CSE 498R

Project Report

A Systematic Review of Scrum in Software development

Team Number: 01

Group Members

Name	ID	Section
Md. Ibrahim Khalil Ullah	1812015042	15
Jahid Akand Nahid	1813142642	15
MD. Mehedi Hasan Bhuiyan Nipu	1911870642	24

Submission Date: 20 June, 2023

Department of Electrical and Computer Engineering

North South University

Spring 2023

Table of Contents—

***Abstract*—** Scrum is one of the frameworks used most frequently among agile software development approaches, which have grown significantly in popularity in recent years. Agile framework Scrum focuses on teamwork, adaptability, and ongoing improvement among development teams.

This study's systematic review will examine the applicability and efficiency of Scrum, an agile framework, in software development projects. The study's goal is to thoroughly review the body of research to evaluate the advantages, difficulties, and overall effects of Scrum in software development.

The research study starts by introducing the fundamental ideas and practices of Scrum, highlighting its emphasis on adaptability, ongoing progress, and self-organization among teams of developers. A systematic analysis of several primary studies, including empirical research, case studies, and industry reports, will then assess the adoption and results of Scrum.

The review highlights several advantages of using Scrum in software development projects. These include accelerated delivery of high-quality software solutions, excellent team communication, increased customer happiness, and improved productivity. Teams can adjust to shifting needs because of Scrum's iterative structure, which enhances responsiveness.

Scrum is also effective in reducing project risk. Thanks to frequent inspection and adaption techniques and regular feedback loops, teams can identify and address difficulties early on. By continuously evaluating and updating project goals, Scrum teams may more quickly react to changing customer needs or market conditions. However, the research also identifies several barriers to implementing Scrum. These problems include dealing with resistance to change, estimating project timelines, and needing skilled and devoted team members. The need for more research and attention on scalability and integration with larger companies is also underlined.

Overall, the systematic study highlights Scrum as a great agile technique for software development that, when used properly, offers a wide range of advantages. Before implementing Scrum, businesses are urged to analyze the unique circumstances and difficulties they confront carefully. They are also urged to give sufficient training and support to guarantee a successful implementation. Future study areas are also recommended to solve the problems and gaps, improving the knowledge and use of Scrum in software development.

Keywords— Scrum methodology, Agile, software development, team.

CHAPTER 1 INTRODUCTION

Background— ***This*** part begins with an introduction to the Scrum methodology, places it in the context of software development, and then provides a more detailed justification for the necessity of this review. The iterative, incremental Scrum project management methodology offers an easy "inspect and adapt" framework. Software is provided using the Scrum methodology in increments termed "Sprints" (often 2–4-week iterations). Planning and review are the two main components of each Sprint. A Scrum team's sprint planning is a time-boxed meeting that can take up to 4 hours. It is committed to growing. They were planned in detail for the Sprint. In sprint review meetings, project participants discuss the company's state, the market, and technology. These sessions may go on for up to four hours. A retrospective meeting may be held to evaluate the finished sprints' teamwork effectiveness. Each team member answers three questions during a 15-minute daily Scrum meeting: what did I do yesterday? What will I accomplish today, and what obstacles are in my path? Burndown charts, sprint backlogs, and product backlogs are the three artifacts produced by Scrum. Customer needs are listed in backlogs, and daily burndown charts display the work to be done.

Motivation—

The Scrum framework facilitates Agile software development, encouraging adaptability, teamwork, and incremental development. Bangladesh may have many software firms, but Scrum is only sometimes used as a development process. This talk will examine why Scrum is essential and how implementing it may completely transform Bangladesh's software sector.

Flexibility and Adaptability: Scrum provides a versatile and adaptable method for developing software. Teams can react swiftly to evolving needs, technological developments, and client feedback. This is essential in the modern, dynamic, and fast-paced software business. Companies may avoid inflexible and protracted development cycles by adopting Scrum and switching to incremental value delivery.

Collaboration and communication: Scrum promotes open communication among team members and cross-functional cooperation. It strongly emphasizes self-organizing teams that collaborate closely and eliminate silos between various departments and functions. Better

comprehension, alignment, and knowledge sharing are fostered by this collaborative approach, resulting in more excellent software solutions.

Continuous Improvement and Iterative Progress: Scrum encourages iterative development through time-boxed iterations known as "sprints." Each Sprint results in a functional product increment, allowing businesses to collect input and test theories early in development. Continuous improvement is possible with this iterative process, which makes it simpler to spot problems and fix them, increase the features of products, and guarantee consumer pleasure.

Accountability and Transparency: Through artifacts like the product backlog, sprint backlog, and burndown charts, Scrum offers a clear view of the development process. This openness allows stakeholders to monitor progress, spot obstacles, and make informed decisions. By designating roles like Scrum Master and Product Owner, Scrum also fosters accountability by ensuring that duties are clear and understood.

Let's now talk about how Scrum adoption could reshape Bangladesh's software sector:

Improved Productivity and Time-to-launch: Scrum's iterative structure and emphasis on collaboration and adaptation may significantly increase productivity and shorten the time it takes to launch software solutions. Companies may gain a competitive edge by adopting Scrum, delivering functional software more frequently, and reacting quickly to shifting market needs.

