Phase I System Presentation

Team Katara

Scenario 1

AS-IS

- James is a busy product owner and doesn't have the time to go through a new set of requirements and specifications for their product.
- He arbitrarily breaks the requirements and specifications into user stories.
- Two developers end up working on closely related functions and waste time doing parallel research.

TO-BE

- James enters the new set of requirements and specifications for their product into an Epic within Jira.
- He adds the AI4Agile app to the project and presses the "Decompose Epic" icon within the selected epic where he entered the documents.
- He is shown a list of suggested user Stories based on related wording.
- He doesn't want two of the stories and deselects them. He also edits another story to provide clarification.
- James presses the "Create Selected" button and the epic is populated with the generated stories

Scenario 2

AS-IS

- James is a busy product owner and doesn't have the time to go through a new set of requirements and specifications for their product.
- He arbitrarily breaks the requirements and specifications into user stories.
- Some developers end up working on a big user story and the development process takes a longer time.

TO-BE

- After James enters the new set of requirements and specifications for their product into an Epic within Jira, it was broken into multiple user stories.
- He presses the "Optimize User Stories" icon within the selected epic where he entered the documents.
- He is shown some new user Stories correspond to a current user Stories.
- He doesn't want some of the stories and deselects them.
- James presses the "Create Selected" button and the new user Stories are added to the project.

Scenario 3

AS-IS

- Now, the actor wants to break things into tasks, but again it’s slow going for him by hand.
- He needs to determine the tasks to determine the deliverables of the product
- The tasks should allow one developer to complete in no longer than a day

TO-BE

- Once the user has broken down their epic to the optimized user-stories
- He can press “Create tasks” button
- He then sees tasks for the specific user-story
- He doesn’t like some of the tasks that were generated so he deselects them and writes in his own
- He presses “Create Selected” and the tasks are added to the project

Scenario 4

AS-IS

- Alex is trying to assign pieces of a new epic to his team. He wants to do this in a way that minimizes occurrences of developers being blocked by someone else's task or story.
- There are a plethora of tasks, and Alex is having a hard time keeping straight which ones are related to which.
- He ends up spending hours opening and closing task detail panes, mentally mapping out the relationships, and shuffling potential assignments around.

TO-BE

- Alex clicks on the epic in question, and scrolls down to the issue relationship graph.
- He's only interested in the dependency relationships, so he clicks the "Filter" button to change it so the graph only shows those.
- He enlarges the graph so he can clearly see this epic's entire tree, and sets to work assigning the tasks or stories to his team.

Creeping Rate

Data Functions and Transactional Functions:

- Internal Logical File (ILF)
    - 1 Point - Suggested Stories
    - 1 Point - Suggested Optimized Stories
    - 1 Point - Suggested Tasks
- External Interface File (EIF)
    - 1 Point - User-entered Epics
    - 1 Point - User-entered Stories
    - 1 Point - User-entered Tasks
    - 2 Points - User-defined dependencies
- External Input (EI)
    - 1 Point - Query user-entered epics
    - 2 Points - Query user-entered stories and accepted suggestion stories
    - 2 Points - Query user-entered dependencies
- External Output (EO)
    - 1 Point - Input Suggested Stories
    - 2 Points - Input Suggested Optimized Stories
    - 1 Point - Input Suggested Tasks
- External Inquiry (EQ)
    - 2 Points - Get user-edited stories
    - 2 Points - Get user-edited tasks

Development Function Point Count

- Size of functions delivered to the user by the development project (ADD) = FP Count (ILFs) + FP Count (EIFs) + FP Count (EIs) + FP Count (EOs) + FP Count (EQs) = 0
- Size of the conversion functionality (CFP) = FP Count (ILFs) + FP Count (EIFs) + FP Count (EIs) + FP Count (EOs) + FP Count (EQs) = 22
- Development Function Point Count (DFP) = ADD + CFP = 22 + 0 = 22

