

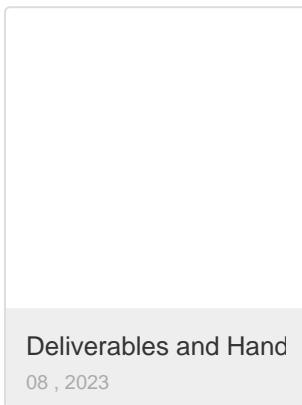
1. Home	2
1.1 Project Background	5
1.1.1 Goal model	6
1.1.2 Scope	7
1.1.3 Project Plan	8
1.1.4 Product Backlog	10
1.1.4.1 Sprint 1	11
1.1.4.2 Sprint 2	16
1.1.4.3 Sprint 3	21
1.1.4.4 Sprint 4	25
1.2 Users and Stakeholders	26
1.2.1 Personas	27
1.2.2 User Stories	31
1.2.3 User acceptance criteria	32
1.2.3.1 Sprint 1 User acceptance criteria	33
1.2.3.2 Sprint 2 User acceptance criteria	34
1.2.3.3 Sprint 3 user acceptance criteria	35
1.3 Development Process	36
1.3.1 Branching strategy	37
1.3.2 Code Reviewing Strategy	38
1.3.3 Development Environment	39
1.3.4 Week8 (Sprint2) Code Review	42
1.3.5 Week 10 code review	45
1.4 Sprints Planning and Review	47
1.4.1 Sprint 1 Plan	48
1.4.2 Sprint 1 Review	49
1.4.3 Sprint 2 Plan	50
1.4.4 Sprint 2 Review	51
1.4.5 Sprint 3 Plan	52
1.4.6 Sprint 3 Review	53
1.4.7 Sprint 4 Plan	54
1.4.8 Sprint 4 Review	55
1.5 Test process	56
1.5.1 Sprint1	57
1.5.2 Sprint2	58
1.5.3 Sprint3	59
1.6 Risk Assessment and Management	61
1.6.1 Risk Assessment	62
1.6.2 Risk Management	63
1.7 Deliverables and Handover	65
1.7.1 Sprint 2 process demo video	66
1.7.2 Final Presentation	67
1.7.3 Environment Setup Instruction	68
1.7.4 Product Setup Instruction	71
1.7.5 Sprint4 Final Demonstration video	73

Home



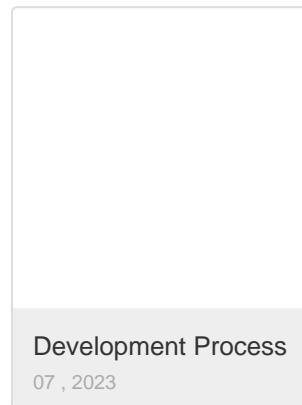
Sprints Planning and R

08 , 2023



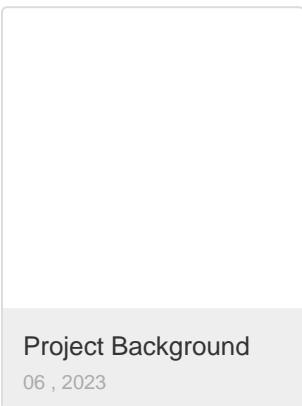
Deliverables and Hand

08 , 2023



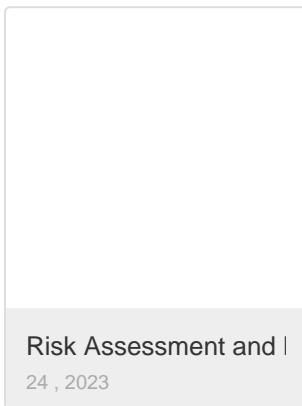
Development Process

07 , 2023



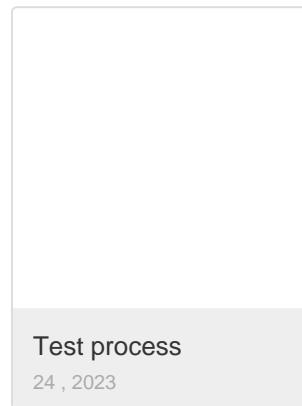
Project Background

06 , 2023



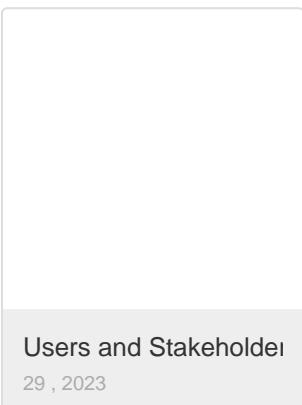
Risk Assessment and I

24 , 2023



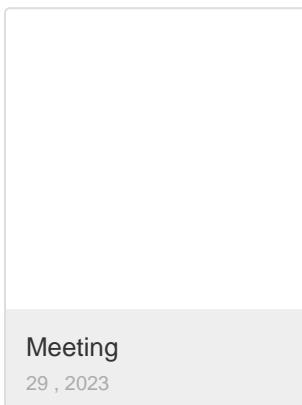
Test process

24 , 2023



Users and Stakeholder

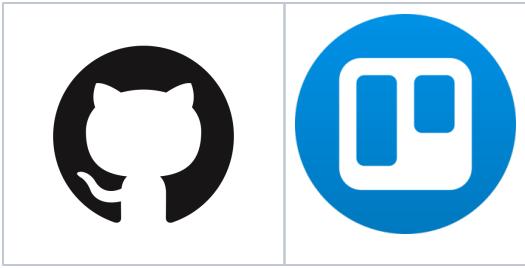
29 , 2023



Meeting

29 , 2023

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



Trello link sprint1:<https://trello.com/invite/b/lBVbLBjj/ATTId270c198df3df8e6019773576f136d3688780042/sprint1>

Trello link sprint2:<https://trello.com/invite/b/0BSXP1S6/ATTI2f7d11233c6ad5b060c2b89b06f2b077C37A1A1F/sprint2>

Trello link sprint3:<https://trello.com/invite/b/lok5Xccb/ATTI4255911dbf1eb229ac5b9aed68c3d0acDAA72C82/sprint3>

Trello link sprint4:<https://trello.com/invite/b/84x4mp8c/ATTIfa39864413f2e735454ecf3c6e55cbadB4161EB1/sprint4>

Stakeholders

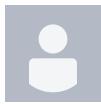
Name	Title	Contact Email	Photo
Wafa Johal	Client	wafa.johal@unimelb.edu.au	
Max Plumley	Supervisor	max.plumley@unimelb.edu.au	
Eduardo Oliveira	Instructor	eduardo.oliveira@unimelb.edu.au	

Team members

Team members	Title	Contact Email	Responsibility
Chengfeng Zhang	Product Owner	Chengfenz@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	<ol 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	<ol 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

Meeting 6/9 (with Wafa Johal) 2 •

Meeting 5/17 (with Wafa Johal) 5 •

Client Meeting 8 •

Sprint 4 Review 10 •

Sprints Planning and Review 15 •

Space contributors

- [Chengfeng ZHANG](#) (2)
- [Yuhang Wang](#) (11)
- [Da Zhang](#) (1)
- [Sijia PEI](#) (1)
- [Minyi Chen](#) (73)
- ...

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 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. Unfortunately, the naoqi section on robot control is currently using Python 2.7 and will continue to be updated in the future.

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
 - An update version of Python(from py2 to py3) and ROS (from ROS1 to ROS2) for `cowriter_letter_learning`
 - An update version of Python(from py2 to py3) for `shape_learning`
 - A new drawing board serves as a tool for drawing letters
 - A communication function for transmitting information is established between the drawing board and ros2
- WP2: Integrate ChatGPT with the NAO robot for autonomous dialogue, personalised selection of words, and autonomous robot motion during conversation
 - An update for a better display of the drawing board
 - A chatgpt voice dialogue function that allows Nao robots to use chatgpt personalized vocabulary selection to communicate with users
 - Autonomous robot motion during conversationTo be developed
- 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. And WP1 and WP2 have been completed according to the client's requirements. WP3 is classified as a future development plan.

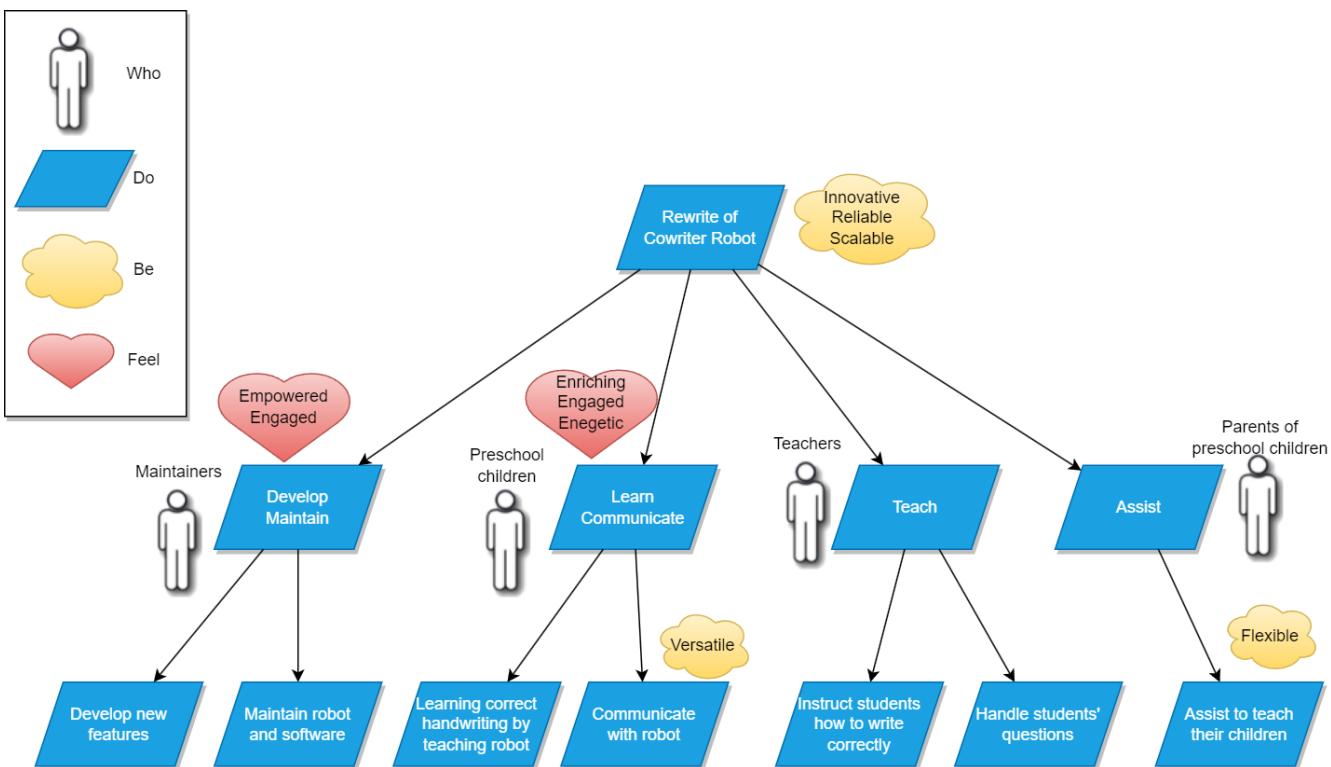
Work package is a term that used to describe a collection of works that groups together to define a specific part of a project.

