

Team Name:

MEGAN PAIGE MULHOLLAND

Team Members:

Connor Flok, Megan Mulholland, Jake Hauff, Ayo Abraham

Contribution justification:

Each team member contributed to:

- Jake – Part (1-2) - overview and method selection
- Connor – Part 3-4 – Backlog and Sprint Planning
- Megan – Parts 5-6 Team Collaboration and Risk Management
- Ayo – Part (7 , 8) - tools/technologies and reflection

Project Overview

Brief description:

A simple AI-based spellchecker built using the Python TextBlob library. The program will take user input text, detect misspelled words, and suggest or automatically correct them. It runs locally through a Jupyter Notebook on Google Colab.

Goals and objectives:

- Implement a functional spellchecking tool using TextBlob
- Practice basic AI and NLP integration
- Apply Agile Scrum methods in development
- Demonstrate teamwork, iterative improvement, and testing in a real project

Stakeholders:

- Development team
- Course instructor
- End users who need a simple spellchecker tool

Agile Methodology Selection

Chosen Agile framework (Scrum):

- Scrum allows small, quick iterations for a short project timeline
- Supports frequent feedback and adjustments
- Encourages collaboration and clear sprint goals

Key roles and responsibilities:

- Product Owner: manages backlog and defines goals - Megan
- Scrum Master: ensures team follows Scrum process - Jake
- Developers: design, code, test, and document the spellchecker
- All team members will work as developers in some capacity

Product Backlog

User stories:

1. I want to enter text so that I can check for spelling errors.
 - a. *Acceptance:* The program accepts any string input.
2. I want the program to show corrected text.
 - a. *Acceptance:* Corrected output displays clearly below the input.
3. I want to see suggested corrections for each misspelled word.
 - a. *Acceptance:* TextBlob suggestions appear when available.
4. I want the system to handle sentences and paragraphs.
 - a. *Acceptance:* No crashes or missed words on larger input.

Prioritization strategy:

1. Core input/output features
2. Correction logic
3. Fixing final interface and optional improvements/visual improvements

Sprint Planning

Sprint schedule:

- Sprint 1 (1 week): Setup environment, import TextBlob, basic spellcheck
- Sprint 2 (0.5 week): Add suggestion feature and handle longer text
- Sprint 3 (0.5 week): Testing, debugging, documentation, demo

Tasks breakdown per sprint:

Sprint 1

- Install and configure TextBlob
- Write initial spellcheck code

- Verify simple word correction works

Sprint 2

- Add multiple-word handling
- Implement suggestions and user feedback
- Improve error handling

Sprint 3

- Write final documentation
- Conduct testing and code review
- Prepare presentation and submit

Estimation techniques used:

- Story points estimated by task complexity (1 = simple, 3 = moderate, 5 = hard)
- Used team discussion to assign points

Team Collaboration Plan

Communication tools and strategies:

- Groupchat/In person for team messages
- Google Docs for notes
- Colab comments for code collaboration
- Spreadsheet for tasks

Meeting schedules:

- Daily stand-up (short check-in)/ messaging
- Sprint review after each sprint in person
- Final meeting at completion to go over project and future work/possible additions

Risk Management

Potential risks:

- Library errors or version issues with TextBlob
- Team scheduling conflicts/Overlapping assignments
- Learning curve
- Loss of Colab session data

Mitigation strategies:

- Use consistent library versions
- Store work in GitHub
- Maintain daily progress updates and keep in contact consistently

Tools and Technologies**Project management tools:**

- Spreadsheet for task tracking
- Outlook for sprint scheduling

Development and testing tools:

- Google Colab (Python, Jupyter environment)
- TextBlob library
- GitHub for version control

Reflection**Challenges anticipated:**

- Managing collaboration across multiple Colab notebooks
- Handling edge cases in text correction
- Keeping sprints short but productive and managing errors quickly

How Agile principles will help address them:

- Frequent sprint reviews ensure quick issue detection
- Daily communication keeps the team aligned
- Incremental improvements reduce large last-minute changes so we can stay on top of it all