



ISSN 2047-3338

Towards Identification and Refinement in Modern Software Development Systems

Maliha Zafar¹, Rizwan Ali²

¹Pir Mehr Ali Shah Arid, Agriculture University, Rawalpindi, Pakistan

²Department of Computer Science, UET, Lahore, Pakistan

¹mali hazafar2@gmail.com, ²rizwanhvl@gmail.com

Abstract– Software system enables us to perform complex business processes by using the modern software development approaches. Identification and refinement of requirements according to rely of the end-users at some time in the software evolution process is a major question. More vigorous issues with rely refers to the end-users is one of the rising features for the software systems. We propose scrumban framework with identification and refinement model to overcome this problem. Because scrumban is flexible approach, we can add changes at any time when we require even during sprint and handle these changes according to kanban board. We have made our development process interactive by involving users during all the phases of scrumban and try to achieve user satisfaction, clarity and trust.

Index Terms-- Scrum, Identification, Requirements, Agile, Software Engineering, Refinement and MSDA (Modern Software Development Approaches)

I. INTRODUCTION

SOME current approach of software development has numerous weakness such as long planning processes requiring long time made it outdated [9]. Therefore, practitioners started to search for other possible software development tools which can best cater such problems. Identification and refinement of requirements has become a major problem in modern software development systems and business operators focus on value creation in the circumstance of making better value of user to their services and products, as well as for the growth of benefits for participants involved in the business [21]. Agile methods are iterative and incremental, in that collaboration between self-organizing cross-functional team provides requirements and solutions. Scrum provides a simple framework by which people can work collectively in form of teams and can better address complex problems [20].

Scrum is characterized by agile framework which contribute to effective management of product development [9]. In modern software development systems different approaches use like scrum, kanban, outmost programming, feature-driven development, adaptive system maturation and lean software

growth etc. [16]. Scrumban merges the rules for scrum and kanban of one draw system. Work of a team was created during installation and without interruption prepares the backlog. In scrumban the same scrum encounter should take place, frequency can modify which depends on the condition and demand. The significant way of scrumban is to assure that the work in WIP limits are complied. We use scrumban when we evolve requirements, no clear roadmap, too fast changes and need to include support or maintenance.

In scrumban distinctive stages are included which incorporate introductory item like accumulation creation, item excess refinement, everyday scrum, meaning of done and dash review. The item build-up creation recognized an ordered rundown of those things that may be required for the generation. A source of requirements to any change can be created for the thing as indicated by the client delight [14]. The product proprietor is responsible for the item backlog, including its content, obtainability and association. Item backlog elaboration will be the demonstration of giving refined element, estimation and requesting the things in the item build-up. This is a continuing process in which the item proprietor, client and the improvement group coordinates on the points of interest of item build-up things. Among item accumulation refined things are reevaluated. The scrum group went to a choice that how and when refinement should be finished. Then again, item backlog can be refreshed at any time by the item proprietor or at the item proprietor's watchfulness [4].

Scrumban takes flakes and parts from both scrum and kanban. For instance, it contains the clarified roles, as usual scrum and meetings from scrum. From kanban, it takes the kanban committee, upholds stream, and quality to change as need of the committee [14]. Scrumban looks more like scrum on the expert grade but on the cultural grade, it will more near alike kanban. Besides of large changes all at once, scrumban promotes gradually change. Kanban board has outmost numbers of stories countenanced in each column at one time. Specified roles in scrum are:

- *Product Proprietor*: The scrum product proprietor aware of what he or she needs to make or discover the awareness for team. The product proprietor concentrates on business and market needs, giving value to all the activities for requirements to be done. He or

she constructs and handles the backlog, give direction on which features to board next and contact with the team and other shareholder to convince for the items in the product backlog. The product proprietor is not a plan director. Besides of directing the position and advancement, their job is to actuate the team [8].

- *Scrum Master*: Usually consider the director of the team, the scrum master assists the team to give the best potential. This means forming, gathering, facing the challenges, and slogging with the product proprietor to assure that the resultant backlog is ready for the next sprint. They do not have control over the team members, but they have governance over the action. For design, the scrum owner cannot express someone what to do but he gives a new sprint step [8].
- *Scrum Team*: The scrum team is incorporating from five to seven represents. Every person that connects with project works together, helps to each other. Unlike traditional evolution teams, there are not clear characters like programmer or designer. Everybody finishes the work together. The scrum group has the method for each dash as they expect that how much work they can do in every repeated performance [4].