Increased Customer Satisfaction: Scrum emphasizes customer involvement throughout the development process. Companies may guarantee better customer satisfaction and forge deeper connections by consistently engaging with consumers, comprehending their needs, and providing incremental value. Positive referrals and recurring business are more likely to come from happy consumers.

Increased Creativity and Employee Engagement: Scrum's collaborative structure encourages team members to think beyond the box. It encourages people to offer ideas, try solutions, and own their work. Due to this high involvement and autonomy, the software sector may see a culture of innovation, improved job happiness, and the retention of top people.

Adapting to Global Market Standards: Leading software businesses worldwide have embraced Scrum. The software sector in Bangladesh may conform to the norms and procedures of the international market by adopting Scrum. This alignment creates possibilities for cooperation, alliances, and market expansion abroad.

Finally, there are several advantages to adopting Scrum in Bangladesh's software sector. It offers adaptability, teamwork, and iterative development, enabling businesses to produce high-quality software quickly. By adopting Scrum, the sector may transform its procedures, boost output, improve client happiness, promote innovation, and conform to international market norms. Although switching to Scrum necessitates dedication, training, and a cultural adjustment, it may be profitable for Bangladeshi businesses in the long run. (Hossain, 2009)

CHAPTER 2 RESEARCH LITERATURE REVIEW

Existing Research and Limitations—

Scrum has been the subject of previous study that has examined a number of its implementation, advantages, and drawbacks. Research has concentrated on determining what aspects affect Scrum's adoption and implementation in various situations and companies. This includes researching the difficulties encountered during implementation, techniques for dealing with opposition, and the effect on team relationships. Scrum's effects on team productivity and project success have been studied. While some studies contend that Scrum may speed up the time to market and increase productivity, others stress the significance of team dynamics and corporate support in obtaining successful results. Scrum places a strong emphasis on team cooperation and regular communication between members. These methods, including Scrum artifacts (like the product backlog and sprint backlog) and rituals (such as daily stand-ups and sprint reviews), have been studied for their efficacy in fostering efficient cooperation and information exchange. Scrum promotes active stakeholder participation throughout the development process. Research has looked at the function of stakeholders, their modes of participation, and difficulties sustaining productive interaction between development teams and stakeholders. Research has also examined the difficulties and solutions as businesses use Scrum on more significant projects or with several teams. This entails researching and analyzing the effects of frameworks like Scrum of Scrums, Nexus, and LeSS (Large-Scale Scrum) on team collaboration, communication, and alignment.

Proposed framework— Four variables were chosen to create the framework after thoroughly analyzing the essential aspects. The suggested framework eventually incorporates the chosen essential aspects, namely the environmental, organizational, technological, and human components.

i. **People factor:** People are the most critical component of any initiative or organization. People make decisions, work on initiatives, and decide if an organization's objectives are attained. Training and learning, societal culture, communication and negotiation, and personal characteristics are the four subfactors that make up this factor. (Kettunen, 2008) (Cho, 2008)

a. Personal characteristics: Communication abilities, honesty, a collaborative attitude, motivation, a desire to learn, and a feeling of responsibility are some personal qualities that may be measured. (Misra, 2009)

b. Negotiation and communication-
1) Individuals who operate in the same time zone can be used to measure communication and negotiation.
2) Work quickly and efficiently with management, operations, business units, customers, developers, and support. (Rola)
3) Communicate with people while being very motivated and believing.

c. Social culture
Social culture may be assessed by team members' progressive attitudes, general communicativeness, and shared social cultures. (Misra, 2009)

d. Education and training Instead of conventional training, professional-led negotiating and mentorship can determine a team's preparedness to teach new members and continually learn from one another. (Misra, 2009)

ii. **Technical factor:** Development and information management are two of these factors subfactors.

a. Development: The following are the main elements of development:

1) The team should design a plan for dealing with problems.
2) Issues at scrum meetings (Ozierańska, 2016)
3) Quick, iterative, and focused on people (Cho, 2008)
4) Sustainable development is emphasized consistently.
5)Organization.
6) The practices of software development methodology are processes, techniques, and simple design. (Dyba, 2008)

b. Information management: Information management involves extensive documentation built upon tacit knowledge management. (Ozierańska, 2016) (Cho, 2008) (Dyba, 2008)

iii. Organizational factors: Customer satisfaction, collaboration, commitment, decision time, corporate culture, team distribution and size, and planning and control are the nine sub-factors that make up the organizational factor.

a. Customer involvement: Customer engagement refers to customers participating in the project. The project will operate more smoothly if team members and clients get along well. Customer commitment enables teams to reduce the possibility of offering unsatisfactory solutions. (Misra, 2009)

b. Decision time: How to make critical decisions fast and within a limited time frame is discussed.

c. Team distribution refers to an organization's participation in globally dispersed initiatives impacted by the cultural and political climate. How geographically near other team members collaborating within or outside the company are located? Because local politics, behavioral patterns, circumstances, and cultures considerably impact the project team's efficiency, the geographic distribution and placement of the teams are important considerations.
(Misra, 2009)

d. Team size Team size is the number of people that make up a team, and it may vary greatly inflorescence on how well they get along with one another. It is advised to divide larger teams into smaller ones if more team members are involved in the project.

