

The programming process term paper

[Business](#), [Company](#)



\n[[Table of Contents](#)]\n

\n \t

1. [Introduction](#) \n \t
2. [Steps in the Programming Process](#) \n \t
3. [Conclusion](#) \n \t
4. [References:](#) \n

\n[/toc]\n \n

Introduction

The objective of computer programming is to develop electronic solutions for a particular company. In today's world, most companies use software systems in order to run their processes efficiently. Companies store their source data details in the system and perform transactions on them through computers. The software users expect certain outputs and reports from a computer system.

Steps in the Programming Process

A company will contract a programmer from within or outside the company to design and implement a software system. The first step in programming is to identify the problem (Jeffries, Anderson & Hendrickson, 2001). The company could be currently performing their operations using a manual system or a computer system. One approach to understanding the problem is to interview the managers and find out what exactly they want the system to do. This process should also be conducted with the end users for example the tellers in a bank or the cashiers at a supermarket. These are the people who work with the system daily. They could be given questionnaires to write

their frustrations with the current system and what they want. The staff in different departments may specify what kind of reports they want from the programs. What are the outputs they expect and more importantly what are the inputs keyed in the system? There are companies which have been using a particular software program and now they are feeling the system is not serving their needs.

The programmer could chart a process flow using flowcharts to understand the processes of the company. The programmer writes the expected requirements of the system by the user. (Jeffries et al, 2001). Only with a clear understanding can the programmer now attempt to design the system. With the design and specifications in place, the programmer then starts writing the program. There are three stages involved in writing a program (Pai, 2008). The first step is known as coding. The design has to be written in a programming language. There are various programming languages in the market such as Java, C++ and visual basic. The programmer chooses the best programming language based on the programming software requirements. This is known as the script writing. The second step is compiling where the instructions are converted by the computer into binary codes. The computer only understands language written in 0 and 1. The computer chips only understand these binary codes. There are programs called compilers that convert the language. Previously, scientists had to write their programs in complicated and complex styles. However, with advances in technological innovations, this has since changed (De Jong, 2006). The programs can now be written in the English language and are more understandable.

As technology progresses, the software programming languages becomes more and more simple to use. The compilers are strict and the programmer has to write the correct programs otherwise they will not be run or processed. They will give error messages on being used. The errors may be as simple as apostrophe use, comma use and other grammatical errors. The programs has to be written in the correct syntax or programming language (Carter, 2001). The third stage in writing the actual program is known as debugging. This is where the programmer corrects the errors in the written program. The compiler gives an error message and highlights the mistakes. This process is usually very long but very important. The correction of errors may even introduce other errors (Pai, 2008).

If the compilers were not to highlight the errors, the program would end up giving wrong outputs or performing the wrong transactions on the source or the raw data. The cycle of coding, compiling and debugging process is a repeated continuous process that the programmer has to keep doing. When compiling the programs, it should be noted that the programmer does not write the whole program then starts to compile and debug it. This process is done on piecemeal basis. First of all, part of the program is written then the programmer compiles it and debugs it. This is wise since it reduces slowly step by step the amount of codes that will be debugged. The programmer then tests the programs to confirm whether the program is giving the required outputs as required and specified by the end users. It is very easy to write the programs but the crucial question is solving the solution. The program is run several times. The codes are documented so that in the

future if the company wants to upgrade the software, they can do so with another programmer and not necessarily the one who designed it.

The programs are tested by the end user and their feedback obtained on whether their problems have been addressed. If the feedback is positive, the programs are implemented either in pilot phases or fully depending on the complexity of company operations or the risk involved in the implementation process. Training needs to be given to the end users on using the programs well and maximizing on all the features of the software. If the feedback is negative, the programmer goes back to work again till the user requirements are met (Carter, 2001).

Conclusion

Designing an efficient software program involves the use of resources such as time, money and people. The programmer works with the company staff to identify what kind of system is desired. He writes the programs and tests the programs with the users to find out if their system needs have been met. The programs are then documented and implemented in the company. The whole process requires teamwork and communication. Every person needs to perform in their role for the process to be a success.

References:

Carter, P. (2001). How to debug programs. Retrieved from.

<http://www.drpaulcarter.com/cs/debug.php>

De Jong, A. (2006). Evolutionary computation: a unified approach. Chicago: McGraw Hill

<https://assignbuster.com/the-programming-process-term-paper/>

Jeffries, R., Anderson, A. & Hendrickson, C. (2001). Extreme programming installed. Boston: Prentice Hall

Pai, D. (2008). What is the Difference Between Compiling and Debugging?.

Retrieved from

<http://ezinearticles.com>