II. PROBLEM STATEMENT

The present research issue is associated to the software creation evolutionary process of applications team within the case of company context. In recent years, a lot of organizations have started using quick methodologies for the effectiveness of software evolutionary projects. Few years back, the context of company also utilize agile methodology called scrum [14]. However, after applying the scrum methodology for a specific period of time, the software development teams, especially applications team, confronted difficulties. Some of the problems are discussed under: Difficult time completing the sprint planning [7], Missing collaboration and troubles in self-organization [5], Incomplete Jobs within a Dash [5], Product owners change priority within sprint [4], Deficient jobs for team members [4], and sudden change to kanban manner in some sprints [5].

Research Questions

Research questions of the current study are described below after thoroughly analysis of problem statement.

Q1: What are the drawbacks in the existing software development systems of scrum and Kanban?

Q2: How scrumban provides the solution for refining of user requirements in the software development?

Q3: How scrumban differentiates himself from the existing software development systems of scrum and Kanban?

III. RELATED WORK

After the detailed literature review, we provide an overview of existing literature that provides problem definition and its scope. The basic objective of this literature is to provide review of related publications that identify the main issues of identification and refinement in scrum and Kanban.

A case company, involving sixty-five employees, tries any online content, images, videos and applications into collaborative and viral

store faces by means of non-intrusive and satisfied related commercial goods for improving the existing agile software development processes. Researcher started with the existing state investigation of the agile, scrum process within the requests change team of the case company. The applicants used for interviews and discussions were from numerous backgrounds and sections of the case company and included specialists such as, scrum master, product proprietor, product executive, quality guarantee team, team supervisor, operation supervisor and software designers [14].

A model of continued agile practice to present a complete understanding of the key factors that are relevant to the continued practice of agile practices. It defines our qualitative study which involves: (i) an attention group with twenty-nine software engineering agile experts, and (ii) semi-structured interviews with twenty agile experts from five different structural backgrounds. Information from both approaches is used to advance the refined model of continued agile practice [32].

In [12] collected data required to report goal, invited SLR authors to join in a collaborative workshop organized around the insignificant group method. The workshop results specified that search & collection and teamwork are the two peak priority tool features. The results also showed that most of the peak importance features are not well maintained in current tools. SLR (Systematic Literature Review) tool, authors can use these findings to guide upcoming progress efforts.

In [35] presents a framework to automatically notice and categorize non-functional requirements from written normal verbal requirements. Method to identify NFRs (Non-functional requirements) is based on removing numerous structures by analyzing the normal verbal requirement whereby the occurrence of a firm mixture and connection among the structures exclusively identifies the requirement as an NFR of a specific group.

A policy refinement approach is proposed to optimize a sequence of dispatching rules (DRs) for a time window of scheduling process in which a GA algorithm evolves the sequences towards an optimum configuration. The policy refinement approach with GA (Genetic Algorithm) is applied to 10 JSS problems of 10x10 (M x N) size from library to optimize the schedules with respect to make span and mean tardiness objectives [26].

In [36] use cases are used as one requirement modeling technique. The proposed use case patterns include the following elements: pattern name, context, problem, solution, examples, consequence, related patterns, and known uses. This paper presents a methodology for object identification and refinement from the software requirements, which is based on object-based formal specification (OBFS). This methodology provides the mean of understanding the object-oriented paradigm easily and supports us with identifying and refining the objects.

In [17] modularity is being increasingly used as an approach to solve for the information overload problem in ontologies. In this article, survey the existing literature to identify and populate dimensions of modules, experimentally evaluate and characterize 189 existing modules, and create a framework for modularity based on these results. The framework guides the ontology developer throughout the modularization process. We evaluate the framework with a use-case for the symptom ontology.

In [4] proposes a structured approach for modeling domain knowledge. The domain knowledge is captured by using patterns in addition to exploring the requirements for avoiding an incorrect specification. His research introduces a semantically based three stage

approach to assist developers in checking the consistency of the requirements models.

Today, taking out change early means being lethargic to business conditions as it were, business disappointment. Correspondingly, customary process administration by consistent estimation, mistake recognizable proof, and process refinements endeavored to drive varieties out of procedures. This approach accepts that varieties are the after effect of mistakes. Today, while process issues unquestionably cause a few blunders, outer ecological changes cause basic varieties. Since we cannot dispense with these progressions, driving down the cost of reacting to them is the main practical system. As opposed to wiping out modify, the new methodology is to decrease its cost. Be that as it may, in not simply pleasing change, but rather grasping it, we additionally should be mindful so as to hold quality. Desires have become throughout the years. The market requests and expects imaginative, astounding programming that addresses its issues soon [1].

The engineers of these systems had an assortment of inspirations, yet principally they were searching for methods for saving the product business from what has all the earmarks of being a difficulty: programming is quite often costlier and conveyed later than anticipated, and to exacerbate the situation, usually inconsistent and neglects to meet a definitive client's needs. Venture directors frequently side step stages or take easy routes with a specific end goal to take care of the issue, however these spontaneous also, spontaneous adjustments of the life cycle simply make the product considerably costlier, later, and more problematic [2].

