

1. Home	2
1.1 Project Background	5
1.1.1 Goal model	6
1.1.2 Scope	7
1.1.3 Project Plan	8
1.1.4 Product Backlog	10
1.1.4.1 Sprint 1	11
1.1.4.2 Sprint 2	16
1.1.4.3 Sprint 3	21
1.1.4.4 Sprint 4	22
1.2 Users and Stakeholders	23
1.2.1 Personas	24
1.2.2 User Stories	28
1.2.3 User acceptance criteria	29
1.2.3.1 Sprint 1 User acceptance criteria	30
1.2.3.2 Sprint 2 User acceptance criteria	31
1.3 Development Process	32
1.3.1 Branching strategy	33
1.3.2 Code Reviewing Strategy	34
1.3.3 Development Environment	35
1.3.4 Sprint2 Code Review	37
1.4 Sprints and Progress	40
1.4.1 Sprint1 plan	41
1.4.2 Sprint1 retrospect	42
1.4.3 Sprint2 plan	43
1.4.4 Sprint2 retrospect	44
1.4.5 Sprint3 plan	45
1.4.6 Sprint3 retrospect	46
1.5 Test process	47
1.5.1 Sprint1	48
1.5.2 Sprint2	49
1.6 Risk Assessment and Management	51
1.7 Deliverables and Handover	56
1.7.1 Sprint 2 process demo video	57
1.7.2 Products	58

Home

Project Background

23 May, 2023

Sprints and Progress

23 May, 2023

Development Process

30 Apr, 2023

Risk Assessment and I

30 Apr, 2023

Test process

30 Apr, 2023

Deliverables and Hand

30 Apr, 2023

Users and Stakeholder

29 Apr, 2023

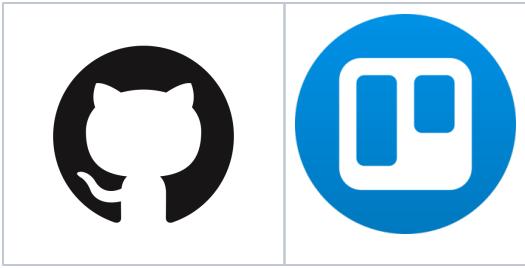
Meeting

29 Apr, 2023

- Project Background
- Users and Stakeholders
- Meeting
- Development Process
- Sprints and Progress
- Test process
- Risk Assessment and Management
- Deliverables and Handover

Github

Trello



Stakeholders

Name	Title	Contact Email	Photo
Wafa Johal	Client	wafa.johal@unimelb.edu.au	
Max Plumley	Supervisor	max.plumley@unimelb.edu.au	
Eduardo Oliveira	Instructor	eduardo.oliveira@unimelb.edu.au	

Team members

Team members	Title	Contact Email	Photo	Responsibility	Add-on
Chengfeng Zhang	Product Owner	Chengfeng@student.unimelb.edu.au	/	<ul style="list-style-type: none"> 1. Keep in touch with client and supervisor with our project progress. 2. Determine the project background, what is the scope of our product. 3. Compose the personas for intended users and come up with user stories. 4. Holding team meetings and record meeting minutes. 	/
Da Zhang	Scrum Master	dzzha1@student.unimelb.edu.au	/	<ul style="list-style-type: none"> 1. Organize and coordinate events such as Sprint planning, Sprint review, and Sprint retrospectives in an efficiently and orderly way. 2. Ensure that Scrum processes are conducted according to specifications and standards 3. Identify, track, and resolve obstacles that teams face which includes interpersonal, technical, resource, and management issues. 4. Responsible for collaboration and self-organization among team members to help the team grow and progress. 	/
Yuhang Wang	Team Developer	yuhang3@student.unimelb.edu.au	/	<ul style="list-style-type: none"> 1. Worked on python and ROS code update, maintenance. 2. ChatGPT deployment 3. Update readme, GitHub 4. Cooperate with other development team members, actively cooperate with scrum master and product owner to achieve customer satisfaction. 	/
Minyi Chen	Backend developer	minyic@student.unimelb.edu.au	/	<ul style="list-style-type: none"> 1. work on developing and maintaining databases, building APIs, implementing security measures, and integrating third-party services. 2. collaborate with other development team members and ensure that the application meets the desired functionality and user experience. 	/

Sijia Pei	Team Developer	sipei@student.unimelb.edu.au	/	<ol style="list-style-type: none"> 1. Update of codes involved in alphabetic interaction. 2. UI design and optimization. 3. Optimize chatGPT dialogue. 4. Maintenance and update of subsequent products. 5. Communicate closely with the members of the development team, regularly show the development progress to client and obtain feedback to ensure that the products meet requirements. 	/
-----------	----------------	------------------------------	---	---	---

Recent space activity



Chengfeng ZHANG

[Project Background](#) updated a minute ago • [view change](#)

[Sprints and Progress](#) updated about an hour ago • [view change](#)

[Sprint3 retrospect](#) created about an hour ago

[Sprint3 plan](#) created about an hour ago

[Development Process](#) updated 30 Apr, 2023 • [view change](#)

Space contributors

- [Chengfeng ZHANG](#) (a minute ago)
- [Da Zhang](#) (22 days ago)
- [Yuhang Wang](#) (23 days ago)
- [Minyi Chen](#) (57 days ago)
- [admin admin](#) (78 days ago)

Project Background

Project Overview

The project has two main objectives. Firstly, to **develop a ROS module that enable NAO robot use ChatGPT** in a versatile manner for different conversation contexts. Secondly, to **create educational writing activity based on the existing CoWriter project**, which involves the user teaching the robot how to write using the robot how to write using their demonstration to improve.

Our product is a unique and interactive learning tool that designed to help improve the handwriting specifically for children. The CoWriter robot will help child to improve their handwriting skills through teaching the robot the correct way to write. The CoWriter robot is set to be a bad writer at first, which encourage the child to play the role of a teacher and guide the robot to improve its writing.

Through a step by step process, the child will teach the robot to write letters, numbers and finally full words. The gradual progression not only enhance the robot's handwriting but also allows the child to refine their own handwriting as well.

We call this process **Learning by Teaching** since it enables children to reinforce their understanding of proper handwriting techniques while simultaneously helping the CoWriter to write in a beautiful way.

Project

The ChatGPT and Nao Robot project aims to enable social robots to use natural language processing capabilities provided by ChatGPT by developing a ROS module. This module will allow social robots to engage in more human-like conversations and to understand context.

The CoWriter software, which was originally programmed in Python 2.7 and has not been maintained in a few years. The project will involve reviewing the current project, updating it to Python 3. By rewriting the CoWriter into Python 3, the project can benefit from the ease of use and powerful features compared with Python 2.7.

Client Goals

The project has been divided into 3 Work Packages(WP) as following:

- WP1: **Update CoWriter project** with python 3 and ROS version compatible with NAO v6
- WP2: Integrate ChatGPT with the NAO robot for **autonomous dialogue, personalised selection of words, and autonomous robot motion during conversation**
- WP3: (optional) Implement a logging and annotation module using the ROS4HRI package

The client **requires WP1 and WP2 to be completed** for handover, and if time allows, we may implement WP3 by choice.

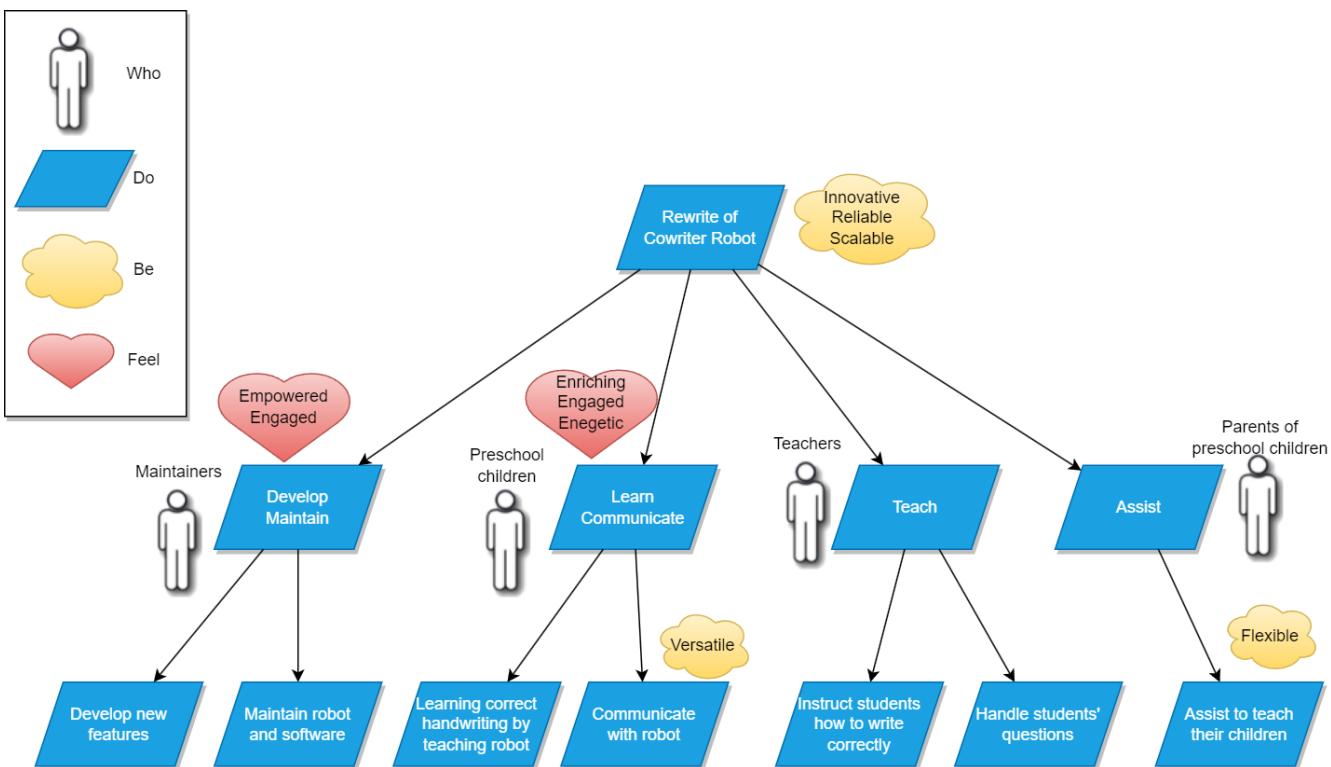
Work package is a term that used to describe a collection of works that groups together to define a specific part of a project.

Goal model

DO-BE-FEEL list

Do	Be	Feel	Who
Develop new features	Configurable	Empowered	Maintainer
Maintain software	Reliable	Empowered	Maintainer
Improve handwriting	Reliable	Innovative	Preschool children
Communicate with robot	Customizable	Enjoyable	Preschool children
Instruct students to write	Reliable	Handy	Teachers
Handle students' questions	Customizable	Relaxed	Teachers
Assist to teach	Versatile	Enriched	Parents of preschool children

Newest Version 24 Apr 2023



Scope

Requirement in scope

	Function	Description
1	Review, Analysis	<ol style="list-style-type: none">1. Code Review: Analyze the existing CoWriter codebase, its dependencies, and repository structure to identify improvements and areas that require updates.2. Dependency Analysis: Examine the project dependencies to ensure compatibility with the latest libraries and frameworks.
2	Rewrite	Rewrite the repository including nodes, shape learning algorithm, launch packages and interface packages.
3	Drawing Board	The user writes letters or numbers on the drawing board, and the board reads the coordinates of these points and passes them to the robot.
4	Robot Writing	By reading the node points passed from the drawing board, the program will analyse and fit in with the most similar letters or numbers to it and write it down on the screen.
5	Learning by Teaching	The robot will learn the handwriting from user in the process of repeating itself over and over again. The gradual progression not only enhance the robot's handwriting but also allows the user to refine their own handwriting as well.
6	Conversation with Robot	Integrate ChatGPT with the Nao robot to enable accurate speech comprehension and meaningful conversations, providing an interactive and user-friendly communication experience to the users.

