AI4Agile Project Description

CPTS 421 - Bolong Zeng
By: Phong Bach, Emily Cawlfield, Nain Galvan, and Aric Monary
Mentor: Skip Baccus
Date Submitted: 9/11/20

Project Description

This project aims to create an application for the Jira platform, in order to assist Agile teams in user story streamlining. It consists of five distinct parts: the planning engine, representation learning engine, optimization engine, analytics engine, and the conversational dialog engine. Overall, these components work 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.

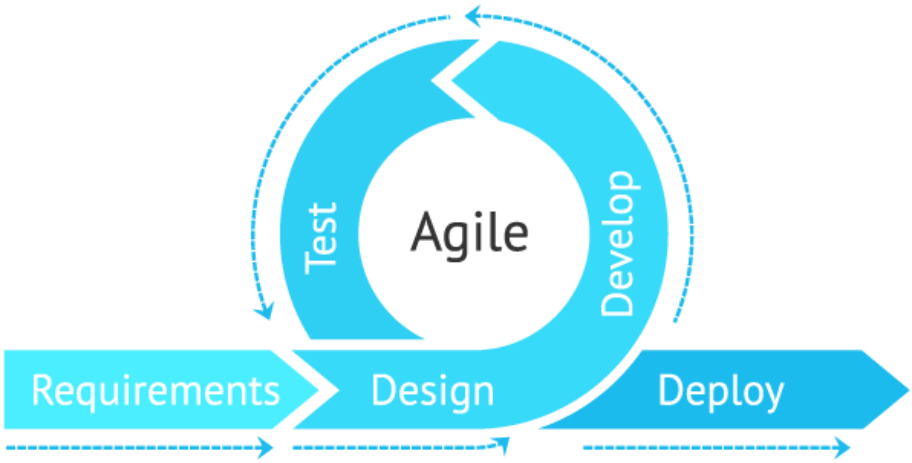

Figure 1: The four parts typically found within the Agile software development process. [7]

Individually, the purposes of the engines are as follows. The representation learning engine's main focus is Natural Language Processing, the optimization engine helps with the planning and analytics tasks, and the planning engine has the responsibility of certain deep learning aspects. The analytics engine is where the visualization elements are generated, and also where stories and tasks are extracted from larger user stories. 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.

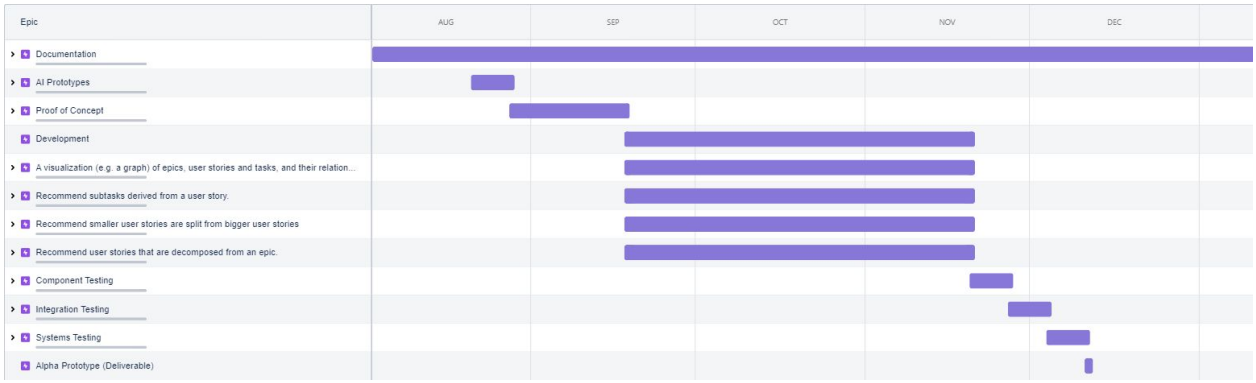

Figure 2: Gantt chart outlining the development plan and deliverables for the project.

Features

Functionality

The main functionalities of the AI4Agile Jira application are as follows:

- Decomposing epics into user stories
- Optimizing stories
- Generating tasks for sprints
- Visualization of explicit and implicit relationships between stories, tasks, and epics

Limitations

The main limitations are in AI training materials. Most data associated with Agile development is from open source projects and may not reflect how Agile is conducted professionally.

A majority of the literature surrounding the application of AI to Agile Development involves addressing issues that pertain to the topic of risk. Although the scope of the project is to leverage AI to improve Agile Development, new features pertaining to risk shall not be implemented until further development outside of the SCORE 2021 competition phase of the project. Such risk features may include:

- Story point estimation [2]
- Delivery predictions for sprints [3]
- Issue due date delays [4]
- Classification of issue-related risks [5]
- Sprint delays related to tasks [6]

Literature Review

The premise of the application is based upon a 2019 research paper titled *Towards effective AI-powered agile project management* [1]; a framework that contains four engines.

