CptS 484: Software Requirements

WRS Evolution

Requirements Elicitation

10/16/2020

Table of Contents

Revision History

Date	Versio n	Changes	Editor
9/22/2020	1.0	Initial Draft	Phong Bach, Emily Cawlfield, Nain Galvan, and Aric Monary
9/25/2020	1.1	Updated Requirements	Nain Galvan
9/25/2020	1.2	Added Storyboard	Emily Cawlfield
9/26/2020	1.3	Resolved Feedback	Emily Cawlfield and Aric Monary
9/27/2020	1.4	Refined Scope	Phong Bach
9/28/2020	1.5	Prototype Insertion	Aric Monary
9/30/2020	1.6	Refined Requirements	Nain Galvan
10/16/2020	1.7	Final Phase I Proofing	Emily Cawlfield, Aric Monary

[1] Introduction

1.1. Purpose

The objective of the project is to develop a software that leverages the power of AI to assist software developers in their agile development process. Although software team members are following the Agile processes and methods, it is usually the case for the vision of creating agile products to be missed. The Agile methodology is often diluted by software developers and businesses developers which renders the process ineffective. The problem increases in larger teams and systems, which inherently makes it difficult to manage.

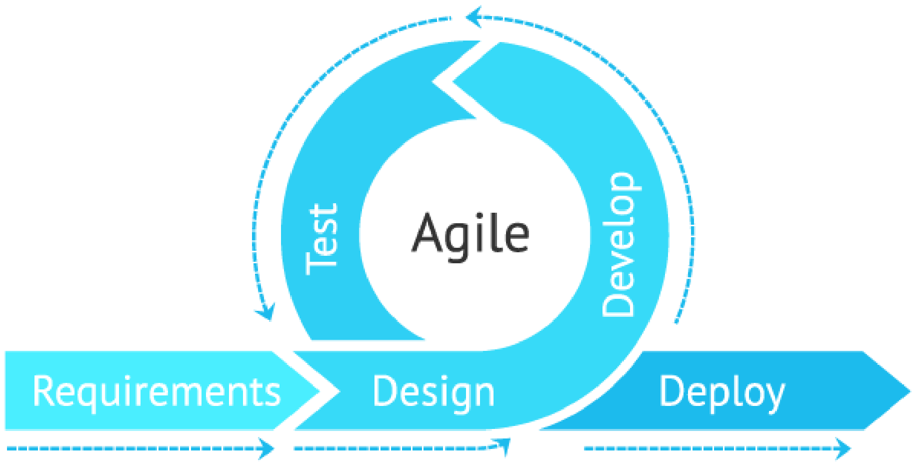


Figure 1: The typical Agile software development process. This loop, representing what Agile terms as a “sprint,” is to be repeated as many times as necessary for a project, in short, time-boxed periods. [1]

At the start of the project, a list of criteria that the ideal prototype should meet was made. These goals are based on the research paper “AI4Agile: Developing AI-enabled tool support for user story refinement in JIRA” by Professor Hoa Khanh Dam [2]. The application is important to help assist the software team by streamlining the Agile development process. The application implementation is similar to what was discussed in the research paper of an AI-power Agile project assistant. It consists of five distinct parts: the planning engine, representation learning engine, optimization engine, analytics engine, and the conversational dialog engine. All these engines work together to let software teams be able to rapidly deliver results that lead to business value.

1.2. Scope

In the five core engines, the representation learning engine’s main focus is Natural Language Processing for project artifacts such as backlog item, product visions, sprint goals, description of backlog item, and team communication. The optimization engine helps with the planning such as computing the optimal selection of backlog items. The planning engine can plan for a sprint with sprint goals using the state of the project. The analytics engine can provide decision support with descriptive analytics such as charts and sprint reports, with predictive analytics such as estimations tools, with prescriptive analytics such as backlog item identification, backlog item refinement and risk mitigation. User interactions with the application are managed through the conversational dialog engine, wherein they can decide if they want to utilize its suggestions, along with giving developer-specific feedback metrics.

For the scope of this project, the application is integrated into the JIRA platform to assist in Agile development. Overall, the application utilizes the five engines that were discussed earlier which works in combination to help teams break epics into progressively smaller user stories and tasks, down to a level where they are ready to be put into sprints. The four core functionalities of the applications are:

1. Recommend user stories that are decomposed from an epic
2. Recommend smaller user stories are split from bigger user stories
3. Recommend subtasks derived from a user story
4. A visualization (e.g. a graph) of epics, user stories and tasks, and their relationships

1.3. Objectives and Success Criteria

The project objectives and success criterias are that the software must:

- be integrated into Jira
- return relevant user stories from epic
- return relevant smaller user stories from bigger user stories
- return relevant tasks from user story
- be able to show visualization of epics, user stories and tasks, and their relationships
- be minimally obstructive
- work with good consistency
- have a fast response time
- be simple and intuitive

1.4. Definitions, Acronyms, and Abbreviations

Term	Definition
AI	Artificial Intelligence
Agile	A software process in which a project is broken up into bite-sized portions and tasks are grouped together to be completed in relatively short time periods.
Backlog	A list of tasks to be completed over an Agile project’s span.
Deep learning	An AI technique that mimics the workings of the human brain in the processing of information.
Epic	A topic that encompasses a series of user stories that fulfill requirements.
Jira	A product made by Atlassian for issue- and bug-tracking in Agile development.
Machine learning	Machine learning is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence.
NLP	Natural Language Processing; a field of AI in which human interactions with computers are dealt with via natural language (any human language that was not artificially created).
Sprint	A short period in which a team completes a set amount of tasks.
Story	An end goal from the perspective of an end-user, written in non-technical language.
Task	The smallest unit of work that a team will undertake, from a developer’s point of view.

1.5. Overview

In section 2, the preliminary domain, preliminary functional requirements as well as the preliminary non-functional requirements were discussed. In section 3, the issues with the preliminary definition given were laid out. It consists of the domain issues, functional requirements issues, and non-functional requirements issues. Options and the rationales for each choice were made.

In section 4, there are two main topics that were discussed, W and RS. The W section states the problem descriptions and the corresponding goals for each of them. For each goal, the backward traceability and forward traceability were listed. From the above information, the improved understanding of domain, stakeholders, functional, and non-functional objectives can be made to see which problem each one alleviates and which goals each one achieves. For the RS section, the function requirements, non-functional requirements, and specifications are listed with the description as well as the satisfied functional requirements issue, objectives and prototype feature.

Section 5 and 6 cover the preliminary prototype and the prototype interface mock-ups. The user manual and the references can be found in section 7 and 8 respectively.

[2] Preliminary Definition

2.1. Preliminary Domain

PD_ID	Preliminary Domain Description
PD1	The app will be deployed on Jira, an agile management and software development tool.

