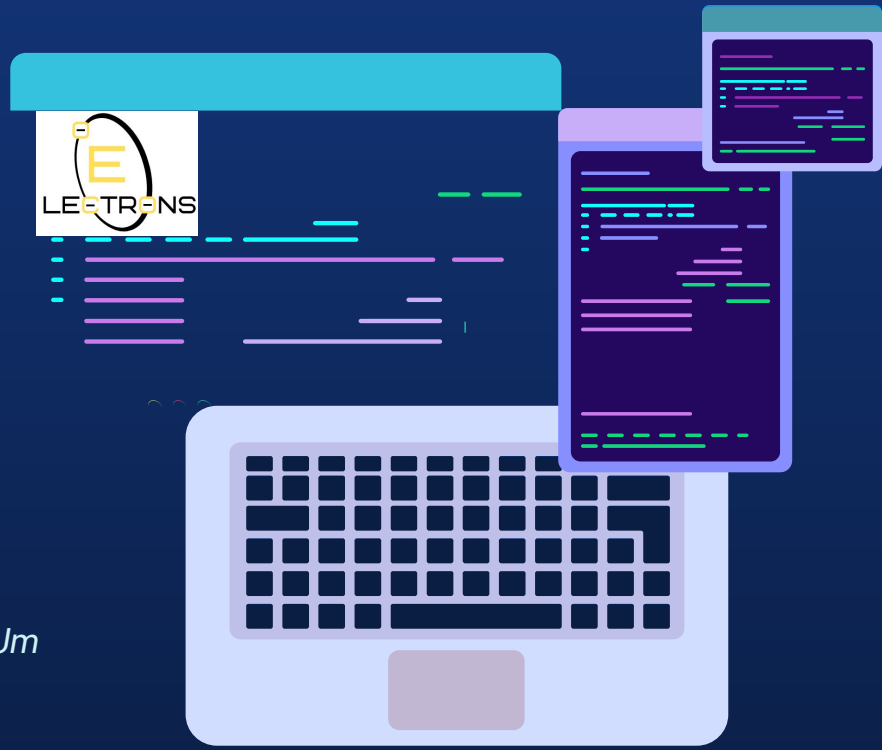
IoT:
Hug The Lanes



Group 8
Justerini Mejia, Justus Neumeister, Philip Russo, TaeSeo Um

01

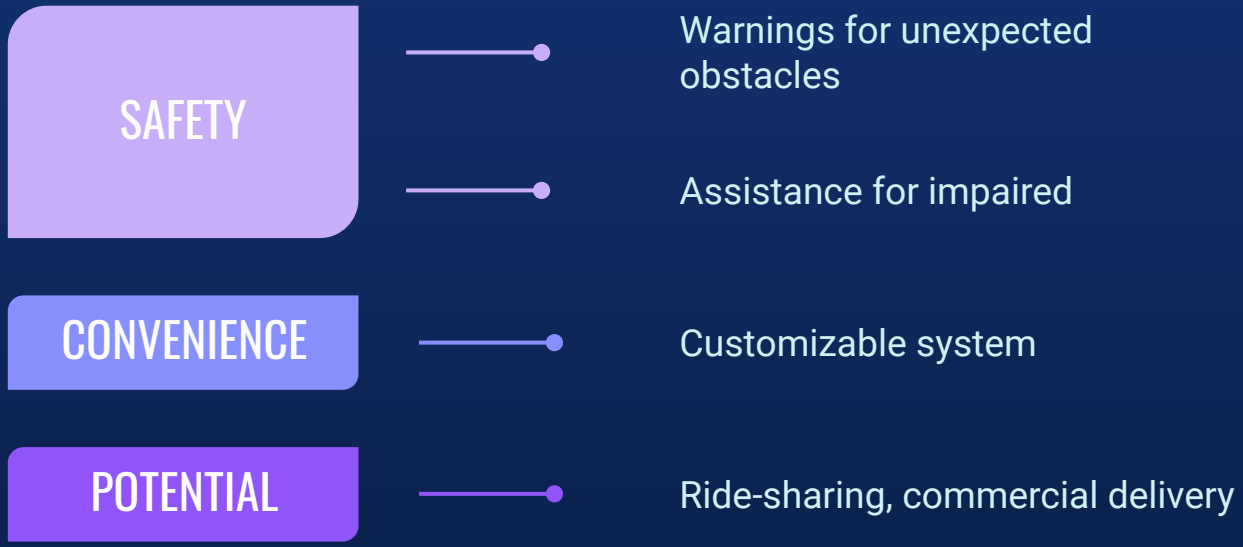
GOAL, USE, VALUE

MISSION STATEMENT

Create innovative self-driving software that:

- Enhances user experience
- Improves accessibility
- Reduces accidents

SOFTWARE USABILITY



VALUE TO THE BUSINESS

Improved Customer Satisfaction

Garner loyalty with reliable software

Access to New Markets

Ability to offer new services and products

Increased Brand Reputation

Perceived as a forward-thinking company

02

CODE DEMONSTRATION

Testing

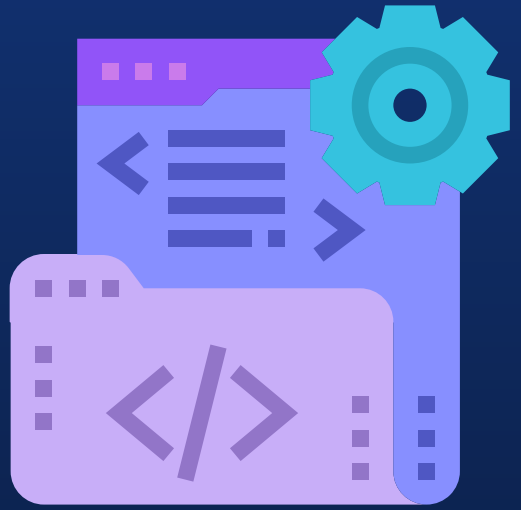
Over the course of multiple days we have tested the effects of data that is:

- Out of the expected range.
- Conflicting with other sensor data.
- Strange.

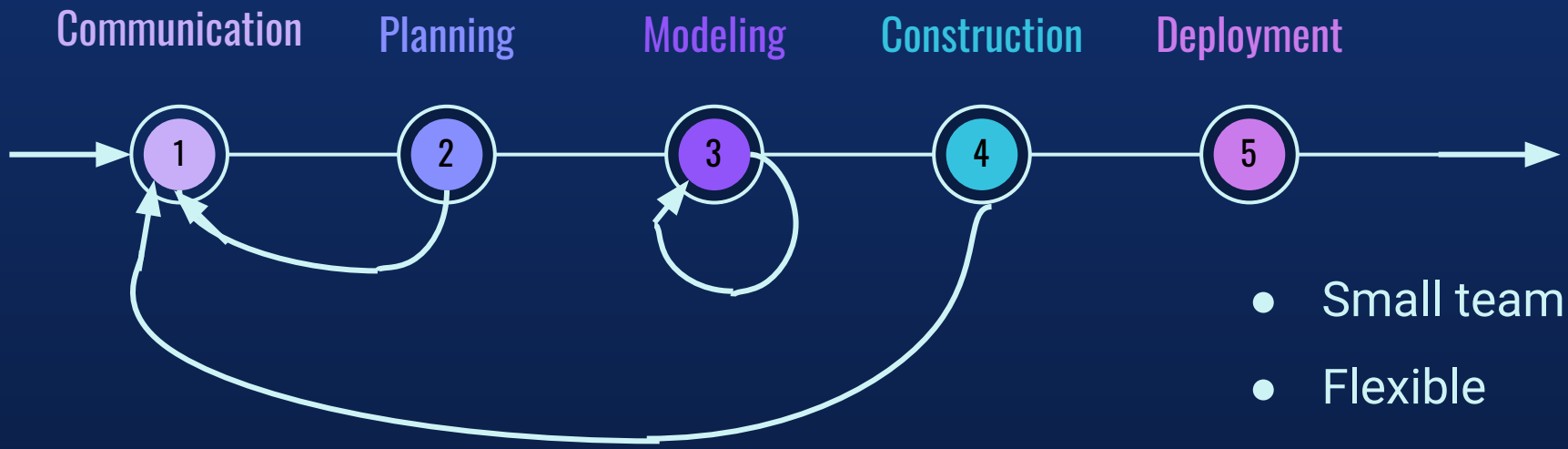
In the event that our system is unable to deal with the current situation, the driver is alerted, decreasing the chances of accident due to over-reliance on our system.

03

SOFTWARE DEVELOPMENT PROCESS



Iterative Development Process



- Small team
- Flexible
- Simple

Object Oriented Architecture

Promotes Modular Design

Easy to test / maintain individual components

Several people working on different parts at once

04

CHALLENGES AND EXCELLENCES

Challenges of this project

- Creating code repository where adding functions would be simple
  - Easy to read input data
  - Sensor Fusion code
- Making sure our code covered all cases that are real-world applicable
  - What type of sensors are needed
  - Within what time should these functions work
  - How do the different modules work together
- Testing the code
  - Making sure all functions work as expected
  - Making sure the raw data input is being used correctly by individual functions

Excellences of this project

- Amazing teamwork and communication
  - As many meeting as possible (in person & remote)
  - Separation of work was easy and smooth
  - Little to no arguments throughout the entire project
- Our current code makes future improvements and adding features easy
  - Just need to add what sensors are needed as input
  - The sensor fusion module code will take care of everything
- Output Log for the technician
  - Shows time
  - Shows each drive session each batch of data is from

Possible Improvements?

Our current code already makes it easy for future functions and features to be added to the system

As a company, we can adapt DevOps strategies, so the development team and operations team can be separated

Generating more data can lead to better testing results

If we were to do this again...

- Code changes
  - Changing the log input from string to tuple -> enables sorting/searching
  - Create an automated data generator file which can feed our code ‘unlimited’ data
  - Refactor some repeated code that might be apparent in the code
  - Different files for each module instead of one file with multiple classes