

SCOPE CONFIRMATION EXERCISE (SCE): A PRE-PROJECT EXERCISE TO ENSURE A SUCCESSFUL CAPITAL MARKET FINTECH PROJECT

Sumit Kumar, Indian Institute of Management Kozhikode

ABSTRACT

Cost and time overruns is persistent problem in software projects. The chaos report (1994) from the Standish group reported the extent of time and budget overruns in a software project delivery, which was quite intriguing. They had reported an average of 178% of the cost overruns for large companies, 182% for the medium, and 214% for the small companies. Time overruns averages are 230%, 202%, and 239%, respectively. Similar findings were reported by McKinsey & Company in 2012 even after 18 years. McKinsey Company said that around 66% of the software projects suffer from cost overrun, 33% from Schedule overrun, and 17% get delivered with benefits shortfall. This is a significant project risk. While there are multiple methods proposed to mitigate this project risk, there is no evaluated project risk management methodology available in the literature. We offered a Pre-Project Scope Confirmation Exercise (SCE) as an activity, if done correctly that can lead to successful project delivery. We proposed a framework of SCE and then evaluated the SCE approach using 20 capital market financial technology projects done for the last 11 years, between 2010 and 2021, and suggested a newly proposed TUB score (Project success score). We have found that more time spent on one SCE drives to a better TUB Score. Essentially a detailed SCE ensures the success of the project. The analysis is done on the primary data collected from those 20 projects.

Keywords: Scope management; Fintech; SCE (Scope confirmation exercise); Agile; Project Score.

INTRODUCTION

On-time and on-budget delivery of a Software project are essential success factors. There are several shreds of evidence of a project being a failure in terms of timeline and budget or both. Only 16.2 % of the software projects get completed in time and budget as per the Standish group (Chaos report, 1994). As per the Chaos Report (1994), in 1995, companies and government agencies in the U.S. had spent \$81 billion on canceled software projects. There were around \$59 billion paid additionally compared to the initial budget to complete the project with massively exceeded the original time allocated. Costs and Time overruns are the two essential parameters on which projects were assessed. The average cost overruns were 178% for the large companies, 182% for the medium companies, and 214% for the small companies. Bloch et al. (2012) also reported a similar state of affairs about Software projects even after 18 years. Nonaka and Babar (2007) concluded that the cost of a software product development overruns because of requirements volatility and poor quality of the condition. Aljohani et al. (2017) identified primarily seven factors of the cost overruns in the software projects; these are frequent design change, contractors' financing, payment delay for completed work, lack of contractor experience,

poor cost estimation, poor tendering documentation, and poor material management. McQuighan & Hammell (2011) highlighted the scope of a project as a leading indicator for managing a software development project. They showed in their analysis the importance of scope determination and Management as a factor leading to the success or failure of the software projects. Damasiotis et al. (2014) stressed that knowing the scope of the project beforehand and knowing it well enough is essential in any successful delivery of the project. They suggested that the complexity of the scope of a software project should be studied by using a proper project management technique. They applied the Project Management body of knowledge (PMBOK) as a framework to analyze the project's complexity. Mirza et al. (2013) alluded to a similar finding of scope management. They agreed that a well-defined and managed scope leads to delivering a quality product in an agreed time and budget. Van Genuchten (1991) investigated the delay in software projects by analyzing 160 activities comprising 15000 hours of work and found that a proper work distribution would have improved the delay. Mancas (2014) explored various software development methods like the Iterative method of software development (e.g., Agile), Lean Software Development (LSD), Waterfall, and others; all suggested a need for the well-defined scope of the projects. The research emphasized the importance of understanding the user requirement upfront with utmost clarity, applying upfront analysis of the complexity of the project and risk involved, and making necessary provisions for that. In another similar study, Lopes & Manas (2013) concluded that stakeholder management failures could lead to the delay of the project delivery.

While there is evidence of delays and overruns in software projects that broadly talk about the reasons behind it, not many papers suggest a solution to reduce these overruns. The Chaos report (1994) presented the magnitude of the losses due to overruns. There are other research papers either independently reported these losses or provided a critical review of the Chaos report. Still, there isn't any research work available which suggests a way to reduce these overruns. In industry, people use various project management practices to minimize budget overruns and time. Some work in some situations, but there is no acceptable method available in the existing literature. Dolo (2013), using confirmatory factor analysis, found that better control and governance together with proper planning and monitoring can reduce the cost overrun significantly. This has motivated us to develop a methodical approach to define scope as a pre-project exercise. We propose a pre-project "*Scope Confirmation Exercise (SCE)*." as an activity to determine the scope definition with several other details that are critical to the success of the project. We recommend an "SCE" to be done before any big software project, especially when there is a considerable amount of complexity involved. We have also evaluated an SCE workshop we have done with our software client from March 2010 to March 2021 and showed with the help of Ordinary least square regression, a causality relationship between the sizes of the SCE to the success the Project delivery. We have indicated that the SCE size is directly related to the success of the project (as measured by the TUB Score).

