

MC 310: Software Engineering Assignment

SUBMITTED BY:

ISHAN BHATEJA : 2K20/MC/061

SUBMITTED TO:

MR. JAMKHONGAM TOUTHANG
Department of Applied Mathematics
DELHI TECHNOLOGICAL UNIVERSITY

May 2023

Write essays on the following topics in about 1000-1200 words each.

The Risks of Using Virus-infected Computer Software.

In today's technologically advanced world, computer software has become an integral part of our daily lives. However, the internet is a vast and complex network of information that is not always trustworthy. One of the greatest risks associated with using computer software is the use of virus-infected software. Virus-infected software is malicious software that infects a computer system and can cause harm to the user in various ways. This essay will discuss the risks associated with using virus-infected computer software and how to prevent them.

The use of virus-infected computer software can lead to a variety of risks. Firstly, it can lead to the theft of personal information. Malicious software can be designed to track the user's online activity and steal sensitive information such as login credentials, credit card details, and bank account information. This information can then be used for fraudulent activities, such as identity theft and financial fraud.

Secondly, virus-infected software can cause damage to the computer system. It can cause the computer to slow down, crash, or even corrupt files and programs. This can lead to a loss of data and productivity, which can be especially detrimental for businesses that rely on their computer systems for day-to-day operations.

Thirdly, virus-infected software can also be used to launch cyber attacks on other computer systems. This can happen when the infected computer becomes part of a botnet, which is a network of infected computers that can be controlled remotely. Botnets can be used for a variety of malicious activities such as launching DDoS attacks, stealing sensitive information, and spreading malware.

To prevent the risks associated with virus-infected software, there are several measures that can be taken. Firstly, it is important to use antivirus software. Antivirus software can detect and remove malicious software from the computer system. It can also provide real-time protection by monitoring the user's online activity and alerting them if they come across any malicious websites or downloads.

Secondly, it is important to keep the computer system up to date with the latest software updates and security patches. Software updates often include security fixes that address known vulnerabilities in the software. By keeping the computer system up to date, users can ensure that they are protected against the latest security threats.

Thirdly, it is important to be cautious when downloading and installing software. Users should only download software from reputable sources, such as the official website of the software provider. They should also read the terms and conditions carefully before installing the software to ensure that they are not unwittingly agreeing to install additional software or malware.

In conclusion, the use of virus-infected computer software can lead to a variety of risks, including the theft of personal information, damage to the computer system, and the launch of cyber attacks. To prevent these risks, users should take measures such as using antivirus software, keeping the computer system up to date, and being cautious when downloading and installing software. By following these best practices, users can ensure that they are protected against the latest security threats and can use their computer systems safely and securely.

Computer Software Piracy and Its Impact on the International Economy

Computer software piracy is a significant global problem that affects the software industry and the international economy. Software piracy refers to the unauthorized use or distribution of copyrighted software, and it has been estimated to cause significant economic losses to software developers and the global economy. This essay will explore the impact of computer software piracy on the international economy, with a focus on specific examples.

The software industry is an essential part of the global economy, and the widespread piracy of software has a considerable impact on the industry's growth and profitability. According to a study by the Business Software Alliance, the global rate of software piracy in 2019 was 37%, resulting in an estimated economic loss of \$46.3 billion. These losses are significant and affect not only software developers but also the broader economy.

One of the most prominent examples of the impact of software piracy on the international economy is the case of Microsoft. Microsoft is one of the largest software companies in the world, and its products, such as Windows and Office, are widely used in homes, businesses, and government offices worldwide. The company has estimated that it loses billions of dollars each year due to software piracy, with China and Russia being the most significant contributors to these losses.

In response to these losses, Microsoft has taken several measures to combat piracy, including anti-piracy campaigns, legal action against infringing companies, and the development of new licensing models. Despite these efforts, software piracy remains a significant problem for the company and the broader software industry.

Another example of the impact of software piracy on the international economy is the case of Adobe Systems. Adobe is a leading software company that produces popular products such as Photoshop and Acrobat. The company has estimated that it loses millions of dollars each year due to software piracy, with high piracy rates in countries such as China and India.

To combat piracy, Adobe has taken several measures, including launching anti-piracy campaigns, working with local law enforcement agencies, and developing new licensing

models. Despite these efforts, the company continues to face significant losses due to piracy, highlighting the challenges of combating piracy in the global marketplace.

The impact of software piracy on the international economy extends beyond the software industry itself. Piracy can have significant effects on the broader economy, including job losses, decreased tax revenues, and reduced economic growth. For example, a study by the International Intellectual Property Alliance found that in 2019, software piracy in China resulted in the loss of over 300,000 jobs and over \$16 billion in lost tax revenues.

