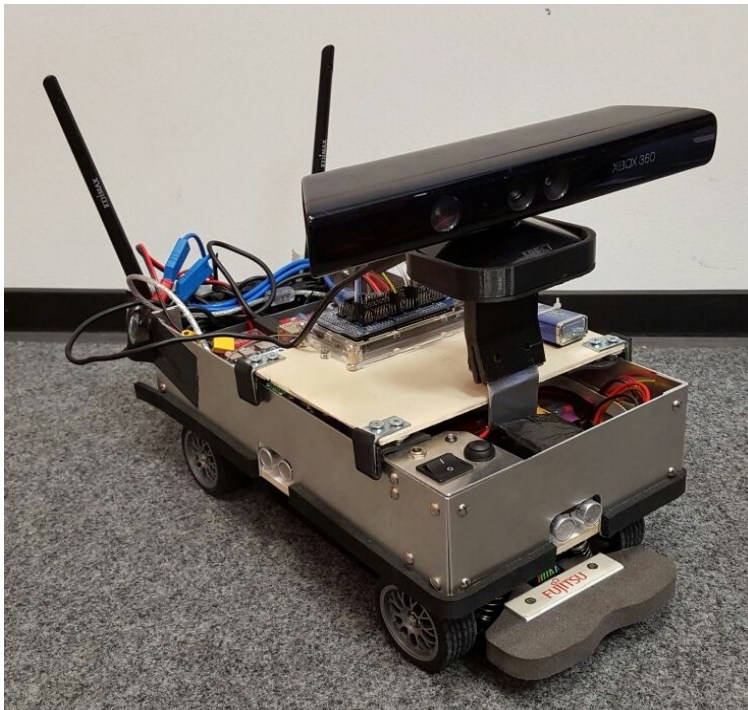


# Endvortrag Echtzeitsystem Projektseminar



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Team Apollo 13: Sebastian Ehmes, Nicolas Acero,  
Huyhn-Tan Truong, Li Zhao



