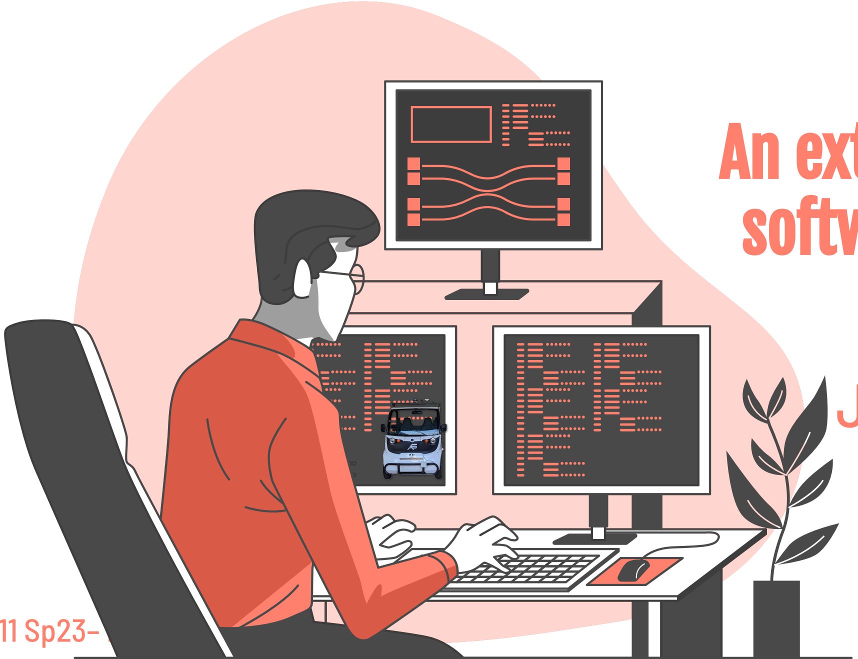


R-IDE

An extension to simplify robotics software development with ROS

CS 411 - Prototype Demo

Josh Peterson, Dominik Soós, Dan Koontz, Gavin St. Clair, Justin Tymkin

Table of Contents

The Team

Josh Peterson

Team Lead, Webmaster,
Frontend UI/UX Specialist

Dominik Soós

Database Specialist

Dan Koontz

Backend/Algorithm Specialist

Gavin St. Clair

Database Specialist

Justin Tymkin

Backend/Algorithm Specialist,
Documentation Specialist

Dr. Lee Belfore

Mentor

Laura Sakos

Mentor

Gavin Alberghini

Mentor

Background

- The Robot Operating System (ROS) is an open source set of software libraries and tools that help developers build robot applications. [1]

- ODU's autonomous vehicles utilize ROS to convert raw sensor data, perform calculations, and make intelligent decisions on the vehicles upcoming route.

- With the combined efforts of the following departments and their respective mentors, ODU hopes to place in the top 6 for the Self-drive challenge at the Intelligent Ground Vehicle Competition (IGVC).
  - Electrical and Computer Engineering
  - Computer Science
  - Mechanical and Aerospace Engineering

Some companies that use ROS-based robotics:

- NASA [4 & 5]
  - Curiosity Rover
  - VIPER program
  - Robonaut 2 [8]
- Microsoft [6]
- SONY [7]

Industries that utilize ROS:

- Autonomous vehicles
- Aerospace
- Health Care
- Agriculture
- Manufacturing
- Military

ROS development contains high barriers of entry for new developers and environments.

ROS workflows leverage several non-attributional windows that elevate the difficulty of debugging, monitoring & understanding

- Each ROS node has at least one terminal window associated with it
- There is no distinction when changes in one node affects a related node
- Simple debugging methodologies become time consuming due to how ROS displays errors

Many parts of the documentation are not up to date and require significant technical knowledge

Environment setup can be difficult in any project, but especially in one that can use multiple versions and distributed dependencies

Our Customers

…

Dr. Belfore

…

Hobbyists

…

Industry

…

Universities

Our Solution

An extension to simplify and speed up the development lifecycle and learning process for ROS developers.

Solution Characteristics

Simplify UI

Create a GUI that simplifies common tasks and commands and accesses visualization tools

Quick Environment Setup

Create a process to quickly build an environment for any ROS project existing or new