The quick disposable prototyping approach addresses the issue of guaranteeing that the product item being proposed truly meets the clients' needs. The approach is to develop a "fast and messy" halfway usage of the framework before (or amid) the prerequisites arrange. The potential clients use this model for a time frame and supply input to the engineers concerning its qualities and shortcomings. This criticism is then used to alter the product prerequisites determination to mirror the genuine client needs. At this point, the engineers can continue with the genuine framework plan and execution with certainty that they are building the "right" framework (with the exception of in those situations where the client needs develop) [4].

An expansion of this approach employs a progression of disposable models, finishing in full scale advancement. Incremental advancement is the way toward developing a fractional execution of an aggregate framework and gradually including expanded usefulness or execution. This approach decreases the expenses brought about before an underlying capacity is accomplished. It additionally delivers an operational framework all the more rapidly, and it in this way decreases the likelihood that the client needs will change among the advancement procedure. In developmental prototyping, the designers build a fractional execution of the framework which meets known necessities [7].

The scrum programming progression process depicted in this article made in a planned exertion between advanced developments strategies. The two associations were enumerating accomplishment gainfulness. This approach secured in a general sense with what we found in postmortems of successful endeavors. Our affiliation has since a long time back used illustrations successfully, not just for plot anyway for affiliation and process also. Numerous current illustrations supported what we got some answers concerning scrum and found in the after-death data [7].

Usage of Kanban in software growth is a developing matter. The main stated welfares of using the kanban technique were enhanced lead time to distribute software, enhanced quality of software, enhanced communication and management, increased reliability of transfer, and decreased client stated shortcomings. The stated encounters included lack of information and particular exercise as well as numerous structural issues. Moreover, proposed practices were removed from the key studies and concise for monitoring the experts concerned in accepting kanban. The results of this literature review are planned for helping researchers and experts to increase a better considerate of the present condition of kanban practice in software expansion. Some innovative has been taken with the extending all out-cost estimation to enhance coherence of the means. Each cycle of the twisting begins with the unmistakable confirmation of the objectives of the bit of the thing being clarified (execution, convenience, ability to oblige change.); the elective strategies for completing this bit of the thing; and the goals constrained on the usage of the decisions (cost, timetable, interface, et cetera.) [8].

IV. MATERIALS AND METHODS

The methodology may include publication research, surveys and other research techniques, and could include both present and historical information. As the item development process is cluttered and complex. Hence most vital adaptability and fitting control is required. Progression supports those that work with most remarkable prologue to natural change and have streamlined for adaptable acclimation to change. Change deselects the general population who have protected themselves from normal change and have obliged tumult and multifaceted nature in their condition. An approach is required that connects with change social affairs to work adaptively inside a flighty condition utilizing free techniques. Complex framework movement happens under quickly creating conditions. Passing on productive structures under turbulent conditions requires most important adaptability [10].

V. RESEARCH METHODOLOGY

Research methodology describes that how we perform our research. The purpose for this study is to describe our investigation viewpoint with decision. Fig. 1 represents our research methodology.

Survey Research

Survey represents one of the famous types of quantitative research. In survey research, the investigators select a sample of respondents from people and directs consistent questionnaires to them. The questionnaires or survey can be a written document that is complete through individual being studied, an online survey, face to face interview, or a phone interview. Applying surveys, it is believable to collect information from large or small population.

VI. PROPOSED FRAMEWORK

This study has proposed framework with the aim to mitigate the challenges which are associated with identification, refinement and user interaction. Scrum is the dominant agile framework, but studies

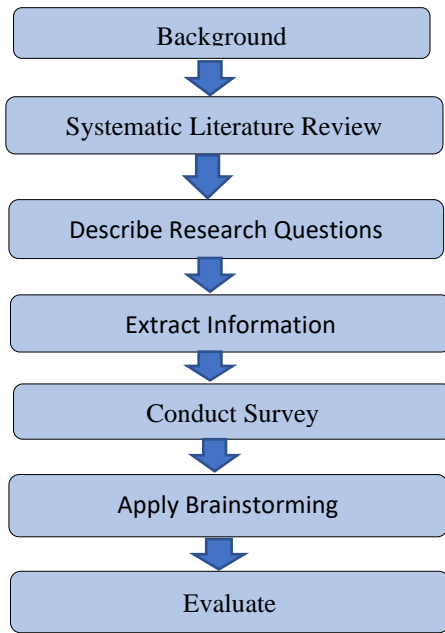


Fig. 1: Research Methodology

show that the stage comes where scrum fails to work alone and comprises with the issues like resist changes, no continuous flow, wastage of time in planning and no maximum number of stories in each column. To solve these issues, we will use a scrumban approach. In scrumban, first we generate all the phases of scrum. Then we defined roles and involve users in all the phases of scrum. We can add changes during the whole development process because we focus on our user satisfaction and clarity.

