

**Q4. List the contraindications of real time customer involvement.****Answer :****Model Paper-III, Q1(c)**

Real-time customer involvement can also lead software towards certain contraindications which includes the following,

- ❖ Real-time customers sometimes focus on their own requirements instead of focusing on the requirements of all the customers. This can lead the development towards the fulfillment of real-time customer requirements only.
- ❖ They usually prefer improvements in their current software rather than adopting a new software. So, it is possible that they make new software turn into their own one.

Real-time customers can be ignored if the developers are confident enough and product managers are experienced enough. However, it is preferable to involve them for developing a successful software product that covers all the user requirements.

**Q5. Discuss the considerations of refining the ubiquitous language.****Answer :**

The process of refining the ubiquitous language must include the following considerations,

- ❖ The entire team should agree with the modifications to be done in the language. For this reason, a meeting is conducted so that every individual working on the project will become aware of the new language.
- ❖ The modified language should be clear enough to capture business requirements. These changes can be done easily but the difficulties lie in reflecting the same with respect to software requirements.
- ❖ In the situations where programmer changes are not possible with respect to the domain-centric language, explain the same to the domain experts.
- ❖ The software should be updated every time when changes are made. This means that recoding must be done whenever the logic of domain is changed.
- ❖ The change ignorance can gradually lead to the mismatches among the result and the design.

**Q6. Write a short note on some coding standards.****Answer :****Model Paper-I, Q1(d)**

Following are some of the coding standards,

**1. Restrict Global Data Types**

This rule restricts the declaration of global data types.

**2. Follow the Naming Conventions**

Naming conventions for global variables, local variables and constant identifiers must be strictly followed. A global variable must begin with an uppercase, a local variable must be in lowercase and a constant identifier must be in uppercase.

**3. Specify Proper Format for the Headers**

The header format for the module must be properly written and declared in such a way that it can be used in the next modules as a standard. Some of the standard header formats are – name of the module, author's name, date on which the module was created, modification history, synopsis of the module, etc.

**Q7. How iteration demo keeps the momentum?****Answer :****Model Paper-II, Q1(d)**

The team working on extreme programming delivers a software for every week that can perform a specific functionality. To keep this momentum, a special kind of approach is adopted which is called as iteration demo. In the first stage of iteration demo, the progress of the team is demonstrated to the stakeholders. As the process proceeds, the stakeholders with some customers join the team and tracks the progress. This builds an honest environment where every individual to expose the truth. In the last stage, the feedback is collected from the customers on a regular basis. This feedback is provided the customers everytime when a demo is released. The major advantage of using iteration demo is that, the problems arising in the software gets exposed to the entire team thereby creating an platform for managing it effectively.

**Q8. Discuss the two key questions to the executive sponsor.****Answer :**

Model Paper-III, Q1(d)

The two key questions raised towards executive sponsor are,

1. Is the work progress running according to the schedule?
2. Should the process be continued?

These questions are important for monitoring the project status and provides a platform for sponsors to express their point of views on the progress. Based on the answers of the above questions, the sponsors are questioned regarding the improvements he/she is expecting. Programmers should consider the suggestions given by the sponsor seriously and work on it. The team should work in a way that the sponsor should not answer 'no' to the second question. If the sponsors answer 'no' to the second question, the project will usually be terminated. The only possibility of continuation is to find out the problems by speaking with the sponsor.

## PART-B

### ESSAY QUESTIONS WITH SOLUTIONS

#### 2.1 TRUST

**Q9. Discuss time boxed half-hour process that can be performed everyday to improve collaboration.**

**Answer :**

Model Paper-I, Q4(a)

Communication and collaboration plays an important role in software development. It seems to be easy to manage collaboration but in reality, it is a complex task. This is because different individuals think differently leading to different points of view. To avoid this, all the individuals working on the project need to raise their queries, conduct meetings and share their views.

Collaboration can be improved by following time boxed half-hour process daily which involves the following steps,

#### **Step-1**

The first step in this process is to create pairs of either programmers and customer, customer and tester etc. Moreover, these pairs are changed regularly.

