International Journal of
Information Systems and Project Management

ISSN (print):2182-7796, ISSN (online):2182-7788, ISSN (cd-rom):2182-780X

Available online at www.sciencesphere.org/ijispm

Continual learning with a Bayesian approach for evolving the baselines of a leagile project portfolio

Sagar Chhetri
Texas Tech University
Department of Industrial, Manufacturing & Systems Engineering
905 Canton Ave., Box 43061, Lubbock Texas 79409
USA
sagar.chhetri@ttu.edu

Dongping Du
Texas Tech University
Department of Industrial, Manufacturing & Systems Engineering
905 Canton Ave., Box 43061, Lubbock Texas 79409
USA
dongping.du@ttu.edu

Abstract:
This article introduces a Bayesian learning approach for planning continuously evolving leagile project and portfolio baselines. Unlike the traditional project management approach, which uses static project baselines, the approach proposed in this study suggests learning from immediately prior experience to establish an evolving baseline for performance estimation. The principle of Pasteur's quadrant is used to realize a highly practical solution, which extends the existing wisdom on leagile continuous planning. This study compares the accuracy of the proposed Bayesian approach with the traditional approach using real data. The results suggest that the evolving Bayesian baselines can generate a more realistic measure of performance than traditional baselines, enabling leagile projects and portfolios to be better managed in the continuously changing environments of today.

Keywords:
leagile project portfolio; evolving Bayesian baselines; continuous planning/learning; performance measurement; decision making.

DOI: 10.12821/ijispm080403

Manuscript received: 3 May 2020
Manuscript accepted: 13 November 2020

1. Introduction

Today's project management environment is much more dynamic and complicated than it has been in the past few decades. These days, organizations often need to continually change their product requirements to adapt to changes in the project environment [1]. Furthermore, the increased demand for fast project delivery with changing conditions has underlined the necessity for project managers to look for better project management solutions and resources.

According to a report on the talent gap for the years 2017–2027 published by the Project Management Institute (PMI), by 2027, for the 11 countries analyzed, employers will need 87.7 million individuals working in project management-oriented roles [2]. This surge in demand for employees could result in a $207.9 billion loss globally. Moreover, the effectiveness of project management execution is rapidly decreasing [3]. The 2018 CHAOS Report found that only 14% of projects completed in 2017 were genuinely successful; the remaining 86% accounted for challenged or failed projects [4]. McKinsey and Company reported that 17% of large information technology (IT) projects with project budgets over $15M go extremely wrong, threatening the existence of the whole company [3]. Project complexity negatively impacts project success, and the percentage of projects with high complexity rose from 35% in 2013 to 41% in 2018 [5]. Increasing project complexity poses significant challenges in assessing project performance. Continually evolving projects and portfolios require an evolving scale of measurement to accurately identity failures and successes. It is certain that the project management world will experience an increase in the complexity of IT projects, where traditional tools and models like the waterfall model will not be sufficient to measure the performance of modern dynamic projects [6].

The published studies discussed in this article (refer to literature review section) mainly focused on the growth of project complexity and the negative impact of massive project failures, risk factors, and success criteria; however, none of them explored whether the scale of the performance measures used in the current project management industry was effective for modern projects. The aim of this study is to establish a new straightforward tool (refer to the proposed evolving baseline method section) for managers that will allow them to measure leagile project and portfolio performance with respect to dynamic and evolving baselines. Specifically, a statistical model is developed (refer to methodology section) to assess the evolving baselines of leagile projects by incorporating continual learning from immediate past performance (refer to results and analysis section). This will facilitate the adoption of leagile project management in a broader range of projects (refer implication of the study section), improving their management and chances of success.

The paper is organized as follows. Section 2 reviews the existing work in project/portfolio management; continuous planning delivery improvement; comparisons of leagile, Scrum, and plan-driven approaches; and existing project management challenges. Section 3 describes the methodology of this study and the SharePoint optimization data used for the evaluation. The methodology outlines the Bayesian continual learning framework and a comparison study to validate the proposed model against the traditional plan-driven model. Section 4 presents the results of this study. Section 5 provides the conclusions and limitations of the current study as well as recommendations for future studies.

2. Literature review

This section presents the findings from related research and case studies to expand the current perceptions about project and portfolio management processes. It begins with traditional plan-driven approaches and the agile delivery model, then explains the latest leagile continuous planning and delivery process.

2.1 Project management approaches and challenges

Theories and concepts about project management are ancient and have been rooted deep in all cultures from the stone age to the modern age. Project management has only become a formal discipline for delivering and managing novel ideas comparatively recently. As defined by the PMI, a project is a unique endeavor that delivers a new or enhanced but always unique solution [5]. It must always have a definitive start and end dates, and is a combination of quality, risk, procurement, time, cost, schedule, resource management, and most importantly, scope, integration, and the communication of management disciplines [5].

Project and portfolio management processes have improved since their inception; however, their failure rate has not decreased [12]. KPMG (Klynveld Peat Marwick Goerdeler) International Limited conducted a survey in New Zealand on projects managed in 2010 and 2012. It found an unexpected increase in project failure rates in 2012 when compared with the 2010 survey data [7]. Similarly, the PMI analyzed their project performance in 2015 and found only 64% of the projects met their goals; the failed projects either had scope creep or simply could not survive [8]. The report recommended the use of lessons learned to improve the project success rate. Furthermore, the 2013 CHAOS Report [9] found a similar result, where only 39% of the projects succeeded. The 2014 CHAOS Report further found that the rate of success—on-time and on-budget— was only 9% [10]. Similarly, a study was conducted to understand the confidence level of project managers regarding project success [11]. It suggested that about 75% of managers lack confidence that their projects will be successful in the end. Most respondents claimed that the uncertainty associated with success criteria makes it difficult to deliver to expectations consistently [11], [13]. A recent study [14] confirms this fact that the larger sized projects are extremely complex; thus, the successful completion rate of such larger projects is much lower than smaller projects. Basit et al. [15] looked into why projects are failing a lot more than past within recently published 33 relevant studies and found the top three reasons for in-house projects as “overrun budget & resources”, “unrealistic estimated schedule,” and “technical complexity”. It is known that the complexity always increases with uncertainty [16] and demand for faster software development [18] are creating unrealistic schedules. These studies leave us with the conclusion that project performance measurement is changing over time [19]; the way we define and measure project success in a complex environment may be outdated [15], [12], [13] and a change is required to establish a common language for success [21], [20].

Traditional plan-driven approaches like waterfall models are falling short in delivering the right product in the modern environment, especially when the project idea is extremely new and the execution happens in an uncertain and complex environment. A plan-driven approach estimates everything during the early phases and the baselines (boundaries) are defined by fixed project plans [22]. Such an approach cannot learn and improve continually based on recent executed events. As a replacement for the traditional approach, multiple types of agile and lean models are emerging to provide better solutions. One of the most famous agile delivery models is scrum. Schwaber was the first known scholar with several publications to support agile scrum as a new iterative and complex adoptive system to deliver pieces of the product in iterations with minimal upfront architecture design and planning effort [23]-[27]. It was reported that waterfall requires ten times more effort than scrum, whereas the velocity of scrum is seven times faster than waterfall, and the customer satisfaction of scrum is significantly better than waterfall [28]. Agile itself has improved in diverse ways in the last two decades. The disciplined agile delivery (DAD) model has gained fame in the last few years. DAD is a people-first agile framework that is specifically generated by picking the best elements of other Agile models like XP, Scrum, and Kanban [29]. Disciplined Agile (DA) became so popular after 2012 that the PMI recently adopted it with four new different certification programs. The DA Toolkit supports continuous improvement and scalability while allowing team members to choose their way of working (WoW) [30].

A continuous process of learning and improvement is required to sustain competitive advantages and thrive in rapidly changing market conditions [31]. It is not an overnight process; continuous improvement, also popularly known as Kaizen, and the process of waste removal for value addition, a Lean approach, cannot be achieved immediately. It is a continually evolving process [31]. Traditional plan-driven and standard agile models still cannot comprehend the possibility of system evolution for a set of complex projects. It requires system thinking, which enables all three aspects: Kaizen, Lean, and Agility, like the leagile delivery model.

