Agile Software Process

Q.1 Set. (A) What are the three main characteristics of agile software development?
In What ways does agile software development differ from other software development approaches?

Ans: If it's decided that agile is the most appropriate development methodology to use, then the three key things that will enable the project to be a success are:
a). **Collaboration**
b). Constant focus on business value
c). and Appropriate level of quality.

- **Early delivery of business value**
- **Continuous improvement** of the project's product and processes
- Scope **flexibility**
- **Team input**
- **Delivering** well-tested products **frequently** that reflect customer needs.

Characteristics of agile development success

- Agile development releases and fixed-length iterations.
- Agile development delivers working, tested software.
- Value-driven development.
- Continuous (adaptive) planning.
- Multi-level planning in agile development.
- Relative estimation.
- Emergent feature discovery.
- Continuous testing.

A host of approaches have agile characteristics. Three, however, are common to many agile projects: lean product development, scrum, and extreme programming (XP). These three approaches work perfectly together and share many common elements, although they use different terminology or have a slightly different focus. Broadly, lean and scrum focus on structure. Extreme programming does that, too, but is more prescriptive about development practices, focusing more on technical design, coding, testing, and integration. (From an approach called extreme programming, this type of focus is to be expected.)

In What ways does agile software development differ from other software development approaches?

Ans: The key difference between heavyweight and agile methodologies is the **adaptability factor**. In an agile methodology if any major change is required, the team doesn't freeze its work process; rather it determines how to better handle changes that occur throughout the project.

The agile model **allows to change the requirements after the development process starts, so it is more flexible**. The waterfall model is rigid, it does not allow to change requirements after the development process starts

S. No.	Traditional Software Development	Agile Software Development
1.	It is used to develop simple software.	It is used to develop complicated software.
2.	In this methodology, testing is done once the development phase is completed.	In this methodology, testing and development processes are performed concurrently.
3.	It follows a linear organization structure.	It follows an iterative organizational structure.
4.	It provides less security.	It provides high security.
5.	Client involvement is less as compared to Agile development.	Client involvement is high as compared to traditional software development.
6.	It provides less functionality in the software.	It provides all the functionality needed by the users.
7.	It supports a fixed development model.	It supports a changeable development model.
8.	It is used by freshers.	It is used by professionals.
9.	Development cost is less using this methodology.	Development cost is high using this methodology.
10.	It majorly consists of five phases.	It consists of only three phases.
11.	It is less used by software development firms.	It is normally used by software development firms.
12.	Expectation is favoured in the traditional model.	Adaptability is favoured in the agile methodology.

Q.1 Set. (B) When did the agile approach emerge?

What was the background of its emergence and evolution?

Ans:

background of its emergence:

In 1986, Hirotaka Takeuchi and Ikujiro Nonaka published an article called **"New New Product Development Game" in the Harvard Business Review.**

Agile was formally launched in **2001**, when a group of technologists (17) drafted the Agile Manifesto. They wrote four major principles for agile project management, intended to guide teams on developing better software:
Individuals and interactions over processes and tools.
Working software over comprehensive documentation.
With Agile software development, **teams can quickly adapt to requirements changes without negatively impacting release dates**. Not only that, but Agile helps also reduce technical debt, improve customer satisfaction, and deliver a higher quality product.

Q.1 Set. (C) What could happen if a bug not planned for arises in development using a traditional methodology?

Ans: Maintenance: Of course, the nature of any software development project is that, through use by customers, bugs will arise and must be squashed. So, past the verification stage, it's typically expected that you will provide maintenance beyond launch. This is an ongoing, post-launch phase that extends for as long as your contract dictates.

Q.2 Set. (A) Map the following Typical tasks / challenges mapped to Cynefin domain: [4]

1. **No release deadline ---** Chaotic, Disorder
2. **Monitoring time spent in phase** – Clear, Simple
3. **Ambitious timeline -** Chaotic
4. **Checking for coding standard** – Clear, Simple
5. **Project scope too large -** Complex
6. **Task estimation -** Clear
7. **Retrospectives without consequence -** Complicated
8. **Fixing a Build** – Clear, Simple

Q.2 Set. (B) Map the following Typical tasks / challenges mapped to Cynefin domain: [4]

1. Reproducing a clearly defined widget - Simple
2. Sentiment analysis and product rating - Complex
3. There is a fire in the theatre, and we need to get out- Chaotic
4. Choosing the wrong tools and taking the wrong actions. - Complicated
5. Building a house - Simple, Complicated
6. Changing requirements - Complicated
7. Knowing when a task is done Page 1 - Simple
8. Lack of trust- Chaotic, Disorder

How to make the best decision depending on the context in which it is taken:

- **Chaotic**: Act, Sense, Respond.
- **Complex**: Probe, Sense, Respond.
- **Complicated**: Sense, Analyze, Respond.
- **Chaotic**: Sense, Categorize, Respond.

If there are no uncertainties, if everything is Clear, there is a "Best Practice" that can be applied.

Q.2 Set. (C) The causality of issues that may develop during the operation of IOT (Internet of Things) systems is mapped onto the Cynefin framework. Based on the examples below, classify IoT systems into the relevant Cynefin domain. [4]

1. Causality becomes unpredictable - Complex

2. Causality is straightforward - **Simple**

3. Causality of issues of the IoT is analyzed based on Artificial Intelligence or machine learning algorithm - Complicated

4. The System or Parts of the system fail with potential increasing damage to the environment it is operating in.- Chaotic

Q.3 Set. (A) Explain how each of the Agile practices listed below supports the following Agile Software Development perspectives (A,B,C). [4]

A. Human: Cognition and social aspects

B. Organizational: managerial, cultural and process considerations

C. Technological: Practical, technical and product considerations

Agile Practices:

1. Whole team

2. Short releases

3. Time estimations

4. Measures

5. Customer collaboration

6. Test-driven development

7. Pair programming 8. Refactoring

A:

Agile Practices:

1. **Whole team – A, B, C**
2. **Short releases – B**
3. **Time estimations – A, B**
4. **Measures – A, B**
5. **Customer collaboration – A, B**
6. **Test-driven development - C**
7. **Pair programming - T**
8. **Refactoring - T**

Use the methods of systematic literature review to complete, collect and review previous related work to find the human factors to affect team performance.

It is essential to study the impact of the individual based factors on **Whole team** performance and several factors can be indicated that show their effect on the performance of a team. Essentially, the study of the factors can be used **to build processes** and teams focused to better their performance which can ultimately lead to better software's. Individuals contribute to the team and studying the factors that rely on their characteristics helps understand the team as a whole in a better way which can in-turn aid in improvement. Keywords: **Individual Human factors, team performance, agile development, personality, high-performing teams**

B: Cultural differences are also an important part of human factors.
The attributes refer to factors such as team climate, team diversity, communications, resources provided etc., t
Collaboration, Communication and Trust are one of the most essential factors. The results have the factors classified into (i) team, (ii) project success, (iii) software development, (iv) requirements, (v) agile practices, (vi) decision-making and (vii) client involvement factors

C: we believe that the principle mentioned in the agile principle (Continuous attention to technical excellence and good design enhances agility) corresponds exactly to these three factors

Q.3 Set. (B) What exactly is a self-organizing team? In what ways does self-organization change the day-to-day life of a developer? Give three explanations. [1+3]

Ans: At the simplest level, a self-organizing team is **one that does not depend on or wait for a manager to assign work**. Instead, these teams find their own work and manage the associated responsibilities and timelines.

