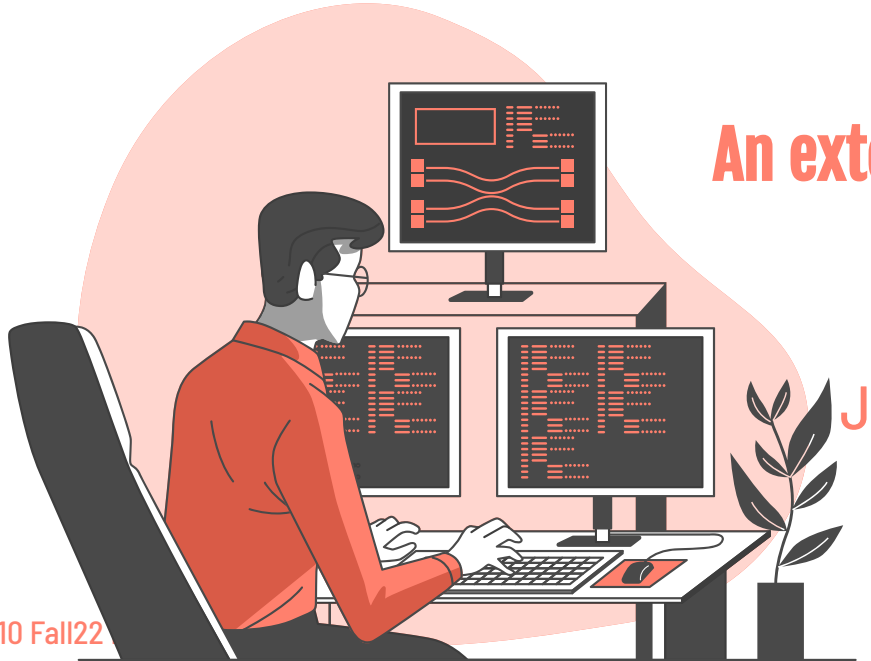
R-IDE

An extension to simplify robotics development with ROS



CS 410 - Prototype
Josh Peterson, Dominik Soós,
Dan Koontz, Gavin St.Clair,
James Hart, 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

