des Sprintablaufs an des Ende, Was kann in der Zusammenarbeit verbessert werden?

Product Backlog Liste der Anforderungen an das Produkt (anwenderorientiert, unabhängig, klein, testbar,...)

Sprint Backlog Plan für den Sprint mit Einträgen aus dem Backlog und dazugehörigen Aufgaben

Product Increment Summe aller Einträge aus dem Product Backlog welche im Sprint umgesetzt wurden

Agile Frameworks – Scrum - Begrifflichkeiten



Wichtige Anmerkung!

Wir arbeiten in Gruppen von Teilnehmern mit unterschiedlichen Vorkenntnissen und unterschiedlichen Zeitbudget für dieses Training!

- Es kann daher vorkommen, dass sich Teilnehmer „abgehängt“ füllen. Kommuniziert dies bitte klar an die Trainer. Es wird genügend Zeit für individuelle Anleitungen und Erklärungen geben. Dafür sind die Trainer u.a. da!
- Die agile Arbeitsweise muss nicht strikt um gesetzt werden! Im Besonderen müssen Teilnehmer mit wenig Programmiererfahrung Arbeitspakete nicht zwingen aufzuteilen. In diesem Fall kann es lehrreicher sein einzelnen Schritte gemeinsam abzuarbeiten.