Key characteristics of a self-organized team

- Teamwork and collaboration. When there is no manager to bring the team together and push orders, it is up to the members to communicate effectively and work with each other.
- Competency.
- Continuous improvement and growth: ...
- Respect and trust in the team. ...
- Ownership.

Take for example **a bank that tried to implement self-organization at some level in one of its departments**. At first, things went smoothly, and the team was very productive. But then the top management felt that they were losing control and decided to go back to a traditional team management approach.

Q.3 Set. (C) Agile Principles P1: Satisfy Customer, P2: Welcome Change, P3: Deliver Frequently, P4: Collaborate Daily, P5: Motivated Individuals and Trust, P6: Face to Face conversation, P7: Working Software, P8: Sustainable Development, P9: Continuous Attention; P10: Simplicity, P11: Self-Organizing, P12: Reflect and Adjust

Q.1.1 Explain how each Agile practice supports one or more Agile Values, Empirical Control Process and Agile principles for each of the Agile practices listed below. Write down the matching Agile value and Empirical control process; however, you do not need to write down the entire principle for Agile principles; instead, use the above labels P1 to P12 to signify the 12 Agile principles. [4]

1. Whole team – P1, P2, P10

2. Short releases - P3

3. Time estimations – P8

4. Measures Page 2 - P12, P10

5. Customer collaboration – P6

6. Test-driven development – P4, P7, P8

7. Pair programming – P8, P4,

8. Refactoring – P9, P12

Q.4 Set. (A) Specify the key issues you face based on your current experience with software development. Will following the Agile techniques help you solve your problem? [2]

Ans: **The continuous nature of agile development processes raises a few serious testing challenges:**

- Changing Requirements.
- Not Enough Information.
- Continuous Testing.

- Technical Skills.
- Frequent Regression Cycles.
- Lack of Communication.
- No Quality Measurement.

An agile method **relies upon incremental and iterative completion of goals with a self-managing team**. It is often presented in opposition to a “waterfall” process that sequentially gathers requirements, completes a design, and then builds a final product.

The primary benefits of Agile, including:

1. Increased visibility.
2. Increased adaptability (agility)
3. Increased alignment.
4. Increased product quality.
5. Increased business value.
6. Increased customer satisfaction.
7. Decreased risk

Better Quality: Because it is iterative, one big benefit of agile methodology is the ability to find problems and create solutions quickly and efficiently.

Q.4 Set. (B) Given one reason why some software products, such as VisiCalc, which was very popular when it was first released, did not survive in the market? [2]

Ans: Sales declined so rapidly, that the company was soon insolvent. **Lotus Development purchased the company in 1985, and immediately ended sales of VisiCalc and the company's other products**.

VisiCalc has now been discontinued. VisiCalc sales nearly vanished when Lotus 1-2-3 was released in 1983, taking full use of the PC's taking full advantage of the enhanced memory and screen. The company's sales decreased so quickly that it went bankrupt. In 1985, Lotus Development bought the company and immediately stopped selling VisiCalc and the rest of the company's products.

Bricklin never received a patent for VisiCalc because **software programs were not eligible for patents by the Supreme Court until after 1981**.

The popular Lotus 1-2-3 spreadsheet was introduced in 1983. Mitch Kapor founded Lotus and used his previous programming experience with VisiCalc to create 1-2-3.

Q.4 Set. (C) In the 2007 "State of Agile Development" survey1 conducted by VersionOne and the Agile Alliance, 60% of respondents predicted a 25% or greater improvement in time-to-market. Explain how Agile practices could have aided in the reduction of time-to-market. [2]

Ans: Become "Well-Oiled" First

One way to look at software development is to see it as problem solving for business. When considering a problem to solve there are two fundamental actions that must be taken:

- Solving the right problem. This is IT/Business alignment.
- Solving the problem right. This is technical expertise.

What Agile Practices Improve Time to Market?

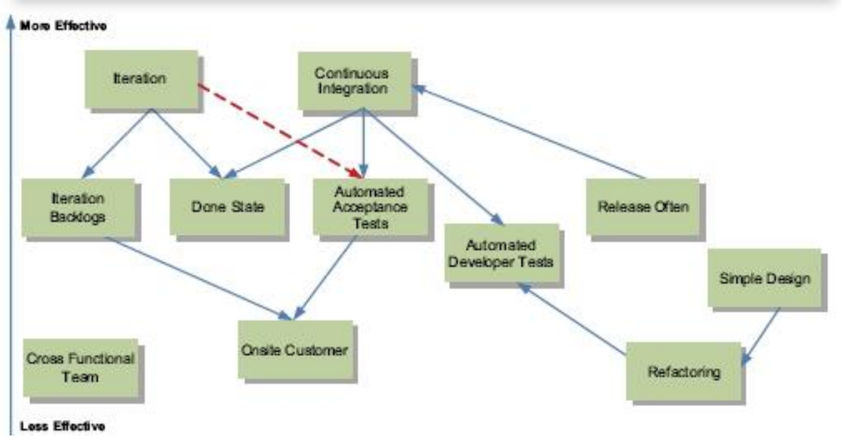


Figure 1-These are the Agile practices that improve time to market. The most effective practices are near the top of the diagram. Therefore, iteration is more effective than Onsite Customer for improving time to market. The arrows indicate dependencies. Continuous Integration depends on Automated Developer Tests for it to be effective.

Iteration: An iteration is a time-boxed event that is anywhere between 1 to 4 weeks long. The development staff works throughout this period – without interruption – to build an agreed upon set of requirements that are accepted by the customer and meet an agreed upon "done state".

Continuous Integration: Release your software to your end customers as often as you can without inconveniencing them.

Iteration Backlog

Automated Acceptance Tests

Onsite Customer

Simple Design

Refactoring

Cross-Functional Team

How to Adopt Agile Practices Successfully

1. Minimize What You Build
2. Cross-Functional Team
3. Learning is the Bottleneck
4. Know What You Don't Know
5. Choosing a Practice to Adopt

Q.5 Set. (A) Some of the time-related issues associated with software development are as follows:

1. Bottlenecks. 2. Project planning and schedule. 3. Time estimation.

4. Time pressure. 5. Late delivery.

For each of the above problems, specify how agile software development methods attempt to overcome it in general and, in particular, what agile practice(s) aims at solving it. [4]

Ans:

Bottlenecks: Start with communication, maximize your Scrum Master role, Refresh your Scrum values.

Project planning and schedule: Sprint planning, Enhance your Product Owner skills. Agile planning is **a project management style with an incremental, iterative approach**.

Time estimation: Improve your Backlog process, Create a compelling product vision

Late delivery: Refresh your Scrum values, Improve your Backlog process

Q.5 Set. (B) Consider the years when the Internet was just getting started and many businesses were getting online. Banks are beginning to consider if they could provide online access to their business customers. One bank considers two strategies.

(A) The bank's IT director suggests that they put together a quick and dirty Web site that allows customers to submit transactions through their browser, to get this up and running, and to try to develop a connection with the "Back-office" legacy mainframe database system.

