

Product and Sprint Backlogs (Epics and User Stories)

Contents

- [Contents](#)
- [Project Essentials: Acceptance Criteria](#)
- [Project Essentials: MoSCoW Prioritisation](#)
- [Project Essentials: T-Shirt Size Estimation and Story Points](#)
- [Epics Overview and Sprint Alignment](#)
- [The Product Backlog \(Entire Project\)](#)
- [Sprint 2 Backlog](#)
- [Sprint 3 Backlog](#)
- [Sprint 4 Backlog](#)
- [Delivered and Undelivered Stories](#)

Project Essentials: Acceptance Criteria

We are using the Behaviour Driven Development (BDD) format for acceptance criteria. BDD is a software development methodology which emphasises collaboration between developers, testers, and non-technical stakeholders. It focuses on specifying the expected behaviour of a software feature using a common language that is easily understood by everyone involved in the project.

The BDD format consists of "Given", "When", and "Then" statements, describing the context, the action, and the expected outcome, respectively. This format helps to ensure that the software meets the desired behaviour and requirements of the user stories. It also facilitates clear communication among team members and stakeholders about the expectations and desired outcomes for each feature, minimising misunderstandings and reducing the risk of producing software that does not meet the users' needs. By using BDD in our acceptance criteria, we can better align the development process with the goals and priorities of the users and stakeholders, ultimately leading to a more successful and valuable software product.

Project Essentials: MoSCoW Prioritisation

Priority of user stories in this project is defined in terms of MoSCoW criteria. MoSCoW stands for Must have, Should have, Could have, and Won't have. This method helps project teams and stakeholders to identify the most important features to focus on and allocate resources more effectively.

In the context of the CoWriter project, the MoSCoW criteria can be defined as follows:

Must have	These are critical requirements that must be implemented for the project to be successful. For the CoWriter project, this includes updating the software and its dependencies to a newer version of Python, integrating ChatGPT as a dialogue manager, and ensuring accurate audio-to-text parsing. Without these features, the CoWriter software would not function correctly or meet the core needs of educators and children.
Should have	These are important requirements that significantly impact the user experience, but the project can still function without them. In this project, this might involve designing an adaptable robot character, or providing personalised word suggestions. While not absolutely critical, these features will greatly improve the effectiveness and user experience of the CoWriter program.
Could have	These are desirable features that would enhance the project but are not necessary for its success. In the CoWriter project, this could include additional customisation options for the robot's character. These features would provide added value, but the software could still function effectively without them.
Won't have	These are low-priority features that are not planned for the current project scope but may be considered for future development. For the CoWriter project, this might involve developing a new web app to capture pen data, or expanding the software to support other educational activities beyond handwriting. While these features could be beneficial, they are not essential for the project's success and can be deferred to a later stage.

By using the MoSCoW criteria, the project team can better prioritise tasks and focus on delivering the most important features first, ensuring that the CoWriter software meets the essential needs of educators and children while also providing a positive user experience.

Project Essentials: T-Shirt Size Estimation and Story Points

The T-shirt size estimation technique simplifies the process of determining task sizes and time commitments for individual team members. Using the categories XS, S, M, L, and XL, this method breaks down tasks into manageable units and facilitates clear communication within the team.

What follows is an outline of each size in terms of time commitment for a single person, and the associated number of story points. Note that a day is considered roughly 8 hours; as the developers are students with other commitments, it is quite common for a task with an estimated effort of one day to take longer than this to complete, similar remarks apply to longer periods.

XS	XS tasks require minimal time and effort, taking up to a couple of hours for one person to complete.	1 story point
S	Small tasks generally take a single person a few hours to a day to complete. 2 story points.	2 story points

M	Medium-sized tasks typically take one person up to a couple of days to complete.	3 story points
L	Large tasks require a few days to a week of work for a single person. Whenever possible, multiple assignees should work on large tasks to ensure timely completion	5 story points
XL	Extra-large tasks represent the most complex and time-consuming work. Due to their complexity, XL tasks should always be decomposed into two or more medium or larger tasks, allowing team members to distribute the workload and optimise project management and productivity	8-10+ story points

Epics Overview and Sprint Alignment

Epics:

Epic 1	Port legacy code to Python 3.x and ROS Noetic
Epic 2	New UI: web app compatible with Wacom and iOS tablets (Deemed Out-of-Scope on April 5th, 2023)
Epic 3	Integrate ChatGPT as a dialogue manager
Epic 4	Prompt and design characters for the robot
Epic 5	Parse audio into text
Epic 6	Provide clear and extensive documentation and usage instructions

The CoWriter project is divided into several epics, each addressing a specific aspect of the project. To ensure effective development and timely delivery, these epics are distributed across three sprints.

