

# AI4Agile Beta Prototype Updates

CptS 423 - Dr. Bolong Zeng

Authors: Phong Bach, Emily Cawlfeld, Nain Galvan, and Aric Monary

Mentor: Skip Baccus

Date Submitted: 2/5/21

# 1. Project Updates

## 1.1 Functionality

At this time, no changes have been made to or planned for in the overall project functionality since the Alpha Prototype presentation. However, if user acceptance testing reveals any needs for adjustments in the upcoming development cycles, such updates may be added later.

## 1.2 System

As a whole, the AI4Agile application is being prepared for release to the Atlassian Marketplace. This necessitates certain updates to individual components to decrease waiting time for users during certain interactions. Additionally, any updates necessary for the application to conform to Atlassian's guidelines for deployment to the marketplace will be discovered in the Marketplace research phase of the project and fixed before deployment continues. From the user side, the team does not currently have plans to change visual elements in any major way.

Due to considerations for the full semi-finalist submission in March, the app may require a transition to be run on Jira Server and made available via hosting due to an uncertain time delay with the approval of app's for the Atlassian Marketplace, where apps are hosted for Jira Cloud. Jira Server is hosted locally and made available through publicly available websites, usually blocked by login credentials. For example, many open-source projects have their own Jira Server due to the large degree of flexibility offered by local hosting.

## 1.3 Components

The task generation component of this project has changes planned in its backend implementation, those being switching from a Python package, Stanford's Stanza NLP [1], to Stanford's CoreNLP package in Java [2]. In addition to this change, point person Nain Galvan will be working to generally improve the sentence outputs for task generation.

For epic decomposition, the planned updates center around providing caching for results so that a user's choice of how many stories they want to be generated from their epic will be easier to quickly adjust and see the differences between.

Story optimization will be adjusted on an algorithmic level in order to make results be available more quickly.

Regarding the relationship visualization tool, plans are now focused even more on user appeal as the app is prepared for public release. This means features such as on-screen panning and zooming tools will be added, as well as increased options for what can be displayed on the graphs. All component changes listed here are planned for sprint cycles later in February, as can be seen in Table 3.1.

## 2. Testing and Validation Plans

### 2.1 Test Data Collection/Generation Plans

Feature		Testing Plan
Epic Decomposition		Inputs: <ul style="list-style-type: none"><li>• Ideal epic</li><li>• Disjoint epic (many unrelated requirements)</li><li>• Semi-ideal epic</li></ul>
Story Optimization		Inputs: <ul style="list-style-type: none"><li>• Already optimized story</li><li>• Minimally related story (story optimization as an entry point)</li><li>• Ideal unoptimized story</li><li>• Semi-ideal unoptimized story</li></ul>
Task Generation		Inputs: <ul style="list-style-type: none"><li>• Unoptimized story (task generation as an entry point)</li><li>• Optimized story</li></ul>
UI	Main GUI	User Trials with mocked processes for intuitiveness of button locations and ease of use testing
	Suggestions	Using mocked data for Epic, Story, and Task types to manually test results from the Create Selected button after editing, selecting, and deselecting various suggestions.
	Graph	Inputs: <ul style="list-style-type: none"><li>• Large and small issue sets for testing zoom and scaling</li><li>• Assignee-only/dependency-only/none/both relationships turned on</li><li>• Ensuring proper querying per issue chosen</li></ul>

Table 2.1: Unit testing plan per component

Integration	Testing Plan
Epic Decomposition	Flows: <ul style="list-style-type: none"><li>• Epic Decomposition directly to Story Optimization</li><li>• Epic Decomposition directly to Task Generation</li></ul>
Story Optimization	Flow: <ul style="list-style-type: none"><li>• Directly to Story Optimization</li></ul>

	<ul style="list-style-type: none"> <li>• Story Optimization directly to Task Generation</li> </ul>
Task Generation	<ul style="list-style-type: none"> <li>• Directly to Task Generation</li> <li>• Epic to Task Generation</li> <li>• Optimized story to Task Generation</li> </ul>
UI/UX	<ul style="list-style-type: none"> <li>• Integration testing per sub-component plugged in</li> <li>• User observation</li> </ul>

*Table 2.2: Integration testing plan*

## 2.2 Scenarios for Testing and Validation

Testing was done for the three AI components to make sure their outputs were correct. For epic decomposition, we used an example epic and manually broke it down into stories. The results of epic decomposition AI were compared to the hand-decomposed epic to validate the component's results. For story optimization, a similar approach was taken. Continuing from the manually broken down epic to stories, the stories were then broken to tasks. The outputs from the story optimization AI were compared to the expected stories for validation purposes.