(B) The bank also gets a report from some outside consultants that suggests they should reengineer the legacy back-end and build an integrated Web frontend to provide a powerful, user-friendly e-banking system engineered to a high standard. What are the advantages and disadvantages of each strategy? Which strategy would be best at the time, and why? [4]

Answer: The IT **director is right** but probably for the wrong reasons. His priority is to protect his department, he wants to be involved in the software development, He hasn't the resources for a big new development, but he could do something that would work up to a point. He doesn't want external consultants taking over and saddening him. He doesn't yet know how to connect the Web front-end to the transaction system but is confident that they can fix something.

In this case, the right reasoning is based around business pragmatics. What the bank actually did was to get the Internet site and front-end working quickly and to commission it. At the time there was no connection with the legacy transaction system, so a large number of typists were employed who printed off the Internet transactions and typed the details into the mainframe system. This was initially expensive but effective. This bank grabbed 70% market share of this new business. This then generated the income to allow them to build an integrated system involving a newly engineered back office produced by their consultants.

Meanwhile, the rival banks were building integrated solutions before going live; they never ever captured market share from the first bank.

Our bank demonstrated agility in both its business planning and in IT deployment. Using people for the data entry instead of a software alternative also demonstrated a low-risk approach as the alternative was, at the time, an untried technology. This illustrates that business considerations as well as technical ones are vital; neither should dominate the other. In an ideal world, the business leaders should have a deep and realistic knowledge of IT, and the IT specialists should have an informed and pragmatic understanding of business realities and the fundamental need to be able to make money, to maximize market share, and to deliver quality products and services. Agile methodologies need to be compatible with these objectives.

Q.5 Set. (C) As an Agile Consultant, how do you deal with or plan for these Project situations? [2+2]

Q.1.1 Your team is in trouble. The client has not been in touch with her feedback on the proposed system. She doesn't have much IT experience and only has a rather vague idea of what she wants. There are no similar systems known to you that you can show

her. You need to start getting some requirements identified and some initial stories prepared.

Answer: Build a simple prototype using your imagination and background research in order to show her something that might stimulate her ideas. The delay may be caused by the client not knowing how to proceed and lacking in confidence; it may be their first experience as a client, 11.5 Conundrums: Discussion 273 and their knowledge of IT is minimal. Some good suggestions from you could be a lifesaver for them. Even if they don't like your ideas, these may stimulate them into suggesting things that are suitable. It also gives you an opportunity to be creative instead of just sitting around waiting.

Q.1.2 Your project includes domain experts and technologists, as well as programming in a language that only one member of your team is familiar with. Three others have a rudimentary understanding of the language but have never written anything significant in it.

Answer: First we need to review the objectives of the project. Sure, we all want to deliver a great system for the client. But we also want to learn more about programming, particularly in this new language, we want to learn how to work in an XP team, and we want to learn how to manage a real project and work with our client. The team needs to discuss all of these things in a rational and calm manner.

Let's look at a possible way forward. We need to look at the project plan. It will contain a number of ongoing tasks, liaising with the client, writing code, working on stories, and so on. We need to include among these the need to learn the new language. Schedule some sessions where the expert gives a tutorial to the others. Of course, this may seem like wasted time because no productive code is being generated, but the benefits will come later. The expert should identify with the others small pieces of code that they can produce in pairs. Meanwhile the expert looks at some other issues such as story analysis and the definition of both functional tests and unit tests. While this is being done, the others are getting up to speed a little. After 2 or 3 weeks, if everyone works hard and with a positive attitude, we might get to a position whereby any pair can program together in a reasonably effective way and they will get better all the time.

Q.6 Set. (A) The original system is very poorly written, has slow performance, a poorly structured database, code that is all over the place, and will be a nightmare to maintain. There is a pressing need to create a new system, and you must persuade the business team to fund it. Should you carefully document the system's functionality and begin reengineering it before adding new functionality? What should your strategy be? [2]

Answer: Your client may have a very good reason for wanting something quick; there may be more business benefit in doing so. Our scenario relates to the business person's need to be able to demonstrate a piece of working software to the business backers who will decide on putting further money into the project. Showing something working, even if it did not have all the functionality required or was a little unreliable, to these backers was much more important than doing a good software engineering job. If the extra funding becomes available, then the proper engineering of a reliable and maintainable system would probably become a priority. This also illustrates that the link between the business 272 Chapter 11

Reflecting on the Process context and the software development process is fundamental. Traditional software engineering textbooks discuss approaches that are dominated by technical issues and the pursuit of quality without looking at how the real business pressures can force the way things have to be done. Although XP provides a number of practices that can guide us toward building high-quality and relevant software solutions, they shouldn't prevent us to respond to real business needs in favour of some abstract notion of how things should be done properly.

Q.6 Set. (B) If someone in your team put forth one of the following claims regarding refactoring, how would you answer him or her? [2]

1. **''I'm paid to write new, revenue-generating features.''**
2. **''Refactoring is an overhead activity.''**
3. **''It is hard for me to see the benefits of refactoring.''**

Answer: My experience suggests that as refactoring becomes part of a routine, it stops feeling like overhead. This statement is easy to make but difficult to substantiate. To the skeptics among you, my advice is just do it, then decide for yourself. Give it time, though.

Q.6 Set. (C) As an Agile Consultant, how do you deal with or plan for these Project situations? An e-commerce system was commissioned by a local retailer. This was successfully completed and installed. One of the sales assistants was assigned the task of printing out Internet orders and processing them through the orders system at the end of their shift. This worked well at first, as the number of orders increased gradually. After a few months of steady growth, the sales figures for orders placed via the Web abruptly dropped.

Answer: Initially, it was thought that the volume of orders was too great for the database system chosen for the application. This hypothesis was soon rejected. Other thoughts focused on the architecture of the system and on the connection between the interface, business, and database layers. Again, no problems were uncovered here. We then looked more carefully at how the system was operating in the business. The reason for the problem was found to be because the increase in the number of orders was not accompanied by a corresponding increase in the people dealing with the orders. The assistant became increasingly frustrated at the volume of work that had to be done at the end of their shift. The desperate solution taken was to delete all the Internet orders instead of dealing with them. This indicates that the system must be designed to include the human dimension as well as the computer. A management strategy should have been designed alongside the introduction of the computer system so that it could adapt to the needs of the business as these changed

Q.7 Set. (A)

Q.1.1 During an estimating meeting five programmers are estimating a story. Individually they estimate the story at two, four and five story points. Which estimate should they use? [2]

Ans: This step often surfaces a different understanding of the scope or implementation, which can then be clarified and agreed. The person that gave a 2 may know an easy solution, but the person that gave a 5 may anticipate a difficultly no one else thought of.

Repeat the estimation process until a consensus is reached. Or sufficient convergence is achieved as per the development teams' own rules.