2.2 Evolving leagile project portfolio baselines

To incorporate lean strategies in agile projects, a new version of the project delivery model has emerged Lean-agile (leagile), as referred as LeAgile. In 1999, Naylor et al. [32] proposed the leagility philosophy for manufacturing production. Later, the leagile idea continued to evolve into many sectors like healthcare, professional services, and most importantly, into software development.[6], [32]-[38]. The leagile method applies lean management to reduce waste in the process and uses agile's iterative strategy to support agility and faster delivery. In this model, lean thinking contributes towards project process evolution, and agile focuses on agility and continuous delivery. As a result, portfolio and project management processes are also continuously improved in the leagile model.

To transform the complexity of modern projects, leagile requires continuous planning and efficient decision-making strategies. In general, existing agile and leagile approaches invest in minimal upfront architecture design and planning; project teams are expected to deliver faster on "not-all-known" scope in smaller packages [24], [27]. In his book [27], Cline argues against the agile teams' mindsets of "no-up-front-anything" and "learning upfront is a waste of time." He suggests that minimal necessary planning and learning are required to deliver a product as expected by business versus no planning at all. In the software development domain, where projects are managed in a dynamic and complex environment, current versions of agile and leagile models are incapable of continually planning for the immediate future [35], [36]. One of the reasons is that these models have not been extensively used in software development, and another is that the technology of software itself is advancing faster than the software development life cycle. These existing project delivery models cannot efficiently address the evolving baselines needed to seek accurate performance measurements for the continuous planning of large project portfolios.

The standard portfolio management is defined as the coordinated management of interrelated projects by which an organization evaluates, selects, prioritizes, and allocates its limited resources to accomplish the best organizational strategies [39]. One of the critical steps in this process is portfolio prioritization based on project baseline measurements, which is prone to extreme missteps because of the complexities involved in decision making during project selection and project task allocation[40]. The traditional plan-driven approach uses a fixed portfolio baseline, which is created during the planning phase and stays fixed until the end of the project [41]. By contrast, the leagile model has a dynamic baseline that evolves over time [41]. Figure 1 illustrates a portfolio with four plan-driven projects and two leagile projects. Plan-driven projects have straight lines, representing the fact that there is no change in the baselines. By contrast, leagile projects have dynamic baselines that constantly shift. In reality, the measurement of success in a complex and dynamic environment should follow evolving baselines rather than the fixed baselines of the plan-driven approach. Besides, a study by Fadaki et al. [19] found that if both leanness and agility equally embedded in system and continually evolved, then the higher performance is achievable.

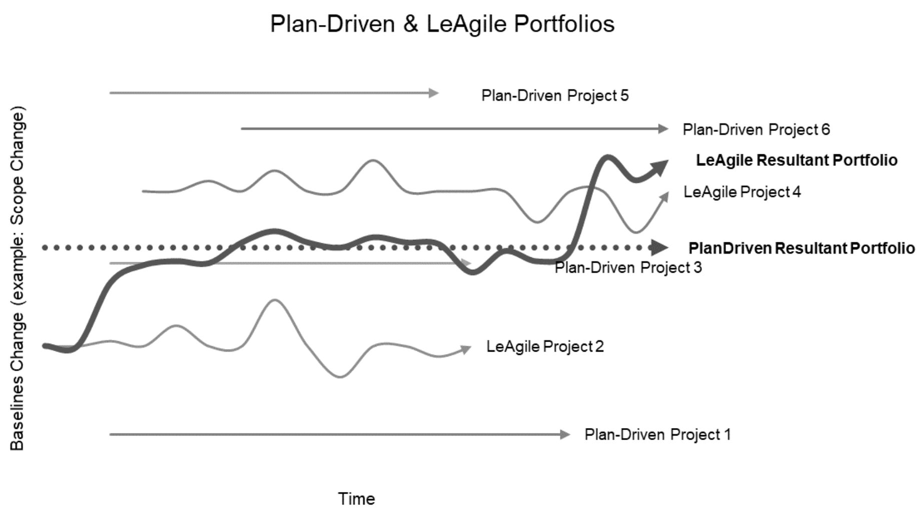


Fig. 1. Portfolio baselines for plan-driven projects and leagile projects

Similarly, the study [40] proposed an IT portfolio management process framework, which references the concept of continually self-organizing portfolios based on learning from the analysis, screening, continuous optimization, and adjustment of the portfolio to achieve evolution and success. In a rapidly changing environment, a portfolio becomes exceptionally complex. The plans and strategies will not work if they stay static throughout the life of the portfolio; instead, they should continually evolve with the experience gained from recent past events [40]. Continuous planning and improvement are crucial in keeping the portfolio alive (reduced risk) given modern complexity [37], [42].

In the IT project management context, according to Fitzgerald and Stol [43], the only forms of continuous planning used are sprint iteration planning, developed from the agile approach, and software release planning. Continuous planning has not yet become widespread throughout all organizations, especially in the context of software development [6]. In addition, a mindset to achieving consistent success has not been established. Only 2.5% of companies complete their projects successfully [44]. Consistently delivering successful projects is the key to the genuine success of a business [45]. Consistent success requires: i) direct “line-of-sight” feedback on project progress; and ii) incorporation of “learning from experience” for the continuous improvement of project management processes and practices [45], [34, p. 106-109].

In modern project management practice, it has become critical to establish a learning system that incorporates lessons from failures with immediate adaptation to sudden changes while maintaining the transparency of knowledge throughout multiple project teams to strategic portfolio leaders [45], [46], [47]. Furthermore, the recently published CHAOS Report [4] introduces a new definition of project success called “pure success.” Pure success is the successful delivery of high customer satisfaction and the generation of a high return on value to the organization [4]. Classic success is the completion of the project on-time and on-budget based on predefined baselines and quality. The report compared pure success with the classic definition of success and found drastic changes in the rates of reported success [4]. When the new definition of success is used, the project success rate decreased to 14% from 36%, and the challenged project rate increased from 45% to 67% [4]. This report reveals that the traditional approach of estimating the performance and baselines produces inconsistent and inaccurate results for modern projects. To achieve pure success, the management team needs to continuously learn from executed tasks and change their product requirements to adapt to changes in the project environment. Pure success requires lean process improvement and learning. Few recent studies used computer-assisted algorithms to establish learning in a project, like learning and feedback loop system [48], work package size optimization for value improvement [49], Bayesian approach for portfolio risk identification and reduction [42], [50], Bayesian approach for traditional waterfall-type earned value planning [51] and modeling uncertainty [16]. The existing studies for success of leagile project system mainly focused on the risk factors, continuous improvement factors, complexity aspects, pros and cons, definitions, acceptance of agile or lean, and causes of failures [6], [15], [18], [27], [37], [38], [48]. However, we found no study which provided a practical and convenient solution for engineering managers on the implementation of learning to reduce these challenges and complexities. This finding supports systematic literature review study by Stefan et al. [20], suggesting IT project complexity is increasing and there are no practical tools and models available yet for managers to achieve true project success.

This article argues that the increase of failure in a large complex project is not just because of the task performance; rather, it is because of the static scale used to measure the tasks. The scale should increase or decrease based on the recent experience of prior tasks. To address these challenges, this study supplies a simplistic learning tool to measure the performance of modern projects. Specifically, the objective of this study is to seek a more accurate estimation of project baselines against which iterative tasks can be measured in a dynamic environment based on continual learning from prior experience.

The study moreover aims to answer whether the evolving baseline provides a better performance measurement scale than the static baseline of the traditional plan-driven approach. A likelihood ratio test and Bayesian model is developed (next section) for the continuous estimation of evolving project baselines based on learning from recent past performance.