#### **Step-2**

The information required by one in a pair should discuss with the other. At this point, different color cards are used. For example, if the time utilized from the moment of realizing the requirement till it is fulfilled is less than 10 minutes, pick a green card. In case if the time involved is less than 24 hours, pick a yellow card, else take red card. Note down the details like,

- ❖ Whose requirements are to be fulfilled?
- ❖ Who will fulfill the requirements?
- ❖ What is the total time consumed?
- ❖ How much time is taken to fulfill the requirement?

#### **Step-3**

The capabilities of various team members are discussed in the pairs so that the time involved in fulfilling the requirements can be minimized.

#### **Step-4**

Finally, all the cards data is discussed among the team and the following information is obtained,

- ❖ The difficulties in obtaining the information.
- ❖ The list of latencies which are responsible for creating hurdles in the process.
- ❖ The reasons that lead the latencies to create hurdles.
- ❖ The flow of information that best suits for optimization of next iteration.
- ❖ The measures that can be adopted for enhancing the information flow.

The team that works collaboratively, easily achieve their goals as the priorities are decided with mutual understanding. Although they might be facing tight deadlines, they continue to work without feeling much pressure as they enjoy working as a team.

If the same task is carried out with a group of competing individuals, some of them might get hurt if their performance is marked as unsatisfactory irrespective of their efforts.

**Q10. Write a short note on trust and discuss various strategies for generating trust in XP team.**

**Answer :**

Model Paper-II, Q4(a)

### Trust

Trust is an important aspect in any relationship. A team must build trust on each other to achieve the targets successfully. However, building trust usually takes time as the team passes through various stages like forming, storming, norming and performing. With all the experiences, they become familiar with the capabilities of each other. With this, they can accept the opportunities with joint responsibility. When any member of the team finds important to do something during the project, he/she starts working on it without searching for someone to work on it. Moreover, if a member faces difficulty in performing a task, he/she must feel free to ask other members. Moreover, the persons who help in resolving issues of other members must not feel that this affects his/her work.

Trust is also important from organization's side on the team. This is because, there will be nothing to show to the managers during the initial stages of XP. So, managers should trust the team that they will complete the project as expected.

### Strategies for Generating Trust

There are many strategies for generating trust among the team members of XP. Some of these techniques are as follows,

#### 1. Mutual Understanding Among Customer and Programmer

It is important to have mutual understanding among customer and programmer for a healthier result. Usually, customers think that the programmers ignore their requirements. Sometimes programmers lose their jobs when they fail to submit their work on time. If the customers pressurize the programmer to complete their work on time, programmer will adopt shortcuts like use of ready-to-use programs that leads to the poor quality of generated product. In such situations, building trust is very much difficult and takes a long time to recover. The reason for this is that both the customer and programmer need to change their attitude. Mutual understanding can be improved by conducting face-to-face meetings. Moreover, a respectful behaviour from both sides can help in understanding technical requirements of each other.

#### 2. Mutual Understanding Among Programmer and Tester

Testers sometimes focus on targeting the mistakes of programmers instead of focussing on the quality improvements. Similarly, programmers show disrespect towards testers. To avoid this, mutual understanding among them need to be generated. For this, both programmer and tester need to show respect towards each other. From the programmer's point of view, they should consider the mistakes found by testers as their areas of improvements. From the tester's point of view, they should focus on releasing the product without any errors.

#### 3. Eat Together

Eating together is another way of generating trust among team members and improve cohesiveness. It wipes out all the barriers and creates a friendly environment. From the manager's side, providing a lunch/dinner once in a week by gathering the entire team together will build helps in understanding each other.

#### 4. Team Continuity

A team usually breaks after the completion of a project and a new team is formed. In this case, the team has to face all the phases to develop collaboration and trust. To avoid this, the teams which have generated productive output can be continued for the next project. Moreover, organizations can consider the teams as resources instead of people.

**Q11. Explain the organizational strategies to build trust.**

**Answer :**

Model Paper-III, Q4(a)

Following are some of organizational strategies to build trust within the organization,

#### 1. Showing Hustle

Hustle generally refers to the action of moving or working quickly in order to accomplish a task. In terms of software development the same term refers to the energized and productive work done by the software team. With this, it can be conveyed that the software team is putting enough efforts daily. The interest of the team can also be observed by the regular reports and iteration demos from the team.

## **2. Deliver on Commitments**

The stakeholders analyze the project based on the end requirements and working of the end product. In extreme programming, these two measures are reported weekly to ensure successful and on-time delivery of software. With such regular updates, trust can be built among stakeholders as they know the status of the development.