REVIEW OF LITERATURE

Requirements for Successful Software Delivery Project

Like any development project, software development (S.D.) has its unique challenges. Most of these challenges are largely obscure. For those projects that have been completed, diverse approaches centered on innovation have been relatively adapted. A successful project, by definition, is one whose deliverables match the stipulated timelines of the project. Therefore, clear

and realistic timelines must be set before the commencement of any given project. For most projects, delays account for most of the risks that lead to stalling the project and eventual failure. For SD projects, it is no different. Team members who do not deliver their work input in time cause a considerable delay in the testing phase, creating a backlog of work that can lead to Death Valley: a situation in SD projects where projects stall and eventually die off. In such an event, the risk of delays and failure is increasingly high, and it is upon project managers to ensure that there are constant updates from team members even when there are no testing activities that are required or anticipated. This writing aims to inform the reader about diverse approaches that can be adapted in S.D. projects to overcome instances of the Death Valley.

Research and Development

In SD projects, there have been cases where emergent and divergent ideas are continuously developed and adapted. In the course of the project, Scaringella et al. (2017) noted a need for S.D. projects to have absorptive capacities to accommodate new customer demands in the development of radical innovations. Spin-offs generally require a blending capability in the market and technical knowledge, market-pull and technological-pull approaches, involvement of customer and parent research centers, and potential and realized absorptive capacities. Lalband and Kavitha (2019) outlined the software development life cycle (SDLC) stages to include planning, analysis, design implementation, testing, deployment, and evolution. R&D, however, determines the project's starting point. Arguably, R&D remains to be one of the most critical elements of a successful project. Briand et al. (2017) explained that it was important for S.D. projects to conduct context-driven research aimed at solving real-world problems that would impact the industrial sector. Where R&D is conclusive, the scope of work is well defined, and therefore, there are far fewer ambiguities that can be expected at the project's inception.

For projects that have carried out extensive R&D activities, spin-offs are very few, mainly because the inadequacies of R&D findings that are not conclusive usually bring about knowledge deficits in the early-stage development (ESTD) in terms of how this knowledge is transformed and applied. Miah et al. (2018) explained design science research (DSR), a tool used to identify the major long-term issues in I.S.s design. The authors further noted two significant issues: the absence of rigorous research in designing artifacts and the nature of I.S. research outputs as not having practical knowledge. Therefore, it is vital that before any development activities, findings from the R&D phase are subjected to a transformation process to synthesize applicable scenarios within the S.D. context. These scenarios are then developed into modules that crystallize the essential functions of the software program. Therefore, what follows R&D should entail a software documentation process where the developer teams led by the project manager jot down all the feasible project modules along with their possible outcomes and how they are handled and rendered.

Funding and Management Issues

The subject of resources is broad as it touches on the essential factors of production, and therefore with this, the question of viability ultimately comes into sharp focus. Stakeholders, like the entrepreneurs, give rise to the project by putting together all other factors through the provision of capital. Salam et al. (2021) explained corporate social responsibility (CSR) and how it was a critical variable that influences the funding of SD (Software Development) projects. Where SD projects improved the general outlook of the organization, in terms of the overall impact generated,

there would be more funding from stakeholders hence increased chances of the project's continuity and success. Meng & Boyd emphasize the need to develop a favorable working environment which is the role of the project manager. The authors explained that project managers are responsible for developing the working relationships that would ultimately determine the success or failure of the project. For stakeholders, the most critical subject remains to be the viability of the project. Arguably, the commercial viability determines the level of funding that a project can get. Effectively budget can make or destroy a project relative to its commercial viability. However, there are projects aimed at scientific breakthroughs that then redirect the attention and priorities of entities such as governments that seek to advance the social and economic prospects of the nation at a more comprehensive level. This might be prospects in public transportation, public health and education, construction, and, more importantly to most governments, is the subject of military advancements.

Therefore, project management (PM) is another essential factor determining how well a project can be completed. Maqbool et al. (2017) explained that whereas project stakeholders strive for the project's success, project managers' competencies played an important role. The authors noted that the leadership styles and approaches of the project managers were key factors in the project's overall success. The vital part of the project manager should be to articulate the needs of the development teams while mediating between project owners and developers on the feasibility of specific project modules. The project manager ensures that the development process is well managed and that there is cooperation amongst the development teams. Dima and Maassen (2018) described situations where S.D. projects were scuttled due to conflicts and competition that arose due to poor communication and lack of sharing. Successfully SD projects are rarely cited with such instances of poor communication. Ideally, development teams should be well connected, with updates being shared promptly across team members in a timely fashion.

Methodologies

For SD projects, there are a variety of SD methodologies are employed. Saeed et al. (2019) state that "*each project contains a system that follows these systems to achieve project completion*". Furthermore, Saeed et al. (2019) noted that knowledge of each of these S.D. methodologies is difficult for project managers and that determining the most appropriate is even tricky. For this reason, it is essential to employ dynamic methods that can accommodate the continuous changes expected in the development stages of the S.D. project. According to Jurado-Navas & Munoz-Luna (2017), this methodology allowed for changing ideas in the course of the project development. It is important to note that the principle methodology employed determines the rate at which deliverables are provided and at what time, along with the stipulated timelines of the project. In SD, there are methodologies such as Waterfall, prototyping, extreme programming, rapid application development, v-mode, spiral and agile (Saeed et al., 2019). Relative to the project's scope, suitable methodologies must be adopted for efficient development and deployment. Dima and Maassen (2018) stated that "*the Scrum method is a version of the Agile model that supports a quick implementation of a new customer requirement*". According to Saeed et al. (2019), Scrum methodology allows for the development of tasks by an independent development team, a traditionally prescribed role for project managers. Projects that employ scrum methodologies in S.D. are bound to overcome project delays since it allows for rapid development and updates from either side of the development teams and project owners.

