**How to reduce the vulnerabilities in a system during software development?**

**Introduction**

In simple terms Software development is the process of creating Software in a specific language, this could be in the form of a mobile app, a game, or a piece of software you download and use on your PC or Mac. In more specific terms and how a programmer, coder or designer would explain it, it is an iterative logical process that’s main goal is to create a computer coded or programmed software that addresses an objective, goal or process that someone or a business might have. When it comes to Software development it is more beneficial to the individual or group that is making the software to plan everything out, to make sure they know what they want to make and when it needs to be made by, to minimize risk, and problems. This makes the whole process a lot more streamline. This plan will consist of several stages that when completed well, and to a professional standard will result in a useful, high quality piece of software ready for personal use or for a business to go on and sell the software for a price. Every good Software Development project will go through these stages. These six stages start with the requirement gathering and analysis stage which is arguably the most important stage as this is where the client will explain everything they want this app, software or system to do. They will go into huge detail on their goal/idea for this app, system or software and will make sure you are both on the same page when it comes to what the finished product should look like. If the project is within a larger business there will be meetings with stake holders, and project managers to determine: who will be using the system or app? And how the system will be used? These are general questions that will be asked and discussed. The second stage is the Design stage. This is where the requirements and analysis will be written up into a design document. This document will also consist of all detailed plan of what the system or app will look like and how it will work. Coding of the project, testing the code, giving back the finished product, and the final stage which can last years; is the maintenance stage. This is a stage which people miss out when speaking about Software development as they do not take into consideration that there is life after the product is handed back to the client. With most client based projects; the client will want you to design, develop and build an app that will be used for years by the public. This is when they will want you to provide updates for the app. These may be every week, month or year. These updates normally consist of bug fixes, small new features and checking up on the app to make sure it is running how it should.