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 2023-4-24



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

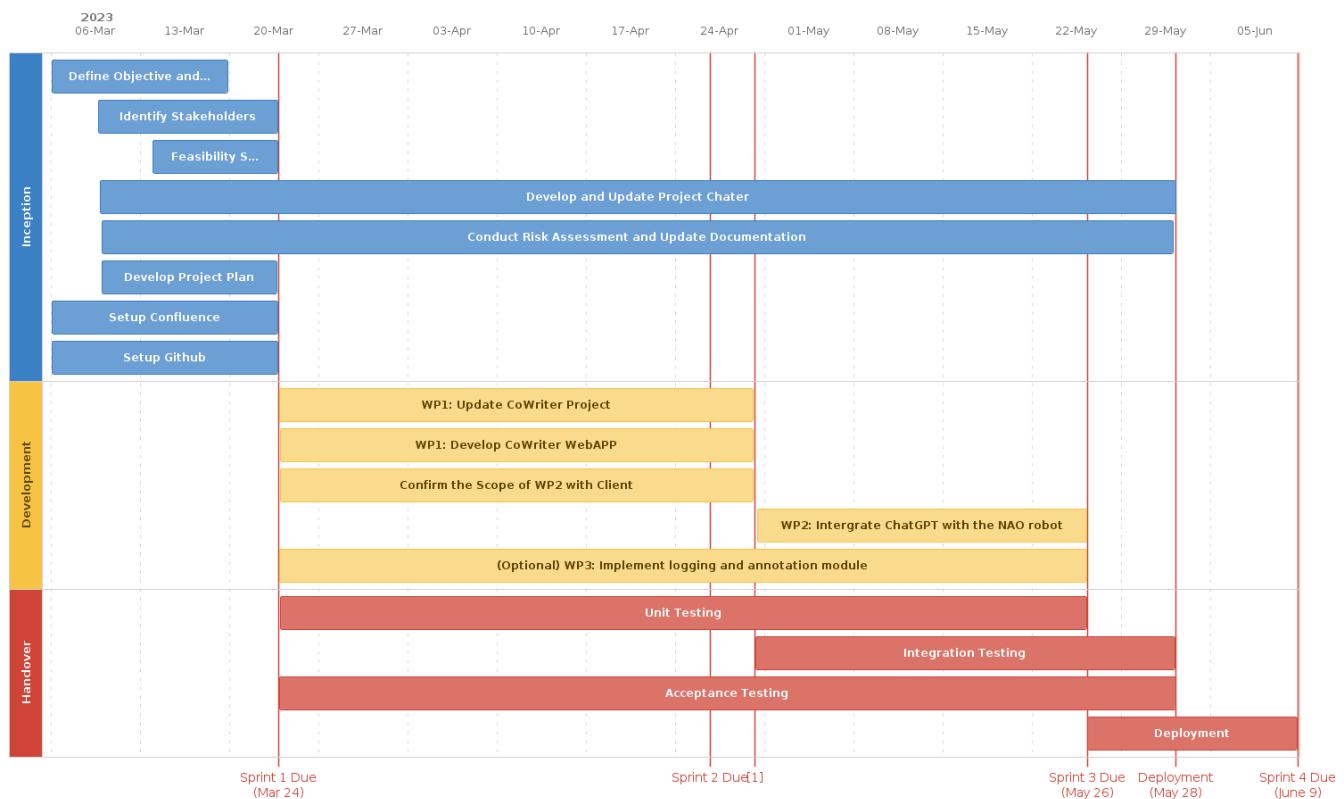
	Function	Description
1	Web Application	In addition to PC, developing a versatile web application that allows users to access and interact with the robot through multiple platforms, including Android, WeCom. By ensuring compatibility across a wide range of devices, users can seamlessly control and communicate 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	2023-4-28	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	2023-3-30	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

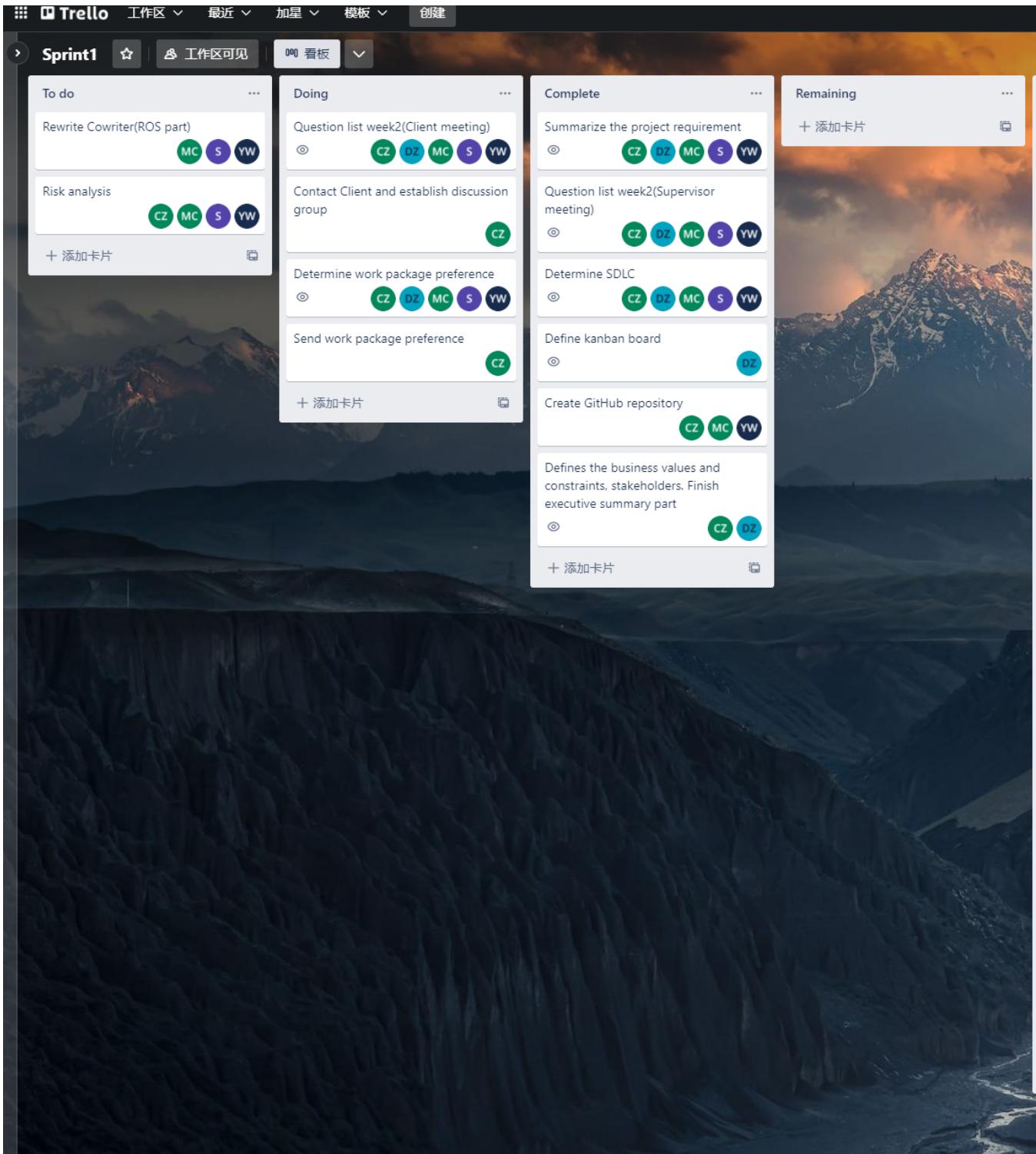
Sprint 1 Backlog

User Story ID	Decomposed userstory	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	2
	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	1
	As a client, I want the README file to show how to use and run the provided code.	-Maintain and update the README file.	1

Kanban board and burndown chart

<https://trello.com/invite/b/lBVbLBjj/ATTId270c198df3df8e6019773576f136d3688780042/sprint1>

Week2



Week3

Trello 工作区 最近 加星 模板 创建

Sprint1 看板

To do

- Rewrite Cowriter(ROS part)
CZ 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