## **3. Manage Problems**

The most effective way to manage problems is to restrict the software to be within the specified limit. This avoids the problems to maximum extent. Moreover, the tasks which are complex should be performed in the initial phases so that the problems can be detected early. These problems can be solved accordingly during the development cycle.

When a problem is detected, conduct a meeting with entire team and discuss the problem so that it can be resolved easily with multiple minds working on it. However, if the intensity of problem is low, the problem can be resolved in the next iteration. In case, if the intensity is high, replan the iteration as soon as possible involving the entire team. Moreover, it should also be discussed with the stakeholders. With such a discussion, the programmers can get some additional time to resolve the problem completely. If it is consuming more time than expected then some problem exists in the system.

## **4. Respect Customer Goals**

Customers feel comfortable in working in their normal offices. To make them comfortable, the programmers should communicate with them in a respectful manner. The programmers must also show interest towards meeting the goals of customers by finding various alternative for both time saving and cost saving with fulfillment of their requirements. With such meetings, communication gap between both programmers and customers will get fulfilled.

## **5. Promote the Team**

The team can be appreciated and promoted in various ways like pasting pictures of work progress in the workspace, inviting higher authorities for attending iteration demos and other ways of appreciation.

## **6. Honesty**

Programmers should report honesty to the stakeholders to avoid disappointment and confusion. This usually happens when some backlogs remain during the reporting time. The programmers might think that recovering the backlogs is an easy task but sometimes it affects the entire schedule of the project. So, it is important to report honestly.

## **2.2 SIT TOGETHER**

### **Q12. Discuss the importance of sitting together.**

**Answer :**

Model Paper-I, Q5(a)

The team sitting together usually has strong bonding among the team members. This allows direct interaction rather than communicating over phone calls, e-mails or other ways of communication. In a traditional non-agile development, such a communication consumes a lot of time. For example, if a programmer gets a doubt in the requirements document of software, he/she needs to communicate with the domain expert. If he/she sends a mail regarding the same, the domain expert might not reply instantly. After some time, when the domain expert replies, the programmer might not find the answer to be exactly according to the question. Again, he/she sends a reply back to the domain expert, the communication consumes a lot of time just for a single clarification that might consume a few seconds when the team sits together.

The communication effectiveness is inversely proportional to the distance between the communicating parties. This means that the effectiveness decreases with increase in the distance thereby leading to misunderstandings and delays. To avoid such delays, team members proceed by guessing thereby committing mistakes. Organizations prefer to avoid direct interaction by creating requirements document during the end-of-phase meeting. Programmers can refer to this document whenever a query is raised. However, the documents need to be written after performing analysis on the queries that can be raised during the development.

The above limitations are eliminated in XP as the whole team sits together. The team includes the following members,

- ❖ Design experts
- ❖ Programmers
- ❖ Testers
- ❖ Business experts.

In such environment, if the programmer gets a doubt in the document, he/she can directly approach the author of document and clarify it within few minutes. This saves time, avoids misunderstandings and improves productivity. A survey revealed that sitting together doubles the productivity and decreases the duration of project by 3 times. The key role in such an improvement is the communication.

A survey on tester's reveled that programmers spend around 20 percent time in coding while 30 percent on programming. The remaining 50 percent of time is utilized in other tasks like conducting/attending meetings, resolving issues etc. However, the programmers of XP team utilize their time usually in programming. The increased percentage is due to the fact that the entire XP team sits together. The issues/queries are resolved instantly and there will be no long meetings. This is because, they share about the work process during the conversations.

Some organizations disallow the team members to interrupt each other for asking doubts. This is usually followed by the organizations who think that interruptions will divert minds of others and can affect their flow of work. However, in XP teams pair programming is preferred. Following such an approach eliminates this limitation because, if one among the pair is engaged in the interruption, the other one continues to work in a flow.

### **Q13. Give an overview of making room and workspace designing.**

**Answer :**

#### **Making Room**

The task of converting your current workspace to XP workspace where the entire team sits together is a complex task. At first, there should be enough space available to make a room that can allow the seating arrangement of entire team. Such an arrangement consumes time and money as the current setup needs to be changed. When the new setup is under construction a conference hall that is large enough to provide enough space for the entire team can be used.