2.2. Preliminary Functional Requirements

PFR_ ID	Preliminary FR Description
PFR1	Recommend user stories that are decomposed from an epic.
PFR2	Recommend smaller user stories are split from bigger user stories.
PFR3	Recommend subtasks derived from a user story.
PFR4	Generate a visualization of epics, user stories, and tasks, and their relationships.

2.3. Preliminary Non-Functional Requirements

PNFR_ ID	Preliminary NFR Description
PNFR1	Decompose epics and user stories without loss of information.
PNFR2	Depict only necessary relationships between epics, user stories, and tasks.
PNFR3	Keep all, potentially confidential, project information secure.
PNFR4	Usable for users without domain expertise.
PNFR5	Usable by users both familiar and unfamiliar with agile development.
PNFR6	Extensible to different platforms
PNFR7	Fast response time for AI processing

[3] Issues with the Preliminary Definition Given

3.1. Domain Issues

Domain Issue ID	Domain Issue Description	
DI1	**PD_ID**	PD1: The app will be deployed on Jira, an agile management and software development tool.
	Ambiguous. Jira is implemented partially into many other Atlassian products while also having a dedicated platform.	
	Option 1	Support the dedicated Jira platform.
	Option 2	Support all platforms Jira is implemented into, along with the dedicated platform.
	Choice	Option 1
	Rationale	Due to the deadline associated with the project submission and the requirements associated with the project, the dedicated Jira platform will be the only platform support while the project stays within the competition scope.
Satisfied by	NFR6	

3.2. Functional Requirements Issues

FR Issue ID	**Description**	
FRI1	**PFR_ID**	PFR1. Recommend user stories that are decomposed from an epic.
	1. The problem that arises is the wrong clustering of user-stories once the epic is broken down. 2. The loss of information from this decomposition.	
	Option 1	Accept our grouping of stories
	Option 2	Give the suggestion to accept our group if user-stories or write their own
	Choice	Option 2
	Rationale	If the user likes the way the user-stories are grouped they have the option to accept them but if they don’t they have the option to write their own.
Satisfied by	FR1	

FR Issue ID	Description	
FRI2	**PFR_ID**	PFR2. Recommend smaller user stories are split from bigger user stories.
	1. The problem that arises is the loss of important information. 2. By optimizing the user-stories could make the new ones too small to be useful	
	Option 1	Accept our break down of user-stories
	Option 2	Give them an optimization button and suggestion to accept our user-stories or write their own
	Choice	Option 2
	Rationale	If the user thinks the user-stories are already small enough they can just accept them. If they think the user-stories need to be smaller than they can click the optimize button to make them smaller with the choice to accept those small user-stories or write their own.
Satisfied by	FR2	

FR Issue ID	Description	
FRI3	**PFR_ID**	PFR3. Recommend subtasks derived from a user story.
	The loss of information from this decomposition.	
	Option 1	Accept the tasks
	Option 2	Give the suggestion to accept the tasks or write their own
	Choice	Option 2
	Rationale	If the user likes the tasks he can keep them but if he doesn't he can write them themselves.
Satisfied by	FR3	

FR Issue ID	Description	
FRI4	**PFR_ID**	PFR4. Generate a visualization of epics, user stories, and tasks, and their relationships.
	1. Showing the implicit or explicit relationships between epics, user-stories, and tasks. 2. Determining the depth needed to show between the relationships	
	Option 1	Show arbitrarily chosen relationship types
	Option 2	Users can choose which relationships to show.
	Choice	Option 2
	Rationale	If the user thinks that the implicit or explicit relationships are wrong they can revise them. The user can also choose
Satisfied by	FR4	

3.3. Non-Functional Requirements (NFR) Issues

NFR Issues ID	Description	
NFRI1	**PNFR ID**	PNFR1: Decompose epics and user stories without loss of information.
	Ambiguous: What is considered a loss of information?	
	Option 1	Any loss of wording.
	Option 2	Some text loss. Keeping all content but losing words with the purpose of readability (transition words).
	Choice	Option 2
	Rationale	Some information, text from an inputted epic, is unnecessary (transition words for example) and serves no purpose besides readability in the initial epic. Additionally, expand the loss of information to cover the story optimization and task generation.
Satisfied by	NFR1	

NFR Issues ID	Description	
NFRI2	**PNFR ID**	PNFR2: Depict only necessary relationships between epics, user stories, and tasks.
	1. What relationships are necessary to show between the epic, stories, and tasks? 2. How many relationships are necessary to show? What should the depth of these relationships be?	
	Option 1	Allow the user to select the type of relationships and how many relationships to show.
	Option 2	Show all relationships (implicit and explicit) with a set depth to the relationships.
	Choice	Option 1
	Rationale	The user should be able to specify the relationships they want to visualize since the decisions may require considerations for a varying amount of relationships. The depth should also have the same degree of flexibility for the previously mentioned need.
Satisfied by	NFR2	

NFR Issues ID	**Description**	
NFRI3	**PNFR ID**	PNFR3: Keep all, potentially confidential, project information secure.
	What does it mean to keep information secure?	
	Option 1	Project information is kept internal only to Jira.
	Option 2	Project information is Internal to Jira and an outside database to query from. Database data deleted after use.
	Option 3	Maintain a database of project information along with using Jira to populate it.
	Choice	Option 2
	Rationale	In order to not have to maintain accurate information about a project, along with removing potential artifacts that could lead to security concerns, the app will not keep long term project data.
Satisfied by	NFR3	

NFR Issues ID	**Description**	
NFRI4	**PNFR ID**	PNFR4: Usable for users without domain expertise.
	1. What sort of information would a domain expert have that could be used? 2. What domain specific information would be needed?	
	Option1	For refinement, ask a user for clarifying information. Requires domain expertise.
	Option2	Allow the user to decompose and refine without clarification. Performance will vary.
	Choice	Option 2
	Rationale	In order for the application to appeal to larger sources.
Satisfied by	NFR4	

NFR Issues ID	Description	
NFRI5	**PNFR ID**	PNFR5: Usable by users both familiar and unfamiliar with agile development.
	What amount of information about Agile is needed to use our app?	
	Option 1	Full Agile understanding (whole process).
	Option 2	Partial Agile understanding (definitions).
	Option 3	No Agile development understanding.
	Choice	Option 2
	Rationale	Limited Agile development knowledge is needed as a user must understand what goes into an epic, story, and task.
Satisfied by	NFR5	

NFR Issues ID	Description	
NFRI7	**PNFR ID**	PNFR7: Fast response time for AI processing
	How fast should the text processing components respond within Jira?	
	Option 1	Instantaneous response time.
	Option 2	5 second response time.
	Choice	30 second response time.
	Rationale	Due to AI processing and external database usage, a time response of 5 seconds is realistic for text-based AI processing and database querying during each step of the decomposition and refinement.
Satisfied by	NFR7	

