# Task-1

According to the scenario the explained case is unethical because they are going to develop the meeting fixing calendar which will track the activities of all the department heads. Where all the low level employee and other people will also be able to know the company departmental heads activities which is also the confidential data of the company.

Instead of that the managing director could have:

* Taken the advices from the client to know with what features and functions are need to be included on the system so that they fell comfortable to use.
* Consulted with client about the security of data protection.

Defending confidential information, delicate data of any organization is known as data protection.

Explaining the principal of data protection in the scenario are as follows:

* The employee should be informed before accessing their personal information and also they must be given them right to refuse it.
* After accessing the employee’s data it must be secreted or shattered.
* Only after informing the employee about using their private information, they can take employee data for the development of meeting fixing system.

ICCP

1. A high standard of skill and knowledge.
2. A confidential relationship with people served.
3. Public reliance upon the standards of conduct and established practice.
4. The observance of an ethical code.

# Task-2

The execution phase is the longest and the third phase on the agile software development life cycle and also this is the phase where developers should invest must of their effort in the project.

I have chosen the agile methodology thinking that this methodology is suitable for this project because agile get faster development process and also has quicker development approach in an operational and well-organized manner. Whereas software development life cycle (SDLC) has orderly approach to developing software in the direction of prosperous product within the timetable.

And Method which is based on iterative and incremental approach where solutions develop over the teamwork between self-organizing universalistic teams is known as Agile.

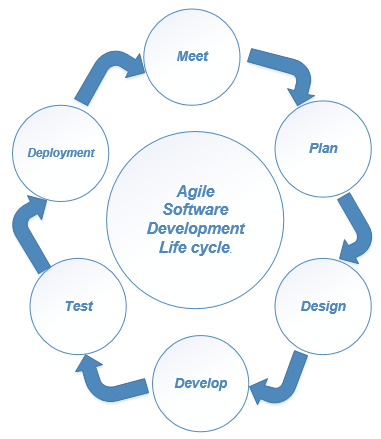


Fig 1: Agile software development life cycle.

All the above agile phases are applied to the scenario:

1. **Meet:** The managing director, department heads and the project manager are the main participants involved. But the end users should give the information to the main participants to include functions into the system.
2. **Plan:** All the department heads including the analyst, developer and end user should plan form where the project should start and then what functions should be included in the meeting fixing calendar. Like meeting length, meeting place and when the meeting should be taken etc. The department heads, managing director, project manager, programmer, analyst and end users are main participants involved here.
3. **Design:** Then the developers should start to make the prototype of the calendar and the real design should only start after end user confirms that. The developer are the main participant involved here.
4. **Develop:** After the completion of the design then the programmer start to integrate the functionality into the system. The programmer and project manager are the main participant involved here.
5. **Test:** When the programmers finish development parts then ‘Quality Assurance’ test the internal and external functionality of the system to assure whether the system’s function works efficiently, smoothly and well secured. The programmer and QA are the main participants involved here.
6. **Deployment:** After the completion of all the testing the team members carried out the demo version to the client and then they provide the feedback if the feedback response is positive then the system will be installed into their system otherwise again the above phase start from initial point.

The three advantages of agile methodologies are:

1. **Continuous involvement:** if departmental heads involve client in project all the time then their relationship to the product is going to evolve and bond to the product is going to increase where they will have greater ownership to the product.
2. **It is iterative development:** when the project is checked again and again it not only reduces the financial risk but also reduce the struggle over unnecessary features.
3. **Easier maintenance:** while developing the product there are multiple developers working in the each and every part of the system together which reduce the distinct point of failure, this is because of understanding of the code while working together in a team in every part of the system.

The three disadvantages of the agile methodologies are:

1. **Disjointed output:** When the team is working on the each modules in different series then the final product often became split rather than one solid unit.
2. **Incomplete documentation:** for developing products the documentation take place during a project but not at the start which becomes fewer detailed and often drops to the low priority.
3. **Additional time and energy:** The interaction between the developers and clients are compulsory. This includes frequent face-to-face discussion, which is the finest portion of the discussion. As a result, agile request more time and energy from every person.

# Task-3

**A**.

The five methods of identifying risk are as follows:

1. Ease of use of hardware, software, approaches and utensils.
2. **Functionality requirements:**
3. Technical requirements.
4. Customer contracts.
5. Development environment.
6. Known user.