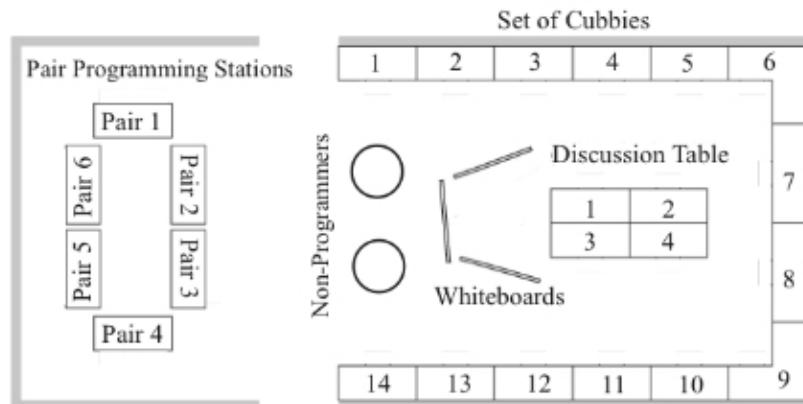
#### **Designing Workspace**

Workspace should be designed in such a way that all the programmers should sit beside one another. Moreover, testers should also be kept near the programmers. However, the domain experts and interaction designers can be kept at some distance but they must be at a distance that they can answer the programmers without raising their voice level. The reason for keeping all the programmers at one place is that they work with continuous collaboration. The testers are kept beside them to keep an eye on the issues getting faced by the programmers.

The managers (product and project managers) are kept at some distance so that the conversations among them will not disturb others. Avoid completely open space as everyone prefer to have atleast enough private space to keep their personal stuff. Moreover, a closed door room away from working area should also be provided so that the team members can make use of it for personal calls, individual meetings and other works.

Apart from these arrangements, the workspace should also include multiple white boards for writing various types of information. Sometimes, a projector is also included in the workspace as it can be used for better explanation. It also saves the time of conducting meetings in the conference hall.

The best way to design pairing stations is to create a set facing each other. An example of workspace design is shown in the figure below,



## 2.3 REAL CUSTOMER INVOLVEMENT

**Q14. Write short notes on real customer involvement.**

**Answer :**

Real-time customer involvement can play a crucial role in the development. Customers provide stories, their priorities and features that can be considered in the next iteration. They provide this information based on their experience and previous iterations. With their involvement, the team gets to know the way in which customers are going to use the software. Moreover, the programmers can also gain knowledge about the end-goals along with the difficulties/limitations existing in the software. Based on this knowledge, the software can be improved in the next release.

Real-time customer involvement can also lead software towards certain contraindications which includes the following,

- ❖ Real-time customers sometimes focus on their own requirements instead of focusing on the requirements of all the customers. This can lead the development towards the fulfillment of real-time customer requirements only.
- ❖ They usually prefer improvements in their current software rather than adopting a new software. So, it is possible that they make new software turn into their own one.

Real-time customers can be ignored if the developers are confident enough and product managers are experienced enough. However, it is preferable to involve them for developing a successful software product that covers all the user requirements.

**Q15. Explain various types of development based on customers.**

Model Paper-II, Q5(a)

The type of development vary with the type of customers. Some of such developments are as follows,

### 1. Personal Developments

Personal development refers to the development of software which is used by the developers themselves. In other words, the developers act as customers.

### 2. In-house Custom Development

In-house custom development refers to the development of software which is used within the organization and is developed in-house. An example usage of such software is the automation of account based calculations. Such development involves fulfillment two types of requirements. Firstly, the requirements of sponsors, secondly the end-users as they belong to the same organization. The requirements of both types of users sometimes mismatch leading to complexities in the development.

The simplicity in such a development is the easy availability of end-users. This makes the end-users to turn into real time customers, sponsors into product managers and some end-users into domain experts. All these will help in development of a better software as the developers can gain full information regarding how the end-users will use the project. The people involved in development should get feedback from their co-workers by demonstrating the plan.

### 3. Outsourced Custom Development

Outsourced custom development refers to the development which is used within the organization but developed outside the organization premises. In this type of development, end-users cannot be involved much as they will not be available all the time. However, it is necessary to involve them by meeting them in person for discussing about the project goals and release plan. If possible, conduct a meeting everytime when an iteration demo is generated or planning session is conducted.