Viability and Feasibility Issues

SD projects are constantly evaluated in terms of their overall viability, mainly in the commerce spectrum. The factors of viability and feasibility are strongly tied to the factors related to funding and Management. However, viability alone raises the question of the overall benefits of the S.D. project relative to the preexisting conditions that may render a given project invalid to certain extents despite achieving the set-out project objectives. Hyrynsalmi et al. (2018) explained the concept of a minimum viable product (MVP), which was an approach involving developing a prototype that would be used to gauge the viability of the final product. It essentially induces a hypothetical use case scenario of the final intended product. Where the MVP fails to reach a target user base, it is rendered non-feasible. Okolo and Baha (2019) explained that selecting a software project was a critical decision that involved ascertaining how much value the project brings to the organization. Where the viability of a project tends to shift, the project is likely to suffer since it would imply that there is no purpose for the project in the long run. Okolo and Baha (2019) explained the need for applying artificial intelligence (A.I.) approaches in developing robust models that could best determine and predict software projects that could bring the most value business-wise. Okola and Baha (2019) described the A.I. approach that applied a branch of A.I. known as artificial neural network (ANN), classified projects into three levels. ANN method proved to be 98.67% accurate with a standard deviation of about 2.6%. The use of diverse and robust research models in software project selection such as ANN ultimately determines how well the viability and feasibility of a project can be evaluated. Viability, therefore, is an important subject that needs to be keenly assessed to ensure that project is sustainable from inception to completion. Sustainable projects are consequently those that are both viable and feasible. Therefore, the sustainability of the S.D. project is an important factor that can significantly determine how well a project can be completed.

A Scope Confirmation Exercise (SCE) Framework

Based on the above section (Introduction and Literature review), there is at least a consensus that scope management is critical to the success of any project. As suggested in the various literatures, to prevent a project from cost and time overruns, a better & robust definition of scope is required. A well-organized SCE delivers:

1. A well-defined project scope, covering an entire user requirement Clarity.
2. Create a well-defined boundary of the project enclosing the scope, good to create an MVP (Minimum Viable Product) of the product line.
3. Risk assessment of the project with a pragmatic mitigation plan.
4. Proper resource planning along with Skill-Matrix mapping.
5. Joint ownership of stakeholders: Formation of steering Committee with identified responsibilities to provide leadership, guidance and support to the project.
6. A set of project documentation agreed upon and signed off by significant stakeholders.

In a normal execution of the project, we generally do not spare time separate for determining the project's scope definition with the above details. While it is not common to run a Scope confirmation exercise as an independent upfront pre-scoping exercise, it has been observed that some project managers do this type of pre-project analysis though not in a well-defined and structured format. In this paper, we wanted to recommend a Scope Confirmation Exercise (SCE) framework in order to best understand the scope of the project from every aspect of the project.

Scope confirmation exercise (SCE)

Scope confirmation exercise is a joint exercise conducted between a team of stakeholders and the development team/vendor team to finalize the points mentioned above. This is an exercise to jointly discuss the scope of the project and create the definition of MVP (Minimum Viable product), hash out the details of every module of the requirement, a high-level software component mapping to those modules, and ultimately provide a detailed scope confirmation document, design blueprint, an indicative solution document, Resource Skill Matrix, and then finally the project documents like Project Plan, Risk Assessment and Mitigation Plan, Budget and contingency plan, A pre-agreed scope modification methodology and agreement, Project governance plan and ultimately the statement of work (SOW). SCE doesn't have a formally defined structure yet, so we propose the following SCE activities.

An agreement to conduct SCE between the development/vendor team and Stakeholders/Project sponsors followed by a detailed objective, schedules, and the essential deliverables post-SCE. Both parties should sign this as to the first thing of SCE.

Pre-scoping questionnaires should be sent to the client on various topics that the client should be ready to discuss in the scope definition workshop. These questionnaires also give the client an idea of what to expect in the SCE workshop, and they can align separate business line SMEs (Subject Matter Experts) to answer those discussions and, most importantly, to discuss the MVP (Minimum Viable Product) as a part of the software solution.

Format and schedule of the SCE workshop: A client and a software vendor must agree to a program of the SCE workshop well in advance. A rule of thumb is that the SCE effort should be around 12% of the tentative budget of the project. Out of these, 40% should be spent in the actual workshop, meeting with clients and SME and the remaining 60% is for the documentation and presentation to the clients.

Participants are the most critical factors in this SCE workshop. It is essential that from the Software vendor or development team, we must have Solution architects, Senior developers, Business Architect, Senior Business Analysts, Senior Project Manager along with an active engagement of SLT (Senior Leadership Team) members who could be C-level executive take part in the SCE. From the client side, it is expected that they should bring their business line SMEs, their software infrastructure team, their team of business analysts, and also their SLT members for the first few sessions.

The SCE exercise should be scheduled at least six weeks before the project's actual start so that the vendor team gets enough time to prepare the deliverable artifacts and the client has enough time to understand and assimilate those artifacts.

Finally, agreed upon SCE documents should be delivered by the Vendor team to the Stakeholders team or Project Sponsors.

Deliverable of the SCE

The SCE should satisfy those seven-objective mentioned in the SCE framework section. While there is no prescribed deliverable here but in general, the deliverable includes but is not limited to the following artifacts:

1. Summary of the SCE workshop
2. Software design blueprint explaining a high-level design aspect of the software. It should also include both functional and technical design and any interfaces needed to be built. A design blueprint has been depicted in Figure 1.