Dr. Lee Belfore
Mentor

Laura Sakos
Mentor

Gavin St. Clair
Database Specialist

Justin Tymkin
Backend/Algorithm Specialist,
Documentation Specialist

James Hart
Frontend UI/UX Specialist

Gavin Alberghini
Mentor

Problem Characteristics

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

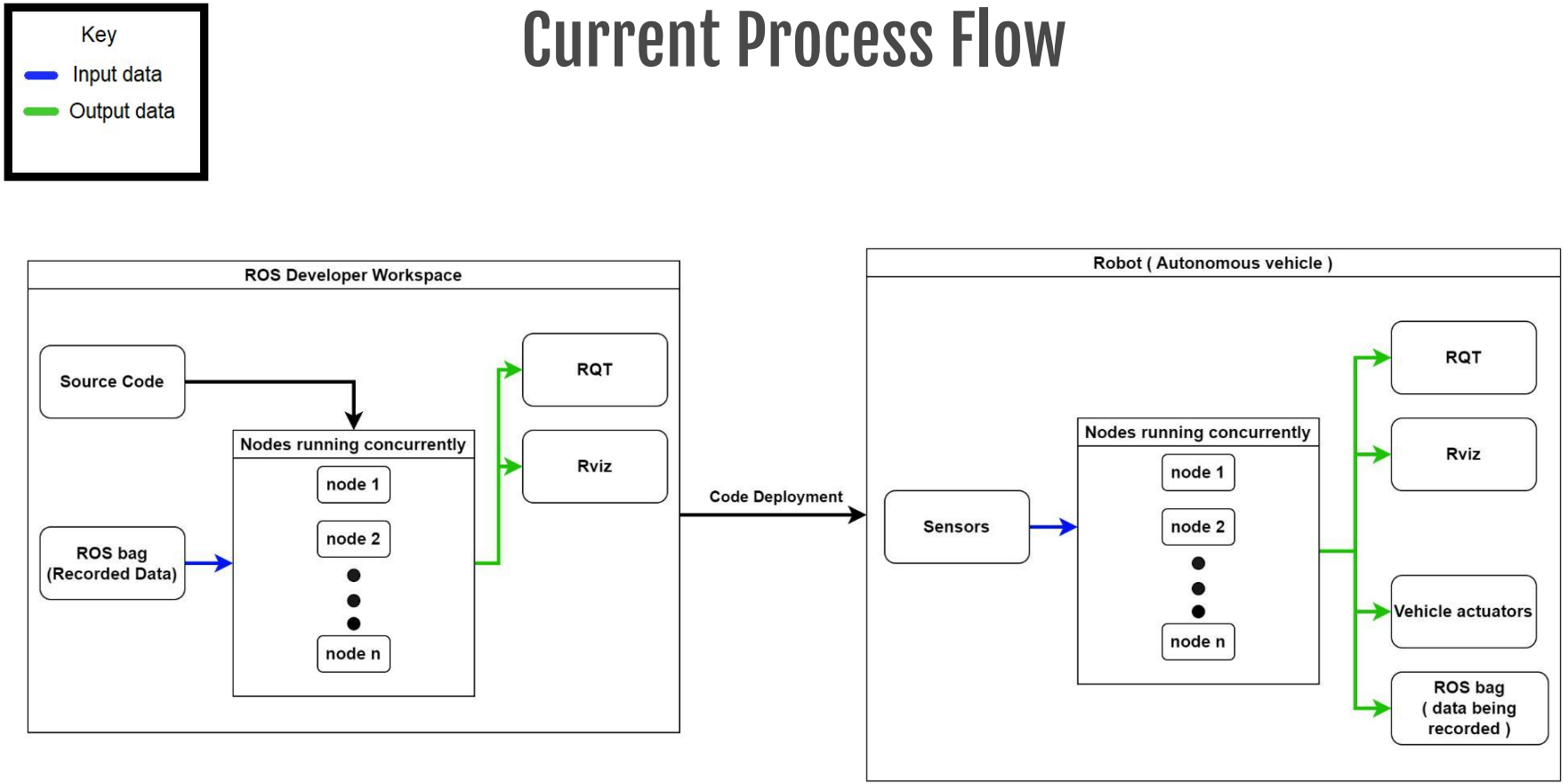
- Each ROS node has at least one terminal window associated with it
- There is no distinction when changes in one node affects a related node
- Simple debugging methodologies become time consuming due to how ROS displays errors