**Software development life cycle:**

The SDLC (software development life cycle) can be created using several different models these models are Waterfall, Prototype, Spiral, RAD (Rapid application development), Agile and Formal. Additionally, I will be speaking about them in this order. Waterfall development has specific goals for each phase of development. Each phase has to be completed before moving on to the next. At the end of each phase, a review is taken place to determine if the project is on the right path and whether we should continue or discard the whole thing. There are 6 phases of the Waterfall model and they are: requirements, which are where the potential requirements of the application are methodically analysed and written down in a specification document that serves as the basis for all future development. Secondly is, Analysis and during this phase the system is reviewed/analysed so we can properly generate the business logic and models that will be used later in the actual application. Next is the design phase this stage covers technical design requirements such as data layers, services and programming languages. Coding is next where all of the core coding is written implementing all models, business logic and service integrations that were specified in the prior stages. The next phase is the testing phase where beta testers and all other testers systematically discover and report any issues they find with the application. Lastly, is the Operations phase, the operations stage entails not just the deployment of the application, but also subsequent support and maintenance that may be required to keep it functional and up-to-date. Some advantages of Waterfall are that it is very simple and easy to understand and use, phases do not overlap as each one is completed before moving onto the next, it is one of the best models to use for smaller projects where requirements are very well understood and finally it is easy to manage because of the rigidity of the model. A few disadvantages of the Waterfall model are it is not good for projects that are complex and object-oriented projects, it is a model that contains high amounts of risk and uncertainty, one the project/application is in testing stage it is very hard to go back and change something that was not well-thought out in the concept stage, it is not a good model for projects that are long and on-going and there is no working software produced until one of the last phases within the life cycle. A good time to use the Waterfall lifecycle is when all of the requirements are well known, clear and there are no ambiguous requirements. Additionally, a good time to use this model is when the product definition is stable. Finally, the perfect project to use this lifecycle in is a small/short project. The main idea of a Prototype model is to create a throwaway prototype to understand the requirements of the project. The prototype is developed based on the requirements we know. By using this model, the developers can get a ‘feel’ of the system; since the interactions with prototype can enable the client to better understand the requirements of the desired system. The advantages of Prototyping are users remain actively involved in the development process, it has quicker user feedbacks than most models leading to better solutions and lastly, confusing or difficult functions can be easily identified. The disadvantages of Prototyping are: practically this methodology might increase the difficulty of the system; as the scope of the system may expand beyond original plans, incomplete or inadequate problem analysis and an incomplete application may cause the application no to be used as the fully system was designed. Some good examples of when you should use the Prototype model are when the desired system needs to have a lot of interaction with the ends users. Additionally, online system and web interfaces have a very high amount of interaction with end users and these are best suited for Prototype model. Lastly, Prototyping makes sure that the end users constantly work with the system to provide feedback which is incorporated in the prototype to result in a usable system. A couple projects that would benefit and give the best result when using the prototype model, would be when the desired system of the project needs to have a lot of interaction with the end users. Normally web interfaces and online systems both have a lot of user interaction so these types of projects would benefit using the prototype model. The Spiral model emphasises the importance of risk analysis. There are four main phases of The Spiral model which include: the planning phase things like Business Requirement Specifications and System Requirement specifications are gathered to make sure you know the ins and outs and what needs to be done/happen within a project. The risk analysis stage which for the Spiral model is one of the most important phases. Within this phase a process is done to identify certain risks you may occur during a project and alternate solutions you can take to avoid these risks or make them easier to deal with. Towards the end of the Risk Analysis phase a prototype will then be produced to help further with finding any bugs, problems and to see how the prototype looks/feels. Thirdly, we have the engineering phase which is where software is developed and a lot of testing is done within the software to make sure it properly works. Finally, is the evaluation phase where the customer will evaluate the output of the project to date before the project continues. There are a few advantages of using the Spiral Model; one being the high amount of risk analysis pays off greatly when it comes to the performance and the quality of the project, it is good for large and mission-critical projects, additional functionality can be added later in time, there is a strong approval and documentation control and lastly software is produced early in the software life cycle. With advantages come disadvantages, these being: it can be a very costly model to use, the high amount of risk analysis involved requires a highly specific expertise, it is not a good model for smaller projects and lastly the project’s success is highly dependent on how the risk analysis phase goes. A few examples of when you should use the Spiral Model, would be when cost and risk evaluation is important within a project, for medium to high risk projects, when the user is unsure of the needs they want for their project, when their requirements are complex and finally for a new product line. Rapid application development (RAD) heavily emphasizes rapid prototyping and iterative delivery; because of this it is good alternative to the Waterfall development model. There are a lot of advantages of RAD but these are just a few crucial ones you should take into consideration. Firstly, it is easy to measure progress as there are so many components, iterations, and prototypes coming down the pipe, and so much information and feedback being given back everything can be maintained and evaluated to make sure you are keeping schedules and budgets. A large percentage of active software developers will switch to the multi-discipline roles, a Rapid application development methodology allows skilled team members to quickly produce prototypes and code that works to illustrate examples that using a slower development technique could take weeks. While most models wait until the end to begin integrations, an application will be integrated pretty much instantly. While developing software is a fairly malleable form, since code can be altered that can change the entire system. It is an advantage to the developers to use this flexibility early on and use it a lot; by iterating concepts or ideas while developing. On the other hand, there are a few disadvantages of RAD, one being that even though RAD gives the developers a lot of flexibility throughout the product's lifecycle; it reduces the control and restrictions which in reality is not all bad but on large projects it can be difficult. When using RAD the team will need to consist of a very skilled group of individuals that can adapt to things within the project evolving. Finally, the project that would benefit using the RAD model would be projects where a system needs to be modularized in 2-3 months. Also when the team has a high availability of designers for modelling and the budget of the project is high enough for the team to be able to afford the cost of the designers alongside with the cost of automated code generated tools. The Traditional x Agile model is based upon adaptability of changing product requirements and also enhancing customer satisfaction by rapidly delivering working product features and involving the client. One of its methods is to break up the entire product into easily developable features, developed through a cycle knows as a sprint. Lastly, there are a few frameworks within the Agile model, but they all boil down to the same idea of breaking down tasks, and responsibilities into smaller tranches. The advantages of the Agile model are the project is continuously improving and so are the team. The project can evolve a lot within this model so the team will adapt and overcome the obstacles they are faced with, improving the project. Additionally, if you the team you are dealing with is experienced change can be catered very well for even though Traditional X Agile has a short planning cycle. A few disadvantages of the Traditional X Agile model are that the planning of a target date consisting of x can be sometimes dubious and not secure. The team will need to consist of highly skilled members on the Agile framework chosen and cross skilled in competencies as the team can be very small. Lastly, time and effort are continually required from the products resources, this is essential to the cycle planning and success. The Agile model should be used when new changes need to be implemented within the project. Also unlike the Waterfall model, within the Agile model there is very limited planning that is required to get started with the project because the Agile model assumes that the end users' needs are forever changing in a dynamic I.T and business world. The Formal Methods are more of the complex models to use as they use rigorously specified mathematical models to build its software and hardware systems. Additionally, to ensure correct behaviour of the system, Formal methods will use mathematical proof as a complement to system testing and as safety becomes a more important issue and as systems become a lot more complicated the Formal approach offers another level of insurance. A few advantages of Formal methods are, because of their rigour they require an engineer to think about his design in a more thorough way; this thorough approach can help identify faulty reasoning far sooner than other traditional designs. Another advantage is the clarity that comes with Formal methods. Formal methods require very clear and defined goals, objectives and approaches, which gives the team a very good idea of what they want to achieve and how they want to go about doing it. For example, when in a safety critical system, ambiguity can be a huge threat to the system as a whole, which is why using a Formal method is better as it eliminates ambiguity entirely. A disadvantage of a Formal approach would be the cost. As there is far more rigour in Formal methods it c makes the approach a lot more expensive. Formal methods involve a large initial cost followed by less consumption as the project advances; this is the complete opposite of traditional models. Finally, most formal methods introduce some form of computational model, which restricts the efficiency of the operations allowed to make the notation elegant and the system provable or work well. These design limitations are usually considered unacceptable from a developer's perspective as it restricts their creativity, and skills.

