Question 1.

The waterfall model is the simplest model of the software development paradigm. It says the all the phases of SDLC will function one after another in a linear manner. That is when the first phase is finished then only the second phase will start and so on. We often describe Waterfall as a 'linear-sequential lifecycle model'. This means that it follows a simple structure of phases where the results of each phase cascade down to the next level of development. In other words, we're not so much looking at one big Niagara Falls, but a series of cascading waterfalls – each with its little pool of activities. The iterative model, says the all the phases of development will linearly function one after another. That is when the first phase is finished then only the second phase will start and so on. This model leads the software development process in iterations. It projects the process of development in a cyclic manner repeating every step after every cycle of the process. Then, on every next iteration, more features and modules are designed, coded, tested and added to the software. After each iteration, the management team can do work on risk management and prepare for the next iteration. Because a cycle includes a small portion of the whole software process, it is easier to manage the development process but it consumes more resources.

Information security requirements refer to the requirement needed for the Protection of information systems against unauthorized access to or modification of information, whether in storage, processing or transit, and against the denial of service to authorized users or the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats.

Waterfall requires everything to be described in written documentation before any code is created. This approach makes it crucially important that all requirements are meticulously defined from the start, as it might be difficult to revise them later. Since technical documentation is a necessary part of the initial requirements phase, this means that everyone understands the objectives. It is easy to avoid potential attacks by first analyzing them in the beginning.

Each step in the waterfall model has a clearly defined starting point and conclusion, which makes progress easy to monitor. This helps reduce errors that may lead to a potential security breach.

Security needs can be difficult to define; it is often challenging to conceptualize the security needs in terms of a functional specification during the requirements phase. Some functionality may evolve which can be difficult to control.

Potential lack of flexibility; there may be issues with the flexibility of the model to cater for new security requirements or changes of security requirements that may occur after the initial consultation. Changes due to potential security breaks may not have been taken into account when planning is all done upfront.

This model does not work smoothly if there are some issues left at the previous step. The sequential nature of the model does not allow us to go back and undo or redo our actions.

One of the advantages of the iterative model is that it produces a working prototype early on in the project. As it is being reviewed and discussed, it's possible to isolate flaws in functions or design. Finding these issues at an early stage may help to address them quickly within a tight budget.

The iterative model allows most risks to be identified during the iteration and higher risks can be dealt with as an early priority.

The development isn't restricted to single modules, and there can be more than one iteration in progress in the development cycle at any given time.

Progress is easily measured

It can sometimes be difficult to find a highly-skilled talent is required for analyzing the potential security breach. It requires a highly skilled development team to help steer the project clear of risks and prevent the project from stalling.

However, Iterative Development is only suitable for larger-scale projects as it may not be possible or realistic to break down small projects into even smaller components.

More resources may be required

Therefore, waterfall development is generally recommended for projects that are not expected to change or to need new developments during the project lifecycle. When you work with an Iterative Development model, the starting point doesn't need a full requirements specification. The process starts with the design and development of just a limited part of the software, and the iterative design process means that you return to expand and enhance this material repeatedly – until the entire system has been implemented and deployed.

Question 2.

Information security requirements refer to the requirement needed for the Protection of information systems against unauthorized access to or modification of information, whether in storage, processing or transit, and against the denial of service to authorized users or the provision of service to unauthorized users, including those measures necessary to detect, document, and counter such threats.

Requirements, which are related to the functional aspect of software fall into the functional requirement category. They define functions and functionality within and from the software system. Let us look at some examples, Search option is given to the user to search from various invoices, the user should be able to mail any report to management.

Requirements, which are not related to the functional aspect of the software, fall into the non-functional requirement category. They are implicit or expected characteristics of software, which users assume. Non-functional requirements include security, performance, and reliability.

Information security is achieved by implementing a suitable set of controls, including policies,
Processes, procedures, organizational structures and software and hardware functions leaving any of these detail may lead to the vulnerability of your system that can be exploited by one or more threats.

The process of specifying non-functional requirements requires knowledge of the functionality of the system, as well as knowledge of the context within which the system will operate.

Leaving any security detail on a non-functional requirement can lead to a potential security breach, the consequences can include the inability of users to access a system, allowing the wrong people to access confidential information, and can cost organizations millions of dollars.

Privacy violation; a strong security requirement imposes the use of cyphering methods including Caesar cyphers, classic cyphers and many more. Those ciphering methods impose the use of data encryption which can dictate protection for sensitive information. The failure to do so may result in a passive attack which is difficult to detect as it doesn't involve any alteration of data. One of the three basic areas of focus when it comes to security is confidentiality, security personnel are urged to prevent at all costs anything that can lead to confidential data being leaked. Encryption has been one of the most common methods for preventing those passive attacks.