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

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

Current Process Flow



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

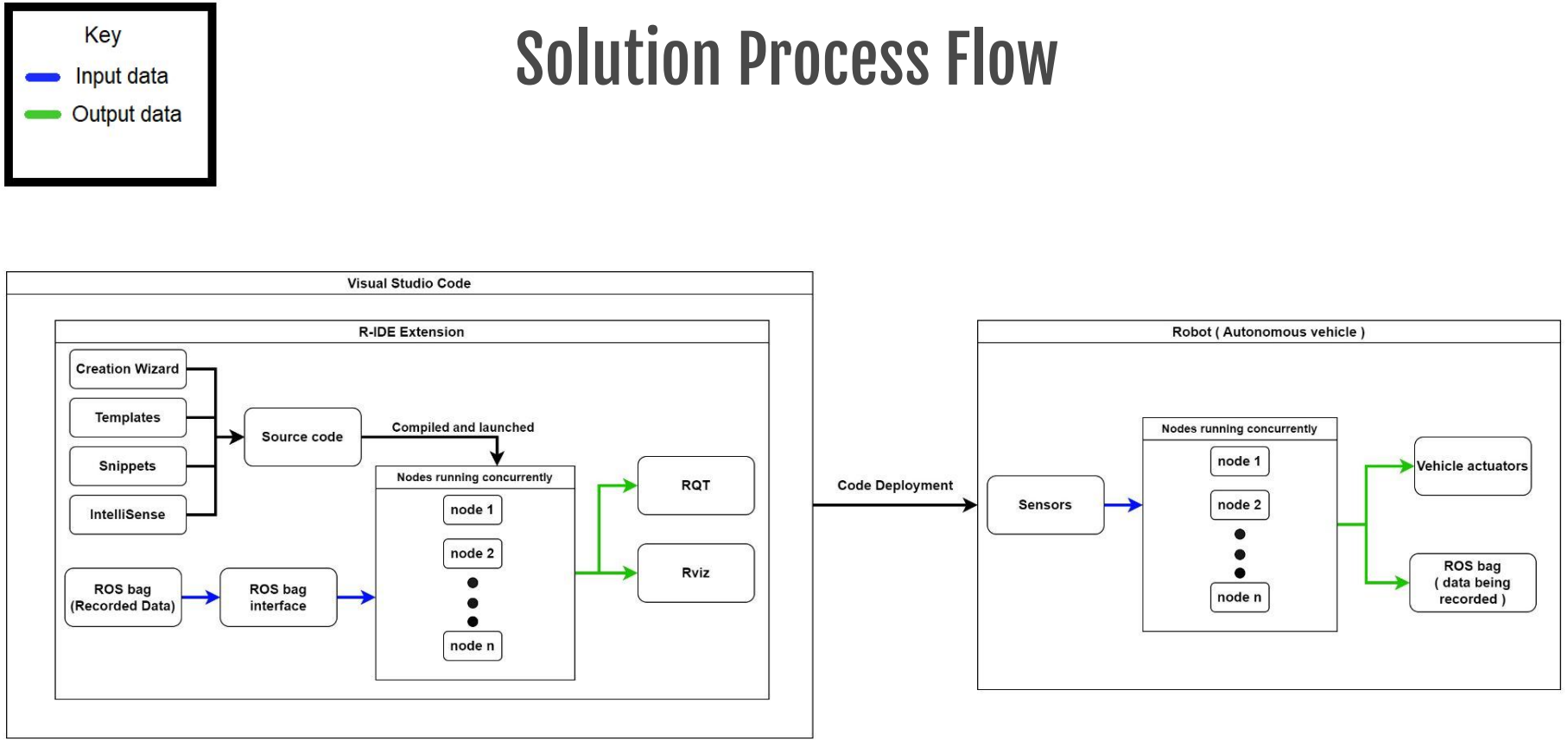
Create custom solutions that can analyze and autofill based on preexisting code

Perspective Shifts

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

An extension pack to simplify and speed up the development lifecycle and learning process.