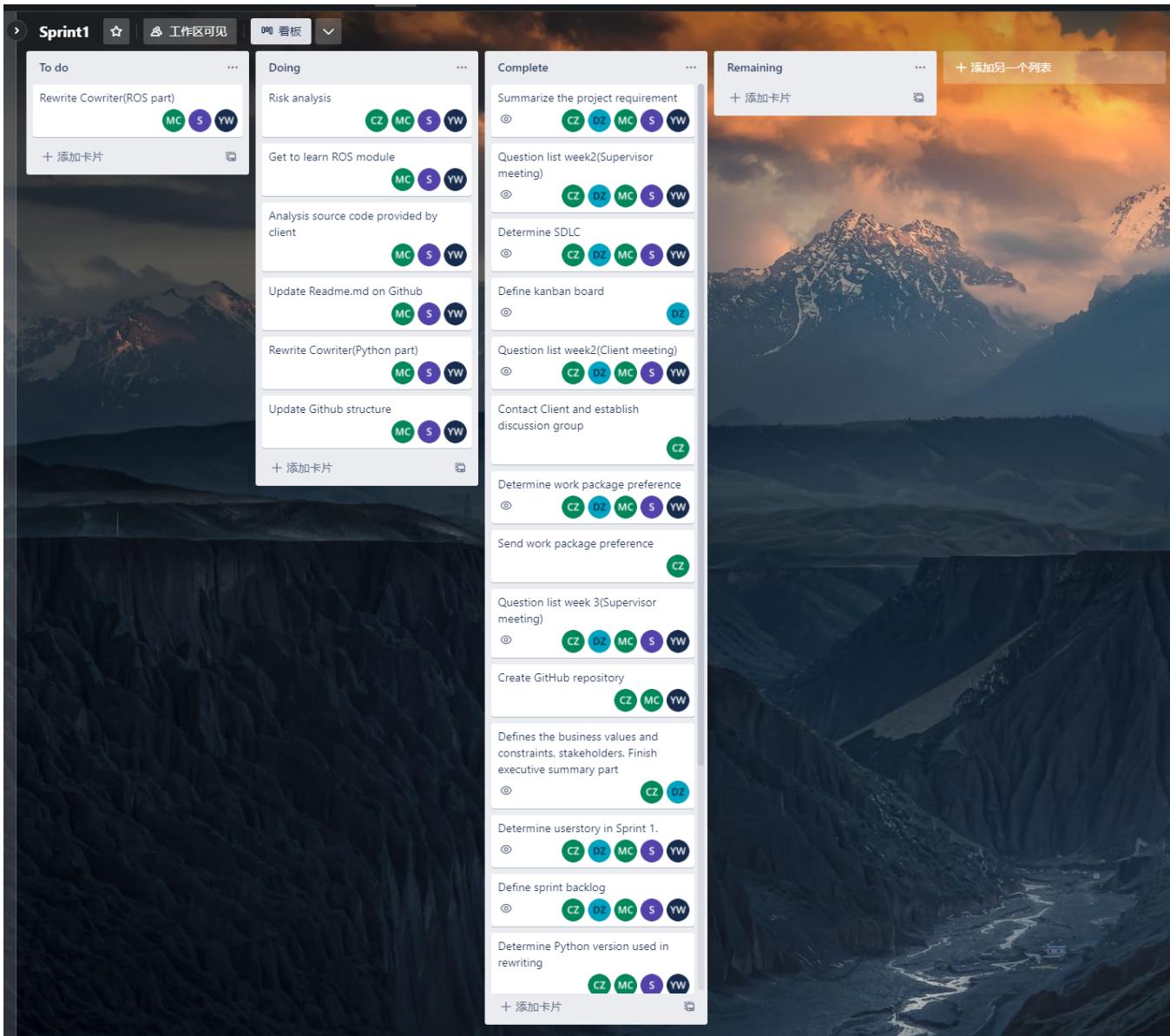
+ 添加卡片

Remaining

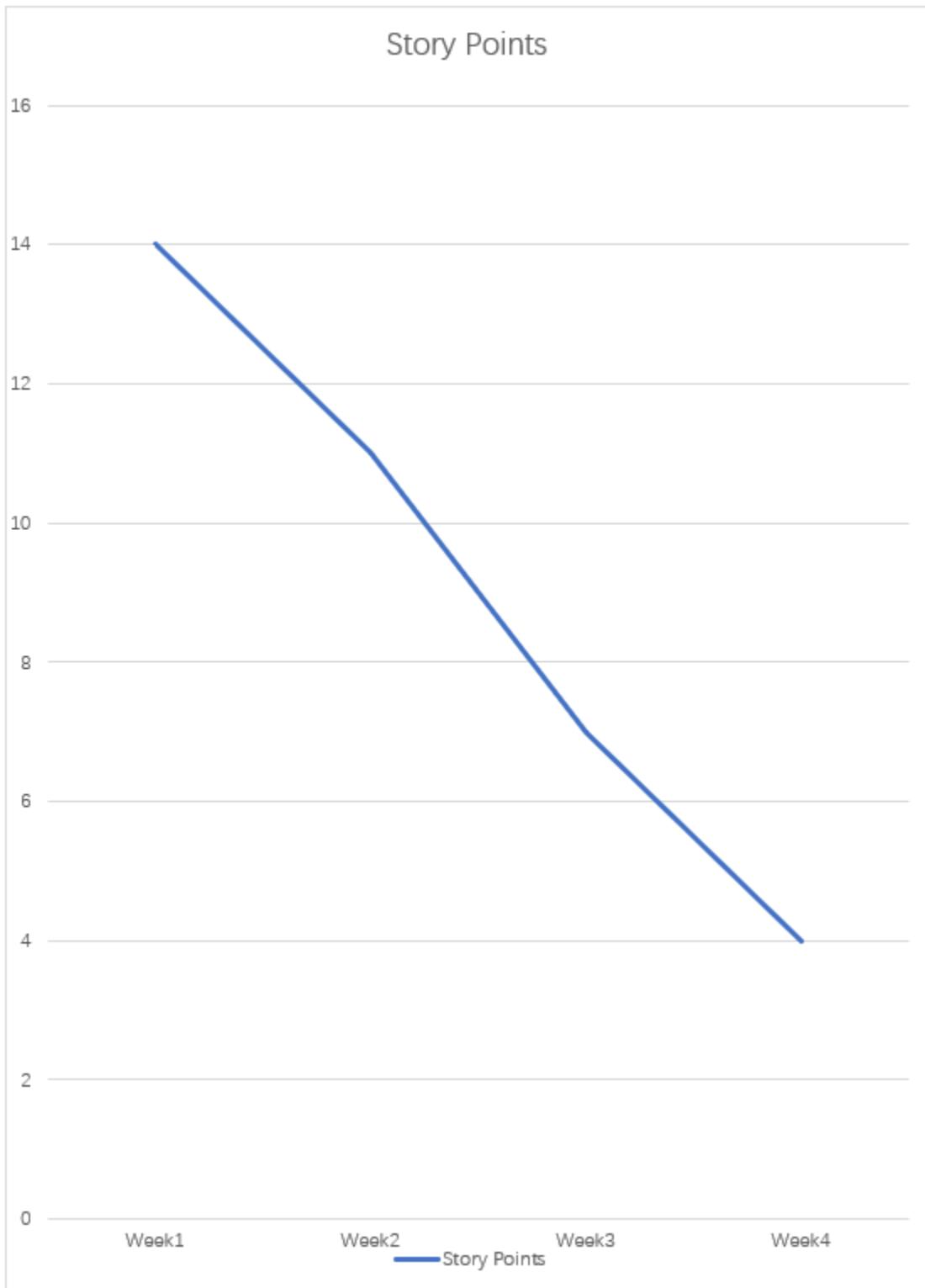
- + 添加卡片

<https://trello.com/c/wMdNXO55/13-get-to-learn-ros-module>

Week4



Sprint 1 burndown chart



Sprint 2

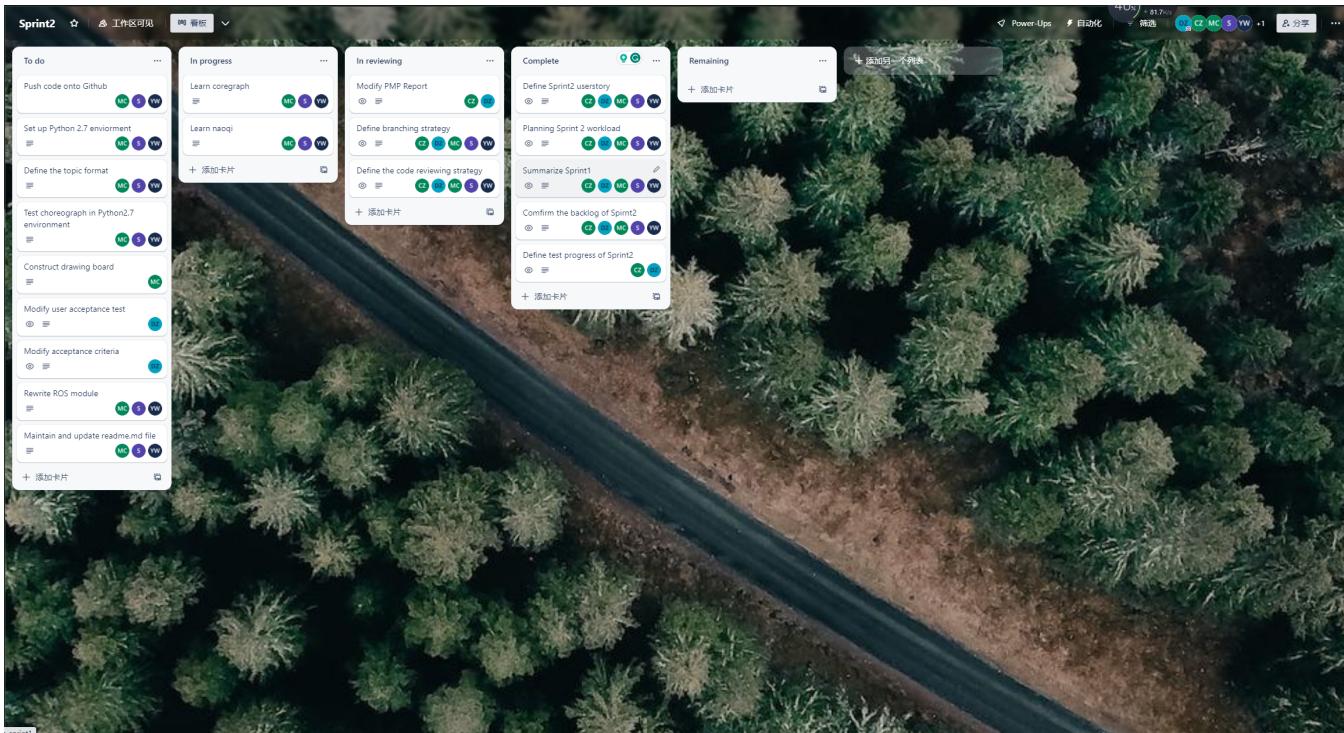
Sprint 2 Backlog

User Story ID	Decomposed userstory	Task	Story Point
5	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.	3
	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.	2
8	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	2
	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	1

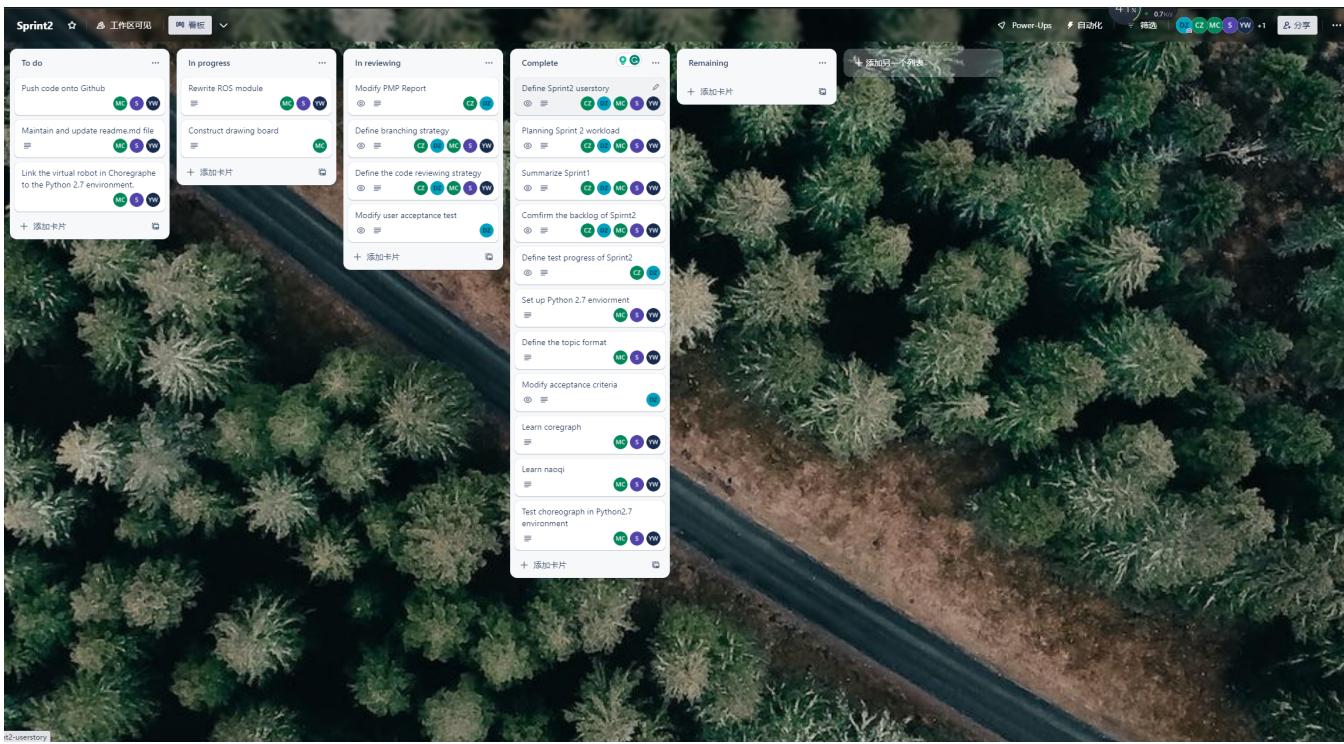
15	<p>As a maintainer, I want the source code provided to be rewritten in the latest Python version so that I can maintain the code more easily in the future.</p>	<ul style="list-style-type: none"> -Determine the latest version of Python used in the project with the Client. -Analysis of the code provided 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 shape_learning file and shape_modeler file. -Update shape_model_ui file. -Debug -Push the source code onto GitHub. -Maintain and update the README file. 	2
	<p>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.</p>	<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 covriter_letter_learning file. -Debug -Push the source code onto GitHub. -Maintain and update the README file. 	3

Sprint 2 week 1 Kanbanboard

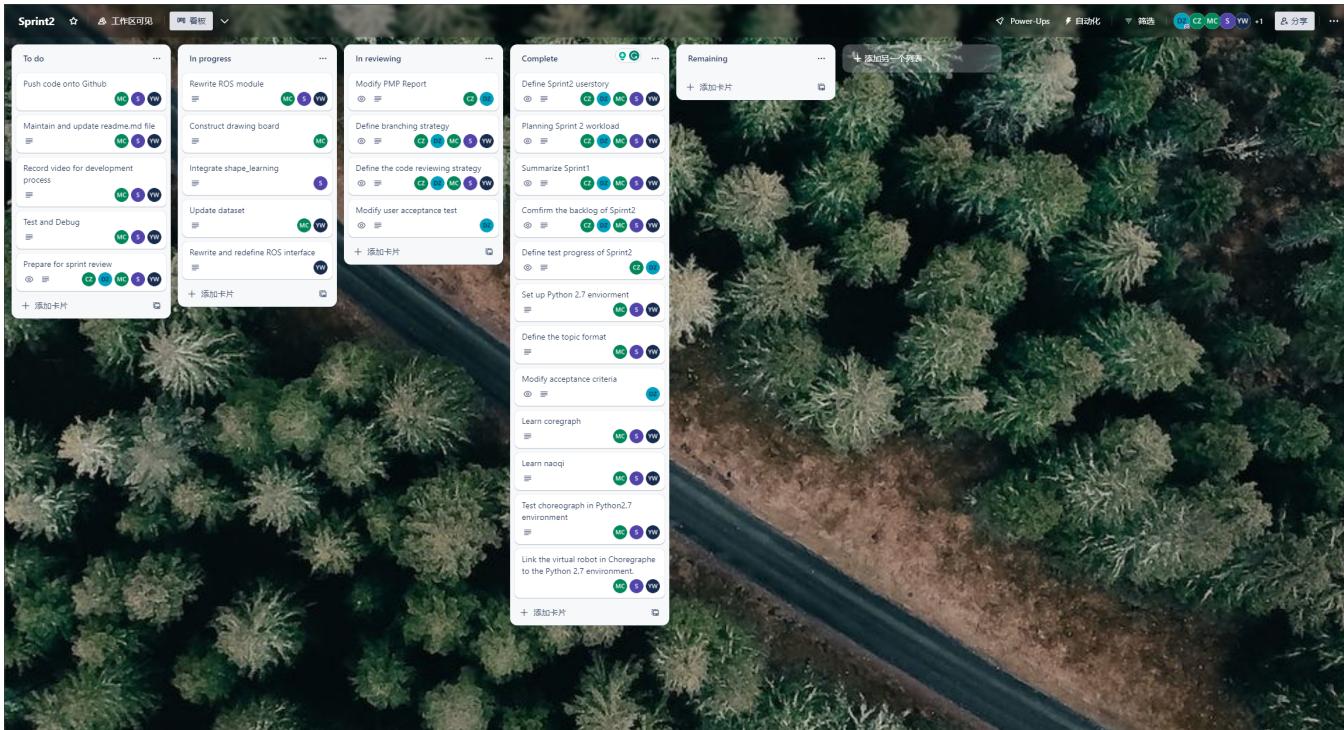
Trello link: <https://trello.com/invite/b/0BSXP1S6/ATTI2f7d11233c6ad5b060c2b89b06f2b077C37A1A1F/sprint2>