[4] WRS

4.1. W

4.1.1. Problem

Problem ID	Problem Description	Corresponding Goals
P1	What is an epic? What is the format and what information is contained within?	G1
P2	How to handle vague information within epics?	G1
P3	How to avoid loss of information during processing?	G1, G2, G3
P4	What is the optimal size of a story? How should the size of the story be calculated?	G2
P5	What format should stories be in?	G1, G2
P6	What format should tasks be in?	G3
P7	What information should be in a task to yield a deliverable?	G3
P8	What relationships need to be known between stories?	G4
P9	How many relationships should be shown?	G4
P10	How should suggestions be presented?	G5
P11	The app may be run on varying computer systems.	G8
P12	Users may have varying levels of domain knowledge to the epic that is being inputted.	G6
P13	Users may not be familiar with Agile development.	G7
P14	Leaking internal project information through an external source.	G9

4.1.2. Goals

Goal ID	Goal Description	Backward Traceability	Forward Traceability
G1	Decompose epic into user stories	P1, P2, P3, P5	IFRO1, INFRO1
G2	Break down user stories into smaller stories	P3, P4, P5	IFRO2
G3	Create tasks from user stories	P3, P6, P7	IFRO3
G4	Visualize the implicit and explicit relationships between an epic, stories, and tasks	P8, P9	IFRO4, INFRO2
G5	Offer generated stories and tasks as suggestions	P10	
G6	No domain knowledge needed from the user	P12	INFRO4
G7	User possesses limited Agile development understanding	P13	INFRO5
G8	App is extensible to varying platforms	P11	INFRO6
G9	Maintain the security of project information.	P14	INFRO7

4.1.3. Improved understanding of Domain, Stakeholders, Functional, and Non-Functional Objectives

4.1.3.1 Improved Domain

Improved Domain ID	Improved Domain Description
ID1	The dedicated Jira platform, a complete agile management and software development tool.

4.1.3.2. Stakeholders

- Sponsors
  - Research: Hoa Khanh Dam, University of Wollongong, Australia
  - Faculty: Bolong Zeng
- Project Mentor
  - Name: Skip Baccus
- Volunteer Faculty Advisor
  - Name: Jeremy Thompson
- Potential Users of AI4Agile
  - Software Developers using Agile software development and Jira Software by Atlassian are the primary users of AI4Agile. Specific user profiles, within Agile development, for the app include:
  - Scrum Masters - Utilize story optimization, task generation, and visualization to speed up their administrative duties along with the ability to make informed risk decisions.
  - Software Requirements Analysts - Enable faster grouping of requirements to be broken into user stories.
  - Software Project Managers - Take potential risk metrics produced by the app to understand the status of their project.

4.1.3.3. Improved Functional Objectives

Based on the above information and our goals, the functional objectives of AI4Agile are:

Improved FR Objective ID	Objective Description	Alleviates Problems	Achieves Goals
IFRO1	The user-stories will be grouped together correctly and will not lose any information due to the break down from epic	P3,P4,P5	G1
IFRO2	The optimized user-stories will be good size and will not lose information	P4,P5	G2
IFRO3	The tasks will not lose information	P6,P7	G3
IFRO4	Will show the implicit and explicit relationships between epics,user-stories, and tasks. Will allow the user to select which relationships to show and edit them	P9	G4

4.1.3.4. Improved Non-Functional Objectives

Improved NFR Objective ID	Objective Description	Alleviates Problem	Achieves Goal
INFRO1	Process epics down into stories and tasks without losing any content-critical information through each transformation.	P1	G1
INFRO2	Show a customizable type and depth to relationships between epics, user stories, and tasks.	P8, P9	G4
INFRO3	Keep project information secure within Jira, primarily, while disposing of all project data stored externally after utilization.	P14	G9
INFRO4	At a minimum, domain knowledge associated to the inputted epic is not needed to perform any tasks.	P12	G6
INFRO5	Decomposition, optimization, and generation require limited, definition only, understanding of Agile development.	P13	G7
INFRO6	The app will be usable on varying operating systems and browsers which support Jira.	P11	G8
INFRO7	The response time to each, Decomposition, optimization, and generation, shall not exceed five seconds.	P14	G9

4.2. RS

4.2.1. Functional Requirements

FR ID	**Description**
FR1	Once user inserts epic it will be broken down into groups of user-stories
Satisfies Functional Requirement Issue	FRI1
Satisfies Objectives	IFRO1
Satisfied by prototype feature	Epic Decomposition

FR ID	**Description**
FR2	Large user-stories will be broken down into smaller user-stories
Satisfies Functional Requirement Issue	FRI2
Satisfies Objectives	IFRO2
Satisfied by prototype feature	Story Optimization

FR ID	**Description**
FR3	Break down user-stories into tasks that will be used as deliverables
Satisfies Functional Requirement Issue	FRI3
Satisfies Objectives	IFRO3
Satisfied by prototype feature	Task Generation

FR ID	Description
FR4	Give the user a visual representation of the implicit and explicit relationships between epics, stories, and tasks.
Satisfies Functional Requirement Issue	FRI4
Satisfies Objectives	IFRO4
Satisfied by prototype feature	Relationship Graph

4.2.2. Non-functional Requirements

NFR ID	Nonfunctional Requirement 1
NFR1	Process text without loss of content-critical information at each stage in the agile development process.
Operationalized Functional Requirements	FR1, FR2, FR3
Satisfies Nonfunctional Requirement Issue	PNFRI1
Satisfies Non-functional Objective	INFRO1
Constrains	P3
Satisfied by prototype feature	Epic Decomposition, Story Optimization, Task Generation

NFR ID	Nonfunctional Requirement 2
NFR2	Allow the user to select the type and depth of relationships shown between epics, user stories, and tasks.
Operationalized Functional Requirements	FR4
Satisfies Nonfunctional Requirement Issue	PNFRI1
Satisfies Non-functional Objective	INFRO2
Constrains	P8, P9
Satisfied by prototype feature	Relationship Graph

NFR ID	Nonfunctional Requirement 3
NFR3	All long term storage of project information will remain internal to Jira.
Operationalized Functional Requirements	FR1, FR2, FR3, FR4
Satisfies Nonfunctional Requirement Issue	PNFRI3
Satisfies Non-functional Objective	INFRO3
Constrains	P8, P9
Satisfied by prototype feature	Epic Decomposition, Story Optimization, Task Generation, Relationship Graph

NFR ID	**Nonfunctional Requirement 4**
NFR4	Allow for full functionality without the input of domain expertises. Domain expertise may be used but not required.
Operationalized Functional Requirements	FR1, FR2, FR3
Satisfies Nonfunctional Requirement Issue	PNFRI4
Satisfies Non-functional Objective	INFRO4
Constrains	P2, P4, P12
Satisfied by prototype feature	Epic Decomposition, Story Optimization, Task Generation