When a malicious user penetrates your system, they look for vulnerabilities in your system whether it is functional or non-functional. They take advantage of the security weakness and exploit your system. This can lead to multiple damages in your system.

Provision of service to unauthorized users; once hackers get into your system they give access to unauthorized users hence that access (malware, spyware) limits our ability to protect the confidentiality of the data.

Loss of data; whether you like it or not If by chance a malicious user hijacks into your system there will be a huge loss of data. As far as I know, data is the greatest asset of the organization once the malicious user gets away with them it can cause a lot of damage to the organization as well as stakeholders.

Deliberate software attacks occur when an individual or group designs and deploys software to attack a system. Most of this software is referred to as malicious code or malicious software, or sometimes malware. These software components or programs are designed to damage, destroy, or deny service to the target systems. Some of the more common instances of malicious code are viruses and worms, Trojan horses, logic bombs, and back doors.
Prominent among the history of notable incidences of malicious code is the denial-of-service attacks conducted by Mafiaboy (mentioned earlier) on Amazon.com, CNN.com, ETrade.com,
ebay.com, Yahoo.com, Excite.com, and Dell.com. These software-based attacks lasted approximately four hours, and are reported to have resulted in millions of dollars in lost revenue.9 The British Internet service provider Cloudnine is believed to be the first business “hacked out of existence” in a denial-of-service attack in January 2002. This attack was similar to denial-of-service attacks launched by Mafiaboy in February 2000.

Countless risks come with neglecting the importance of non-functional security requirements. But there is something one can do, which is imposing strong security patterns for both functional and non-functional security requirements leaving no room or system vulnerabilities for the potential threats.

Q3.

Present-day information systems are vulnerable to a host of threats. What is more, with the increasing complexity of applications and services, there is a correspondingly greater chance of suffering from breaches in security. In our contemporary Information Society, depending as it does on a huge number of software systems that have a critical role, the Information system must be ensured as being safe right from the very beginning.

The biggest problem, however, is that in the majority of software projects security is dealt with when the system has already been designed and put into operation. Added to this, the actual security requirements themselves are often not well understood. This being so, even when there is an attempt to define security requirements, many developers tend to describe design solutions in terms of protection mechanisms, instead of making declarative propositions regarding the level of protection required

A very important part of the achieving of secure software systems in the software development process is that known as Security Requirements Engineering, which provides techniques, methods and norms for tackling this task in the IS development cycle. As a project manager, it is vital to involve a person with security knowledge in the early stage of the development.

The use of repeatable and systematic procedures to ensure that the set of requirements obtained is complete, consistent and easy to understand and analyzable by the different actors involved in the development of the system. a good requirement specification document should include both functional requirements and non-functional. As far as security is concerned, it should be a consideration throughout the whole development process, and it ought to be defined in conjunction with the requirements specification.

Security requirements are identified by a methodical assessment of security risks. Expenditure on controls needs to be balanced against the project harm likely to result from security failures.
The results of the risk assessment will help to guide and determine the appropriate management action and priorities for managing information security risks, and for implementing controls selected to protect against these risks. Risk assessment should be repeated periodically to address any changes that might influence the risk assessment results.

Security expert helps in the early project development stages ensuring that every detail is taken into account with respect to security. Security expert person might help in implementing the project guidelines, information security policy, and activities that reflect project objectives, an approach and framework to implementing, maintaining, monitoring, and improving information security that is consistent in the project.

Stakeholders and all people involved in the project are provided with appropriate awareness, training, and education by someone whose background is security. Everyone involved in the project needs to be trained and aware of information security, but not everyone in the project needs a formal degree or certificate in information security that is why we need appropriate individuals in security.
Security training involves providing project stakeholders with detailed information and hands-on instruction designed to prepare them to perform their duties securely. Management of information security can develop customized in-house training or outsource the training program. one of the least frequently implemented, but the most beneficial programs is the security awareness program; designed to keep information security at the forefront of the users' minds, need not be complicated or expensive. If the program is not actively implemented, employees begin to 'tune out', and the risk of employee accidents and failures increases

We cannot guarantee that our project will be risk-free; that is why our security expert must address actions needed should an unexpected risk become a problem. And some risks are simply part of a project. When we acknowledge that a significant problem cannot be prevented, we can use controls to reduce the seriousness of a threat. For example, you can back up files on your computer as a defense against the possible failure of a file storage device. Through a risk assessment, threats to assets are identified, vulnerability to and the likelihood of occurrence is evaluated and potential impact is estimated.