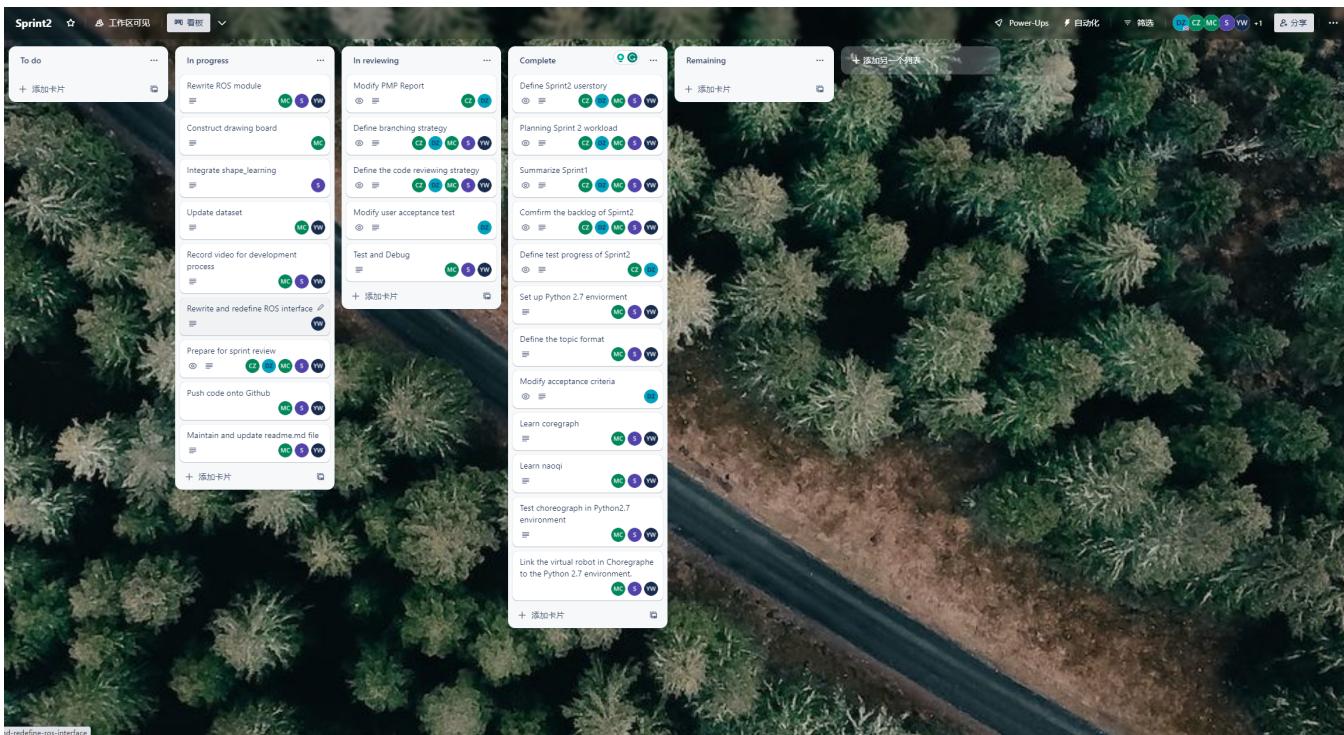
Sprint 2 week 2 Kanbanboard



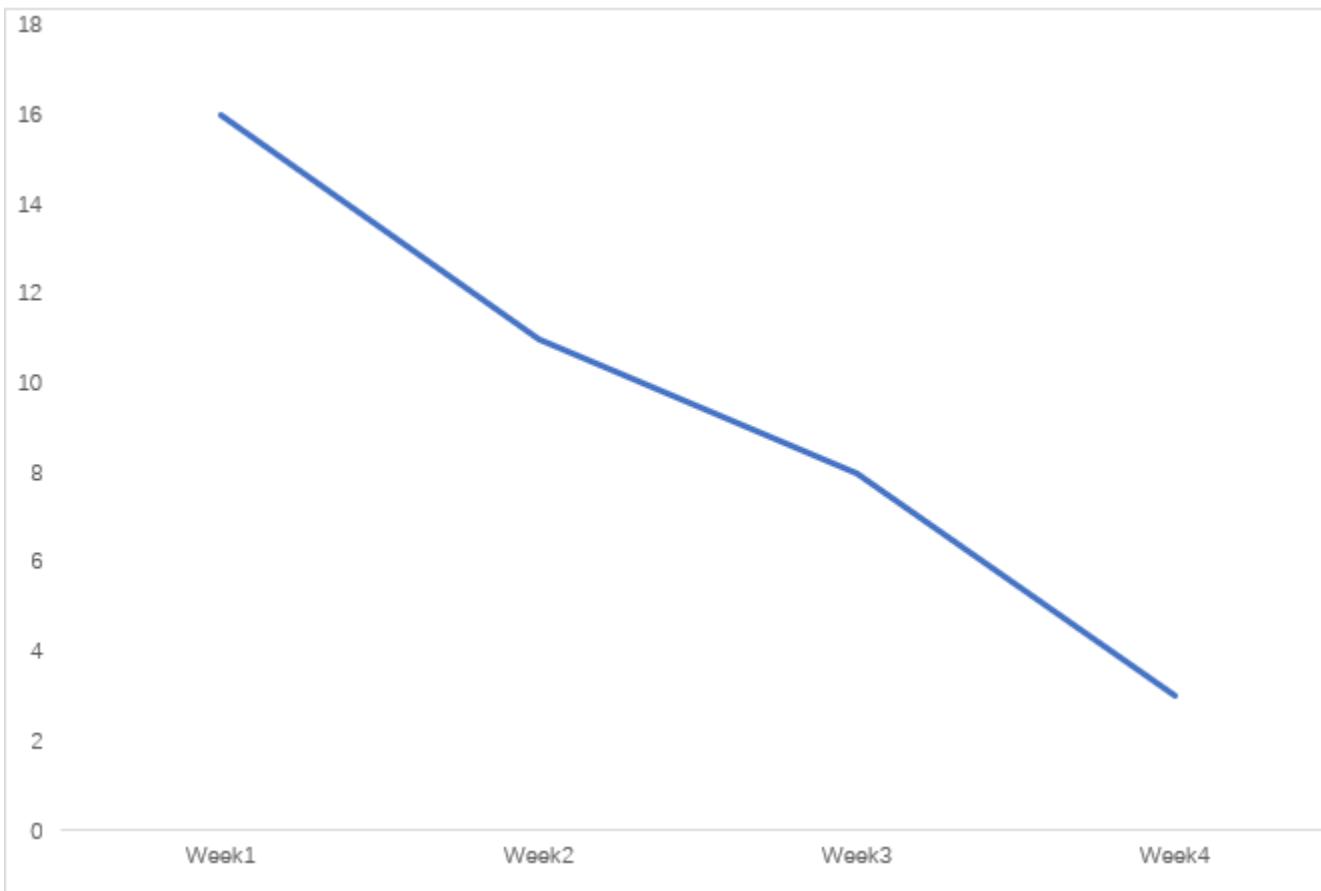
Sprint 2 week 3 Kanbanboard



Sprint 2 week 4 Kanbanboard



Burndown chart of Sprint 2



Sprint 3

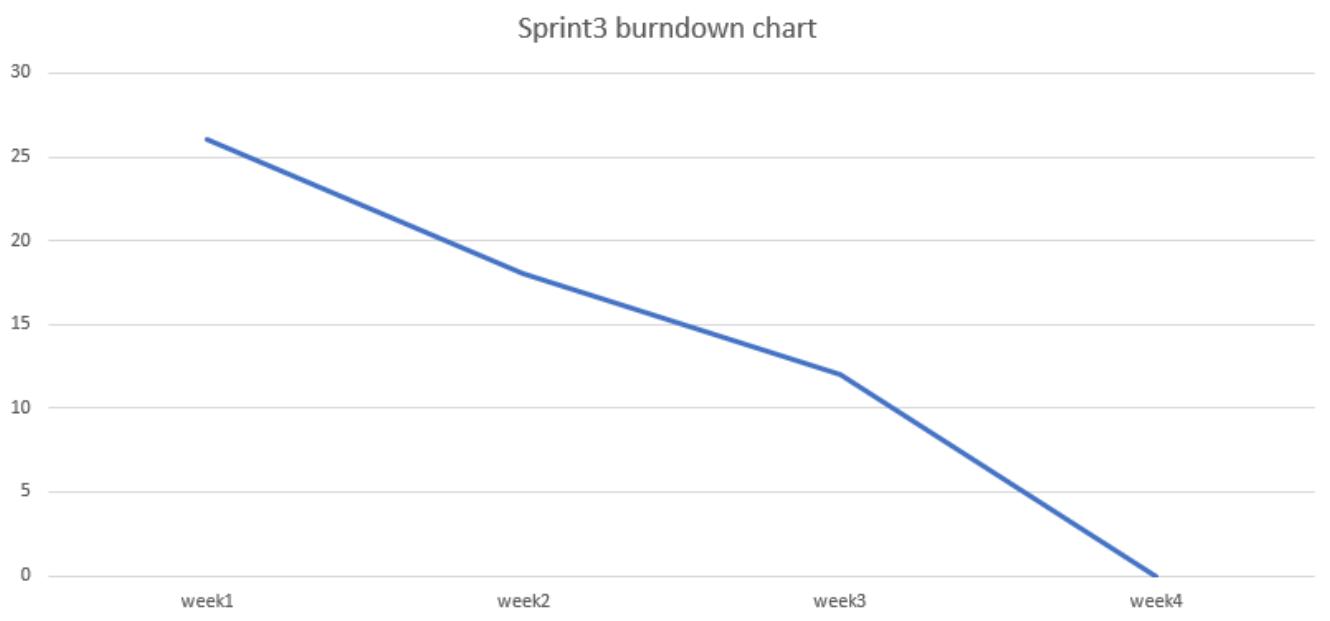
User Story ID	Decomposed userstory	Task	Story Point
2	As a preschool child, I want NAO robot to understand and respond accurately to my random questions so that I can get the information I need.	<ul style="list-style-type: none"> -Define and implement NLP models to accurately understand a wide range of questions. -Develop an algorithm for NAO robot to generate accurate and helpful responses. -Create a set of test questions and evaluate NAO robot's ability to understand and respond accurately. 	3
	As a preschool child, I want my interactions with NAO robot to be enjoyable and engaging.	<ul style="list-style-type: none"> -Implement speech synthesis to make NAO's responses sound natural and engaging. -Develop a feedback mechanism for users to rate their interaction and engagement. -Test the engagement of interactions with a focus group of users and iterate on feedback. 	2
3	As a preschool child, I want to activate Naobot through a simple voice command.	<ul style="list-style-type: none"> -Research suitable voice recognition technology for Naobot. -Integrate selected voice recognition technology into Naobot. -Test the effectiveness of activation with different voice commands in different environments. 	1
	As a preschool child, I want Naobot to accurately understand and process my voice instructions.	<ul style="list-style-type: none"> -Develop and enhance Naobot's language processing capability to understand children's speech patterns. -Test command processing accuracy and refine the model based on test results. 	3
	As a preschool child, I want Naobot's responses to be child-friendly and easy to understand.	<ul style="list-style-type: none"> -Design a response generation mechanism that ensures child-friendly and easy-to-understand responses. -Test responses with a group of children to validate understandability and appropriateness. -Iterate and refine the responses based on test feedback. 	2
4	As a preschool child, I want Naobot to generate a personalized collection of words based on my vocabulary usage.	<ul style="list-style-type: none"> -Develop a mechanism for Naobot to learn from a child's vocabulary usage over time. -Implement a function to generate a personalized word collection. -Test the accuracy and relevance of the generated word collection with children's input. 	5
	As a preschool child, I want Naobot to create vocabulary exercises based on my personalized collection of words.	<ul style="list-style-type: none"> -Design a feature that generates vocabulary exercises based on the child's personalized word collection. -Implement gamified elements to make the vocabulary exercises engaging. -Test the vocabulary exercises with children to ensure they are challenging and engaging. 	2
	As a preschool child, I want to receive appropriate feedback from Naobot after completing vocabulary exercises.	<ul style="list-style-type: none"> -Design a feedback mechanism that provides positive and constructive feedback to children. -Implement the feedback feature, ensuring it aligns with child-friendly language and tones. -Test the feedback mechanism with children, refining it based on their responses and comprehension. 	1
6	As a preschool child, I want NAO robot to accurately demonstrate the motion of writing specific characters.	<ul style="list-style-type: none"> -Program Nao robot's arm movement to accurately mimic the strokes of writing specific characters. -Validate the accuracy of the motion with a handwriting expert. 	3
	As a preschool child, I want NAO robot's demonstration to help me learn to write characters.	<ul style="list-style-type: none"> -Set up Nao bot functionality to ensure clear visibility of NAO's demonstrations. -Set up a user testing session to evaluate the effectiveness of NAO's demonstrations in teaching character writing. 	2

Sprint 3 Kanban board

Trello link:

<https://trello.com/invite/b/lok5Xccb/ATTI4255911dbf1eb229ac5b9aed68c3d0acDAA72C82/sprint3>