### 4. Vertical Market Software Development

Vertical market software development refers to the development of software which is used by the customers belonging to different organizations of some industry. Such software are customized based on requirements of the end-users. In this type of development, the customers are given priority for directing software development but it should not ignore the requirements of others customers. It is the responsibility of product manager to consider all the needs of the users, integrate all these needs and provide a single view of the software.

An alternative way of considering the user requirements is to get the feedback instead of recruiting them as real customers. Few of the organizations make use of review boards where the potential customers are provided with sample software for trials. Moreover, these customers are instructed to arrange real-end users for joining the development team.

### 5. Horizontal Market Software Development

Horizontal market software development refers to the development of software which is used by the customers belonging to different organizations of different industries. It is very much similar to vertical-market software development. The difference here is the use of beta versions of software, experience testing etc., in place of review boards.

## 2.4 UBIQUITOUS LANGUAGE

### **Q16. What is the need of ubiquitous language and discuss the use of common language?**

#### **Answer :**

Ubiquitous language refers to the language that can be easily understood by both the programmer and the domain expert. Such language is needed to provide a stable platform for the entire team as the domain experts are usually unaware of coding standards whereas the programmers are unaware of the standards followed in problem domain. It provides a common language instead of two different languages to avoid confusions and disappointment after development. Use of such language saves time that is consumed in translating the language of domain experts to programming terms and vice versa.

However, if it is not translated well, the product development might turn into a wrong direction.

#### **Example**

If programmers explain about a problem to the domain experts using their programming constructs like – I, parent element, child element etc., domain expert might respond with incorrect suggestion as domain experts are usually not familiar with such terms. However, if the same problem is explained as domain terms, the domain expert can easily understand the actual problem and respond with valuable suggestions.

Apart from this, use of ubiquitous language also helps all the team members to join in the discussions whenever they have suggestions regarding the problems existing in other areas of development. In the use of ubiquitous language, domain terms are preferred. However, if the software product involves more technical terms like development of web servers, compilers etc., then technical terms can be used. Such designs are usually considered good when designing simple and small codes. Moreover, these designs can lead to bugs and complexities in case of larger projects.

### **Q17. Explain the usage of ubiquitous language in code and describe the process of refining it.**

#### **Answer :**

Model Paper-I, Q4(b)

#### **Ubiquitous Language Code**

Programmers usually face difficulties in translating the code to a domain-centric language. An alternative way to simplify the process of translation is to use domain-centric language based terms in the code. To do this, name the classes, methods, variables etc., using the terms used by domain experts. For this, the programmers need to gain knowledge about domain thereby bridging the gap in translating the technical terms to domain terms.

Domain-centric code can be developed by developing a domain model. These models can be created in different ways based on the type of design.



### Refining the Ubiquitous Language

The process of refining the ubiquitous language must include the following considerations,

- ❖ The entire team should agree with the modifications to be done in the language. For this reason, a meeting is conducted so that every individual working on the project will become aware of the new language.
- ❖ The modified language should be clear enough to capture business requirements. These changes can be done easily but the difficulties lie in reflecting the same with respect to software requirements.
- ❖ In the situations where programmer changes are not possible with respect to the domain-centric language, explain the same to the domain experts.
- ❖ The software should be updated every time when changes are made. This means that recoding must be done whenever the logic of domain is changed.
- ❖ The change ignorance can gradually lead to the mismatches among the result and the design.

## 2.5 MEETINGS

### Q18. Discuss in detail about conducting meetings.

**Model Paper-II, Q4(b)**

Meetings are organized to get the status of work progress from the individuals working on the software project. In such meetings, most of the time gets wasted while the actual meeting usually takes 10 minutes. To avoid this, a stand-up meeting is conducted on a daily basis for not more than 10 minutes. In this meeting, the teammates gather at one place and each individual/pair tells about the new information that is needed to be shared with the team. In agile environment, such meetings are called scrum. These meetings reveals the tasks performed previous day, the tasks to be performed on that day and the problems existing in the system that are creating complexities.

Stand-up meetings should be conducted later in the day rather than conducting it in the morning. The reason for this is the delay that can occur because of late comings and other reasons that can affect early start of meeting. As this meeting concluded within a short time, it should provide just an overview of the ongoing process. The time taken per each member is usually 30 seconds. If there is a requirement of discussing any further, individual/personal meeting is conducted.