Requirement out of scope

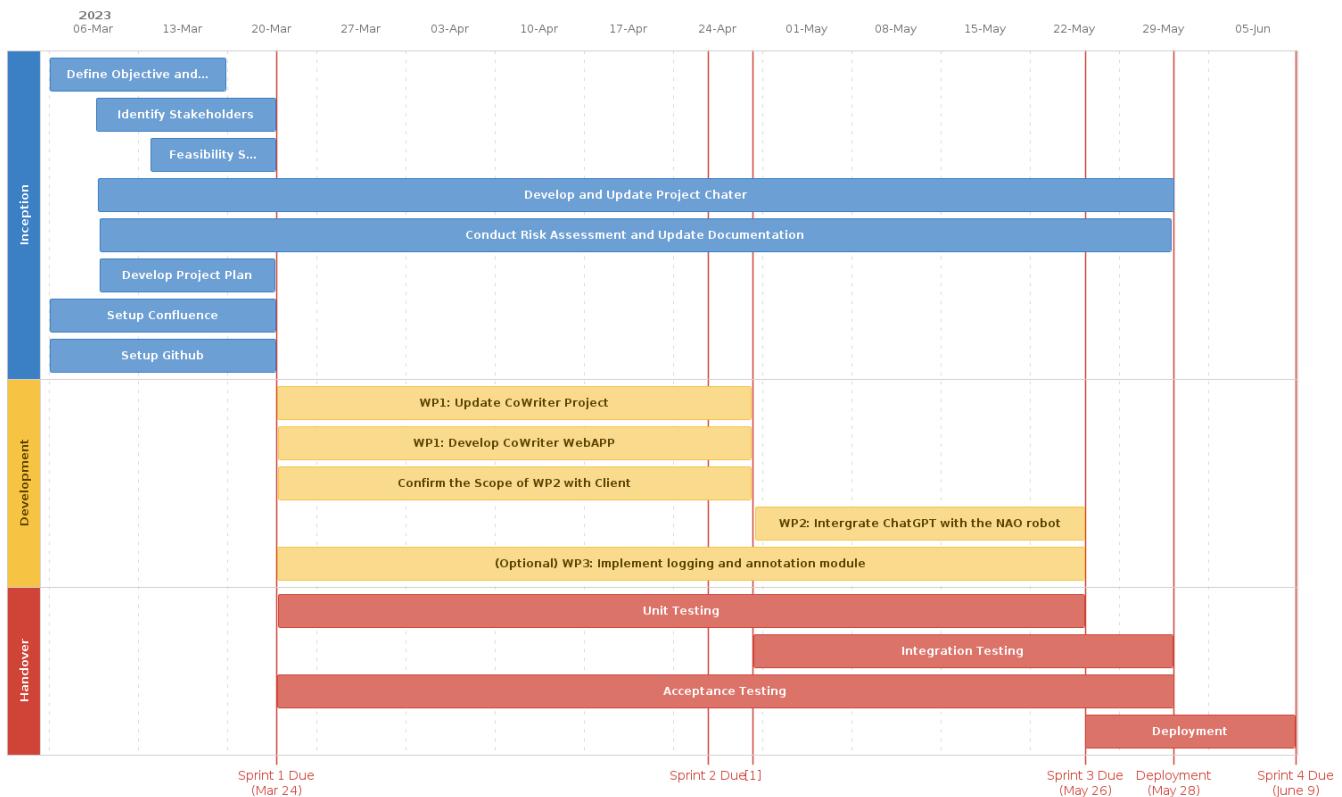
	Function	Description
1	Web Application	In addition to PC, developing a versatile web application that allows users to access and interact with the robot through multiple platforms, including Android, WeCom. By ensuring compatibility across a wide range of devices, users can seamlessly control and communicate with the robot, enhancing the overall experience and convenience.
2	Guide children in concentrating	Robot that integrated with ChatGPT can engage children through interactive, personalized conversations that capture their attention, while providing tailored guidance and encouragement to help them maintain focus on tasks or learning activities.
3	Data storage and Analysis	Collecting the data generated by users as Rosbags for analysis. It aims to develop another stand-alone webapp that will be used to analyse the data.

Project Plan

Plan

We were currently work on the work package 1 that client needs us to rewrite the CoWriter by Python 3. By completing the rewrite and implementation of web app, we are able to integrate the ChatGPT with the NAO robot in sprint 2. We are planning to debug and restrict the ChatGPT's behavior and tone to make it more childlike. In addition, we plan to perform comprehensive testing on the robot, including unit, integration and stress testing, to address the potential errors or vulnerabilities.

Project Timeline



[1]: Client will select the best port solution between Team RedBack, BoxJelly and BlueRing

Project Progress and Milestones

		Completed	<ul style="list-style-type: none"> • Project Charter (Initial Version) • Project Plan (Initial Version) • Risk Assessment (Initial Version) • Project Objective and Goal Confirmed
WP1 Completed	28 Apr 2023	On Track	<ul style="list-style-type: none"> • CoWriter software updated to adapt Python 3 and Ros 2 Humble • A working corresponding UI for the CoWriter software
Sprint 1 Review	30 Mar 2023	On Track	<ul style="list-style-type: none"> • Client approve our design solution • Client confirmed all the required functionalities are completed
(More to be Added)			

Status Key Note:

1. **On track**: The project is progressing according to plan and is on schedule to meet its milestones and deliverables.
2. **Behind schedule**: The project is not progressing as quickly as planned, and some milestones or deliverables may be delayed.
3. **Completed**: The project has been successfully completed, and all milestones and deliverables have been achieved.

Product Backlog

- Sprint 1
- Sprint 2
- Sprint 3
- Sprint 4

Sprint 1

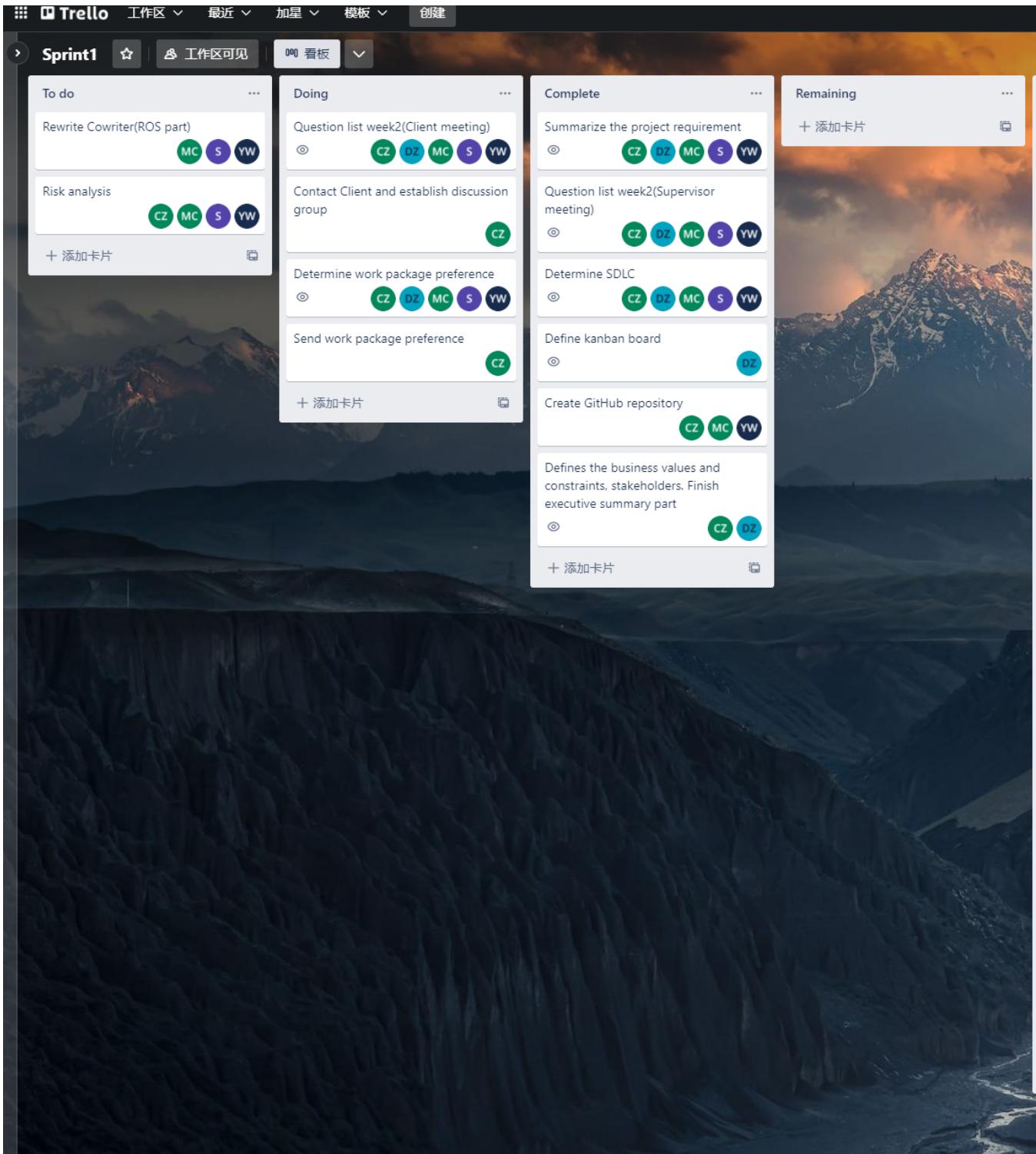
Sprint 1 Backlog

User Story ID	Description	Task	Story Point
12	As a client, I want the functionality to be modelled in the project so that it will be helpful to develop other functionalities in the future.	-Create a repository on GitHub -Define the structure of the project with the client	3
13	As a client, I want clear annotations to facilitate my quick understanding of the code and subsequent work.	-Analysis of the code provided by the Client. -Clear comments when rewriting code or writing new code	3
14	As a client, I want the README file to clearly show the project's structure so that it is convenient to maintain the code in the future.	-Maintain and update the README file	5
	As a client, I want the README file to clearly show how to install the related package needed in the project so that I can run the code.	-Maintain and update the README file when a new module or package is used in the development	
	As a client, I want the README file to show how to use and run the provided code.	-Maintain and update the README file.	
15	As a client, I want the source code provided to be rewritten in the latest Python version so that I can maintain the code more easily in the future.	-Determine the latest version of Python used in the project with the Client. -Analysis of the code provided by the Client. -Rewrite the source code provided by the Client, -Push the source code onto GitHub.	5
	As a client, I want the ROS module provided to be rewritten in the latest ROS version, ROS2, so that I can maintain the code more easily in the future.	-Determine the latest version of the ROS module used in the project with the Client. -Learning how to use the ROS module. -Analysis of the code provided by the Client. -Rewrite the source code provided by the Client, -Push the source code onto GitHub.	

Kanban board and burndown chart

<https://trello.com/b/lBVbLBjj/sprint1>

Week2



Week3

Trello 工作区 最近 加星 模板 创建

Sprint1 看板

To do

- Rewrite Cowriter(ROS part)
MC S YW
- Risk analysis
CZ MC S YW
- Rewrite Cowriter(Python part)
MC S YW

+ 添加卡片

Doing

- Determine work package preference
CZ DZ MC S YW
- Update Readme.md on Github
MC S YW
- Define the ROS module used in the project
CZ MC S YW
- Update Github structure
MC S YW
- Get to learn ROS module
MC S YW

+ 添加卡片

Complete

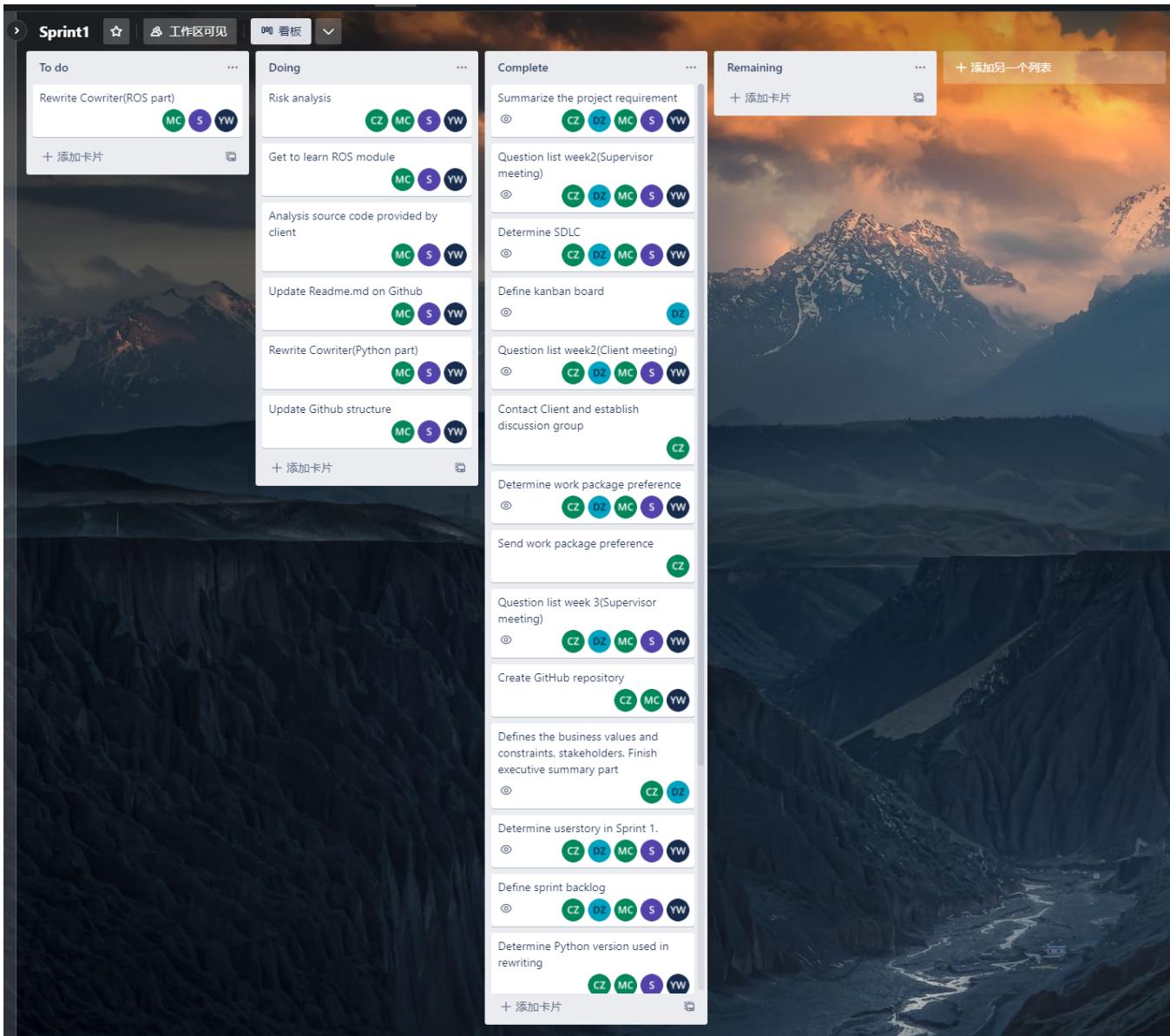
- Summarize the project requirement
CZ DZ MC S YW
- Question list week2(Supervisor meeting)
CZ DZ MC S YW
- Determine SDLC
CZ DZ MC S YW
- Define kanban board
CZ DZ
- Create GitHub repository
CZ MC YW
- Send work package preference
CZ
- Question list week2(Client meeting)
CZ DZ MC S YW
- Contact Client and establish discussion group
CZ
- Defines the business values and constraints, stakeholders. Finish executive summary part
CZ DZ
- Question list week 3(Supervisor meeting)
CZ DZ MC S YW
- Determine Python version used in rewriting
CZ MC S YW
- Deploy virtual machines and Ros module.
MC S YW
- Analysis source code provided by client
MC S YW