3. Methodology

This study is one of the first efforts to establish a practical performance measurement using the Bayesian continual learning approach for leagile portfolio management. This article focuses on the actual process improvement for a whole portfolio using the project-level tasks' experience. The proposed framework provides a simple formula to achieve learning and reduce uncertainty. This study follows the principle of Pasteur's quadrant from systems engineering (Figure 2) to both enhance project management knowledge and realize the immediate use of Bayesian continuous learning [52]. Furthermore, the likelihood ratio test is performed to compare the accuracy of the proposed model against a traditional model (refer to section comparison of approaches).

Pasteur's quadrant was named after Louis Pasteur, whose work exemplifies both advancements in knowledge on the subject matter and results with high social benefits by making them immediately available for use.

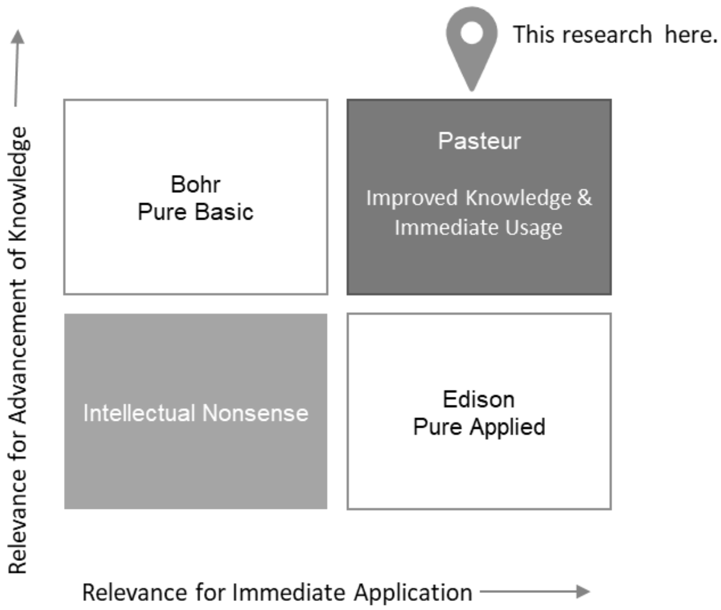


Fig. 2. Pasteur's quadrant

The static baseline approach in project management is an example of the Edison quadrant, which has high immediate usability but little improvement in knowledge, as presented by the bottom right block of Figure 2. Our proposed evolving baseline approach incorporates both the immediate applicability and improvement in knowledge located in the top right block of Figure 2. Specifically, Bayesian theory is used in our approach to estimate the evolving baseline by continually measuring the performance of executed tasks and predicting the confidence bounds of the baseline based on the newly learned posterior distributions. Figure 3 provides an overview of this study, which illustrates the proposed Bayesian evolving baseline approach, the traditional static baseline approach, and their comparisons to choose the model with the best performance.

This section is further divided into three subsections—the first subsection presents the details of the process flow and steps taken during analysis. The second subsection develops the proposed evolving baseline approach further by mathematically describing how the evolving baseline is generated from learning and Bayes rule. The third subsection presents a brief description of the traditional baseline approach used for comparison.

3.1 *Methodology flow steps*

In a traditional static baseline approach, the project team uses the historical lessons learned from past projects or make a rough order-of-magnitude estimation to establish baselines (e.g., mean, upper, and lower bounds of the probability of task failure) for future measurement. The baselines are often determined during the initiation and planning phases; they are then used throughout the entire life of the project.

For the traditional static baseline approach, as seen in the left section of Figure 3, the same POC baseline is used until the end of project life to measure performance. By contrast, in the proposed approach, the right section of Figure 3 continually updates its as soon as new learning occurs. In each measurement iteration, the count of failed tasks and total tasks from the completed bucket is grabbed and passed instantly to the Bayesian model. Measurement iteration in this article is defined as the cycle of measurements done for the completed tasks. It is not the same as the terms "iteration" or "sprint", which are used in adaptive models and agile scrum. A new event means a task or a set of similar jobs have been completed at a certain rate of success or failure when a measurement is collected.

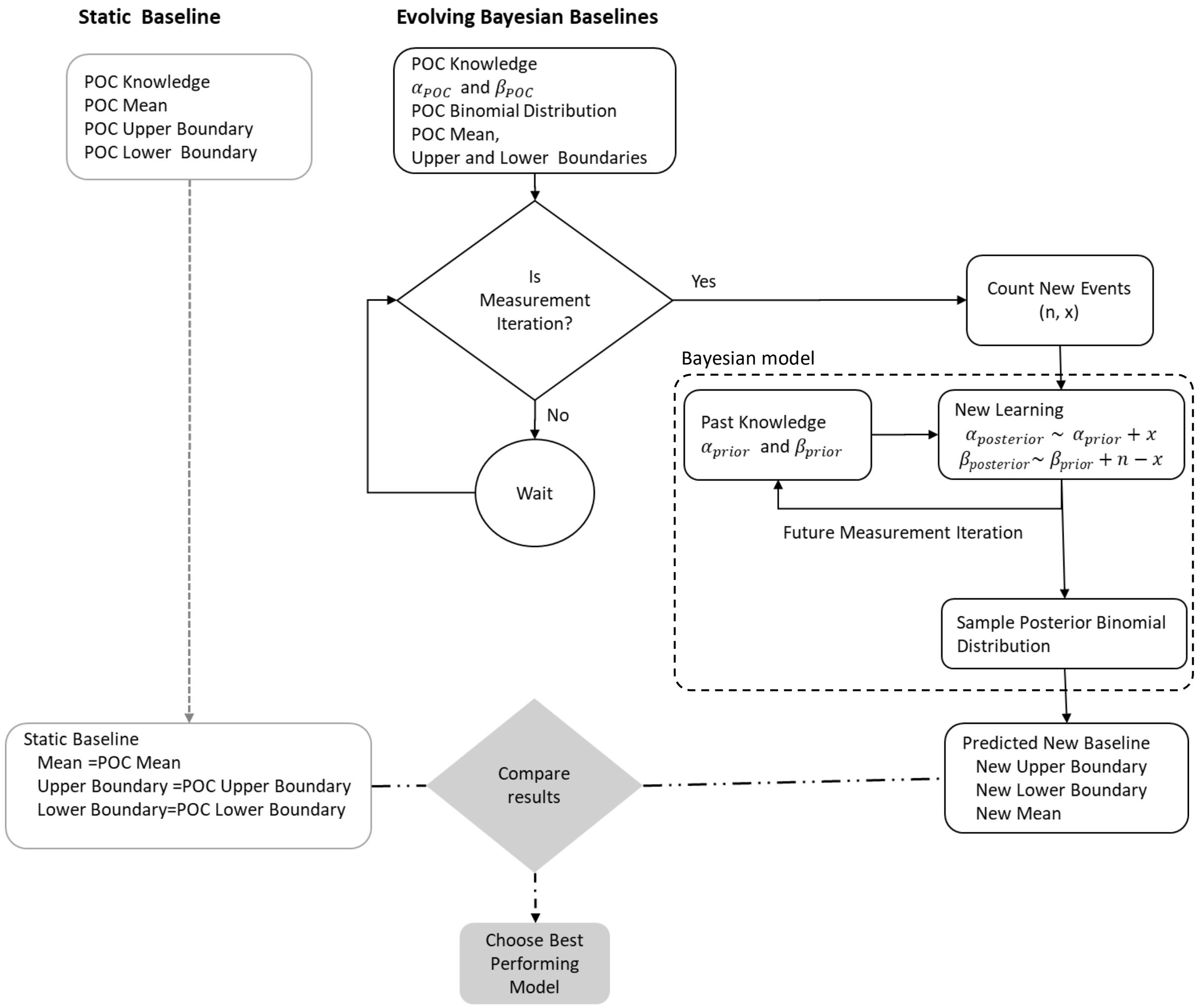


Fig. 3. Methodology: comparison of static and evolving baselines

In the last step of this study, we compare the traditional static baseline approach and the new Bayesian evolving baseline approach to identify the best performing model (refer to gray blocks in Figure 3 and section comparison of approaches). A baseline is often described by its mean and confidence bounds. The baselines generated by both approaches are compared with each other to evaluate their usability and accuracy. The model with the most realistic baseline is chosen as the best performing model.