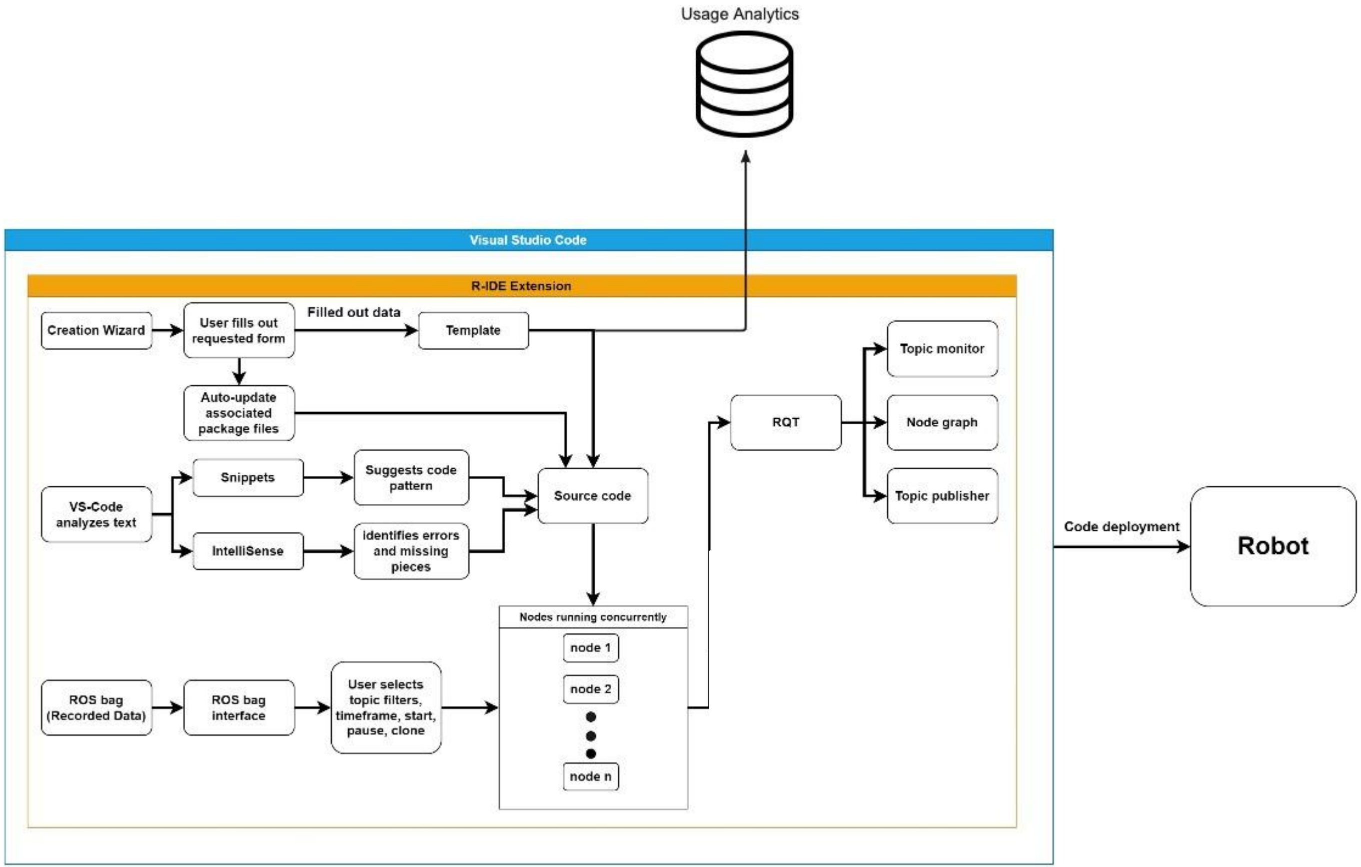
Custom Solutions

Create custom solutions that can analyze and autofill based on preexisting code

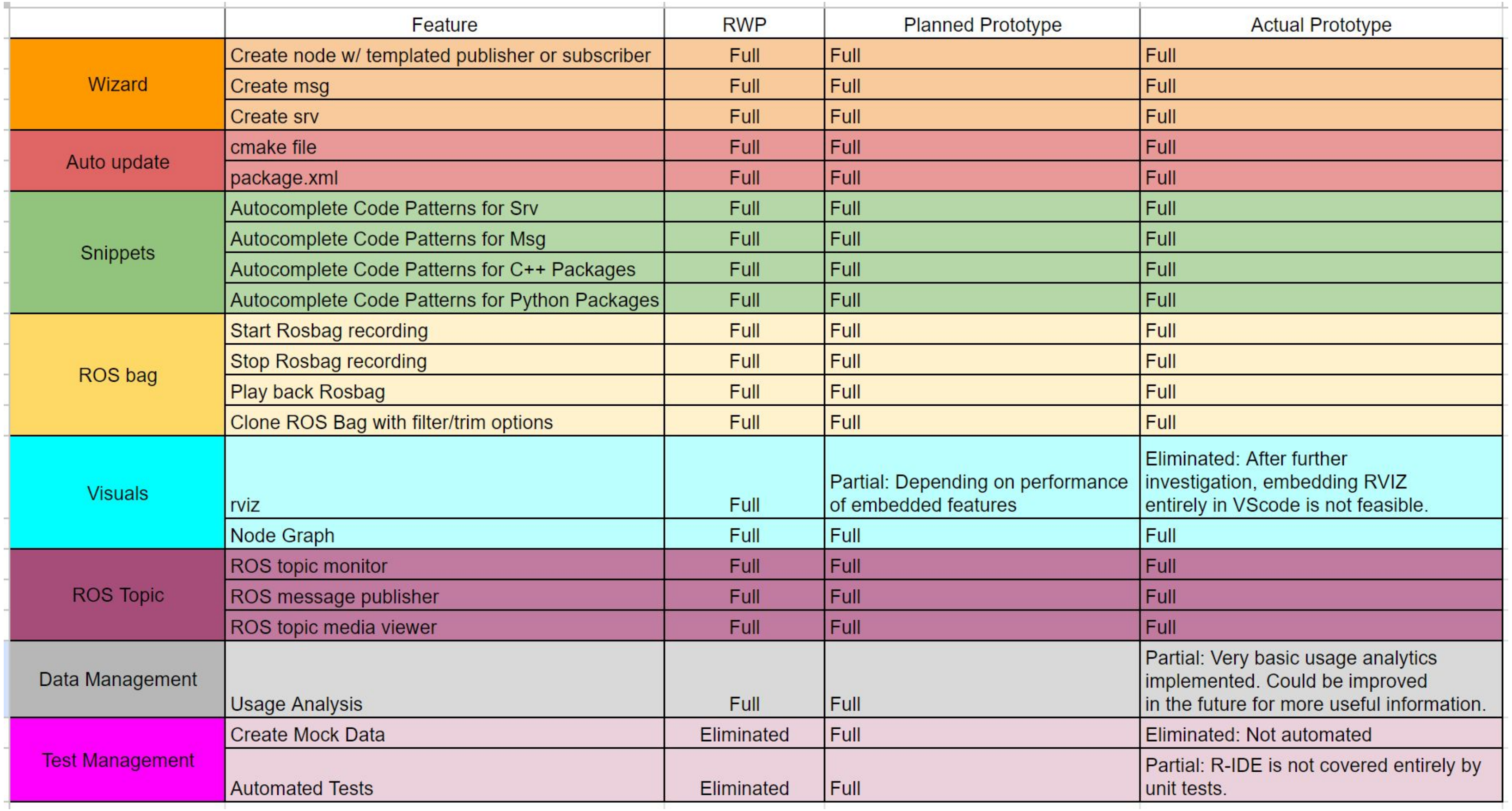
Perspective Shifts

Create a system that allows for perspectives to quickly change and mutate as required

Major Functional Component Diagram



Features - RWP vs Prototype



	Feature	RWP	Planned Prototype	Actual Prototype
Wizard	Create node w/ templated publisher or subscriber	Full	Full	Full
	Create msg	Full	Full	Full
	Create srv	Full	Full	Full
Auto update	cmake file	Full	Full	Full
	package.xml	Full	Full	Full
Snippets	Autocomplete Code Patterns for Srv	Full	Full	Full
	Autocomplete Code Patterns for Msg	Full	Full	Full
	Autocomplete Code Patterns for C++ Packages	Full	Full	Full
	Autocomplete Code Patterns for Python Packages	Full	Full	Full
ROS bag	Start Rosbag recording	Full	Full	Full
	Stop Rosbag recording	Full	Full	Full
	Play back Rosbag	Full	Full	Full
	Clone ROS Bag with filter/trim options	Full	Full	Full
Visuals	rviz	Full	Partial: Depending on performance of embedded features	Eliminated: After further investigation, embedding RVIZ entirely in VScode is not feasible.
	Node Graph	Full	Full	Full
ROS Topic	ROS topic monitor	Full	Full	Full
	ROS message publisher	Full	Full	Full
	ROS topic media viewer	Full	Full	Full
Data Management	Usage Analysis	Full	Full	Partial: Very basic usage analytics implemented. Could be improved in the future for more useful information.
Test Management	Create Mock Data	Eliminated	Full	Eliminated: Not automated
	Automated Tests	Eliminated	Full	Partial: R-IDE is not covered entirely by unit tests.

Challenges We Faced :’(

- ROS
- IDE to Extension - Josh
- Database - Dom
  - MySQL (Docker) → MongoDB

- ROSbag Buffer - Dan
- Unit/Regression Testing - Gavin
- Recreating Visual Tools - Justin

Planned Vs Actual Development Tools

Software Requirements

ROS → ROS 1 Noetic**
~~Rviz~~ → Eliminated
~~OpenGL~~ → Rviz Dependency

Database

~~MySQL~~ → MongoDB

*ROS1 requires some kind of linux environment such as WSL or a virtual machine with linux
** Code templates are known to work in Noetic, but may work in others

Languages

Javascript/Typescript
Html/CSS = Svelte Framework
Python → Template Code
C++ → Template Code
XML → Automated Packages
CMake → Automated Packages
Ros CLI → For ROS library limitations

Developer OS

Linux
Windows*
Mac*

IDE

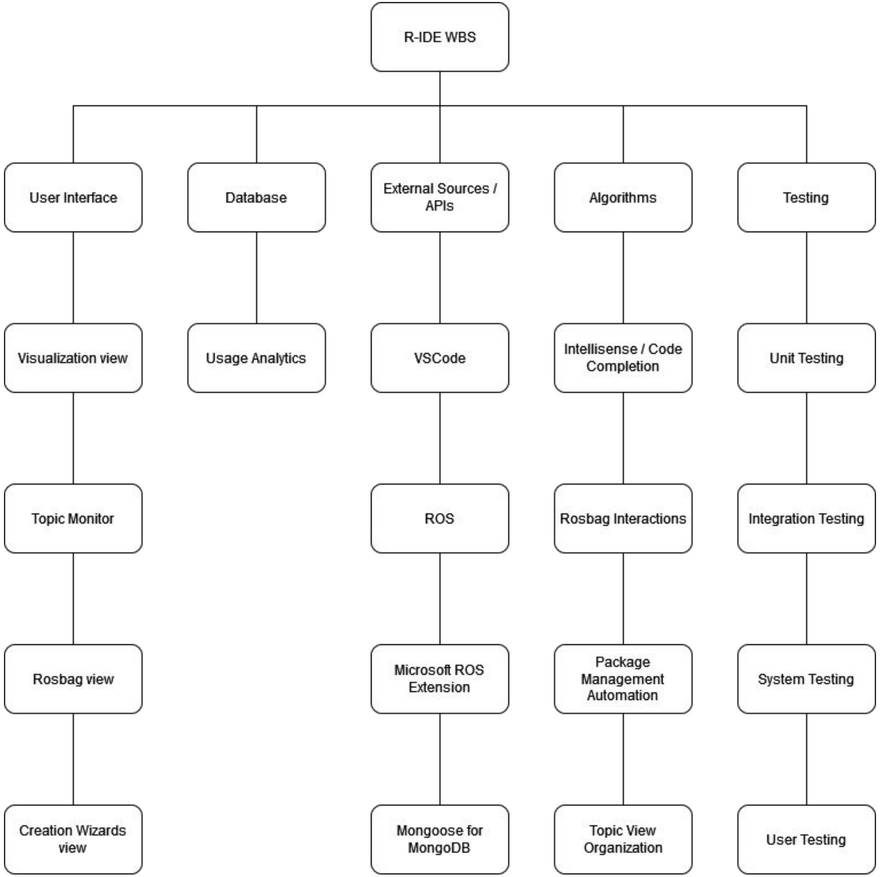
VS-Code
w/ Microsoft ROS
Extension

Hardware

Computer that meets inherited hardware requirements for VS code, ROS, and ~~Rviz~~