3. Module map is a technical mapping of the software modules to the user requirement with a dependency diagram that can help project scheduling and planning purposes.
4. It is a comprehensive solution document that depicts the future state of the flow with relevant screenshots, diagrams, and other details. This solution document can later be used as an implementation guidebook, so it needs to be as comprehensive as possible with the given amount of time and budget of SCE.
5. A mutually agreed project execution methodology (e.g., Agile, Waterfall, Hybrid, etc.)
6. A detailed analysis of Software gaps, plan to address those gaps with a mutually agreed timeline of delivery if these gaps include the method of delivery (e.g., Software upgrades or patches or workarounds)
7. Project Risk assessment and mutually agreed on a mitigation plan.
8. A mutually agreed scope change/scope management methodology
9. A resource matrix is generally a blended organization chart with roles and responsibilities from both sides. This will be used as a Project organization chart with well-defined roles and responsibilities.
10. Project governance details may include meeting schedules of SLTs, Milestone checks, Escalation process, and other information.
11. A detailed project plan, effort estimate, SOW (Statement of work), and Total Cost of Delivery (COD).
12. Any other details client may ask for.

Historically we have observed that a detailed SCE (Scope confirmation exercise) has resulted in more predictable software delivery with better risk mitigation, scope management, resource planning, and ultimately a project with minimum budget overruns and timelines. We validated our hypothesis that more time spent on SCE (Scope Confirmation Exercise) ensures successful project delivery.

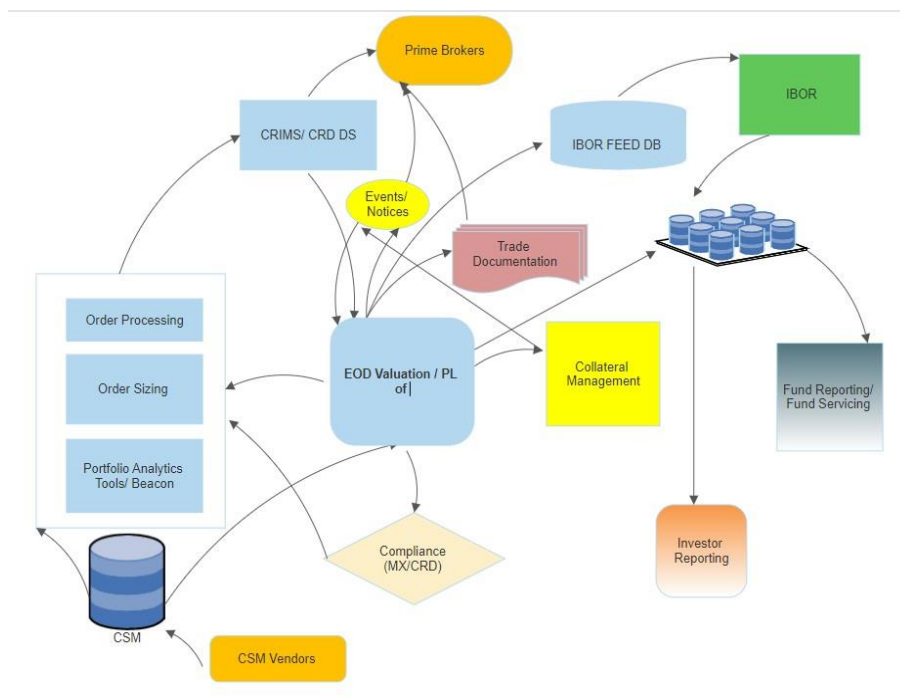


FIGURE 1

A HIGH-LEVEL DESIGN BLUE OF AN INSTITUTIONAL ASSET MANAGER

Validation of Pre-Project SCE (Scope Confirmation Exercise) as a Success Criterion of a Software Project

We have tested and validated the SCE as a criterion for successfully delivering the project, including minimum overruns of time and budget. We also included other factors, which measure the successful delivery of software projects. While we didn't borrow our project success factors directly from the CHAOS Report (1994), we were inspired by that report. We surveyed our end clients' responses, recorded their responses post project delivery, and prepared a Project Score. Variables in the project score are:

1. Timeliness score
2. Usefulness Score
3. Budget score

We wanted to call this a Project's TUB (Timeliness, Usefulness, Budget) variables and a TUB Score overall. Point to be noted here that Timeliness, Usefulness & Budget score cannot be precisely measured in hours is a range. That range is often subject to perception and interpretation, so a feedback-based index is required. Generally, 10-12% variance in the budget and time is said to be good in one organization, while for other organizations, it needs to be less than 5% to be called good. Therefore feedback from the broad spectrum of users/stakeholders was required to add coherence to the TUB score. We prepared an overall TUB score per project and compared a comparative analysis of how a project has been done compared to the effort and time spent on the SCE (Scope Confirmation Exercise). We expected that more effort and time spent on the SCE would ensure a better project score, which means a project delivered with minimum overruns of time/budget and provides high client satisfaction.

Timeliness Score

Projects completed within time have a higher Timeliness score.

Usefulness Score

How much original objective the project has achieved in terms of business process re-engineering, Operational Efficiency, successful decommissioning of any existing software module/functionality, etc.

Budget Score

Projects completed within budget has a higher Budget score, while those with high budget overruns have a lower Budget score.