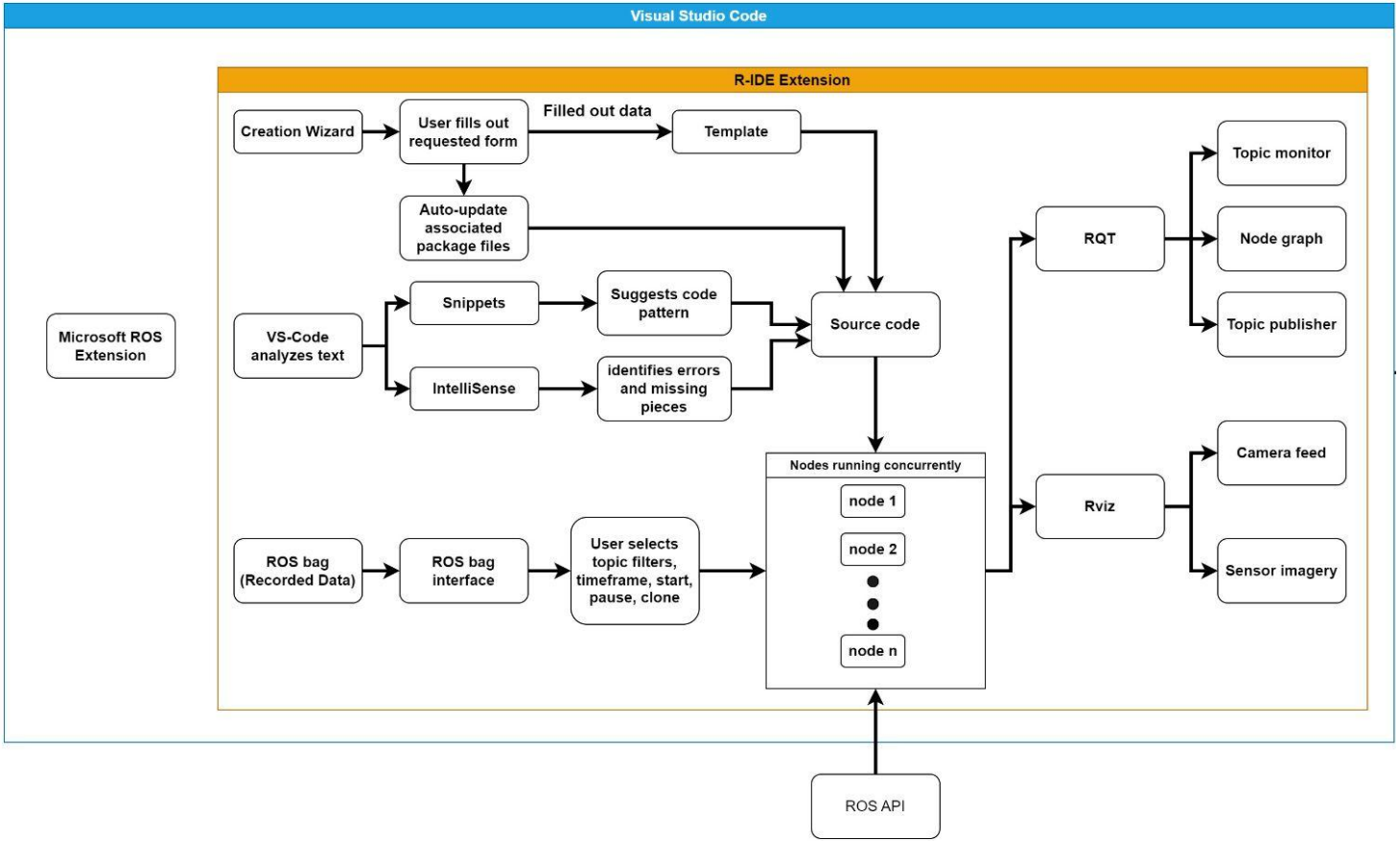
Solution Process Flow



Real World Product vs Prototype

	Feature	RWP	Prototype
Wizard	Create node	Full	Full
	Create msg	Full	Full
	Create srv	Full	Full
Auto update	cmake file	Full	Full
	package.xml	Full	Full
Snippets	Autocomplete Code Patterns	Full	Full
ROS bag	Start Rosbag recording	Full	Full
	Stop Rosbag recording	Full	Full
	Play back Rosbag	Full	Full
Visuals	rviz	Full	Partial: Depending on performance of embedded features
	Node Graph	Full	Full
ROS Topic	ROS topic monitor	Full	Full
	ROS topic publisher	Full	Full
Data Management	Usage Analysis	Full	Full

Major Functional Component Diagram



Extension Software/Hardware Development Resources

Software Requirements

ROS
Rviz
OpenGL

Database

MySQL

Languages

Javascript
Html/CSS
Python
C++
XML

Developer OS

Linux
Windows*
Mac*

IDE

VS-Code

Hardware

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

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

*ROS1 requires some kind of linux environment such as WSL or a virtual machine with linux