(Qumer, 2008) (Qumer, A framework to support the evaluation, adoption and improvement of agile methods in practice, 2008)

e. Corporate culture: Corporate culture describes how a company persuades consumers to provide quick feedback. A user-centric organizational culture is possible.

f. Regulate and planning: Control and planning examine the use of informal, unstructured, and undocumented tactics by software development teams, as well as the team's ability to regulate quality. (Misra, 2009)

iv. Environmental factor: This factor has three key aspects: team issues, everyday tool use, and customer participation. By attending to team members' needs and placing their faith in them to finish their work, an organization may establish a high-quality working environment for its employees. The developer should operate in a supportive and persistent atmosphere. The other team members' trust is necessary to achieve high confidence. Based on the context of the development environment, the organization should determine how much documentation is appropriate for each project.

(Cho, 2008) (Qumer, An evaluation of the degree of agility in six agile methods and its applicability for method engineering, 2008) (Qumer, A framework to support the evaluation, adoption and improvement of agile methods in practice, 2008)

a. Customer engagement: Customer involvement is vital to a product's scent. This section describes how consumers collaborate with developers to carry out tasks and are wholly involved in software development. Agile methodology states that customers should be involved in product development. Customers should be invited to participate in decision-making processes and develop excellent project execution strategies for organizations. (Cho, 2008)
b. Creating a supportive and helpful atmosphere for team members to do their jobs is called the "working environment." The Scrum methodology suggests an open office setting since it can make it easier for developers to work, aid in self-organization, and foster collaboration.
c. Team-specific tools and issues: Team-specific tools and issues are crucial to the environmental element. Team members can reduce the number of defects in software development by using tools. In a similar vein, teamwork can assist in resolving problems.
(Cho, 2008)

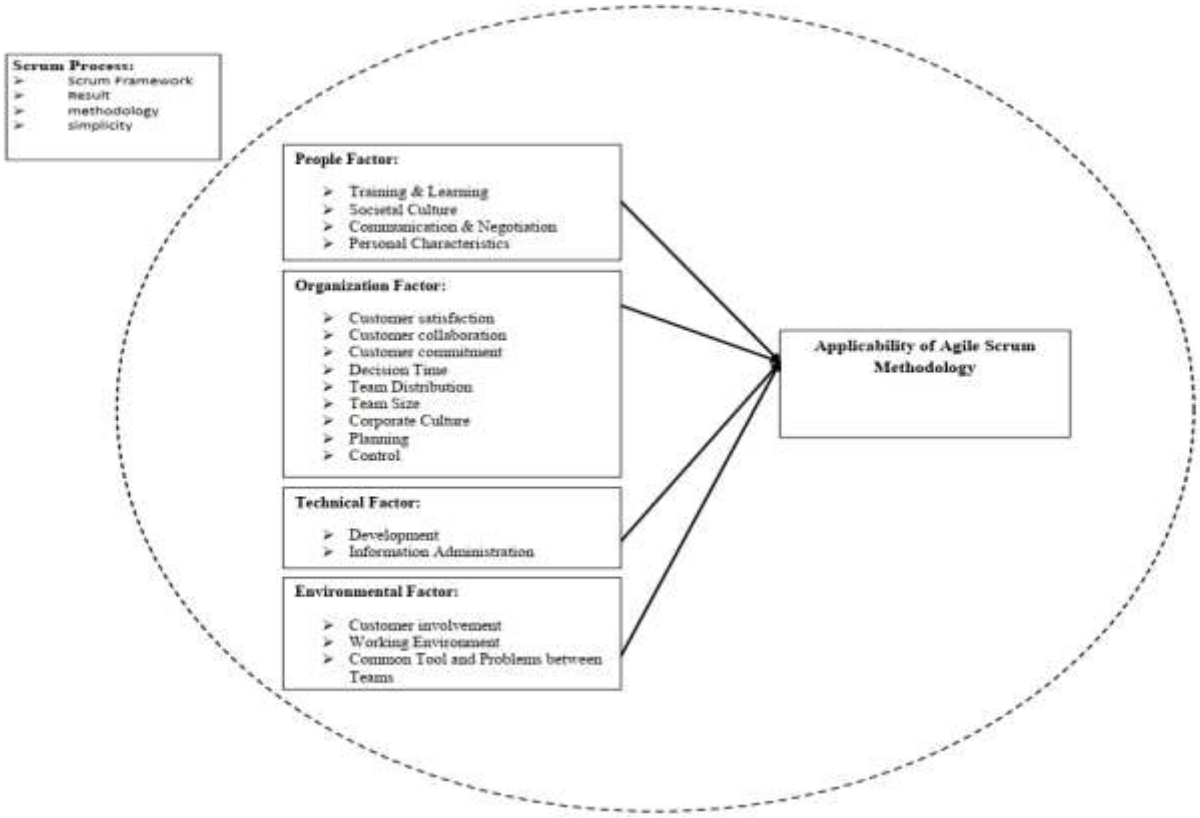


Fig: Framework for applicability of agile scrum methodology.