Estimation Meeting

The estimation meeting proceeds as follows:

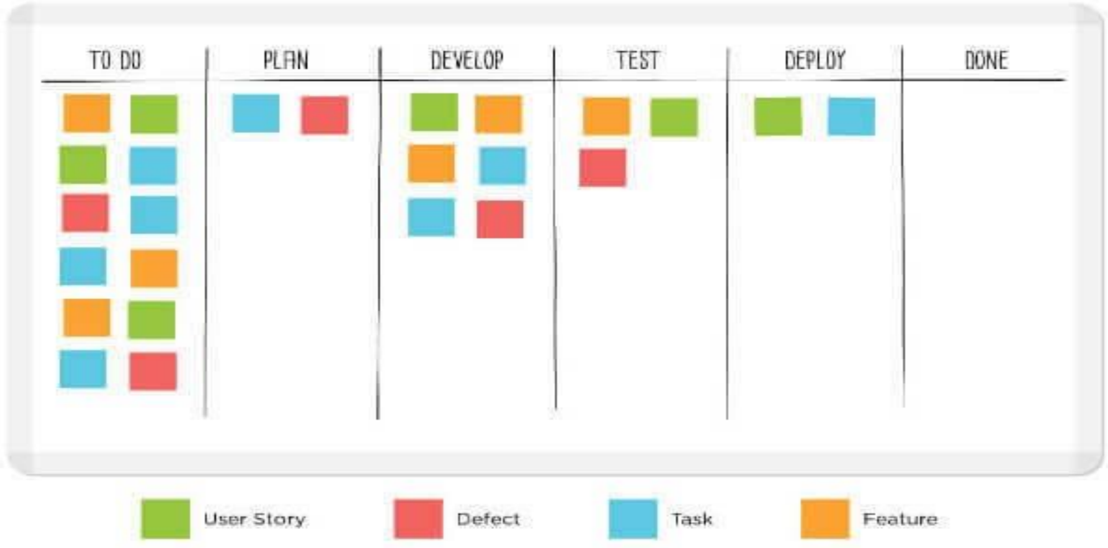
1. The Scrum Master, who will not play, chairs the meeting.
2. The Product Owner provides a short overview of one product backlog item (PBI) to be estimated. The development team asks questions and discuss to clarify assumptions and risks. A summary of the discussion is recorded by the development team.
3. Each member of the development team compares the size of the PBI relative to the calibration PBIs and chooses their size estimate. During the discussion, numbers must not be mentioned at all in relation to feature size to avoid anchoring.
4. Everyone calls their estimates simultaneously. Only the development team may estimate. The product owner is permitted to enquire as to why something is a certain size and may enter a negotiation on the scope which may affect the size of an item.
5. People with high estimates and low estimates offer their rationale for their estimate, and then discussion continues. During the debate, assumptions should be aired, and the Product Owner can provide clarity.
6. This step often surfaces a different understanding of the scope or implementation, which can then be clarified and agreed. The person that gave a 2 may know an easy solution, but the person that gave a 5 may anticipate a difficultly no one else thought of.
7. Repeat the estimation process until a consensus is reached. Or sufficient convergence is achieved as per the development teams' own rules.

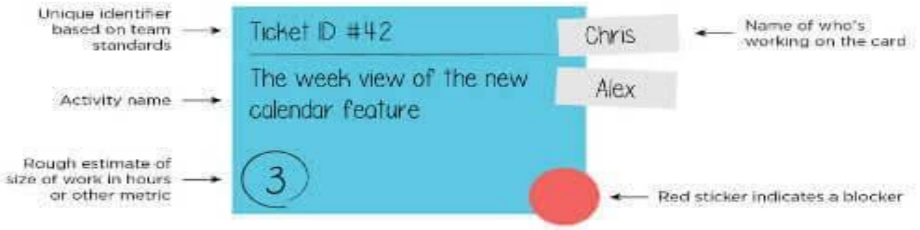
Q.1.2 A group of people is gathering to estimate user stories and arrange them on the wall according to their size. What method is the team attempting to employ? What is the goal/purpose of this technique? [2]

Ans:

Kanban board, workflow: it’s visual representation.

A Kanban workflow defines explicit policies and principles, followed by the Kanban team. Its main objective is to represent the rules and procedures of work while the work is flowing across different stages of its development and delivery cycle.




Unique identifier based on team standards
Activity name
Rough estimate of size of work in hours or other metric
Ticket ID #42
The week view of the new calendar feature
3
Chris
Alex
Name of who's working on the card
Red sticker indicates a blocker

Q.7 Set. (B) Determine the velocity of each iteration in the table below. [4]

	Iteration-1	Iteration-2	Iteration-3
Story points at start of iteration	100		
Completed during iteration	35	40	36
Changed estimates	5	-5	0
Story points from new stories	6	3	

Ans:

it's simply the number of Story Points "burned" in a sprint (**velocity**)

Iteration-1: 35, Iteration-1: 40, Iteration-1: 36,

Q.7 Set. (C)

User Story-1; Story Points= 8; Value Point=55	**User Story-2; Story Points= 3; Value Point=13**
User Story-3; Story Points= 8; Value Point=21	**User Story-4; Story Points= 5; Value Point=55**
User Story-5; Story Points= 2; Value Point=8	**User Story-6; Story Points= 13; Value Point**

The team velocity is 13 and the iteration length is 2 weeks. **Which user stories will be targeted for the next iteration?** The customer desired that releases be made after every second iteration. This project begins on March 15, 22? **What is the anticipated deadline for the final release?** [2+2]

Ans:

Which user stories will be targeted for the next iteration? :

The term *Moscow* itself is an acronym derived from the first letter of each of four prioritization categories: M - *Must have*, S - *Should have*, C - *Could have*, W - *Won't have*.

Weighted Shortest Job First (WSJF) is a prioritization model used to sequence jobs (eg., Features, Capabilities, and Epics) to produce maximum economic benefit. In SAFe, WSJF is estimated as the Cost of Delay (CoD) divided by job size.

Value of story 1 = 55/8 = 6.87

Value of story 4 = 55/5 = 11

We will take story four and three as these stories has high business values.

What is the anticipated deadline for the final release?

If the table represents a prioritized backlog (this is a must in Scrum!), you would take the first X Stories depending on your estimated velocity for the next sprint.

The order only depends on the Product Owner is the only one who prioritizes the backlog. There are several approaches to do this:

Total Story Points = (8+3+8+5+2+13) = 39 (Story points)

Total Sprint = 39/15 = 2.6

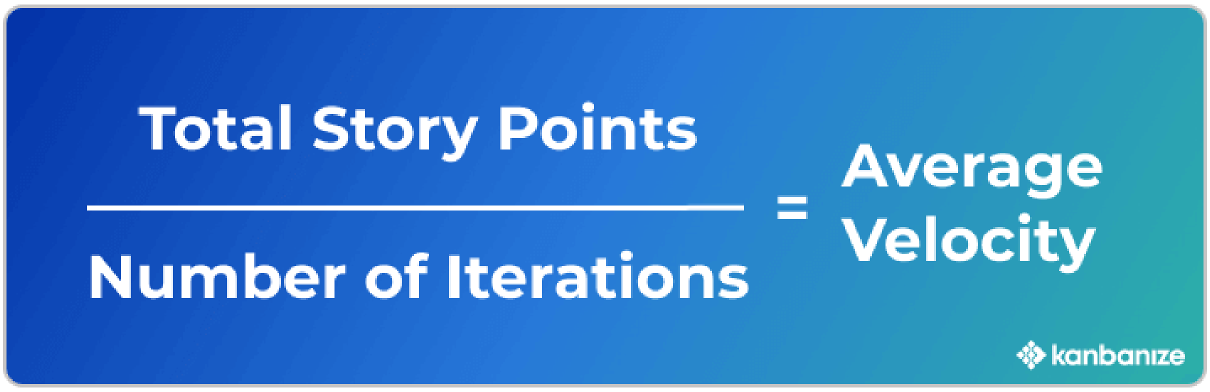
So in theory we might finish the project within three Sprints.

Anticipated deadline for the final release is Mid of May 2022.