3.2 *Proposed evolving baseline method*

The iterative nature of tasks and activities in a leagile type model creates the possibility of qualitative measurements of the smallest tasks or activities. Furthermore, quantifying task scope/deliverables depends on the approach to the work breakdown structure (WBS) [53]. It is practically impossible to implement continuous improvement without a quantifiable work package or task [53]. In project management, a "rule of thumb" for task estimation is the "80-hour" rule: it suggests decomposing the whole project scope until task size reaches 80 hours per deliverable. It helps in determining when to stop dividing deliverables into smaller elements. It is also followed in an agile scrum, where the standard sprint size is two weeks long. This study uses data with the "80-hour" rule to quantify the task as a failure or success (refer to the section on research data for details). This study uses success and failure probabilities to measure the performance of tasks and projects. A Bayesian model is used to derive the evolving baselines; the equations and computational steps are described in detail here.

As shown in Figure 3, the Bayesian model combines the lessons from the new events and past knowledge to continually predict the new posterior parameters, which provides an updated and more accurate estimation of the baseline parameters such as average success and/or failure probabilities as well as their upper and lower bounds. The posterior parameters also become prior parameters (past knowledge) for future measurement iterations. The mathematical details are described as follows.

Each task can either succeed or fail, which can be considered a Bernoulli trial. Therefore, the probability $P(x)$ of observing x failures in n tasks can be obtained from the binomial distribution as

$$P(x) = \binom{n}{x} p^x (1-p)^{n-x} \tag{1}$$

where p is the probability of failure per task. For complex projects/portfolios in a dynamic environment, the failure probability of each task may change as the projects develop. The failure rate may depend on shifts in market conditions, technological advancements, legal requirements, project environment, and resources. Therefore, it is crucial to continuously update the failure probability p based on learning from the immediate past. This can be achieved through the Bayesian learning algorithm described below.

In the Bayesian framework, priors $g(p)$ and likelihood $L(x|p)$ function are required to compute the posterior $g(p|x)$ as follows:

$$\text{Posterior } g(p|x) \sim \text{Likelihood } L(x|p) * \text{Prior } g(p) \tag{2}$$

where symbol "~" represents "directly proportional to" and the likelihood of observing x failure from n tasks can be calculated using the binomial distribution as

$$L(x|p) = \binom{n}{x} p^x (1-p)^{n-x}, \text{ where } x = 0,1,2, \ldots n \tag{3}$$

For binomial likelihood, a natural choice of the prior for failure probability p is the beta distribution [54], where the prior (beta distribution) probability density function (PDF)) $g(p)$ with shape parameters α, β > 0 is given as

$$g(p) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha,\beta)} \tag{4}$$

where

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \sim \frac{(\alpha-1)!(\beta-1)!}{(\alpha+\beta-1)!}$$

Further, using Equation (2), the posterior distribution of p can be derived as follows [54]:

$$g(p|x)\sim \frac{p^{\alpha+x-1}(1-p)^{\beta+n-x-1}}{B(\alpha+x,\beta+n-x)} \tag{5}$$

The posterior distribution of failure probability p also follows a beta distribution with parameters $\alpha_{posterior}$ and $\beta_{posterior}$

$$\alpha_{posterior} \sim \alpha_{prior} + x \tag{6}$$

$$\beta_{posterior} \sim \beta_{prior} + n - x. \tag{7}$$

where α_{prior} and β_{prior} are the prior parameters α and β in Equations (4)–(5). The posterior beta distribution can then be used to estimate the baseline measurement, i.e., the failure probability and confidence bounds. Specifically, the following formulas can be used to estimate the baseline parameters.

The mean of the posterior beta distribution (i.e., the mean failure probability) can be computed using [54, p. 530]:

$$m(p) = \frac{\alpha_{prior}+x}{\alpha_{prior}+ \beta_{prior}+n} \tag{8}$$

The credibility interval of the failure probability (p) at 90% credibility can be calculated using the following equations [54, p. 530]:

Lower Credibility Interval: LCI =$BETAINV(0.05,\ \ \alpha_{prior} + x\ ,\ \beta_{prior} + n - x\ )$ (9)

Upper Credibility Interval: UCI= $BETAINV(0.95,\ \ \alpha_{prior} + x\ ,\ \beta_{prior} + n - x\ )$ (10)

where *BETAINV* is the inverse of the beta distribution. The posterior parameters are passed to the next iteration as new priors to continuously update the beta distribution of failure probability for baseline estimation. The proposed model offers a continually evolving baseline based on newly learned information as compared to the static baseline approach where the baseline measurements stay constant throughout the project lifetime.

3.3 Traditional static baseline method

In the traditional static baseline approach, the binomial distribution (Equation (1)) is used to calculate the POC baseline. Similar to the Bayesian approach where a 90% credibility interval is used, for the traditional approach we also used a 90% confidence interval. The upper and lower bounds of failure probability (p) at the confidence level 90%, given x failures in the n total tasks, can be calculated using the beta distribution as [54].

Lower bound: BETAINV$(0.05, x, n - x + 1)$

Upper bound: BETAINV$(0.95, x + 1, n - x)$

The POC baseline is static throughout the life of the project.

3.4 Research data

We used real case data from the ABC Health Care company for our “SharePoint optimization (SO)” portfolio. “ABC” is not a real name as the company wishes to stay anonymous. The main goal of the SO effort was to optimize the usage of SharePoint by incorporating continual learning from the performance of each SO task. The SO effort was initiated because of a sudden increase in the chargeback of the SharePoint service, which increased from $67 per Gigabyte (GB) in 2016 to $85 per GB in 2020. The business case for this SO portfolio was to realize a direct benefit of $19.28 M within two years.

Furthermore, the SO effort focused on establishing a self-learning process to continually optimize the performance of all SharePoint accounts. Six weeks of data were gathered for the first “outreach” phase of the SO effort. It included

3,113 SharePoint accounts with at least two site control admins and multiple site business owners. The SO portfolio followed a continuous delivery model with leagile strategies for process optimization. All SharePoint tasks of projects continually moved from the “to-do” bucket to “in-progress” and then to the outreach “completed” bucket.

Each task was associated with each SharePoint account and was completed independently by different site control admins and site business owners from a different department. Each task contained 17 questions to gather analytical data regarding the effective usage of the SharePoint account. The site control admins and site business owners had to run the few reports from their SharePoint dashboard to complete the task. The completed bucket contained all the project tasks completed successfully, and the failed tasks stayed in the in-progress bucket until they were fixed. We counted the task as failed if the task exceeded the due date. The due date for each task was set to two weeks after generation. Successful tasks were color-coded green. The failed and challenged tasks were grouped together and marked red. The overall portfolio status was measured every two weeks and reported in strategic leadership meetings. A breakdown of the project tasks for each measurement iteration is summarized in Table 1. A measurement iteration in this study is defined as a status-reporting cycle of the whole portfolio, a two-week cycle.

Table 1. SO outreach data

Measurement iterations	SO projects	Challenged (red)	Succeeded (green)
Iteration 1	59	17	42
Iteration 2	303	66	237
Iteration 3	267	22	245

The POC for process improvement and optimization was used before the start of the SO portfolio. Forty early adopters, who wanted to move to optimization as soon as possible, were engaged in the POC effort, which generated ten failed project tasks out of the 40 POC tasks, and this failure rate was used as the starting baseline for the whole project portfolio.

