Introduction to networking

Full Paper Microsoft. Net The Microsoft. Net framework is considered as a software component that can be integrated with Windows operating systems, as it provides added features by pre-coded programs to fulfill common program requirements (Microsoft . net. 2007). Likewise, it also administers program execution that is coded precisely for the framework. Moreover, it is also considered as a major add-on that is compatible with all new applications developed for Windows platform (Microsoft . net. 2007). Microsoft. Net solutions are comparatively cost effective to develop, cost effective to implement and cost effective to maintain (Miller, 2003). Applications associated with Microsoft. Net do not have to be dependent for example Visual studio, as Microsoft. Net based applications execute within the Common Language Runtime (CLR) that is similar to J2EE based application executing within the Java Virtual Machine (JVM) (Miller, 2003). Likewise, the Microsoft. Net framework incorporates a rich archive of functionality to the CLR that is far greater than J2EE. Moreover, another advantage that the Microsoft. Net framework provides a rich environment for programming, as less coding is required as compared to other programming languages. This is a significant advantage, as less coding concludes to low cost and less time for project completion (Miller, 2003). Furthermore, Microsoft. NET applications are robust and provide features to connect and support many users comparatively. This concludes that the total cost of a solution incorporates the amount of money is required to develop and implement an application. Hence, Microsoft. Net provides a significant advantage over other programming languages in this domain.

(Cabral, Sacramento, & Marques, 2007) one of the researches illustrated the issues associated with Microsoft. NET framework documentation. The research highlighted the nonexistence of appropriate documentation and quality in several Microsoft. NET applications. However, the weightage of the research was on the documentation quality, as different issues were found associated with absence of proper documentation while gathering assemblies for the research. Likewise, the study demonstrated that 87% associated exceptions were not documented. The values were ranging from 80% to almost 98%. However, for the issues associated with inadequate quality in documentation, part of the data associated with inadequate accounts of exceptions that can possible emerge in the process of calling other methods (Cabral, Sacramento, & Marques, 2007). However, the remaining explicit is fair enough and documented. Moreover, the research also concluded that by following call chains, automated documentation can be initiated. Therefore, it is evident that further research needs to be carried out for Microsoft. NET, as the coders are not documenting their codes along with the use of the unchecked mode, lack of adequate documentation associated with exception classes and the classes utilized for catching arguments are generic and do not facilitate any resolution of errors (Cabral, Sacramento, & Marques, 2007). In addition, in Microsoft. NET programming, exceptions are not being managed efficiently, as an effective use of an error handling tool can rectify issues to an extent.

References

Cabral, B., Sacramento, P., & Marques, P. (2007). Hidden truth behind . NETs exception handling today. IET Software, 1(6), 233-250.

Microsoft . net.(2007). Network Dictionary, , 309-309.

Miller, G. (2003). . NET vs. J2EE. Communications of the ACM, 46(6), 64-67.