TITLE: HOME-BASED SOFTWARE ENGINEERING BUSINESS OPPORTUNITIES
FROM: Robert Adams
EMAIL: robert.adams@whatifwe.com
WEB-SITE: www.whatifwe.com
DUNS: *********
CAGE: *****
DATE: April 14, 2010

1.0   OVERVIEW

ISO-9001 requires that enforceable standard methods and processes must be used on a project. Specifically, a conscious application of the four steps of the Deming Cycle at all levels is the most effective way of getting maximum quality.

Applying these principles to software engineering is a significant challenge:

> **STANDARD METHODS AND PROCESSES:** Standard methods are, at best, strongly recommended. It is not currently practical to rigorously enforce their use.

> **THOROUGH ECONOMICAL TESTING:** Thorough, economical testing of the product is always a problem. The acceptance test is never the right time to find errors in the product.   A thorough test plan must be included in the development process for all work products used in the product.

> **DIFFERENT VERSIONS:** The development of different versions of the product can create a source-code management nightmare. Often, conditional compilations statements are used to implement these upgrades resulting in larger and more complex source files.

> **PATCH ELIMINATION:** The Patch is often used to correct an error discovered at the last minute. The patch is usually not compliant with the standards that have been selected for the product and often creates additional problems that are must be corrected at a later time.

Robert Adams has spent 20 years developing methods of meeting the above challenge for his own home-based software engineering consulting business. These techniques have matured enough to be potentially useful to the Federal Government relative to software development opportunities for small and home-based businesses.

2.0   TECHNICAL SOLUTION
2.1   PROGRAMMABLE SOFTWARE DEVELOPMENT ENVIROMENT

The *Programmable Software Development Environment (PSDE),* a command line Win32 and Linux program, is fundamentally defined by the following **two step process:**

> The development of software parts libraries.
> The development of source files from these software parts libraries.

The **PSDE** project is divided into any number of **Development tasks** followed optionally by a **Test task.**

Each **Development task**s contains any number of library and source development sub-tasks. The **test task**, also based on the **two step process,** is capable of thoroughly testing a command-line product, module such as a class or subroutine, or a software parts library to verify proper operation as well as proper error detection handling capabilities.

2.2   STANDARD METHODS AND PROCESSES
2.2.1   OVERVIEW

The **two-step process** provides the opportunity of insuring that software developed using this tool will be in strict compliance with the standard methods and processes requirements of ISO-9001.

TITLE: HOME-BASED SOFTWARE ENGINEERING BUSINESS OPPORTUNITIES
FROM: Robert Adams
EMAIL: robert.adams@whatifwe.com
WEB-SITE: www.whatifwe.com
DUNS: *********
CAGE: *****
DATE: April 14, 2010

Each library sub-task prepares a software parts library from a collection of software parts; each of which has a name and list of commands.. Some of these commands define the preparation of product source code files.

Each source development sub-task is defined by a software parts library and one or more software development files; each of which contains the names of the software parts to be executed. The Product is prepared by the execution of the software parts in the order specified in the software development files.

If the standard methods and processes as required by ISO-9001 are implemented in the software parts libraries, these standard methods will be strictly applied in the development of product using the source development sub-task.

2.2.2   ERROR DETECTION

There are many techniques for detecting, reporting, and properly responding to an error encountered during the execution of a product. Many of these techniques are application specific; others apply to all applications.   In as much as many errors take a long time to occur and are difficult to correct, it is very important to rigorously include these techniques in the delivered product. *Strict Error Detection can easily be accomplished by including the associated techniques in the software parts libraries exclusively used to develop the product.*

2.2.3   ERROR PREVENTION

There are many techniques that are know to be more error resistant than others. For example, structured programming is very error resistant where as the use of global storage is very error-prone. *The instruction set contains a comprehensive set of error prevention commands that can enforce the proper use of the selected error-prevention methods.*

2.3   THOROUGH ECONOMICAL TESTING

The *test task* provides the capability of not only testing a complete command-line product, but also various work-products such a classes and subroutines and software parts libraries   This is accomplished by preparing a test program, building an executable, and running the executable for each test condition. Errors are reported from all steps in this process. The ability to evaluate the error resistant strength of software parts libraries can be evaluated by the two consecutive source development sub-task used to prepare the test program.

This test task is very efficient and economical on all work-products that do not require human intervention; that is, pushing buttons on a GUI.

2.4   DIFFERENT VERSIONS

Strict Portability between a Windows and a Linux operating system has been achieved. Two kinds of files are involved: 1) A text file and 2) a system independent binary file.. This coupled with nine user defined parameters enables all of the files to be used in either operating system without change.

Command-Line C++ product development in Windows and Linux is quite similar.   There are differences in the static libraries and the behavior of a few system routines. All of these differences can be solved with a few well defined system specific software parts libraries.   These system specific libraries can be specified in the command-line.

Strict Portability between C++ and C# is currently being developed.

TITLE:	HOME-BASED SOFTWARE ENGINEERING BUSINESS OPPORTUNITIES
FROM:	Robert Adams
EMAIL:	robert.adams@whatifwe.com
WEB-SITE:	www.whatifwe.com
DUNS:	*********
CAGE:	*****
DATE:	April 14, 2010

2.5 PATCH ELIMINATION

Longitudinal parity has been used for many years as an error-detection algorithm. Encrypting the data with a pseudo-random generator before computing the longitudinal parity byte greatly increases its error-detection capability.

The Cryptographic Commands of the **PSDE** include the generation of a pseudo-random longitudinal parity byte. A parity byte can be attached as a comment at the end of each line of source code.   These commands can be used to certify that:

  ➢ The proper software parts libraries have been used in the design.
  ➢ No patches have been inserted into the product source code.

 Furthermore, the PDSE can perform the certification analysis of the product source code files.

3.0 OFFERING
3.1 OPEN SOURCE TOOLS

The command-line based PSDE is available from www.whatifwe.com free-of-charge under an open-source license. It is well documented. As in all open-source programs, there are no guarantees, expressed or implied.

3.2 SOFTWARE ENGINEERING SERVICES

Robert Adams will answer routine questions regarding the use of the PSDE; a BLOG has been linked to the web-site for this purpose. Other services such as software parts library development must be purchased. Every effort will be made to keep all projects well within the Micro-Purchase level.

4.0 CONCLUSION

I, Robert Adams, am a free-lance software developer licensed as a home-based business in the City of Los Angeles. I developed this tool to be able to develop high-quality software at far less cost then the tradition software development companies.

This tool has now been developed to a point that it can now be beneficial to the Federal Government. *Specifically, it can provide a workable pathway for other free-lance software engineers to service the government software development needs. Specifically:*
  ➢ The procuring agency can specify the standard methods and processes to be deployed on the specific project by specifying the required software parts libraries and where to obtain them.
  ➢ The procuring agency can verify that specified software parts library was exclusive used in the development of the product from longitudinal parity bytes attached to each source file line.

The devil, of course, is in the details which we must work out. If we can collaborate on those details, we will be able to create a significant opportunity for the free-lance software engineer who was laid-off or needs to be at home to perform parenting duties. I believe that the home-based software engineering business can significantly reduce the cost of the development of high quality software.

I propose that we provide this opportunity to the home-based business environment.