4. Results and analysis

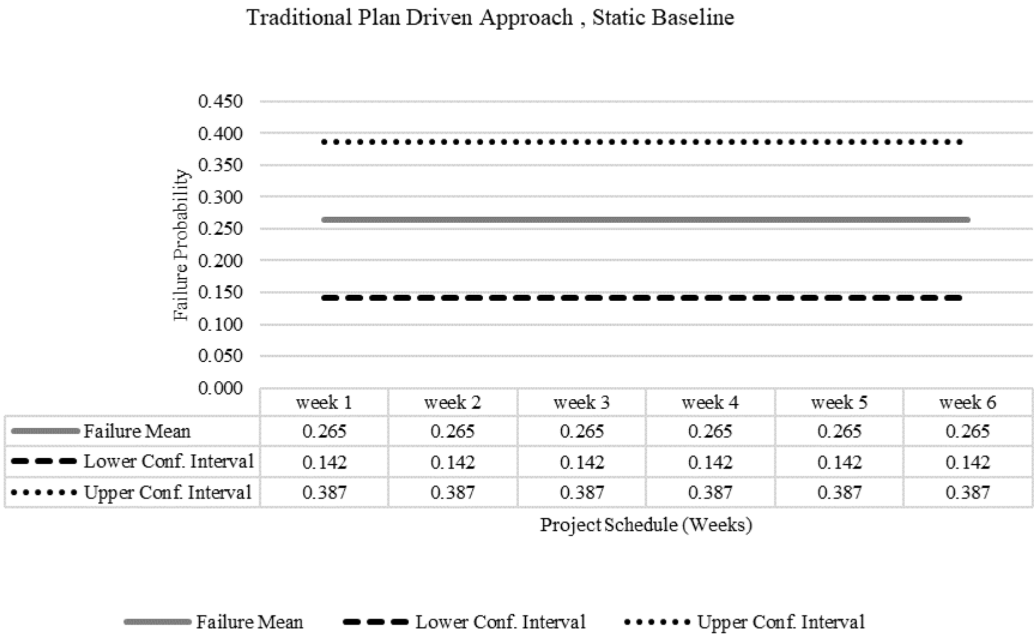
4.1 Results for the traditional static baseline

The traditional plan-driven approach uses a historical point of reference to estimate all the baselines during the inception of the project. The baseline stays fixed and is the only baseline used to measure the performance of future tasks for all measurement iterations. Baseline estimates in the traditional approach is given in Table 2 and Figure 4, where the point estimation and the confidence interval of the point estimation are calculated respectively and stay the same over several iterations.

The point estimation of the failure probability of 0.25 is obtained, given that 10 out of 40 tasks failed in the project portfolio. As explained in Section 3 (Equation (5)), POC effort predicts that the estimated failure probability will fall within the lower confidence interval of 0.142 to the upper confidence interval of 0.387 at a 90% confidence level. The mean, lower, and upper bounds are presented in Figure 4 by solid, dashed, and dotted lines, respectively.

Table 2. Traditional static baseline results

Binomial distribution parameters	Historical knowledge	Iteration 1	Iteration 2	Iteration 3
Point Estimate (Mean)	0.25	0.25	0.25	0.25
Lower Conf. Interval	0.142	0.142	0.142	0.142
Upper Conf. Interval	0.387	0.387	0.387	0.387



	week 1	week 2	week 3	week 4	week 5	week 6
Failure Mean	0.265	0.265	0.265	0.265	0.265	0.265
Lower Conf. Interval	0.142	0.142	0.142	0.142	0.142	0.142
Upper Conf. Interval	0.387	0.387	0.387	0.387	0.387	0.387

Fig. 4. Static baseline based on the POC effort

In a plan-driven approach, significant efforts are invested in controlling the baselines of project plans [22]. Changes in such models must usually go through a strict change control process, which is not efficient in a dynamic leagile environment. By contrast, enterprise leagile projects and portfolios continue to adapt to the changes in requirements and the environment. For the leagile model, it is critical to continually update the baseline and measure the success and failure adaptively as the projects and portfolios progress. In the next section, we illustrate the proposed continual learning strategy to dynamically update the baseline after each iteration as new failure data become available.

4.2 Results for the proposed evolving Bayesian baseline

In the previous section, the POC identified the prior failure probability of a portfolio, i.e., on average, 10 out of 40 SO outreach tasks failed. This information was used in the Bayesian learning approach to update the posterior distribution of failure rate at each iteration. The posterior produces a new baseline, which can be used to measure the performance of future tasks.

It is assumed that the initial failure probability from POC data (previous section) follows a beta distribution with parameters $\alpha_{prior} = 10$ and $\beta_{prior} = 30$ before iteration 1 of Weeks 1 and 2. After iteration 1, failure data were collected (see Table 1), where 17 failures were observed out of a total of 59 SO targets. Following the equations given in Section 3, the posterior distribution of the failure probability p can be obtained as a beta distribution with the shape and scale parameters calculated as follows:

$$\alpha_{posterior} \sim \alpha_{prior} + x = 27$$

$$\beta_{posterior} \sim \beta_{prior} + n - x = 72$$

Here, $\alpha_{posterior}$ is α_{prior} increased by the number of observed failures x and $\beta_{posterior}$ is β_{prior} increased by the number of successes $(n - x)$, as shown in Equations (6) and (7).

Given the parameters of the posterior distribution of p, the average failure probability can be calculated using Equation (8) as follows:

$$m(p) = \frac{\alpha_{prior} + x}{\alpha_{prior} + \beta_{prior} + n} = 0.273$$

Accordingly, the LCI and UCI at 90% confidence level are

$$\textit{Lower Credibility Interval}_{\ alpha=0.05} = 0.202$$

$$\textit{Upper Credibility Interval}_{\ alpha=0.95} = 0.349$$

This procedure is repeated for multiple measurement iterations to update the baselines. As shown in Table 3, for each iteration, the posterior is updated, generating new Bayesian baselines for future tasks.

Table 3. Predicted Bayesian posterior and beta parameter results

Parameters	**Prior**	**Posteriors**		
		Iteration 1 [a]	**Iteration 2** [b]	**Iteration 3** [c]
α	10	27	93	115
β	30	72	309	554
$m(p)$	0.25	0.273	0.231	0.172
LCI		0.202	0.198	0.148
UCI		0.349	0.267	0.196

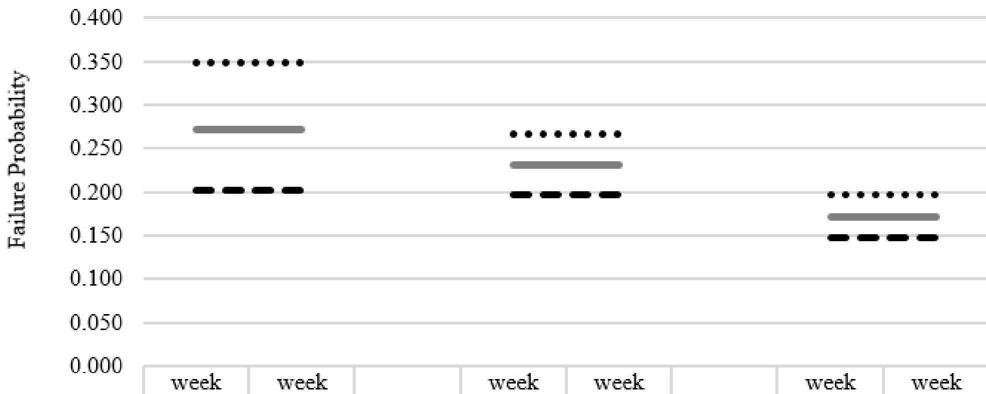
[a] Weeks 1 and 2, where failed x=17, total n= 59
[b] Weeks 3 and 4, failed x= 66, total n=303
[c] Weeks 5 and 6, failed x=22, total n=267

Figure 5 shows the evolution of the baseline based on the information learned from each iteration (every two weeks). The lesson from Weeks 1 and 2 suggests an average failure probability of 0.273 with an LCI of 0.202 and a UCI of 0.349. The estimated credibility interval from Weeks 1 and 2 will be used as the new baseline to measure the performance of Weeks 3 and 4. During Weeks 3 and 4, more tasks were assigned, and a few failures occurred; the mean reduced to 0.231 with a credibility interval of (0.198, 0.267) at a 90% confidence level. The failure probability for Weeks 3 and 4, shown by the middle three lines in Figure 5, stayed below the upper bound of credibility interval predicted by Weeks 1 and 2. This means that Weeks 3 and 4 performed better than Weeks 1 and 2. Moreover, the gap between the UCI and LCI of Weeks 3 and 4 is smaller than that of Weeks 1 and 2, which is an indication of the improvement in task performance during Weeks 3 and 4.