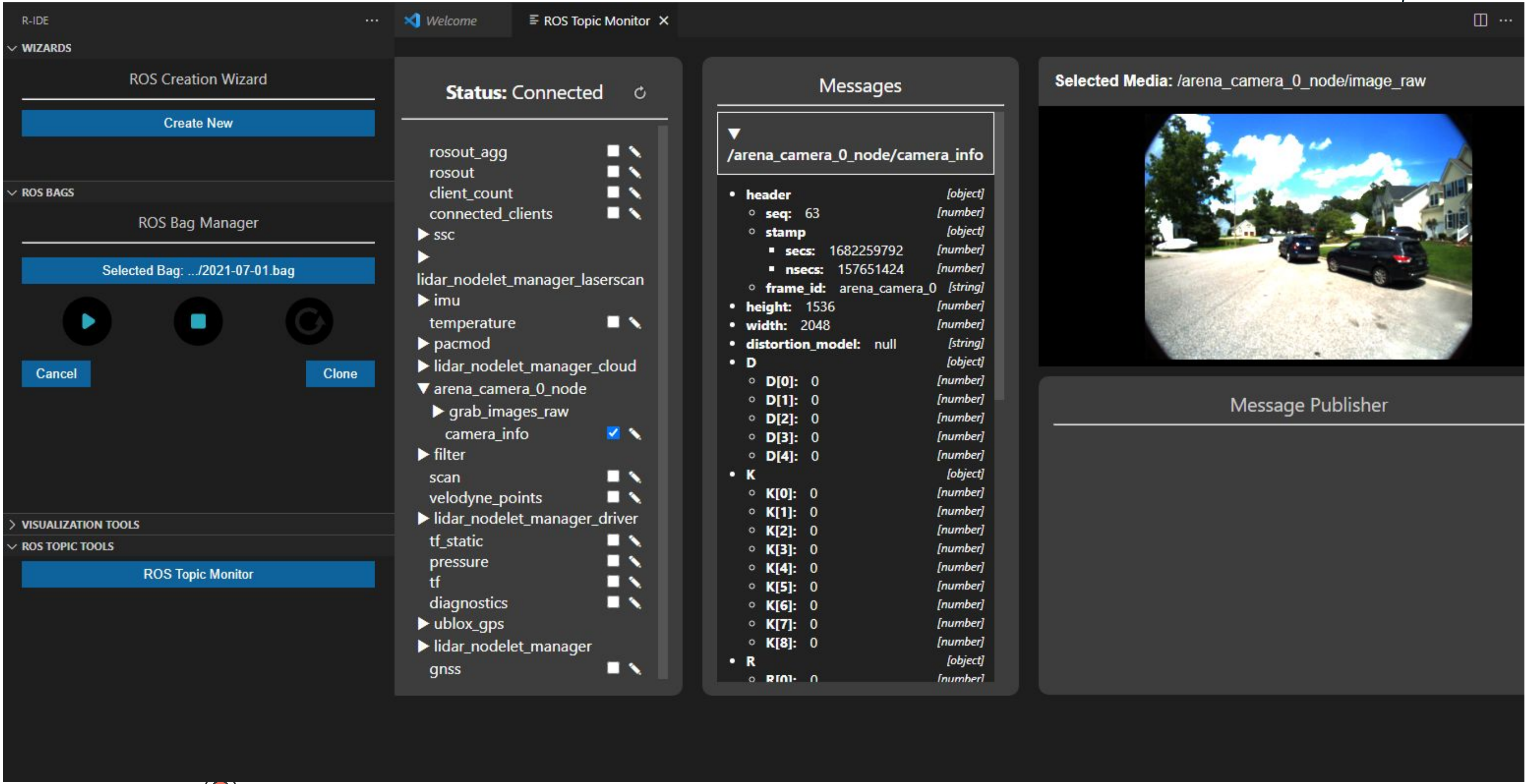
https://code.visualstudio.com/docs/supporting/requirements

Work Breakdown Structure



- R-IDE WBS
  - User Interface
    - Visualization view
    - Topic Monitor
    - Rosbag view
    - Creation Wizards view
  - Database
    - Usage Analytics
  - External Sources / APIs
    - VSCode
    - ROS
    - Microsoft ROS Extension
    - Mongoose for MongoDB
  - Algorithms
    - Intellisense / Code Completion
    - Rosbag Interactions
    - Package Management Automation
    - Topic View Organization
  - Testing
    - Unit Testing
    - Integration Testing
    - System Testing
    - User Testing

UI Screenshots



Status: Connected

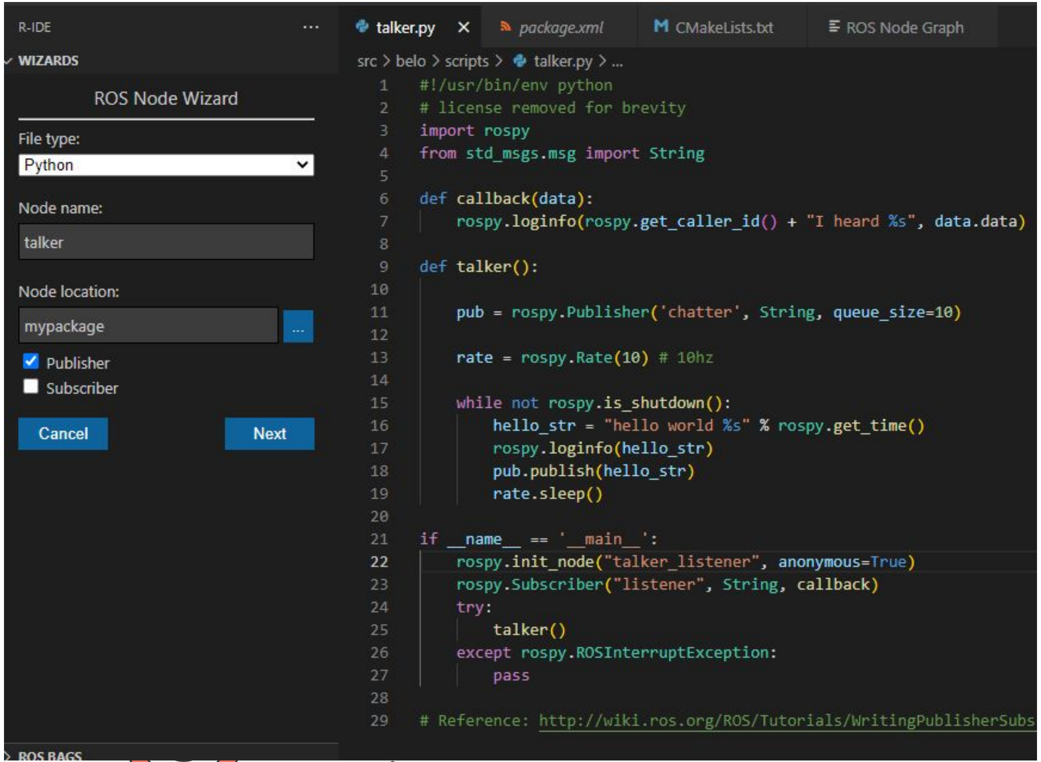
- rosout_agg □
- rosout □
- client_count □
- connected_clients □
- ▶ ssc
- ▶ lidar_nodelet_manager_laserscan
- ▶ imu
- temperature □
- ▶ pacmod
- ▶ lidar_nodelet_manager_cloud
- ▼ arena_camera_0_node
  - ▶ grab_images_raw
  - camera_info ☑
- ▶ filter
- scan □
- velodyne_points □
- ▶ lidar_nodelet_manager_driver
- tf_static □
- pressure □
- tf □
- diagnostics □
- ▶ ublox_gps
- ▶ lidar_nodelet_manager
- gnss □

▼ /arena_camera_0_node/camera_info

```
• header                              [object]
  ○ seq:   63                         [number]
  ○ stamp                             [object]
    ▪ secs:   1682259792              [number]
    ▪ nsecs:   157651424              [number]
  ○ frame_id:   arena_camera_0        [string]
• height:   1536                      [number]
• width:   2048                       [number]
• distortion_model:   null            [string]
• D                                   [object]
  ○ D[0]:   0                         [number]
  ○ D[1]:   0                         [number]
  ○ D[2]:   0                         [number]
  ○ D[3]:   0                         [number]
  ○ D[4]:   0                         [number]
• K                                   [object]
  ○ K[0]:   0                         [number]
  ○ K[1]:   0                         [number]
  ○ K[2]:   0                         [number]
  ○ K[3]:   0                         [number]
  ○ K[4]:   0                         [number]
  ○ K[5]:   0                         [number]
  ○ K[6]:   0                         [number]
  ○ K[7]:   0                         [number]
  ○ K[8]:   0                         [number]
• R                                   [object]
  ○ R[0]:   0                         [number]
```

Selected Media: /arena_camera_0_node/image_raw

Creation Wizard and Node Graph



ROS Node Wizard

File type: Python

Node name: talker

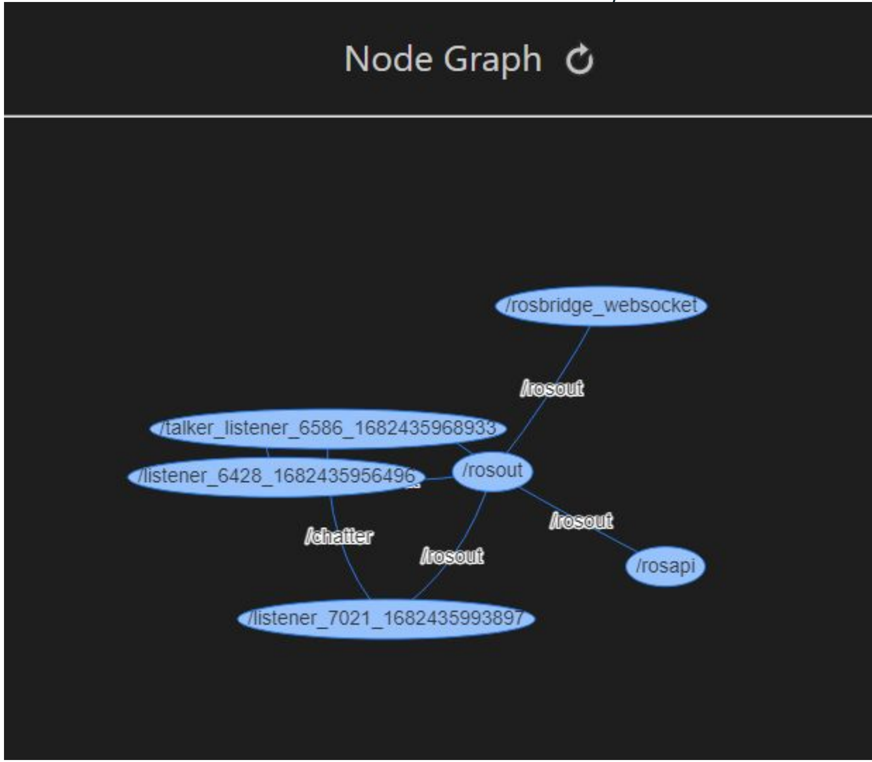
Node location: mypackage ...

