Application of Bayesian Belief Network for Agile Kanban Backlog Estimation

Eric Weflen, Kevin Korniejczuk, Sharon Lau, Steve Kryk, Cameron MacKenzie, Iris V. Rivero
Iowa State University, Department of Industrial and Manufacturing Systems Engineering, 3004 Black Engineering Building, Ames, IA, 50011-2164, USA

Abstract
This paper presents an approach based on influence diagrams for reducing uncertainty in Agile Kanban backlog feature completion time. Agile project management techniques, including SCRUM and Kanban, are prevalent in software development and spreading to other product development fields. A key artifact of Agile is the product backlog, containing work which needs to be completed by the development team. Internal and external stakeholders often require projections for completion of backlogged requests or features. Current estimation techniques such as duration assignments through planning poker and the use of story points to calculate velocity require persistent team input, while task counting has limited accuracy. Therefore, an influence diagram (also known as a Bayesian belief network) was generated to probabilistically assess factors influencing the completion time of backlog items. Statistical functions and uncertainty nodes were validated through data collected from a product development team practicing Agile Kanban. In addition to lowering the barrier to adopting backlog estimation, this model accounts for factors influencing lead time that current techniques disregard such as re-prioritization and feature or request additions. This approach can provide a simpler, more robust representation of project backlog while effectively using team resources.

Keywords
Agile methods, Influence diagram, Project management, Kanban, Project estimation

1. Introduction
Project management methods are important elements in aiding a company’s efficiency. Agile project management practices, including SCRUM and Kanban, are starting to gain widespread acceptance due to their adaptability and versatility. Agile methods are iterative and incremental approaches that were initially developed for the software development community with the goal of providing visibility and transparency, making it more adept to change than traditional project management methods.

Kanban is a highly visual method of managing projects through the use of a physical board displaying tasks as they flow through the development process. This process shares a name with the lean manufacturing concept on which it was based. The Kanban board typically consists of a backlog containing tasks to be worked on and several columns to indicate the status of those tasks, each with a done column for completed tasks as demonstrated in Figure 1. Many variations of columns between backlog and the final done column exist to suit the process that the product development team follows. New tasks are added to the backlog section based on priority level. Kanban is a pull system, which means that tasks can only move from one column to the next when the latter task is completed, creating a vacancy. An established work-in-progress (WIP) limit caps the number of allowed tasks in each column to reduce the volume of tasks that are actively being worked on, with the goal of improving the team’s ability to respond to change. Similar to the use of WIP limits in lean manufacturing, the team may uncover inefficiencies or issues with their process that can be corrected instead of being covered up.
The benefits of adopting an Agile style process include the reduction of cost in the transfer of information and the reduction in time between a decision and the outcome of that decision. However, many uncertainties affect the planning process to ensure timely delivery of any product launch. This makes it challenging to obtain accurate time estimates for when a task in the backlog will be completed. Inaccurate time estimates may lead to a loss of trust from key stakeholders and in the Kanban method. While solutions have been developed to aid the Agile project planning process, much of this research has focused on techniques other than Kanban.

Currently, some Agile teams use a method called Story Point Estimation to calculate approximate completion dates for tasks. While traditional methods of estimation are typically done in time formats such as days, weeks, or months, Story Point Estimation assigns each task, sometimes referred to as a user story, a point value known as story points, typically following a modified Fibonacci or base-2 number sequence. The team assigns points to each task, and subsequently, the team’s velocity can be measured based on how many points they complete in a given time frame. This velocity can then be used to compute an estimated completion date range for specific tasks. Even though an estimate of time can be derived from velocity, this approach ignores several factors which influence the tasks completion date, such as backlog reprioritization and the addition of new tasks.

To address this issue, a probabilistic model representing the factors that influence the movement of tasks in the backlog was developed to obtain lead time estimates for when work will start on tasks in the backlog. Task completion time is an important piece of information regularly requested by key stakeholders. While this model will only provide an estimate for work start date, this is often the portion of the lead time with the most uncertainty. Time estimates remain a challenge for Agile Kanban practices, and little research has been done to solve this problem. It can be difficult for project teams to justify the ongoing effort needed to support current estimation techniques in order to fulfill periodic requests, leading to teams discontinuing this service.

This study presents an approach for modeling the probabilistic relationship between task completion lead times and variables affecting them through the creation and application of an influence diagram. This influence diagram is then applied in the creation of an estimation tool that may eliminate the need for active project sizing and velocity calculations. Through this method, it may be possible to reduce the level of effort needed for project estimation by up to a factor of ten, by using data collected from a few weeks to form probabilistic estimates for the remainder of the year. The team needs only to maintain the model when team dynamics change, such as the addition of new team members or the implementation of a change to their process. The drastic reduction in maintenance allows teams that are required to produce estimates to do so with fewer wasted resources. There may also be teams that are currently not providing project estimates due to resource constraints even though their customers may desire estimates. This model may allow these teams to provide better customer service through estimation without an ongoing resource drain.