DATA AND DATA COLLECTION METHODOLOGY

We sampled 20 capital market fintech projects delivered between March 2010 to March 2021. These projects implement "*Trading and Risk Management*" software for Sell sides, Buy sides, Market places, and various corporate treasuries. Sell-side clients include top-tier Banks, Broker-dealers, Tier 2 banks wanting to automate the corporate client business, and other small banks. Buy-side clients included Hedge Funds, Institutional Asset Managers, Pension Funds, Insurance management firms, Endowment funds, and corporate treasuries.

Software Used

A Capital Market Software was used. The functionality of this software includes but is not limited to:

1. Trade generation and execution of assets like Equities, Bonds, Derivatives, and some of the alternate assets,
2. Pricing and valuation of the assets using various models and methodologies.
3. Management of Market data, Reference data, Static data, and other relevant data.
4. It has Comprehensive coverage of post-trade processing capabilities such as clearing, settlement, recording of transactions, IBOR (Investment Book of record), trade confirmation, accounting, etc.

Clients implement this software for automating their business process and streamlining their business by gaining operational efficiency and reducing the cost of business.

We grouped the Clint side team into three broad categories. First Category was comprised of SLT (Senior Leadership Team) members like Senior Portfolio Managers, Head of Program Management, Head of the Trading desks, CTOs, etc. The second category comprised Mid-Level Managers like Senior Business Analysts, Junior Portfolio Managers, Financial Engineers, Quants, Tech Lead and Back Office Managers, etc. Our third Category was from junior staff that has done the major hands-on work in the projects; they are Business Analysts, Software Engineers, Project staff, etc. We gave equal weights to the responses of each Category of responders and finally created a TUB Score for each of those 20 projects done between March 2010 to March 2021 (11 years). The score was recorded out of 100. This score is compared with SCE (Scope confirmation Exercise) score. SCE score is defined as the ratio of Estimates of SCE (In Man days) to the total estimates of the actual project multiplied by 100. So basically

$$CE\ Score = SCE\ Estimates\ (In\ Man\ Days) / Total\ Actual\ project\ estimates\ (In\ Man\ Days)$$

We asked a few more questions, which were later excluded from the TUB index; the purpose of these additional questions was just to gauge the consistency of the responses. Some of those questions were: Engagement level in the project, Opinion about vendor resources, Ease of doing business with the vendor, Project transparency, etc. While we did not follow the exact recommendation of the Likert Scale as suggested by Croasmun & Ostrom (2011), we kept the underlying spirit of the Likert Scale in our study (Figure 2).

We also came up with a simple regression-based model to show the causality relationship between the SCE Score and Project TUB Score, which can be a valuable insight for any project.

It was expected that the TUB score would be directly influenced by the SCE Score:

$$Aggregate\ TUB\ Score = \beta_0 + \beta_1.SCE\ Score$$

$$Aggregate\ TUB\ Score = \beta_0 + \beta_1.SCE\ Score$$

Where,

β_0 =regression constant; and β_1 =slope of the regression line

In most projects, the quality of resources was almost consistent, but it could be a minor limitation of the data. More details on this little data limitation are available in the discussion section.

Following is our sample of projects done in 11 years from March 2010 to March 2021 (Table 1). The client list includes both sell-side and buy-side clients. From the sell side, software implementation projects were delivered for Broker-Dealers, Tier 2 Banks, Market places (Exchanges and Clearinghouses). From the Buy-side space, the client list includes Institutional

Asset Managers, Hedge Funds, Pension Funds, Insurance Firms, etc. More details of financial market participants and institutions can be found in Saunders and Cornett (2012).

Project ID	Business Side	Business Line
1001	Sell-Side	Broker-Dealer
1002	Sell-Side	Tier 2 Bank
1003	Buy-Side	Institutional Asset Manager
1004	Sell-Side	Broker-Dealer
1005	Sell-Side	Broker-Dealer
1006	Buy-Side	Pension Fund
1007	Buy-Side	Hedge Fund
1008	Buy-Side	Hedge Fund
1009	Market Place	Exchange
1010	Buy-Side	Insurance Fund Manager
1011	Market Place	Clearing House
1012	Sell-Side	Broker-Dealer
1013	Buy-Side	Institutional Asset Manager
1014	Sell-Side	Broker-Dealer
1015	Sell-Side	Valuation Platform
1016	Buy-Side	Alternative Investment Firm
1017	Sell-Side	Tier 2 Bank
1018	Buy-Side	Institutional Asset Manager
1019	Sell-Side	Tier 2 Bank
1020	Sell-Side	Valuation & Risk Platform

TUB (Timeliness, Usefulness & Budget) Index score as reported by three sets of personnel from the client-side per project (Table 2). TUB Index score provided reported by three Categories of Client personals.