NFR ID	**Nonfunctional Requirement 5**
NFR5	Provide all needed understanding and definitions to the agile development process. Require limited prior knowledge of agile.
Operationalized Functional Requirements	FR1, FR2, FR3, FR4
Satisfies Nonfunctional Requirement Issue	PNFRI5
Satisfies Non-functional Objective	INFRO5
Constrains	P1, P4, P13
Satisfied by prototype feature	Epic Decomposition, Story Optimization, Task Generation, Relationship Graph

NFR ID	Nonfunctional Requirement 6
NFR6	The app will support the dedicated Jira platform.
Operationalized Functional Requirements	FR1, FR2, FR3, FR4
Satisfies Nonfunctional Requirement Issue	PNFRI6
Satisfies Non-functional Objective	INFRO6
Constrains	P11
Satisfied by prototype feature	Epic Decomposition, Story Optimization, Task Generation, Relationship Graph

NFR ID	Nonfunctional Requirement 7
NFR7	The response time for each phase of decomposition will not exceed five seconds. Generating new results will not exceed one second.
Operationalized Functional Requirements	FR1, FR2, FR3
Satisfies Nonfunctional Requirement Issue	PNFRI7
Satisfies Non-functional Objective	INFRO7
Constrains	N/A
Satisfied by prototype feature	Epic Decomposition, Story Optimization, Task Generation

4.2.3. Specifications

Functional Specification ID	Functional Requirement
FS1	Decompose an epic into user stories without the loss of any content-critical information. Decomposition will cluster requirements by topic into individual, large stories.
Satisfies Functional Requirement	FR1
Satisfies Objectives	IFRO1, INFRO1
Satisfied by prototype feature	Epic Decomposition

Functional Specification ID	Functional Requirement
FS2	Optimize user stories into, potentially, smaller user stories without the loss of any content-critical information. The optimization should populate into Jira within five seconds. Stories should be broken into individual action groups based on action wording.
Satisfies Functional Requirement	FR2
Satisfies Objectives	IFRO2, INFRO1, INFRO3, INFRO4, INFRO5, INFRO7
Satisfied by prototype feature	Story Optimization

Functional Specification ID	Functional Requirement
FS3	Generate tasks from user stories without the loss of content-critical information. The generation should populate into Jira within five seconds. Tasks should be individual actions derived from a user story.
Satisfies Functional Requirement	FR3
Satisfies Objectives	IFRO3, INFRO1, INFRO3, INFRO4, INFRO5, INFRO7
Satisfied by prototype feature	Task Generation

Functional Specification ID	Functional Requirement
FS4	Show a directed graph of relationships for a selected object (i.e. epic, story, task, etc.). Allow the user to select the depth of relationships along with the types of relationship (i.e. dependencies, developers, etc.). The graph should render as the object is selected.
Satisfies Functional Requirement	FR4
Satisfies Objectives	IFRO4, INFRO2, INFRO3, INFRO4, INFRO5, INFRO7
Satisfied by prototype feature	Relationship Graph

Functional Specification ID	Functional Requirement
FS5	The application will be implemented into the dedicated Jira platform.
Satisfies Functional Requirement	FR1, FR2, FR3, FR4
Satisfies Objectives	INFRO6
Satisfied by prototype feature	Epic Decomposition, Story Optimization, Task Generation, Relationship Graph

Functional Specification ID	**Functional Requirement**
FS6	The implicit parent-child relationship of developers to the epics, user-stories, and tasks will be shown as clusters.
Satisfies Functional Requirement	FR4
Satisfies Objectives	IFRO4, INFRO2, INFRO3, INFRO4, INFRO5, INFRO7
Satisfied by prototype feature	Relationship Graph

Functional Specification ID	**Functional Requirement**
FS7	The explicit parent-child relationship between the epics, user-stories, and tasks will be shown as a tree.
Satisfies Functional Requirement	FR4
Satisfies Objectives	IFRO4, INFRO2, INFRO3, INFRO4, INFRO5, INFRO7
Satisfied by prototype feature	Relationship Graph

Functional Specification ID	**Functional Requirement**
FS8	The user is able to see the tree of the explicit relationships, or the cluster of implicit relationships separately or together
Satisfies Functional Requirement	FR4
Satisfies Objectives	IFRO4, INFRO2, INFRO3, INFRO4, INFRO5, INFRO7
Satisfied by prototype feature	Relationship Graph

[5] Preliminary Prototype

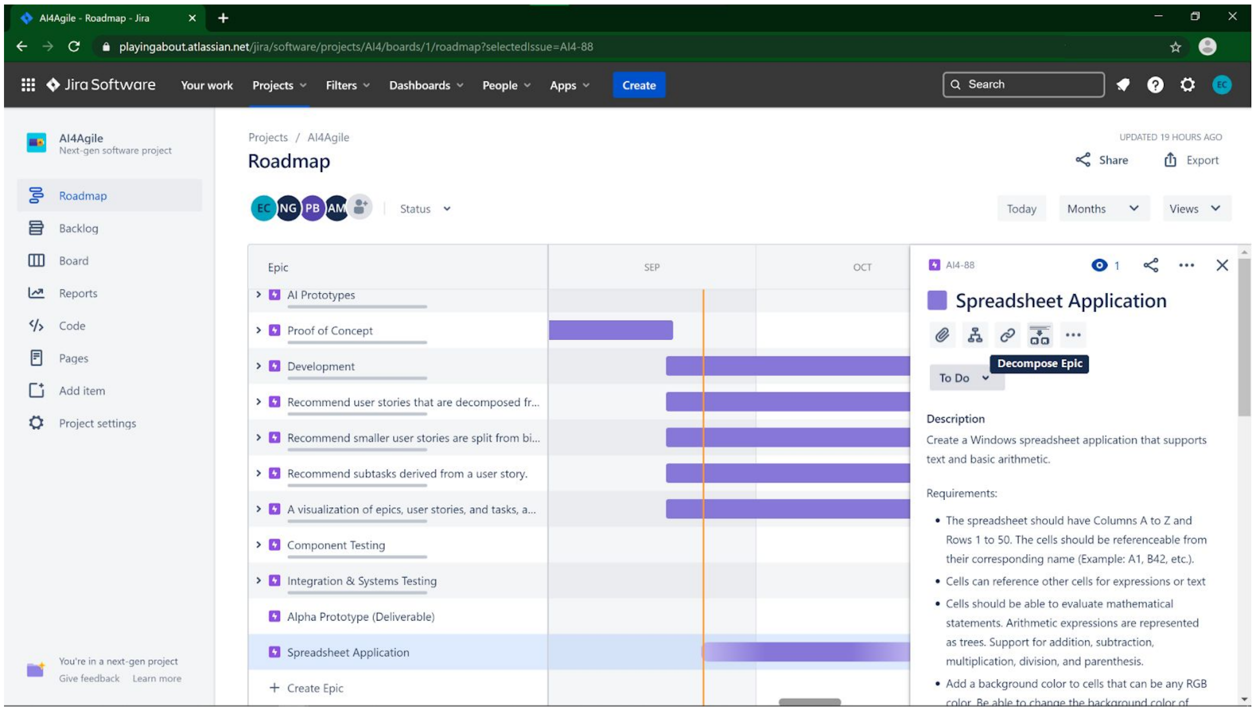
The preliminary prototype of the AI4Agile application is outlined within individual documents on a component basis. Each document contains a description, implementation, and testing metric