Prototype User Stories

User Type	Need/Want	Story
Developer	Need	Support various file extensions like urdf, srv, msg.
Developer	Need	A wizard to create a new ros node
Developer	Need	A wizard to create ROS messages
Developer	Need	A wizard to create a service
Developer	Need	Update cmake/package files on creation of new node
Developer	Need	Ability to download the plugin from the VS code extension marketplace
Developer	Want	Integrate and use ROS/rqt plugins
Developer	Want	create visuals for Rviz and rqt
developer	want	double click node on graph and have source for node appear
Developer	Want	Easily start a ROS bag recording
Developer	Want	Strip transforms and create a subset bag
Developer	Want	Easily Play a ROS bag recording
Developer	Want	Observe and manipulate ROS topics
Developer	want	Securely subscribe to ROS topics on a different device
Developer	Want	to create launch configs and files
Developer	Want	to create buttons associated with launch files I created
Developer	Want	oversee error log
Developer	Want	Utilize ROS 1 or ROS 2 seamlessly
Developer	Want	I can use snippets to build common code patterns
Developer	Want	Remotely connect to ROS applications
Developer	Want	Identify top-level/initial errors in runtime
RIDE Developers	Want	Telemetry/ Analytics support for RIDE Developers

Tester User Stories

User Type	Need/Want	Story
Tester	Need	Verify ROS bag recording interacts correctly with the ROSbag interface
Tester	Need	Verify that the wizard for creating a new node is easy to use and guides the user through the process step-by-step
Tester	Need	test the functionality of the wizard for creating a new message (to ensure that it generates the correct code and allows te user to customize the message as needed)
Tester	Need	verify that the srv wizard is able to generate the correct code for creating a new service (so that the user can easily add new functionality to their ROS system)
Tester	Need	test the ROS bag recording feature to ensure that it correctly captures and saves all of the data from the ROS system (so that it can be used for analysis and playback)
Tester	Want	verify that the rqt and rviz visuals are easy to use and provide useful information to the user (so that they can better understand and analyze their ROS system)
Tester	Need	verify that snippets suggests correct templates based on analyzed code pattern
Tester	Need	verify that intellisense marks errors and/or missing pieces accurately
Tester	Need	Verify that RIDE is accessible from the Extension marketplace and downloads and configures correctly on a variety of machines and operating systems
Tester	Need	I need to verify that the correct snippets are suggested and that they fill code correctly
Tester	Need	I need to verify that wizards correctly fill templates and that the file is placed in the correct directory
Tester	Need	I need to verify that the Wizards remember file and directory locations between uses

Prototype Sprint Plan

Sprint 0: Setting up development environment
- Repository to hold RIDE's source code
- Installing required development tools and dependencies

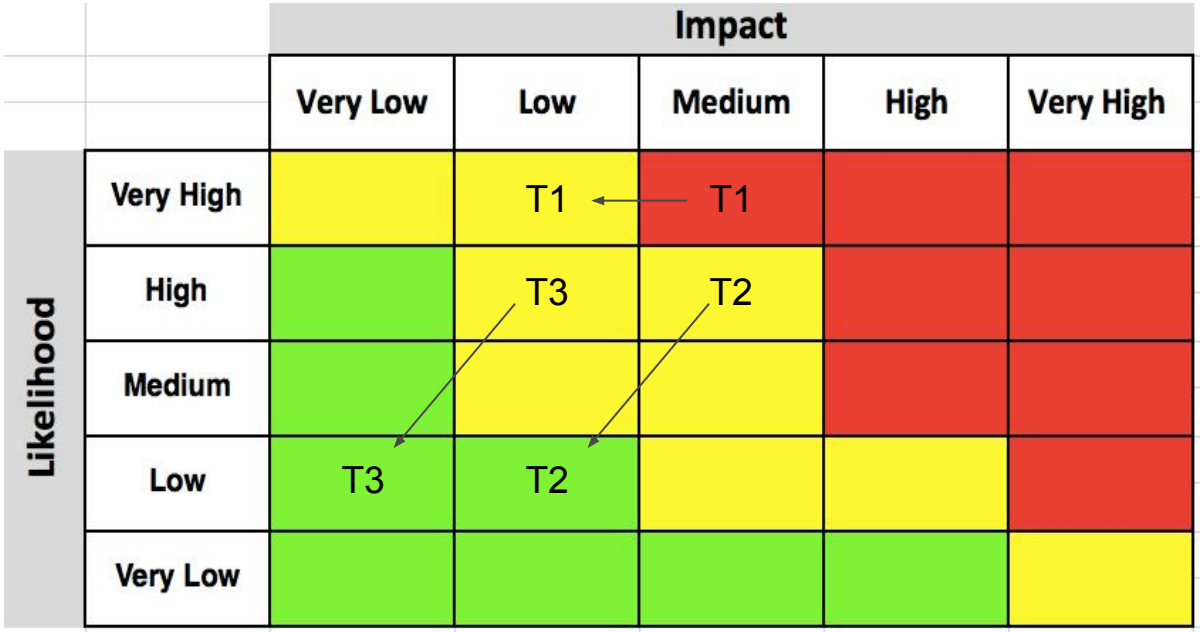
Sprint 1: Basic views, backend, and database
- Usage analytics database configuration
- Basic frontend views, and backend built
- Ensure frontend can communicate with backend
- Ensure backend can populate database with mock data