Kanban board is used to handle user stores. We proposed framework in which customer creates the user stories which produces product backlog. From product backlog the requirements are to be prioritized on the kanban board by product owner and scrum master. From kanban chart the prioritized requirements are selected by the team or an individual according to the specialties. Then from the kanban board requirements are pull into the task board by pull system just like in kanban.

Each task board has three columns To Do, Doing, Done to represent the flow of work. Each activity in the task board has WIP limit to limit the tasks performed by the developer per time. Our users are involved in the following phase's product backlog creation, product backlog refinement, sprint planning, daily scrumban meeting and sprint review. Specialist utilized the progression shrewd ID and refinement demonstrate which comprises of a few phases beginning from client prerequisites stage to the documentation stage. A concise clarification of each phase in step-wise identification and refinement shown is done in the accompanying 7 stages process.

A) User Stories

In user stories phase, we find some clients/users who interact with our software product. We make meeting with them and get information about product. In user story, the interaction between developer and user will more because the only user gives the right feedback to developers, so we interact with our clients and users for helping us and getting more to know how our product is.

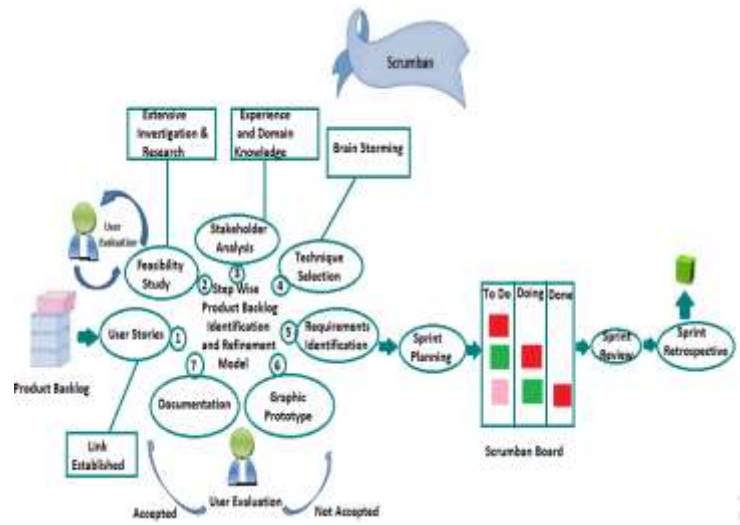


Fig. 2: Scrumban Framework

B) Feasibility Study

In feasibility study, we examine the technical stuff, non-technical stuff of our software like how much features needs, which language will be used, which platform need to develop this software, which requirement of clients are more important and which are less, how much time required to complete this project.

i). Technical achievability: In technical study, we examine the core and technical features of software like how much features needs, which language will be used, which platform need to develop this software.

ii). Legal practicality: In this we confirm that our software which is to be developed does not conflicts with legal requirements. This software will be different because of his features, performance, and compatibility.

iii). Economic practicality: In this phase, we analyze our software become profitable or not. Because as in market analysis we need to make place also.

If all these three factors are agreeable to the product at that point, no change is asked for and it is carried on to the following stage or else change demand will be made. In the last case, the progressions are to be assessed by the clients and based on which choices on confirmation is made. Henceforth client cooperation assumes a vital part in this stage.

C) Stakeholder Analysis

In stakeholder analysis, we clarify the stakeholder who used this software, or for which use this software will make like software houses, home users, corporate companies etc. These are main stakeholder who interact with our software. We take different steps for assumption. Because we did not select those users who don't have knowledge of this software, we chose random person in early stage then we chose some technical and some non-technical stakeholders.

D) Technique Selection

The technique we used is brainstorming. We used this technique on chosen clients/users who come for meeting with us. As we said in stakeholder phase, we gather some technical persons for brainstorming who will tell us about more features to add, about the feedback for this software. Then we chose some non-technical person

who don't have knowledge about software, we give training to them how to use this software and we analyze after that how much our software is friendly user. How much user can use easily, and we did through brainstorming of users.

E) Identify Requirements

After technique selection, we find the functional requirement of our software like performance of software, availability of software, scalability of software, capacity of software, reliability and maintenance of software. These are the functional features that identify in brainstorming technique that we done in previous section. Gather the requirements from client and user and tell the developer to implement these changes in our software and make more valuable.