To address these challenges, governments and international organizations have taken steps to combat software piracy. For example, the World Intellectual Property Organization (WIPO) has developed international agreements and treaties aimed at protecting intellectual property rights, including software copyrights. Many countries have also developed laws and regulations aimed at combating piracy and protecting intellectual property rights.

In conclusion, computer software piracy is a significant global problem that has a considerable impact on the international economy. The piracy of software products such as those developed by Microsoft and Adobe causes significant economic losses to software developers and the broader economy. The impact of piracy on the international economy extends beyond the software industry, and it can result in job losses, decreased tax revenues, and reduced economic growth. To address these challenges, governments, software companies, and international organizations must work together to combat piracy and protect intellectual property rights.

Software Testing: Manual and Automated Web-Application Testing Tools

Software testing is a crucial step in the software development lifecycle that ensures the quality and reliability of software applications. Testing can be performed manually or using automated testing tools. In web application development, both manual and automated testing tools are used to identify defects and bugs in the application. This essay will explore the differences between manual and automated web application testing tools and their respective advantages and disadvantages.

Manual Testing:

Manual testing is the process of manually testing the functionality and features of a web application. This process involves testers using the application as an end-user and verifying that the application works as intended. Manual testing is a time-consuming process that requires a significant amount of effort from testers. However, it has some advantages over automated testing tools.

One advantage of manual testing is that it is more flexible than automated testing. Testers can change the test cases on the fly and test the application based on their experience and intuition. This makes it easier to identify potential issues that may not be caught by automated testing tools. Manual testing also allows testers to explore different scenarios and user workflows, which is difficult to do with automated testing tools.

Another advantage of manual testing is that it is more cost-effective for smaller applications or projects. Since manual testing does not require any additional tools or software, it is easier to set up and manage than automated testing tools. This makes it an ideal choice for smaller projects or applications that have limited resources.

However, manual testing has some disadvantages as well. It is prone to human errors, and testers may miss defects or bugs due to the large amount of testing required. Manual testing is also time-consuming and may not be feasible for large applications or projects.

Automated Testing:

Automated testing is the process of using software tools to test the functionality and features of a web application. This process involves using scripts or software tools to automate the testing process. Automated testing is faster and more efficient than manual testing and can be used to test large applications or projects.

One advantage of automated testing is that it is more accurate and reliable than manual testing. Automated testing tools can run the same tests repeatedly, ensuring that the application works consistently and efficiently. Automated testing tools are also less prone to human errors, and testers can catch defects or bugs that may have been missed in manual testing.

Another advantage of automated testing is that it is faster and more efficient than manual testing. Automated testing tools can run the same test cases repeatedly, saving time and effort for testers. This makes it easier to test large applications or projects and identify defects or bugs quickly.

However, automated testing has some disadvantages as well. Automated testing tools require additional software or tools, making it more challenging to set up and manage than manual testing. Automated testing tools also require skilled testers or developers to create and manage the test scripts, which can be a significant investment for small or medium-sized businesses.

Choosing Between Manual and Automated Testing:

Choosing between manual and automated testing depends on the specific needs of the application and project. Smaller projects or applications with limited resources may benefit from manual testing, while larger applications or projects may require automated testing.

Manual testing is an ideal choice for applications that have complex user workflows or user interfaces that are difficult to automate. Manual testing is also an excellent choice for applications that require exploratory testing, which is difficult to do with automated testing tools.

Automated testing is an ideal choice for applications that require testing to be done repeatedly, such as regression testing. Automated testing is also an excellent choice for applications that require testing to be done quickly and efficiently.

Conclusion:

In conclusion, software testing is a crucial step in the software development lifecycle that ensures the quality and reliability of web applications. Both manual and automated testing tools have their advantages and disadvantages, and choosing between them depends on the specific needs of the application and project. Manual testing is more flexible and cost-effective, while automated testing

Ethical Issues Involved in Software Project Management

Software project management involves planning, organizing, and overseeing software development projects. Ethical issues can arise during software project management due to conflicts of interest, inadequate communication, and other factors. This essay will explore the ethical issues involved in software project management and their impact on the development process.

One of the ethical issues involved in software project management is conflicts of interest. Project managers may have personal or financial interests that conflict with the interests of the project stakeholders. This can lead to biased decision-making and a lack of transparency in the development process. To address this issue, project managers should disclose any potential conflicts of interest and take steps to ensure that their decisions are objective and in the best interests of the project stakeholders.