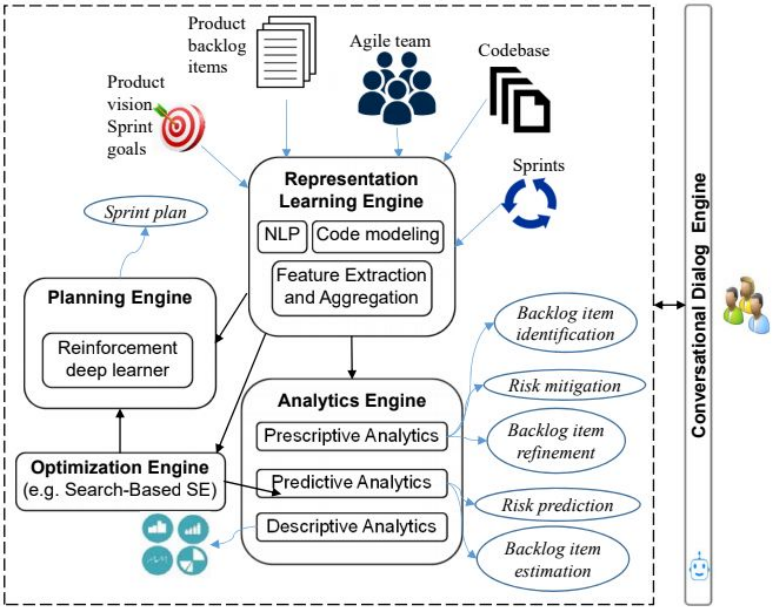

Figure 2: The complete AI4Agile software architecture as laid out by Hoa Khanh Dam et al. [1].

There are quite a few Agile project management software. Some Agile solutions available are Atlassian Jira, Axosoft, Assembla, Agilean, SprintGround and VersionOne. However, there are not any existing applications that provide support for the Agile development process and are AI-powered. The solutions mentioned are not capable of leveraging users' information to predict and offer recommendations. To be able to assist the Agile teams, the program must be able to understand both structured and unstructured data that form epics, user stories, or tasks.

Figure 3: Places within the sprint process where AI will be implemented [1].

According to *Towards effective AI-powered Agile project management* [1], some of the latest advances in deep learning-based NLP techniques such as word2vec, paragraph2vec, Long Short-Term Memory, or Convolutional Neural Networks can be useful to produce quality results for NLP tasks. The pros and cons of the related techniques are shown below in Tables 1 and 2.

Techniques	**Pros**	**Cons**
Word2vec	● Scalable model that generated word embeddings for large collection of texts ● Feed the model raw text and it outputs word vectors	● Word sense is not captured separately
Glove	● Faster training time than word2vec ● Give better semantic comparing to word2vec	● Larger memory footprint ● Word sense is not captured separately
Adagram	● Capture different sense of a word based on the surrounding words ● Perform context sensitive operation	● Written in Julia
Fasttext	● Useful in word sentence similarity task on corpus with misspelling ● Better quality for even word that occur rarely in collection of texts	● Multiple sense embeddings is not captured
Wang2vec	● Takes into account word order	● Word sense is not captured separately

Table 1: Words embedding models

Techniques	Pros	Cons
Paragraph2vec	● Adaptation of Word2vec ● Useful to generate sentence or document embeddings	● May not be sufficient in some cases where sentence similarity is high priority
Skip-thought	● Useful where input collection of texts has sentence continuity	● Neighboring sentences have to be semantically related
Long Short-Term Memory	● Does better than word order independent models ● Remember state information	● Needs labeled data for training ● Requires a lot of resources and time to train
Convolutional Neural network	● Fast prediction ● Can be used for both regression and classification problems ● Can be trained with any number of inputs and layers	● Cannot know how much each independent variable is influencing the dependant variables ● Computationally expensive ● Depends on a lot of training data

Table 2: Sentence/Document embeddings

Some other potential predictive models that can be used are Random Forests, Stochastic Gradient Boosting Machines, Deep Neural Networks with Dropout, and Recurrent Highway Network. From “Predicting Delivery Capability in Iterative Software Development” by Choetkiertikul et al. [3], ensemble methods such as Random Forest or Gradient Boosting Machines are highly recommended when it comes to building software analytic models.

Stakeholder Identification and Considerations

Sponsors
Research: Hoa Khanh Dam, University of Wollongong, Australia
Faculty: Bolong Zeng

Mentor Contact Information
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.

Special Considerations and Needs of the Stakeholders

The software requirements analyst group of potential users in particular might have more considerations for the speed and efficiency of the decomposition process as they are likely already experts at the process.

A need of the scrum master group of users could be the relationship visualizations being available outside of the steps of epic/user story/task creation, for viewing to not be inconvenient outside of that process.

Due to potential changes to Jira, the application may need continual support to remain functional. The specific components that are subject to change are:

- Atlassian Connect Express (ACE) framework - The core component that connects the application to JIRA.
- Atlassian User Interface (AUI) library - Basis for all UI within JIRA.
- Programming Specific libraries - Support libraries used with components of the application. Such examples include, Keras and Tensorcore python libraries.

Such changes to either framework or library may cause the application to degrade, hindering the usability of the app for all stakeholders. Specific functions can become unusable pending specific library changes.

References

1. 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.
2. M. Choetkiertikul, H. K. Dam, T. Tran, T. T. M. Pham, A. Ghose, and T. Menzies, *A deep learning model for estimating story points*, IEEE Trans. Softw. Eng., vol. PP, no. 99, p. 1, 2018.
3. M. Choetkiertikul, H. K. Dam, T. Tran, A. Ghose, and J. Grundy, *Predicting Delivery Capability in Iterative Software Development*, IEEE Trans. Softw. Eng., vol. 14, no. 8, pp. 1–1, 2017.
4. M. Choetkiertikul, H. K. Dam, T. Tran, and A. Ghose, *Predicting the delay of issues with due dates in software projects*, Empir. Softw. Eng., vol. 22, no. 3, pp. 1223–1263, 2017.
5. M. Choetkiertikul, H. K. Dam, T. Tran, and A. Ghose, *Characterization and prediction of issue-related risks in software projects*, Proceedings of the 12th Working Conference on Mining Software Repositories (MSR), 2015, pp. 280–291.
6. M. Choetkiertikul, H. K. Dam, T. Tran, and A. Ghose, *Predicting delays in software projects using networked classification*, Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2015, pp. 353–364.
7. https://devcom.com/tech-blog/agile-advantages-for-business/