## I. Recap:

- Mission 1 & Mission 2

## II. Mission 2:

- Selbstlokalisierung
- Mapping
- Wegplanung
- „sbpl“ - Lösung der Wegplanung

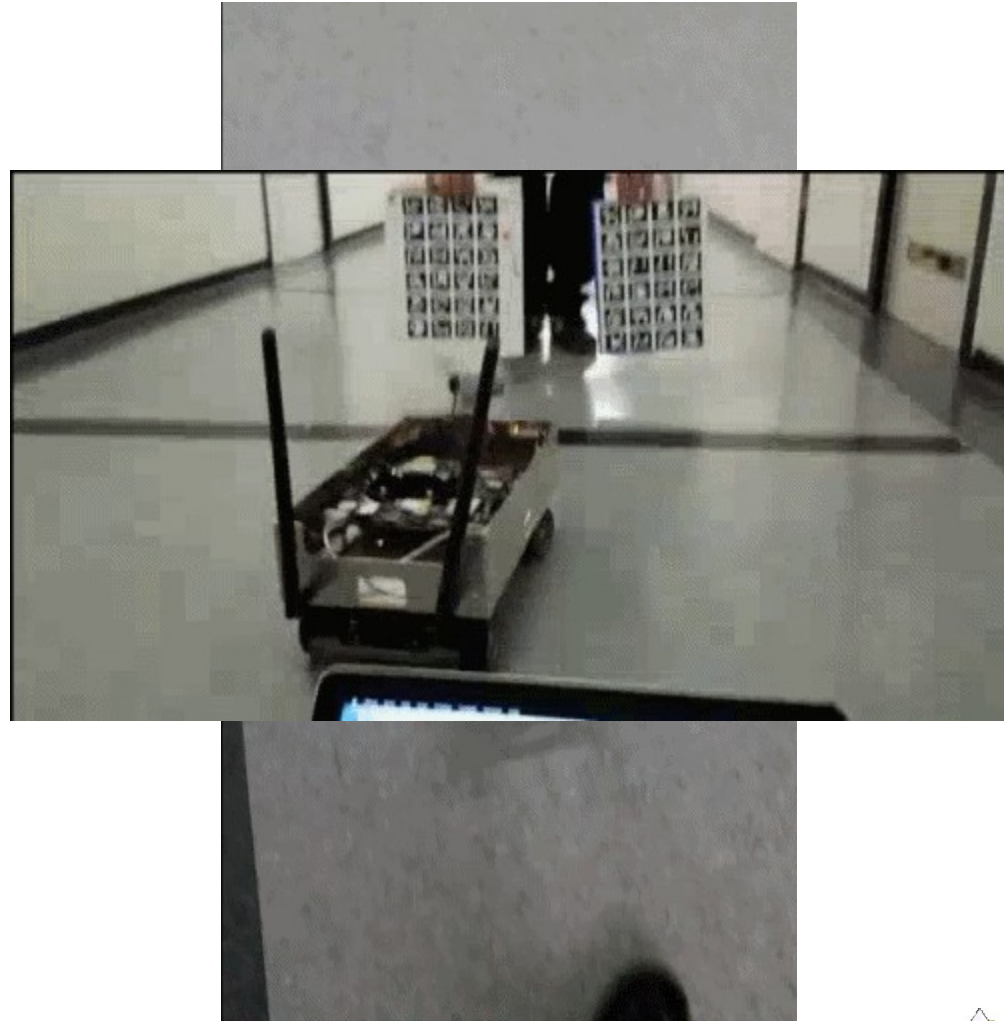
## III. Bonus Mission

## IV. Fazit & Ausblick

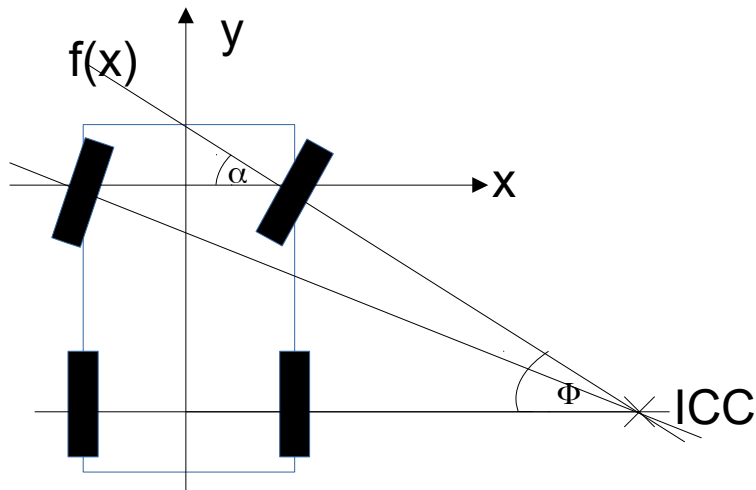
# I: Recap: Mission 1 - Regelung



- Intuitiv nahliegendste Überlegung:
  - Radstellung in Abhängigkeit des Abstandes zur Wand einstellen durch einen PD-Regler mit empirisch bestimmten Werten
- „Aruco-Tore“ durchfahren mit Hilfe des ROS-Packages `ar_sys`



# I. Recap: Mission 2 – Odometrie & Vorw. Kinematik



- **Odometrie:** Berechnung v. Position und Orientierung eines Fahrzeuges allein durch Messung der Radbewegung mit internen Sensoren und der kinematischen Vorwärtslösung
- **Vorw. Kinematik:** Bestimmung der eigenen räumlichen Anordnung des Fahrzeug aus den steuerbaren Größen (hier Geschwindigkeit und Radwinkel)
- **ICC** (instantaneous center of curvature) = momentane Drehzentrum
- $\Phi$  = Drehwinkel um den ICC/ Orientierung bezüglich Ursprungs

$$(I): \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} R \cdot \cos(\Phi + \alpha) - \frac{k}{\tan(\alpha)} \\ R \cdot \sin(\Phi + \alpha) - k \end{pmatrix}, \quad (II): \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} R \cdot \cos(\pi - \Phi + \alpha) - \frac{k}{\tan(\alpha)} \\ R \cdot \sin(\pi - \Phi + \alpha) - k \end{pmatrix}$$

I. Fall: rechts Kurve

II. Fall: links Kurve

$${}^0T_i = \begin{pmatrix} \cos(\Theta) & -\sin(\Theta) & 0 & x \\ \sin(\Theta) & \cos(\Theta) & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{mit } \Theta = \begin{cases} -\Phi, & \text{wenn } \alpha < 0 \\ \Phi, & \text{wenn } \alpha > 0 \end{cases}$$