Sprint 3 burndown chart

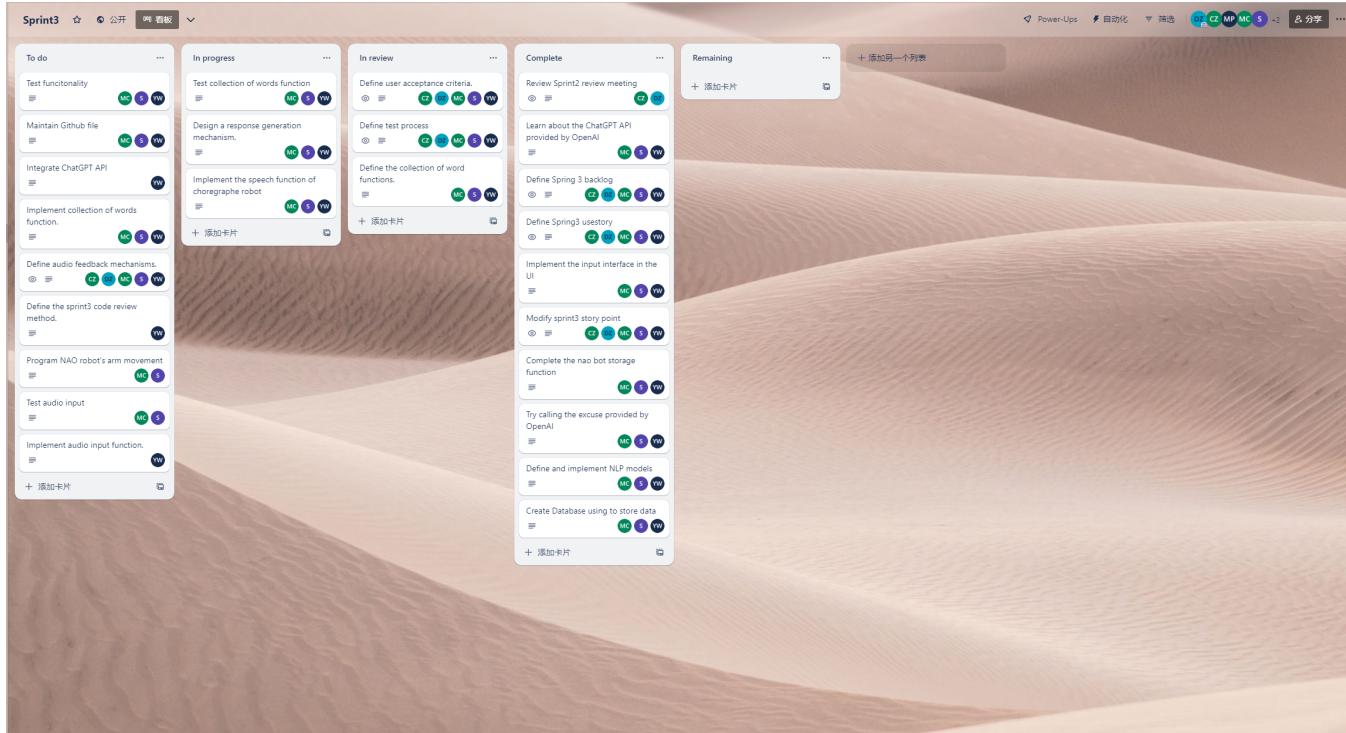


Week 1 Kanban board

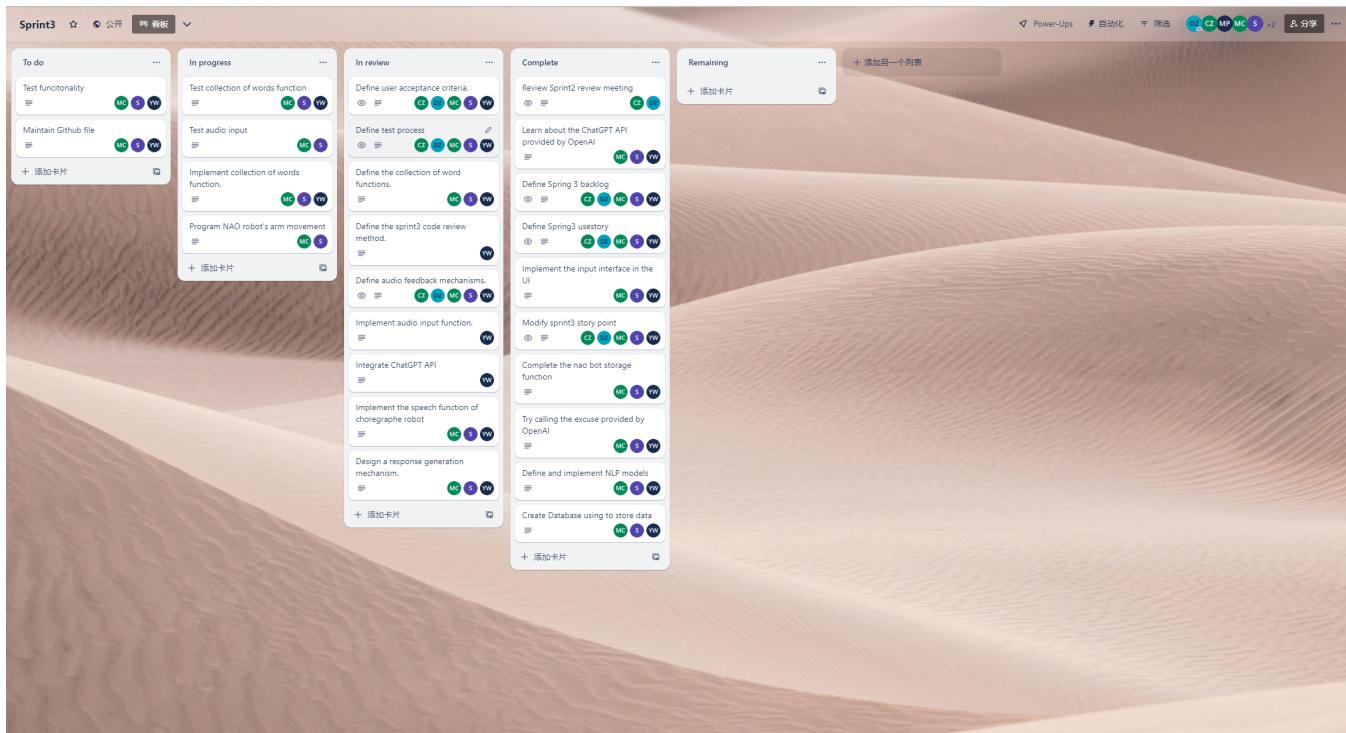
The Kanban board displays tasks across five columns: To do, In progress, In review, Complete, and Remaining. Each column contains several cards with task descriptions and status indicators. A new column, "Remaining", is shown on the right.

- To do:**
 - Test functionality
 - Maintain Github file
- In progress:**
 - Try calling the excuse provided by OpenAI
 - Implement the input interface in the UI
 - Implement the speech function of choregraphe robot
- In review:**
 - Define Spring 3 backlog
 - Create Database using to store data
 - Complete the nao bot storage function
 - Modify sprint3 story point
 - Define and implement NLP models
 - Design a response generation mechanism.
- Complete:**
 - Review Sprint2 review meeting
 - Learn about the ChatGPT API provided by OpenAI
 - Define Spring3 usestory
- Remaining:**
 - + 添加卡片

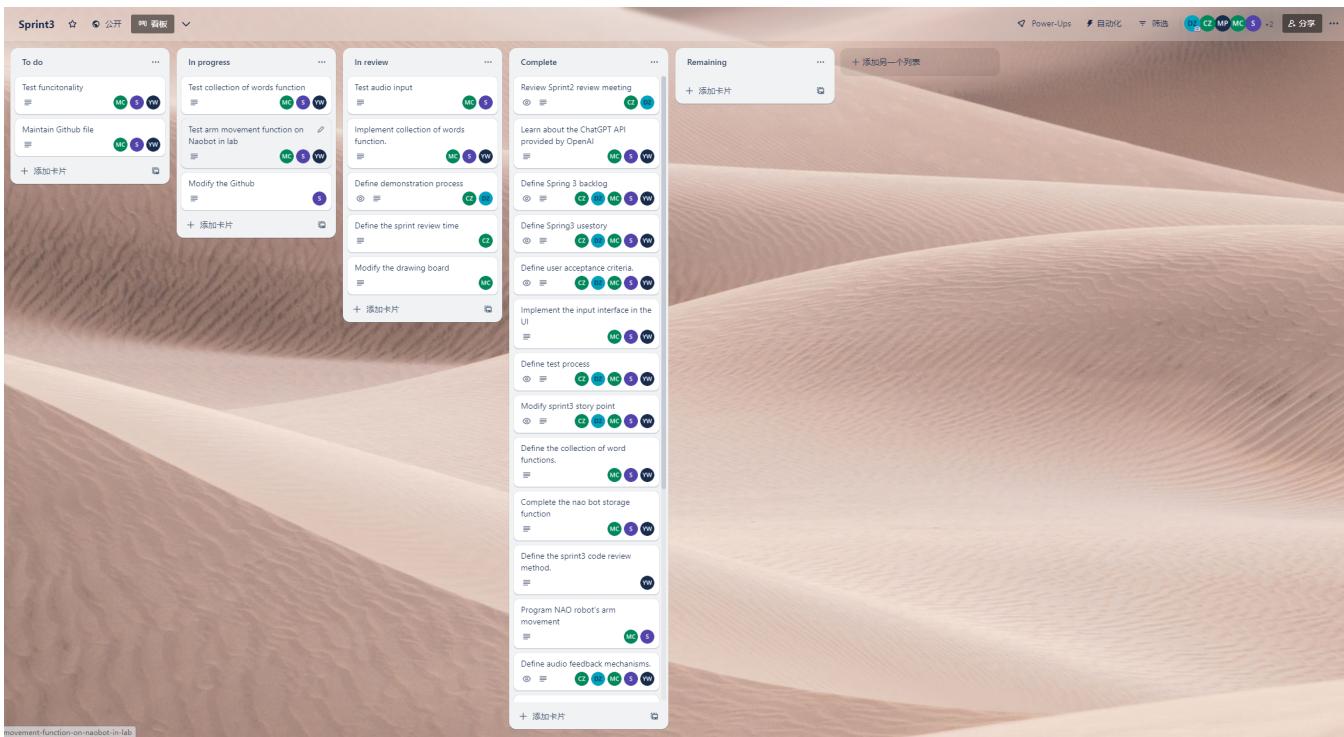
Week2 kanban board



Week 3 kanban board



Week 4 kanbanboard



Spirnt 4

Task ID	Task Name	Description
1	Create Release Notes	Create a detailed document explaining to the client how to access developed resources and how they are organized on GitHub, and also provide details on how to run their project.
2	Generate Release TAG	Generate a release TAG on Github containing all project resources.
3	Produce Demonstration Video	Produce a 3-5 minute video demonstrating the product, and upload it to Confluence.
4	Create ZIP File	Create a ZIP file containing all project resources, send it to the client, and add it to Confluence.
5	Prepare Final Presentation Slides	Ensure that the final presentation slides have been added to Confluence and Github.
6	High-Fidelity Digital Prototype and Data Sample	Ensure the industry partner has access to the final version of the digital prototype and can run it independently after the end of the teaching semester. Generate a document with SAMPLE DATA. The industry partner must simulate tasks and scenarios in the final digital prototype.
7	Deploy Product	Ensure the product is deployed and an URL is available on Github README so the client can access the current version of the software.

Trello link: <https://trello.com/invite/b/84x4mp8c/ATTIifa39864413f2e735454ecf3c6e55cbadB4161EB1/sprint4>

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: 

Power: 

Social: 

Personality

Introvert	Extrovert
	
Analytical	Creative
	
Loyal	Fickle
	
Passive	Active
	

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.

Preferred Channels

Social Media

Mobile

Email

Traditional Ads


Influencers



Sophia Isabelle Martinez - Parent of Preschool Children



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

Kind Caring Passionate Empathetic

Goals

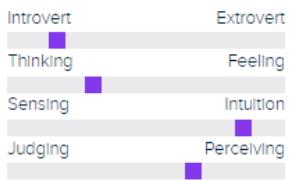
- Let her child Mia get accompanied
- Keep her work-life balance
- Improve Mia's handwriting skills

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

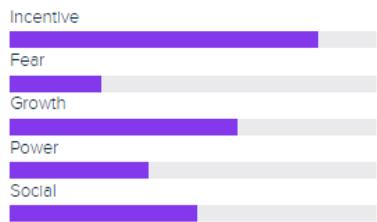
Personality



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.

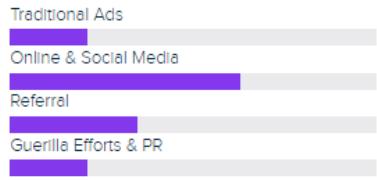
Motivation



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**

Hardworking Organized
Practical Protective

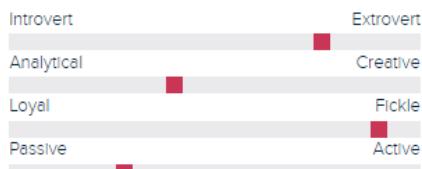
Brands & Influencers



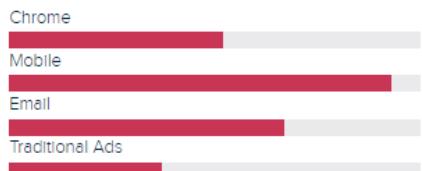
Bio

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



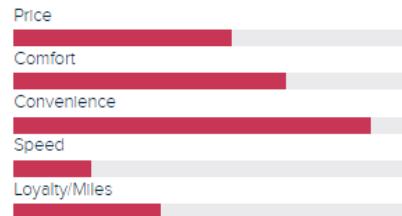
Goals

- 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



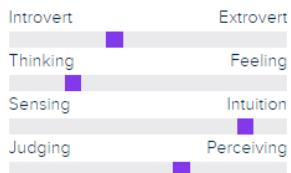
Alex Thompson - Maintainer



"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

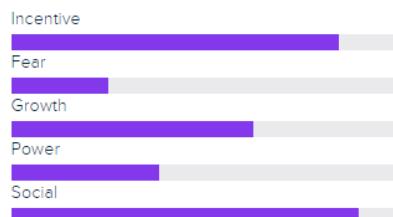
Bio

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



User Stories

Epic (Features)	ID	Owner	User Stories	Story Points	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.	3	Should have	Assigned	No
Nao robot functionalities	2	Preschool children	As a preschool child, I would like to ask random questions to an NAO robot rather than improve my handwriting so that it makes me feel enjoyable.	5	Could have	Assigned	No
	3	Preschool children	As a preschool child, I want to communicate with Naobot through audio input so that it can make my learning more efficient.	8	Must have	Assigned	No
	4	Preschool children	As a preschool child, I want to have my personalized collection of words in which the Naobot can generate relevant practice vocabulary based on my personal needs so that I can be more interested in learning.	8	Must have	Assigned	No
	5	Preschool child	As a preschool child, I would like to improve my handwriting so that I can write letters or words more fluently than the former version.	5	Must have	Assigned	No
	6	Preschool child	As a preschool child, I want the NAO robot to move its arm and show how to correctly write the characters so that I can learn how to write.	5	Must have	Assigned	No
	7	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.	1	Could have	Assigned	No
	8	Parents of the 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
	9	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.	1	Could have	Assigned	No
	10	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
ROS2 and Python enabled Port and APIs	11	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.	5	Must have	Assigned	No
Maintenance and follow-up project development	12	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
	13	Future Maintainer	As a future maintainer, I want clear annotations to facilitate my quick understanding of the code and subsequent work.	3	Should have	Assigned	No
	14	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
	15	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

The term MoSCoW itself is an acronym derived from the first letter of each of four prioritization categories: Must have, Should have, Could have, and Won't have.