☑ Publisher

☐ Subscriber

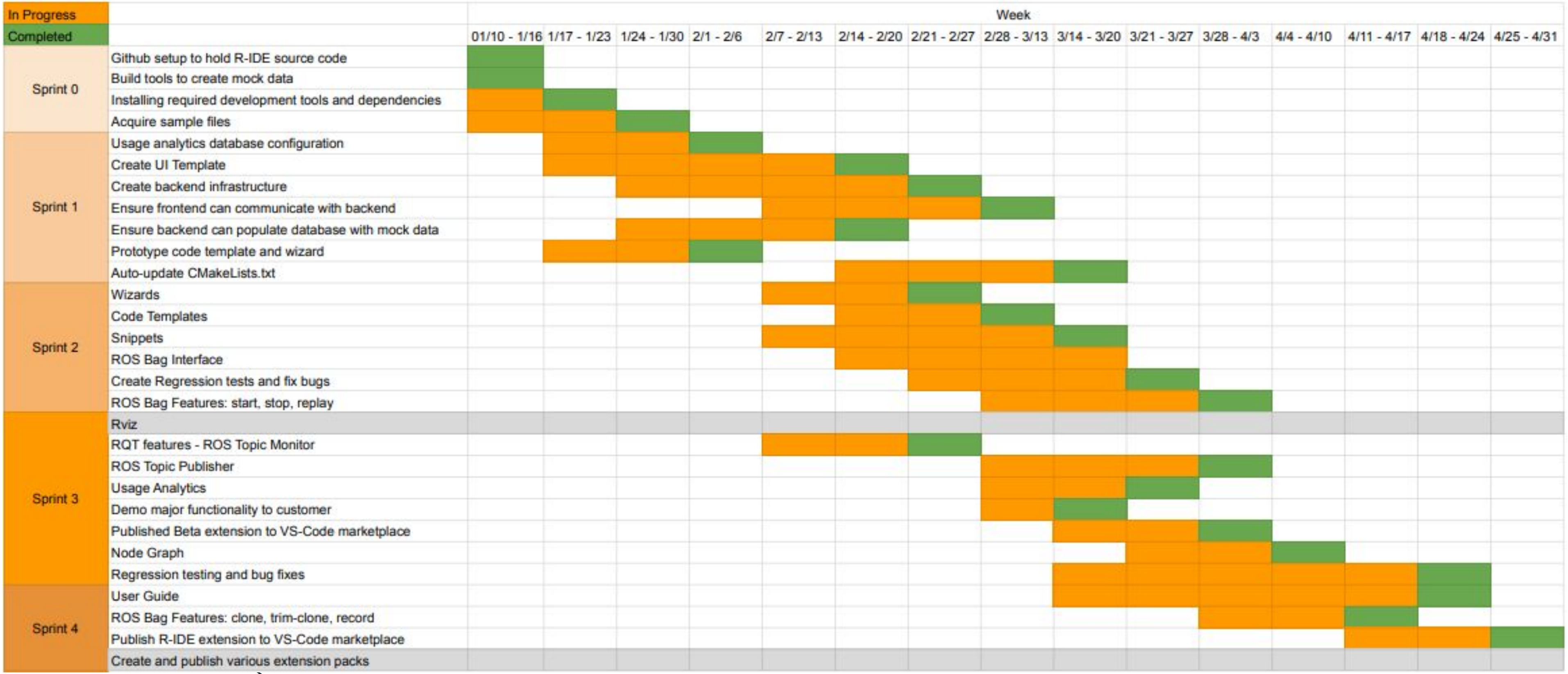
Cancel Next

```python
#!/usr/bin/env python
# license removed for brevity
import rospy
from std_msgs.msg import String

def callback(data):
    rospy.loginfo(rospy.get_caller_id() + "I heard %s", data.data)

def talker():

    pub = rospy.Publisher('chatter', String, queue_size=10)

    rate = rospy.Rate(10) # 10hz

    while not rospy.is_shutdown():
        hello_str = "hello world %s" % rospy.get_time()
        rospy.loginfo(hello_str)
        pub.publish(hello_str)
        rate.sleep()

if __name__ == '__main__':
    rospy.init_node("talker_listener", anonymous=True)
    rospy.Subscriber("listener", String, callback)
    try:
        talker()
    except rospy.ROSInterruptException:
        pass

# Reference: http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubs
```



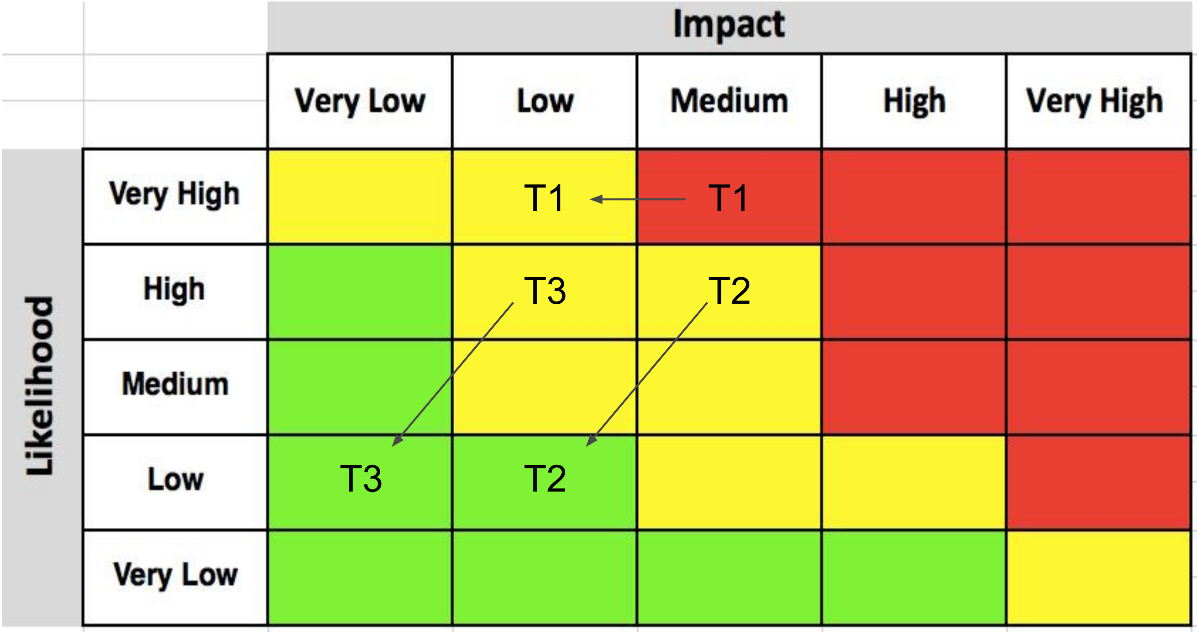
Node Graph
/rosbridge_websocket
/rosout
/talker_listener_6586_1682435968933
/listener_6428_1682435956496
/rosout
/rosout
/chatter
/rosout
/rosapi
/listener_7021_1682435993897

Development/Sprint Profile



In Progress 🟧		Week														
Completed 🟩		01/10 - 1/16	1/17 - 1/23	1/24 - 1/30	2/1 - 2/6	2/7 - 2/13	2/14 - 2/20	2/21 - 2/27	2/28 - 3/13	3/14 - 3/20	3/21 - 3/27	3/28 - 4/3	4/4 - 4/10	4/11 - 4/17	4/18 - 4/24	4/25 - 4/31
Sprint 0	Github setup to hold R-IDE source code	🟩														
	Build tools to create mock data	🟩														
	Installing required development tools and dependencies	🟧	🟩													
	Acquire sample files	🟧	🟧	🟩												
Sprint 1	Usage analytics database configuration		🟧	🟧	🟩											
	Create UI Template		🟧	🟧	🟧	🟧	🟩									
	Create backend infrastructure			🟧	🟧	🟧	🟧	🟩								
	Ensure frontend can communicate with backend					🟧	🟧	🟧	🟩							
	Ensure backend can populate database with mock data			🟧	🟧	🟧	🟩									
	Prototype code template and wizard		🟧	🟧	🟩											
	Auto-update CMakeLists.txt						🟧	🟧	🟧	🟩						
Sprint 2	Wizards					🟧	🟧	🟩								
	Code Templates						🟧	🟧	🟩							
	Snippets					🟧	🟧	🟧	🟧	🟩						
	ROS Bag Interface						🟧	🟧	🟧	🟧						
	Create Regression tests and fix bugs							🟧	🟧	🟧	🟩					
	ROS Bag Features: start, stop, replay								🟧	🟧	🟧	🟩				
Sprint 3	Rviz															
	RQT features - ROS Topic Monitor					🟧	🟧	🟩								
	ROS Topic Publisher								🟧	🟧	🟧	🟩				
	Usage Analytics								🟧	🟧	🟩					
	Demo major functionality to customer								🟧	🟩						
	Published Beta extension to VS-Code marketplace									🟧	🟧	🟩				
	Node Graph										🟧	🟧	🟩			
	Regression testing and bug fixes									🟧	🟧	🟧	🟧	🟧	🟩	
Sprint 4	User Guide									🟧	🟧	🟧	🟧	🟧	🟩	
	ROS Bag Features: clone, trim-clone, record											🟧	🟧	🟩		
	Publish R-IDE extension to VS-Code marketplace													🟧	🟧	🟩
	Create and publish various extension packs															