+ 添加卡片

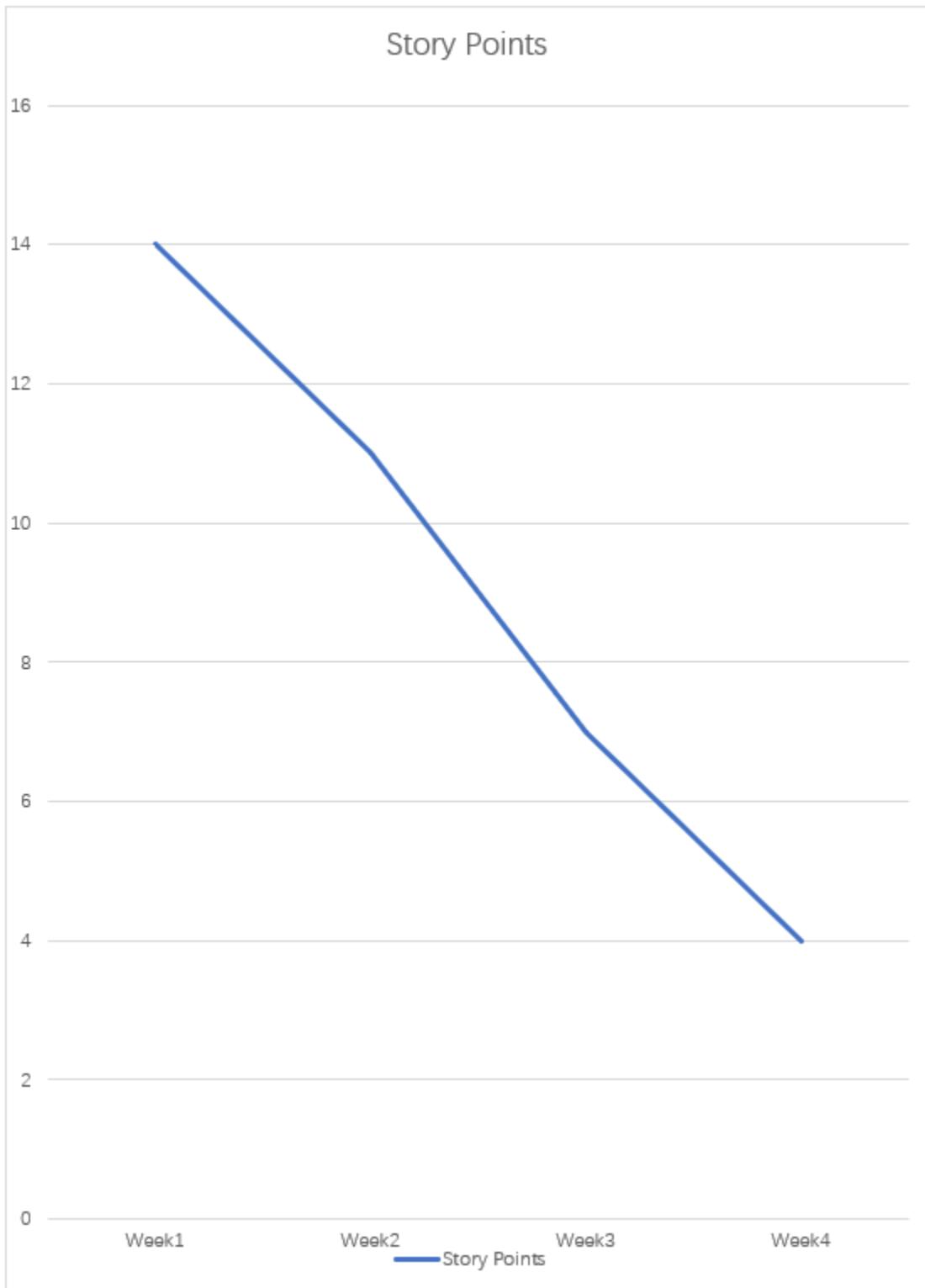
Remaining

- + 添加卡片

Week4



Sprint 1 burndown chart



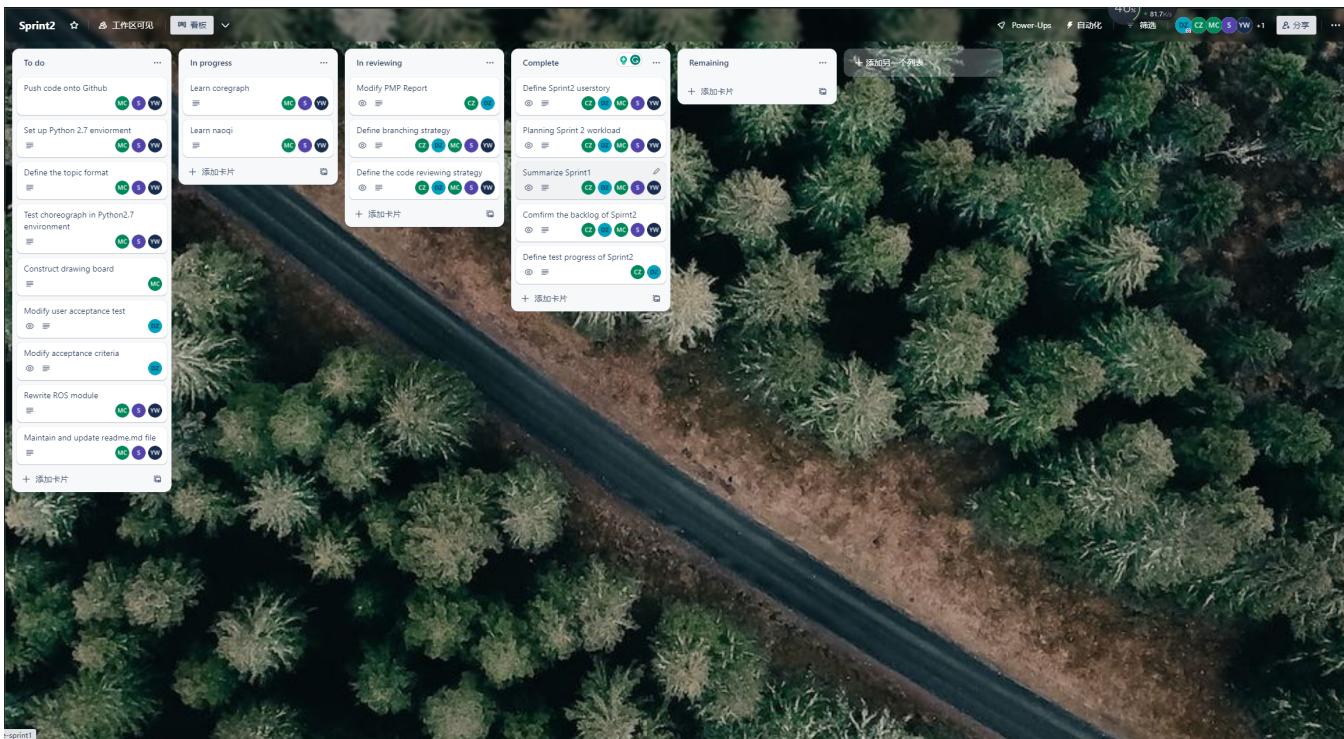
Sprint 2

Sprint 2 Backlog

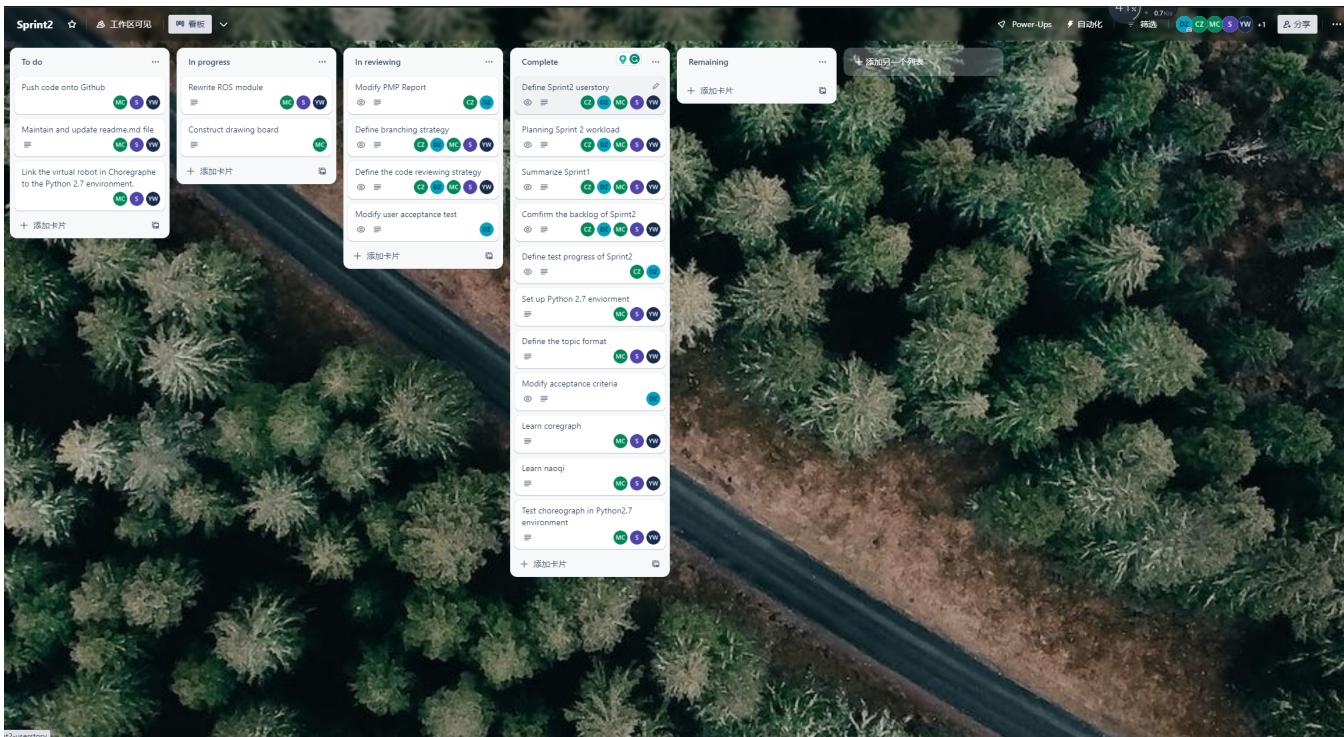
User Story ID	Description	Task	Story Point
3	As a preschool child, I want the robot can recognize the characters I write on the input device so that the robot can understand which characters I want to learn.	<ul style="list-style-type: none"> -Construct input node and backend transmission. -Construct a React-based drawing board. -Integrated a shape-learning algorithm with ROS2. -Update the dataset(the data file used to identify textual graphics). -Debug -Push the source code onto GitHub. -Maintain and update the README file. 	5
	As a preschool child, I want the robot to demonstrate correct writing so that I can learn how to improve my writing skills.	<ul style="list-style-type: none"> -Set up a Python 2.7 environment in Ubuntu. -Link the virtual robot in Choregraphe to the Python 2.7 environment. -Use the Naoqi SDK in a Python 2.7 environment. -Rewrite and configure the interface packages. -Test whether the virtual robot in Choregraphe can correctly receive and execute instructions. -Update the dataset(the data file used to generate textual graphics). -Debug -Push the source code onto GitHub. -Maintain and update the README file. 	
7	As a parent of a preschool child, I want to have a login system so that I can have a personal account to store the learning process.	<ul style="list-style-type: none"> -Design the user interface of the login system -Develop back-end authentication capabilities -Design and implement a user database for storing user information -Integrate front-end and back-end functions -Develop a personal account function -The development API is used to store and get user learning progress data -Test 	3
	As a parent of a preschool child, I want the robot to record and store my child's learning process.	<ul style="list-style-type: none"> -Choose the database used to store data -Create a unique storage space for each user -Integrate data acquisition functions into the robot -Test 	

10	As a maintainer, I want the message transferred between nodes to be efficient and reliable so that I can easily maintain the project.	<ul style="list-style-type: none"> -Rewrite the code of the nodes in the ROS module. -Define the format of the topics(which topic is a type of message format.) between the subscriber and publisher of the ROS module. -Implement the interface provided in <code>create_service</code> and <code>create_client</code>. -Debug -Push the source code onto GitHub. -Maintain and update the README file. 	3
16	As a maintainer, I want the source code provided to be rewritten in the latest Python version so that I can maintain the code more easily in the future.	<ul style="list-style-type: none"> -Determine the latest version of Python used in the project with the Client. -Analysis of the code provided by the Client. -Install sl and related libraries and files according to the readme provided in the client's repository -Record the running results before running each Python script. -Look for updates to the required packages. -Rewrite <code>shape_learning</code> file and <code>shape_modeler</code> file. -Update <code>shape_model_ui</code> file. -Debug -Push the source code onto GitHub. -Maintain and update the README file. 	5
	As a maintainer, I want the ROS module provided to be rewritten in the latest ROS version, ROS2, so that I can maintain the code more easily in the future.	<ul style="list-style-type: none"> -Determine the latest version of the ROS module used in the project with the Client. -Learning how to use the ROS module -Analyze the code provided by the Client. -Rewrite the <code>covriter_letter_learning</code> file. -Debug -Push the source code onto GitHub. -Maintain and update the README file. 	