The team found a sample set of compound sentences with their corresponding simple sentence sets that could be used to test the task generation component's ability to create simple sentences from compound ones. Using the expected outputs given in the sample set, the AI's results were able to be validated. The plans in Tables 2.1 and 2.2 were followed as given, before and during integration of the components into the Jira platform. These tables will continue to guide this team's testing process as the project continues, with more varied sets of inputs to be added for validation.

## 2.3 Test Result Analysis and Presentation Plans

After running the tests for the Jira features and AI components and comparing them to the predetermined functional and non-functional requirements for this project, the results showed they performed as intended, but improvements could still be made for better results. Future testing efforts will continue using the functional and non-functional requirements to analyze testing results, with heavier emphasis on the non-functional now that the main project functionality has been implemented.

For presentation purposes, the team will create an epic sample to be used and broken down using epic decomposition, story optimization, and task generation. During the decompositions, suggestions generated from the AI components will be shown, as well as specific sub-features, such as editability of suggestions. Team members will be assigned to specific tasks on an example Jira board, in order to present the relationship visualization graph capabilities and features.

### 3. Summary of Remaining Work

The outline of remaining works and tasks for AI4Agile, as seen in Table 3.1, are in the event that Team Katara does not move forwards as a Semi-finalist in the SCORE 2021 competition, which will be announced on February 8th. The primary focus, regardless of competition outcome, will be to deploy the app on the Atlassian marketplace.

Epic	Task	Point Person	Start Date	End Date
Atlassian Marketplace	Review Marketplace Guidelines	Emily Cawfield	2/1	2/14
	Review Existing App's Compliance with Guidelines	Phong Bach		
	Evaluate Security Compliance	Aric Monary		
	Integrating Authentication	Nain Galvan		
Hosting Service	Research Hosting Options	Nain Galvan	2/8	2/15
	Evaluate Software Hosting Needs		2/16	2/21
	Retrofit Software to be Hosted	Aric Monary		
Deployment Pipeline	Evaluate DevOps services based on Hosting Service	Phong Bach	2/16	2/21
	Deploy the Pipeline			
	Integrate Pipeline into development			
	Link Pipeline to Jira			
User Learning Experience	Prioritize use cases from customer	Emily Cawfield	2/8	2/21
	Develop Written User Guide			
	Develop Integrated Instructions into app			
AI Features	Caching AI results	Aric Monary	2/22	3/14
	Filtering "bad" generated Tasks	Nain Galvan		
	Improve Noise Handling	Phong Bach		
Graph Features	User settings for Rendered Fields	Emily Cawfield	2/22	3/14
	Additional Relations			
	Additional Fields			

Table 3.1: Remaining Task Table for Spring Semester

<b>Epic</b>	<b>Task</b>	<b>Point Person</b>	<b>Start Date</b>	<b>End Date</b>
UI Features	Additional Warnings for Edge Cases	Aric Monary	2/22	3/14
	Revise Layout by User Feedback			
Alpha Deployment	Evaluate Initial User Feedback	Nain Galvan	3/15	4/4
	Fix Critical Bugs from Alpha	Depends on Bug		
	Create New Features per User Feedback	Depends on Feature		
Beta Deployment	Deploy new Alpha Features	Depends on Feature	4/5	4/26
	Fix Critical Bugs from Beta	Depends on Bug		
	Evaluate Beta User Feedback	Phong Bach		
Documentation	Beta Prototype Updates	N/A	2/1	2/5
	Midterm Progress Report		3/7	3/12
	Final Report		4/26	5/7

Table 3.1 (continued): Remaining Task Table for Spring Semester

In the event that Team Katara does move forward as a semi-finalist in the SCORE 2021 competition, the Alpha and Beta Deployment sections of the plan will be altered to focus efforts on the polish of existing features along with the creation of a presentation that demonstrates AI4Agile.

Due to the remote nature of the competition because of Covid-19, the project will still move towards a deployment on the Atlassian Marketplace to allow for remote evaluation of the app. Additional adjustments to the timeline of the tasks will be made as the semi-finalist deliverable must be submitted by March 1st. The submission must include a fully deployed application, and in the case of our web-based application, must be available via a "...URL pointing to a publicly accessible server where the project has been fully deployed" [3]. Such requirement will be fulfilled through the app being published on the Atlassian Marketplace

In the event that Team Katara is invited as a finalist, the project timeline will need further revision as it goes outside the scope of the semester, with final presentations being in the last week of May. The final submission does not require any further deliverable outside of previous deliverables. A well-rehearsed and thorough presentation should be sufficient and will be further discussed and refined if Team Katara is selected as a finalist.

Sources:

1. <https://stanfordnlp.github.io/stanza/>
2. <https://stanfordnlp.github.io/CoreNLP/>
3. <https://conf.researchr.org/home/icse-2021>