The alignment of sprints with epics is as follows:

Sprint 2	Initially, the focus was on Epics 1 & 2. However, Epic 2 was deemed out-of-scope on April 5th, 2023. Therefore, Sprint 2 will now concentrate solely on Epic 1, porting the legacy code to Python 3.x and ROS Noetic.
Sprint 3	During the third sprint, the team will focus on Epics 3, 4, and 5, aiming to integrate ChatGPT as a dialogue manager, design engaging characters for the robot, and develop the ability to parse audio into text.
Sprint 4	The final sprint will emphasise completing Epic 6, which involves providing clear and extensive documentation and usage instructions. Additionally, any remaining Must-Have tasks in the product backlog will be addressed during this sprint to ensure a comprehensive and polished project outcome.

The Product Backlog (Entire Project)

Epic	Story ID	User	Story	Story Points	Acceptance Criteria	MoSCoW Priority
1	US01	Dani Muhamad	As a Developer, I want the Letter Learning Interaction ROS package to be updated to a newer version of Python, so that it remains compatible with modern systems and libraries.	20	Given the Letter Learning Interaction ROS package has been updated to Python 3.8.10, When the test cases are ran, Then the test cases should all pass. When the CoWriter program is executed, Then the program should run without Letter Learning Interaction causing errors.	Must Have
1	US02	Dani Muhamad	As a Developer, I want the Nao Writing package, on which CoWriter depends, to be updated to a newer version of Python, so that it remains compatible with modern systems and libraries.	5	Given the Nao Writing ROS package has been updated to Python 3.8.10, When the test cases are ran, Then the test cases should all pass. When the CoWriter program is executed, Then the program should run without Nao Writing causing errors.	Must Have

1	US03	Dani Muham mad	As a Developer, I want the Shape Learning library, on which CoWriter depends, to be updated to a newer version of Python, so that it remains compatible with modern systems and libraries.	5	Given the Shape Learning library has been updated to Python 3.8.10, When the CoWriter program is executed, Then the program should run without Shape Learning causing errors.	Must Have
1	US04	Dani Muham mad	As a Developer, I want the Choose Adaptive Words package, which provides the current CoWriter UI, to be updated to a newer version of Python, so that it remains compatible with modern systems and libraries.	9	Given the Choose Adaptive Words ROS package has been updated to Python 3.8.10, When the CoWriter program is executed, Then the program should run without Choose Adaptive Words causing errors.	Should Have
1	US05	Dani Muham mad	As a Developer, I want the CoWriter program and its dependencies to be updated and built using the latest version of ROS (Noetic), so that it remains compatible with libraries, and receives ongoing support and updates beyond May 2023.	10	Given the CoWriter packages have been updated to use ROS Noetic, When catkin_make is ran in the root folder of the workspace, Then the workspace should build without errors. When the CoWriter program is executed using ROS Noetic, Then the program should run without errors.	Must Have
1	US06	Lucía Castro	As an Educator, I want to see that the ported version of CoWriter works on a simulated robot before having children interact with a real robot, to ensure the updated software provides a safe and effective learning experience for the students.	11	Given the CoWriter packages have been ported to Python 3.8.10 and ROS Noetic, When CoWriter is ran with a virtual Nao using Choregraphe, The program should demonstrate the expected functionality.	Must Have
2	US07	Lucía Castro	As an Educator, I want CoWriter to use a web app that can capture pen data from Wacom tablets and iOS devices, so that I can easily collect writing samples from various devices used by my students.	Undeter mined	Out of scope as of April 5th 2023, no acceptance criteria required.	Won't Have
2	US08	Zoe Suzuko	As a Younger Child, I want to use a web app to submit my writing samples from my iOS device, so that I can participate in the CoWriter program using my preferred device.	Undeter mined	Out of scope as of April 5th 2023, no acceptance criteria required.	Won't Have
2	US09	Dylan Rees	As an Older Child, I want to use a web app to submit my writing samples from my Wacom device, so that I can have a seamless experience participating in the CoWriter program.	Undeter mined	Out of scope as of April 5th 2023, no acceptance criteria required.	Won't Have
3	US10	Lucía Castro	As an Educator, I want the robot to use ChatGPT as a dialogue manager, so that it can engage in more natural conversations with children and reduce the need for my intervention.	16	Given the robot has ChatGPT integrated as a dialogue manager, When the robot receives a spoken prompt, Then the robot should generate a natural and contextually appropriate response using ChatGPT.	Must Have
3	US11	Lucía Castro	As an Educator, I want the robot to respond to a wider range of prompts so that the activity can be adapted to different skill levels and needs.	2	Given the robot has been programmed to respond to an extended range of prompts, When the robot is in operation, Then the robot should be able to respond to various prompts suitable for different skill levels and needs.	Should Have
3	US12	Zoe Suzuko	As a Younger Child, I want the robot to provide personalized word suggestions during the CoWriter program, so that I can practice writing words that are relevant and engaging for me.	8	Given the robot provides personalized word suggestions using ChatGPT, When the developers test the robot's word suggestions with sample younger child profiles, Then the suggested words should be relevant and engaging for a younger audience.	Should Have