Examples of people reporting in stand-up meetings are as follows,

- ❖ A programmer can mention that he/she has modified the way in which database connects to the server. He/she can also mention the time at which he/she can be available for clarifying the doubts.
- ❖ A product manager can mention the feedbacks he/she collected from any of the surveys conducted.
- ❖ A programmer who has finished his/her task can offer others to take help from him/her.

It is important to timebox the meeting to avoid unnecessary delays in the stand-up meetings. Some members unnecessarily extend the meetings. For such people, an alternative way should be adopted to keep them out of the meetings. Moreover, follow the timing of the stand-ups even if some people are not present.

If it is not possible to conduct a meeting daily, an alternative way should be adopted to keep in touch with all the members.

## 2.6 CODING STANDARDS

### Q19. Write about the creation of coding standards.

**Model Paper-III, Q4(b)**

Coding standards refer to the guidelines that must be followed by the programmers in writing programs.

Following are some of the standards,

#### 1. Restrict Global Data Types

This rule restricts the declaration of global data types.

**2. Follow the Naming Conventions**

Naming conventions for global variables, local variables and constant identifiers must be strictly followed. A global variable must begin with an uppercase, a local variable must be in lowercase and a constant identifier must be in uppercase.

**3. Specify Proper Format for the Headers**

The header format for the module must be properly written and declared in such a way that it can be used in the next modules as a standard. Some of the standard header formats are – name of the module, author's name, date on which the module was created, modification history, synopsis of the module, etc.

**4. Adopt a Standard Error Return and Exception Handling Policy**

The method of returning errors and the method of handling exceptions in a function must be standard adhering to the standards of organization. Generally, a function returns 0 or 1 when an error occurs.

Creating such standards is the first most task of programmers before coding. These standards are updated as the coding proceeds. For this reason, a minimal set of standards are generated and are updated accordingly for improvisations. The initial set is usually taken from an already available standards. These standards should consider,

- ❖ Various development methods.
- ❖ Tools and utilities
- ❖ Layout of files and directions
- ❖ Handling of errors
- ❖ Various events and logging approaches.

The above standards should be decided by conducting a meeting of all the programmers. However, it is very difficult to make standards that will be agreed by all the programmers. But as the time passes, programmers get use to it.

In case a programmer disobeys the standard, the development should be continued and later on, the standard should be updated whenever a problem arises.

**Q20. Discuss in brief about adhering to the standards.****Answer :**

Pair programming helps in adhering to the standards as it offers a platform on which coding and formatting related standards can be discussed. With this, the standards get improved time-to-time. Another way to make the team follow the standards is to employ collective code ownership. This approach can use used to make multiple members to modify a single code.

Some other methods include automated standards checking tools and customized version control systems. These tools will either return errors or rectify the codes automatically. In both the cases, there can be chances of getting false bugs. A better approach is to handover the responsibility to the programmers themselves and they don't follow the standards, they need to mention the reason. Ask the question to the programmers not following the standards with positive attitude in a polite way. Ask them to give suggestions to improve the standards. Consider the suggestions obtained if possible or else explain about the drawbacks of using the standard suggested by the programmer.

If the standards are good enough, it is possible that programmer might misunderstood those standards. In such situations, a detailed one-to-one discussion with the programmer will be a better choice. However, if he/she is a junior programmer, assist him/her in completing his/her job in coordination with the team.

Sometimes, the team simply denies to use even simple standards as they are new to extreme programming. To avoid this is to let the team violate the standards for months so that they can understand as with time and experience. However, if the traditional legacy code works well, then that will be a better choice than the new one. The standards, bugs, risk factors etc., can be considered in different iterations whenever modifications are done. For this purpose, an automated tool can also be used.

When new standards are implemented successfully, the code becomes easily maintainable and readable. However, if team works collaboratively, they can skip defining and implementing the coding standards.



# UNIT

# 2

## COLLABORATING



### PART-A SHORT QUESTIONS WITH SOLUTIONS

**Q1. What information is obtained in discussing the data present on cards?**

**Answer :**

Model Paper-II, Q1(c)

The following information is obtained when the data present on cards is discussed among the team,

- ❖ The difficulties in obtaining the information.
- ❖ The list of latencies which are responsible for creating hurdles in the process.
- ❖ The reasons that lead the latencies to create hurdles.
- ❖ The flow of information that best suits for optimization of next iteration.
- ❖ The measures that can be adopted for enhancing the information flow.