**Software vulnerabilities:**

Software Vulnerabilities can create huge setbacks, loses and even game changing problems for any business. In this day and age every business will use some sort of technology within its day to day running. This can be in the form of a website, an app, computer and systems within the business and even the passes staff use to enter and exit the building; and each one of these pieces of technology can be hacked or used against the business by an individual or organisation. A software vulnerability is seen as a weakness, a ‘backdoor’ or a problem within the systems code and the system as a whole. These flaws can all be used by a hacker to change the normal running of said system. These hackers may delete lines of code, change lines of code or gain access to customer’s details via the database of the system; all having huge consequences on the business for possibly a simple mistake or error in the code or system that if looked over in more detailed would have saved them losing business, money, time, reputation and customers increasing the chance of the business failing and becoming bankrupt. These attacks can come in a number of ways and each one with a different level of severity. Firstly, we have the phishing scam, which is by far the easiest way to infiltrate a system. An email containing a malicious link is sent to an employee or individual that is within the company with the hopes they click the link and unleash the virus onto the system and computer. A way that has been around for years is simply guessing the password. According to a 2016 survey by SplashData some of the most frequent password used are sadly “12345”, “password” and “qwerty”. This leaves a lot of the country vulnerable to cyber-attacks. Lastly, we have A Trojan Horse. This type of malware is designed to hide its true identity from the user and system. It will normally hide itself as “an executable file in an email attachment or free software from an unsecured website”.

