

Robert Adams
robert.adams@whatifwe.com

To Whom It May Concern:

Pioneers must receive a few arrows before settlers can get any land.

The human being is a tool-based animal that instinctively learns from experimentation. A baby begins this process by putting things into his mouth.

Schools, on the other hand, are training process. Students who perform in accordance to the teacher's expectations, methods and standards are rewarded. Students who find alternate means of accomplishing the same goals are often penalized.

The success of our way of life is critically dependent upon a pioneer spirit; i.e., an ample collection of pioneers. Schools, for the most part, produce settlers, not pioneers.

The personal computer is perhaps the most powerful experimental tool that has ever been developed. It provides the experimenter with a unique opportunity to explore a large range of subjects on his own time and budget. It can greatly help the experimenter to become a pioneer.

I am a software engineer with over 45 years of experience. Over the last 15 years, I have been aggressively pursuing the quest of economically developing error-free software and I have made much progress in this pursuit. During this pursuit, I have discovered a proactive interface between the training process delivered by a school and an individual's personal learning process.

Please examine the attached documents for a more detailed description of my discovery and the benefits resulting from it.

In order to reduce my discovery to practice, I need to find a few pioneers willing to accept a few more arrows. Perhaps our experimental efforts will increase the number of pioneers available to a number of critical technologies. Increasing the number of pioneers will help preserve our way of life.

Sincerely,

Robert Adams

TITLE: I HAVE A SOFTWARE DREAM
FROM ROBERT ADAMS
WEB-SITE www.whatifwe.com
EMAIL: robert.adams@whatifwe.com
DATE: November 29, 2007

1.0 MY DREAM

1.1 OVERVIEW

My Dream is to provide the United States with the benefits of an economical Error Free Software technology. This technology is already strictly compliant with the Deming Cycle and can be fully utilized by a free lance software engineer or telecommuting employee. We are rapidly approaching the time when a major disaster will occur from a software error.

1.2 ROLE OF EDUCATION

My technology has greatly improved the quality of the software that I have developed and my clients have greatly benefited from the quality of the resultant product. Unfortunately, I will not be able to bring my dream to fruition without involvement of our educational system at all levels:

- Exploration and experimentation by the University professors is needed to mature the technology.
- The legal definition of a licensed professional engineer is defined in a large part by academic achievement. We will need to soon consider a licensed professional software engineer.
- To obtain an ample supply of talent, the educational process must occur during high-school. We will need to improve the value of a high-school education and the availability of a college education in the mind of the high-school student.

1.3 SCHOOLS AND THE LEARNING PROCESS.

The `human being is a tool-based animal that instinctively learns from experimentation. A baby begins this process by putting things into his mouth.

Schools, on the other hand, are training process. Students who perform in accordance to the teacher's expectations, methods and standards are rewarded. Students who find alternate means of accomplishing the same goals are often penalized for using different methods.

1.4 ROLE OF THE DEMING CYCLE IN EDUCATION

The Deming Cycle can greatly improve the quality of our educational process. It is defined by the following four steps:

- Plan: Make a plan for doing the task.
- Do: Perform the task in accordance with the plan
- Check: Evaluate the success of the completed task.
- Analyze: Upgrade the plan as necessary to improve its success.

I believe that a workable interface between the training of a classroom and an individual's learning process can be found in the Deming Cycle. The student can be trained to use the Deming cycle and a learning process is required in the Check and Analysis steps.

1.5 ROLE OF THE PERSONAL COMPUTER

The personal computer is perhaps the most powerful experimental tool that has ever been developed. It provides the experimenter with a unique opportunity to explore a large range of subjects on his own time and budget. The experimenter

TITLE: I HAVE A SOFTWARE DREAM
FROM ROBERT ADAMS
WEB-SITE www.whatifwe.com
EMAIL: robert.adams@whatifwe.com
DATE: November 29, 2007

is no longer dependent upon the support of a grant or other organization to perform a significant piece of original research.

2.0 PERSONAL EXPERIMENTAL ACTIVITIES

2.1 PERSONAL OVERVIEW

I have been an experimenter for most of my life. I started in electronics at approximately the age of eight. In the beginning, it was in the form of a hobby. As time proceeded, I began to focus upon the problems that I cared about. All of my employment was defined in a large part by this experimental process.