F) Graphic Prototype

At this stage the clients are demonstrated an outline of the product. The model is made utilizing any realistic programming. This model outline fills in as the premise on which the improvement procedure starts. Subsequently, this stage is the essential stage for additionally process. On the off chance that the client favors the realistic model, at that point it goes to next stage. Something else, the necessities must be legitimately recognized once more. From now on changes are made in the model as per the given new prerequisites and in this way assessed by the client, so the procedure goes ahead till the client endorses it and continues it with the following stage.

engineering process. Then it can be passed onto other development stages like sprint planning, daily scrum, development, testing, deployment, review and release. In this manner these are the means which ought to be sought after to pick up an effective prerequisite distinguishing proof and refinement.

By the methods for this model, we can demonstrate that client contribution in each period of the advancement of the product is of vital significance as they guarantee the improvement of value arranged programming. After this, perform sprint planning. Sprint board will come and after this step, review the sprint and then finally release the product.

Thus, these are the steps which should be pursued to gain a successful requirement identification and refinement. By the means of this model, we can prove that user involvement in each phase of the development of the software is of paramount importance as they ensure the development of quality-oriented software.

VII. RESULTS AND DISCUSSION

In proposed framework, we discuss about scrumban framework for identification and refinement of requirements. Here discuss the results of survey that has been conducted for the evaluation of our proposed framework.

A) Evaluation Method

We have formulated a questionnaire in order to conduct the survey to validate the results of our proposed framework and also perform statistical analysis on our survey results.

This result will get after evaluation of different user that we gather in brainstorming techniques. Two different methodologies will apply in our software in sense of scope, budget, risk and quality of software and we identify the different mathematical evaluation because of we did brainstorming, did not perform implementation and did any other technique, so we calculate the result on online through software as we discuss on two methodologies.

We calculate the results online through software. We find out the mean and standard deviation of both methodologies to identify which methodology is better from one another as given in Table I.

Table I: Average and standard deviation values of scores for each factor, for both Scrum and Kanban

Factor	Scrum		Kanban	
	Mean	Std. Div.	Mean	Std. Div.
Schedule	3.67	1.20	4.05	0.85
Scope	3.60	1.08	3.68	1.02
Budget	3.87	1.12	3.79	0.92
Risk	3.83	0.99	3.93	0.94
Resources	3.88	1.13	3.93	0.86
Quality	3.98	0.96	4.02	0.87

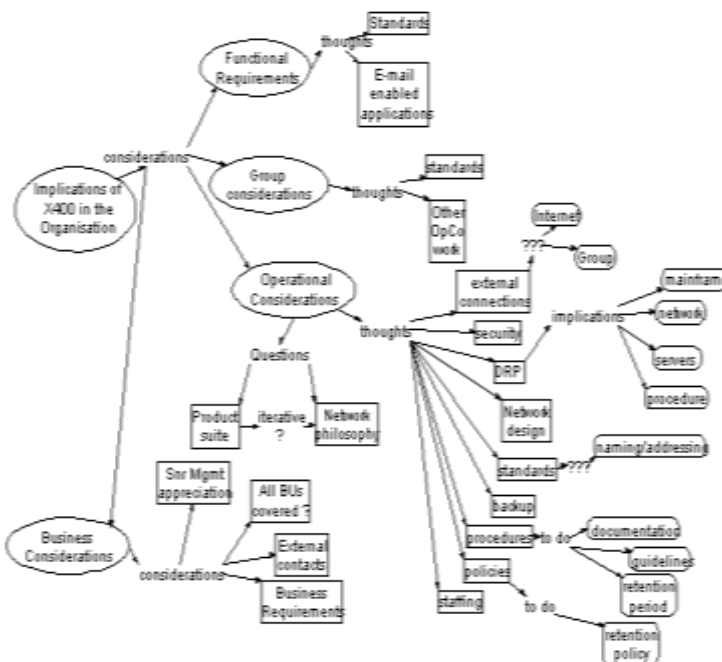


Fig. 3: Brainstorming of Software Prototype

Fig. 3 represents graphic prototype of our software. This outline of our software is how we make and create this software. What we get after brainstorming with clients and user, which steps need to do first, which feature are more important, and which are less.

G) Documentation

The identification stage ends with the evaluation of graphic prototype by the stakeholders. After requirements identification and refinement requirements are documented to complete the requirement