**What can impact the software process:**

Vulnerabilities within software development can cause expensive, time consuming problems for the team and therefore they need to be sussed out and dealt with before they cause any problems that can’t be undone. An example of one of these vulnerabilities that can impact the software development process is Social engineering. Today hackers try to focus more on the staff than the technology when they are trying to get their hands-on software or information. Hackers will try to convince or manipulate staff to ‘handing’ over password, security codes, and so on. A way they do is by breaking the employee in person finding out the persons weakness and leading them to either think about it or talk about it causing the worker to feel weak and more likely to give up and hand over passwords. Or on the other hand the hacker could spend time getting to know the employee and will try to build a bridge of trust with the hopes the employee tells the hacker a password or anything valuable. Lastly, if an employee is not properly told about phishing scams the hacker may try to send emails containing malicious links with the hopes that the employee opens this link at work and causes the hacker to gain access to the software. If a competitor is serious enough they may hire a hacker or do this themselves. If they find out a company is creating a new type of software. Secondly, we have race conditions, a race condition is when the order of two or more events is changed, if they need a correct type of order to function and they do not have this, it is bug. Hackers will exploit small gaps during the processing of code and try to insert malicious code and change the running of the software. As you can guess when a hacker changes the code of your software during development or insert malicious code within it, it is going to either break the software or affect it negatively. Additionally, another big problem when it comes to the software process would be a buffer overflow. A buffer overflow is when an application will try to write data past the beginning or end of a buffer, this can cause applications to crash, can change data, these can also create open windows for other attacks and buffer overflows are one of the highest sources of vulnerabilities. When creating software, it gives people huge pride coding everything themselves but when it comes to encryption and securing your software unless you are an expert in the field should leave it to others and try to outsource encryption software. This is a huge problem when people write their own code and encryption for their software as they think they are an expert but realistically they haven’t properly secured their software from hackers.

Each of the vulnerabilities I have mentioned can all be reduced or eliminated and here are the ways they can. Firstly, when it comes to social engineering there is a simple way of reducing this and that’s to make sure that within your company that every employee is taught about the ways that hackers will use them to gain access to projects and software. To maximise the amount of information you can give to your staff about preventing hackers from using social engineering against them is to take them all to a seminar all about social and engineering and how to prevent. These are a good way of exposing your staff to a huge amount of information and tips to help keep your business safe. Additionally, race conditions can be a huge problem for developers and an easy way to reduce or prevent them is to synchronize sections of the code with a synchronized block. Synchronized blocks work by only allowing one thread to execute inside them at any given time and any other thread that tries to enter the block will be held until the first thread finishes. Additionally, you will need to make sure the critical section is executed by an atomic instruction, this means that while a thread is executing it, no other thread is able to enter or start to execute until the first thread has left the section. Certain languages do not allow for buffer overflow so to prevent buffer overflow simply chose a language that does not allow for it. During development if you have already started and are aware buffer overflows are happening then try to use the functions ‘strcopy’ and ‘strcat’ these functions will often copy a string into a buffer and attach the information or contents of one buffer onto another. Lastly, when we are talking about developing software there is a huge pride that comes with being able to say I created that alone or we created that but when it comes to securing that software it can be a different story. Developers like to create their own software that will help defend or protect their other software this can cause them more problems than good and for that reason when it comes to preventing your software from being open to attacks buy software from companies that sell this kind of software. Buying software and not developing it yourself might be more expensive but you will be able to sleep at night knowing your software is secure and encrypted because you decided to buy it from a someone that knows what they’re doing.

**Case Studies**

Yahoo is similar to google, as Yahoo contains a search engine and an internet portal. Additionally, it incorporates a library of World Wide Web sites organized into a list of the top categories it has to offer. Yahoo provides users with access to “hundreds of thousands of webpages and websites”. With its staggering 1 billion monthly active users you can tell people love Yahoo, but when something is loved there are going to be people out there that either hate it or want it to be no more. Lastly, Yahoo is worth close to 5 billion which is an expensive company.