CHAPTER 3 METHODOLOGY

System Design—

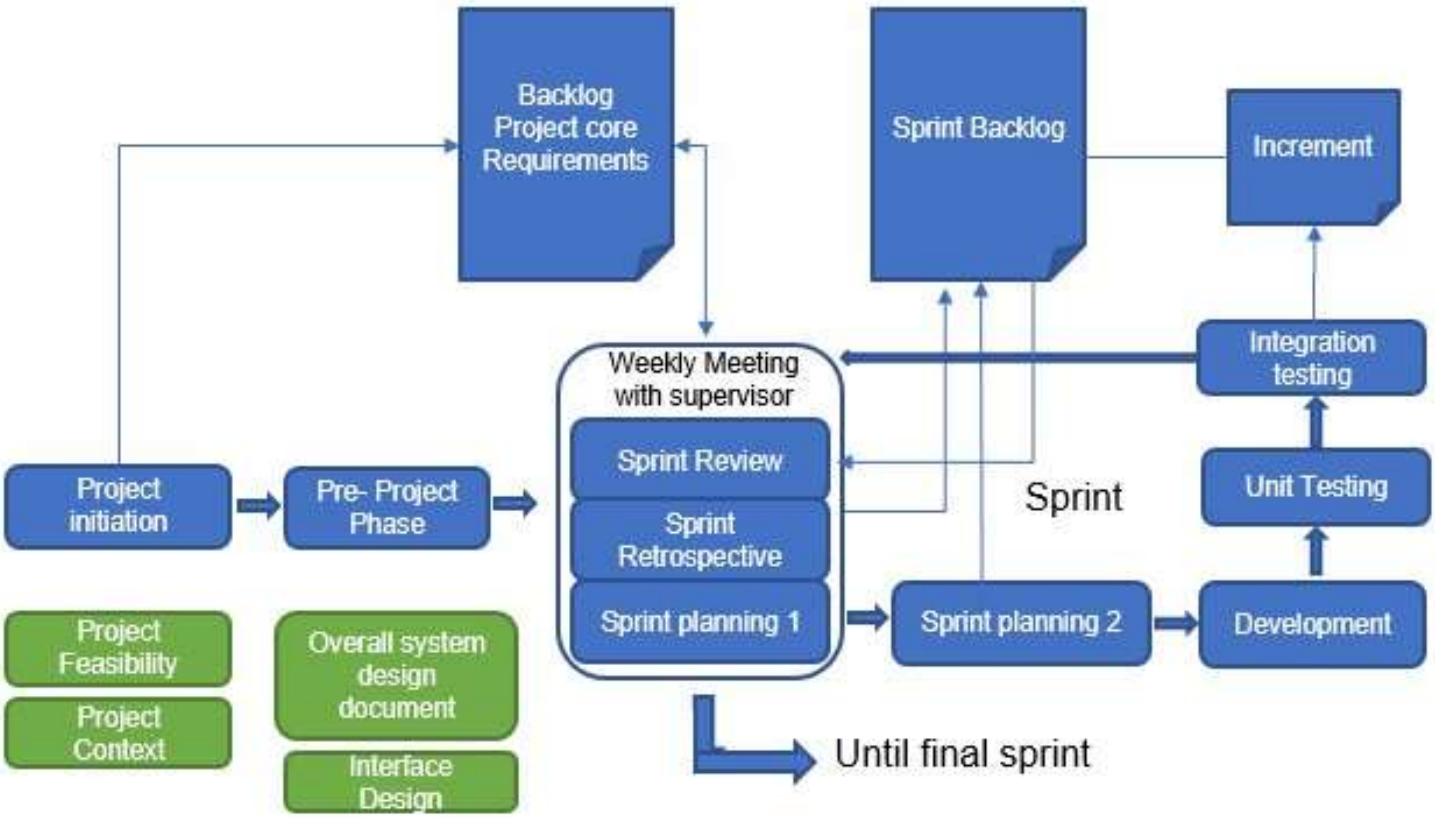


Fig: Visualization of the scrum process.

Although the Scrum methodology is frequently utilized in software development, it may also be employed in various other fields and projects. Small, cross-functional teams work in brief sprints, which are iterations that last one to four weeks, at the heart of Scrum. The Scrum Master, Development Team, and Product Owner all have crucial roles to play in the procedure. The product owner is responsible for building and maintaining the product backlog, generating a list of all requested changes and additions, and identifying the project's requirements. The Product Owner collaborates extensively with stakeholders to ensure the team provides value. The Scrum Master serves as the team's facilitator and coach, ensuring adherence to the Scrum rules and removing any barriers to advancement.

Additionally, they encourage the team to self-organize and support efficient communication and teamwork. The specialists in the development team are in charge of delivering the product

increment. These people often perform the tasks of developers, testers, designers, and other pertinent professionals. The team decides how to accomplish its objectives inside the Sprint because it is self-organizing.