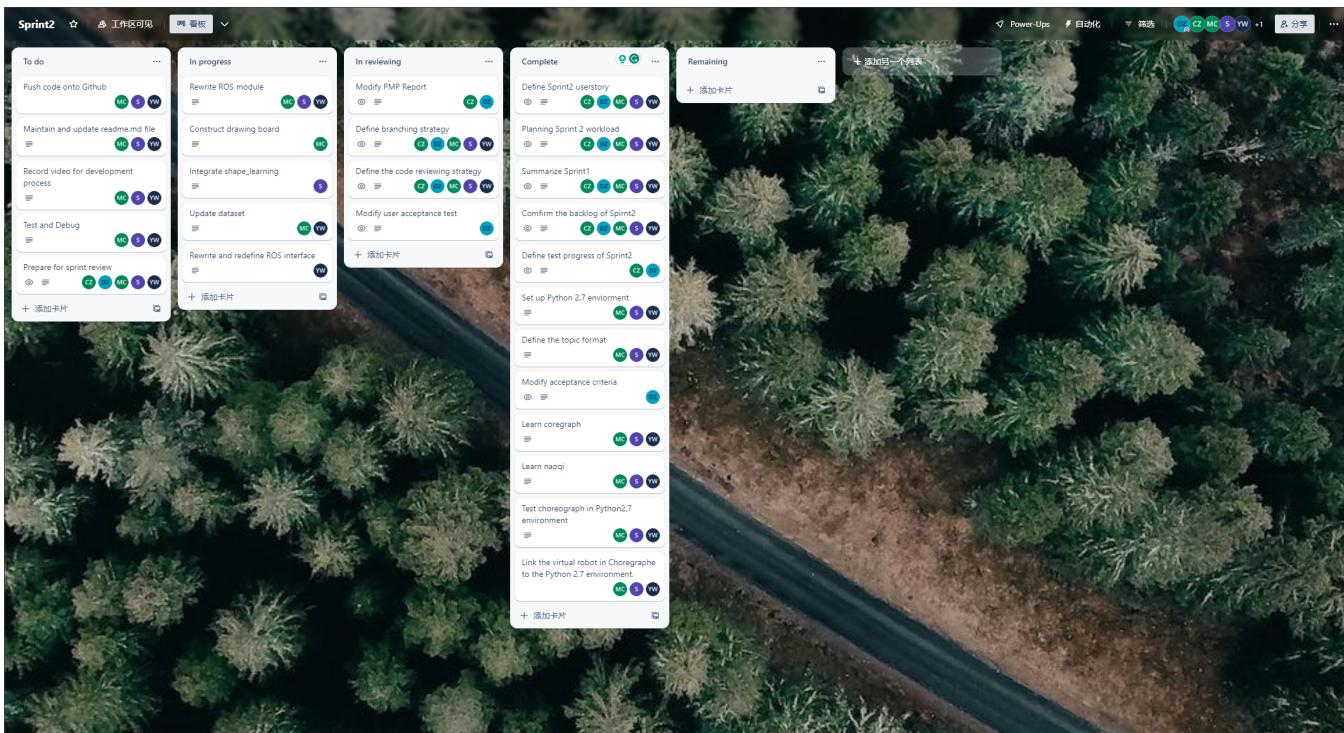
Sprint 2 week 1 Kanbanboard



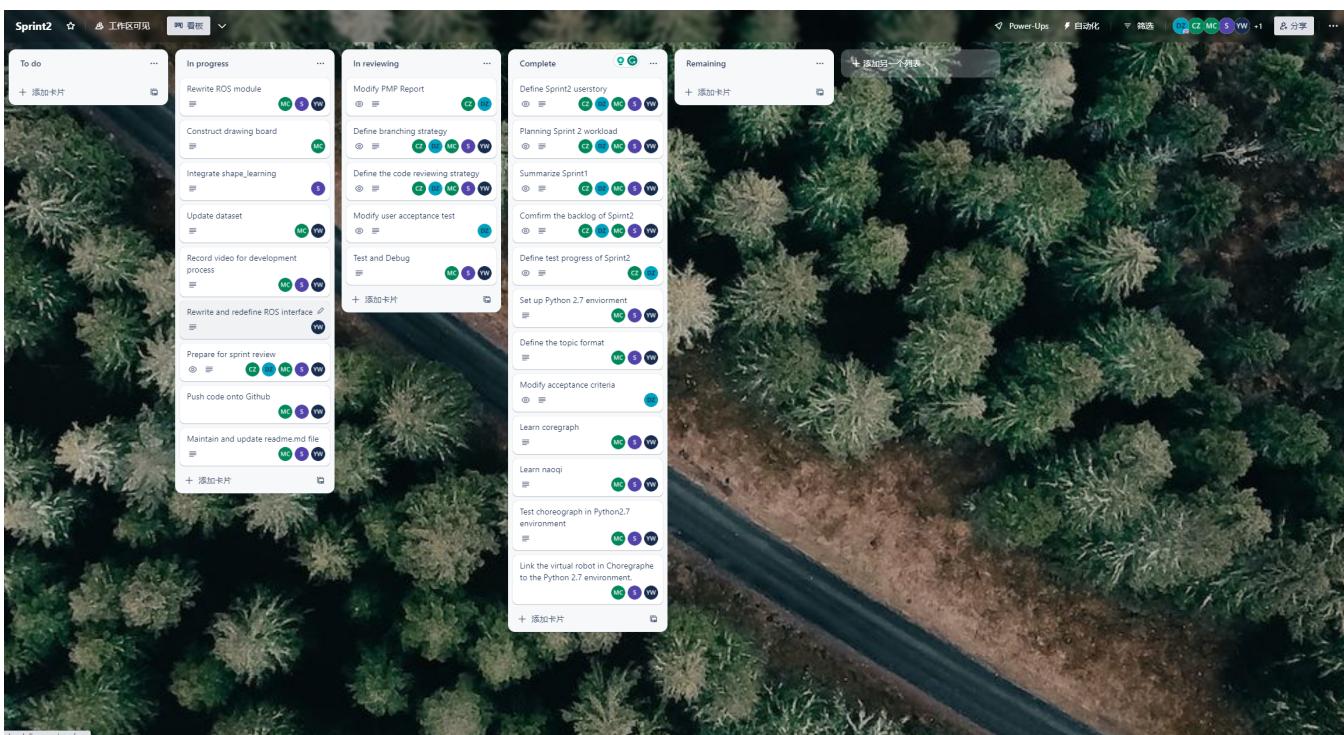
Sprint 2 week 2 Kanbanboard



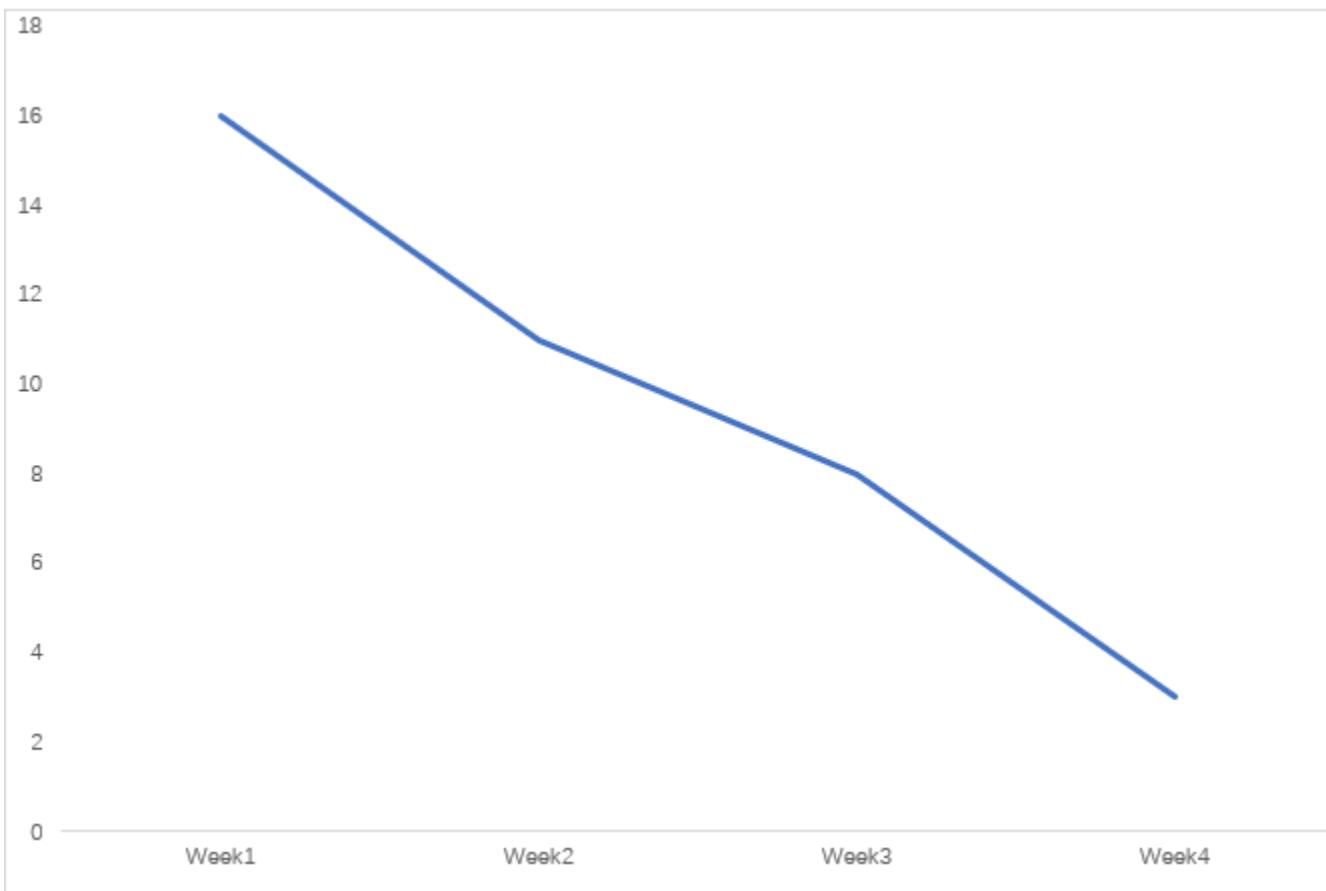
Sprint 2 week 3 Kanbanboard



Sprint 2 week 4 Kanbanboard



Burndown chart of Sprint 2



Sprint 3

Sprint 4

Users and Stakeholders

This chapter covers the intended users, user stories and user acceptance criteria based on these stories.

- [Personas](#)
- [User Stories](#)
- [User acceptance criteria](#)

Personas

Stakeholders

Name	Internal/External	Role	Power and Influences
Wafa Johal	Internal	Industry Partner (Client)	Client of the Project
NA-RedBack	Internal	Development Team	Project Team
Clark Andrew (fictional)	External	Preschool child	End User
Jill Anderson (fictional)	External	Teacher	End User
Sophia (fictional)	External	Parents of preschool child	Parent of users
Alex Thompson (fictional)	External	Maintainer	Maintenance

4 Fictional personas for our project

Clark Andrew - Preschool Children



Attributes:
Friendly, Clever, Shy
Demographics:
Age: 6
Work: Preschool child
Family: Single
Location: San Jose, CA
Character: Primary user

Motivation

Fear	<div style="width: 70%; background-color: #00AEEF;"></div>
Power	<div style="width: 5%; background-color: #00AEEF;"></div>
Social	<div style="width: 50%; background-color: #00AEEF;"></div>

Goals

- Improving his handwriting skills through teaching
- Getting accompanied by the CoWriter Robot
- Communications with CoWriter

Frustrations

- Struggling with having a conversation with his friends.
- Becoming frustrated when he knows his handwriting needs improving.
- Having difficulty finding the right words to convey his thoughts and emotions

Bio

Clark is a vibrant and enthusiastic ten-year-old who has captured the hearts of everyone around him. Born on a sunny day in April, Oliver has grown into a curious, kind, and creative preschooler, always eager to explore and learn new things. His wide, twinkly eyes and infectious laughter make it impossible not to fall in love with his cheerful spirit.