Project ID	Senior Leadership Team	Middle Management	Project Staff
1001	86	76	70
1002	84	74	69
1003	90	80	76
1004	55	45	35
1005	56	44	42
1006	78	75	72
1007	48	38	32
1008	76	64	60
1009	42	36	30
1010	85	80	80
1011	84	80	78
1012	50	42	44

1013	80	80	82
1014	60	60	56
1015	48	46	46
1016	85	80	76
1017	70	70	62
1018	45	40	35
1019	70	68	76
1020	40	35	35

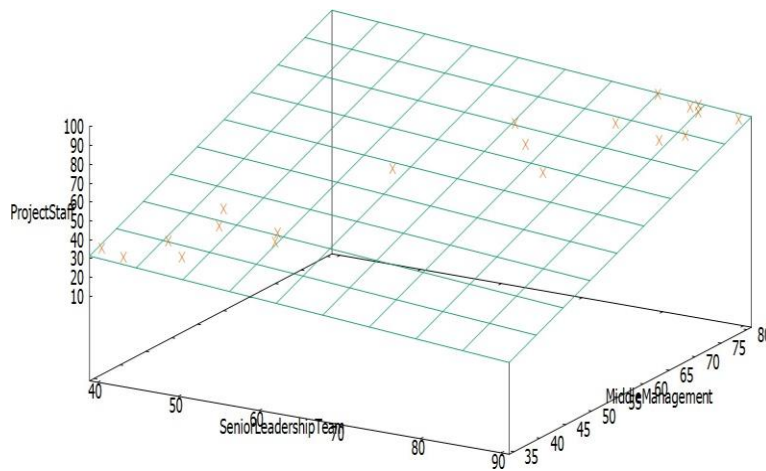


FIGURE 2

TUB INDEX RESPONSE FROM THE THREE GROUPS OF CLIENT PERSONALS

The reason behind taking responses from the three categories of stakeholders is to get a holistic view of the project's feedback. Restricting ourselves into one particular group would give us a biased opinion of the feedback. Another reason for reaching out to 3 various groups was to incorporate every section of the project staff.

Summary Statistics of the TUB Index score

Table 3				
SUMMARY STATISTICS OF TUB SCORES BY RESPONDENTS' CATEGORIES				
Variable	Mean	Median	Minimum	Maximum
Senior Leadership Team	66.6	70	40	90
Middle Management	60.65	66	35	80
Project Staff	57.8	61	30	82
Variable	Std. Dev.	C.V.	Skewness	Ex. kurtosis
Senior Leadership Team	17.267	0.25926	-0.17163	-1.5393
Middle Management	17.673	0.2914	-0.23409	-1.6415
Project Staff	18.58	0.32146	-0.209	-1.5452
Variable	5% Perc.	95% Perc.	I.Q. range	Missing obs.
Senior Leadership Team	40.1	89.8	35.5	0
Middle Management	35.05	80	36.5	0
Project Staff	30.1	81.9	39.25	0

Table 3 and Table 4 represent summary statistics of TUB scores by respondents' categories and TUB score vs. SCE score respectively.

Project ID	Total Actual project estimates (In Man Days)	SCE Estimates (In Man Days)	SCE Score	Aggregate TUB Score
1001	1320	162	12.27	77.33
1002	2400	304	12.67	75.67
1003	600	80	13.33	82.00
1004	1260	105	8.33	45.00
1005	960	72	7.50	47.33
1006	1020	138	13.53	75.00
1007	360	20	5.56	39.33
1008	768	94.8	12.34	66.67
1009	864	60	6.94	36.00
1010	792	132	16.67	81.67
1011	840	96	11.43	80.67
1012	1134	84	7.41	45.33
1013	392	51.2	13.06	80.67
1014	744	60	8.06	58.67
1015	1260	84	6.67	46.67
1016	828	94.8	11.45	80.33
1017	996	144	14.46	67.33
1018	1140	90	7.89	40.00
1019	1246	175	14.04	71.33
1020	440	20	4.55	36.67

Summary Statistics of TUB Score and SCE Score

Summary Statistics of TUB Score and SCE Score, using the observations 1-20 has been provided in Table 5.

Variable	Mean	Median	Minimum	Maximum
SCE Score	10.408	11.439	4.5455	16.667
AggregateTUB Score	61.683	67.000	36.000	82.000
Variable	Std. Dev.	C.V.	Skewness	Ex. kurtosis
SCE Score	3.4494	0.33141	-0.049875	-1.2128
AggregateTUB Score	17.606	0.28543	-0.22434	-1.6219
Variable	5% Perc.	95% Perc.	I.Q. range	Missing obs.
SCE Score	4.5960	16.556	5.8348	0
AggregateTUB Score	36.033	81.983	34.500	0

RESULTS, DISCUSSION, AND IMPLICATIONS

Table 6 presents regression output of Ordinary Least Square model where dependent variable is Aggregate TUB Score.

Table 6 REGRESSION OUTPUT				
	Coefficient	Std. Error	t-ratio	p-value
Const	14.2240	5.91310	2.405	0.0271
SCE Score	4.55975	0.540608	8.434	<0.0001
Mean dependent var	61.68333	S.D. dependent var	17.60631	
Sum squared residual	1189.290	S.E. of regression	8.128447	
R-squared	0.798072	Adjusted R-squared	0.786853	
F(1, 18)	71.14052	P-value(F)	1.14e-07	
Log-likelihood	-69.23256	Akaike criterion	142.4651	
Schwarz criterion	144.4566	Hannan-Quinn	142.8539	

White's test for heteroskedasticity -

Null hypothesis: heteroskedasticity not present Test statistic: LM=2.8248

With p-value= $P(\text{Chi-square}(2) > 2.8248) = 0.243558$

Therefore, Null hypothesis cannot be rejected

Breusch-Pagan test for heteroskedasticity - Null hypothesis: heteroskedasticity not present Test statistic: LM=0.94374

With p-value= $P(\text{Chi-square}(1) > 0.94374) = 0.331318$

Therefore, Null Hypothesis cannot be rejected.

