THE SDLC AND SIX SIGMA
AN ESSAY ON WHICH IS WHICH AND WHY?

Dr. Roy A. Boggs, Florida Gulf Coast University, rboggs@fgcu.edu

ABSTRACT

The Systems Development Life Cycle (SDLC) has been in existence for almost half a century, while Six Sigma has only recently appeared in IT literature. Given all of the areas of concern addressed by the SDLC, where does Six Sigma, and Design for Six Sigma, fit it? Is it something new, simply a matter of passion or is it a new paradigm replacing the old? These questions are addressed from a historical standpoint, with a review of current implications for IT scholarship. The answer is that it is none of the above. Rather it is a subset seeking renewal and rededication.

Keywords: SDLC, Systems Development Life Cycle, Six Sigma, Design for Six Sigma

INTRODUCTION

In December 2003, a round table involving systems analysts from major software developers was held in Ft. Myers, Fl. Those taking part in the round table discussion represented international firms for medical and hospitality software as well as significant Internet applications. The purpose was to discuss the application of current Systems Development Life Cycles (SDLC) as presented in IT literature. (20) Even though the discussion begin with a position paper reviewing the SDLC and its history, it became quickly apparent that many developers were in reality still practicing the analysis and 'loop until you get it right method' of development, while others had become converts to Six Sigma.

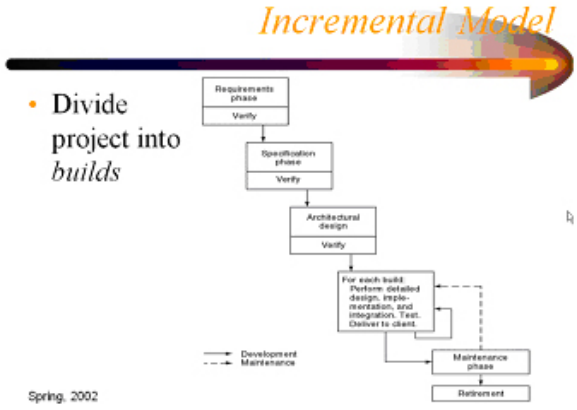
The word 'convert' is used guardedly. However, there existed a distinct difference in the level of enthusiasm between those of the traditional Waterfall SDLC and the more recent Six Sigma proponents. Given this difference, for those in attendance the question became: Which is Best and Why? Which should we practice, and why? And what about Design for Six Sigma (DFSS)? Is this a new SDLC or a replacement for the SDLC, or simply something else?

Further discussion revealed that, of all of the important topics such as documentation, adoption, maintenance, management control, outsourcing, quality control, risk management, time management and scope, two topics stood out as central to Six Sigma: user involvement and financial management. These two topics are indeed mirrored in SDLC history. However, they have been represented with various levels of emphasis and often with limited success.

Using these two topics, user involvement and financial management, the following looks first at the discordant array of SDLCs and then reviews Six Sigma and DFSS. Finally comparisons will be drawn and summary given.

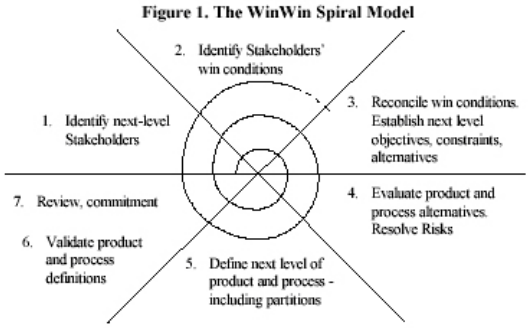
THE SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC): A BRIEF REVIEW

A literature review of history of the Systems Development Life Cycle (SDLC) dates back to the mid 1960s. While working for the United States Department of Defense, A. Enthoven and Henry Rowan developed a heuristic process for managing large information systems projects. (10) This process developed into a *linear* list of *stage*s in project development that could be re*cycle*d as needed. Winston Royce introduced the first formal SDLC model in 1970. (14) This model was to become the well-known "Waterfall Model", also known as the Linear Sequential Model or Classic Life Cycle. All subsequent direct permutations of this model can be summed as some permutation of the 4Ds: **D**iscover, **D**esign, **D**evelop and **D**eliver. The early history of this model did not include the end-user to any great extent and cost overruns were legend. Both of these failings can perhaps be understood, given the nature of the hardware and the real lack of financial experience versus the scope of the projects. This was to a large extent the realm of the engineer.

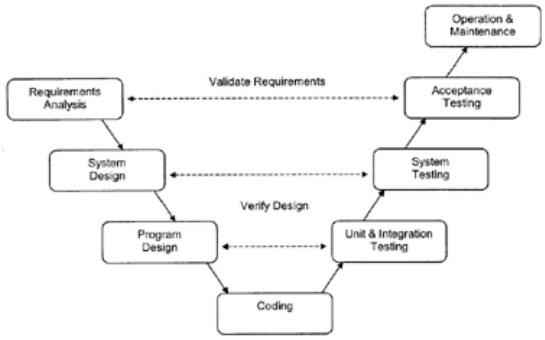


In the early 1980s, the more flexible Incremental Model was introduced. Linear sequential processes are applied in a staggered fashion over a period of time. Each sequence produces a deliverable output, or increment, of the project. All additional feature features remain undeveloped until the user reviews and approves the current build. This process is repeated until the project is completed. Here the user representative is expressly included in the *review* of each sequence, which quite often causes 'maintenance' delays and cost overruns when the user fails to accept a particular build. If in the end the project proved useful in some degree to the user, the project was often considered a success in spite of 'cost creep' concerns. This was to a large extent the realm of the analyst.