Personality

Introvert	<div style="width: 10%; background-color: #00AEEF;"></div>	Extrovert
Analytical	<div style="width: 5%; background-color: #00AEEF;"></div>	Creative
Loyal	<div style="width: 5%; background-color: #00AEEF;"></div>	Fickle
Passive	<div style="width: 10%; background-color: #00AEEF;"></div>	Active

Preferred Channels

Social Media	<div style="width: 50%; background-color: #00AEEF;"></div>
Mobile	<div style="width: 10%; background-color: #00AEEF;"></div>
Email	<div style="width: 10%; background-color: #00AEEF;"></div>
Traditional Ads	<div style="width: 90%; background-color: #00AEEF;"></div>

Influencers



Sophia Isabelle Martinez - Parent of Preschool Children



"I would like to let my sweetie Mia get accompanied."

Kind Caring Passionate Empathetic

Goals

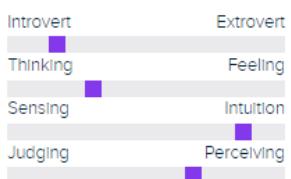
- Let her child Mia get accompanied
- Keep her work-life balance
- Improve Mia's handwriting skills

Frustrations

- Work-life balance: Juggling her responsibilities as an accountant and a devoted mother to Mia can be challenging at times. Sophia may sometimes struggle to find the perfect balance between her career and personal life.
- Mia's handwriting is relatively poorer than a normal child, but Sophia does not have enough time to spend with the child. She was trying to find a way to help Mia improve her handwriting skills.

Age: 32
Work: Accountant
Family: Married, one kid
Location: Melbourne, VIC
Character: Parent of user

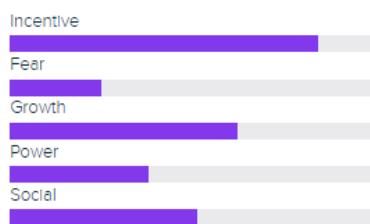
Personality



Bio

A devoted and caring mother to her delightful preschooler, Mia. As a company accountant, she is always busy in her daily life, trying to balance her work and life. But she can always manage to find time to spend with her. Her warm and nurturing presence encourages Mia's curiosity and imagination to soar.

Motivation



Brands & Influencers



Preferred Channels



Jill Anderson - Educator



"I'm looking for a tool that helps my students to be educated."

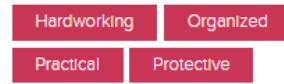
Age: 39

Work: Educator

Family: Married

Location: New York, NY

Character: Teacher



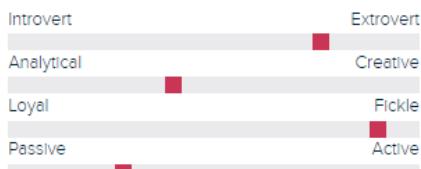
Brands & Influencers



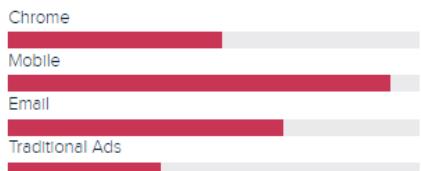
Bio

Lily is a dynamic and innovative educator with a passion for integrating technology into the classroom. Lily is dedicated to inspiring her students to excel in writing by incorporating the NAO robot as a unique teaching tool. Her enthusiasm and expertise in merging cutting-edge robotics with traditional teaching methods create a stimulating learning environment. By leveraging the interactive and engaging capabilities of the NAO robot, Lily motivates her students to hone their writing skills, fostering their creativity and self-expression. Her dedication to enhancing education through technology makes Lily a trailblazer in the world of modern teaching.

Personality



Preferred Channels



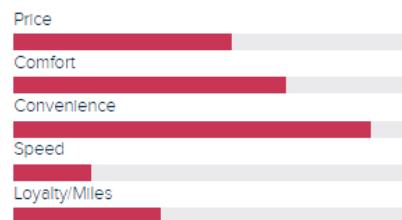
Goals

- Using the robot's assistance, Jill aims to develop students' fine handwriting skills and penmanship through personalized feedback and tailored activities.
- Jill plans to use the robot with ChatGPT integration for interactive language learning, providing customized conversation practice and grammar exercises to support students' development.

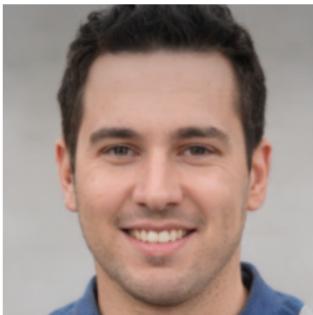
Frustrations

- Jill struggles to provide personalized guidance to each student, with a large number of students in her class.
- She finds it difficult to maintain students' motivation and engagement in handwriting and language exercises, as some students may not find traditional teaching methods stimulating enough.

Motivation



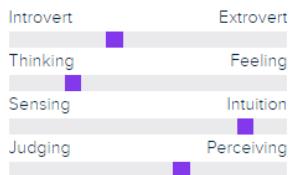
Alex Thompson - Maintainer



"I wanna try my best to give better user experience to my customer"

Age: **29**
Work: **Software Engineer**
Family: **Single**
Location: **Sydney, NSW**
Character: **Maintainer**

Personality



Engaging Innovative Adaptive

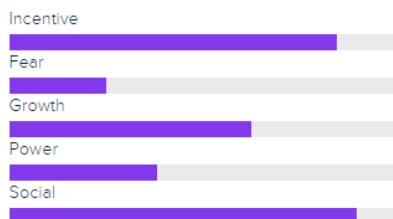
Bio

Alex Thompson is a skilled robotics engineer specializing in educational robot maintenance and optimization. With a degree in Electrical Engineering and Robotics, Alex has a decade of experience in the industry, focusing on robotic teaching assistants. As the lead maintainer of the handwriting improvement robot integrated with ChatGPT, Alex ensures smooth operation and continuous system improvement. Passionate about enhancing the learning experience, Alex also shares their knowledge as a guest lecturer and mentor, inspiring future engineers in the education sector.

Goals

- Aim to improve reliability to minimize downtime
- Keep the system up-to-date
- Provide a good user experience

Motivation



Frustrations

- Struggles with the robot's disorganized codebase and insufficient documentation, leading to increased time and effort in debugging, updating, and maintaining the system.
- Staying current with the latest technological advancements and integrating them into the robot's system can be demanding, as Alex needs to balance innovation with stability and reliability.

Brands & Influencers



Preferred Channels



Powered by [Xtensio](#)

User Stories

Epic (Features)	ID	Owner	User Stories	Story Points (1-5)	MosCoW Priority	Status	ChatGPT Generated
ChatGPT deployment platform	1	Parents of preschool children/ Educator	As a parent of preschool children/educator, I want to control the NAO robot through a web app so that I can access it on different platforms.	5	Must have	Assigned	No
Nao robot functionalities	2	Preschool children/ Parents of preschool children	As a preschool child /parent of preschool, I would like to ask random questions to an NAO robot rather than improve my handwriting so that it makes me feel enjoyable.	5	Must have	Assigned	No
	3	Preschool child	As a preschool child, I would like to improve my handwriting so that I can write letters or words fluently as the former version.	5	Must have	Assigned	No
	4	Educator	As an educator, I want to use the NAO robot to help me handle the question asked by my students so that I can have spare time to spend with other students.	3	Should have	Assigned	No
	5	Educator	As an educator, I want to use the NAO robot to guide distracted students to concentrate on handwriting so that they will not be disturbed by other students.	3	Should have	Assigned	No
	6	Individuals with Disability	As an individual with a disability, I want the social robot to write according to my voice command to help me with writing during daily life.	1	Could have	Assigned	No
	7	Parents of preschool child	As a parent of a preschool child, I want the robot to view my kids' handwriting so that it can correct and improve their handwriting.	3	Should have	Assigned	No
	8	Parents of preschool children/ Educator	As a parent of preschool children/educator, I want to integrate ChatGPT with the NAO bot to improve the robot's writing and comprehension capability.	3	Should have	Assigned	No
	9	Parents of preschool children/ Educator	As a parent of preschool children/Educator, I want to log in to the web with my personal account so that the robot can save my personalized content.	3	Should have	Assigned	No
	10	Future Maintainer	As a future maintainer, I want to have the functional modules of the Nao robot independent of each other so that I can add new functions based on its existing functional modules.	3	Must have	Assigned	No
ROS2 and Python enabled Port and APIs	11	Future Maintainer	As a future maintainer, I would like to use the ROS APIs developed from WP1 and be guided by its documentation so that the development of WP2 will be easier.	1	Could have	Assigned	No
	12	Future Maintainer	As a future maintainer, I would like to make requests through HTTP protocols to use the APIs remotely.	1	Could have	Assigned	No
	13	Future Maintainer	As a future maintainer, I want the port to have a tidy structure so that it will be helpful to develop other functionalities in the future.	3	Should have	Assigned	No
Maintenance and follow-up project development	14	Future Maintainer	As a future maintainer, I want clear annotations to facilitate my quick understanding of the code and subsequent work.	3	Must have	Assigned	No
	15	Future Maintainer	As a future maintainer, I want to have a clear readme that clearly explains the introduction and operation of the project, which will be helpful for future handover with other developers.	5	Must have	Assigned	No
	16	Future Maintainer	As a future maintainer, I want to be able to use the latest version so that I can maintain the code more easily.	5	Must have	Assigned	No

User acceptance criteria

- Sprint 1 User acceptance criteria
- Sprint 2 User acceptance criteria

Sprint 1 User acceptance criteria

User story ID	Acceptance Criteria
13	-The code structure should be easy to understand and navigate.
	-Code should be properly commented on and documented to clearly explain each module, function, and variable.
	-It should be easy to add new features to your code without introducing errors or breaking existing code.
	-Any problems or suggestions for improvement related to the code structure are resolved or documented.
14	-Verify that comments are clear and clear to help understand the logic and purpose of the code.
	-Ask the development team what they think of code comments and if they think they are helpful for understanding code and developing new features.
	-Record any problems or suggestions for improvement related to code comments.
15	-The README should be clear and explain the purpose, function, and operation of the project.
	-The README should contain the necessary information, such as project name, version number, dependencies, etc.
	-The operation instructions in the README should be clear and clear to help users complete the project operation.
	-The development team should confirm with the customer that the README is very helpful for the transition.
	-Any problems or suggestions for improvement related to the README are clearly documented.
16	-The code in the project, both client-supplied code and code developed by the development team, has been written in the latest version of the programming language as required.
	-When code is upgraded to a higher version of a programming language, the difficulty and impact of the upgrade should be minimized.
	-The development team should confirm that using the latest version of the programming language improves code quality and maintainability.
	-Any problems or suggestions for improvements related to the way the code was written are clearly documented.

Sprint 2 User acceptance criteria

User story ID	Acceptance Criteria
3	<ul style="list-style-type: none"> -Design a set of structured handwriting practice courses according to the kindergarten children's learning ability and age characteristics of adjustment. -Provide progressively more difficult practice materials, including basic strokes, letters, numbers, words, and short sentences. -Various teaching methods, such as video tutorials, animations, and physical demonstrations, are incorporated to suit different children's learning needs and preferences. -Design evaluation and feedback mechanisms so parents and educators can continuously track their child's handwriting progress and adjust and improve if necessary. -Provide handwriting exercises in conjunction with other learning activities, such as reading, jigsaw puzzles, etc., to help children improve their handwriting and overall skills. -Design interactive and cooperative learning activities that allow children to build teamwork and social skills as they learn to write. -Ensure that local educational standards and policy requirements implement the curriculum.
7	<ul style="list-style-type: none"> -The robot should provide clear and concise feedback to the child regarding their handwriting errors and suggest improvements. -The robot should track the child's progress over time and adapt its feedback based on their improvement. -Parents could access progress reports and review the robot's suggestions for improvement. -The robot's guidance should comply with local educational standards and policies.
10	<ul style="list-style-type: none"> -The robot should be able to capture images or videos of the child's handwriting in real time. -The robot should store the recorded handwriting samples in an organized manner, including dates and other relevant metadata. -The robot should provide a user-friendly interface for parents to access and review the recorded handwriting samples. -The robot should ensure the privacy and security of the recorded handwriting samples, allowing access only to authorized users. -The robot should allow parents to compare handwriting samples from different dates to visualize the improvement process. -The robot should allow parents to export the handwriting samples for further analysis or sharing.
16	<ul style="list-style-type: none"> -The project code, both client-supplied and developed by the development team, has been written in the latest version of the programming language as required. -When code is upgraded to a higher version of a programming language, the difficulty and impact of the upgrade should be minimized. -The development team should confirm that using the latest version of the programming language improves code quality and maintainability. -Any problems or suggestions for improvements related to how the code was written are documented.