Similarly, information learned from Weeks 3 and 4 creates a new baseline for Weeks 5 and 6. The performance of the tasks for Weeks 5 and 6 is evaluated against the baseline from Weeks 3 and 4, as shown in Table 3. The mean failure probability from Weeks 5 and 6 is estimated as 0.172, which is also a sign of improvement in the performance during these Weeks when compared with the means and Credibility intervals of Weeks 3 and 4 and Weeks 1 and 2. Furthermore, the gap between the UCI and LCI has been reduced significantly in Weeks 5 and 6 when compared to those of prior iterations.

When looking at the whole iteration sets, as presented in Figure 5, the mean failure probability continued to decrease, nearing 17% in the last iteration. The failure probability decreased continually, and the performance of the task increased iteration by iteration. Similarly, the width of the credibility intervals gap reduced with each new iteration.

This confirms that the variations in task failure probability are decreasing, and that the task performance is becoming more consistent.



	week 1	week 2		week 3	week 4		week 5	week 6
Failure Mean	0.273	0.273		0.231	0.231		0.172	0.172
Lower Credibility Interval	0.202	0.202		0.198	0.198		0.148	0.148
Upper Credibility Interval	0.349	0.349		0.267	0.267		0.196	0.196

Project Schedule (Weeks)

Failure Mean   Lower Credibility Interval   Upper Credibility Interval

Fig. 5: Evolving baselines using the Bayesian learning approach

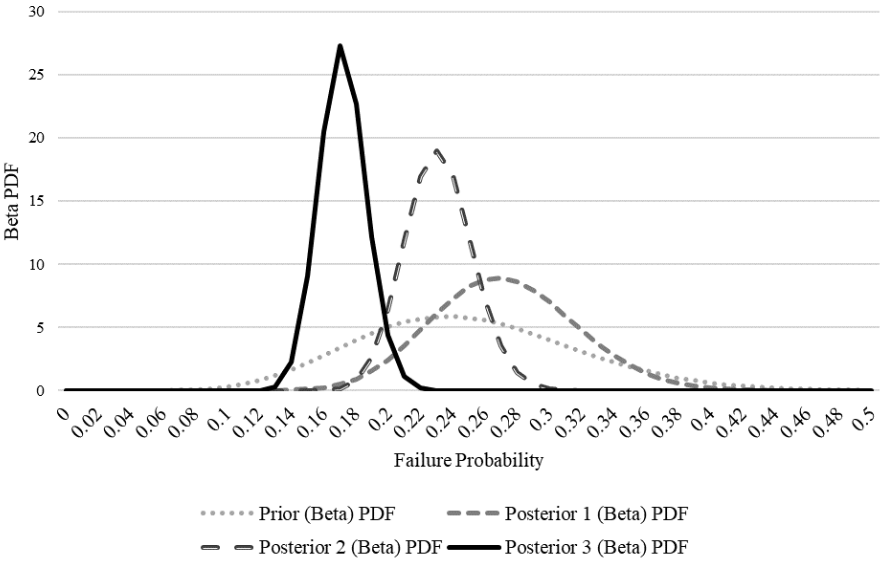


Fig. 6. Predicted PDF of the posterior distributions for each iteration result

Moreover, as the number of iterations increase, the PDF of the posterior distribution of the failure probability p moves left and its tails PDF become thinner (Figure 6). The posterior distribution of iteration 3 has a peak centered at 0.17 with thinner tails than the posterior distributions of iterations 1 and 2. This again shows continuous growth towards a lower rate of failure and tighter confidence bounds. In simpler terms, iteration 3 predicts that the failure probability of iteration 4 will stay within 0.148 to 0.196 at a 90% confidence level. If the failure rate in the fourth future iteration goes above 0.196, then the project portfolio is considered to be challenged, in contrast to the traditional approach, where the portfolio would not be considered challenged until the failure rate reaches 0.387.

The evolving baseline of the Bayesian approach showed a decrease in the posterior mean and a decrease in the spread between the upper and lower limits. This stands for the fact that with each iteration, the performance improves. That is, the failure rate (p) decreasing as effort count (n) increases—a genuine intention of the leagile delivery model [54].

4.3 Comparison of approaches

The traditional plan-driven approach identifies a baseline during the start of the project, and the baseline stays static throughout all iterations (Table 2 and Figure 4). By contrast, the evolving baseline approach continues to predict new baselines for future measurement iterations. As an example, the experience of the second measurement iteration predicts the new baseline for the third iteration. The failure probability of the task for the third iteration is predicted to be within 0.198 to 0.267 at a 90% confidence level. The task portfolio is considered to be challenged if the rate of actual task failure exceeds 0.267 in the third measurement iteration, versus the traditional approach where the task will not fail until the rate exceeds 0.387. As a result, the baselines evolved using the proposed Bayesian model are more accurate and realistic than those of the traditional approach.

A likelihood-ratio test (LRT) [55, p. 511] was conducted to find a better model of evolving project baselines. During LRT, we compared the llikelihood values of the traditional model against the proposed Bayesian model. The null hypothesis is defined as the performance of the Bayesian model is the same as the traditional model, and the alternative hypothesis is Bayesian model has better performance. The likelihood-ratio test statistic (LRT statistic) is calculated as $-2\ log_e(\frac{L_{traditional}}{L_{Baysian}})$, where $L_{traditional}$ is the likelihood values of the traditional model and $L_{Baysian}$ is the likelihood value of the Bayesian model. The LRT statistic is 5.919. This provides a significantly small p-value, 0.015. Reject the null hypothesis at $\alpha = 0.05$. The LRT test supports the fact that the Bayesian approach is a better model than the traditional model.

The Bayesian approach provides a more accurate measurement of project and portfolio performance than the plan-driven method. The Bayesian approach responds quickly to changing project variables that can positively or negatively impact project performance. These variables can be changes in the team environment, market, resources, law/regulations, technology, weather, or the recent coronavirus impact. The confidence bounds of the evolving baseline can increase or decrease and move up or down based on learning from the immediate past, unlike the static baseline of the traditional approach, where the confidence bounds stay the same throughout the project lifetime. Continuous forecasting is much easier if managers can immediately get a new predicted baseline for future iterations.

Our proposed approach recommends the maintenance of only two parameters (α, β) to estimate evolving baselines continually. Managing only two parameters simplifies the "applicability" of the proposed approach. The computation required to calculate the updated baseline is straightforward; anyone with Excel can use the built-in BETAINV function to obtain the posterior distribution, mean failure probability, and upper/lower confidence limits for new baselines.

5. Conclusions

It is evident in the project management world today that most organizations have moved towards agility and lean delivery models. Nevertheless, the leaders of project management offices and project managers are still trying to catch up with this trend. This transformation is rapid, and limited resources and tools are available to aid continuous planning and decision making. This article provided an applied framework (a Bayesian evolving baseline approach) for modern leagile projects. The analysis demonstrated the advantages of the proposed approach over the traditional static baseline approach using SO portfolio data. The LRT findings of this study suggest that the evolving Bayesian baseline is a more accurate and realistic scale for measuring the success or failure of a leagile project and portfolio than the traditional static baseline. The result suggests that the continuous evolution of baselines based on learning can better estimate task performance for future planning. The proposed model can be easily integrated into any existing leagile project for continuous decision making. Furthermore, it is applicable to any type of project delivery model as long as the tasks of the project can be measured in terms of success or failure; they are independent and very similar in nature.