Technical Risk Matrix



		Impact				
		Very Low	Low	Medium	High	Very High
Likelihood	Very High	🟨	🟨 T1	🟥 T1	🟥	🟥
	High	🟩	🟨 T3	🟨 T2	🟥	🟥
	Medium	🟩	🟨	🟨	🟥	🟥
	Low	🟩 T3	🟩 T2	🟨	🟨	🟥
	Very Low	🟩	🟩	🟩	🟩	🟨

T1: VSCode API dependencies

- **Risk**: New updates may cause dependency issues
- **Mitigation**: Use vscode-test npm package

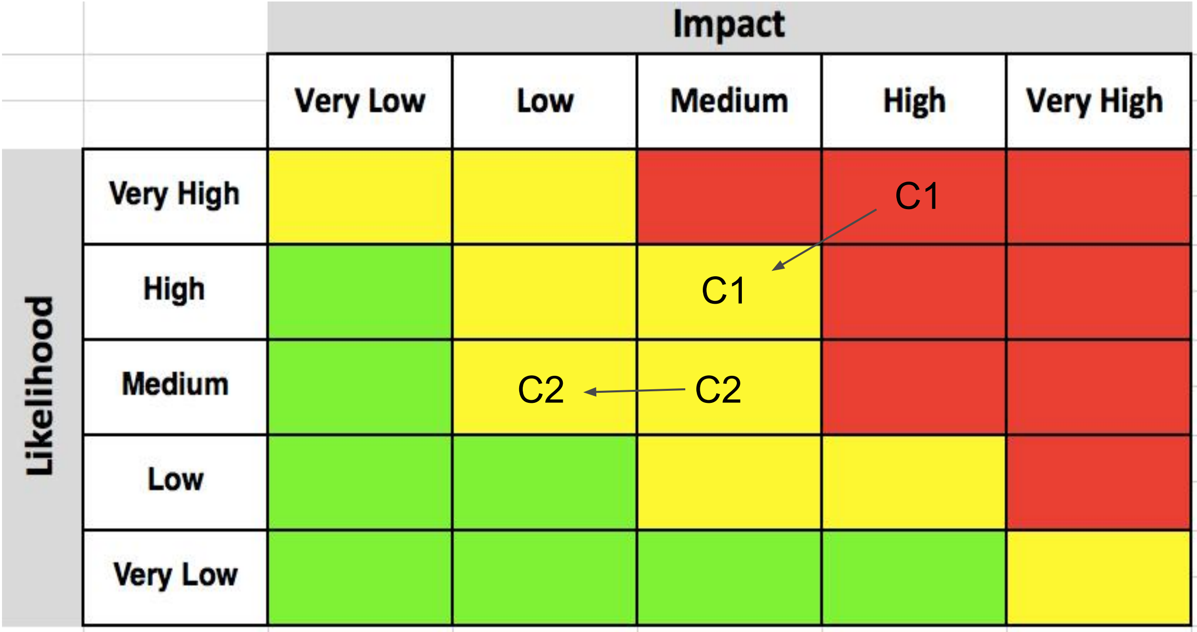
T2: Dependency issues between wizards

- **Risk**: New ROS versions make breaking changes
- **Mitigation**: Pinned version of ROS

T3: ROS1 → ROS2

- **Risk**: Communication between ROS1 and ROS2
- **Mitigation**: Using ros1_bridge, which allows bidirectional communication

Customer Risk Matrix



		Impact				
		Very Low	Low	Medium	High	Very High
Likelihood	Very High	🟨	🟨	🟥	🟥 C1	🟥
	High	🟩	🟨	🟨 C1	🟥	🟥
	Medium	🟩	🟨 C2	🟨 C2	🟥	🟥
	Low	🟩	🟩	🟨	🟨	🟥
	Very Low	🟩	🟩	🟩	🟩	🟨

C1: Knowledge gap

❖ **Risk**: Tutorials are too complex or too easy for users

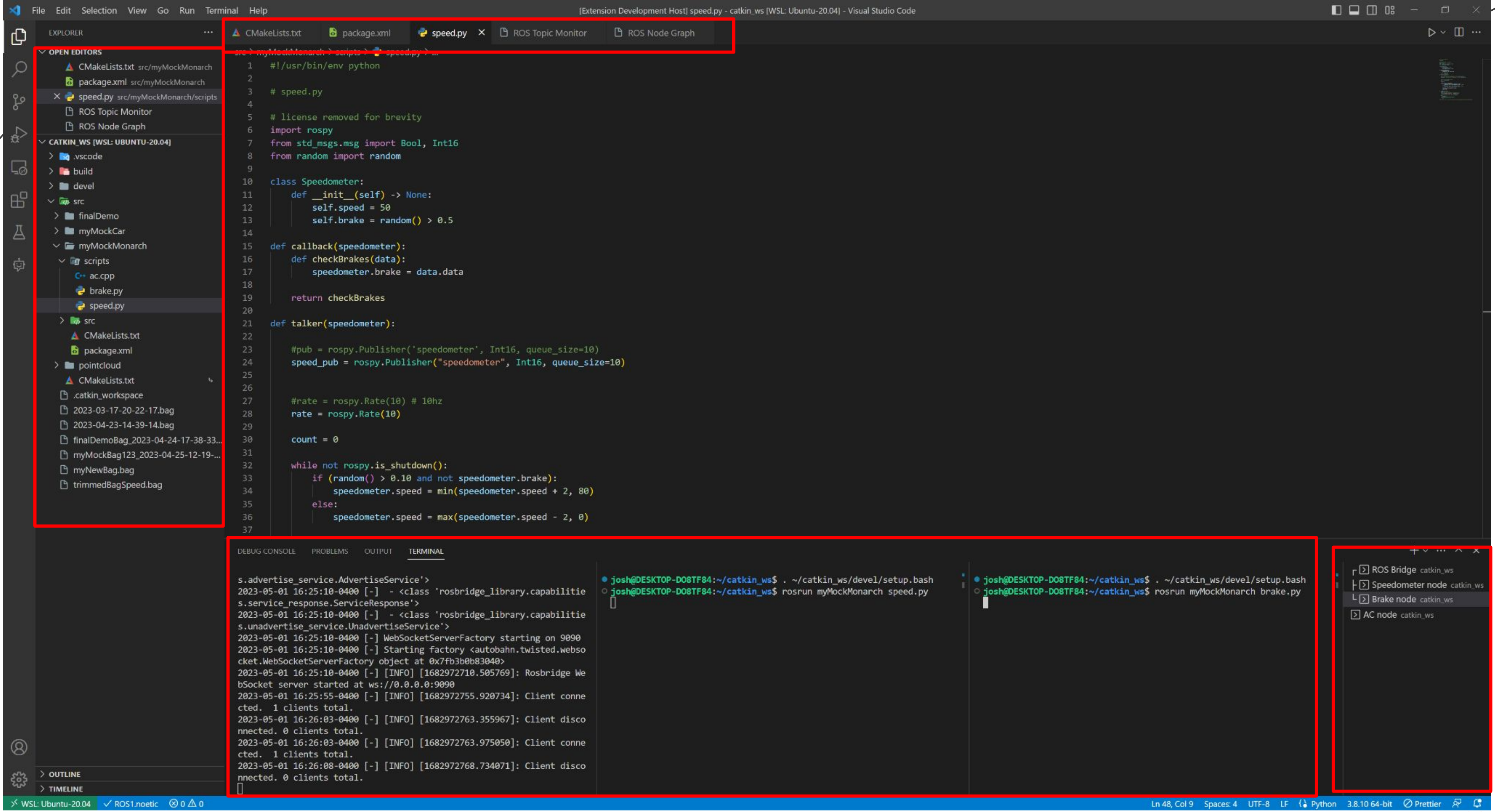
❖ **Mitigation**: Design in a modular manner

C2: Lack of Resources

❖ **Risk**: May not be able to cover all topics

❖ **Mitigation**: Documentation and well-written user requirements

Current Developer Interface



```python
#!/usr/bin/env python

# speed.py

# license removed for brevity
import rospy
from std_msgs.msg import Bool, Int16
from random import random

class Speedometer:
    def __init__(self) -> None:
        self.speed = 50
        self.brake = random() > 0.5

def callback(speedometer):
    def checkBrakes(data):
        speedometer.brake = data.data

    return checkBrakes

def talker(speedometer):

    #pub = rospy.Publisher('speedometer', Int16, queue_size=10)
    speed_pub = rospy.Publisher("speedometer", Int16, queue_size=10)


    #rate = rospy.Rate(10) # 10hz
    rate = rospy.Rate(10)

    count = 0

    while not rospy.is_shutdown():
        if (random() > 0.10 and not speedometer.brake):
            speedometer.speed = min(speedometer.speed + 2, 80)
        else:
            speedometer.speed = max(speedometer.speed - 2, 0)