**Q2. Discuss any two organizational strategies to build trust among team members.**

**Answer :**

The two organizational strategies to build trust among team members are,

#### 1. Showing Hustle

Hustle generally refers to the action of moving or working quickly in order to accomplish a task. In terms of software development the same term refers to the energized and productive work done by the software team. With this, it can be conveyed that the software team is putting enough efforts daily. The interest of the team can also be observed by the regular reports and iteration demos from the team.

#### 2. Deliver on Commitments

The stakeholders analyze the project based on the end requirements and working of the end product. In extreme programming, these two measures are reported weekly to ensure successful and on-time delivery of software. With such regular updates, trust can be built among stakeholders as they know the status of the development.

**Q3. What are the difficulties faced by the team when not sitting together?**

**Answer :**

Model Paper-I, Q1(c)

In a traditional non-agile development, communication consumes a lot of time. For example, if a programmer gets a doubt in the requirements document of software, he/she need to communicate with the domain expert. If he/she sends a mail regarding the same, the domain expert might not reply instantly. After some time, when the domain expert replies, the programmer might not find the answer to be exactly according to the question. Again, he/she sends a reply back to the domain expert, the communication consumes a lot of time just for a single clarification that might consume a few seconds when the team sits together.

The communication effectiveness is inversely proportional to the distance between the communicating parties. This means that the effectiveness decreases with increase in the distance thereby leading to misunderstandings and delays. To avoid such delays, team members proceed by guessing thereby committing mistakes. Organizations prefer to avoid direct interaction by creating requirements document during the end-of-phase meeting. Programmers can refer to this document whenever a query is raised. However, the documents need to be written after performing through analysis on the queries that can be raised during the development.



## 2.7 ITERATION DEMO

**Q21. How iteration demo keeps the momentum? Discuss the process of conducting it.**

**Answer :**

Model Paper-I, Q5(b)

### Iteration Demos

The team working on extreme programming delivers a software for every week that can perform a specific functionality. To keep this momentum, a special kind of approach is adopted which is called as iteration demo. In the first stage of iteration demo, the progress of the team is demonstrated to the stakeholders. As the process proceeds, the stakeholders with some customers join the team and tracks the progress. This builds an honest environment where every individual to expose the truth. In the last stage, the feedback is collected from the customers on a regular basis. This feedback is provided the customers everytime when a demo is released. The major advantage of using iteration demo is that, the problems arising in the software gets exposed to the entire team thereby creating an platform for managing it effectively.

### Conducting Iteration Demo

Iteration demo can be conducted by any of the team members. Usually, it is conducted by product managers as they frequently interact with stakeholders. In the iteration demo, the team of developers, sponsors along with stakeholders are involved. However, customers are also considered whenever their presence is important. Moreover, the other teams interested in attending iteration demo are also included over a teleconference. The duration of the demo should be around 10 minutes as it considers the work of only a week.

It is important to follow the timings strictly to give a message to all the attendees that time is precious. The detailed discussion on the same is conducted after the meeting with product manager and the team. The discussion during the iteration demo is regarding the features of the demo or the changes in the demo. Expose all the problems instead of covering them. However, describe the ways in which these problems can be resolved.

In the next step, the stories are discussed one after the other. Moreover, use the customer tests for further demonstration. At the end of demo, demonstrate the ways in which stakeholders can use the software.

**Q22. Explain the following with respect to iteration demo,**

- (a) **The two key questions to the executive sponsor**
- (b) **Weekly deployment.**

**Answer :**

Model Paper-III, Q5

**(i) The Two Key Questions to the Executive Sponsor**

The two key questions raised towards executive sponsor are,

1. Is the work progress running according to the schedule?
2. Should the process be continued?

These questions are important for monitoring the project status and provides a platform for sponsors to express their point of views on the progress. Based on the answers of the above questions, the sponsors are questioned regarding the improvements he/she is expecting. Programmers should consider the suggestions given by the sponsor seriously and work on it. The team should work in a way that the sponsor should not answer 'no' to the second question. If the sponsors answer 'no' to the second question, the project will usually be terminated. The only possibility of continuation is to find out the problems by speaking with the sponsor.