3	US13	Dylan Rees	As an Older Child, I want the robot to provide word suggestions tailored to my abilities, so that I can practice writing words which challenge me to improve.	3	<p>Given the robot provides word suggestions tailored to the abilities of older children using ChatGPT,</p> <p>When the developers test the robot's word suggestions with sample older child profiles,</p> <p>Then the suggested words should be appropriately challenging for an older user.</p>	Should Have.
4	US14	Lucía Castro	As an Educator, I want the robot's character to be adaptable, so that it can adjust its behaviour to suit the needs of each child.	3	<p>Given the robot's character is adaptable,</p> <p>When the developers test the robot with different profiles,</p> <p>Then the robot should adjust its behaviour to suit the needs of each profile.</p>	Could Have
4	US15	Zoe Suzuko	As a Younger Child, I want the robot's character to be personalized, so that it feels like it's designed specifically for me and helps me feel more comfortable during the CoWriter program.	2	<p>Given the robot's character is personalized,</p> <p>When the developers test the robot's character with sample younger child profiles,</p> <p>Then the robot should display a character that would make a younger audience feel comfortable using the CoWriter program.</p>	Could Have
4	US16	Dylan Rees	As an Older Child, I want the robot's character to be age-appropriate, so that it doesn't come across as childish and I feel good about needing to interact with it.	3	<p>Given the robot's character is age-appropriate,</p> <p>When the developers test the robot's character with sample older child profiles,</p> <p>Then the robot should display a character that would not come across as overly childish.</p>	Could Have
5	US17	Lucía Castro	As an Educator, I want the robot to be able to accurately parse audio into text, so that it can better understand and respond to the children's spoken words during the CoWriter program.	5	<p>Given the robot has been programmed to parse audio into text,</p> <p>When the developers test the robot's audio parsing with sample spoken words from children,</p> <p>Then the robot should correctly convert the spoken words into text and provide appropriate responses.</p>	Must Have
5	US18	Zoe Suzuko	As a Younger Child, I want the robot to accurately parse audio into text, so that it can understand my spoken words and provide appropriate feedback and guidance.	6	<p>Given the robot has been programmed to parse audio into text,</p> <p>When the developers test the robot's audio parsing with sample spoken words from younger child profiles,</p> <p>Then the robot should correctly convert the spoken words into text and provide feedback and guidance suitable for younger children.</p>	Must Have
5	US19	Dylan Rees	As an Older Child, I want the robot to accurately parse audio into text, so that it can understand my spoken words and engage in more meaningful and responsive conversations.	5	<p>Given the robot has been programmed to parse audio into text,</p> <p>When the developers test the robot's audio parsing with sample spoken words from older child profiles,</p> <p>Then the robot should correctly convert the spoken words into text and engage in meaningful and responsive conversations tailored to older children.</p>	Must Have