Breusch-Pagan test for heteroskedasticity (robust variant) - Null hypothesis: heteroskedasticity not present

Test statistic: LM=1.71491

With p-value= $P(\text{Chi-square}(1) > 1.71491) = 0.19035$

Therefore, Null Hypothesis cannot be rejected.

Test for normality of residual -

Null hypothesis: error is normally distributed

Test statistic: Chi-square(2)=0.467938 with p-value=0.791386

Therefore, Null Hypothesis cannot be rejected

Model is given by

$$\text{Aggregate TUB Score} = 4.56 * \text{SCE Score} + 14.2$$

Here Aggregate TUB score is a measure of the success of the project delivered (Figure 3). A higher TUB score ranks a project higher in terms of the quality. A lower TUB score signifies a project not paid within time and budget and low client satisfaction. SCE score, as defined earlier, is the ratio of effort/budget spent on the Scope confirmation exercise (SCE) compared to the effort/budget of the actual project. A higher SCE score means a reasonable amount of time/resource spent on SCE, while a lower SCE score suggests that less amount of time and resources have been spent on SCE as compared to the actual project. The above equation explains the relationship between the Aggregate TUB score and the SCE score. A Higher SCE score explains that a considerable amount of time has been given in understanding

the project before the actual execution of the project, which also means that every small issue has been discussed among vendors/ development teams and stakeholders. This also sounds logical that more time spends and pains taken before the project would reduce the pain in the project. While the example of this Scope confirmation exercise is from Capital Market Fintech products, but it can very well be applied to any project which involves complexity. One of the major implications of doing an SCE prior to the project's actual start is getting to know the people from both sides well in advance. The in-person SCE meeting and SCE artifact preparation exercise involves frequent interactions between the vendor team and Stakeholders/end clients. This gives an opportunity to build a relationship between the two teams, which generally proves very fruitful in the long-term success of the project.

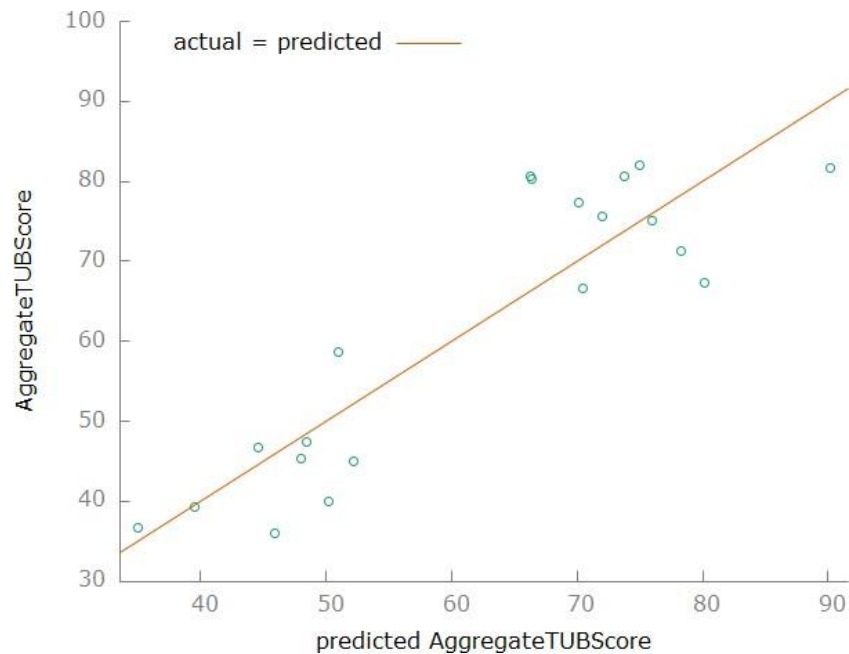


FIGURE 3

ACTUAL VS. FITTED

Analysis of Results and Its Implications

This model establishes a relationship between SCE effort and the Success index. It is a very important relationship that a fintech company can leverage to deliver a successful project. A company can target a success rate by targeting a TUB scope, and that will give them an idea of how much SCE (Scope confirmation exercise) should be done in order to meet that success criterion. This model can be re-purposed for any industry once it is calibrated for that industry data. It is expected that the model may have variations in predictability if used in different sectors, but conceptually, this model has portability to various other industries also. This model can also use to benchmark the project. Projects can be ranked based on a cut-off TUB Score. It is possible to say that a project with a TUB score higher than a level can be ranked one, which signifies the highest level of achievement in Project delivery (Table 7).

Project TUB Score	Rank	Interpretation
$TUB \geq 75$	1	Beat Client Satisfaction
$50 \leq TUB < 75$	2	Meet Client satisfaction
$TUB < 50$	3	Doesn't meet Client Satisfaction

Therefore, this model can be used in various different ways from project planning, resource and budget estimations, overall project management.

CONCLUSION

Limitations and further improvements: One of the minor limitations is that the survey questions were formed much later after the project is completed. One of the data collection issue was not having the actual person available to answer those survey questions. We decided to run this analysis based on the client satisfaction survey question that we generally ask. Some of the questions which were important to evaluate the project were not part of the original client survey questionnaires, and we went back to our previous clients to ask those questions. There were a couple of challenges to that; at one place, we didn't find all the people who were part of the project back then, and another challenge was the time elapsed. We might have got slightly different answers which would have been more relevant in today's time. Technology in the last ten years has changed considerably. Another limitation of the data is not having resource quality as a separate factor. While generally, our team had normally distributed skill sets but that aspect may be researched and reported.