In 2016 while in talks with Verizon about selling to them they came out and stated that between 2013 and 2014 they were the victims of the biggest data breach in history. When asked about the hackers Yahoo stated that it was “likely by a state-sponsored actor”. Yahoo came out and said that “this attack had compromised the real names, email addresses, dates of births and telephone numbers of over 500 million users”. Additionally, Yahoo stated that “the vast majority of the password involved had been hashed using the robust bcrypt algorithm”. When the word had gotten out that this information had been leaked as you can guess the public were not happy at all. If this was not bad enough a few months after this attack that earlier records stated that there was a different group of hackers that had compromised a whopping 1 billion accounts, again with information like names, dates of birth, email addresses and passwords being leaked but this time security questions and answers were also leaked. A few years later Yahoo said that in fact all of the 3 billion accounts on Yahoo had been compromised. These attacks knocked a huge $350 million off of Yahoo’s sale price, meaning that Verizon and Yahoo came to an agreed price of $4.48 billion for Yahoo’s core internet business.

Ebay is a huge marketplace where people can buy and sell items. You make an account and can upload pictures and descriptions of items you wish to sell or can search for items you wish to buy.

In May of 2014 Ebay announced they had been the victims of a cyberattack that exposed names, addresses, dates of births and encrypted passwords of nearly all of its 145 million users. The company stated that “the hackers for into the company network using the credentials of three corporate employees and had complete inside access for 229 days, during this time they were able to make their way to the user database” As you can see these hackers must have been experienced to be able to get the credentials of three employees and then to have access for 229 days without being stopped. Fortunately, Ebay stated that “financial information, such as credit card numbers, was stored separately and were not compromised” which was a huge relief for a lot of Ebay users. Additionally, Ebay encouraged users to “change their passwords” just to make sure the hackers were gone and did not still have access to accounts. During this time, they received a lot of back lash about its lack of communication with its users and implementation of the password-renewal process.

**Analysing research:**

My secondary research consisted of researching two case studies that would help me answer the question ‘How to reduce vulnerabilities in a system during Software development?’. Within my report I spoke about the methodology of Software development and the key parts of any Software development project such as the six stages to any good project. For my Primary research I handed out a questionnaire to students that were in courses that were themed around computers to see if other students were taught and were aware about software vulnerabilities. My questions varied from “what is the definition of a phishing attack” to asking them about what password they felt was the stronger. I gave my questionnaire out to 20 students as I felt this would give me a variety of answers I could work with. As I mentioned before I asked the students what the definition of a phishing attack is and to my surprise only 40% of the students wrote the correct answer. This suggests that a lot of the students are vulnerable to a phishing attack, this being said it puts the college at risk as if one of these student who answered incorrectly were to open a phishing email at college it could disrupt the whole college system. Additionally, when asked about their password and if they use the same password for most social media accounts and accounts in general 50% of the student said they use the same password for most of their accounts which means if a hacker is able to access on account and find out the password they have access to every other account that student has making them extremely vulnerable of hackers gaining access to information about the students, the students would rather keep private. Which in essence can put the students wellbeing at risk, as well as their financial status being affect if hackers get their hands on card details. Additionally, I asked student to give me the definition of an SQL infection and 70% of them were able to give me the correct answer meaning they have bene taught about these types of attacks but not so much about phishing attacks which is arguable the easier attack to understand. A pattern I observed that was certain students knew the more difficult, harder answer but could not answer the simpler questions. For example, when asked about phishing scams 60% could not give me the correct definition where as 70% could give me the correct definition of an SQL injection, which a lot of people would say is the harder attack to understand but these students seem to understand the harder ones and not the basics, and in essence if you do not understand the basics of something there is no point in knowing or understanding the harder definitions/ knowledge. Another trend I noticed is that not many of the student are worried about being hacked and don’t take many precautions when it comes to going about their internet life. When I asked them “Write 2 precautions you take when downloading music, games, movies to avoid being hacked” only 35% of the students gave me a precaution they take, most of the student wrote something along the lines of “I don’t” or “none”. A few precautions they take are checking the URL to make sure there are no suspicious characters or symbols within it and checking a link before the click it to make sure it does not contain any suspicious characters or symbols. This I was disappointing as it showed only a small percent of them actually realises how serious and easy it can be for hackers to gain access to your system if you’re not careful. To back this up when I asked them if they had ever been hacked 80% of them admitted to being hacked at least once in the past which is just not good at all. In essence I feel my research has opened my eyes and showed me that students are not taught much about software vulnerabilities, how to prevent them and how to going about your internet life while being safe. Most students gave answers that suggests they are vulnerable to being hacked, which means they put their information at risk and the college’s information at risk, when the come to college as use the college systems.