Development Process

This section of our management plan covers the branching strategy, code reviewing strategy, development environment, and Sprint2 code review. These components help streamline the development process, maintain high-quality code, and foster a collaborative working environment, ensuring efficient delivery of our software projects.

- [Branching strategy](#)
- [Code Reviewing Strategy](#)
- [Development Environment](#)
- [Sprint2 Code Review](#)

Branching strategy

Industry branching strategy

We are using an industry branching strategy(GitFlow) in this project. Git Flow is a workflow model for version control using Git that defines a strict set of branch and merge policies, as well as the lifecycle of how branches should be handled and managed at different stages. The core idea of Git Flow is to separate the development and distribution of your code so that you can better control the quality and stability of your code. The main branches of Git Flow include two long-running branches: the master branch and the develop branch. The master branch is used to store stable, released versions of the code, and the develop branch is used to store the latest versions in development.

- **Master branch:** Used to store stable versions that have been released.
- **Develop branch:** Store the latest version in development.
- **Feature branch:** used to support the development of new features;
- **Release branch:** used for testing and fixing the release before release.
- **Hotfix branch:** Used for emergency bug fixes after publication.

Code Reviewing Strategy

- All code must be reviewed by at least one colleague to ensure code quality, readability, and maintainability.
- The code reviewer should be a colleague who is familiar with the code in question and, preferably, has domain knowledge.
- The code review process should be completed before the code is committed so that problems can be detected early and changes can be made.
- The code reviewer should document problems found in the code, such as potential bugs, performance issues, or readability issues, and provide suggestions for improvement.
- Code review should be a combination of tools and people. Tools can help find common errors and potential problems, while human reviews can provide deeper analysis and insight.
- The code review process should focus on consistency and standardization of code style, such as naming, indentation, comments, and formatting.
- There should be effective communication and feedback between code reviewers and code submitters to ensure that the effectiveness of code reviews is maximized.

Test:

For testing, we will use the Automated tests provided by ros.org(<http://wiki.ros.org/Quality/Tutorials/UnitTesting>). Automated tests can make development easier and more efficient by allowing for incremental updates, confident refactoring, better-designed code, prevention of recurring bugs, and easier collaboration between developers. Developers should use language-specific testing frameworks like gtest (C++) and unittest (Python) and integrate tests into the project scripts to run them alongside the build and test infrastructure. Testing should be done at three levels: library unit test, ROS node unit test, and ROS node integration/regression test.

1. Testing Tools

The main testing tools in ROS include Python's unittest framework for testing Python code at the library level, Google test framework gtest for testing C++ code at the library level, and rostest for testing at the ROS node level, which involves ROS as a communication middleware. Tests should be integrated into project scripts and run by the build and test infrastructure using catkin/roslaunch integration. Test nodes should be introduced using the tag in launch files, and rostest will interpret these tags to start nodes responsible for running node-level tests.

2. Levels of testing

Level 1. Library unit test (unittest, gtest): a library unit test should test your code sans ROS (if you have a hard time testing sans ROS, you probably need to refactor your library). Make sure that your underlying functions are separated out and well-tested with both common, edge, and error inputs. Make sure that the pre-and post- conditions of your methods are well-tested. Also, make sure that the behaviour of your code under these various conditions is well documented.

Level 2. ROS node unit test (rostest + unittest/gtest): node unit tests start up your node and test its external API, i.e. published topics, subscribed topics, and services.

Level 3. ROS nodes integration/regression test (rostest + unittest/gtest): integration tests start up multiple nodes and test that they all work together as expected. Debugging a collection of ROS nodes is like debugging multi-threaded code: it can deadlock, there can be race conditions, etc... An integration test is often the best way of uncovering the bugs.

Development Environment

According to customer requirements, we need to use ROS2 humble and python3 for refactoring the existing repository and development. There are good reasons to support this decision:

The reasons for ROS:

Robot Operating System (ROS) is a flexible, open-source framework for developing robotics software. Currently, there are two major versions of ROS1 and ROS2. ROS2 (Robot Operating System 2) is the second generation of Robot Operating System, which is an open-source framework for building robot software. Nowadays the most popular version of ROS2 is ROS2 humble

Advantages of ROS2 over ROS1:

ROS2 offers several advantages over its predecessor, ROS1, including:

1. Improved Real-time Performance: ROS2 has a more modular architecture that allows better real-time performance and improved handling of time-critical operations.
2. Better Security: ROS2 offers better security features compared to ROS1. It has added features such as encryption, authentication, and access control to ensure that robots are secure and protected from cyber threats.
3. Cross-Platform Compatibility: ROS2 can run on different platforms, including Linux, Windows, and macOS. This makes it easier to develop, test, and deploy robot applications across multiple platforms.
4. Modularity and Scalability: ROS2 is designed to be more modular, which makes it easier to develop, test, and deploy complex robotic systems. It also allows for better scalability and reusability of software components.
5. Easier Integration with Other Tools: ROS2 has a more streamlined build system and is designed to work with other development tools such as Git and CMake. This makes it easier to integrate with other software tools and frameworks.
6. Besides, this version is currently supported for the longest maintenance.

The reason for Python:

1. Consistency and efficiency: Since the existing repository code is made up of python2, for consistency and efficiency we will stick with python and use the new version of Python 3.10.

2. Comparison between python and Java/C

2.1 Rich libraries and modules: Python has a large number of libraries and modules covering various domains, such as data processing, computer vision, and machine learning. These libraries and modules help developers implement functionalities quickly and reduce development time.

2.2 High code readability: Python's code has high readability, making it easy to understand and maintain. For a team project, code readability is crucial for team collaboration and code maintenance.

2.3 Ease of use: Python has a simple and clear syntax, which makes it easier for beginners to learn and understand. This allows developers to write and debug code more quickly compared to Java and C.

2.4 Rich libraries and modules: Python boasts a vast number of libraries and modules covering various domains, such as data processing, computer vision, and machine learning. These libraries and modules help developers implement functionalities quickly, reducing development time.

2.5 Integration with ROS2: While ROS2 does offer support for Java and C, the support and available resources for Python are more extensive, making it easier to integrate and work with ROS2 for robotic applications.

Environment setup instructions:

Python3.10 setup:

Execute the following command in the command prompt based on your personal system:

Windows:

Visit the official Python website's download page at <https://www.python.org/downloads/>

Download the Windows installer for Python 3.10 by clicking on the appropriate link (either 32-bit or 64-bit, depending on your system).

Run the installer and follow the on-screen instructions.

Make sure to check the box that says "Add Python 3.10 to PATH" during installation. This will make it easier to run Python from the command line.

Complete the installation process.

macOS:

Visit the official Python website's download page at <https://www.python.org/downloads/>

Download the macOS installer for Python 3.10.

Open the downloaded package and follow the on-screen instructions to complete the installation process.

Linux (Ubuntu, Debian):

Python 3.10 may not be available in the default repositories for some Linux distributions. In that case, you can use the "deadsnakes" PPA repository to install it. Run the following commands in the terminal:

```
sudo apt update  
sudo apt install software-properties-common  
sudo add-apt-repository ppa:deadsnakes/ppa  
sudo apt update  
sudo apt install python3.10
```

After completing the installation, you can check the Python version by running `python3.10 --version` in the command line or terminal.

ROS2 setup

Execute the following command in the command prompt based on your personal system:

Ubuntu Linux:

```
sudo apt update sudo apt upgrade  
sudo locale-gen en_US en_US.UTF-8 sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8 export LANG=en_US.UTF-8  
sudo apt install curl gnupg2 lsb-release curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -  
sudo sh -c 'echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" > /etc/apt/sources.list.d/ros2.list'  
sudo apt update  
sudo apt install ros-humble-desktop  
echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc source ~/.bashrc
```

macOS and Windows:

ROS2 is primarily targeted at Ubuntu Linux, but there are experimental installations available for macOS and Windows. The official ROS2 installation documentation provides instructions for these platforms, but keep in mind that support may be limited, and not all features and packages may be available. For the most up-to-date instructions, refer to the official ROS2 documentation:

macOS: <https://index.ros.org/doc/ros2/Installation/OSX-Development-Setup/>

Windows: <https://index.ros.org/doc/ros2/Installation/Windows-Development-Setup/>

Remember to always check the official ROS2 documentation for the most accurate and up-to-date installation instructions.

Sprint2 Code Review

Date:

29/4/2023

Participants:

Minyi Chen, Sijia Pei, Yuhang Wang

ChatGPT prompt illustration:

Sprint2's code review is done in the prompt manual way with ChatGPT. All the critical/core code is inspected and evaluated in the following prompt to ChatGPT:

[Here is the code]

Please evaluate the code above:

Use the following checklist to guide your analysis:

1. Documentation Defects:
 - a. Naming: Assess the quality of software element names.
 - b. Comment: Analyze the quality and accuracy of code comments.
2. Visual Representation Defects:
 - a. Bracket Usage: Identify any issues with incorrect or missing brackets.
 - b. Indentation: Check for incorrect indentation that affects readability.
 - c. Long Line: Point out any long code statements that hinder readability.
3. Structure Defects:
 - a. Dead Code: Find any code statements that serve no meaningful purpose.
 - b. Duplication: Identify duplicate code statements that can be refactored.
4. New Functionality:
 - a. Use Standard Method: Determine if a standardized approach should be used for single-purpose code statements.
5. Resource Defects:
 - a. Variable Initialization: Identify variables that are uninitialized or incorrectly initialized.
 - b. Memory Management: Evaluate the program's memory usage and management.
6. Check Defects:
 - a. Check User Input: Analyze the validity of user input and its handling.
7. Interface Defects:
 - a. Parameter: Detect incorrect or missing parameters when calling functions or libraries.
8. Logic Defects:
 - a. Compute: Identify incorrect logic during system execution.
 - b. Performance: Evaluate the efficiency of the algorithm used.

Provide your feedback in a numbered list for each category. At the end of your answer, summarize the recommended changes to improve the quality of the code provided.

Issues identified, ChatGPT feedback and comments on feedback:

I tem	Artifact (on GitHub)	Location (where the issue was found in the reviewed artifact?)	S e v er ity	Ty pe	Defects Category	Description	Fixed by the autho r?	Is the evaluation suggestion of ChatGPT helpful?
1	branch main/src /nao_drawing_board/src /components	useEffect, comment at line 72-73	Trivial	Imprudent	Documentation Defects (Comment)	Missing comment on the useEffect block at line 23-29. This block handles the canvas resizing. The comment at line 72-73 might be improved by mentioning that it redraws the line on the canvas while the user is painting.	No	Yes
2	branch main/src /nao_drawing_board/src /components	shape_type	Medium	Imprudent	Check Defects (Check User Input)	User input is taken for "shape_type" through an input field, but there is no validation or sanitization in place.	No	Yes
3	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /input_interpreter.py	general	Medium	Imprudent	Documentation Defects (Comment)	There are very few comments in the code. More comments should be added to provide better documentation.	No	Yes

4	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /input_interpreter.py	s in the process_user_input function (line 78)	Trivial	Improvement	Documentation on Defects (Naming)	Variable name s in the process_user_input function (line 78) can be more descriptive, such as stroke_data.	No	Yes
5	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /input_interpreter.py	Line 29	Trivial	Improvement	Visual Representation on Defects (Indentation)	Line 29 has an extra indentation, which should be removed for consistency.	No	Yes
6	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /input_interpreter.py	send_strokes function (line 52-59)	Medium	Investigate	Check Defects (Check User Input)	The send_strokes function (line 52-59) processes user input without validating it.	No	Yes
7	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /learning_word.py	init_datasetDirectory	Trivial	Improvement	Visual Representation on Defects (Long Line)	Break long lines into multiple lines for better readability, such as the line starting with init_datasetDirectory in the generateSettings function.	No	Yes
8	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /learning_word.py	show_shape	Medium	Investigate	Documentation on Defects (Comment)	Add more comments in the show_shape method to explain the purpose of different lines of code.	No	Yes
9	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /nao_writer_naoki.py	lines 4-10	Trivial	Investigate	Structure Defects (Dead Code)	There are commented-out lines of code at the beginning that should be removed (lines 4-10).	No	Yes
10	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_learner.py	lines containing "USE ONLY FIRST PARAM FOR SELF-LEARNING ALGORITHM ATM".	Medium	Investigate	Visual Representation on Defects (Long Line)	Consider breaking these lines for better readability.	No	No
11	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_learner.py	general	Medium	Investigate	Check Defects (Check User Input)	No explicit user input handling is present in the code. If user input is required, consider adding input validation and error handling.	No	Yes
12	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_learner.py	line 14	Trivial	Improvement	Documentation on Defects (Comment)	The comment on line 14 is a little vague ("# Handling file"). Consider providing more context or detail.	No	Yes
13	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_learner_manager.py	general	Medium	Improvement	Documentation on Defects (Comment)	The code lacks comments. It would be beneficial to add comments explaining the purpose of each code block, function, or class.	No	Yes
14	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_learner_manager.py	line 447-586	Medium	Improvement	Structure Defects (Dead Code)	Many lines of code are commented, should be removed	No	Yes
15	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_learner_manager.py	general	Medium	Investigate	Check Defects (Check User Input)	No explicit user input handling is present in the code. If user input is required, consider adding input validation and error handling.	No	Yes
16	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_modeler.py	general	Medium	Improvement	Documentation on Defects (Comment)	There are some comments in the code, but they could be improved by providing more context and explanation for certain functions and methods.	No	Yes
17	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_modeler.py	general	Trivial	Investigate	Visual Representation on Defects (Long Line)	There are a few lines that are longer than the recommended maximum of 79 characters, but this does not significantly hinder readability.	No	Yes
18	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_modeler.py	normaliseShapeWidth()	Medium	Improvement	Structure Defects (Duplication)	There is some duplication of code in the normaliseShapeWidth() method, which could be refactored to improve code maintainability.	No	No
19	branch main/src /nao_ros2_ws/src /nao_writing/nao_writing /shape_modeler.py	general	Medium	Investigate	Check Defects (Check User Input)	There is some error handling for file paths, but other user input is not extensively validated.	No	No

Code review's number summary:

Number of severe/critical errors:	0
Number of medium errors:	12
Number of trivial errors:	7

Number of helpful evaluation suggestion from ChatGPT	16
Number of useless evaluation suggestion from ChatGPT	3
Ratio of helpful evaluation suggestion from ChatGPT	84.21%

Sprints and Progress

This section captures plans and retrospectives for each Sprint, offering insights into project progress and promoting continuous improvement in execution and team performance.