==================================

To calculate initial velocity, it’s recommended that you start by agreeing with the developers on what each story point (SP) means in terms of working hours. For example, 1 SP equals 4 hours, 2 SPs should respectively equal 8 hours, 3 SPs = 12 hours, 5 SPs = 20 hours of effort, etc.

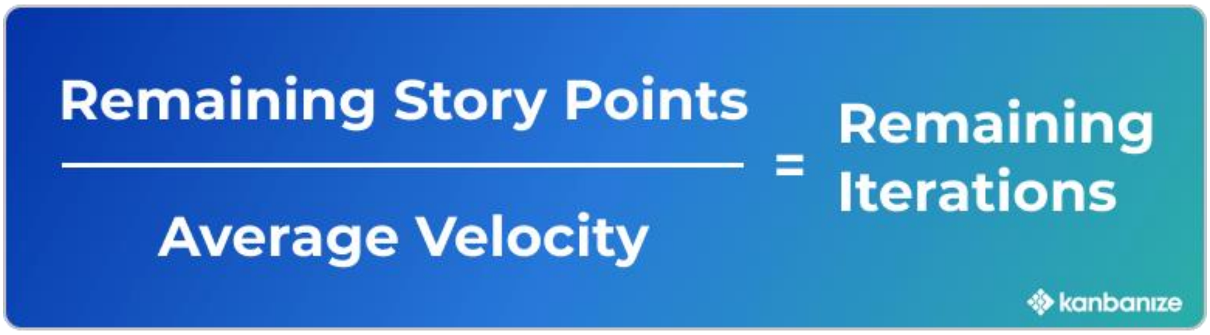
The formula for Agile velocity is below.



Now imagine that you’ve been through 4 iterations, practical and your velocity for them is the following: ***Iteration 1: 21 points, Iteration 2: 13 points, Iteration 3: 28 points, Iteration 4: 15 points.***

The average Agile velocity that we can derive from here is below.

Going forward in the project, you can use that Agile velocity to plan how much work you can commit to. For example, suppose that you have estimated that your backlog consists of 286 story points in total. After you subtract the completed ones (77), this means you have 209 points remaining. You can then divide 209 by your average velocity (19), which means your project could be completed after 11 more iterations.



$$\frac{\text{Remaining Story Points}}{\text{Average Velocity}} = \text{Remaining Iterations}$$

How to Improve Team Velocity in Agile?

There are a few techniques that you can use to improve team velocity in Agile. One of the most prominent ones is to analyze where work gets stuck, uncover dependencies and blockers so you can find ways to reduce their impact.

Other useful techniques include:

- Focus on visualizing the “Definition of Done” through process policies and acceptance criteria to make it clear for everybody on the team. This will help you prevent bugs and reworks.
- Reduce context switching to keep the developer’s focus on one thing at a time. This can be done by imposing Work In Progress limits, but also the Scrum Master should aim to shield the team from external distractions.
- Break down the user stories into smaller pieces (if possible) to improve their flow.
- Increase the Product Owner’s involvement in the work process to reduce misunderstandings and make the user story’s details clearer.

Q.8 Set. (A) Here are the velocities of two teams working on the same type of project in nearly identical environments and with nearly identical resources.

Iteration	**1**	**2**	**3**	**4**	**5**	**6**	**7**

Team-1	**8**	**20**	**18**	**14**	**16**	**24**	**19**
Team-2	**5**	**11**	**6**	**12**	**14**	**29**	**24**

Consider team-1's iteration-1 velocity as an outlier. Determine the velocity of team-1.

Calculate team 2's velocity using "yesterday's weather" and only the last two sprints. Which team is the most productive? What are some of the possible causes for Team 1's velocity to drop to 10 on the next sprint? [2]

Ans:

Team-1 velocity: (8+20+18+14+16+24+19)/7 = 119/7 = 17 (Normal velocity)

Team-1 velocity: (8+20+18+14+16+24+19)/6 = 119/6 = 19.8 (As 1st iteration is outlier)

Team-2 velocity: (29+24)/2 = 26.5 (Due to yesterday's weather)

Team 2 is most productive on iteration five and iteration Six.

Some of the possible way to drop team one velocity may be team member availability, Swap team member or new member adding in team, Other elements that affect velocity could be external to the team, such as the stability of the infrastructure. If the development team is continually called upon to fix production problems, then they will have less time to spend writing new code.

Q.8 Set. (B) Here are some excerpts from user stories. Using Story points, size them appropriately. Your assumptions should be listed.

1. **Migrate a legacy datastore to a new database technology.**
2. **Add a text field to the form.**
3. **User Id validation and storage to the form.**
4. **Data is retrieved using a third-party API that has been integrated.**
5. **Redesign a web application's layout**
6. **Migrate from a complex form to a new form in stages.**

The team's velocity over the last three iterations has been: 20,25,28 story points. How long will it take the team to complete the first four stories? [2]

Ans:

1. Migrate a legacy datastore to a new database technology. – 8 **story points**
2. Add a text field to the form. – 3 **story points**
3. User Id validation and storage to the form. – 2 S **story points**
4. Data is retrieved using a third-party API that has been integrated. - 10 **story points**
5. Redesign a web application's layout – 15 **story points**
6. Migrate from a complex form to a new form in stages. - 7

Average velocity from last three sprint = (20+25+28)/3 = 73/3 = 24.33 SP

Time to take the team to complete the first four stories = (8+3+2+10) = 23 SP. It will take one sprint to complete the first four stories.

Q.8 Set. (C) In their most recent two-week iteration, Team-A completed 43 story points. Team-B is working on a different project and has twice the number of developers as Team A. In their most recent two-week iteration, they also completed 43 story points. How is that possible?
Can we conclude that Team A is more productive than Team B? [2]

Ans: As per question, Team A is working on different project and Team B is on different projects.

Team-A completed 43 story points with X number of team members.

Team-B completed 43 story points with 2X number of team members.

We can conclude on basis of story point completion. **The story points of one team are not comparable to the story points of any other team.** Team B might be having some complex project, Complex IT infra, Resource un-arability…etc.

Q.9 Set. (A) An Agile team is working on improving the SLA for level-1 tickets in the production support system of an e-commerce site. Currently, Level-1 tickets take approximately 6 weeks to deploy in production.

Q.1.1 Determine two lean software development approaches that the team can employ to improve the process.

Q.1.2 Briefly describe the steps involved in both approaches. [2+4]

Ans: The Lean approach to business processes, originally derived from the enormously influential Toyota production system (TPS), is based on three fundamental principles: **delivering value as defined by the customer, eliminating waste, and continuous improvement**.

- **Continuous Improvements**: Leaders who use the lean model, think that there is always room for improvement. They might use software to monitor their company's continuous improvement by planning, measuring, and documenting the lean initiatives and operations. These leaders frequently design their processes using an integrated reform cycle like **DMAIC (Define, Measure, Analyze, Improve, and Control), PDSA (Plan, Do, Study, & Act), etc**.
- **Appreciating People**: According to corporations that use lean philosophy, individuals who are responsible for making the goods or services frequently have the best ideas. They cherish the opinions of their staff equally to the opinions of their consumers. Lean firms can more effectively find opportunities for improvement by enabling employees to contribute their observations and suggestions.