Sprint 2: Implement "Smart" features
- Wizards
- Templates
- Snippets
- ROS Bag Interface
- Regression testing and bug fixes

Sprint 3: Usage analytics and visuals
- Rviz
- RQT features
- Usage analytics
- Demo major functionality to customer
- Regression testing and bug fixes

Sprint 4: Extension pack creation, and documentation
- User guide
- Publish R-IDE extension to VS-code marketplace
- Create and publish various extension packs
- Regression testing and bug fixes

Technical Risk Matrix



T1: VSCode API dependencies

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

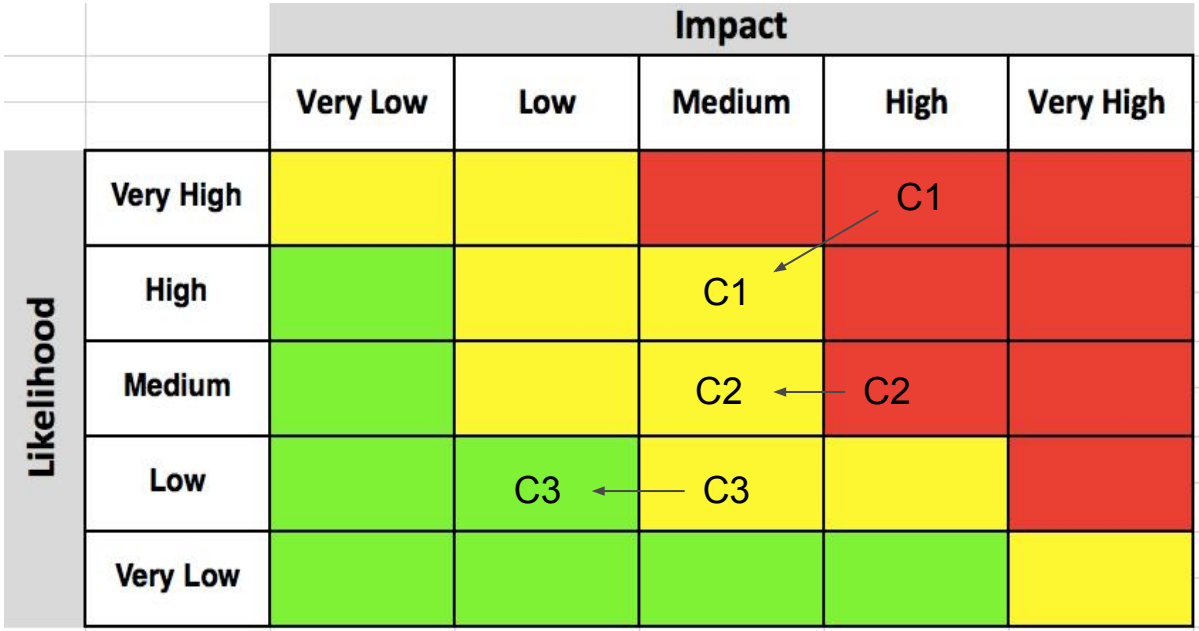
T2: Dependency issues between wizards

- **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



C1: Knowledge gap

- **Risk**: Tutorials are too complex or too easy for users
- **Mitigation**: Design in a modular manner