```

```
s.advertise_service.AdvertiseService'>
2023-05-01 16:25:10-0400 [-]  - <class 'rosbridge_library.capabilitie
s.service_response.ServiceResponse'>
2023-05-01 16:25:10-0400 [-]  - <class 'rosbridge_library.capabilitie
s.unadvertise_service.UnadvertiseService'>
2023-05-01 16:25:10-0400 [-] WebSocketServerFactory starting on 9090
2023-05-01 16:25:10-0400 [-] Starting factory <autobahn.twisted.webso
cket.WebSocketServerFactory object at 0x7fb3b0b83040>
2023-05-01 16:25:10-0400 [-] [INFO] [1682972710.505769]: Rosbridge We
bSocket server started at ws://0.0.0.0:9090
2023-05-01 16:25:55-0400 [-] [INFO] [1682972755.920734]: Client conne
cted.  1 clients total.
2023-05-01 16:26:03-0400 [-] [INFO] [1682972763.355967]: Client disco
nnected. 0 clients total.
2023-05-01 16:26:03-0400 [-] [INFO] [1682972763.975050]: Client conne
cted.  1 clients total.
2023-05-01 16:26:08-0400 [-] [INFO] [1682972768.734071]: Client disco
nnected. 0 clients total.
```

```
josh@DESKTOP-DO8TF84:~/catkin_ws$ . ~/catkin_ws/devel/setup.bash
josh@DESKTOP-DO8TF84:~/catkin_ws$ rosrun myMockMonarch speed.py
```

```
josh@DESKTOP-DO8TF84:~/catkin_ws$ . ~/catkin_ws/devel/setup.bash
josh@DESKTOP-DO8TF84:~/catkin_ws$ rosrun myMockMonarch brake.py
```

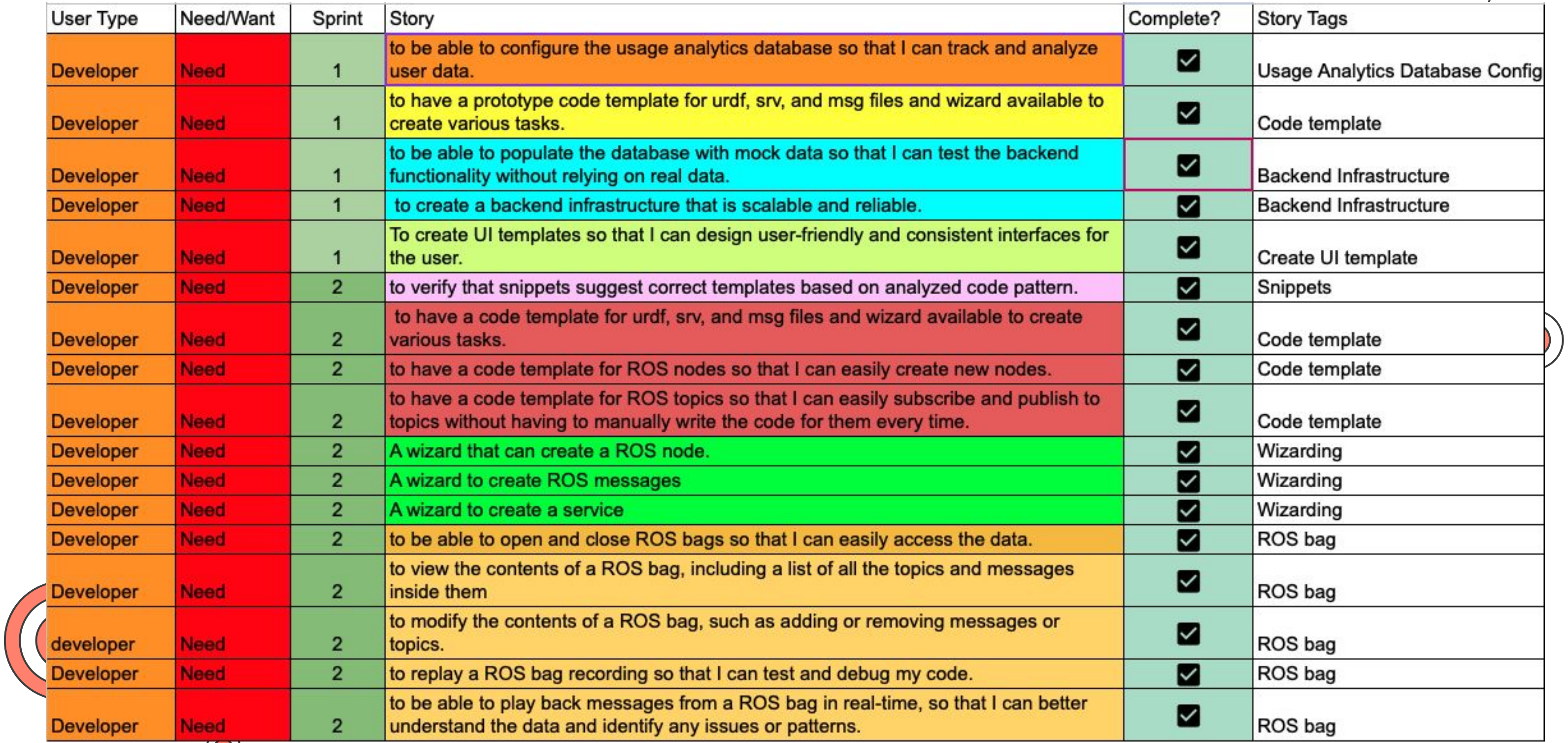
Demo Summary

- Where to find us - Dan
- ROS Package Creation - Dan
- Creation Wizard - Gavin
- Snippets - Justin
- Create Executable - Dan
- Node Graph - Justin
- Topic Monitor - Josh
- ROS Bag Manager - Dominik

Appendix A: If we had more time...

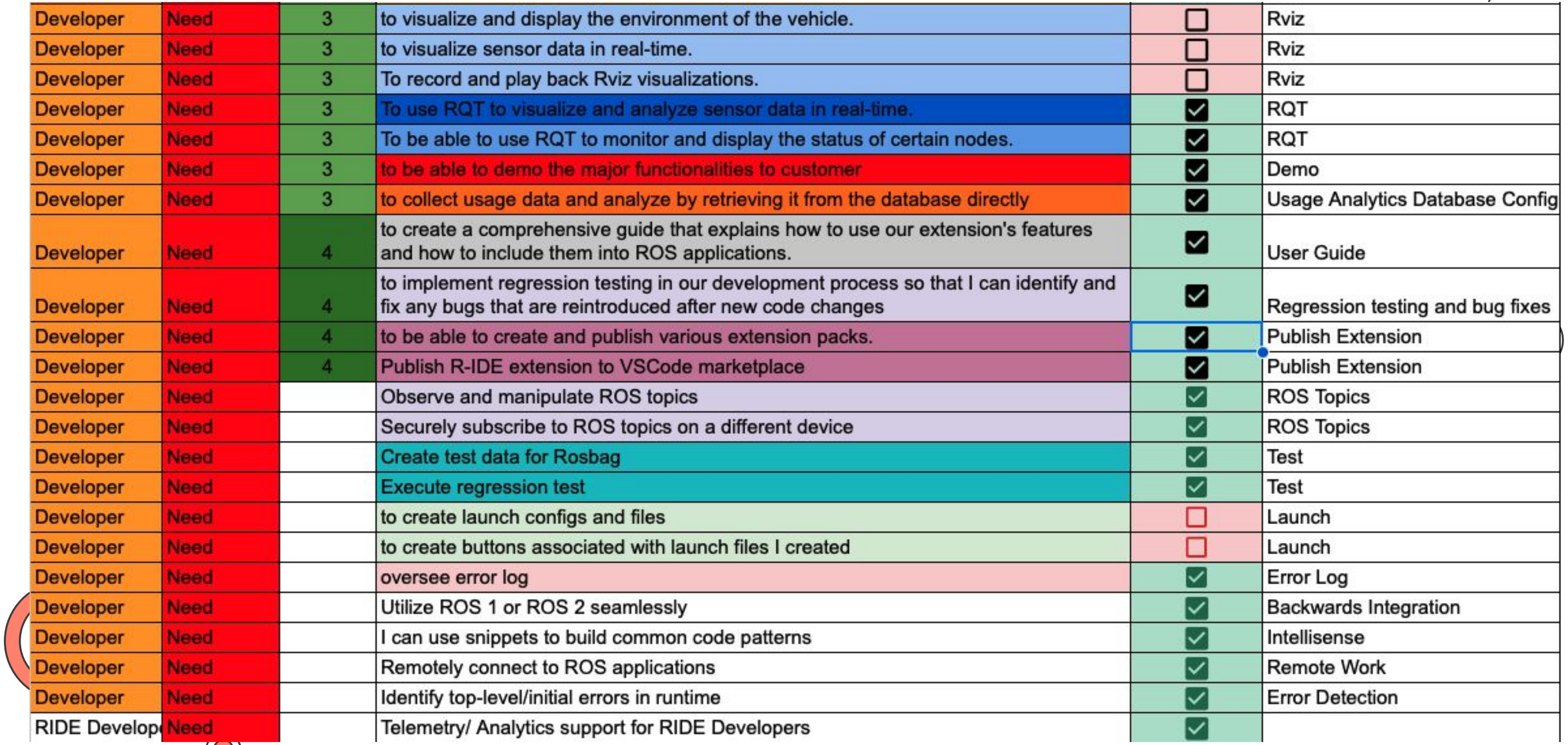
- Added functionality to the node graph
- More regression testing
- Variable rosbag playback speed
- More ROS development snippets
- Further improvements on auto updating config files
- Improvements to Automated testing
- Recreating relevant RVIZ functionality
- Improve usage analytics to provide more useful information

Appendix B: Prototype user stories as implemented.