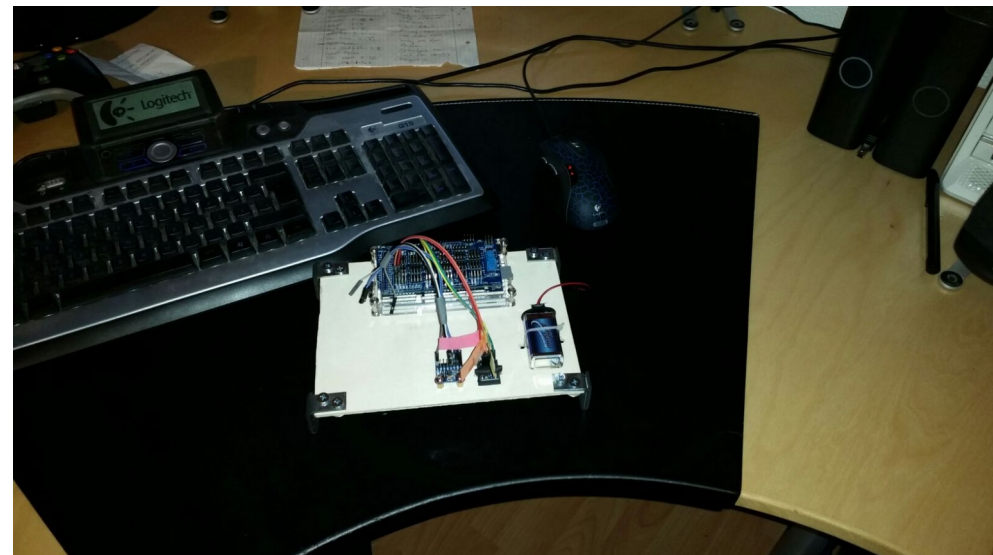
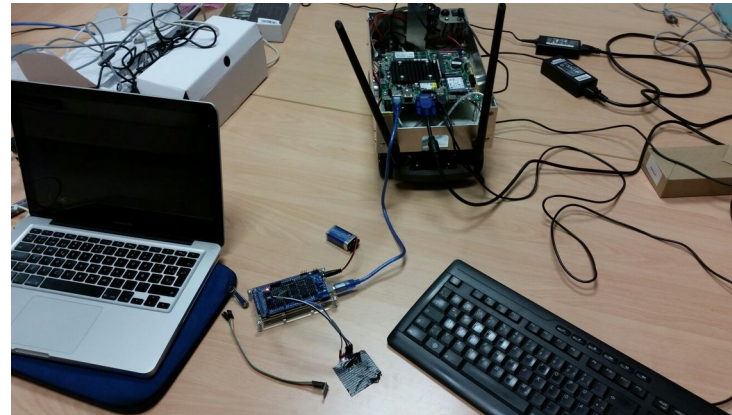
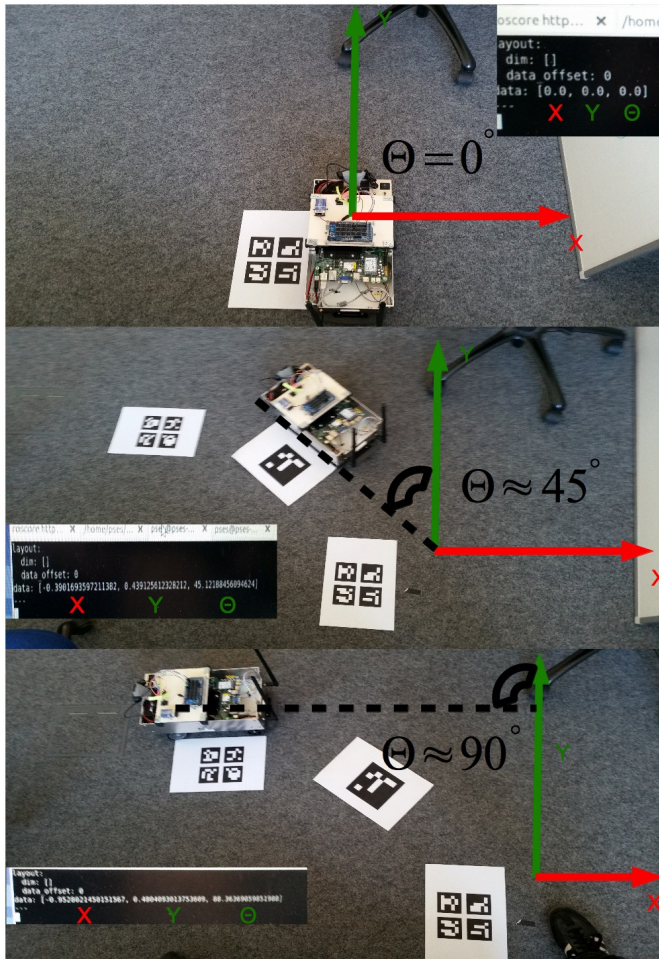
Transformationsmatrix  
des Autos