2.2 RESEARCH AT LOCKHEED

In 1966, my first experimental computer program, the Markov simulation of an atomic bomb, caused me to be transferred to the Scientific Computer Division where I got a rare opportunity to perform a number of IRAD projects, one of which was an extension of the original Markov Simulation.

Around 1968, Lockheed developed a “compiler-writer” (a program that writes programs) with a guarantee that every correct statement would be properly parsed and every error would be caught. Error-free software was now feasible. I used this program to develop a multi-dimensional numerical functional analysis computing lab.

2.3 CONFERENCE PUBLICATIONS AT IMACS

In 1980, I bought my first personal computer. I developed a PC based Multi-Dimensional Functional computing laboratory and used it on Markov simulation of a highly disturbed gas. I presented the results of my experiments in four IMACS conferences in Oslo (1986), Prague (1987), Paris (1988), and Lille (1991). The participants that I met in Prague were amazed that I personally funded all of my research and attended the conferences on my vacation.

2.4 INDEPENDENT CONTRACTOR CHALLENGES

In 1992, I became an independent contractor specializing in the economical fixed-price development of software. To be able to profitably accomplish my objective, I aggressively pursued economical methods for developing quality software; I discovered how “software breaks”, how to protect the customer from these failures, and developed a software development tool, a programmable code generator to facilitate what I had discovered.

2.5 ISO9001 AND THE DEMING CYCLE

In 2003, I was exposed to the Deming Cycle. I immediately saw the connection between its requirements and the tools that I had developed. I upgraded the Programmable code generator to satisfy the Plan step and developed a Monte-Carlo Programmable Test System to satisfy its Check and Analysis steps.

3.0 SOFTWARE DEVELOPMENT, EDUCATION, AND THE DEMING CYCLE

As I developed a personal implementation of the Deming Cycle, it became apparent that it was the interface between the personal learning process and the training provided by schools. Specifically, a person can be trained to perform in accordance to the Deming Cycle which will require a learning process in the C and A steps. Furthermore, I believe that major software industries would be very interested in individuals whose IT education is based on the Deming Cycle.

EXPERIENCE SUMMARY

- ❖ **FORTY FIVE YEARS IN SOFTWARE, DIGITAL, ANALOG ENGINEERING** with an emphasis in the development and test of high reliability software. This experience includes the application of the Deming Cycle to software engineering. This experience includes **employment**, **independent contracting**, and **long-term personally funded independent research and development efforts**.
- ❖ **EXPERIENCE HIGHLIGHTS:**
 - **1992-PRESENT: SOLE PROPRIETOR – WHAT IF WE**
 - **PROGRAMMABLE DEVELOPMENT ENVIRONMENT:** Currently combining the *Programmable Code Generator* and *Programmable Monte-Carlo Test system* into an *Integrated Programmable Development Environment*. Included in the upgrade are the following features
 - **Combine multiple software kit libraries into a project library** to enables the division of a large project into manageable tasks.
 - **Multiple development tasks within a project** enable the designer to create a user-defined input language translator
 - **Sixteen Output Files** facilitates the simultaneous development of both documentation and product source code.
 - **Line-Oriented Text File Input** provides the input needed for a user-defined input language translator.
 - **Multi-dimensional state machines** are needed for development of a simple-precedence based input language translator as well as the linkages required for a UML model.
 - **PROGRAMMABLE MONTE-CARLO TEST SYSTEM:** Developed software test system for measuring the resistance of software parts and modules to use-related programming errors for both Windows and Linux operating systems.
 - **EXPORTABLE ELINT:** Using the MKS environment developed the Tuner-Antenna Controller and IEEE488 Interface for an exportable ELINT system. Monte-Carlo methods were used to rapidly and thoroughly test the software.
 - **EXPORTABLE CRYPTOGRAPHY:** Upgraded firmware for an exportable KY58-equivalent Wideband Secure Voice System. Developed IDE for simulating and producing the prom-files for its custom bit-slice processor using the *Programmable Code Generator*.
 - **STANDARD SOFTWARE PARTS:** Developed the basic *Programmable Code Generator* for both Windows and Linux operating systems that implemented a two step software development process consisting of:
 - Development of a Project Library of standard software parts
 - Development of Product exclusively from the software parts in this project library
 - **SOFTWARE ERROR DETECTION:** Software “breaks” irretrievably when it makes an erroneous write. Formal error detection methods were developed which would anticipate an erroneous write, prepare a report, and terminate execution of the program. Determined that a *Programmable Code Generator* based on the Macro Assembler’s macro instruction would greatly facilitate the implementation of these formal error methods.
 - **MEDICAL PROGRESS NOTES EDITOR:** The goal of this effort was to be able to prepare a custom Medicare Progress Notes Editor in about 10 minutes at the customer’s site. A program that wrote this program was developed and successfully used in preparing several custom editors. The primary challenge was verifying that these editors were error-free.

