

The True Cost of Operation and



A long, long time ago, in a country far, far away, there lived a Mighty King. The Mighty King was happy, as his computing needs were very simple. What simple programs he needed to run his kingdom could be punched up on less than 2000 punch cards. When the programs became obsolete, the King simply had the programs rewritten as needed. This method served the King's needs well.

And the King was happy.

But as computing power became greater, and the needs of the kingdom became more complex, the King had to change his method of program production. He summoned forth his wise consultants, and explained that the programs were becoming larger and more complex. He was at a loss as to what to do.

And the King was unhappy.

The wise consultants of his kingdom explained to the benefits of an RFP. And the RFP was put forth. And responses were collected. And contracts were let. And requirements were gathered. And prototypes were developed. And the prototypes created great amounts of fertilizer.

And the King wept.

And JAD sessions were held, and requirements modified, and new versions of the software created. And this begat Version 1.

And the King was well pleased.

But the users began to complain, and asked for more and more features. And the King assembled a maintenance team together these additional requirements. But the requirements

Maintenance

were not prioritized. And the maintenance team tried to add every request into the second version of the system. And the days of coding were many, but unfortunately the days of testing were few. And even though there were still many errors, this effort prematurely brought forth Version 2. And it was slow. And it crashed frequently. And it had many features that very few people used. And it ate up all available memory. Yea, verily, did I mention it crashed frequently?

And the King waxed wroth.*

And the King summoned his advisers, who explain to him about software bloat. And the King threatened to behead some developers. And lo and behold, a new design team for Version 3 was assembled. And requirements were re-gathered and re-verified. In the verified requirements were prioritized. And agile methodologies and object oriented design were used. In the days of analysis and design were great, and the days of coding were few. And because of the design process, and proper analysis and design, The days of testing were even fewer. And the system ran well, And the users were happy.

And the system performed as expected, and the King was well pleased.

And the days passed, and eventually the host hardware and the operating system were upgraded. And minor changes were made to the software, and it still ran well. For a while. And then more and more upgrades were required. Hardware changed again. The system was re-hosted to a different platform. And the changes grew and grew. And version 3.14 8.6g begin to crash with great regularity.

And the King was upset. His treasury was also bankrupt, and therefore he did not have the money to completely rewrite the software a fourth time

And the King summoned his advisers, who explained to him about re-factoring in maintenance. And the King was tutored in

the areas of technical debt, and the magical spells for perfective, corrective, and adaptive maintenance. And as the system kept evolving, it worked less and less well. And the King endeavored to rewrite the system from scratch, but there was never enough time, and never enough money.

And the King was confused.

"Surely", the King thought, "somebody in the kingdom can tell me when I need to rewrite the software, when maintenance no longer brings forth any benefits." And many wise men were summoned and queried but no one had an answer that suited the king. Each contractor wanted their own company to do the rewrite, and each quoted an exorbitant price. And, the new federal oversight initiatives being what they were, not one contractor had the decency to offer the king a kickback.

Finally, after his advisers had given up, a very old sage was located, and summoned to appear before the king. This sage supposedly had the skill and expertise borne of many, many years of operations and maintenance experience, and even the wisest consultants considered him full of wisdom. He had truly earned the title of "graybeard". At last he appeared, and went before the king and knelt. "How may I assist you" the sage asked. "Oh, great sage", the King said, "Nobody seems to when to quit maintenance changes and rewrite the software again. I don't even have an estimated cost. Do you have any idea when I can afford to rewrite the software again?"

"Never", set the wise sage. "It is never affordable. However, the day will come when you can't afford not to!"

And the King saw the wisdom of these words, and realized that he had been doing the best that he could. The King immediately commissioned a task force to ask Congress for federal aid, and he also formed a committee to petition Congress to allow the new development money to fall under O&M funds.

And Congress agreed, and they all lived happily ever after.

David A. Cook, Ph.D.
Professor of Computer Science
Stephen F. Austin State University
 cookda@sfasu.e

*Note by author: I have wanted to use this phrase all my life. I heard it in the Groucho Marx movie Horse Feathers, made in 1932, well before my time. The lines were:

Secretary: "The Dean is furious. He's waxing wroth."

Quincy Adams Wagstaf [Groucho]: "Is Roth out there too? Tell Roth to wax the Dean for a while."