5.1 Discussion

Most complex enterprise projects are challenged more now than they were in the past few decades. The use of the outdated static baseline models to measure leagile project progress could be one of the reasons for the increase in project failures. The static baseline of the traditional plan-driven model does not apply to all types of contemporary projects and portfolios, especially when there is a constant change in the project scope, budget, resources, and environment. It is a known fact that a static baseline does not account for the recent changes in the project environment. This study showed that the performance measurement of a static baseline produces suboptimal results for modern leagile projects, as continuous learning and improvement are not considered in the traditional approach.

This article recommends the use of the Bayesian learning approach to estimate a continually evolving baseline and then use the learned baseline to measure success and reduce complexity. Our analysis found that the proposed evolving baseline provides more accurate performance predictions for the future effort of leagile projects/portfolios than the traditional static baseline. The evolving Bayesian baseline can closely capture the nature of project and portfolio progress despite the ever-changing project variables and environmental factors. The Bayesian learning-based evolving baseline approach can achieve both continuous learning and continuous planning in a joint framework for any leagile project portfolio.

5.2 Implications of the study

Learning from recent events has become a crucial element in complex projects with the unknown project scope. Projects that follow the leagile model for continuous delivery can benefit from the proposed strategy. This study developed a continual learning approach to estimate evolving baselines in a complex and dynamic project environment and proved that constant improvement is achievable through iterative learning. Evolving baselines generated from the continuously updated posterior predictions can incorporate “lines of sight” and “feedback loops” for a whole portfolio of leagile project systems.

This article is not limited to the data (SO optimization tasks) and the leagile model we used for our research. The mathematical solution provided by this study can be used in all types of projects and their portfolio as long as they maintain measurable task performance metrics like any simple work order to a complex project system. It can be implemented practically in any project as long as the work packages or tasks are iterative, measurable, and independent. It can benefit project and portfolio models such as DevOps, microservices, and leagile, which require continuous planning, continuous improvement, and continuous delivery. Furthermore, this study opens a new avenue for machine learning and artificial intelligence technologies to be applied in the software project management field to optimize existing project management processes and performance measurement standards.

In contrast to the static nature of the traditional approach, continual learning from recent experiences of proposed approach provides more accurate and reliable estimates of project and portfolio baselines. The continual learning from recent experiences is more recent and closely trails the changes in the project environment, thus reduces uncertainty. The justification for integrating Bayesian theory into project delivery models is that the Bayesian approach allows all possible subjective and objective input variables to be incorporated while producing quantifiable results. The outputs of the Bayesian model are measurable posterior metrics that are generated using continuously updated inputs due to changes in environments, changes in project structures, and even unknown priors. The prediction becomes more accurate as it matures with new learning. The results are impactful, especially when the project environment and scope are dynamic, and the baselines continue to change. Hence, the major implications of the study are the following:

- The study provides a straightforward and accurate tool for forecasting the performance of leagile projects and portfolios;
- The study uses the binomial distribution, which is widely used in project management to measure task performance and status;
- The evolving baseline approach is easy to use, and users with minimal statistical knowledge can implement it in leagile projects or portfolios;
- The proposed tool can contribute to informing decision making and planning. For example, it will empower managers and leaders to obtain reliable estimations of the performance of in-progress tasks/teams/projects and accurately plan upcoming projects in the portfolio pipeline.

5.3 Limitations and further research

This study was limited to leagile-type projects and portfolios. It used the binomial distribution to ensure the straightforward applicability of the evolving baselines in leagile project and portfolio. The binomial distribution can easily incorporate the most popular approach of task status reporting (task failure or success) to model task performance and predict future events. However, other models like the exponential or proportional hazards models could be used to describe failure mechanisms concerning project time, budget, and cost. Additional reliability models and measurements, such as survival models, hazard functions, and reliabilities, were not fully explored in this article. Future studies could incorporate such reliability models to predict overall project portfolio system reliability. A comparison study can be done to identify the most accurate model with reliable performance estimates.

As a final remark for future works, it is important to note that the task experience and learned performance estimates used in the article are highly quantitative. They must be quantifiable enough to be used easily in the proposed solution in order to make exceptionally reliable decisions. Future work may attempt to use a qualitative learning approach or deep machine learning approach in a hugely dynamic project to identify if evolving baselines perform better than static baselines.

References

[1] A. B. M. Moniruzzaman and D. S. A. Hossain, "Comparative Study on Agile software development methodologies," Global Journal of Computer Science and Technology, vol. 13, no. 7, pp. 25, 2013.

[2] Project Management Institute (PMI). (2013). *Talent Gap: Project Management through 2020* [Online]. Available: pmi.org

[3] M. Bloch, S. Blumberg, and J. Laartz, "Delivering large-scale IT projects on time, on budget, and on value," *Harvard Business Review* (2012), pp. 2–7, 2012.

[4] The Standish Group, *CHAOS Report: Decision Latency Theory: It Is All About the Interval,* The Standish Group International Inc., 2018.

[5] Project Management Institute. (2018). *Pulse of the profession 2018: Success in disruptive times* [Online]. Available: pmi.org.

[6] B. N. D. França, R. V. Simões, V. Silva and G. H. Travassos, "Escaping from the time box towards continuous planning: an industrial experience," *IEEE/ACM 3rd International Workshop on Rapid Continuous Software Engineering (RCoSE),* pp. 43–49, 2017.

[7] KPMG, *Project Management Survey Report 2013 Strategies to Capture Business Value,* Athens, Greece: KPMG, 2013.

[8] Project Management Institute (PMI). (2015). *Pulse of the profession 2015: Capturing the Value of Project Management* [Online]. Available: pmi.org.

[9] The Standish Group, *Manifesto, CHAOS Report. Think Big, Act Small*, The Standish Group International Inc., 2013.

[10] The Standish Group, *CHAOS Report, Project Smart*, The Standish Group International Inc., 2014.

[11] Geneca, *Doomed from the Start? Why a Majority of Business and IT Teams Anticipate Their Software Development Projects Will Fail,* Oak Brook, USA: Geneca, 2011.

[12] I. Carmen and B. Sussy, "IT projects success factors: a literature review", *International Journal of Information Systems and Project Management*, vol. 8, no. 2, pp. 49-78, 2020, doi: 10.12821/ijispm080203.

[13] B. Muamer, B. Sabina and G. Manfred, "The pivotal factors of IT projects' success – Insights for the case of organizations from the Federation of Bosnia and Herzegovina", *International Journal of Information Systems and Project Management*, vol. 8, no. 1, pp. 23-41, 2020, doi: 10.12821/ijispm080102.

[14] Ö. Meltem, G. Furkan and Ç. Burç, "An empirical analysis on the effects of investment assessment methods on IS/IT project success", *International Journal of Information Systems and Project Management*, vol. 7, no. 4, pp. 33-52, 2019, doi: 10.12821/ijispm070402.

[15] B. Shahzad, K. M. Awan, M. I. Lali and W. Aslam, "Identification of Patterns in Failure of Software Projects," *Journal of Information Science and Engineering*, vol. 33, no. 6, pp. 1465-1480, 2017.

[16] I. Dikmen, A. Qazi, H. Erol and M. Birgonul, "Meta-Modeling of Complexity-Uncertainty-Performance Triad in Construction Projects", *Engineering Management Journal*, pp. 1-15, 2020, doi: 10.1080/10429247.2020.1772698.

[17] W. XIA and G. LEE, "Complexity of Information Systems Development Projects: Conceptualization and Measurement Development", *Journal of Management Information Systems*, vol. 22, no. 1, pp. 45-83, 2005, doi: 10.1080/07421222.2003.11045831

[18] W. Hong, J. Thong, L. Chasalow and G. Dhillon, "User Acceptance of Agile Information Systems: A Model and Empirical Test", *Journal of Management Information Systems*, vol. 28, no. 1, pp. 235-272, 2011, doi: 10.2753/mis0742-1222280108.