- 1. Must have:** These are critical requirements that the project must deliver to be considered a success.
- 2. Should have:** These are important requirements but are not vital. While they are valuable and necessary for project success, they are not as critical as "Must have" requirements. In other words, if a "Should have" requirement is not included, the project can still be a success.
- 3. Could have:** These are desirable requirements but not necessary. These requirements will only be included if there is enough time and resources.
- 4. Won't have (or Would like to have):** These are low-priority requirements that will not be implemented in the current version of the product but could be considered for the future.

User acceptance criteria

- Sprint 1 User acceptance criteria
- Sprint 2 User acceptance criteria
- Sprint 3 user acceptance criteria

Sprint 1 User acceptance criteria

User story ID	Acceptance Criteria		
	Pre-condition(Given)	Action Description(When)	Acceptance Criterion(Then)
12	The future maintainer is preparing to add some new features to the robot.	Future Maintain is ready to work on the code structure of the project	<ul style="list-style-type: none"> -The code structure should be easy to understand and navigate. -Code should be properly commented on and documented to explain each module, function, and variable clearly. -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.
13	The future maintainer is preparing to add some new features to the robot.	Future Maintain is ready to read the code provided.	<ul style="list-style-type: none"> -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.
14	The future maintainer is preparing to add some new features to the robot.	Future Maintain is ready to read the readme file provided.	<ul style="list-style-type: none"> -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.

Sprint 2 User acceptance criteria

User story ID	Acceptance Criteria		
	Pre-condition(Given)	Action Description(When)	Acceptance Criterion(Then)
5	The robot is starting up normally, and the preschool child is ready to engage in a series of interactions with the robot.	preschool interacts with the robot through voice input and says it wants to practice vocabulary	<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.
8	The robot is starting up normally, and the parent is ready to engage in a series of interactions with the robot.	Parents log in through their personal accounts and view some relevant information.	<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.
15	The future maintainer is preparing to add some new features to the robot.	The future maintainer reviewed the code version used in the project.	<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.

Sprint 3 user acceptance criteria

User story ID	Acceptance Criteria		
	Pre-condition(Given)	Action Description(When)	Acceptance Criterion(Then)
2	The robot is starting up normally, and the preschool child is ready to engage in a series of interactions with the robot.	A preschool child asks questions to the naobot robot using its voice input feature.	<ul style="list-style-type: none"> -The NAO robot must accurately comprehend and process random questions from the user. -The NAO robot must provide a response that is correct and understandable. -The NAO robot should handle unclear or ambiguous queries, asking for clarification if necessary. -The interaction with the NAO robot should be enjoyable and engaging for the user.
3	The robot is starting up normally, and the preschool child is ready to engage in a series of interactions with the robot.	A preschool child attempts to converse with the naobot robot using its voice input feature.	<ul style="list-style-type: none"> -Naobot can accurately understand and process the child's voice instructions. -Naobot responds to the child's queries or commands within a reasonable time frame. -The child can ask Naobot to perform tasks related to learning. -Naobot's responses are child-friendly and easy for a preschooler to understand. -Naobot can recognise and appropriately handle unclear or ambiguous voice commands from the child. -Naobot's voice communication function works consistently without major errors or crashes. -Naobot can assist in teaching or reinforcing educational concepts.
4	The robot is starting up normally, and the preschool child is ready to engage in a series of interactions with the robot.	The preschool child has learned some words or shown interest in certain words, and they transmit this information to the naobot through voice input or hardware input.	<ul style="list-style-type: none"> -Naobot can create a personalized collection of words for each child. -Naobot can generate vocabulary exercises based on the words in the child's personalized collection. -The vocabulary exercises are age-appropriate and match the child's learning level. -The child can complete the vocabulary exercises using Naobot independently or with minimal assistance. -The system keeps track of the child's progress and adjusts the difficulty level of exercises according to the child's learning curve. -Naobot ensures the protection and privacy of the child's personalized collection of words, adhering to necessary data protection standards. -The child can access their personalized collection of words and practice anytime they want. -Parents or guardians can oversee the child's learning progress and adjust settings as necessary.
6	The robot is starting up normally, and the preschool child is ready to engage in a series of interactions with the robot.	The preschool child wants to learn how to write certain words through interaction with the naobot.	<ul style="list-style-type: none"> -The NAO robot must be able to replicate the motion of writing specific characters accurately. -The NAO robot's demonstration must be clear and easily visible to the user. -The user should be able to mimic the NAO robot's motion and write the character correctly. -The user must find the NAO robot's demonstration helpful in learning to write characters.

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](#)
- [Week8 \(Sprint2\) Code Review](#)
- [Week 10 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:

Python 3.10 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.

Python 2.7 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/download/releases/2.7/>

Download the Windows installer for Python 2.7 by clicking on the appropriate link (either [Windows x86](#) or [Windows x86-64](#), depending on your system).

Run the installer and follow the on-screen instructions.

Make sure to check the box that says "Add Python 2.7 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/download/releases/2.7/>

Download the macOS installer([Mac Installer](#) or [32-bit Mac Installer](#)) for Python 2.7.

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

Linux (Ubuntu, Debian):

Python 2.7 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 python2
```

After completing the installation, you can check the Python version by running `python -V` 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.

Docker Setup

Ubuntu

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

macOS