Another ethical issue is inadequate communication. Project managers may fail to communicate effectively with stakeholders, resulting in misunderstandings and delays. This can lead to dissatisfaction among stakeholders and damage the reputation of the project team. To address this issue, project managers should establish clear lines of communication and provide regular updates to stakeholders. They should also ensure that stakeholders have the opportunity to provide feedback and ask questions.

A third ethical issue is the use of proprietary software. Proprietary software can limit competition and restrict access to knowledge and innovation. This can result in higher costs for users and limited access to technology for those who cannot afford it. To address this issue, project managers should consider using open-source software, which is freely available and can be modified and distributed by anyone. This can promote collaboration and innovation and lead to better outcomes for all stakeholders.

Another ethical issue is the use of unlicensed software. Unlicensed software can infringe on the intellectual property rights of others and expose the project team to legal liability. To address this issue, project managers should ensure that all software used in the project is properly licensed and that the project team is aware of the consequences of using unlicensed software.

A fifth ethical issue is the use of offshore development teams. Offshore development teams can offer cost savings and access to specialized skills, but they can also present cultural and language barriers, as well as legal and ethical concerns. To address this issue, project managers should ensure that offshore development teams are properly vetted and that they adhere to ethical standards and legal requirements. They should also provide training and support to help overcome cultural and language barriers.

Finally, a sixth ethical issue is the use of artificial intelligence (AI) and machine learning (ML) in software development. AI and ML can offer significant benefits, such as improved efficiency and accuracy, but they can also present ethical concerns, such as bias and discrimination. To address this issue, project managers should ensure that AI and ML are used ethically and that they are transparent in their decision-making processes. They should also establish protocols for addressing any ethical concerns that may arise.

In conclusion, ethical issues can arise during software project management due to conflicts of interest, inadequate communication, the use of proprietary or unlicensed software, offshore development teams, and the use of AI and ML. Project managers must be aware of these issues and take steps to address them to ensure that the development process is ethical and meets the needs of all stakeholders. By doing so, project managers can build trust with stakeholders and promote the success of software development projects.

Indian Software Industry: Growth Patterns, Constraints, and Government Initiatives

The Indian software industry has seen significant growth over the past few decades, making it one of the most important sectors of the Indian economy. This essay will explore the growth patterns, constraints, and government initiatives that have contributed to the success of the Indian software industry.

The Indian software industry has experienced remarkable growth over the past few decades. In the 1990s, the industry saw a boom in growth due to the Y2K bug and the increasing demand for software services from overseas clients. This demand was met by the emergence of a large number of small and medium-sized software companies in India, which led to the creation of a highly competitive industry. The Indian software industry has since diversified its offerings to include IT-enabled services, business process outsourcing, and software product development.

However, the Indian software industry also faces significant constraints that limit its growth potential. One of the main constraints is the shortage of skilled workers, which limits the industry's ability to meet the growing demand for software services. Another constraint is the lack of innovation, as most Indian software companies focus on providing low-cost services rather than developing new products or services. In addition, the industry is also heavily reliant on overseas clients, which exposes it to fluctuations in the global economy.

To address these constraints and promote growth in the Indian software industry, the government has initiated several policies and programs. One of the most significant initiatives is the National Policy on Information Technology, which was launched in 2012. The policy aims to promote the development of the Indian software industry by encouraging innovation, providing support for research and development, and improving the quality of education and training in the sector.

Another important government initiative is the establishment of software technology parks, which provide infrastructure and support services to software companies. The software technology parks are located in several cities across India and offer a range of services, including high-speed internet connectivity, office space, and training facilities. These parks

have been instrumental in attracting overseas clients to India and promoting the growth of the Indian software industry.

The Indian government has also launched several programs to address the shortage of skilled workers in the software industry. One of these programs is the National Skill Development Corporation, which aims to train and certify workers in various industries, including software development. The program provides training to both new and experienced workers and offers certification in various software technologies.

In addition, the Indian government has also initiated several programs to promote entrepreneurship in the software industry. One of these programs is the Startup India initiative, which aims to support and promote startups in the software industry. The program provides funding, mentorship, and other support services to startups and has helped to create a vibrant startup ecosystem in India.

In conclusion, the Indian software industry has experienced significant growth over the past few decades due to the emergence of a highly competitive industry. However, the industry also faces significant constraints that limit its growth potential, including a shortage of skilled workers and a lack of innovation. To address these constraints, the Indian government has initiated several policies and programs, including the National Policy on Information Technology, software technology parks, and skill development programs. These initiatives have been instrumental in promoting the growth of the Indian software industry and ensuring that it remains a key sector of the Indian economy.