The Scrum process follows a set of defined events or ceremonies. The most notable ones include:

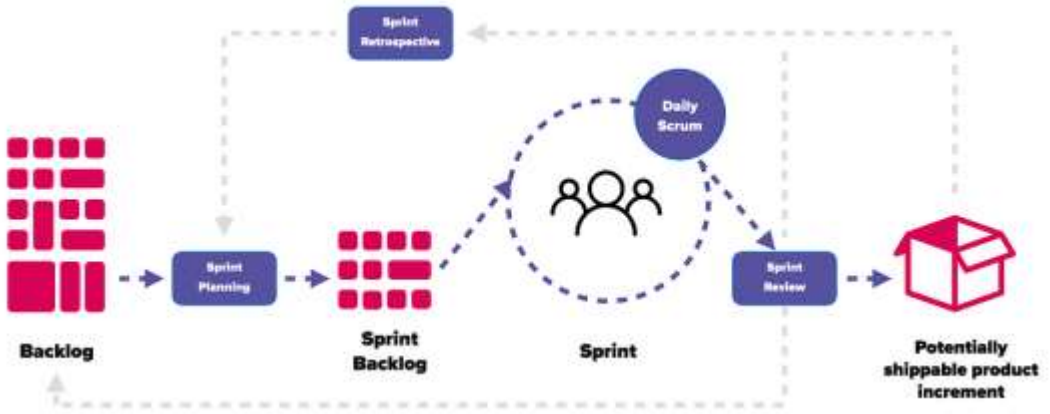
1. Sprint planning: The team and the product owner work together to create the sprint goal and choose which backlog items to focus on during the Sprint at the start of each iteration.

2. Daily Scrum: A quick daily meeting when team members coordinate their efforts, review progress, and spot any possible problems. The emphasis is on what has been completed, what is scheduled for the day, and what roadblocks stand in the way of advancement.

3. Sprint Review: The team shows the finished product to stakeholders and solicits input after each Sprint. The Product Owner evaluates the project's overall progress toward its objectives and may revise backlog items or priorities as necessary.

4. Sprint Retrospective: Following the Sprint Review, the team organizes a retrospective meeting to assess their progress, pinpoint areas for improvement, and develop concrete strategies to improve their procedure going forward.

Transparency is encouraged by the Scrum methodology through several artifacts. Some examples include the sprint backlog, a portion of the product backlog chosen for the current Sprint, and the increment—a possible shippable product version after each Sprint. Scrum's incremental and iterative processes let teams react more quickly to adjustments, client feedback, and new needs. It promotes teamwork, flexibility, and ongoing development. Scrum helps teams to provide value more often and maintain a high level of flexibility throughout the development process by breaking projects into manageable increments.



Scrum roles:

The three main roles in the scrum methodology are those of the development team, the product owner, and the scrum master.

Product owner: The primary responsibility of the product owner in the Scrum team is to collaborate with the user group to choose the features that will be included in the product release. They serve as the team's leader and guide the project's growth in the right direction. The winners are them.

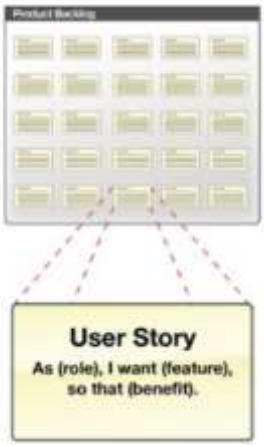
Scrum Master: The team's Scrum Master is in charge of promoting and assisting Scrum. They provide Scrum guidance to the team, product owner, and company while looking for ways to improve their methods. A competent scrum master collaborates with the team to increase transparency and delivery effectiveness while fully understanding the team's workflow.

Development Teams: To create product increments, cross-functional development teams are prepared. This team comprises developers, testers, designers, etc., to lessen dependency on the external party. Self-organizing scrum team members ensure that each Sprint is completed by breaking down the Product Backlog into units of potentially release-ready functionality. (Yerukala)

Scrum workflow Step:

1. **Product backlog Creation:** The first stage of the Scrum workflow process is the visioning phase when the stakeholders gather to select which features should be

implemented and create a product roadmap. The product owner is where the Scrum work process begins. The product owner creates user stories or project requirements with the team. A scrum team's user stories or product requirements are organized into a prioritized list called a product backlog. User stories featured in the Scrum methodology are written from the end user's viewpoint. The product owner decides which user stories or products are added to the product backlog. Release of the backlog comes next when the product backlog is created.

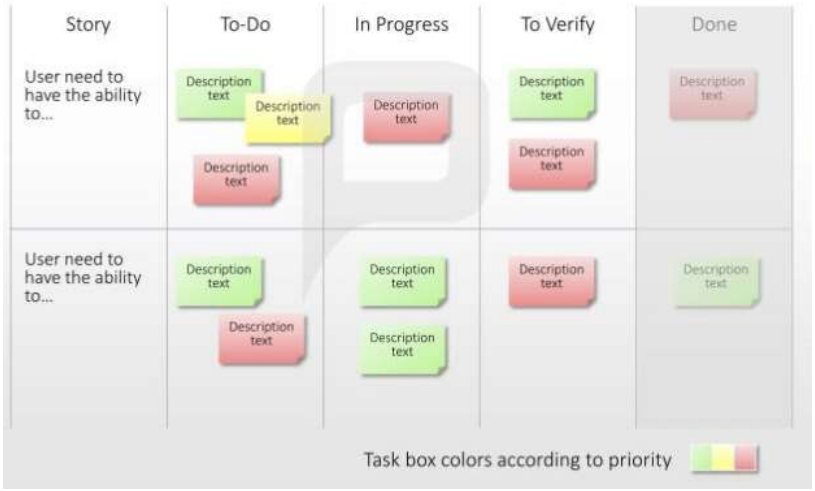


2. **Release backlog:** The team selects how to divide user stories into releases based on the product roadmap created with the product owner. Delivering a portion of the release backlog, often the product backlog, is the release's goal. After deciding which ones will be included in a particular release, the development team assigns a time estimate to each user story. The user stories are chosen for a sprint when the release planning is finished.