It is the security expert's role to protect the data that the project collects and use. Without data, an organization loses its record of transactions and/or its ability to deliver value to its customers. Any business, educational institution, or government agency operating within the modern context of connected and responsive services relies on information systems. Even when transactions are not online, information systems and the data they process enable the creation and movement of goods and services. Therefore, protecting *data in motion* and *data at rest* are both critical aspects of information security. The value of data motivates attackers to steal, sabotage, or corrupt it. An effective information security program implemented by information security protects the integrity and value of the organization's data.
We can't neglect the role of security experts in the project development as I've outlined. A project to be successful must put into place security guidelines throughout the project early stage of implementation and that's a project manager's duty to hire the person who can do such work.

Q4

Many popular applications from the 90s are not available on the market anymore. New Internet users will never hear about RealPlayer or ICQ—products that millions were using just ten years ago. One reason they're gone is that a plethora of new features turned those simple, usable applications into hulking space stations, resulting in a bad user experience. So adding features, new requirements may cause an intentionally misuse, we want firstly to know what is requirement? is described as the function which the system as a whole should follow to satisfy the stakeholder needs, it is commonly used in engineering design such as software engineering, enterprise engineering, or system engineering, and other important key that should be explained before is system which means combination or set of components, like people, machines, assets, that are related and work together on transaction, processing, to achieve the same goal. And sometime new requirement, or feature or use case may be intentionally misused meaning using them in wrong manner with the wrong purpose, that may cause someone to think about the following thoughts before; insecurity on existing system and the insecurity that may come up with the new feature, thinking about new features or use case, need to be documented because it may cause intentionally misuse, making the system complicated than how it was, complicating the existing system, and new feature may lead the equipment feature, so it is very important to think about new requirement, or accepting new feature.

At the beginning, we should know what the system insecurity is? A system insecurity can be described as an attempt to disrupt or damage the system. Adding new feature is good, some systems and users will love it, by loving new technology, and the owner will like how the new technology basically a feature is easing the tasks they want to do, however, the new feature can come up with new insecurities on existing system, those insecurities might be cyber-attack means creating opportunities for hackers to infiltrate systems and allowing to shut down application, or robbing the data. Wrong development phases, especially during its implementation phase, can bring the vulnerabilities on system at its level of security. Wrong development phase especially implementation of new feature, might lead the existing system to be insecure, this is potential issue why every owner should think about new feature before being accepted.

Secondly, it’s very important to think about new feature or requirement, creating new feature, or accepting the request of new requirement into the system is good, but, it may cause equipment failure, basically equipment failure refers to any event in which any equipment cannot accomplish its intended purpose or tasks, it may moreover mean that equipment stopped working or performing as desired. In here creating new requirement, or feature, data center physical infrastructure is always vulnerable to failure of some kind, making it one of the leading causes of downtime.

Thirdly, taking time to think about a new feature that can be created to the existing system may cause complexity of the whole system, being complicated will force the owner to provide a new well prepared document to the users or consumers that describe the whole system, and changes that happens to the system, before being accessed. When the owners delay to provide that document, the interaction with system will be complicated to them that may lead intentionally misused.

By concluding this, in my opinion, I strongly encouraging to take a long time to think about new requirement, feature or use case because it is extremely essential, due to keeping your system secure, protecting it from cyber-attacks, and once you think about them before adding new feature, will reduce complexity of system, and keep the equipment up and working properly.

References

Donald Firesmith, “Engineering Security Requirements”, Journal of Object Technology, Volume 2, no. 1 (January 2003), pp. 53-68, doi:10.5381/jot.2003.2.1.c6.

Firesmith, D.G.: Engineering Security Requirements. Journal of Object Technology 2(1), 53–68 (2003)

Karl Wiegers, Software requirement, 2nd edition, Microsoft Press, 2003.

Kotonya, G., Sommerville, I.: Requirements Engineering Process and Techniques. Hardcovered, 294 (1998)

Michael E.Whiteman, Herbert J.Mattord, Principles of information security, 4th edition, Course Technology, 2002.

Software engineering body of knowledge (http://www.swebok.org/)

Olchówka, B. (2014, March 25). *Beware of feature overload: A case study*. UXmatters. https://www.uxmatters.com/mt/archives/2014/03/beware-of-feature-overload-a-case-study.php.

Tom Banta on September 27, 2019, Server downtime, Common causes, and how to prevent them

https://www.vxchnge.com/blog/common-causes-of-server-downtime

System documentation. Definition: system documentation.

https://www.its.bldrdoc.gov/fs-1037/dir-036/_5261.htm#:~:text=system%20documentation%3A%20The%20collection%20of,computing%2C%20or%20information%20processing%20system.