Advantages of Lean

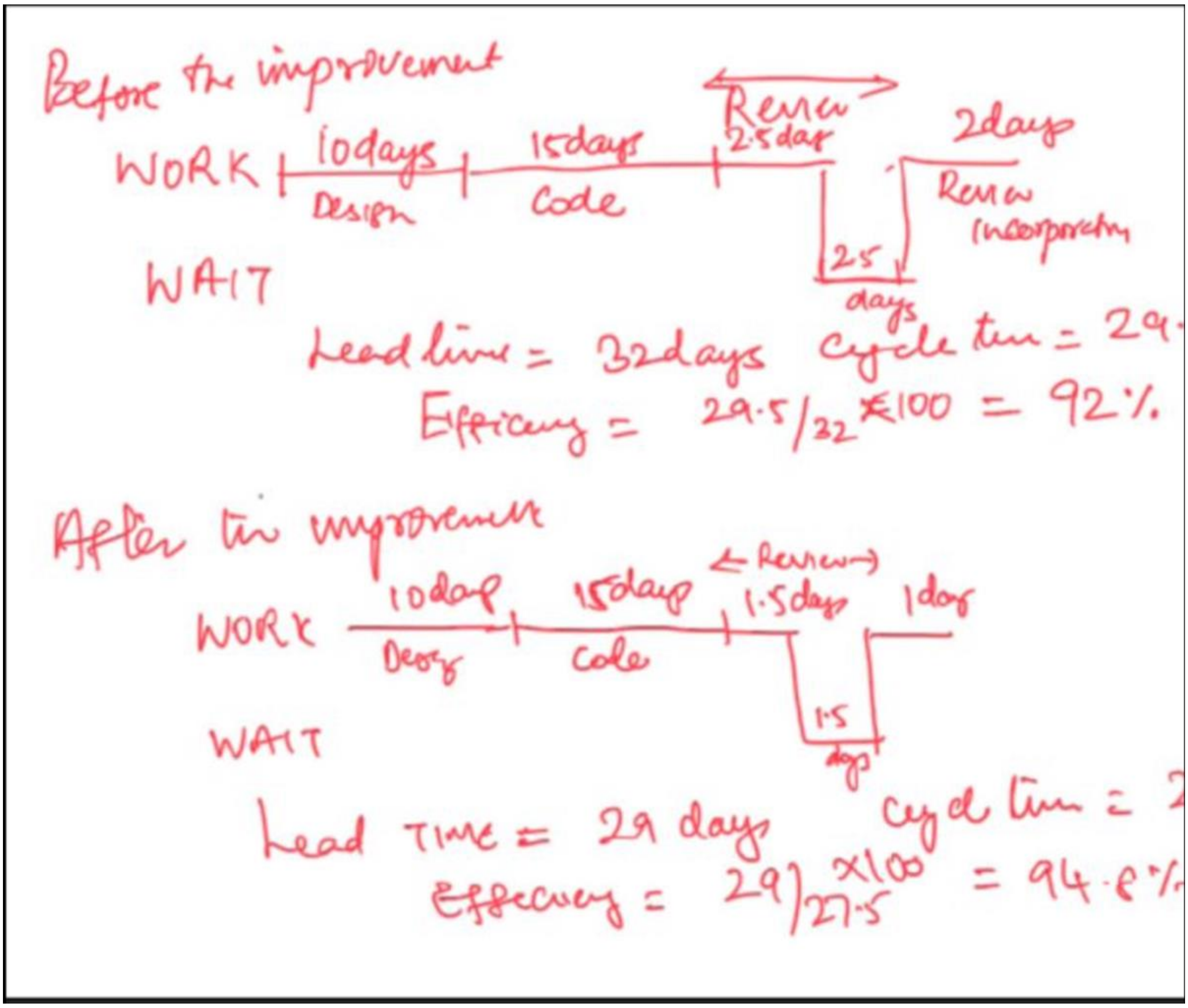
Here are some of the key benefits that you can avail by following the Lean Methodology:

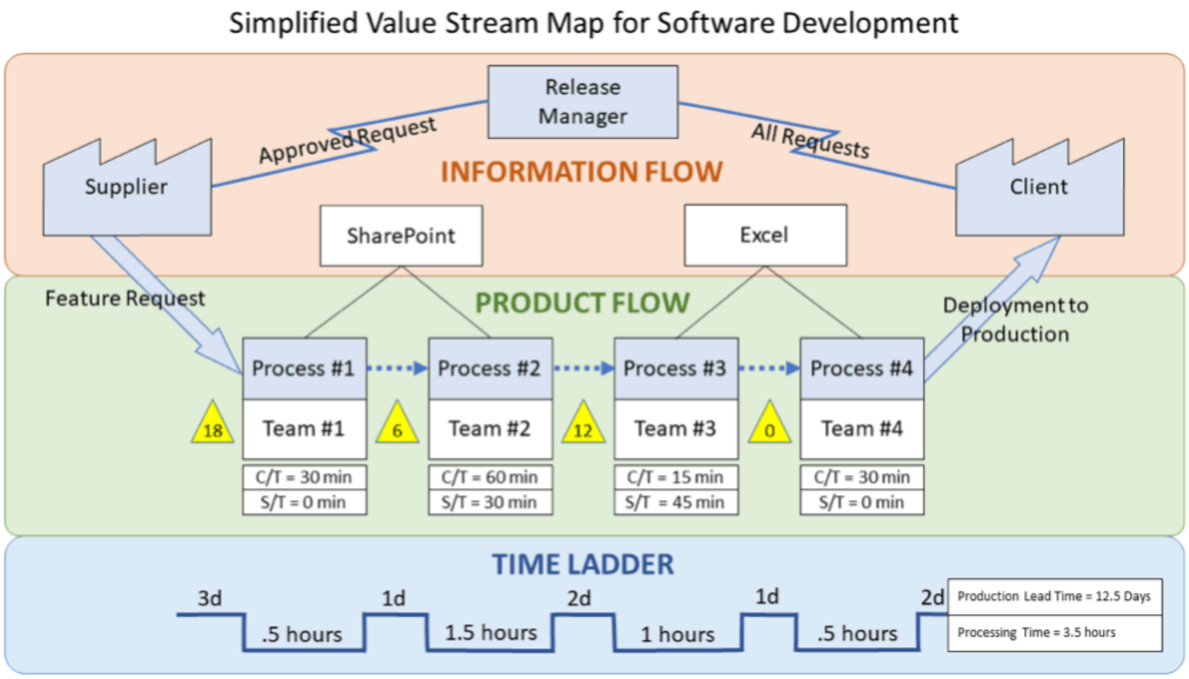
- **Reduced waste**: Lean methodology aids businesses in determining better resource allocations in order to reduce waste, give greater value to customers, and boost profit margins.
- **Better Productivity**: It can help you increase the operational efficiency of your company by removing distractions and giving employees instructions that enable them to concentrate on critical tasks.
- **Increased Process Efficiency**: Companies that use lean methodologies can optimize their procedures by embracing the concept of pull, which involves only creating things whenever there is a market demand for them. This enables businesses to guarantee that each stage of their value chain is running well.
- **Better Focus**: By decreasing unproductive operations like downtime, extra inventory, and needless duties, lean methods can assist an entire organization to improve its focus. This frees up staff's time to focus on tasks that add value to their clients.
- **Reduction in Costs**: The paradigm of the lean technique focuses on maximizing profits by minimizing waste and determining the greatest price consumers can justify spending for an item.
- **Better Quality**: Companies can increase the general value of their offerings by focusing solely on what delivers benefits to clients. They can also better satisfy the demands of their clients and provide them with fresh, inventive solutions.