**B**.

The explanation of the three possible risk based on part (a) are:

1. **Functionality requirements:** if the functional requirements is not identified then the risk could be rise in cost if some additional function should be add or improve in the system. for example:
2. **Known Users:** The team of TBTR Ltd should be given training if they are not at ease with using a computerized system.
3. **Technical requirements:** The condition of the computer would not permit the new system to function proficiently, if there is sarcasm in the size of completed system.

**C.**

**Proactive** refers to the activities that address observed risk before it actually occurs. It is usually maintained as the highest form of the risk management as proactive activities are observed before, if TBR Ltd use proactive then the team will be aware from the risk and can prepared for the reduction of risk which save their time, money and prestige as well.

**Re-active** refers to the movements in comeback after risk incident it is usually observed as the basic form of the risk management. if the organization uses the re-active then the team cannot prepare to reduce risk as re-active is done after the risk. So, the company’s will have high chance of losing time, money and prestige.

**D**.

The risk management framework according to the ISO 31000 refers to the traditional modules that maintain and tolerate risk management during the project. Replacing the all present principles, approaches to risk management and providing one common adjustment across all industries insensitive of business area are the objectives of the **Iso 31000**.

Currently the problem of the TBR ltd is resource management, not having have clear requirement so that product won’t finished in time or if finished also then the product will not developed as they want or the new developed product will not run in the current system or they are now facing the security problem. So, Iso 31000 should be used because it will reduce all this kind of problems.

* If the product need to be changed either in future or in present also then the product can be changed from the existing product they need not to develop new on again, because Iso 31000 is iterative, dynamic and then responsive to change and provide repetitive improvement of the organization. Which saves the organizations money and value.
* As per current situation company is having security and resource management problem. Talking about resource management if the organizations systems OS is outdated then the developed product won’t run in it. So, after using Iso 31000 it clearly address insecurity and then team would prepared to reduce and also manage resource by communicating with the client.

# Task-4

A.

The delivery of the last version of the application is known as software release. It is anticipate by the delivery of the beta and alpha version of the product. A decent release will make call and make a request of the genuine product before its official lunch. The release of product get valuable reactions from the user so it is important from the technical viewpoint. Which provide the time to fix bugs, reduce errors etc. before releasing to the public. There are four software development methods which are as follows:

* **Web-based methods:** it is typical in software organization because it does not require installation and is consistency. We are developing a desktop application not a website. So, web methods in not suitable for software release.
* **Pilot methods:** it is a distribution of production application but earlier in time to a full release of the application. This method is suitable for software release because it minimize the impact and identify the errors before installation.Due to this there won’t be any other errors in future regarding the software because it checks the errors before installation on the client system.
* **Big bang methods:** it refers togetting relieve from the current system and shifting all the users to the new system. It is not feasible to release software because it does not reduce impacts and taking about the data protection it is not secure as it shift all the users to the new system and if there will be any issue in the product in future then there will be problem in data, cost and in time. So, that it is not feasible to use.
* **Parallel methods:** it is a policy for a system implementation where new system runs next to old one while system equally works at the same time. This methods is not possible to use in software release because the new software will be totally different with the organizations old software where the two software will not be user friendly.

**B.**

Finally, I have chosen the pilot method for software release because it minimize the impact and identify the errors before installation.Due to this there won’t be any other errors in future regarding the software because it checks the errors before installation. It also give chance to the project team to weigh the capability of the new application and make changes that make more efficient, which improve the process of the system.

# Task-5

A.

The advantages of the outsourcing for the development of the new computerized calendar system are as follows:

1. **Diminish the risk of project failure:**

There is the contribution of expert and experienced team which diminish the possibility of project failure. That concentrate on TBR Ltd.’s central business activities and provide more profits.

1. **Additional arrangement is not needed:**

The TBTR Ltd should not invest cash for additional infrastructure to complete the project because the development center has all basic infrastructure.

1. **Great level of trustworthiness and effectiveness:**

Due to the contribution of the expert team and finest technology as explained above, this confirms less faults and rework.

The disadvantages of the outsourcing for the development of the new computerized calendar system are as follows:

1. **Possibility of information sharing:**

The organization should share the information with another person after signing the agreement with the third party where there is high chances of losing the confidential data.