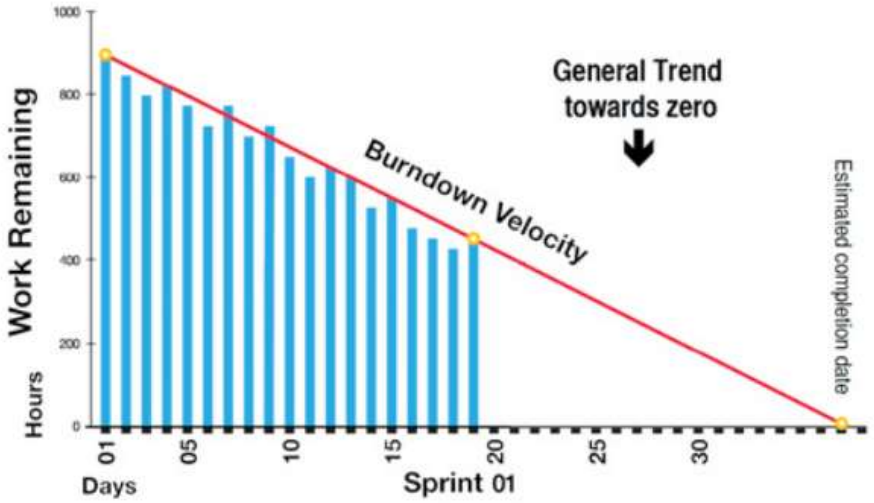
3. **Sprint Backlog Creation:** The team uses an assigned period known as a "Sprint" to complete several tasks from the Backlog. Each Sprint lasts between two and four weeks.

A controllable portion of the release backlog is completed in a ship-steady condition during each Sprint. A sprint backlog is a collection of product backlog items that must be completed during a single sprint iteration. After deciding on the sprint backlog, the team breaks down each user story into a task. After then, the product is created during each Sprint.

4. **Working on Sprint and Scrum Meetings:** The development process starts after the user stories for the current phase have been chosen. A task board, which contains individual user stories with a description of tasks required for implementation, is frequently used to track the current working process. Daily scrums are held once the code has been produced and incorporated into the system. Development teams do daily stand-ups or scrums to review the Sprint Goal and Sprint Backlog progress and to make adjustments to the plan for the remainder of the Sprint. Each meeting has a time limit of around 15 minutes. Obtaining accurate information about the current project state is the primary objective of these discussions. Everyone on the team must communicate with one another to let them know what duties they have completed, what jobs they wish to complete next, and any difficulties or obstacles they encountered while working. Update your sprint burndown chart after that.

5. **Burndown**  **Charts:** The team uses a burndown chart to track its progress. It provides a daily evaluation of the

work necessary to finish a certain sprint or release. By contrast, the number of hours worked with the initial project estimation, one may calculate the slope of the curve (burndown velocity), which displays the average productivity for each day. This graph makes it easier to calculate the current labor rate. In light of those conclusions, the number of user stories for the subsequent Sprint could change.



6. **Product demonstration and testing:** If every user story is finished, the sprint backlog will also be finished, signaling the Sprint's end. After the Sprint, a sprint review is performed, during which the customers are shown and allowed to approve the functional software. Based on their comments, stakeholders determine what additional adjustments should be made to the project.
7. **Planning the following Sprint retrospectively:** The team then conducts a sprint retrospective to assess the Sprint and identify any areas for improvement. The team specifically focuses on three things during a retrospective:
   What had success?
   What specifically went wrong?
   What needs to change?

Retrospectives might run up to 90 minutes in total. They aid in integrating ongoing changes into our sprint cadence and team. Finally, the team velocity is updated, providing information radiators to show the project's state and progress. These updates eventually return to the user stories, and the cycle is repeated until the project is finished. (Yerukala)

CHAPTER 4 RESULT, ANALYSIS, AND DISCUSSION

Results:

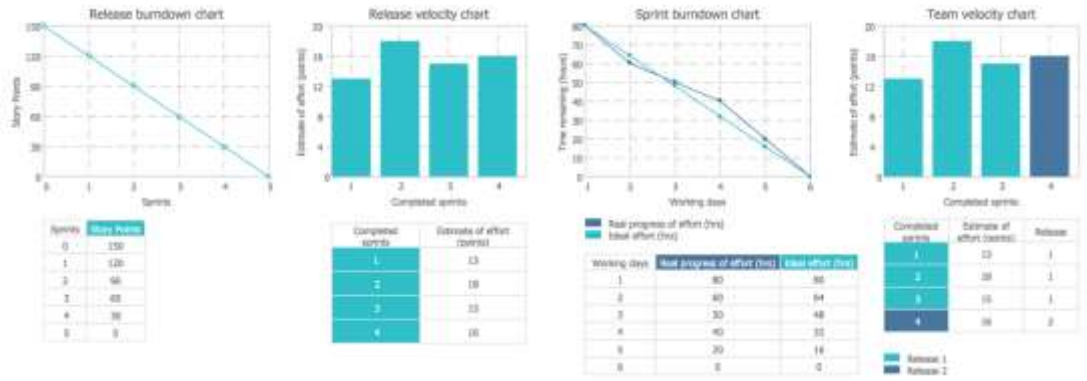


Fig: Result Analysis of Scrum in Software Development

The systematic review of Scrum in software development yielded overwhelmingly positive results, highlighting the numerous benefits and advantages of implementing Scrum methodology. The analysis of various studies and industry reports revealed that Scrum has consistently demonstrated positive outcomes in terms of team collaboration, project flexibility, and product quality. Teams utilizing Scrum reported improved communication and collaboration among team members, resulting in enhanced productivity and efficiency. Additionally, the iterative nature of Scrum facilitated better adaptability to changing project requirements, allowing teams to deliver high-quality products that meet client expectations. The findings also indicated that Scrum fostered a sense of ownership and empowerment among team members, leading to increased motivation and job satisfaction.

Key Benefits of Scrum Methodology in Software Development:

1. Agile approach
2. Improved project visibility
3. Customer Satisfaction
4. Increased Productivity
5. Risk Reduction
6. Continuous Improvement
7. Flexibility and adaptability

These advantages make Scrum an appealing software development technique, allowing teams to provide value more quickly, communicate successfully, and constantly improve their processes.

Scrum Limitations:

1. Scrum requires high-level, experienced personnel; without these individuals, there is a chance that the process will enlarge its scope.

2. Experience and dedication are necessary for a scrum team.

3. A less skilled scrum master might completely sabotage the development process.

4. The project may result in errors if the task is inadequately stated.

Issues in Scrum Implementation:

There are problems with management, development, and release processes that directly hurt Scrum adoption. Here is a thorough explanation of each.

1. **Quality Items Pileup:** Due to the agile nature of Scrum, teams must provide something promptly, regardless of the breadth of their Sprint Planning. As a result, teams may need to pay more attention to software quality and accumulate quality-related issues. This may also be relevant to actions taken to increase performance. This is the main problem with the Scrum Methodology and any Agile Methodology.

2. **Issues with Module Integration:** Because products are published often throughout a sprint, testing and quality assurance can only sometimes be completed on time for module integration. Critical flaws can readily arise when QA time for large or complicated systems is inadequate.

3. **Code Quality:** Because of their agility, teams often have short deadlines. The developer must put in extra time to make up for a missed deadline. Issues with code quality will result from this. However, it is impossible to always produce flawless code, especially when under time pressure.

4. **Lack of Scrum Training:** According to the results, 50% of the team members have no formal scrum training and need to become more familiar with the scrum methodology. They either learned this information via their slime masters or their teammates on the other teams.

5. **Release Process:** The release and deployment processes are another significant scrum problem. Scrum introduces agility into the workplace; sprint deployment is the main issue for every team.

(Kumar, 2020)

Chapter 5 Project Planning

Project Timeline	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
Product Backlog Refinement:	On going Process				
Sprint Planning	Typically, few hours				
Sprint	Two to four weeks				
Daily Scrum	Around 15 minutes daily				
Sprint Review	One to two hours				
Sprint Retrospective	One to two hours				
Final Product Launch					

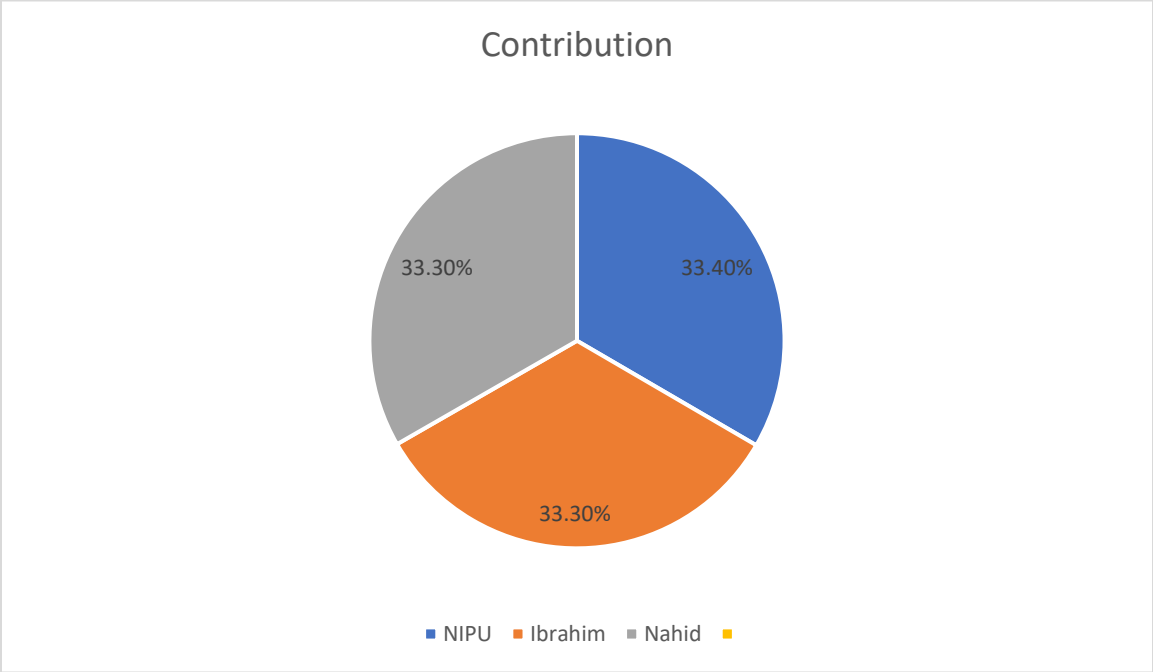
According to the Scrum methodology, these are the primary stages and times. It's vital to remember that the precise times might change based on the team's size, the project's difficulty, and other elements. Scrum places a strong emphasis on adaptation and flexibility. Therefore, the durations listed below are broad suggestions rather than rigid restrictions.