User Type	Need/Want	Sprint	Story	Complete?	Story Tags
Developer	Need	1	to be able to configure the usage analytics database so that I can track and analyze user data.	☑	Usage Analytics Database Config
Developer	Need	1	to have a prototype code template for urdf, srv, and msg files and wizard available to create various tasks.	☑	Code template
Developer	Need	1	to be able to populate the database with mock data so that I can test the backend functionality without relying on real data.	☑	Backend Infrastructure
Developer	Need	1	to create a backend infrastructure that is scalable and reliable.	☑	Backend Infrastructure
Developer	Need	1	To create UI templates so that I can design user-friendly and consistent interfaces for the user.	☑	Create UI template
Developer	Need	2	to verify that snippets suggest correct templates based on analyzed code pattern.	☑	Snippets
Developer	Need	2	to have a code template for urdf, srv, and msg files and wizard available to create various tasks.	☑	Code template
Developer	Need	2	to have a code template for ROS nodes so that I can easily create new nodes.	☑	Code template
Developer	Need	2	to have a code template for ROS topics so that I can easily subscribe and publish to topics without having to manually write the code for them every time.	☑	Code template
Developer	Need	2	A wizard that can create a ROS node.	☑	Wizarding
Developer	Need	2	A wizard to create ROS messages	☑	Wizarding
Developer	Need	2	A wizard to create a service	☑	Wizarding
Developer	Need	2	to be able to open and close ROS bags so that I can easily access the data.	☑	ROS bag
Developer	Need	2	to view the contents of a ROS bag, including a list of all the topics and messages inside them	☑	ROS bag
developer	Need	2	to modify the contents of a ROS bag, such as adding or removing messages or topics.	☑	ROS bag
Developer	Need	2	to replay a ROS bag recording so that I can test and debug my code.	☑	ROS bag
Developer	Need	2	to be able to play back messages from a ROS bag in real-time, so that I can better understand the data and identify any issues or patterns.	☑	ROS bag

Appendix B: Prototype user stories as implemented.



Developer	Need	3	to visualize and display the environment of the vehicle.	□	Rviz
Developer	Need	3	to visualize sensor data in real-time.	□	Rviz
Developer	Need	3	To record and play back Rviz visualizations.	□	Rviz
Developer	Need	3	To use RQT to visualize and analyze sensor data in real-time.	☑	RQT
Developer	Need	3	To be able to use RQT to monitor and display the status of certain nodes.	☑	RQT
Developer	Need	3	to be able to demo the major functionalities to customer	☑	Demo
Developer	Need	3	to collect usage data and analyze by retrieving it from the database directly	☑	Usage Analytics Database Config
Developer	Need	4	to create a comprehensive guide that explains how to use our extension's features and how to include them into ROS applications.	☑	User Guide
Developer	Need	4	to implement regression testing in our development process so that I can identify and fix any bugs that are reintroduced after new code changes	☑	Regression testing and bug fixes
Developer	Need	4	to be able to create and publish various extension packs.	☑	Publish Extension
Developer	Need	4	Publish R-IDE extension to VSCode marketplace	☑	Publish Extension
Developer	Need		Observe and manipulate ROS topics	☑	ROS Topics
Developer	Need		Securely subscribe to ROS topics on a different device	☑	ROS Topics
Developer	Need		Create test data for Rosbag	☑	Test
Developer	Need		Execute regression test	☑	Test
Developer	Need		to create launch configs and files	□	Launch
Developer	Need		to create buttons associated with launch files I created	□	Launch
Developer	Need		oversee error log	☑	Error Log
Developer	Need		Utilize ROS 1 or ROS 2 seamlessly	☑	Backwards Integration
Developer	Need		I can use snippets to build common code patterns	☑	Intellisense
Developer	Need		Remotely connect to ROS applications	☑	Remote Work
Developer	Need		Identify top-level/initial errors in runtime	☑	Error Detection
RIDE Develop	Need		Telemetry/ Analytics support for RIDE Developers	☑	

Thanks for your time!

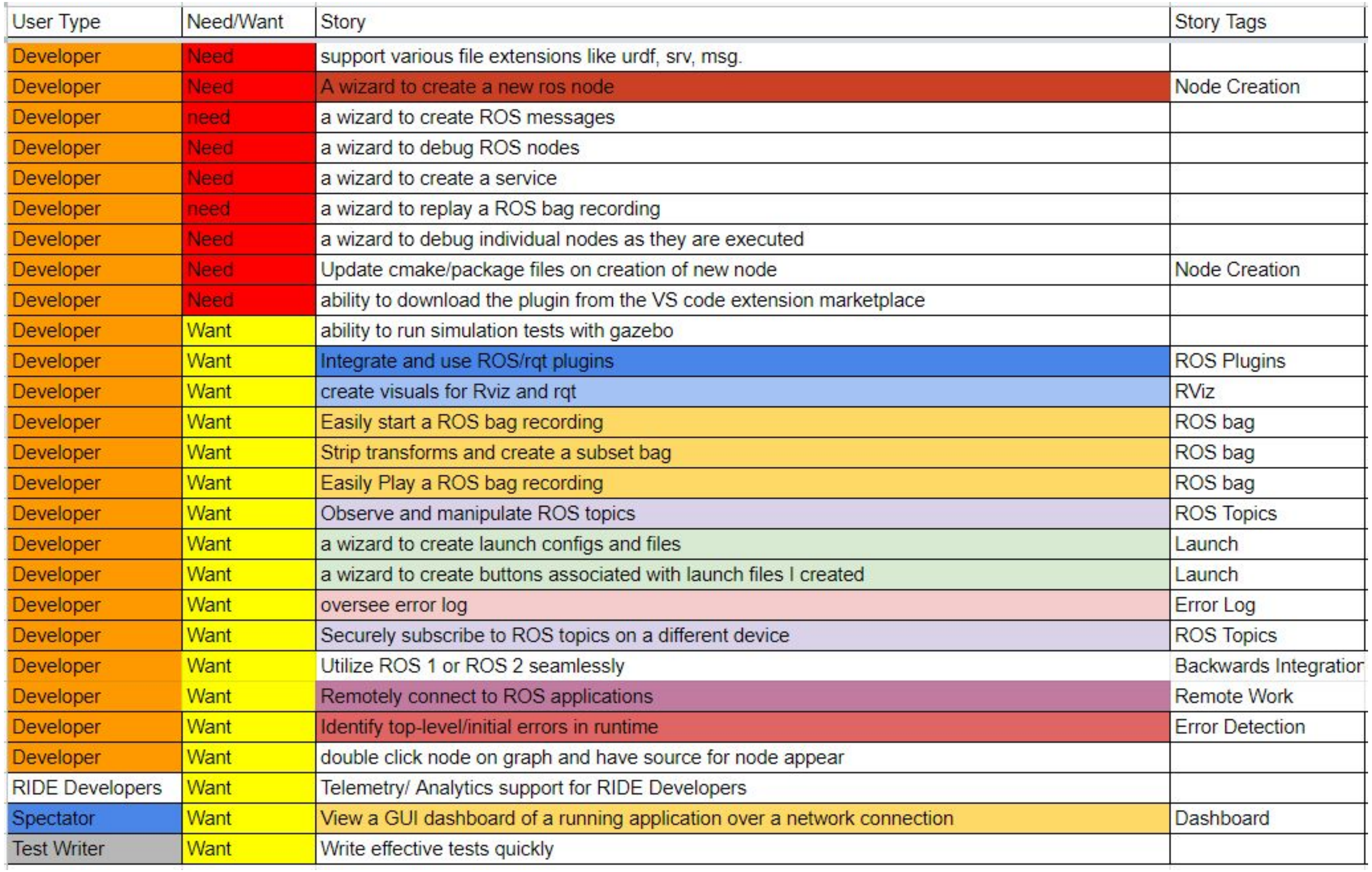
Any questions?

Appendix C: References

Citation

1. "Robot Operating System." ROS, www.ros.org/.
2. "Global Robot Operating System Market Research Report 2022(Status and Outlook)." Market Reports, www.marketreportsworld.com/global-robot-operating-system-market-21185690.
3. "Robot Operating System Market." Future Market Insights, www.futuremarketinsights.com/reports/robot-operating-system-market.
4. "Lunar Rover." Open Robotics, www.openrobotics.org/customer-stories/lunar-rover.
5. Wessling, Brianna. "Open Robotics Developing Space ROS with Blue Origin, NASA." The Robot Report, 11 Feb. 2022, www.therobotreport.com/open-robotics-developing-space-ros/.
6. Ackerman, Evan. "Microsoft Announces Experimental Release of ROS for Windows 10." IEEE Spectrum, IEEE Spectrum, 24 June 2021, spectrum.ieee.org/microsoft-announces-experimental-release-of-ros-for-windows-10.
7. Tomoya Fujita. "aibo with ROS", https://roscon.ros.org/2018/presentations/ROSCon2018_Aibo.pdf
8. Brian Gerkey on September 1, 2014 8:07 AM. "ROS Running on ISS." Ros.org, www.ros.org/news/2014/09/ros-running-on-iss.html.
9. M. Cardona, A. Palma and J. Manzanares, "COVID-19 Pandemic Impact on Mobile Robotics Market," 2020 IEEE ANDESCON, 2020, pp. 1-4, doi: 10.1109/ANDESCON50619.2020.9272052. https://ieeexplore.ieee.org/abstract/document/9272052
10. International Federation of Robotics. Annual installations of industrial robots 2019-2020 and 2021*-2024* [Graph]. Retrieved from https://ifr.org/downloads/press2018/2021_10_28_WR_PK_Presentation_long_version.pdf
11. "Wiki." Ros.org, wiki.ros.org/Security.
12. Dieber, Bernhard, et al. "Security for the robot operating system." *Robotics and Autonomous Systems* 98 (2017): 192-203.