Component document Links:

- Epic Decomposition
- Story Optimization
- Task Generation
- Relationship Graph

[6] Prototype Interface Mock-ups

Scenario 1:



AI Prototypes
Proof of Concept
Development
Recommend user stories that are decomposed fr...
Recommend smaller user stories are split from bi...
Recommend subtasks derived from a user story.
A visualization of epics, user stories, and tasks, a...
Component Testing
Integration & Systems Testing
Alpha Prototype (Deliverable)
Spreadsheet Application

Spreadsheet Application

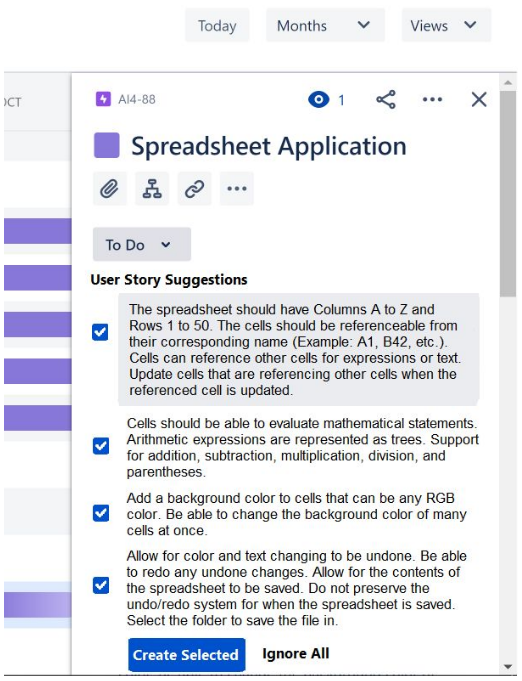
Description

Create a Windows spreadsheet application that supports text and basic arithmetic.

Requirements:

- The spreadsheet should have Columns A to Z and Rows 1 to 50. The cells should be referenceable from their corresponding name (Example: A1, B42, etc.).
- Cells can reference other cells for expressions or text
- Cells should be able to evaluate mathematical statements. Arithmetic expressions are represented as trees. Support for addition, subtraction, multiplication, division, and parenthesis.
- Add a background color to cells that can be any RGB color. Be able to change the background color of

Frank is a software requirements analyst, and he wants to speed up the process of breaking an epic full of requirements into smaller user stories. For this purpose, he installs the AI4Agile plugin to Jira. Frank creates a new epic, puts the requirements in with the format specified in the plugin's user manual, and clicks the *Decompose Epic* button.



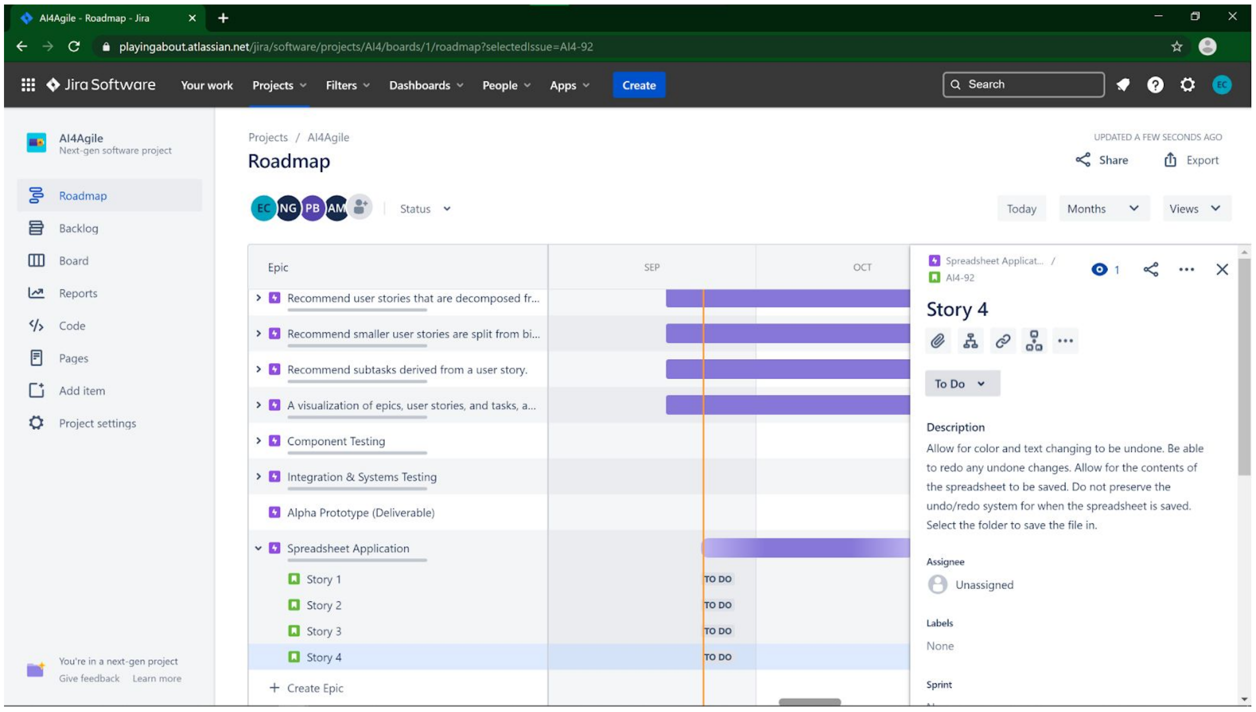
Spreadsheet Application

User Story Suggestions

- ✓ The spreadsheet should have Columns A to Z and Rows 1 to 50. The cells should be referenceable from their corresponding name (Example: A1, B42, etc.). Cells can reference other cells for expressions or text. Update cells that are referencing other cells when the referenced cell is updated.
- ✓ Cells should be able to evaluate mathematical statements. Arithmetic expressions are represented as trees. Support for addition, subtraction, multiplication, division, and parentheses.
- ✓ Add a background color to cells that can be any RGB color. Be able to change the background color of many cells at once.
- ✓ Allow for color and text changing to be undone. Be able to redo any undone changes. Allow for the contents of the spreadsheet to be saved. Do not preserve the undo/redo system for when the spreadsheet is saved. Select the folder to save the file in.