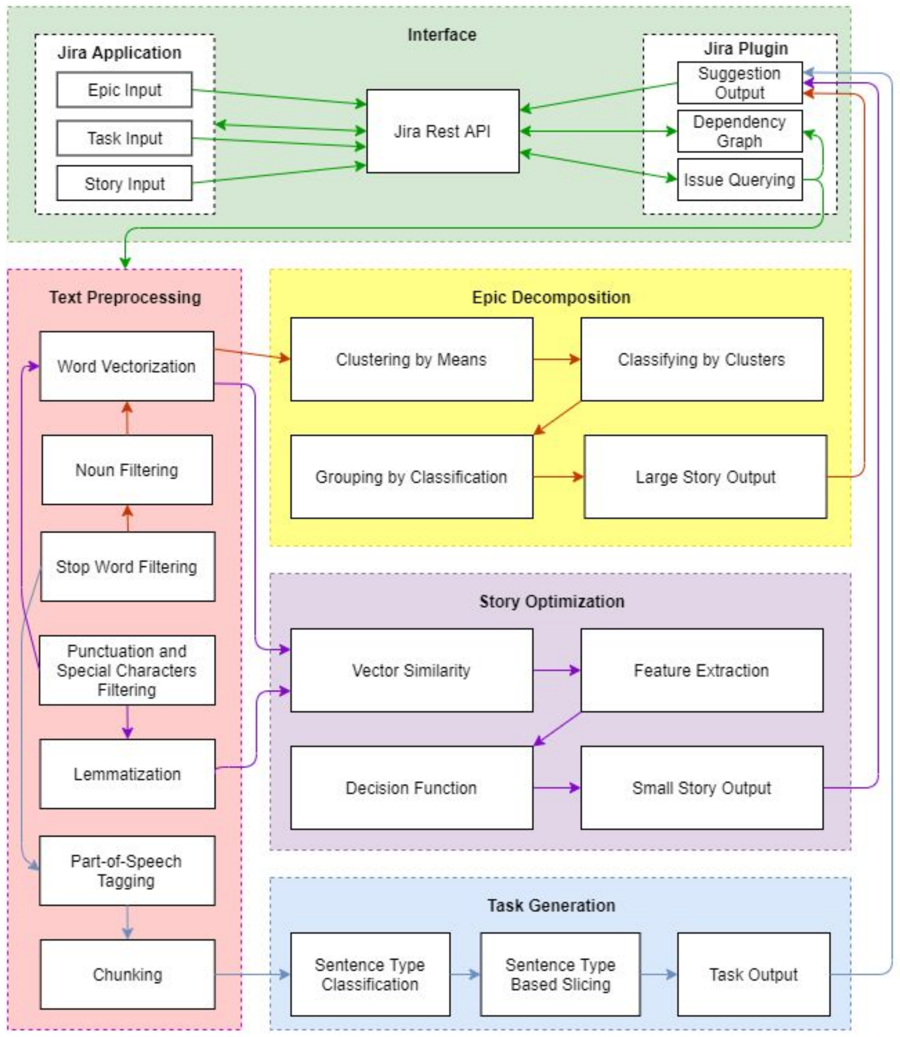
Creep Rate:

- Estimated Beginning Number of Function Points: 22
- Estimated End Number of Function Points: 30
- Number of Months for Project: 2
- Creep Rate: ((30-22)/22)/2 = **~18%**

Why use AI4Agile?

- Complimentary tool for Agile development
  - Editable suggestions
  - Previewable results
- Integrated into a robust and accessible platform
- Little required Agile development experience
- Low required Domain knowledge
- Enhanced Risk Analysis
  - Dependency Visualization

Block Diagram Overview


Interface
Jira Application
Epic Input
Task Input
Story Input
Jira Rest API
Jira Plugin
Suggestion Output
Dependency Graph
Issue Querying
Text Preprocessing
Word Vectorization
Noun Filtering
Stop Word Filtering
Punctuation and Special Characters Filtering
Lemmatization
Part-of-Speech Tagging
Chunking
Epic Decomposition
Clustering by Means
Classifying by Clusters
Grouping by Classification
Large Story Output
Story Optimization
Vector Similarity
Feature Extraction
Decision Function
Small Story Output
Task Generation
Sentence Type Classification
Sentence Type Based Slicing
Task Output