5	US20	Lucia Castro	As an educator, I want the robot to be able to distinguish my speech from the speech of my students, to minimise the possibility of misuse or misunderstanding.	10	Given the robot has been programmed to distinguish educators from children, When the educator is speaking to the robot, Then the robot should correctly identify that the educator is speaking. When a child is speaking to the robot, Then the robot should correctly identify that a child is speaking.	Could Have
6	US21	Lucía Castro	As an educator, I want to be able to easily install, set up and run the CoWriter software on my computer so that I can prepare for the handwriting activity with children.	TBD	Given the CoWriter software is fully developed and the usage documented, When the developers test the installation, setup, and running process on a computer, Then the process should be straightforward and efficient.	Must Have
6	US22	Dani Muhamad	As a Developer, I want to be able to read documentation for every class, method, and function used in the CoWriter software.	TBD	Given that the CoWriter software is completed, When the developers review the documentation, Then they should be able to quickly understand the purpose and functionality of each class, method, and function in the software, without the need to scrutinise the code.	Should Have

Sprint 2 Backlog

The sprint 2 backlog initially consisted of epics 1 and 2, which encompasses user stories US01-US09. As of April 5th 2023, epic 2 was deemed by the client to be out-of-scope, so the backlog of this sprint has been epic 1, encompassing user stories US01-US06.

Below we break user stories into subtasks and estimate their size:

US01

Task	T-Shirt Size	Story Points
Update *.py files in letter_learning_interaction/scripts/ to use Python 3.8.10	XS	1
Write tests for updated *.py files in letter_learning_interaction/scripts/	XS	1
Update *.py files in letter_learning_interaction/include/ to use Python 3.8.10	S	2
Write tests for updated *.py files in letter_learning_interaction/include/	S	2
Update *.py files in letter_learning_interaction/nodes/ (excluding learning_words_nao.py) to use Python 3.8.10	S	2
Write tests for updated *.py files in letter_learning_interaction/nodes/	S	2
Update letter_learning_interaction/nodes/learning_words_nao.py	L	5
Write tests for updated learning_words_nao.py	L	5

US02

Task	T-Shirt Size	Story Points
Update *.py files in nao_trajectory_following/ to use Python 3.8.10	M	3
Write tests for updated *.py files in nao_trajectory_following/	S	2

US03

Task	T-Shirt Size	Story Points
Update *.py files in shape_learning/ to use Python 3.8.10	M	3

Write test for updated *.py files in shape_learning/	S	2
--	---	---

US04

Task	T-Shirt Size	Story Points
Update *.py files in choose_adaptive_words/nodes/ to use Python 3.8.10	M	3
Extend choose_adaptive_words/ to have functioning UI	M	3
Write tests for updated *.py files in choose_adaptive_words/	M	3

US05

Task	T-Shirt Size	Story Points
Integrate updated packages into single branch, make sure the *.py files are consistent (following code guidelines, using correct method names for imported classes, etc.)	L	5
Fix ROS files in nao_trajectory_following/ (CMakeLists.txt, package.xml, launch file, etc.)	XS	1
Fix ROS files in choose_adaptive_words/	XS	1
Fix ROS files in letter_learning_interaction/	S	2
Resolve issues with ROS workspace dependencies on external libraries	XS	1

US06

Task	T-Shirt Size	Story Points
Research and install working Nao v4 simulation software (to run simulator on legacy code)	S	2
Build old workspace in ROS Indigo	XS	1
Run legacy code on simulated Nao **abandoned due to difficulties and time constraints**	S	2
Research and install working Nao v6 simulation software	M	3
Get our ported code running on simulated Nao	M	3

US07

Task	T-Shirt Size	Story Points
Research how to implement web app for iOS and Wacom tablets, compatible across platforms and working with ROS	L	5
No further tasks as deemed out of scope		

US08

No tasks as deemed out of scope

Some additional tasks came up which are not specific to a user story, either because they relate to multiple or because they relate to other tasks like code review or preparation for showcase/sprint review

Task	T-Shirt Size	Story Points
Update letter_learning_interaction/ unittests to use rostest	S	2
Update nao_trajectory_following/ unittests to use rostest	XS	1
Find suitable replacement for naoqi functionality and update codebase to use replacement instead of naoqi	S	2
Integrate PEP 8 checking on GitHub	XS	1
Perform code review using ChatGPT	XS	1
Try extend functionality of shape_learning/ to send shapes to robot, for use in sprint review/showcase	M	3

Sprint 3 Backlog

The focus in this sprint is on epics 3, 4, and 5, which encompasses user stories US10 to US19.