User Stories



User Type	Need/Want	Story	Story Tags
Developer	Need	support various file extensions like urdf, srv, msg.	
Developer	Need	A wizard to create a new ros node	Node Creation
Developer	need	a wizard to create ROS messages	
Developer	Need	a wizard to debug ROS nodes	
Developer	Need	a wizard to create a service	
Developer	need	a wizard to replay a ROS bag recording	
Developer	Need	a wizard to debug individual nodes as they are executed	
Developer	Need	Update cmake/package files on creation of new node	Node Creation
Developer	Need	ability to download the plugin from the VS code extension marketplace	
Developer	Want	ability to run simulation tests with gazebo	
Developer	Want	Integrate and use ROS/rqt plugins	ROS Plugins
Developer	Want	create visuals for Rviz and rqt	RViz
Developer	Want	Easily start a ROS bag recording	ROS bag
Developer	Want	Strip transforms and create a subset bag	ROS bag
Developer	Want	Easily Play a ROS bag recording	ROS bag
Developer	Want	Observe and manipulate ROS topics	ROS Topics
Developer	Want	a wizard to create launch configs and files	Launch
Developer	Want	a wizard to create buttons associated with launch files I created	Launch
Developer	Want	oversee error log	Error Log
Developer	Want	Securely subscribe to ROS topics on a different device	ROS Topics
Developer	Want	Utilize ROS 1 or ROS 2 seamlessly	Backwards Integratior
Developer	Want	Remotely connect to ROS applications	Remote Work
Developer	Want	Identify top-level/initial errors in runtime	Error Detection
Developer	Want	double click node on graph and have source for node appear	
RIDE Developers	Want	Telemetry/ Analytics support for RIDE Developers	
Spectator	Want	View a GUI dashboard of a running application over a network connection	Dashboard
Test Writer	Want	Write effective tests quickly	

References Cont.

Competition Matrix References

1. CLion IDE: https://www.jetbrains.com/clion/
2. CLion Hatchery: https://plugins.jetbrains.com/plugin/10290-hatchery
3. CLion ROS Support: https://plugins.jetbrains.com/plugin/11235-ros-support
4. VS Code: https://code.visualstudio.com/
5. QT Creator with ROS: https://github.com/ros-industrial/ros_qtc_plugin/blob/devel/gh_pages/index.rst
6. Emacs with Rosemacs: http://wiki.ros.org/rosemacs
7. VIM ROS: https://github.com/taketwo/vim-ros
8. TMUX: https://github.com/tmux/tmux
9. GNU Screen: https://www.gnu.org/software/screen/
10. Terminator: https://github.com/gnome-terminator/terminator
11. "Software Design Report for the ODU Monarch I", Version 1.3, Jul 8, 2022, page 3. Author Lee Belfore”

Glossary

https://www.cs.odu.edu/~410orang/#/glossary

ROS - The Robot Operating System (ROS) is an open source set of software libraries and tools that help developers build robot applications.

ROS Node - A node is a process that performs computation. Nodes are combined together and communicate with one another using streaming topics, RPC services, and the Parameter Server. These nodes are meant to operate at a fine-grained scale; a robot control system will usually comprise many nodes. For example, one node controls a laser range-finder, one Node controls the robot's wheel motors, one node performs localization, one node performs path planning, one node provides a graphical view of the system, and so on.

Autonomous Machine - A machine capable of sensing its environment, carrying out computations to make decisions, and performing actions in the real world.

ROS Bag - A bag is a file format in ROS for storing ROS message data. Bags are the primary mechanism in ROS for data logging, which means that they have a variety of offline uses.

ROS Topic - Topics are named buses over which nodes exchange messages. Topics have anonymous publish/subscribe semantics, which decouples the production of information from its consumption.

Glossary cont..

Rviz - (Ros Visualization) A 3D visualizer for displaying sensor data and state information from ROS

Gazebo - Gazebo is a real-world physics simulator that creates a world and simulates the robot

RQT - A QT-based framework for GUI development for ROS. It contains tools that support ROS topics, bags, node graphs, and many other tools for visualization and manipulation of ROS nodes

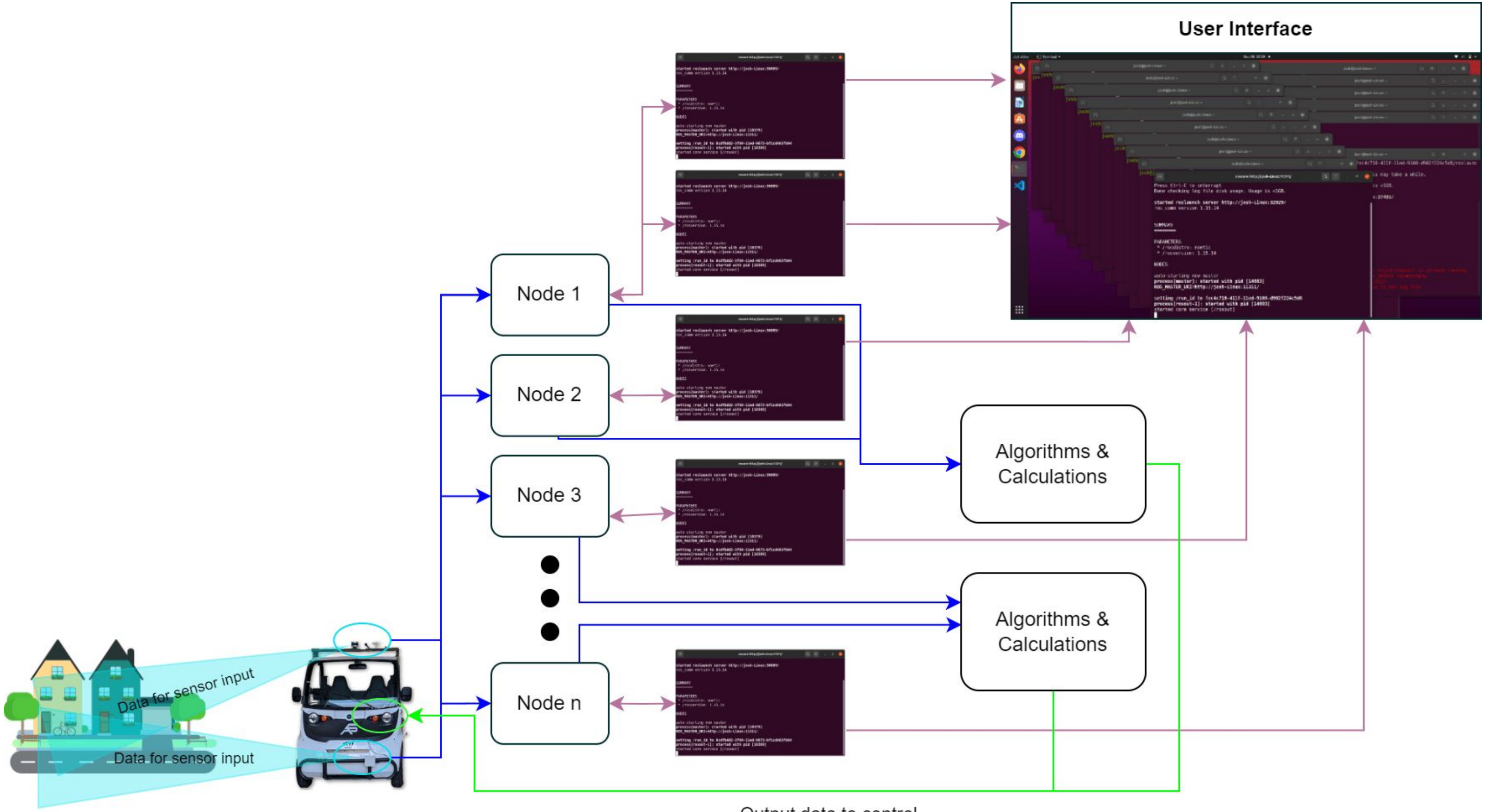
Wizard - A user interface that presents dialog to lead a user through a sequence of steps. Often used to configure a service for the first time or to simplify a complex or unfamiliar process.

Template - An editable text or code snippet that can be filled with given values from another tool like a wizard.

Tutorial - A method of transferring knowledge that teach via example and supplies the information to complete a given task

VSCode Extension - A tool designed to add additional features and capabilities to VSCode. It can create new dialogues, add functions, or change the appearance of VSCode

Current Developer Interface Diagram



Output data to control vehicle actuators