Create Selected Ignore All

Now, Frank looks at the suggestions the AI came up with. If he decides he doesn't like any of these suggestions, he can click the *Ignore All* button to go back to his Spreadsheet Application epic. To edit a story, he can click on its text box, then save or cancel those edits. To ignore a story suggestion entirely, he can uncheck its box. Once Frank is happy with his resulting user stories, he clicks *Create Selected*.


Projects / AI4Agile

Roadmap

Epic
Recommend user stories that are decomposed fr...
Recommend smaller user stories are split from bi...
Recommend subtasks derived from a user story.
A visualization of epics, user stories, and tasks, a...
Component Testing
Integration & Systems Testing
Alpha Prototype (Deliverable)
Spreadsheet Application
Story 1 — TO DO
Story 2 — TO DO
Story 3 — TO DO
Story 4 — TO DO

Story 4

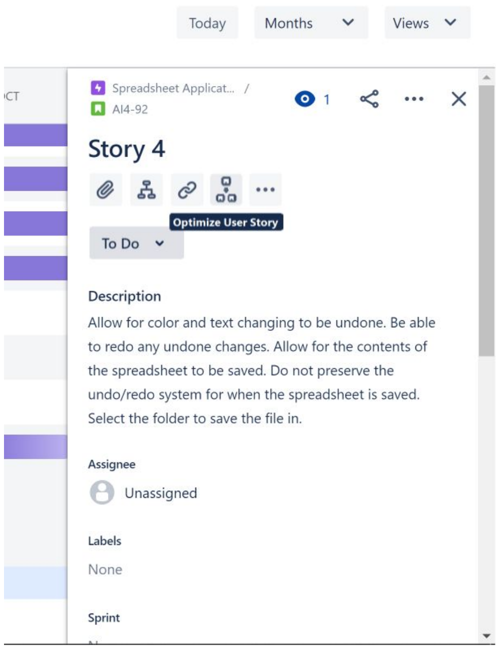
Description

Allow for color and text changing to be undone. Be able to redo any undone changes. Allow for the contents of the spreadsheet to be saved. Do not preserve the undo/redo system for when the spreadsheet is saved. Select the folder to save the file in.

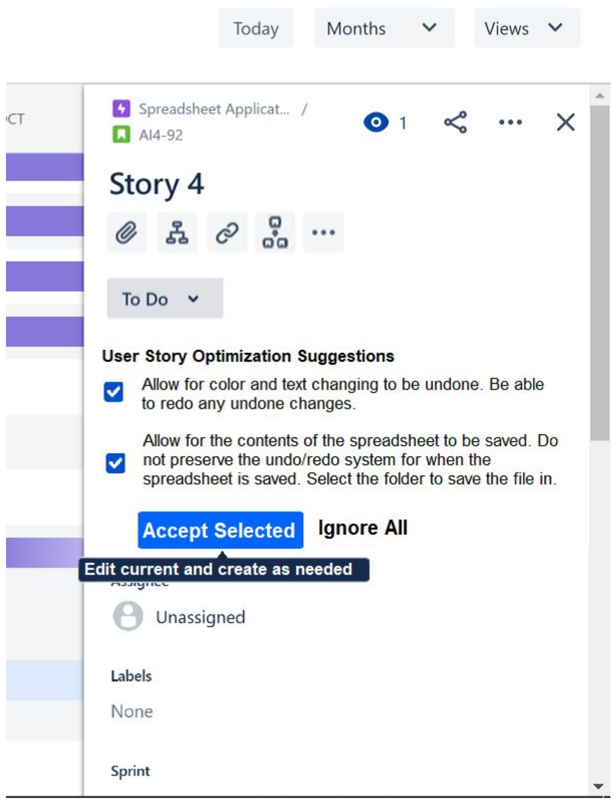
Assignee: Unassigned

Labels: None

The app has now created the 4 user stories that Frank approved under the heading of the Spreadsheet Application epic.



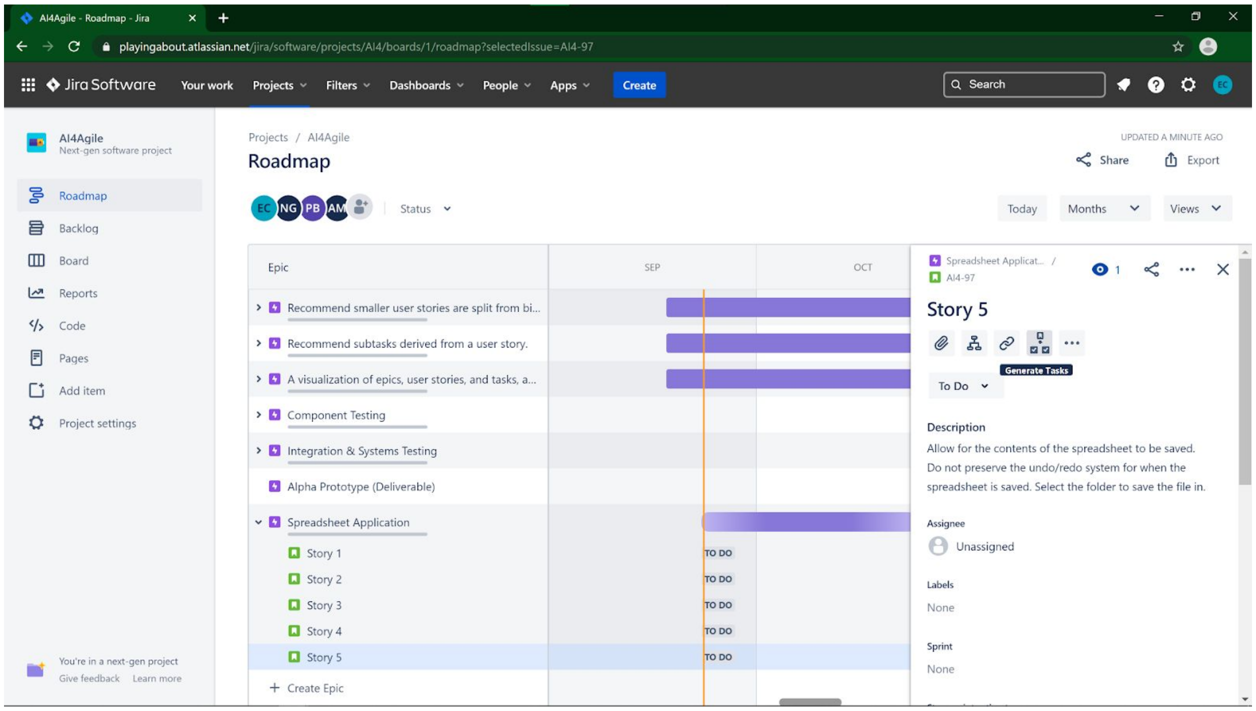
If Frank wants to continue the process of breaking up epics, he can go into one of the user stories and click *Optimize User Story*. Depending on the size of the user story already, it might be broken down into multiple smaller stories, or left alone if it’s already optimized.