Q.9 Set. (B) A team spent 10 days designing a feature and 15 days coding it. The team distributed the design documents and code and had them reviewed by a few experts. The review took 40 hours because the reviewers requested clarifications back and forth. It took two days to incorporate the review. During the review, the team's productivity was 50% of the review effort. 1day= 8 working hours. [2+4]

1. **Q.1.1 To improve the review process, propose a process improvement step.**
2. **Q.1.2 Assume that your process improvements resulted in a 40% reduction in review effort and a 1day reduction in review incorporation. Can you create a value stream map of this process before and after the improvement? How much has overall efficiency increased?**

Ans: Minimize the reviewers requested clarifications back and forth.
"Value stream mapping is a lean manufacturing or lean enterprise technique used to document, analyze and improve the flow of information or materials required to produce a product or service for a customer."

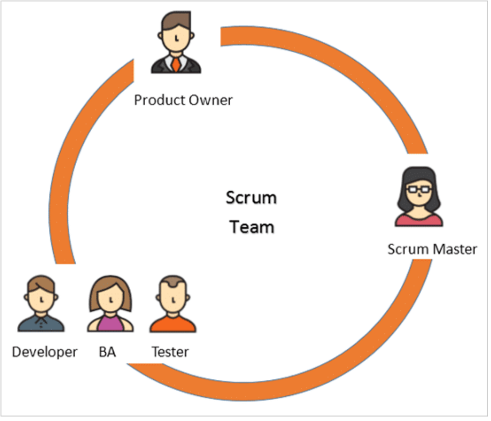


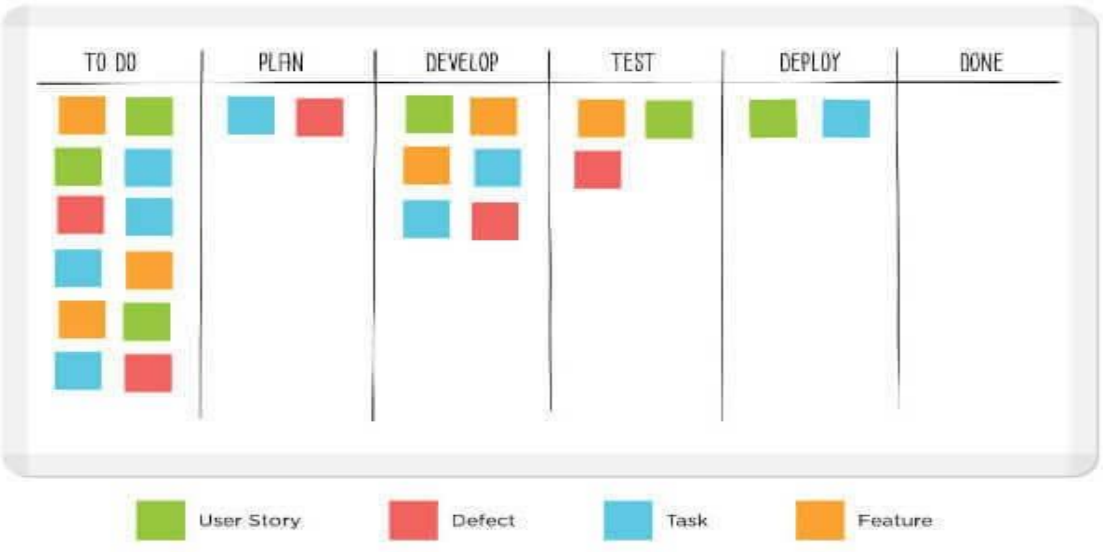


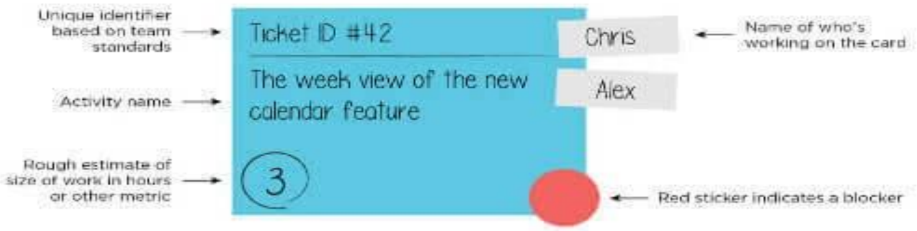
Q.9 Set. (C) A team consists of two business analysts, four developers, and three testers. Business Request, Scheduled, Building, Testing, Business Review, Ready for Release are the process steps for delivering work items. [2+3+1]

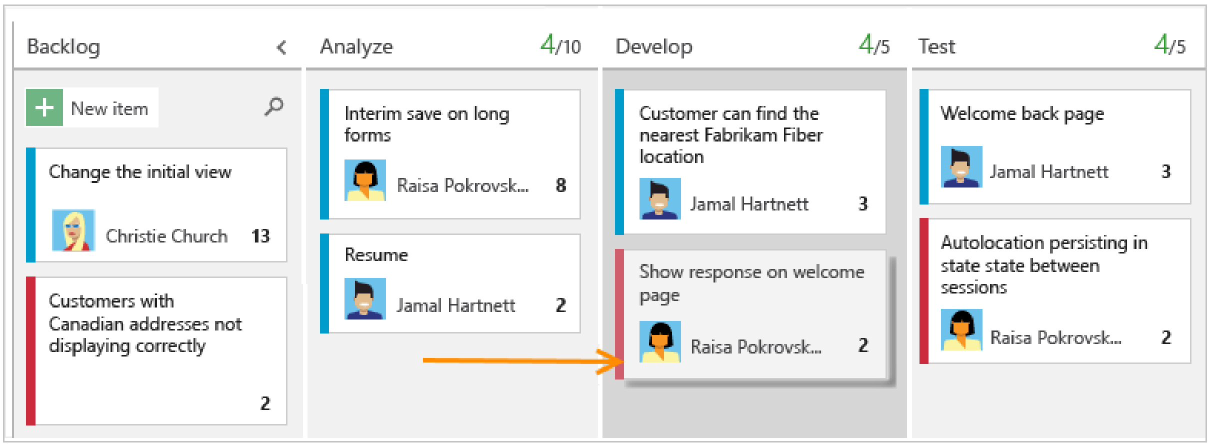
1. **Q.1.1 Create a visual representation of this workflow**
2. **Q.1.2 After a few weeks of implementing Kanban, you discovered that there are ten work items in the testing phase and one in the business review phase. This situation has been going on for a month. What could be the problem, and what can you do to solve it?**

Q.1.3 To begin, what WIP limit will you set?