```
sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
sudo systemctl start docker
```

Windows : Please follow the [installation manual](#) to install the desktop version of Docker

Week8 (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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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	Important	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%

Week 10 code review

Date:

12/5/2023

Participants:

Minyi Chen, Sijia Pei, Yuhang Wang

Issues identified, ChatGPT feedback and comments on feedback:

ORGANISING YOUR CODE REVIEW (A FEW DAYS/HOURS BEFORE MEETING)								
Name and local of artifact (on Github) to be reviewed:								
What to be reviewed?	Naoqi2robot files, Nao_writing files.							
When's the code review meeting happening:	12/05/2023							
Reviewers:	Minyi Chen, Yuhang Wang, Sijia Pei							
CODE REVIEW MEETING (STARTING THE MEETING)								
Team:	NA-Redback							
Date:	12/05/2023							
Time:	<13:00:00>							
Facilitator:	Minyi Chen, Yuhang Wang, Sijia Pei							
Reviewers:	Minyi Chen, Yuhang Wang, Sijia Pei							
CODE REVIEW (DURING MEETING TIME)								
Item	Artifact (on GitHub)	Location (where the issue was found in the reviewed artifact?)	Severity	Type	Defects Category	Description	Fixed by the author?	Verified by the Moderator?
1	tests/Naoqi2robot /movearm.py	The angle_a and angle_b	Trivial	Improvement	Documentation Defects (Naming)	The variable and function names in the script are generally clear and descriptive. However, angle_a and angle_b in calculate_joint_angles function might be made more descriptive.	No	No
2	tests/Naoqi2robot /movearm.py	general	Moderate	Improvement	Documentation Defects (Comment)	The code lacks comments entirely. It would benefit from comments explaining the purpose and functionality of functions and key operations within the script.	No	No
3	tests/Naoqi2robot /movearm.py	calculate_joint_angles function	Moderate	Improvement	Logic Defects (Compute)	The computation in the script seems to be correct. However, the calculation for angle_a and angle_b in calculate_joint_angles function should be reviewed for the specific context.	No	No
4	src/nao_ros2_ws/src /nao_writing/nao_writing /audio_chat.py	audio function	High	Issue	Check Defects (Check User Input)	Add input validation to ensure the user provides a non-negative integer	No	No
5	src/nao_ros2_ws/src /nao_writing/nao_writing /audio_chat.py	general	Moderate	Improvement	Documentation Defects (Comment)	The code lacks sufficient comments, which makes it harder for others to understand its logic. It would be beneficial to add comments describing the purpose and functionality of each method and major code block.	No	No
6	src/nao_ros2_ws/src /nao_writing/nao_writing /audio_chat.py	if "let's practise" in user_input or "let's practice" in user_input:	Moderate	Improvement	Structure Defects (Duplication)	Some of the operations in the if "let's practise" in user_input or "let's practice" in user_input: block and the else: block are duplicated and could be consolidated to improve code readability and maintainability.	No	No
7	src/nao_drawing_board /src/components /CanvasManager.tsx	general	Trivial	Improvement	Documentation Defects (Comment)	There are no comments in the code. Although the code is quite self-explanatory, adding comments, especially for the functions would be helpful for better understanding.	No	No
8	src/nao_drawing_board /src/components /CanvasManager.tsx	ALPHABETS_LOWER	Trivial	Improvement	Visual Representation Defects (Long Line)	There are a few long lines in the code, such as when defining ALPHABETS_LOWER. Splitting these lines could improve readability.	No	No

9	src/nao_drawing_board /src/components /CanvasManager.tsx	General	Me di um	Inv esti gate	Check Defects (Check User Input)	The code does not appear to directly handle user input. The messages from ROS2 topics and services are assumed to be valid.	No	No
10	src/nao_ros2_ws/src /nao_writing/nao_writing /nao_writer_naoki.py	general	Me di um	Imp rov em ent	Structure Defects (Dead Code)	The code includes some commented-out parts. If these are not necessary, they should be removed.	No	No
11	tests/Naoqi2robot /nao_bot_controller.py	"DRAW" command	Me di um	Inv esti gate	Check Defects (Check User Input)	The script does not seem to handle errors in user input robustly. For example, if the "DRAW" command was sent with malformed content, the script would likely crash.	No	No
12	tests/Naoqi2robot /nao_bot_controller.py	general	Me di um	Imp rov em ent	Structure Defects (Dead Code)	There are some commented out lines of code that could be removed if not needed.	No	No
13	tests/Naoqi2robot /nao_bot_controller.py	general	Me di um	Imp rov em ent	Documentation Defects (Comment)	There is a lack of comments in the code, which makes it harder for others to understand the purpose of each function and the flow of the program.	No	No
14	tests/Naoqi2robot /nao_bot_controller.py	Line 37, line 39	Tr ivi al	Inv esti gate	Visual Representation Defects (Long Line)	Some lines are too long and could be broken down for improved readability.	No	No

END OF CODE REVIEW MEETING

Number of severe/critical errors:	1							
Number of medium errors:	9							
Number of trivial errors:	4							
Total inspection time (hs):	120mins							

Sprints Planning and Review

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

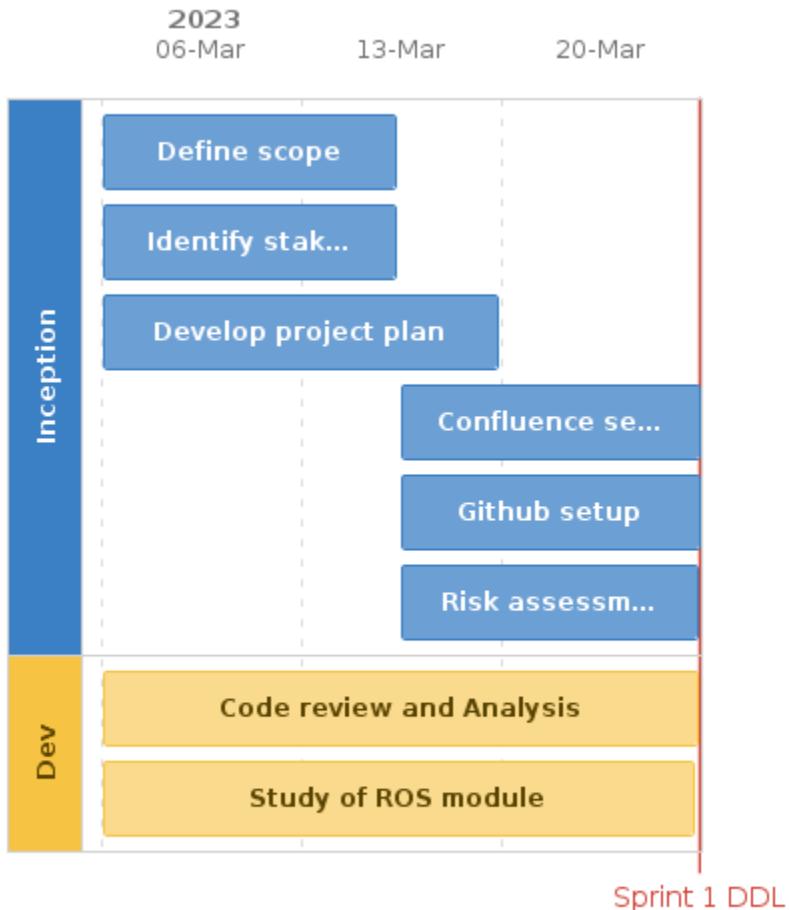
- [Sprint 1 Plan](#)
- [Sprint 1 Review](#)
- [Sprint 2 Plan](#)
- [Sprint 2 Review](#)
- [Sprint 3 Plan](#)
- [Sprint 3 Review](#)
- [Sprint 4 Plan](#)
- [Sprint 4 Review](#)

Sprint 1 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.



Sprint 1 Review

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:

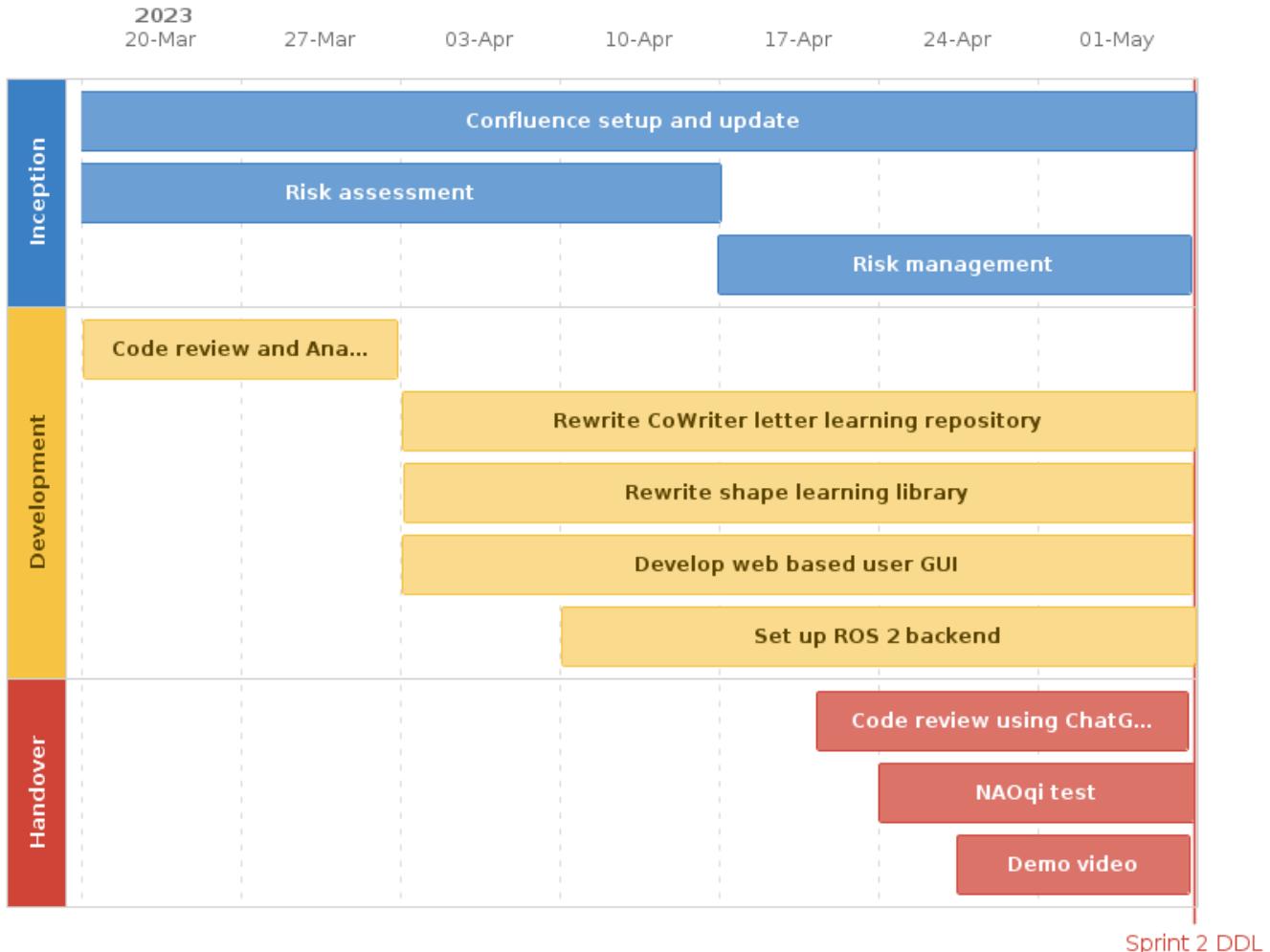
- Team members worked well, communicated effectively.
- Each team member attended booked meetings and shared their thoughts.

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.

Sprint 2 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.



Sprint 2 Review

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:

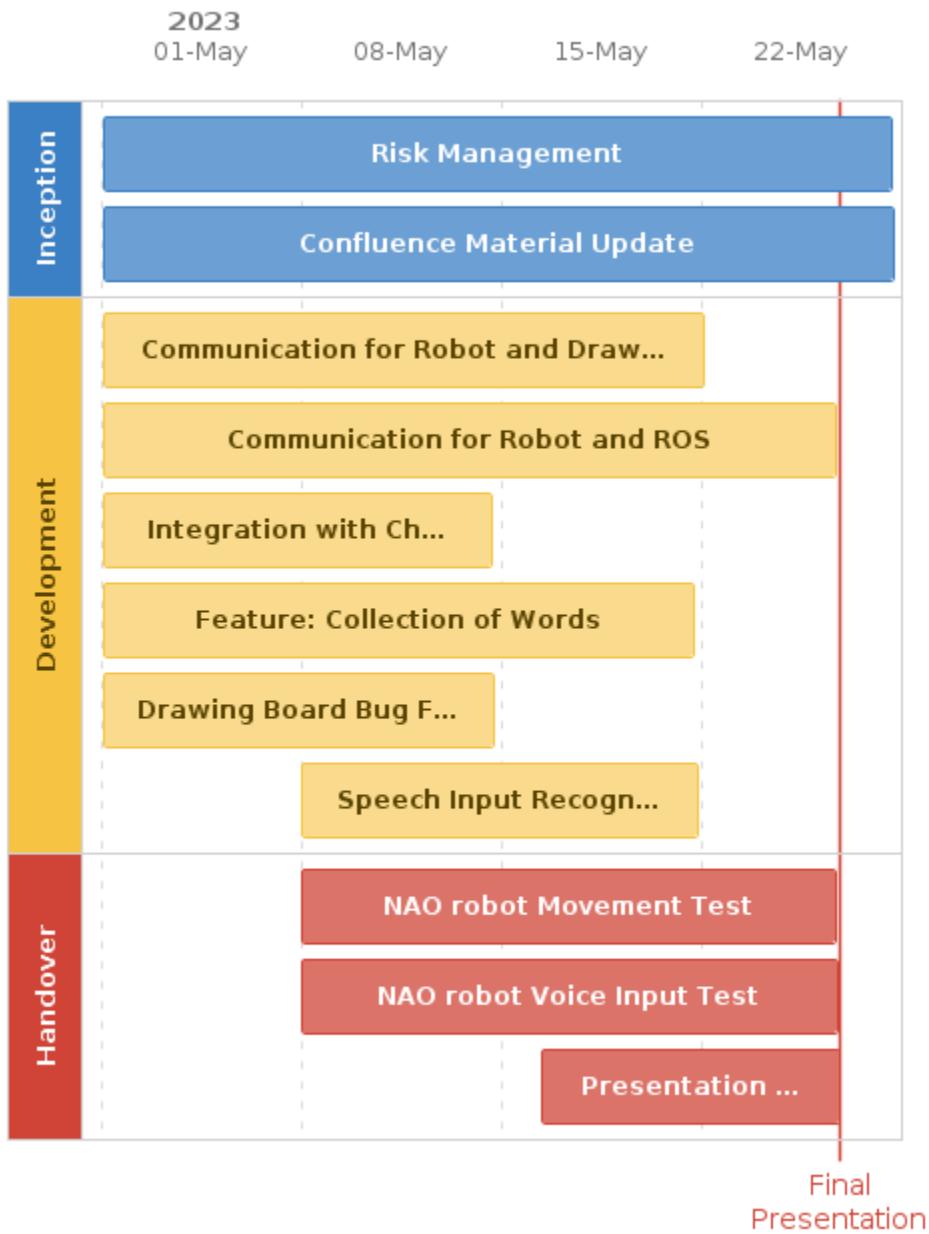
- Each team member has shown great efficiency.
- Developers completed the demo for Sprint 2 in cooperation.

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.

Sprint 3 Plan

- In the inception part of our sprint, we'll concentrate on robust risk management and updating our Confluence materials for a solid project foundation.
- In the development stage, we're focusing on technology enhancements. We'll establish communication pathways between the NAO robot, ROS module, and drawing board, integrate ChatGPT, complete the 'Collection of Words' feature, and implement the speech input recognition for the robot.
- In the handover phase, we'll undertake comprehensive tests on the NAO robot's movement and voice input, followed by preparations for our final presentation, summarizing our project milestones.



Sprint 3 Review

Attendees:

- @Wafa Johal
- @Sijia Pei
- @Chengfeng Zhang
- @Yuhang Wang
- @Minyi Chen
- @Da Zhang

Sprint Goal:

- Risk management
- Confluence material update
- Establish the communication between NAO robot and drawing board
- Establish the communication between NAO robot and ROS module
- Integration with ChatGPT
- Collection of words
- Drawing board bugs fix
- Implement the speech input recognition to the NAO robot

Sprint reflection on the completed sprint:

Accomplishment:

- Establish the communication between NAO robot and drawing board
- Establish the communication between NAO robot and ROS module
- Integrate the robot with ChatGPT
- Complete the feature: collection of words
- Update the drawing board
- Implement the speech input recognition

Not finished:

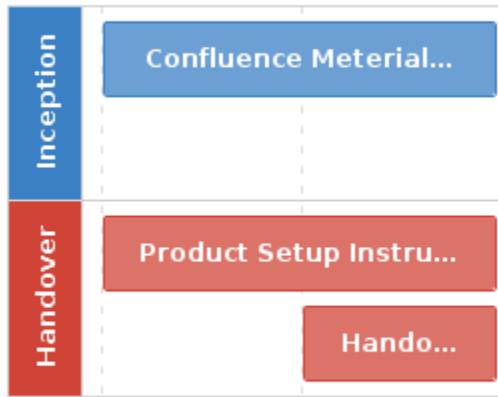
- More interactive interface for the feature Selection of words
- Comprehensive model for the robot that can record user's handwriting and improve its own

Team performance:

- Our client highly commended for the robot movements
- Our client was satisfied with the integration of NAO robot and ChatGPT.

Sprint 4 Plan

2023
29-May 05-Jun



- In the final sprint, our team will focus on closing the project and updating our Confluence and GitHub materials.
- In the development stage, all work has been completed in the previous sprint, and there are no further development plans in this sprint.
- In the handover phase, the team plans to add Product Setup Instruction that are convenient for other development teams to refer to in the future, as well as a video that can showcase project functionality for learning.

Sprint 4 Review

Attendees:

- @Wafa Johal
- @Sijia Pei
- @Chengfeng Zhang
- @Yuhang Wang
- @Minyi Chen
- @Da Zhang

Sprint Goal:

- Confluence material update
- Create environment setup instruction
- Create product setup instruction

Sprint reflection on the completed sprint:

Accomplishment:

- Complete a clear and detailed environment setup instruction
- Modified the Confluence and Trello board to the final version

Team performance:

- Each team members have engaged into the process of product setup instruction

Test process

This part covers our project test steps for each Sprint:

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

Sprint1

User story ID	Target	Pre-condition	Testing steps	Status
12	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
13	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
14	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

Sprint2

User story ID	Target	Pre-condition	Testing steps	Status
5	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
8	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
15	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. -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.) -Try running various features and check if the results match the expectations and record the bug(if it occurred). -Verify that the code is written as required in the latest programming language version (Python3, ROS2). -Document any problems or improvements related. 	Assigned

Sprint3

User story ID	Target	Pre-condition	Testing steps	Status
2	Ensure Naobot can accurately receive voice commands from preschool children and generate responses according to the voice input.	-The Nao bot starts normally -The user has a normal language ability	-The child or parent initiates interaction with the Nao robot using the designated command. -The child or parent asks various random questions. -Verify that the Nao robot processes the questions accurately and responds within an acceptable time frame. -Check the appropriateness and understandability of the Nao robot's responses. -Check whether the Nao robot correctly handles unclear or ambiguous queries. -Verify that interactions with the nao robot are enjoyable and engage the interest of the child or parent.	Assigned Assigned Assigned Assigned Assigned Assigned
3	Ensure Naobot can accurately receive, process and respond to voice commands from preschool children.	-The Nao bot starts normally -The child has a certain language foundation -The child is ready to give voice commands.	-Have the child activate Naobot using the designated voice command. -The child issues various voice commands related to learning (e.g., asking to recite the alphabet, count, or tell a story). -Verify that Naobot processes the commands accurately and responds within the agreed time frame. -Check the appropriateness and understandability of Naobot's responses. -Check whether Naobot correctly handles unclear or ambiguous voice commands. -Ensure that there is a mechanism for parental control to limit the child's interaction with Naobot.	Assigned Assigned Assigned Assigned Assigned Assigned
4	Ensure Naobot can create a personalized word collection for the child and generate relevant vocabulary exercises.	-The Nao bot starts normally -The child has a certain language foundation -Naobot has interacted with the child and gathered enough data about their vocabulary usage and learning preferences to generate a personalized word collection.	-Interact with Naobot using a simulated child's vocabulary and observe if it can create a personalized word collection based on this interaction. -Verify that Naobot generates a word collection that matches the child's vocabulary usage and learning level. -Check that Naobot generates vocabulary exercises based on the words in the child's personalized collection. -Have the child attempt to complete the vocabulary exercises, verifying whether they suit the child's age and learning level. -Ensure that Naobot provides appropriate feedback upon completion of the exercises. -Verify that the system tracks the child's progress and adjusts the difficulty of exercises over time. -Check that the child can access their personalized word collection and vocabulary practice anytime. -Ensure that the child's personalized collection of words is securely stored and adheres to data protection standards.	Assigned Assigned Assigned Assigned Assigned Assigned
6	Ensure the Naobot can demonstrate how to write the characters and show the child.	-The Nao bot starts normally -The child can follow the movement shown by the Naobot	-The child requests the NAO robot to demonstrate how to write a specific character. -Check if the NAO robot can move its arm to mimic the process of writing the character.	Assigned Assigned

		-Verify that the NAO robot's arm movements accurately represent the strokes and sequence involved in writing the character.	Assigned
		-Check whether the child is able to understand and mimic the NAO robot's arm movements.	Assigned
		-Verify that the NAO robot's demonstration is helpful and effective in teaching the child how to write the character.	Assigned

Risk Assessment and Management

In the initial phase of our sprint, a crucial component is the execution of thorough risk assessment and management. We understand the importance of proactively identifying potential challenges that could affect our project timeline, quality, or resources. Therefore, we have a dedicated team committed to assessing these risks and developing strategies to mitigate them effectively. Our risk management process involves continuous monitoring to ensure that any new risks are promptly identified and addressed, thus maintaining the integrity and progress of our project. This thorough approach helps us stay on track, ensures we're prepared for any eventualities, and bolsters the likelihood of our project's success.

- [Risk Assessment](#)
- [Risk Management](#)

Risk Assessment

Specific risk:

Risk ID	Risk Type	Owner	Description	Probability (0-1)	Impact (1-10)	Exposure	Justification
1	Project	Minyi Chen, Yuhang Wang, Sijia Pei	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.
2	Product	Minyi Chen, Yuhang Wang, Sijia Pei	Due to the upgraded CoWriter which integrated with ChatGPT, the Conversational interactions may be incorrect or incompletem 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.
3	Product	Miryi Chen, Yuhang Wang, Sijia Pei	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.
4	Project	Minyi Chen, Yuhang Wang, Sijia Pei	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.
5	Product	Minyi Chen, Yuhang Wang, Sijia Pei	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.

Generic risk:

Risk ID	Risk Type	Owner	Description	Probability (0-1)	Impact (1-10)	Exposure	Justification
1	Project	Chengfen g Zhang, Da Zhang	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.
2	Project	Chengfen g Zhang	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.
3	Project	Chengfen g Zhang, Da Zhang	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.
4	Product	Minyi Chen, Yuhang Wang, Sijia Pei	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.
5	Product	Chengfen g Zhang, Yuhang Wang	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.

Risk Management

Specific risk:

Risk ID	Response	Response Strategy	Resource Required
1	Mitigation	<ul style="list-style-type: none"> Analyse 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	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	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	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	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. 	UX/UI designers to develop a user-friendly interface

Generic risk:

Risk ID	Response	Response Strategy	Resource Required
1	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	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	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	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.

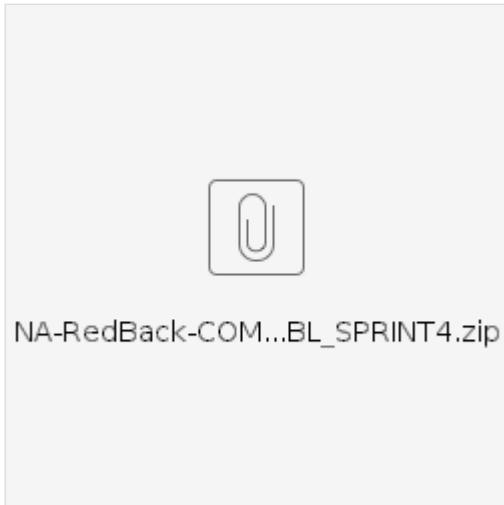
5	Mitigation	<ul style="list-style-type: none">• Develop a comprehensive test plan that covers functional, performance, and security testing.• Allocate sufficient time and resources for thorough testing during each stage of the project.	<ul style="list-style-type: none">• Skilled testers or developers 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
- Final Presentation
- Environment Setup Instruction
- Product Setup Instruction
- Sprint4 Final Demonstration video

Handover zipfile:



Sprint 2 process demo video

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

Final Presentation

Presentation details

Project Title: Team ChatGPT and Nao Robot (code: NA)

Presentation Time: 5/25/2023 -12:15:00 PM

Attendees:

- @Max Plumley
- @Luke Rosa
- @Sijia Pei
- @Chengfeng Zhang
- @Yuhang Wang
- @Minyi Chen
- @Da Zhang

Presentation slides: [google_slides_NAO robot](#)

Participants	Presentation Content
Chengfeng Zhang	<ul style="list-style-type: none">• Introduction to the project• Components of the project
Minyi Chen	<ul style="list-style-type: none">• Introduction to the structure• Live demo for NAO robot handwriting
Sijia Pei	<ul style="list-style-type: none">• Presentation of speech input recognition• Live demo for NAO robot integration with ChatGPT
Yuhang Wang	<ul style="list-style-type: none">• Implementation of Collection of Words• Live demo for NAO robot feature: Collection of Words
Da Zhang	<ul style="list-style-type: none">• Feature outlook• Conclusion

Environment Setup Instruction

Product Environment setup instructions:

Python 3.10 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.

Python 2.7 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/download/releases/2.7/>

Download the Windows installer for Python 2.7 by clicking on the appropriate link (either [Windows x86 MSI Installer](#) or [Windows x86-64 MSI Installer](#), depending on your system).

Run the installer and follow the on-screen instructions.

Make sure to check the box that says "Add Python 2.7 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/download/releases/2.7/>

Download the macOS installer([Mac Installer](#) or [32-bit Mac Installer](#)) for Python 2.7.

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

Linux (Ubuntu, Debian):

Python 2.7 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 python2
```