**(ii) Weekly Deployment**

The programmers should develop an iteration demo for every week. This demo should be real enough that stakeholders can work on it. This demo should be created irrespective of the interest of stakeholders. If the team fails to create a demo, then the project is considered to be in trouble. Moreover, the time involved in product completion and delivery is considered as a major risks because it can take more than the time expected for the delivery. To avoid such time consumption, weekly deployment is necessary.

The team that successfully makes the code to be deployed weekly, eliminates the chances of facing technical debt. Moreover, it makes the code releasable in each iteration. However, it is not necessary to be deployable in case of legacy codebase due to its incompleteness.

## 2.8 REPORTING

**Q23. Who need reporting? Explain the types of reports.**

**Answer :**

Model Paper-II, Q5(b)

### Reporting

Reporting is usually needed by the programmers to show the project status to the higher authorities and stakeholders. The management and stakeholders need these reports to keep track of the project in which they have invested.

### Types of Report

Following are the major types of reports used in software development,

1. Progress reports
2. Management reports.

#### 1. Progress Reports

Progress reports refer to the type of reports which are used for indicating the current progress of the team. It includes information regarding iteration demo and delivery of product. These reports also help in building trust on team from the stakeholders point of view. The different types of progress reports generated by the team are as follows,

##### (i) Vision Statement

Vision statements are created by on-site customers and are updated on a regular basis. These statements include the details like the main intent of product, the tasks under process, the reasons for performing the current task and the ways in which it can be completed successfully. These statements are considered as reference in the meetings.

##### (ii) Weekly Demo

Weekly demos increase confidence among team members. It includes detailed information regarding the work progress.

##### (iii) Release and Iteration Plan Boards

Release and iteration plan boards are installed in the workspace which provides information regarding the work progress with respect to the next iteration and the product delivery dates. These boards help the stakeholders to track work progress.

##### (iv) Burn-up Chart

Burn-up charts provide the complete blueprint of the project along with its expected completion date. These charts are created while updating the release plan.

#### 2. Management Reports

Progress reports refer to the type of reports which are used for analyzing goals and trends set by the management. It also includes the information regarding the defects rate. The different types of management reports are as follows,

##### (i) Productivity

Productivity reports includes the role of the team in increasing the productivity of business. It is evaluated in terms of different measures like return on investment, savings, revenue etc. This measure is completely based on the metrics which important to the management or organization. Initially, it is considered that the team is not returning any revenue but this figure will increase as the project proceeds.

##### (ii) Throughput

Throughput refers to the count of features offered by the development team in a specific time period. This information is different for different tasks/features.

##### (iii) Defects

Defects tracking records the defects in the development process. It is performed by recording the date of entry and closure for every defect. The rate of defect introduction and closure can be checked by project manager at any instance of time. This helps in ensuring that the difference between the amount of defects introduced and closed are in control.

##### (iv) Time Usage

Time usage reports help the stakeholders in identifying whether the team under pressure, utilizing its time efficiently or not. With this comparison, stakeholders can match the velocity with efforts.



**Q24. Discuss about the reports that can be avoided.**

**Answer :**

Some of the reports that can be avoided are as follows,

**1. SLOC and Function Points**

SLOC (Source Lines Of Code) and function points are measuring units used for measuring the size of software. However, they are sometimes used for measuring the productivity which is an incorrect choice. This is because, the use of SLOC makes the programmer focus on increasing the number of lines in a code instead of focussing on the productivity and the number of bugs. In reality, the number of bugs is directly proportional to the number of lines in a code. Therefore, generation of reports using these metrics should be avoided specifically in the weekly reports.

**2. Number of Stories**

The use of number of stories to measure the productivity should also be avoided. The reason for this is that the stories are modified accordingly based on the requirements by the programmers. If they are used in the estimation of productivity, the programmers will split the stories even if it is not needed to show that they did a productive work.

**3. Velocity**

Velocity can be used as a measure of calculating productivity. However, it incurs additional cost if it is considered as primary focus. Therefore, use of velocity as a productive measure should be avoided.

**4. Code Quality**

There are no specific metrics for estimating the quality of code. They only provide the piece of code where the code needs to be reexamined. Therefore, such metrics should be avoided as they can mislead the stakeholders.