User Story Optimization Suggestions

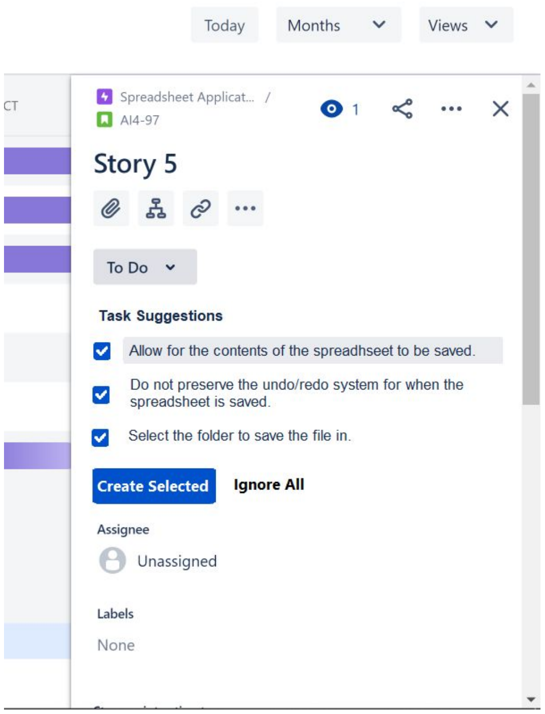
- [x] Allow for color and text changing to be undone. Be able to redo any undone changes.
- [x] Allow for the contents of the spreadsheet to be saved. Do not preserve the undo/redo system for when the spreadsheet is saved. Select the folder to save the file in.

Once the Story Optimization Suggestions are generated, Frank has the same options as with the previous stories: to select or deselect stories via the checkboxes, make edits, or ignore all suggestions. As he hovers over the *Accept Selected* button, Frank is warned that this option will edit his currently selected user story as well as create a new one to accommodate for the splitting. He clicks anyway, since this is what he wanted.



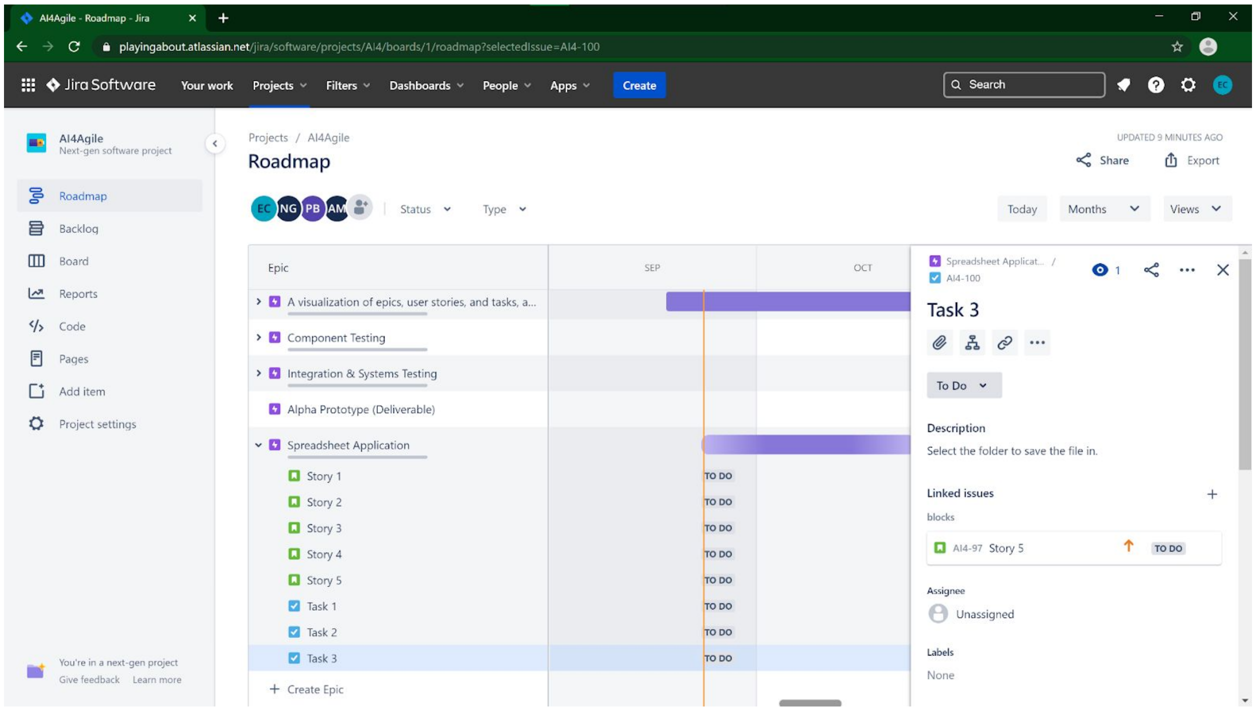
Allow for the contents of the spreadsheet to be saved. Do not preserve the undo/redo system for when the spreadsheet is saved. Select the folder to save the file in.

Now there are five user stories, since Story 4 was optimized into two separate stories. To continue the decomposition process, Frank opens a story and clicks *Generate Tasks*.



