M. O. Svyrydenko
student of Faculty of Informatics and Computer Technique
National Technical University of Ukraine
"Kyiv Polytechnic Institute"
Research supervisor: O.A.Amons, associate professor
National Technical University of Ukraine
"Kyiv Polytechnic Institute "
Language tutor: A.V.Polischuk

QUALITY ASSURANCE ENGINEERING

Software quality assurance engineers and testers have an eye for detail and are committed to improving the performance of the products a company sells. They ensure that the software packages being sold meet or exceed the customer's expectations in performance and value. The biggest software companies spend millions of dollars of their budget on hiring quality software assurance engineers to head their quality assurance departments. This money is well spent, because the software quality assurance engineer will make sure that the product is top quality before it is marketed and sold.

A software quality assurance engineer is involved in the entire software development process to ensure the quality of the final product. This can include processes such as requirements gathering and documentation, source code control, code review, change management, configuration management, release management and the actual testing of the software. Software quality assurance is often confused with software testing, but should not be. Testing is a big part of software quality assurance, but it is not, by any means, the only part of it.

Software quality assurance engineers must be familiar with the entire software development life cycle in order to be effective. They may start out as software engineers or as testing engineers. They should have strong communication skills, be good at documenting their work, and must be able to work well with different groups. They often become the middle person between the developers and the customers, so they need to be able to understand, and convey, both viewpoints. In addition to running tests, software quality assurance engineers diagnose problems, recommend solutions, and determine whether program requirements have been met.

LITERATURE

1. C. Kaner, J. Falk, H. Nguyen "Testing computer software" (1999)
2. M. Andrews "How to break software"