After completing the installation, you can check the Python version by running `python -V` 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.

Docker Setup

Ubuntu

```
sudo apt-get update  
  
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

macOS

```
sudo yum install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin  
  
sudo systemctl start docker
```

Windows : Please follow the [installation manual](#) to install the desktop version of Docker

Product Setup Instruction

Setup NA-RedBack Repository

1. Set Git username information

```
git config --global user.name "github's Name"  
git config --global user.email "github@xx.com"
```

2. Configure ssh key

```
ssh-keygen -t rsa -C "github@xx.com"  
cd .ssh
```

Enter GitHub, in Your Profile, click on SSH Keys on the left, and then click on the Add SSH key button on the right

Among them, fill in the title casually, copy the content from id_rsa.pub to Key, click the Add Key button below, and then enter the following command on the command line to check if the configuration is successful:

```
ssh -T git@github
```

If the welcome statement is returned, the configuration is successful. If it fails, please check the previous operation process for any errors

3. Clone Repository

```
git clone git@github.com:COMP90082-2023-SM1/NA-RedBack.git
```

4. Use Docker to build project

```
cd NA-RedBack/src/nao_drawing_board/  
docker compose -f docker-compose.dev.yml up --build
```

Note that each time a new repository is installed, a new build is required, or you may install Node.js and run the dev server locally

5. You may now visit the web applications with the address of the dev server

Launch Naoqi Controller

Since the naoqi SDK is written in python 2.7, you may only import it with python 2.7.

1. Prepare python 2.7 venv
2. Get the ip address and port number of the Nao Robot and edit the code
3. Run the file tests/Naoqi2robot/nao_bot_controller.py

Now the robot should be connected and ready to accept input from the drawing board and the chatbox.

Launch ROS2 Nodes

1. Navigate to the folder src/nao_ros2_ws
2. Build the project with command colcon build
3. Source the project with command source install/setup.bash
4. Run the launch file with ros2 launch launch/nao_writing.launch.py

Now, four nodes will be launched, which are input_interpreter, learning_word, nao_writer_naoqi, audio_chat

You should also notice a chat box also pops up on the screen.

Launch Naoqi Controller

Since the naoqi SDK is written in python 2.7, you may only import it with python 2.7.

1. Prepare python 2.7 venv
2. Get the ip address and port number of the Nao Robot and edit the code

3. Run the file `tests/Naoqi2robot/nao_bot_controller.py`
Now the robot should be connected and ready to accept input from the drawing board and the chatbox.

OpenAI key setup

Since this project incorporates chatGPT, an openAI key needs to be set up to invoke the GPT model. Here are the steps to set up openAI key:

1. Create an OpenAI account. Visit the OpenAI [website](#) and follow the instructions to create a new account.
2. Log into the [website](#) to set up billing information for using OpenAI API.
3. Log into the [API keys website](#) to create new secret key.
4. When the key is created, copy and paste it to save it. The complete key may only appear once.
5. In the project folder, navigate to: `src/nao_ros2_ws/src/nao_writing/nao_writing/audio_chat.py`, replace the value of the global variable: `api_key` in the `audio_chat.py` file with the key you just obtained in step3

Choregraphe installation guidance

Choregraphe is a software for programming and monitoring SoftBank Robotics' Nao and Pepper robots. Here are the steps on how to install and launch Choregraphe on Linux and Windows systems.

Linux:

1. Go to SoftBank Robotics' official website and download the Choregraphe suite for Linux systems.
2. Open your Linux terminal.
3. Navigate to the folder containing the `.tar.gz` file you downloaded. You can do this using the `cd` command.
4. Extract the downloaded `.tar.gz` file using the `tar -xvzf [filename].tar.gz` command.
5. Navigate into the extracted directory using the `cd` command.
6. To start Choregraphe, you can run the choregraphe script in the bin folder. The command would be `./bin/choregraphe`.

Windows:

1. Go to SoftBank Robotics' official website and download the Choregraphe suite for Windows systems.
2. After downloading the `.exe` file, navigate to the download location.
3. Double-click on the `.exe` file to launch the installer.
4. Follow the instructions provided by the installer.
5. Once installed, you should be able to launch Choregraphe either from the Start menu or the desktop shortcut if you chose to create one during installation.

To switch robot model:

Navigate to the "View" tab, find and select the "Robot View" option, then navigate to the top-right corner. You should see a drop-down menu where you can select the robot model you want to use.

Use the drawing board

The drawing board consisted of two button and one input box, the two buttons are 'Done', 'Clear'.
'Done' button will send the drawing strokes to backend and 'Clear' button will clear all inputs.

1. Input the word you want to write in the input box, you should see equal numbers of canvas being created for each char
2. Draw each char as suggested by the char on top of each canvas, you may use the 'Rewrite' button to clear the corresponding canvas
3. Click 'Done' to send the drawings to backend

Audio input, conversation with robot and collection of words guidance

The process for users to input with their own microphone is as follows:

1. The user first clicks on "record". There might be some warnings in the terminal after clicking, but it won't affect the process (this might be due to issues between the Ubuntu virtual environment and the Windows audio device).
2. After seeing the warning, the user starts to speak. When finished, the user waits for a second or two.
3. The "stop" button will light up, indicating that the parsing of user input is complete. The user clicks "stop", and the parsed user input will appear in the prompt box. Sometimes the parsing might not be accurate, in which case the user can manually modify the input.
4. The user clicks "submit" to submit the question and wait for the reply.

The implementation logic of "collection of words" is almost the same as a regular chat. The user mentions a topic of interest, and then ChatGPT returns 5 related words for the user to choose and practice. However, the difference is that, whether the user types in the input or speaks it, the phrase "let's practice" or "let's practise" must appear in the user's input. Only then will the program enter the "collection of words" mode. An example of collection of words user input request: Hi, I'm interested in space. **Let's practice** some english words about space!

Sprint4 Final Demonstration video

Link to the demonstration video: <https://youtu.be/fyKoKvxCB0M>