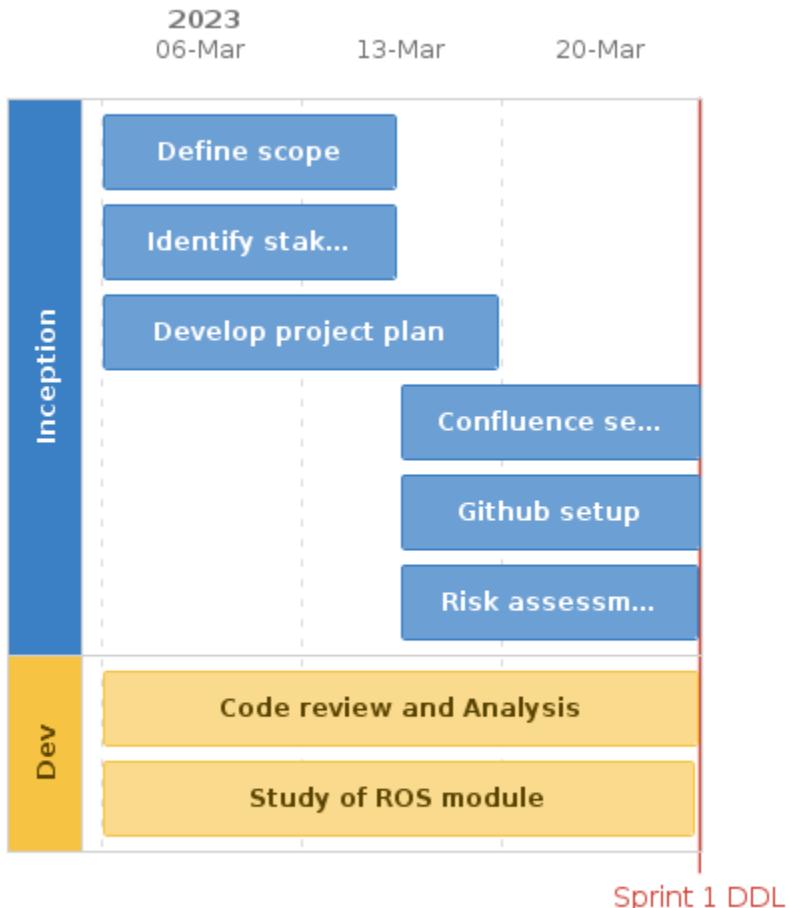
- [Sprint1 plan](#)
- [Sprint1 retrospect](#)
- [Sprint2 plan](#)
- [Sprint2 retrospect](#)
- [Sprint3 plan](#)
- [Sprint3 retrospect](#)

Sprint1 plan

In the first sprint, our team focused on defining the scope, identifying the users and stakeholders, and Developing our project plans.

After finishing the work above, we start to set up our working space like Github, Confluence.

As soon as we get the code repository from our client, our developers start to review and analyse the dependency of the code.



Sprint1 retrospect

Sprint reflection on the completed sprint:

Accomplishment:

- Define the scope of our project.
- Identify intended user groups, create goal model for them.
- Basic confluence set up including [project background](#), [goal model](#), [scope](#), [project plan](#) and etc..
- Setup of Github repository.
- Study of ROS module

Challenges:

- Setup up the confluence page.
- create user stories for intended users.
- Split user stories into each sprint.

Not finished:

- Code review and analysis of previous code repository.

Team performance:

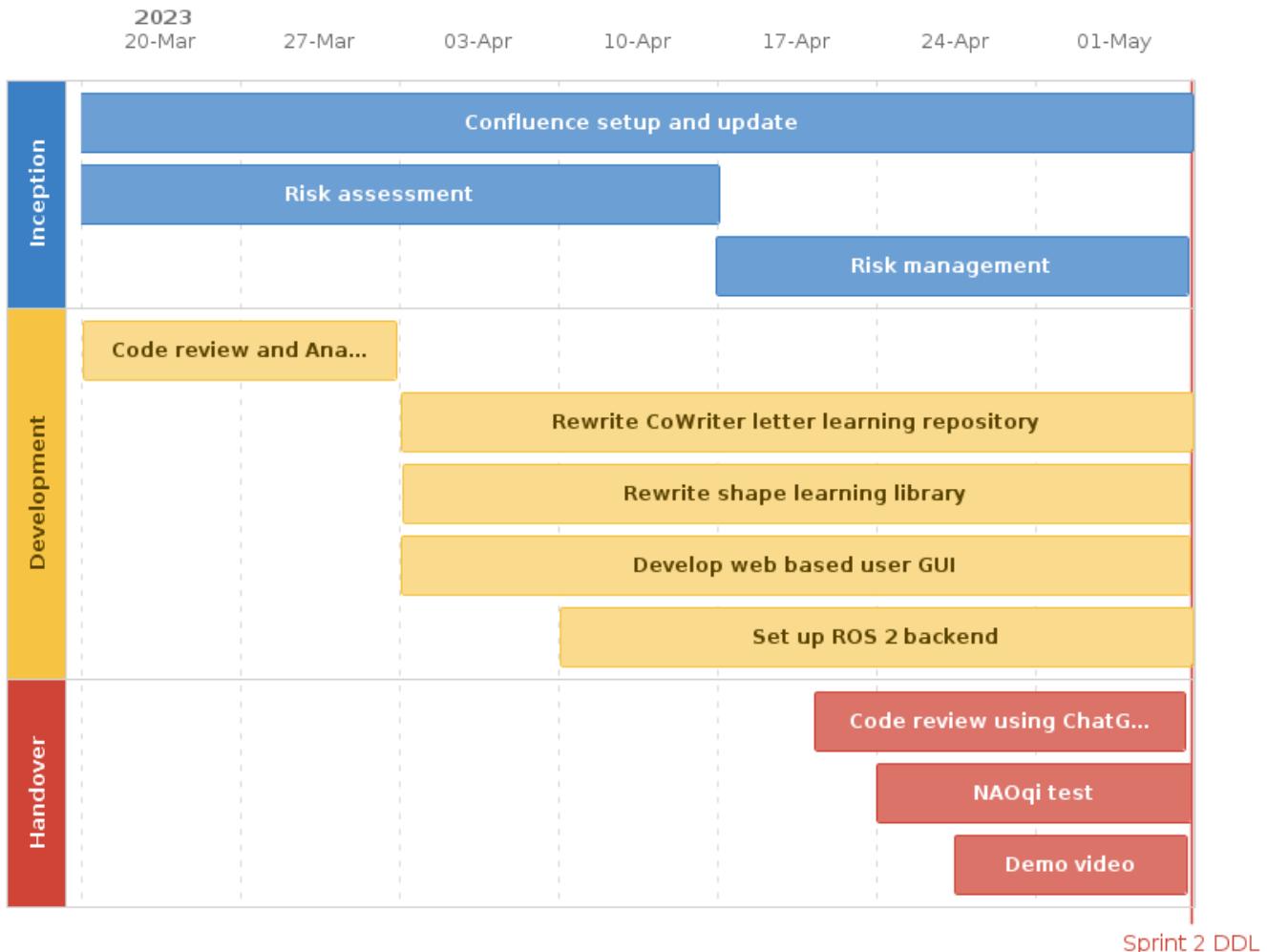
- Team members worked well, communicated effectively.
- Each team member attended booked meetings and shared their thoughts.

Improvement items:

- Update the Trello board up-to-date and catch up with the plan.
- Discard some of the nodes in the previous code repository.

Sprint2 plan

In this sprint, the team plans to make significant progress across various aspects of the project. Confluence will be set up and updated to keep documentation organized and accessible. A thorough risk assessment will be conducted to identify potential challenges and mitigate their impact. The team will perform a code review and analysis, which will lead to the successful rewriting of both the CoWriter letter learning repository and the shape learning library. A web-based user GUI will be developed to enhance user interaction, and the ROS 2 backend will be established to streamline communication between components. The team also plans to utilize ChatGPT for code review and conduct tests with the NAOqi SDK to ensure the proper functioning of the robot. Finally, a demo video will be created to showcase the progress made during this sprint.



Sprint2 retrospect

Sprint reflection on the completed sprint:

Accomplishment:

- Rewrote cowriter_letter_learning repository.
- Rewrote shape_learning library.
- Developed web based user GUI.
- Set up ROS 2 backend.
- Updated Confluence materials with respect to the feedback from Sprint 1.

Challenges:

- Using NAOqi SDK enables the virtual robot in choregraphe can move and make posture by given coordinates.

Not finished:

- Feature enables NAO robot to write whole words

Team performance:

- Each team member has shown great efficiency.
- Developers completed the demo for Sprint 2 in cooperation.

Improvement items:

- Allocate more time for tasks that proved to be more complex than initially anticipated, such as enabling the NAO robot to write whole words.
- Ensure that high-priority tasks, like completing the whole-word writing feature for the NAO robot, are addressed early in the sprint to minimize the risk of not completing them on time.

Sprint3 plan

Sprint3 retrospect

Test process

This part covers our project test steps for each Sprint:

- [Sprint1](#)
- [Sprint2](#)

Sprint1

User story ID	Target	Pre-condition	Testing steps	Status
13	Verify that the port structure is clean and orderly and that future functionality can be easily developed.	-Port development is complete. -The code has been reviewed and approved by the development team.	-Open the port code file and examine the code structure. -Verify that the code is organized logically and consistently. -Check that the code is properly commented and documented to explain each module, function, and variable clearly. -Add new functionality to the port and verify that the code structure makes it easy to implement. -Ask the development team what they think of the code structure and if they find it easy to use. - Clearly document any problems or suggestions for improvements related to the code structure.	Assigned Assigned Assigned Assigned Assigned Assigned
14	Verify that the comments in the code are clear so that customers can quickly understand the code and follow up.	-Code development is complete. -The code has been reviewed and approved by the development team.	-Comments should be clear and clear to help understand the logic and purpose of the code. -Comments should cover all important code segments, features, variables, and functions involved. -Comments should be updated to reflect code changes and modifications. -The development team should confirm that comments help understand the code and develop new features. -Clearly document any problems or improvement suggestions related to code comments.	Assigned Assigned Assigned Assigned Assigned
15	Verify that the README in the project is clear, introduces the project clearly, and helps with future handoffs with other developers.	-Project development has been completed. -README has been reviewed and approved by the development team.	-Open the README file and examine its contents. -Verify that the README clearly describes the project's purpose, function, and operation. -Try to follow the instructions in the README and verify that the operation completes successfully. -Ask the development team what they think of README and if they think it will be very helpful for the transition. -Clearly document any problems or suggestions for improvement related to the README.	Assigned Assigned Assigned Assigned Assigned
16	Verify that code in the project, including customer-provided code and code developed by the development team, is written in the latest version of the programming language as required to facilitate the development of new functionality by the customer.	-Project development has been completed. -The code has been reviewed and approved by the development team.	-Review the code in the project, both customer-supplied code and code developed by the development team. -Verify that the code is written as required in the latest programming language version (Python3, ROS2). -Ask the development team for their opinion on how the code is written and using the latest programming language version and talk to the customer. -Document any problems or improvements related to how the code is written.	Assigned Assigned Assigned Assigned