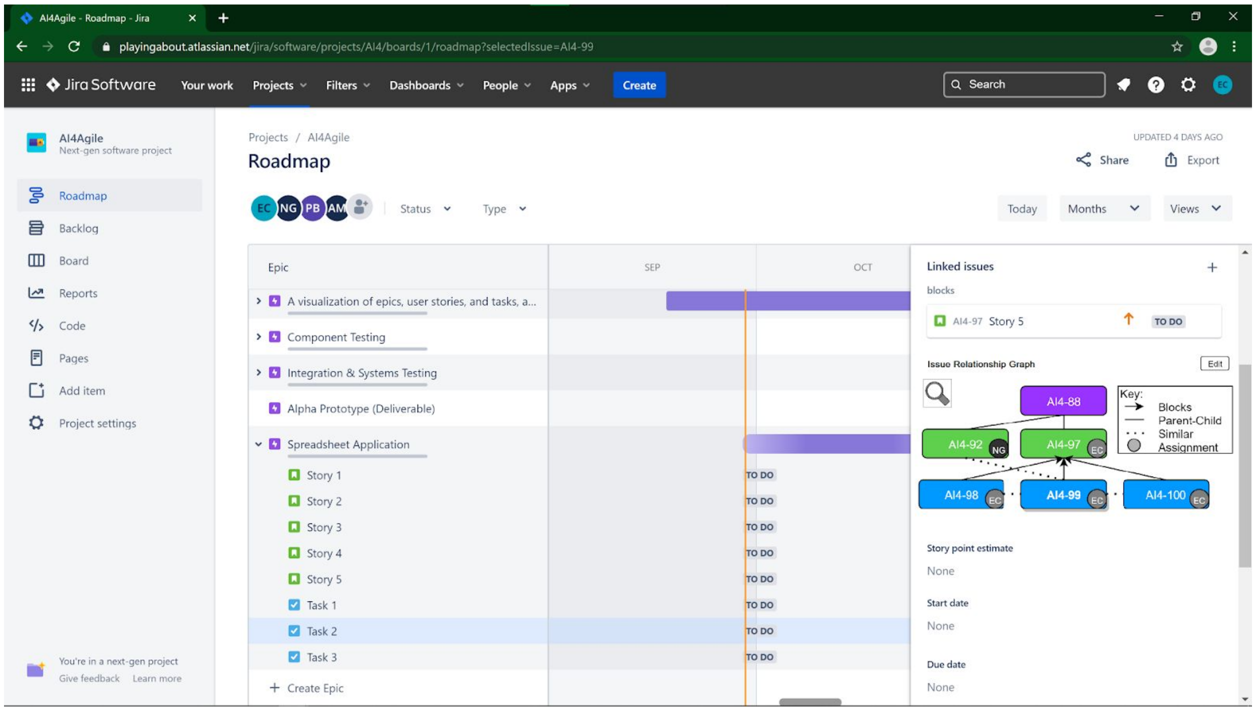
- [x] Allow for the contents of the spreadhseet to be saved.
- [x] Do not preserve the undo/redo system for when the spreadsheet is saved.
- [x] Select the folder to save the file in.

As before, the options include selecting or deselecting individual suggestions, editing, and ignoring all suggestions. Once he's happy with the tasks, he clicks *Create Selected*.

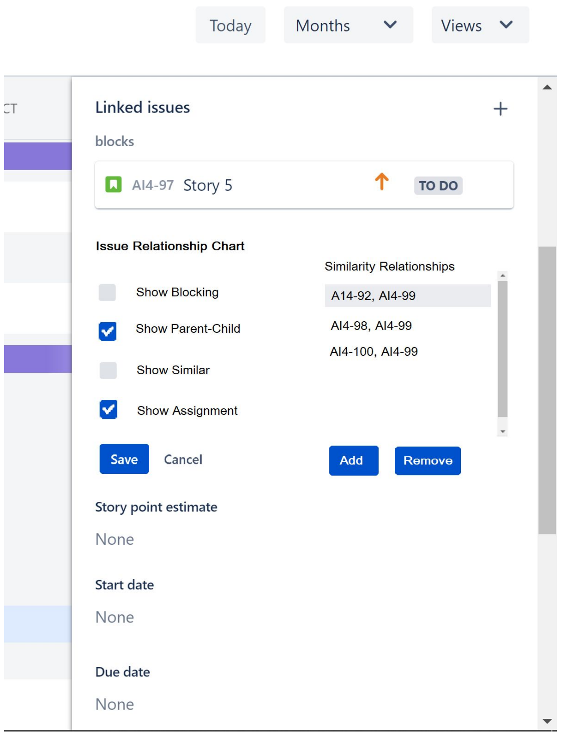


The tasks have now been created, and linked to their parent story to indicate a blocking relationship. All Frank had to do was make some decisions and maybe edits, and now he's got one story fully decomposed!

Scenario 2:



Madison is a scrum master, and she wants to take a closer look at some of the tasks to make sure the developer assignments make sense. To do this, she uses the AI4Agile Issue Relationship Graph feature by navigating to a certain epic, user story, or task, and finding the graph portion. From here, she has the options to *Zoom In* on the graph, or *Edit* it to change which types of relationships are shown. Madison wants to show more relationships, so she clicks *Edit*.

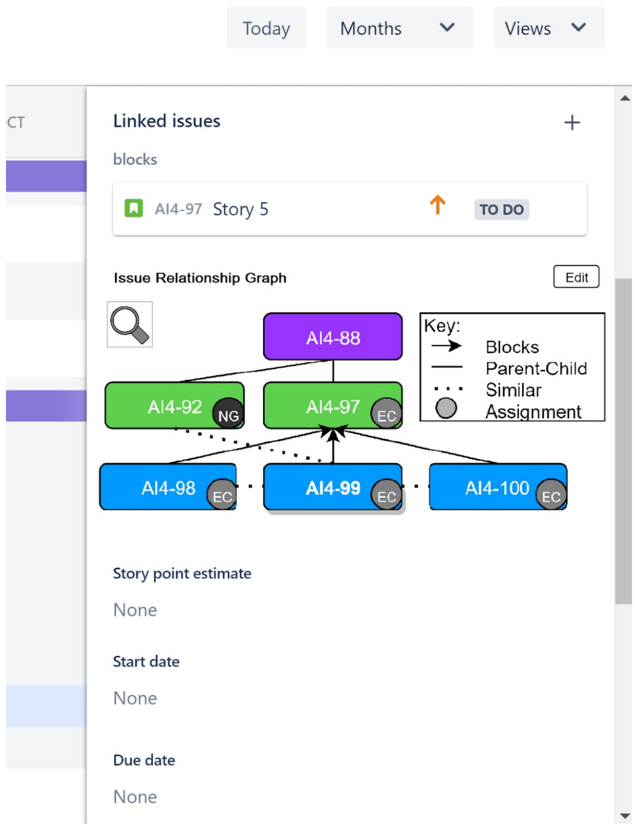


- [ ] Show Blocking
- [x] Show Parent-Child
- [ ] Show Similar
- [x] Show Assignment

Similarity Relationships

A14-92, AI4-99
AI4-98, AI4-99
AI4-100, AI4-99

In the graph's edit page, Madison checks the boxes to *Show Blocking* and *Show Similar* so that those types of relationships will populate on the graph. If she thought any of the suggested similar issues were incorrect or lacking, she could use the menu on the right to change them. Next, she clicks to *Save* these options.



Now, Madison can see all the available relationships between the task she has selected and other tasks, stories, and the epic they came from. She is better able to decide whether the team members she assigned to these pieces make sense given the relations between them.

[7] References

1. Agile Advantages For Software Development And Your Business. (n.d.). Retrieved from https://devcom.com/tech-blog/agile-advantages-for-business/
2. Hoa Khanh Dam, Truyen Tran, John Grundy, Aditya Ghose and Yasutaka Kamei, *Towards effective AI-powered agile project management*, Proceedings of the 41st International Conference on Software Engineering (ICSE 2019), New Ideas and Emerging Results.