[19] M. Fadaki, S. Rahman and C. Chan, "Leagile supply chain: design drivers and business performance implications", *International Journal of Production Research*, vol. 58, no. 18, pp. 5601-5623, 2019, doi: 10.1080/00207543.2019.1693660.

[20] M. Stefan, P. Liliane and K. Rob, "Definitions, characteristics and measures of IT project complexity - a systematic literature review", *International Journal of Information Systems and Project Management*, vol. 8, no. 2, pp. 5-21, 2020, doi: 10.12821/ijispm080201.

[21] J. Pinto and S. Mantel, "The causes of project failure," *IEEE Transactions on Engineering Management*, vol. 37, no. 4, pp. 269–276, 1990.

[22] J. Pinto and P. Morris, *The Wiley guide to managing projects*. Hoboken, N.J: John Wiley & Sons, 2010.

[23] K. Schwaber, "SCRUM Development Process," In *Business Object Design and Implementation*, London, UK: Springer, pp. 117–134., 1997.

[24] K. Schwaber and M. Beedle, *Agile Software Development with Scrum* (Vol. 1), Upper Saddle River, USA: Prentice Hall, 2002.

[25] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, 2004.

[26] K. Schwaber and J. Sutherland. (2017). *The Scrum Guide* [Online]. Available: scrum.org.

[27] A. Cline, *Agile Development in the Real World*, Berkeley, USA: Apress, 2015.

[28] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Addison-Wesley, 2012.

[29] M. L. Scott W. Ambler, *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*, IBM Press, 2012.

[30] S. Ambler, M. Lines, *Choose Your WoW: A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working*, Project Management Institute, 2020.

[31] O. a. M. J. Al-Baik, "Kaizen cookbook: The Success Recipe for Continuous Learning and Improvements.", in 2016 49th Hawaii International Conference on System Sciences (HICSS), Koloa, HI, 2016.

[32] J. B. Naylor, M. M. Naim, and D. Berry, "Leagility: Integrating the lean and agile manufacturing paradigms in the total supply chain," *International Journal of Production Economics*, vol. 62, no. 1, pp. 107–118, 1999.

[33] V. Mishra, C. Samuel and S. Sharma, "Lean, agile and leagile healthcare management – A case of chronic care", *International Journal of Healthcare Management*, vol. 12, no. 4, pp. 314-321, 2018, doi: 10.1080/20479700.2018.1428520.

[34] F. Rahimnia and M. Moghadasian, "Supply chain leagility in professional services: how to apply decoupling point concept in healthcare delivery system", *Supply Chain Management: An International Journal*, vol. 15, no. 1, pp. 80-91, 2010, doi: 10.1108/13598541011018148.

[35] G. Kasims, "Applying Lean to Improve Quality in Software Development Projects", *Proceedings of the 2nd International Conference on Business and Information Management*, pp. 130-134, 2018, doi: 10.1145/3278252.3278254.

[36] P. Rodríguez, J. Markkula, M. Oivo and J. Garbajosa, "Analyzing the Drivers of the Combination of Lean and Agile in Software Development Companies", in *Product-Focused Software Process Improvement*, O. Dieste, A. Jedlitschka and N. Juristo, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 145-159, 2012, doi: 10.1007/978-3-642-31063-8_12.

[37] X. Wang, K. Conboy and O. Cawley, ""Leagile" software development: An experience report analysis of the application of lean approaches in agile software development", *Journal of Systems and Software*, vol. 85, no. 6, pp. 1287-1299, 2012, doi: 10.1016/j.jss.2012.01.061.

[38] T. Suomalainen, R. Kuusela and M. Tihinen, "Continuous planning: an important aspect of agile and lean development," *International Journal of Agile Systems and Management*, vol. 8, no. 2, pp. 132–162, 2015.

[39] Project Management Institute (PMI), A guide to the Project Management Body of Knowledge (PMBOK guide), 6th ed. Newton Square, PA: Project Management Institute, 2017.

[40] N. Archer and F. Ghasemzadeh, "An integrated framework for project portfolio selection," *International Journal of Project Management*, vol. 17, no. 4, p. 207–216, 1999.

[41] R. Sweetman and K. Conboy, "Portfolios of agile projects: A complex adaptive systems' agent perspective," *Project Management Journal*, p. 18–38, 2018.

[42] D. Guan, K. W. Hipel, L. Fang and P. Guo, "Assessing project portfolio risk based on Bayesian network", *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1546-1551, 2014, doi: 10.1109/SMC.2014.6974136

[43] Fitzgerald and K.-J. Stol, "Continuous software engineering and beyond: Trends and challenges," in *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, 2014.

[44] PWC, *Project Management: Improving Performance, Reducing Risk*, 2014 [Online] Available: www.pwc.com/jg.

[45] T. Cooke-Davies, “The ‘real’ success factors on projects,” *International Journal of Project Management*, vol. 20, no. 3, pp. 185–190, 2002.

[46] Sommer, “Project failure—12 mistakes to avoid,” in *PMI Global Congress North America*, Atlanta, GA., 2004.

[47] R. Discenza and J. B. Forman, “Seven causes of project failure: How to recognize them and how to initiate project recovery,” in *PMI Global Congress North America*, Atlanta, GA, USA, 2007.

[48] O. Krancher, P. Luther and M. Jost, “Key Affordances of Platform-as-a-Service: Self-Organization and Continuous Feedback”, *Journal of Management Information Systems*, vol. 35, no. 3, pp. 776-812, 2018, doi: 10.1080/07421222.2018.1481636.

[49] C. Li and N. Hall, “Work Package Sizing and Project Performance”, *Operations Research*, vol. 67, no. 1, pp. 123-142, 2019, doi: 10.1287/opre.2018.1767.

[50] F. Ghasemi, M. Sari, V. Yousefi, R. Falsafi and J. Tamošaitienė, “Project Portfolio Risk Identification and Analysis, Considering Project Risk Interactions and Using Bayesian Networks”, *Sustainability*, vol. 10, no. 5, pp. 1609, 2018, doi: 10.3390/su10051609.

[51] F. Caron, F. Ruggeri and A. Merli, “A Bayesian Approach to Improve Estimate at Completion in Earned Value Management”, *Project Management Journal*, vol. 44, no. 1, pp. 3-16, 2013, doi : 10.1002/pmj.21303.

[52] E. Stokes, *Pasteur’s Quadrant: Basic Science and Technological Innovation*, Brookings Institution Press, 1997.

[53] K. Ghane, “A model and system for applying Lean Six sigma to agile software development using hybrid simulation”, *2014 IEEE International Technology Management Conference*, 2014, doi: 10.1109/itmc.2014.6918594

[54] P. A. Tobias and D. C. Trindade, *Applied Reliability*, Boca Raton, USA: CRC Press Taylor & Francis Group, 2012.

[55] K. Aho, *Foundational and applied statistics for biologists using R*. Boca Raton, FL: CRC Press, 2014.

Biographical notes

Sagar Chhetri
Sagar Chhetri is a Systems Excellence Scientist, Researcher and ML Project/Portfolio Management professional. He has been working as a project management professional and chief scientist for the past 15 years. He is an MBA holder and Ph.D. candidate in the Systems and Engineering Management (SYEM) program at Texas Tech University. He holds PMP certification from PMI and Data Scientist certification from Microsoft (edEx). His research interests include enterprise project system improvement using learning, computation, reliability, machine learning, and LeAgile Portfolio.

Dongping Du
Dr. Du is an Assistant Professor in the Department of Industrial, Manufacturing and Systems Engineering at Texas Tech University. She received her BS in Electrical Engineering from the China University of Mining and Technology and her MS and PhD in Industrial Engineering from University of South Florida. Her research interests are data mining and machine learning, modeling and simulation, reliability analysis and risk assessment. She is a member of Institute of Industrial and Systems Engineers (IISE), Institute for Operations Research and the Management Sciences (INFORMS), and Institute of Electrical and Electronics Engineers (IEEE).