Section two of this paper introduces our model and assumptions made in creating the influence diagram. The model’s application to Agile project management is shown in section three. Finally, the model’s limitations and impact on project management are discussed in section four.
2. Model and Methodology

2.1 Modeling Approach
This study presents a model to assist Kanban teams to provide an accurate estimate of when work would start on a specific backlogged task by taking into account team performance as well as other factors, such as task sizes and movement in the backlog. In this section, we discuss the influence diagram that was developed for this purpose. We will review the factors that influence the length of time before a project in the backlog reaches the active workflow and how they link together to form a network diagram. Every team sets up a unique workflow that tracks tasks that have left the backlog and become WIP. We do not try to include the workflow in our model and instead focus on when work will start. While others have tried to include this workflow in their model, it was intentionally left out of this model to reduce complexity, and thus the effort needed for teams to adopt this tool. Kanban aims to be a lightweight, easy to implement Agile methodology, and thus we present an easily implemented tool created with the goal of fitting a broad spectrum of project teams with minimal input. While building the influence diagram, assumptions were made about how multiple factors will influence each other. There are limitations to this approach, which will be discussed later in this paper, due in part to the intentional attempt to keep the system simple to use by Agile teams whose members may not be familiar with influence diagrams or statistical modeling.

2.2 Developing the Model
To generate the model, first, the nodes were developed by listing the factors that influence the backlog item movement. These factors were identified through a review of the literature and were validated by a Kanban project team at our industry partner Andersen Corporation (Bayport, Minnesota US) [6, 7]. Once the nodes were identified, links were formed between them to represent how they influence each other (refer to Figure 2). The final deterministic node in this diagram is the lead time to exit the backlog, which is represented by a rounded rectangle. Based on the task’s backlog position, the lead time to exit the backlog will change. The team must balance the needs of all the customers who may have competing priorities when it comes to the order projects should be completed. Using this tool, they can provide the customers with feedback on when work will start on their requested project.

![Figure 2: Bayesian network representing factors that influence lead time until work starts on a task](image)

The first node in this diagram, the rectangle on the left side of Figure 2, is the Backlog Position. The team makes a decision of where to position a new task within the backlog. This Backlog Position node is connected to five nodes as seen in Figure 2, indicating the input of this node affects the other nodes to which it is connected.

Following the Backlog Position node is a series of four chance nodes, represented by circles, which are probabilities derived from historical data input from the Kanban team. New Tasks Added represents the number of tasks that could be added to the backlog. Tasks added ahead of the task of interest will delay the start of work. Tasks Canceled
expresses how many backlogged tasks were deleted from the backlog. Deleting tasks from a backlog opens the
opportunity for tasks ranked below these deleted tasks to move up in the backlog, therefore affecting the lead time for
those tasks to exit the backlog. Reprioritized Backlog observes how far the task in question moves backward when
tasks previously ranked below this task are moved ahead of its position in the backlog queue. Tasks located further in
the backlog are exposed to this uncertain environment for longer periods of time which may impact lead time until the
task exits the backlog. The last in this series of nodes is Average Backlog Item Size. This node records the average
story points per task prioritized above the task of interest.

This series of four nodes in addition to the original Backlog Position node all feed into the Virtual Backlog Position
node. Since velocity calculations are based on the current backlog points and the team’s historical velocity, the Virtual
Backlog Position in Figure 2 represents an intermediate consequence node. This node is the summation of the task’s
original backlog position with the new tasks being added, the tasks removed from the backlog subtracted, and any
changes based on reprioritization, giving the virtual location of the project in the backlog. This virtual backlog position
is an improvement to the current story point estimation technique which assumes none of these factors are present.
This is one area where this type of statistical modeling may be able to provide information that current modeling
techniques do not.

Average Team Daily Velocity is impacted by the Virtual Backlog Position but also contributes to the lead time for a
task to exit the backlog. The team’s velocity is measured based on how many story points the team completes in a
given timeframe. A combination of Average Team Daily Velocity and Virtual Backlog Position are used in calculating
the Days Until Work Starts on Task by dividing the Virtual Backlog Position by Average Team Daily Velocity.