Unique identifier based on team standards
Activity name
Rough estimate of size of work in hours or other metric
Ticket ID #42
The week view of the new calendar feature
3
Chris
Alex
Name of who's working on the card
Red sticker indicates a blocker



Backlog

New item

Change the initial view
Christie Church 13

Customers with Canadian addresses not displaying correctly
2

Analyze 4/10

Interim save on long forms
Raisa Pokrovsk... 8

Resume
Jamal Hartnett 2

Develop 4/5

Customer can find the nearest Fabrikam Fiber location
Jamal Hartnett 3

Show response on welcome page
Raisa Pokrovsk... 2

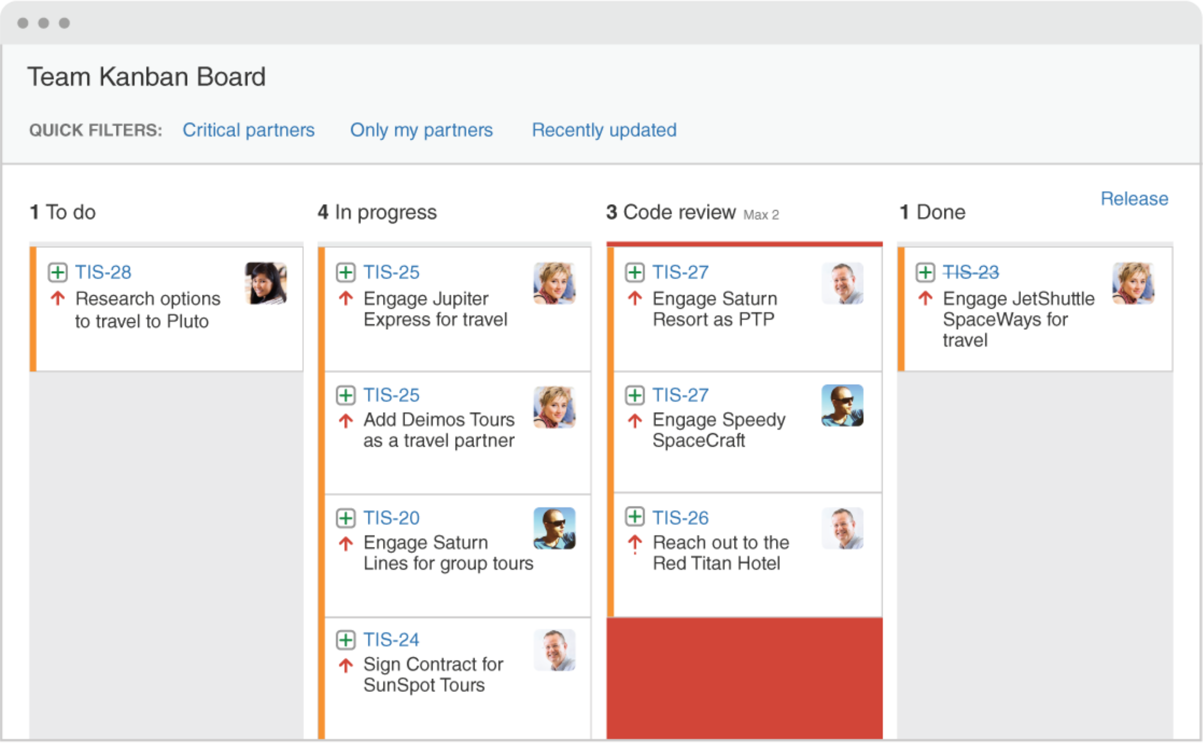
Test 4/5

Welcome back page
Jamal Hartnett 3

Autolocation persisting in state state between sessions
Raisa Pokrovsk... 2

Ans: 2nd Q: Swarming

Ans: WIP limit: Depending on the kind of work your team does and the number of people on it, a good starting point is probably somewhere **between the number of team members plus 1 and twice the number of team members**. For example, a good WIP limit for a team of 5 people is probably somewhere between 6 and 10 tasks. The main idea behind WIP limits can be explained by this simple phrase: **Stop starting, start finishing**.

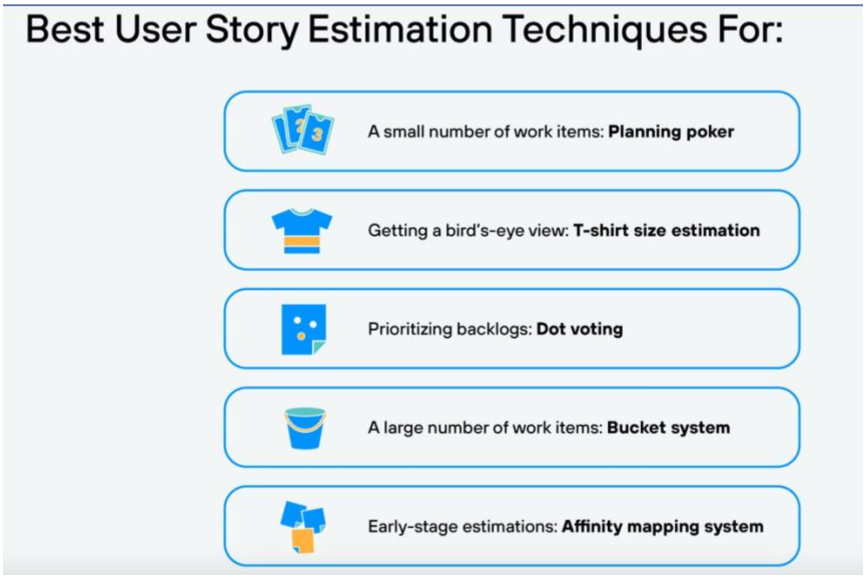


Team Kanban Board

QUICK FILTERS: Critical partners Only my partners Recently updated

1 To do	4 In progress	3 Code review Max 2	1 Done Release
TIS-28 Research options to travel to Pluto	TIS-25 Engage Jupiter Express for travel	TIS-27 Engage Saturn Resort as PTP	~~TIS-23~~ Engage JetShuttle SpaceWays for travel
	TIS-25 Add Deimos Tours as a travel partner	TIS-27 Engage Speedy SpaceCraft	
	TIS-20 Engage Saturn Lines for group tours	TIS-26 Reach out to the Red Titan Hotel	
	TIS-24 Sign Contract for SunSpot Tours		

Q. What are the way estimate user stories?



Best User Story Estimation Techniques For:

A small number of work items: **Planning poker**

Getting a bird's-eye view: **T-shirt size estimation**

Prioritizing backlogs: **Dot voting**

A large number of work items: **Bucket system**

Early-stage estimations: **Affinity mapping system**

Planning Poker. Planning poker is an agile estimation technique that makes use of story points to estimate the difficulty of the task at hand.

Q. Advantages of Agile

Many benefits are accrued by those who follow the Agile methodology:

- **Better Quality Products**: Testing is an integral aspect of the project implementation stage in Agile methodology, which implies the total grade of the finished product is higher. The client is kept informed throughout the development cycle and has the ability to request changes based on market conditions.
- **Increased Customer Satisfaction**: Customers are always included in the judgment cycle in Agile, which leads to higher customer loyalty. You provide greater benefits to the customer and guarantee that the finished product is actually suited to their expectations by keeping them in the fold and making adjustments based on their input.
- **Improved control**: Because of its openness, feedback inclusion, and quality aspects, Agile enables managers to have more influence over the process. All through the project's execution phase, quality is assured, and all participants are kept informed with daily status reports using robust reporting instruments and processes.
- **Greater adaptability**: When Agile is properly applied in a development team, it allows for unprecedented flexibility. Teams operate in shorter bursts, with the product owner providing frequent feedback and interaction.
- **Lower risks**: Any project that follows the Agile technique should never fail, in theory. Agile operates in tiny sprints with the goal of delivering value continually. Even if a certain technique does not go as anticipated, there must be a small element that can be retrieved and utilized in the future.