# I. Recap: Mission 2 – Odometrie & Vorw. Kinematik



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT



## II. Mission 2 - Selbstlokalisierung



- Selbstlokalisierung durch Odometrie und Vorw. Kinematik
  - Herzstück für die Wegplanung
- Angewiesen auf viele und gute Daten für gefahrenen Weg und Orientierung
  - Hallsensor liefert zu wenige Daten (<1Hz Publish)
- Verbesserung durch Drehgeber
  - 45Hz Daten, Visualisierung funktioniert in Echtzeit
  
- Verbesserung der klassischen Odometrie (Positionsbestimmung anhand von Stellgrößen) durch Inertialsensorik
  - Lenkstellgrößen bilden nicht die Wirklichkeit ab
  - Fehler der klassischen Odometrie wächst schnell mit gefahrener Strecke



## II. Mission 2 - Mapping

- Ziel: Punkt in der Welt angeben  
→ Auto fährt den Punkt an ohne durch Hindernisse zu fahren
- Karte von der Umgebung wird benötigt  
→ Mapping durch gegebene Ultraschallsensoren und Kamera nicht möglich
- Einbau der Kinect als Ersatz für Kamera und Ultraschallsensoren  
→ Darstellung der Welt als Pointcloud(Punktwolke)



## II. Mission 2 - Mapping



- Mapping mit Hilfe der ROS-Package „gMapping“ „Slam\_gmapping“ , „depthimage\_to\_laserscan“  
→ benötigt Odometrie und Informationen von einem Laser
- Kinect liefert jedoch eine Pointcloud  
→ „depthimage\_to\_laserscan“:  
Benutzung von Punkten d. Punktwolken nur einer bestimmten Höhe und liefern diese als ein Laser\_Topic zurück
- Mapping durch manuelles Durchfahren des Flures in einer Loop





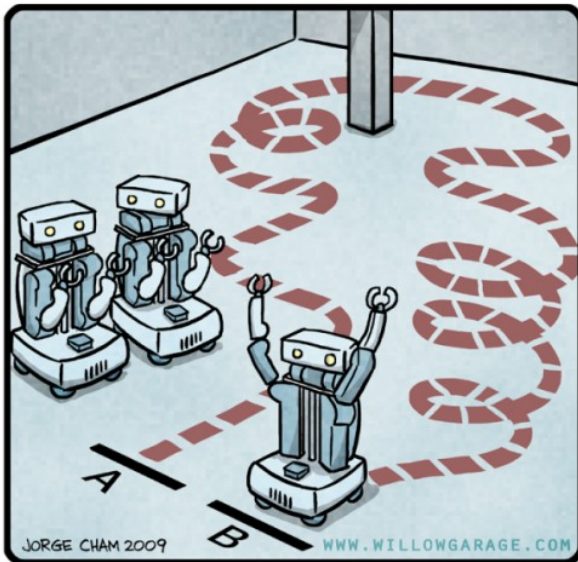


# [Video 1]



## II. Mission 2 - Wegplanung

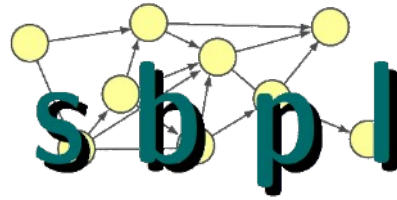
R.O.B.O.T. Comics



"HIS PATH-PLANNING MAY BE  
SUB-OPTIMAL, BUT IT'S GOT FLAIR."