2.3 Assumptions and Limitations
A number of assumptions are built into this model. It is assumed that 1) the project team is re-prioritizing projects on
a regular basis. If there are recurring events where a large re-prioritization occurs, they may not be accounted for in
this model. 2) There is a continuous flow of new requests. If there are irregular events that cause a change in the
volume of new project requests that are not captured in the historical data, this model will not take them into account.
3) Team dynamics and workflow setup do not change, including the number of team members, WIP limits, and
workflow breakdown. While these items are not directly modeled, new data would need to be collected by the team
to recalibrate the distributions. Because this model is based on historical data, the volume of data will improve the
estimates. At the same time, there is a risk that events occur that are not represented by historical data, in which case
an outcome may occur to which the model had assigned a probability of zero.

3. Application and Results

3.1 A Review of the Agile Kanban Model
The software Netica (Norsys Software Corporation) was used to transform the influence diagram (Figure 1) into a
model. As can be seen in Figure 3, the model contains the same nodes and interactions as in Figure 2. However, the
model contains categories for each node where team data can be entered, which are shown as rows in each node shown
in Figure 3. By collecting and monitoring team data and entering the data into each respective node, the distributions
are tailored towards the Kanban team. In order to determine how many days until work begins on a backlogged item,
the backlog position is utilized as a decision node titled Backlog Position. By adjusting the data in the Backlog Position
node, the distribution of estimated days to completion is presented in the final consequence Days Until Work Starts
on Task node. This output can be utilized by the project team to communicate estimates to customers when work will
start on their requested task, or to decide where to place a customer request within the backlog.

3.2 Data Collection Methods
In order for the model to provide value to the Kanban team, effective data collection is needed. For each chance node
in Figure 3, data was provided and collected from a Kanban team at Andersen Corporation. Data was collected for:
the amount of story points in a task, the amount of tasks canceled while an observed task is in the backlog, the amount
of tasks reprioritized while an observed task is in the backlog, the amount of story points completed each day, and the
number of items added with a higher priority. As more data is available to populate the model, the forecast will be
more accurate. Data were collected for 6 weeks at Anderson Corporation to estimate the probabilities for each
uncertainty in Figure 3. The values in Figure 3 represent the collected probabilities.
3.3 Calculations
The Virtual Backlog Position node is calculated as the summation of Backlog Position, Reprioritized Backlog, and New Tasks Added with Tasks Canceled subtracted from this value. The Days Until Work Starts on Task node is calculated by dividing the Virtual Backlog Position by the Average Team Daily Velocity. Even though the lead time for a task to exit the backlog is a deterministic function of two variables, since those variables are either uncertain or determined by other uncertainties, the lead time will also be uncertain. As indicated in the influence diagram in Figure 3, adjusting the backlog position impacts the virtual backlog position, therefore impacting the number of days until the task begins. Additionally, if a team shows recent unusual consistency in daily velocity over a period of time, the daily velocity can be fixed.

3.4 Results
By utilizing the probabilities provided at the final consequence node by fixing the backlog positions to 5, 15, and 25, three discrete cumulative distribution functions (CDF) were created Figure 4. The x-axis indicates the number of days until the task starts while the y-axis indicates the probability that the task begins within a number of business days as specified by the x-axis.
Figure 4 shows that as the position in the backlog increases, the time to completion increases. The results from Figure 4 indicate realistic expectations since as the Backlog Position increases, more tasks have a higher priority to be completed. Therefore, since more tasks need to be completed before the new task is pulled from the backlog, the lead time should increase. When deciding where to place the task in a backlog, it is important to know when that task ideally should be completed. For example, if a customer wants a task completed within 26 days, according to Figure 4, placing the task in Position 5 would ensure 100% probability that the task will be completed on time, placing the task in Position 10 would ensure a near 80% probability that the task will be completed on time, and placing the task in Position 15 would roughly guarantee a 10% probability that the task will be completed on time.

4. Conclusions
The objective of this study was to provide an alternative approach to project estimation for Agile Kanban project teams to reduce the level of effort required to generate these estimates. This was done with the goal of maintaining or improving the accuracy of these estimates by accounting for the most significant uncertainties and risks that may influence lead time. In addition, it was intended to create a tool that was easy enough to use by those lacking a statistics or risk analysis background. The results are in the form of probabilistic forecasts, which are more appropriate for describing the uncertain future than deterministic approaches which are likely to be wrong because the future is uncertain. Future studies are still needed to validate the robustness of results generated by this model.

Acknowledgments
The authors would like to thank Kaine Wadsley and the Kanban project team at Andersen Corporation for their feedback and data collection efforts.

References
5. Buckl, S. et al., 2010, Essential Scrum: a practical guide to the most popular agile process, Addison-Wesley, Boston.