1. **Timing problem:**

There is five to twelve hours’ time gap with several countries. In this situation, there will be communication problem between the two countries.

1. **Problem of language and culture between two countries:**

The problem of language and culture between two countries might lead to confusion which will effect on output of the project.

**B**.

In-house development should be used because if the organization outsource the product then the organization will lose the managerial control. Where the company won’t get the final output form the team as like getting from own team. Hence, the TBTR Ltd need to possess control of their system. Likewise, in-house development provide the autonomy to choose what function should be include and not to include in the system as well as can modify anytime. So, the project team can develop the product according to TBTR Ltd.’s needs.

And again the next thing is there is time gap between the team and the company while outsourcing the product but in in-house there won’t be any communication gap between the team and company. The company can communicate any time, either to give feedback or to modify the product and also to know the development progress status.

Similarly, the analyst should not have to write code from the starting because they might already have some re-usable code which saves the organizations time, effort etc. it also lets separate form the organization’s opponents because as per organizations desires the system is being developed.

Again, the team can immediately spot the issues because developing the software using in-house development empowers project team to observe all the phases in the process. While in outsourcing there won’t be any chance to observe the each steps which leads to cause more bugs and product also won’t developed as the organization needs.

The in-house software is developed by the own team of the company which provides familiar support. Instead of dealing with developers who won’t understand the project heads unique situation but the team head can get support from the own team and reduce the interruption from technical errors. Which will make the software familiar and user friendly because of having personalized software and developed from the own team too.

Finally, finally there will be team’s full loyalty and effort to the organizations project only because there will not be any other client and product will be developed by the own team. Due to this the product will be developed in time, it will be efficient and reliable, on the other hand there will be full support from the team and also no need to hire other for maintenance.

# Task-6

**A**.

The explanation of the two ways to measure the quality of software are as follows:

1. **Quantitative:** Quantitative measure is based on the actual program where the measurement of loading time of program and performance is easy, as per faults per line of code. It can also be collected from inspections, questionnaires or from program tracking tools.
2. **Qualitative:** Qualitative measure is based on the judgment of somebody which is also very difficult to measure. To decide the quality of software the standard is place and hence, the quality of the software is decided via the result of quantity analysis. So, for taking the circumstance nearby the results of the program it is frequently used.

**B**.

By mixing both of them would be beneficial to measure the quality of the new TBTR Ltd system. As quantitative measure is based on the actual program where the measurement of loading time of program and performance is easy, as per faults per line of code. And It can also be collected from inspections, questionnaires or from program tracking tools. At the end, the system analyst and the team of programming will work to measure the quality of the new system of TBTR Ltd.

Whereas the Qualitative measure is based on the judgment of somebody which is also very difficult to measure. And for taking the circumstance nearby the results of the program it is frequently used. So, to decide the quality of software the again standard is place and hence, the quality of the software is decided via the result of quantity analysis. At last, the system analyst and the team of programming will work to measure the quality of the new system of TBTR Ltd.

**C**.

**D**.

Explanation of measurement of the quality of software by ISO 1926.

* **Functionality:** It is the important purpose of any product which depends on the software complication and helps to define the come across the necessity of the client. As the software has capability to fulfill a task and keep to its persistence. Now the software developed for TBTR Ltd should able to:
* Fix the meetings with meeting participants.
* Type the participant’s details with the time interval of the meeting.
* Fix where and when the meeting need to take place on a sequence of dates which every person is free at the similar time.
* **Reliability:** It decides the ability of TBTR Ltd.’s software to tolerate its usage when putting under the different situations.
* **Usability:** The software’s usability depends on the functional use of software. Likewise, TBTR Ltd.’s calendar system is not influenced by entering the details of the meeting participants, with the time interval to fix the meeting.
* **Efficiency:** It is concerned with the system source used when providing the required functionality. The amount of disk space, memory used by the calendar system delivers the worthy signal of this measurement.
* **Maintainability:** The capability to detect the fault, bugs inside calendar system developed for the TBTR Ltd.’s and fixing that fault is what speaks about the maintainability. The consistency or difficulty of the code impact the maintainability of the software.
* **Portability:** it defines how nicely the software developed for TBTR Ltd can be adopt modifications with its requirement easily.