- Unser Ansatz: Umgebung in eine Art Graph darstellen
  - Gitter auf die Map und Kreuzungen werden Knoten
  - Wegplanung durch zB. Dijkstra-Algorithmus
- Probleme: Welchen Knoten darf man verbinden damit man nicht durch Objekte fährt, sodass die Kurven fahrbar sind, damit es verienbar mit der Odometrie ist ... .
- Erkenntnis: Wegplanung ist eine aktuelles Forschungsthema

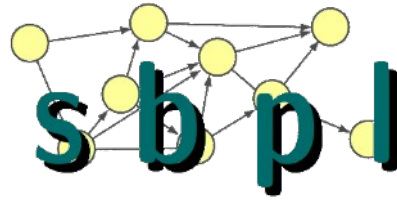
## II. Mission 2 -



- **search-based-planning-laboratory** ist eine Arbeitsgruppe der Universität Carnegie Mellon in Pittsburgh
- „sbpl“ hat ein ROS-Package entwickelt mit einem Planner, der für fast jeden Roboter angepasst werden kann
- Funktioniert wie nach unserer Vorstellung:  
→ Punkt im Raum vorgeben, Punkt anfahren im Raum
- Herausforderung: Integration in unser System

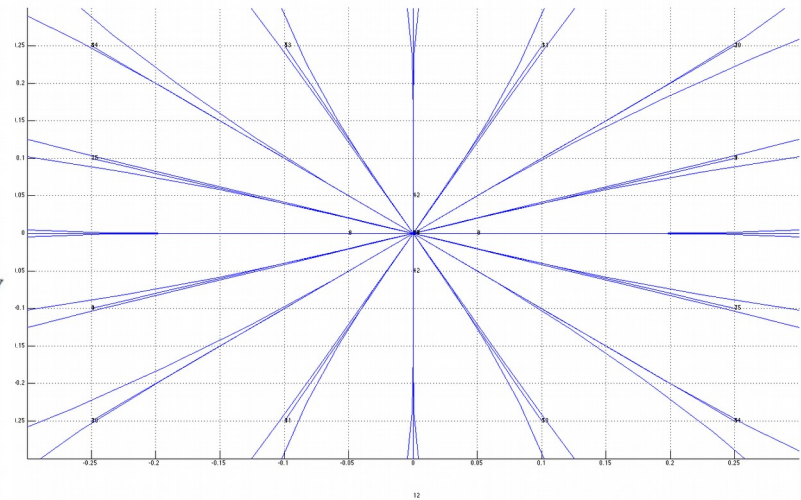
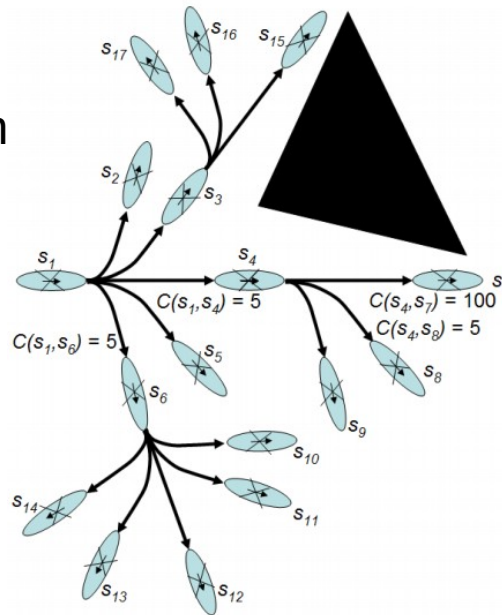


## II. Mission 2 -

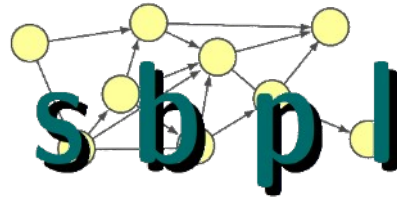


- 1. stellt die Umgebung als Graph dar → Knoten sind diskrete Punkte an der der Roboter sich befinden kann
- 2. Gültigkeit der Knoten → gibt es Kollisionen wenn der Roboter sich am besagten Knoten befindet
- 3. Anwendung von bekannten Graph Algorithmen zb. Dijkstra