- **1975-1989, 1991-1992: OCEAN TECHNOLOGY, INC.**
 - **STATISTICAL MECHANICS – SEVERELY DISTURBED GAS**
 - **Purpose:** It had become quite apparent that major economic and political problems would occur over the increasing use of petroleum products and that a possible solution was a thermonuclear reactor. Furthermore, ball-lightning, if properly implemented in a deuterium-tritium environment might be a practical solution.
 - **Challenge:** The basic challenge was to be able to perform a statistical mechanics simulation of a highly disturbed gas on a Heathkit-H89 personal computer (Z80 processor, 64k memory, 3-floppy disk drives).
 - **Effort:** To meet the above challenge, the operating system was extensively upgraded and a version of the numerical functional analysis computing laboratory was developed. This program was used to simulate a disturbed gas.
 - **Conference Publication -- 1986:** The results were presented in Oslo, Norway while on vacation. The experiment clearly demonstrated the power of the personal computer. Unfortunately the results tended to indicate that not enough points were used to represent the functions.
 - **Conference Publication-1987:** The numerical approximation used in the statistical mechanics experiment presented in Oslo was explored for adequacy in the number of points. Found that the discrete time-dependent representation did not converge to the continuous representation as required by the definition of a continuous function. The results of this effort were presented in Prague while on vacation.
 - **Conference Publication - 1988:** Found a potential "Goedel's Incompleteness Theorem" explanation of the findings presented in Prague. Specifically, Demonstrated that one can not prove that time is either continuous or discrete thus rendering the use of time-dependent differential equations of motion potentially invalid. Presented these findings in Paris while on vacation.
 - **WIDE-BAND SECURE VOICE SYSTEM:** Developed an exportable equivalent of KY-58 wideband voice encryption system. 16 kilobit/sec CVSD digitized voice was encrypted by a multi-level pseudo-random generator defined by 128-bit cryptographic key and a 32-bit session-key. A custom bit-slice processor was developed for this application.
- **1961-1975: LOCKHEED CALIFORNIA COMPANY**
 - **MARKOV SIMULATION:** My first computer program was matrix-based statistical mechanics simulation of a sphere of fissionable material (atomic bomb). An estimate of the critical mass was made by running various diameters. This program was prepared on my own time and run at the Scientific Computer Services Division by a friend.
 - **IRAD PROGRAMS**
 - **PICARD INTEGRATION SIMULATION:** The Markov-base Atomic Bomb experiment was upgraded to a Picard Integration method using a Partial Spline approximation to the probability functions. The results were far superior to those obtained in the discrete Markov Simulation.
 - **FUNCTIONAL ANALYSIS COMPUTING LABORATORY:** A comprehensive computing system for the manipulation of multi-dimensional functions approximated by the Partial Spline was developed.
 - **BAYESIAN METHODS – NORAD FEASIBILITY STUDY:** Using the Functional Analysis Computing Laboratory, explored the feasibility of using Bayesian methods to improve the capability of NORAD in a full scale attack. "Is Los Angeles under attack and how much time remains" was the question selected for the Bayesian Instrument. First a statistical instrument was defined for this question using a single Synchronous satellite (fly eye). Next the instrument was used to develop the probability of threat from various missile trajectories. The program was quite successful in evaluating single and multiple threats faster than real time on a non-dedicated machine.

❖ **EDUCATION:**

B.A. Physics, U.C. Berkeley, 1962.

❖ **SECURITY CLEARANCE:**

Secret – Last Active in January 2004