This model can be further extended to various other technology project implementations, and in fact, with data availability, it can be calibrated and used for other industries. A logistic regression model can also be built to develop a probability model for the success and failure of the project.

REFERENCES

- Aljohani, A., Ahiaga-Dagbui, D., & Moore, D. (2017). Construction projects cost overrun: What does the literature tell us?. *International Journal of Innovation, Management and Technology*, 8(2), 137.
- Briand, L., Bianculli, D., Nejati, S., Pastore, F., & Sabetzadeh, M. (2017). The case for context-driven software engineering research: generalizability is overrated. *IEEE Software*, 34(5), 72-75.
- Chaos Report (1994). *The Standish Group*. The Standish Group.
- Croasmun, J.T., & Ostrom, L. (2011). Using likert-type scales in the social sciences. *Journal of Adult Education*, 40(1), 19-22.
- Dima, A.M., & Maassen, M.A. (2018). From Waterfall to Agile software: Development models in the IT sector, 2006 to 2018. Impacts on company management. *Journal of International Studies*, 11(2), 315-326.
- Doloi, H. (2013). Cost overruns and failure in project management: Understanding the roles of key stakeholders in construction projects. *Journal of Construction Engineering and Management*, 139(3), 267-279.
- Damasiotis, V., Fitsilis, P., & O'Kane, J. (2014). Scope management complexity in software projects: An approach to evaluate it. *British Academy of Management (BAM)*.
- Van Genuchten, M. (1991). Why is software late? An empirical study of reasons for delay in software development. *IEEE Transactions on Software Engineering*, 17(6), 582.
- Hyrnsalmi, S., Klotins, E., Unterkalmsteiner, M., Gorschek, T., Tripathi, N., Pompermaier, L. B., & Prikladnicki,

- R. (2018, October). What is a minimum viable (video) game?. In *Conference on e-Business, e-Services and e-Society* (pp. 217-231). Springer, Cham.
- Jurado-Navas, A., & Munoz-Luna, R. (2017). Scrum Methodology in Higher Education: Innovation in Teaching, Learning and Assessment. *International Journal of Higher Education*, 6(6), 1-18.
- Lalband, N., & Kavitha, D. (2019). Software engineering for smart healthcare applications. *International Journal of Innovative Technology and Exploring Engineering*, 8(6S4), 325-331.
- Lopes, L.T., & Mañas, A.V. (2013). Delays in IT Projects Due to Failures in the Stakeholders Management. *Future Studies Research Journal: Trends and Strategies*, 5, 155-186.
- Mancas, C. (2014). Will Software Development Projects Always Risk Delays. *Journal of Information Technology & Software Engineering*, 4, e123.
- Maqbool, R., Sudong, Y., Manzoor, N., & Rashid, Y. (2017). The impact of emotional intelligence, project managers' competencies, and transformational leadership on project success: An empirical perspective. *Project Management Journal*, 48(3), 58-75.
- Bloch, M., Blumberg, S., & Laartz, J. (2012). *Delivering large-scale IT projects on time, on budget, and on value*. *Harvard Business Review*, 5(1), 2-7.
- McQuighan, J. M., & Hammell II, R. J. (2011, August). Scope as a leading indicator for managing software development. In *2011 Ninth International Conference on Software Engineering Research, Management and Applications* (pp. 235-241). IEEE.
- Meng, X., & Boyd, P. (2017). The role of the project manager in relationship management. *International Journal of Project Management*, 35(5), 717-728.
- Miah, J., S., Gammack, J. & Hasan, N. (2018). Methodologies for designing healthcare analytics solutions: A literature analysis. *Health Informatics Journal*, 26(4), 2300-2314.
- Mirza, M.N., Pourzolfaghar, Z., & Shahnazari, M. (2013). Significance of scope in project success. *Procedia Technology*, 9, 722-729.
- Nonaka, M., Zhu, L., Babar, M.A., & Staples, M. (2007, July). Project cost overrun simulation in software product line development. In *International Conference on Product Focused Software Process Improvement* (pp. 330-344). Springer, Berlin, Heidelberg.
- Okolo, O., & Baha, B.Y. (2019). Business viability of potential software projects using artificial neural network. *Journal of Computer Science and Its Application*, 26(1).
- Saeed, S., Jhanjhi, N. Z., Naqvi, M., & Humayun, M. (2019). Analysis of software development methodologies. *International Journal of Computing and Digital Systems*, 8(5), 446-460.
- Salam, A., Hikmat, I., Haquei, F., & Badariah, E. (2021). The Influence of Share Ownership, Funding Decisions, Csr and Financial Performance of Food Industry. *Annals of the Romanian Society for Cell Biology*, 12698-12710.
- Saunders, A., Cornett, M. M., & Erhemjamts, O. (2012). *Financial markets and institutions*. McGraw-Hill/Irwin.
- Scaringella, L., Miles, R. E., & Truong, Y. (2017). Customers involvement and firm absorptive capacity in radical innovation: The case of technological spin-offs. *Technological Forecasting and Social Change*, 120, 144-162.

Received: 13-Jan-2022, Manuscript No. JMIDS-22-10862; **Editor assigned:** 20-Jan-2022, PreQC No. JMIDS-22-10862(PQ); **Reviewed:** 22-Feb-2022, QC No. JMIDS-22-10862; **Revised:** 26-Feb-2022, Manuscript No. JMIDS-22-10862(R); **Published:** 28-Feb-2022