**Recommendations:**

There are a lot of things you can do to prevent flaws, issues, bugs and exploits during the development process and I am going to give some tips/recommendations on how to reduce and prevent these vulnerabilities. Firstly, when developing a system its best to use an authentication mechanism that cannot be bypassed or changed. If not designed correctly they can be tampered with so make sure an expert sets it up. This mechanism will stop bypasses within your software. While creating your system you should identify the sensitive data that you will be holding within the software and how you should hold it, this will make it easier and safer when it comes to the finish project, as you will know where to put the sensitive data and how to hold it. A small recommendation that can go along way is just to think about the data you have and make sure it is explicitly validated and do not make implicit assumptions about said data. When assumptions are made about data hackers can exploit these situations and get what they want. During development it is important to arm your software with security levels and parts that are going to protect is now and years to come. During development I would recommend installing software that will protect your project from being bypassed or hacked, with a firewall. During development and in the future a good way to help prevent malicious code being injected into your software/system is to whitelist applications this approach will whitelist applications and not allow them to run on your computer or system, instead it only allows a single pre-defined set of applications to run on it. During development of software to avoid the wrong people getting their hands of the software, I would recommend that the software is in a place only accessible by the right people for example a room that only certain people can enter or a drive that only certain people can access. Additionally, you can add passwords to the drives and computer to stop them being accessed. An easy mistake that developers make is having messy code that is not laid out correctly and efficiently. An easy way to fix this is to add indentations and spaces between paragraphs of code this makes it easier to read, go back to and fix. Lastly, we have one of the best ways to prevent bugs and that is just to test the code. Testing code is the best and easiest way to see if the code works.

**Reflect/Conclusion:**

In conclusion I feel the research I have done and carried out has helped me a lot to answer the question “How to reduce the vulnerabilities in a system during software development?”. I have gained lots of information, tips and research to help me identify what things can be done to prevent hackers gaining access and how to decrease these vulnerabilities before anything catastrophic happens to my systems and information. For my research I asked 20 students that were in computer based courses about vulnerabilities and hackers. To my surprise not many of the answers I got made my confident that these students were taking the necessary action to keep them self-safe from attacks as well as them keeping the college safe when they use its computers. I hope from these recommendations they will learn to think about what they are doing first before they give a hacker the operand over them. For my secondary research I wrote about two case studies one being Yahoo and one about Ebay. With Yahoo they were involved in the largest data breach in history with over 3 billion accounts being compromised and information like, full names, email addresses, telephone number and date of births being leaked. Additionally, with the Ebay case study similar to the Yahoo case study, over 145 million accounts were affect and the hackers were able to have access to the information for 229 days which is absurd. These case studies made me realise that no matter how much money or how big you are there is no escaping a hacker, and arguably when you are bigger you attract hacker’s attention a lot more as the information you have is normally worth a lot more. The recommendations I have created should help all developers when it comes to preventing bugs and exploits within their code and system. Finally I feel this research has opened my eyes a lot more to what precautions I should be taking as well and has hopefully open the eyes of lots of new developers and even old developers.