Lesson learned—

- **Clear and Regular Communication:** Scrum relies heavily on effective communication. Setting up a transparent culture and promoting open dialogue among team members, stakeholders, and the scrum master are essential. This ensures that everyone is on the same page, aware of the project's status, and capable of responding quickly to any problems.

- **Self-organizing and Cross-functional Teams**: Scrum strongly emphasizes empowered and cross-functional teams. It is crucial to provide team members the freedom to decide for themselves and to do excellent work.
- **Iterative and Incremental Development:** Scrum is an iterative methodology. By segmenting the project into smaller, more manageable pieces, it is possible to receive regular feedback, make adjustments, and make continual improvements.
- **Prioritization and Flexibility:** Scrum allows stakeholders to assign a priority to the product backlog, ensuring that the most important features are delivered first. The backlog might be reprioritized during the project to consider evolving requirements and market demands.
- **Continuous Improvement:** Through regular retrospectives, Scrum encourages a culture of continuous improvement. The team reviews its work, finds areas for improvement, and establishes concrete plans to improve its performance in the following sprint.
- **Sprint Planning:** A key component of Scrum is sprint planning. Setting time limits for meetings like daily stand-ups, sprint planning, reviews, and retrospectives may help attendees stay focused and maximize their time. To provide a disciplined and effective development environment, sprint planning entails goal definition, backlog item selection, work estimation, and creating a sprint backlog.
- **Effective Product Backlog Management:** The product roadmap comprises a constantly changing list of features, improvements, and defects. Effectively managing the product backlog requires ongoing item prioritization and refinement.
- **Dedicated and Empowered Scrum Master:** The Scrum Master is a critical player in assisting the Scrum process. Thus, they must be committed and empowered. The team may better grasp and adopt Scrum concepts, overcome obstacles, and create a productive work environment with the aid of a committed and empowered Scrum Master. The Scrum Master also acts as a coach and mentor for the team members while ensuring that the Scrum methodology is implemented appropriately.

Contribution of an individual team member—



Conclusion: The agile scrum technique is most frequently employed in the software industry, although there currently needs to be a foundation for this methodology in the literature. An application framework for the agile scrum technique has been presented based on four variables: organizational, technological, human, and environmental. These four components are further broken down into supporting variables. All of these elements serve as essential building pieces for the suggested structure.

References

1. Cho, J. (2008, January). An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum. Retrieved from https://www.researchgate.net/publication/43410168_An_Exploratory_Study_on_Issues_and_Challenges_of_Agile_Software_Development_with_Scrum
2. Dyba. (2008, 1 06). Empirical Studies of Agile Software Development: A Systematic Review. Information and Software Technology. doi:http://dx.doi.org/10.1016/j.infsof.2008.01.006
3. Hossain, E. M. (2009, July 13). Using Scrum in Global Software Development: A Systematic Literature Review. doi: 10.1109/ICGSE.2009.25
4. Kettunen, P. (2008, December 3). Adopting key lessons from agile manufacturing to agile software product development—A comparative study. *Volume 29*(,Issues 6–7), Pages 408-422. Retrieved ,June–July 2009, , from Adopting key lessons from agile manufacturing to agile software product development—A comparative study
5. Kumar, M. (2020). *Applicability of scrum methods in Software.* SSRN. Retrieved May 26, 2020, from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3610759
6. Misra, S. C. (2009, November). Identifying some important success factors in adopting agile software development practices. doi: 10.1016/j.jss.2009.05.052
7. Ozierańska, A. (2016, July). The Critical Factors of Scrum Implementation in IT Project - the Case Study. doi:10.22367/jem.2016.25.06
8. Qumer, A. (2008, November). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Volume 81*(, Issue 11), 1899-1919. doi:https://doi.org/10.1016/j.jss.2007.12.806
9. Qumer, A. (2008, March). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Volume 50*(Issue 4), 280-295. doi:https://doi.org/10.1016/j.infsof.2007.02.002
10. Rola, P. (n.d.). Conceptual model of working space for Agile (Scrum) project team. *Volume 118*, 49-63. doi:https://doi.org/10.1016/j.jss.2016.04.071
11. Yerukala, M. (n.d.). *Scrum Workflow.* Retrieved from https://mindmajix.com/scrum-workflow