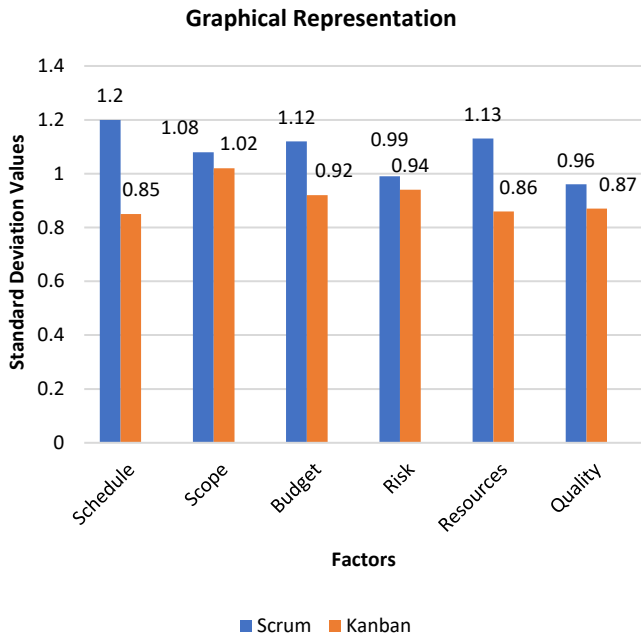


Fig. 4: Results of Survey

B) Results of Survey

In our research data from different gatherings were accumulated and a survey conducted to

evaluate the user’s feedback for the scrum process in which a total of 21 responses collected to evaluate the questions of the survey.

In the survey, question for each factor like schedule, scope, budget, risk, resources and quality were conducted. There were five categories were identified to check the customer response such as strongly agree, agree, neutral, disagree and strongly disagree. Survey questions related to each factor are given in Table II.

Table II: Summary Responses for Scrumban Users

Factor	Question (number)	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Ag. score	Response count
Schedule	Project teams are aware of project status (11)	4.8%	9.5%	23.8%	28.6%	33.3%	3.76	21
	Project teams can adapt changes quickly (12)	4.8%	14.3%	34.3%	38.1%	28.6%	3.71	21
	Project is delivered on time according to schedule (13)	9.5%	9.5%	28.6%	23.8%	28.6%	3.52	21
Scope	Project usually has well defined scope (21)	0.0%	28.6%	34.3%	42.9%	34.3%	3.43	21
	Project management methodology effective to make scope clearer (22)	4.8%	4.8%	28.6%	33.3%	28.6%	3.76	21
Budget	Project is delivered within budget (31)	9.5%	0.0%	33.3%	33.3%	23.8%	3.62	21
	Project provides good return on investment (32)	4.8%	0.0%	19.0%	33.3%	42.9%	4.10	21
Risk	Project risks and opportunities are managed (41)	4.8%	4.8%	28.6%	47.6%	34.3%	3.62	21
	Business objectives are met (42)	4.8%	0.0%	34.3%	47.6%	33.3%	4.05	21
Resources	Human/material resources are mostly available (51)	4.8%	14.3%	38.1%	23.8%	19.0%	3.38	21
	Teams work well together to achieve expected results (52)	4.8%	0.0%	0.0%	42.9%	52.4%	4.38	21
Quality	Quality requirements are met (61)	4.8%	0.0%	34.3%	52.4%	28.6%	4.00	21
	Client satisfaction is met (62)	4.8%	4.8%	19.0%	52.4%	19.0%	3.76	21
	Project is successful overall (63)	4.8%	0.0%	4.8%	52.4%	38.1%	4.19	21

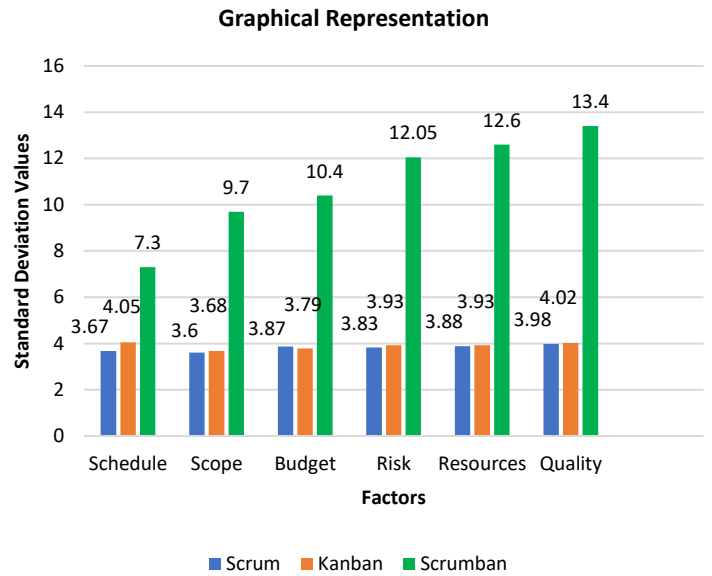


Fig. 5: Results of Survey after Applying Brainstorming Technique

Above results are related to the responses of user collected after the survey in which total number of responses of each category are mentioned in each column of the category. In the mean column we calculate the Mean of each value of our survey of each question and same for the standard deviation is calculated according to the responses of the users. Mean and Standard Deviation is calculated with the following formulas:

$$\bar{X} = \frac{\sum X}{N}, \quad \sigma = \sqrt{\frac{\sum (X - \bar{X})^2}{n - 1}}$$

We put the values calculated from the formula and get the values of mean and standard deviation which will be helpful in the statistical analysis.