Sprint2

User story ID	Target	Pre-condition	Testing steps	Status
3	To significantly improve the handwriting ability of preschool children, including writing letters, numbers, words, and short sentences when using the robot.	<ul style="list-style-type: none"> -Children already have basic writing skills. -Parents and educators are willing and able to participate in the children's writing practice process, providing necessary support and feedback. 	<ul style="list-style-type: none"> -Conduct a baseline test of the children's handwriting ability before the course starts, recording their levels in writing letters, numbers, words, and short sentences. -During the semester, conduct a handwriting ability check every two weeks, assess the children's progress in various aspects, and compare it with the baseline test. -Collect feedback from parents and educators to understand their satisfaction with the course and suggestions for improvement. -At the end of the course, conduct a summative assessment to comprehensively examine the children's handwriting ability and understand their progress within a semester. -Compare the summative assessment results with the baseline test, analyze the children's improvement in various aspects, and whether the expected target has been achieved. -Based on the assessment results and feedback, optimize and adjust the course better to meet the user story's acceptance criteria. 	Assigned
7	The robot can save and record the children's reading and writing practice. Parents and educators can keep track of the children's progress through the robot record.	<ul style="list-style-type: none"> -The robot installs and integrates the software to capture and analyze handwriting. -The child is familiar with the robot and is comfortable with its presence when practising writing. -Parents know how to use the robot properly. 	<ul style="list-style-type: none"> -Conduct a pre-test to evaluate the child's handwriting quality before using the robot's guidance. -Introduce the robot to the child's handwriting practice sessions and ensure that it functions correctly in capturing, analyzing, and providing feedback on the child's handwriting. -Monitor the child's interactions with the robot during handwriting practice sessions over a period of 8 weeks, collecting data on the frequency and effectiveness of the robot's feedback. -Obtain feedback from the child and parent regarding the usability of the robot and its impact on the child's handwriting. -Compare pre-test and post-test results to assess the effectiveness of the robot in improving the child's handwriting. -Optimize the robot's guidance methods and functionalities based on the test results and feedback to better meet the acceptance criteria. 	Assigned
10	Enable the addition of new functions to the Nao robot with minimal impact on existing modules, reducing the difficulty and cost of maintaining existing functionality and subsequent feature development.	<ul style="list-style-type: none"> -The Nao robot has a modular architecture with well-defined interfaces for communication between modules. -The maintainer has access to the robot's documentation, source code, and development tools. -The maintainer has a good understanding of the robot's functional modules and their dependencies. 	<ul style="list-style-type: none"> -Check whether there is a ROS interface that can be implemented. -Check the communication between publishers and subscribers of topics across different nodes to ensure that it functions normally. T -Try rewriting a new interface and check if the insertion of new functionality works properly. -Evaluate the difficulty and cost of maintaining the existing functionality and developing the new function, comparing it to previous development experiences. 	Assigned

		<ul style="list-style-type: none"> -Collect feedback from users (e.g., developers, maintainers, end-users) on the ease of adding new functions to the robot and any potential improvements. 	Assigned	
		<ul style="list-style-type: none"> -Based on the test results, feedback, and evaluation, make any necessary adjustments to the robot's modular architecture or integration process to better meet the target. 	Assigned	
16	Verify that code in the project, including customer-provided code and code developed by the development team, is written in the latest version of the programming language as required to facilitate the development of new functionality by the customer.	<ul style="list-style-type: none"> -Project development has been completed. -The code has been reviewed and approved by the development team. 	<ul style="list-style-type: none"> -Review the code in the project, both customer-supplied code and code developed by the development team. 	Assigned
			<ul style="list-style-type: none"> -Verify that the code is written as required in the latest programming language version (Python3, ROS2). (Notes: The part about calling Naoqi is written in Python 2.7.) 	Assigned
			<ul style="list-style-type: none"> -Try running various features and check if the results match the expectations and record the bug(if it occurred). 	Assigned
			<ul style="list-style-type: none"> -Verify that the code is written as required in the latest programming language version (Python3, ROS2). 	Assigned
			<ul style="list-style-type: none"> -Document any problems or improvements related. 	Assigned

Risk Assessment and Management

Specific risk:

Risk ID	Risk Type	Description	Probability (0-1)	Impact (1-10)	Exposure	Justification	Owner	Response	Response Strategy	Resource Required
1	Project	Since upgrading and optimizing CoWriter software requires modifications and extensions to existing systems, there may be technical risk such as compatibility issues.	0.8	5	4	Acknowledging this risk allows our team to plan for thorough testing and troubleshooting, ensuring a smooth transition and minimizing disruptions during the upgrade process.	Minyi Chen, Yuhang Wang, Sijia Pei	Mitigation	<ul style="list-style-type: none"> Analyze existing systems and dependencies before modifications. Conduct compatibility testing between upgraded and existing components. 	<ul style="list-style-type: none"> Skilled developers experienced with CoWriter technologies. Time for research, testing, and implementation.
2	Product	Due to the upgraded CoWriter which integrated with ChatGPT, the Conversational interactions may be incorrect or incomplete and conversational scenarios may not be accurately identified.	0.3	5	1.5	Knowing this risk helps us to prioritize testing on the communication between the user and the NAO robot in specific conversational scenario.	Minyi Chen, Yuhang Wang, Sijia Pei	Mitigation	<ul style="list-style-type: none"> Test ChatGPT using diverse conversational scenarios. Update and refine model training data for domain-specific context. 	<ul style="list-style-type: none"> Testers or developers experienced in conversational AI systems. Access to domain-specific training data.
3	Product	Nao Robot requires consideration of robot performance, stability and durability, there may be risks associated with robot deployment and configuration.	0.2	9	1.8	Acknowledging this risk allows us to keep our eyes on the configurations of NAO robot. Make sure its stability and durability.	Minyi Chen, Yuhang Wang, Sijia Pei	Mitigation	<ul style="list-style-type: none"> Conduct comprehensive testing of the robot's performance and stability under various conditions and scenarios. Implement regular maintenance procedures to ensure the robot's durability and prevent hardware issues. 	<ul style="list-style-type: none"> Testers or engineers experienced with NAO Robot. Appropriate testing environments and tools to evaluate robot performance.

4	Project	Since upgrading CoWriter software may involve sensitive data, such as user information and learning records, there may be data leakage and security issues.	0.3	9	2.7	Upgrading CoWriter software involves handling sensitive data, which may introduce data leakage and security issues. Recognizing this risk allows us to focus on implementing robust security measures and data protection protocols, ensuring the confidentiality and integrity of user information.	Minyi Chen, Yuhang Wang, Sijia Pei	Avoid	<ul style="list-style-type: none"> Implement data encryption and secure communication protocols to protect sensitive data during transmission and storage. Conduct a thorough security review of the upgraded software to identify. 	<ul style="list-style-type: none"> Security experts or developers with experience in data protection. Tools and technologies to establish access controls and authentication mechanisms.
5	Product	The NAO robot is integrating with ChatGPT. Due to the complexity, there may be a risk that our young age users cannot understand the words and actions made by robots.	0.8	7	5.6	Integrating ChatGPT with the NAO robot may result in complex words and actions that young users find difficult to understand. Acknowledging this risk enables us to prioritize simplicity and user-friendliness in the robot's interactions, ensuring an engaging and accessible experience for young users.	Minyi Chen, Yuhang Wang, Sijia Pei	Mitigation	<ul style="list-style-type: none"> Conduct user testing with a representative sample of the target age group to identify potential comprehension issues and make necessary adjustments. Implement a simple and intuitive user interface to facilitate user interaction with the robot. 	<ul style="list-style-type: none"> UX/UI designers to develop a user-friendly interface

Generic risk:

Risk ID	Risk Type	Description	Probability (0-1)	Impact (1-10)	Exposure	Justification	Owner	Response	Response Strategy	Resource Required
---------	-----------	-------------	-------------------	---------------	----------	---------------	-------	----------	-------------------	-------------------

1	Project	Since upgrading and optimizing CoWriter software requires consideration of multiple aspects, such as the special needs of dyslexic children, system performance and stability, there may be time risks, such as project delays.	0.8	9	7.2	By recognizing this time risk, we can allocate resources effectively and plan accordingly to minimize delays while ensuring the software meets the diverse needs of its users.	Chengfeng Zhang, Da Zhang	Mitigation	<ul style="list-style-type: none"> Develop a detailed project plan with clear milestones and deadlines to track progress and ensure timely completion. Allocate sufficient resources and prioritize tasks to minimize bottlenecks and avoid delays. 	<ul style="list-style-type: none"> Skilled project managers to develop and oversee the project plan. Adequate team size and expertise. Time and resources for continuous risk management and monitoring.
2	Project	We have to integrate the NAO robot with ChatGPT, it needs a series of tests and adjustments will be required, and with a tight budget, we may risk overspending.	0.2	7	1.4	Acknowledging this financial risk enables us to allocate resources efficiently, prioritize essential tasks, and implement cost-effective strategies to minimize budget overruns.	Chengfeng Zhang	Mitigation	<ul style="list-style-type: none"> Develop a detailed budget plan, allocating resources efficiently to prioritize essential tests and adjustments. Monitor and track expenses regularly to ensure alignment with the budget plan and identify potential cost overruns early. 	<ul style="list-style-type: none"> Skilled financial planner or project manager to develop and manage the budget plan effectively. Access to cost-effective testing tools, technologies, and strategies to optimize the testing process.

3	Project	The client's envisioned project differs from the target user and team's understanding, potentially leading to a less-than-ideal outcome that may not fully meet the client's expectations.	0.5	7	3.5	Understanding this risk enables us to prioritize clear communication, collaboration, and feedback loops to ensure alignment, ultimately delivering a project that meets the client's expectations and requirements.	Chengfeng Zhang, Da Zhang	Mitigation	<ul style="list-style-type: none"> Establish clear and open communication channels with the client, encouraging regular feedback and discussion throughout the project. Conduct thorough requirement gathering and analysis to ensure a deep understanding of the client's expectations and needs. 	<ul style="list-style-type: none"> Skilled project manager to facilitate effective communication between the client and the team.
4	Product	The possibility of substandard software development, including inadequate code specification and logic errors, may result in compromised system stability and reliability, potentially affecting the overall performance and user experience.	0.2	5	1	By acknowledging this risk, we can prioritize thorough code reviews, rigorous testing, and adherence to best practices, ensuring the delivery of a high-quality product that meets performance standards and provides a satisfactory user experience.	Minyi Chen, Yuhang Wang, Sijia Pei	Mitigation	<ul style="list-style-type: none"> Implement strict coding standards and best practices to ensure high-quality code development. Use automated testing tools and frameworks to perform continuous testing of the software, validating functionality, stability, and performance. 	<ul style="list-style-type: none"> Skilled developers with experience in maintaining high-quality coding standards and best practices.

5	Product	Insufficient testing could result in undetected errors or vulnerabilities, potentially compromising the system's stability and reliability, and negatively impacting overall performance and user experience.	0.3	5	1.5	Acknowledging the risk of inadequate testing allows us to prioritize comprehensive testing strategies, including unit, integration, and stress testing, to identify and address potential errors or vulnerabilities. This ensures system stability, reliability, and a positive user experience, contributing to a successful product launch.	Chengfeng Zhang, Yuhang Wang	Mitigation	<ul style="list-style-type: none"> Develop a comprehensive test plan that covers functional, performance, and security testing. Allocate sufficient time and resources for thorough testing during each stage of the project. 	<ul style="list-style-type: none"> Skilled testers or developers experienced in creating and executing comprehensive test plans.
---	---------	---	-----	---	-----	---	------------------------------	------------	---	---

Deliverables and Handover

This part covers the deliverables of our project at different development schedules.

- Sprint 2 process demo video
- Products

Sprint 2 process demo video

Link to the demonstration video: <https://www.youtube.com/watch?v=3e0Nvp-He80>

Products

Product Environment setup instructions:

Python3.10 setup:

Execute the following command in the command prompt based on your personal system:

Windows:

Visit the official Python website's download page at <https://www.python.org/downloads/>

Download the Windows installer for Python 3.10 by clicking on the appropriate link (either 32-bit or 64-bit, depending on your system).

Run the installer and follow the on-screen instructions.

Make sure to check the box that says "Add Python 3.10 to PATH" during installation. This will make it easier to run Python from the command line.

Complete the installation process.

macOS:

Visit the official Python website's download page at <https://www.python.org/downloads/>

Download the macOS installer for Python 3.10.

Open the downloaded package and follow the on-screen instructions to complete the installation process.

Linux (Ubuntu, Debian):

Python 3.10 may not be available in the default repositories for some Linux distributions. In that case, you can use the "deadsnakes" PPA repository to install it. Run the following commands in the terminal:

```
sudo apt update  
sudo apt install software-properties-common  
sudo add-apt-repository ppa:deadsnakes/ppa  
sudo apt update  
sudo apt install python3.10
```

After completing the installation, you can check the Python version by running `python3.10 --version` in the command line or terminal.

ROS2 setup

Execute the following command in the command prompt based on your personal system:

Ubuntu Linux:

```
sudo apt update sudo apt upgrade  
sudo locale-gen en_US en_US.UTF-8 sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8 export LANG=en_US.UTF-8  
sudo apt install curl gnupg2 lsb-release curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -  
sudo sh -c 'echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" > /etc/apt/sources.list.d/ros2.list'  
sudo apt update  
sudo apt install ros-humble-desktop  
echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc source ~/.bashrc
```

macOS and Windows:

ROS2 is primarily targeted at Ubuntu Linux, but there are experimental installations available for macOS and Windows. The official ROS2 installation documentation provides instructions for these platforms, but keep in mind that support may be limited, and not all features and packages may be available. For the most up-to-date instructions, refer to the official ROS2 documentation:

macOS: <https://index.ros.org/doc/ros2/Installation/OSX-Development-Setup/>

Windows: <https://index.ros.org/doc/ros2/Installation/Windows-Development-Setup/>

Remember to always check the official ROS2 documentation for the most accurate and up-to-date installation instructions.