Update 01 May 2023 following sprint 2 review, and getting to the end of sprint 2, the following tasks are rolling over from the sprint 3 backlog into the sprint 3 backlog (Note: both tasks are in-progress and should be completed fairly quickly):

Task	T-Shirt Size	Story Points
Get our ported code running on simulated Nao	M	3
Extend choose_adaptive_words/ to have functioning UI	M	3

Note that we adopt test driven development for this sprint, so a development task automatically involves testing; thus there is no need to have separate testing tasks, as we did in the previous sprint.

Below we break user stories into subtasks and estimate their size:

US10

Task	T-Shirt Size	Story Points
Integrate ChatGPT API into letter_learning_interaction	S	2
Have NaoSettings class interact with API to generate dialogue	S	2
Create states/methods in StateManager class to handle ChatGPT interaction with child	L	5
Make states/methods for ChatGPT interaction are compatible with not using ChatGPT	M	3
Create some prompt templates which interact between input and ChatGPT	S	2

US11

Task	T-Shirt Size	Story Points
Formatting or choosing prompt templates	S	2

US12 and US13

Task	T-Shirt Size	Story Points
Add a word suggestion related states to StateManager and have this interact with NaoSettings (generate suggestions, receive choice, etc)	XL	8
Interaction between Child Profile (in Choose Adaptive Words) and Nao to generate personalized word suggestions	M	3

US14, 15, 16

Task	T-Shirt Size	Story Points
Add some initial interaction to establish appropriate persona, or derive this from Child Profile	L	5
Some vetting of responses for appropriateness (embedding neighbours, or some other analysis)	M	3

US17-19

Task	T-Shirt Size	Story Points
Integrate microphone input data to NaoSettings (Node which is listening to microphone, publishing data; subscriber to receive this data, may involve custom message, or maybe just String message)	L	5

Use speech to text to convert sound data to text (Whisper or Google speech to text)	M	3
Define interaction between recorded user input and StateManager class	M	3
Define how StateManager class handles recorded user input for conversational state(s)	L	5

US20

Task	T-Shirt Size	Story Points
Ability to distinguish educator speech from child speech	XL	10

Sprint 4 Backlog

We agreed that all 'Must Have' user stories from epics covered in previous sprints have been delivered, and all incomplete stories with lower priority from these sprints is deemed out-of-scope. The focus of this sprint, therefore, is entirely on Epic 6. Epic 6 consists of user stories US21 and US22. Below we break these user stories into sub-tasks and estimate their size.

US21

Task	T-Shirt Size	Story Points
Install the required software on client laptop in IxT lab	S	2
Video showing how to launch the program and interact with the robot when it is running	XS	1
Extensive documentation on how to launch the program with both a physical robot and a simulated robot, provided on both Confluence and GitHub	S	2
Extensive installation instructions, including instructions on how to use automated set-up script	XS	1

US22

Task	T-Shirt Size	Story Points
Finish documentation for code associated with letter_learning_interaction	XS	1
Finish documentation for code associated with nao_writer_naoqi	XS	1
Finish documentation for code associated with choose_adaptive_words	S	2
Finish documentation for code associated with shape_learning	XS	1
Generate RQT graph and state machine graph for the program	XS	1
Fix test cases which were broken in last run of development	M	3
Clear documentation of what features are implemented, and what remains to be done (ideas for future, features we would have liked to implement but didn't have time, etc.)	S	2

Delivered and Undelivered Stories

We managed to deliver all of the must have stories, and some lower priority work. But we did not deliver the entire product backlog in this project. This section will list the stories which were completed, and the stories which were not completed. Stories will be listed by their story ID, further details on the stories can be found by consulting the product and sprint backlogs above.

The stories which were delivered were the following:

- US01
- US02
- US03

- US04
- US05
- US06
- US10
- US17
- US18
- US19
- US20
- US21
- US22

The stories which we did not have time to deliver are as follows:

- US07
- US08
- US09
- US11
- US12
- US13
- US14
- US15
- US16