VIII. COMPARISON

We compare the different existing frameworks with our proposed framework in terms of schedule, scope, budget, risk, resources and quality. Table III shows that our proposed scrumban is better than existing frameworks.

As could be anticipated, a tremendous piece of the bugs occurred at all beyond question knew parts, foul up recovery and code age. Undertaking Analysis for example in an undertaking to legitimize that the investigative iterative change figuring gives quantitative occurs as proposed.

According to our research questions which are identified in the first chapter our proposed scrumban approach has better results than scrum and kanban.

We did literature review, which is based on our research questions and identified different factors like schedule, scope, risk, budget, resources and quality. We also identified the drawbacks in the existing approaches of scrum and kanban. The values of standard deviation of the scrumban process is far better than the scrum and kanban

methodology. Scrumban is way better process for the software development according to the survey we done.

Table III: Comparison between Scrum, Kanban and Scrumban

Factor	Bin Han (2011)	Magnus Strale (2012)	Zahoor Ahmad Khan (2014)	Proposed Scrumban Model
Schedule	P	X	✓	✓
Scope	✓	✓	✓	✓
Budget	X	P	P	✓
Risk	X	P	X	P
Resources	✓	X	P	✓
Quality	✓	X	X	✓

IX. CONCLUSION

Kanban board inappropriate for complex advancement ventures, however much of the time it ought to be adjusted to serve particular needs of a specific group and additionally venture. In such cases it can be utilized as a managing reference containing a rundown of possible working states and standard procedures. In this area a genuine illustration is given of how a subset of proposed segments was utilized and how WIP limits were set in a test kanban venture that was led at the faculty of PC and Information Science. The undertaking required the advancement of an online apparatus for overseeing kanban ventures and endured 3.5 months.

The advancement group comprised of three graduate understudies of computer science, while the creator assumed the part of the product proprietor. Among the development we tried different things with two unique sheets and distinctive WIP limits. The board comprised of a subset of segments proposed "Build-up" and "Chose" segments relate to the "Item Backlog" and "Dash Backlog", separately. The WIP limits utmost reaches of the "Chose" segment was communicated regarding speed, i.e., the quantity of story focuses that the group was relied upon to finish in a sprint. At each sprint arranging meeting the relating stories were exchanged shape the "Build-up" segment to "Chose". The WIP limits utmost reaches of different segments were communicated as far as work things, showing the maximal number of client story cards in every segment.

The "Next" segment contained a predetermined number of high need stories as depicted. The WIP restrict was set to 3 in light of the fact that there were 3 designers chipping away at the venture. The undertaking was little and every engineer was expected to build up a client story from starting to end; along these lines, it appeared to be sensible to combine the "Investigation and Design", "Improvement", "Testing" and "Documentation" segments into a solitary "Improvement" segment.

This WIP confine gave off an impression of being as well low because of generous number of stories that were dismissed by the product owner among "Acknowledgment". These stories were returned back to the "Following" section and regarded as "silver shots"

having higher need than different stories. With a specific end goal to complete these stories as fast as could be expected under the circumstances and abbreviate the lead time, the WIP furthest reaches of the "Advancement" segment was expanded. Be that as it may, we imagine that in typical conditions an ideal esteem.

The "Convey" segment was discarded since the venture was test and did not have a genuine client. In this manner, the client stories acknowledged by the item owner moved specifically to the last "Done" section.

Most often, software development houses are reluctant to use new techniques for software development due to associated uncertainties. Current study is an attempt to offer an agile framework which better addresses the team collaboration methods. This study will propose scrumban based framework which addresses today's problems. Previously research studies attempted to provide scrum-based software's however they were not as effective as they should. Previous scrum-based software's [2] were difficult in sprint planning processes, low collaboration, and other relevant problems as mentioned in literature review. Therefore, this research is an attempt to propose an effective and efficient software which addresses such conventional problems.