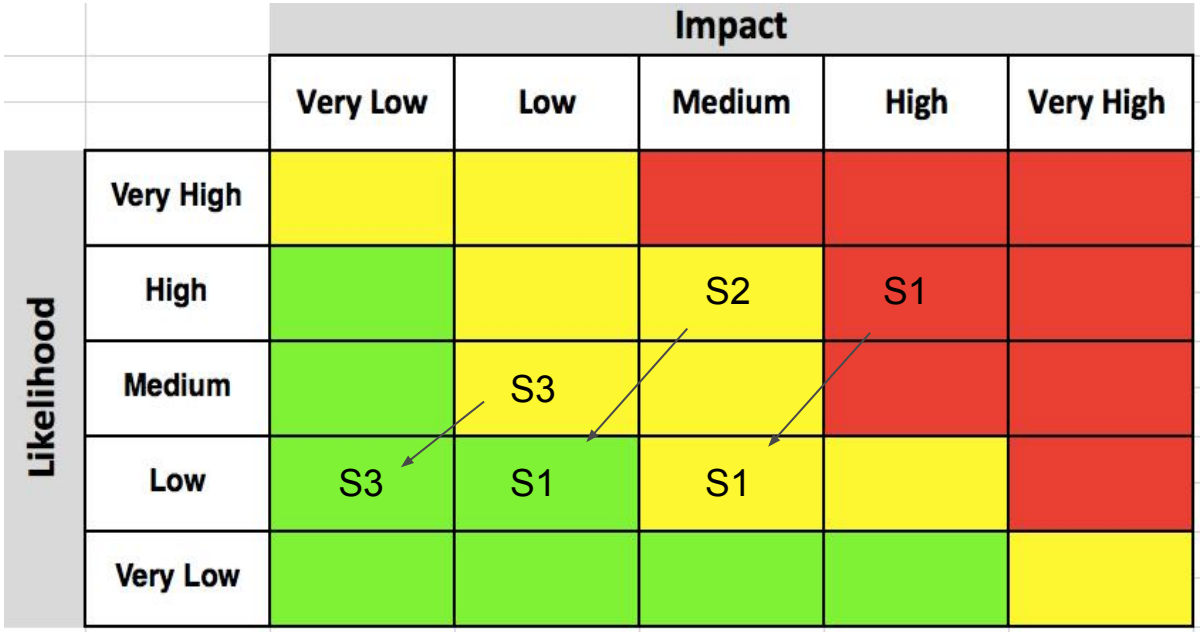
C2: How do we know we improved user experience?

- **Risk**: Progress is difficult to quantifiably measure
- **Mitigation**: Frequent status checks and surveys with users

C3: Lack of Resources

- **Risk**: May not be able to cover all topics
- **Mitigation**: Documentation and well-written user requirements

Security Risk Matrix



S1: Unauthorized data access

- **Risk**: Gain sensitive data about user from ROS nodes
- **Mitigation**: Avoid collecting sensitive data and secure communication channel

S2: Uploading data into the database

- **Risk**: Unauthorized access to data while inserting
- **Mitigation**: Encryption while transmitting data

S3: MySQL Server

- **Risk**: Unauthorized access to data while inserting
- **Mitigation**: User authentication and encryption

Conclusion

- ROS developers need an accessible way to build ROS projects
- RIDE will simplify and speed up development with code automation tools
- RIDE will allow developers to visualize and manipulate data from ROS nodes
- Our system will allow us to monitor our product and make changes quickly

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.

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