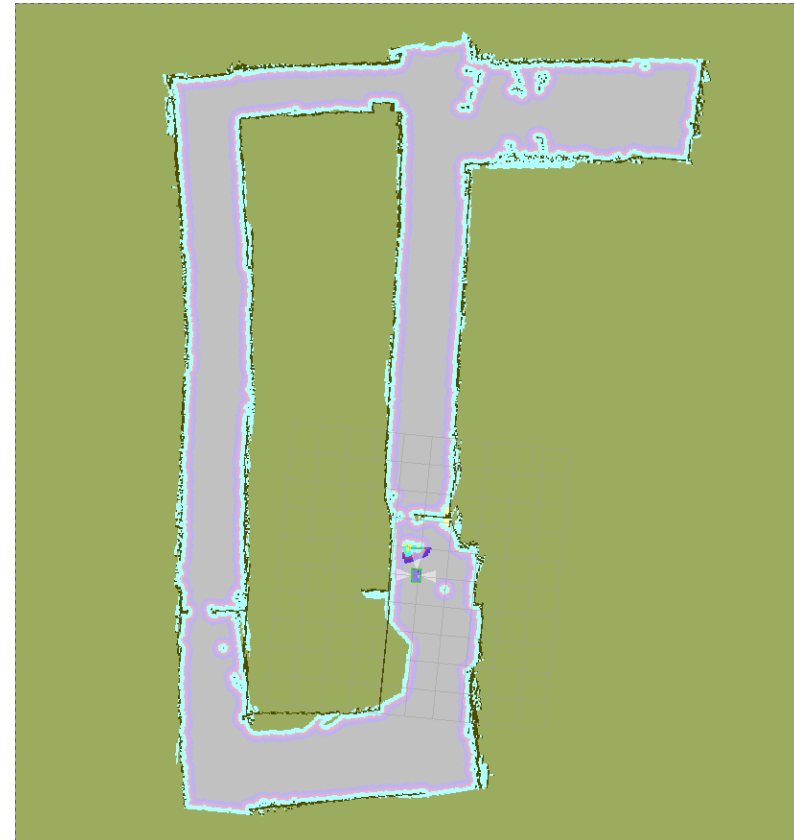
- „sbpl“ benötigt Motion Primitives – verschiedene definierte Grundmanöver mit Information wo Roboter sich danach befindet



## II. Mission 2 -



- Notwendigkeit von einer sehr guten Odometrie , damit Roboter weiß von wo er planen muss und sich während der Fahrt nicht verliert
- Gute Karte von der Umgebung
- AMCL (**A**daptive **M**onte **C**arlo **L**ocalisation)  
→ liefert eine optische Odometrie der unsere bereitgestellte Position zur Laufzeit korrigiert





# [Video 2]



### III. Bonus Mission

- Bis hier bieten wir nur ein Grundgerüst um die verschiedensten Aufgaben zu lösen.
- Da zum Schluss noch Zeit war haben wir uns eine Anwendung überlegt.
- Idee: Anfängliche Aufgabenstellung „Fahre durch Aruco markierte Tore“ noch einmal aufgreifen.
- Vorteil: Diese Aufgabe stellt nun kein Sonderfall mehr da und kann quasi während des „normalen“ Betriebs gelöst werden.



## III. Bonus Mission



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

**[Video 3]**





# III. Fazit & Ausblick

- Aufgabenstellung sehr offen → gut machbar im Zeitraum und ermöglicht einen viele eigene Ideen einzubringen
- Arbeiten an aktuellen Forschungsthemen mit aktuellen Werkzeugen(ROS)
- Sensoren für autonomes Fahren nicht ganz optimal  
→ Radwinkel, Drehgeber, Gyroskop, Kamera mit Tiefeninfo.
- Verbesserung der Motion Primitives
- Autonomes Mapping
- Ziel in der Zukunft: Carolo Cup  
→ Ausbau der Plattform,  
→ weiterentwickeln außerhalb des Projekt Seminars  
→ verschiedene Aufgabenstellung (große Kooperation der Teams)





**Vielen Dank  
für Ihre  
Aufmerksamkeit**