REFERENCES

- [1]. Ahmad, M. O. (2013). Kanban in software development: A systematic literature review. In Software Engineering and Advanced Applications (SEAA), 39th EUROMICRO Conference on (pp.9-16).
- [2]. Anwar, S. A. (2014). Towards a Framework for Scrum Handover Process. PAKISTAN ACADEMY OF SCIENCES, 265.
- [3]. Banijamali, A. (2017). An Empirical Study on the Impact of Scrumban on Geographically. In Model-Driven Engineering and Software Development (MODELSWARD), 2016 4th International Conference on (pp. 567-577).
- [4]. Beydoun, G. (2014). Identification of ontologies to support information. Information Systems, 46,45-60.
- [5]. Boehm, B. W. (2013). A Spiral Model of Software Boehm, B. W. (1988). Computer, 21(5),61-72.
- [6]. Critical Analysys Of The Scrum Project, (2012).
- [7]. Casamayor, A. (2010). Identification of non-functional requirements in textual specifications. Information and Software Technology, 52(4), 436-445. <https://www.sciencedirect.com/science/article/abs/pii/S0950584909001918>
- [8]. DAVIS, A. M. (2009). A Strategy for Comparing Alternative Software.
- [9]. Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010).
- [10]. Ghazel, M. (2015). A pattern-based method for refining and formalizing informal specifications in critical control systems. Journal of Innovation in Digital Ecosystems, 2(1-2), 32-44.
- [11]. Han, B. (2012). Practical Experience: Adopt Agile Methodology Combined With Kanban.
- [12]. Hassler, E. (2016). Identification of SLR tool needs – results of a community workshop. Information and Software Technology, 70,122-129 <https://www.sciencedirect.com/science/article/abs/pii/S0950584915001779>
- [13]. Highsmith, J. (2002). Agile Software Development. (Vol. 13)
- [14]. Khan, Z. A. (2014). Scrumban - Adaptive Agile Development Process.
- [15]. KUHN, G. N. (2008). Performance evaluation of two-stage multi-product kanban system. IIE Transactions, 40(3),265-283.

- 07408170701499554s
- [16]. Kniberg, H. (2011). Lean from the trenches: Managing large-scale projects with Kanban. Pragmatic Bookshelf.
- [17]. Khan, Z. C., & Keet, C. M. (2015). An empirically-based framework for ontology modularisation. *Applied Ontology*, 10(3-4),171-195.
- [18]. Kumaresh, S. (2013). Managing requirement elicitation issues using step-wise refinement model.
- [19]. Kniberg, H. (2009). Kanban vs Scrum. *Crisp AB. Viitattu*, 1, 1-41.
- [20]. Lee, S. (2010). Distributed agile: project management. *Empirical Software Engineering*, 15(2),204-217.
- [21]. Lepak, D. P., Smith, K. G., & Taylor, M. S. (2007). Value creation and value capture: a multilevel perspective. *Academy of management review*, 32(1), 180-194.
- [22]. MAHNIC, V. (2013). Improving Software Development through Combination of Scrum and kanban. *Communications and Information Technology, Espanha*, 281-288.
- [23]. Mohammadi, N. G. (2016). Patterns for Identification of Trust Concerns. In *Proceedings of the 21st European Conference on Pattern Languages of Programs* (p. 31).
- [24]. Moitra, J. D. (2001). *Global Software*. 18(2),16-20. <https://ieeexplore.ieee.org/document/914732/>
- [25]. Nikitina, N. (2012). From Scrum to Scrumban: A Case Study of a Process Transition. In *Software and System Process (ICSSP), 2012 International Conference on* (pp. 140-149).
- [26]. Ogur, E. (2013). Refining Scheduling Policies with Genetic Algorithms. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation* (pp.1513-1518).
- [27]. Pries-Heje, L. (2011). Why Scrum works: A case study from an agile distributed project in Denmark and India. In *Agile Conference (AGILE), 2011* (pp. 20-28).
- [28]. Riemenschneider, C. K. (2002). Explaining Software Developer Acceptance. (12), 1135-1145.
- [29]. Rising, L. (2000). *The Scrum Software Development*. 17(4),26-32.
- [30]. Ruitenburg, R. R., & Braaksma, A. J. (2017). Evaluation of the Lifetime Impact Identification Analysis: Two tests in a changeable context. *CIRP journal of manufacturing science and technology*, 17, 42 -49.
- [31]. Schwaber, K. (2000). *SCRUM Development Process*. (pp. 117-134)
- [32]. Senapathi, M. (2017). Refining a model for sustained usage of agile methodologies. *Journal of Systems and Software*, 132, 298-316.
- [33]. Sidky, A. S. (2002). Problem identification and decomposition within the requirements generation process.
- [34]. Smits, H., & Pshigoda, G. (2007). Implementing scrum in a distributed software development organization. In *Agile Conference (AGILE), 2007* (pp. 371-375). IEEE.
- [35]. Vibhu. (2014). A Framework for Identifying and Analyzing Non-functional. In *Proceedings of the 4th International Workshop on Twin Peaks of Requirements and Architecture* (pp.1-8).
- [36]. Wohono, R. S. (2015). A Framework for Object Identification and Refinement in object-oriented analysis and design. In null (p.351).
- [37]. Yilmaz, M., & O'Connor, R. V. (2016). A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. *Tehnički vjesnik*, 23(1), 237-245.