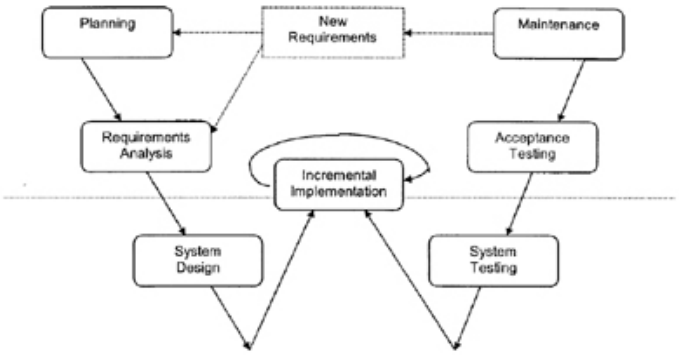
By the 1990s, newer SDLC models began to appear. Of particular important was the Spiral Model of Barry W. Boehm. (1 – 7) This model was designed in the first instance to improve risk management and system survivability. This it did by beginning with a limited prototype using polar coordinates, which enabled each build to expand on each coordinate, with risk management always a central factor. This model does include the user representative as an input to the success of each cycle, it does not provide a strong vehicle for end-user involvement. It also is difficult to apply cost, time, and schedule estimates across a project portfolio.



Boehm with his subsequent Next Generation Process Model (NGPM) addressed some of these difficulties in 1994. This iteration of the Spiral Model involved the stakeholders to a much larger degree and better identified user's needs at the outset of the project. However, there remained the real possibility of conflicts among stakeholders; and by the end of the decade, Boehm had introduced the Win-Win Spiral Model. This iteration added activates to each spiral that expressly identified the stakeholders, their win-conditions, and methods for resolving any conflicts. At the same time, stronger risk management techniques were added. The result was an SDLC which provided a great rate of acceptance and buy-in by the stakeholders and much stronger risk control which led to much stronger financial control. But stakeholder should not in every case be considered synonymous with the end-user; and the identification methodology for risk needed clarification. This is the realm of the manager.



By the turn of the century, a new direction for the SDLC had been introduced in Germany, the V-Model. (17 – 19) Currently this model is the model required in Germany for all military, civil and federal IT projects. It is also a guideline for anyone that works on federal projects. The V-Model focuses on project management, software development, quality assurance, and configuration management, with special attention for better communication between developer and customer. Cost are expected to be significantly lowered though its quality assurance modules. The V-Model has become internationally accepted and reflect standards, such as ISO/IEC 12207 or ISO 9001. A newer model, the Project WEIT model, is scheduled for April 2005 to address issues such as scalability, change adoption and application formulation.



One addition model with a similar format is the W-Model of the Frauenhofer Institute of Production Technology. (16) This Model focuses primarily on coding and testing to achieve more cost effective code under risk evaluation. In the same direction is the Rational Unified Model and Unified Modeling Language. Prominent developers include Microsoft, Oracle, HP, IBM and Texas Instruments. There is ample opportunity for user feedback, but the primary interest remains code and coding practices. This is the realm of the software developer.

In this brief historical review of the SDLC, from the realm of the engineer to that of the software developer, the end-user is seldom totally left out of consideration. However, one might speak more of distant concern than of actual methodologies for total end-user participation. (At the same time, it is instructive that major projects have been known to train end-users to develop interfaces.) Cost concerns can be found in all the realms and have driven a great many changes in Systems Analysis and Project Management. However, massive failures are well known, especially in public projects where the public has access to project data. On the other hand, IT is not without successes: payrolls are, excepting in rare instances, processed as expected.

SIX SIGMA (6^{σ})

Sigma, standard deviation, indicates in Six Sigma literature, the extent to which a product deviates from the ideal norm. The number 'six' indicate statistically how many failed parts (3.4 defects) per million are acceptable. This includes manufacturing processes as well as data processing. When someone ferrets out a defective process, the correction of which will lead to better customer satisfaction or an improved delivery process, Six Sigma kicks in. The core of this process is labeled **DMAIC**: **D**efine the problem, **M**easure where you stand, **A**nalyze the problem, **I**mprove the situation, and **C**ontrol the new process to see that the problem has been corrected. (12)

In corporations from General Electric, Motorola, and Allied Signal/Honeywell to Home Depot and even in military training schools, Six Sigma is presented as a challenge for every employee. Every level of the organization is recruited with titles such as (in order from the top): Executive Leadership, Champion, Master Black Belt, Black Belt, Green Belt and Team Member. **DMAIC** focuses on improvement; and successes can be demonstrated. The end-user is foremost and cost improvement at the core of any proposal. "Six Sigma organizations believe that they can be creative *and* rational, focus on the big picture *and* minute details, reduce errors *and* get things done faster, and make customers happy *and* make a lot of money … Six Sigma is an operating philosophy that is customer focused and strives to drive out waste, raise levels of quality, and improve financial performance … "(15) A user/cost combination that has proven itself effective. At the core is the realm of the engineer – identify the opportunity for improvement and make the product better.

DESIGN FOR SIX SIGMA (DFSS)

By the turn of the century, the analyst was not far behind. Six Sigma received a new definition: "a comprehensive and flexible *system* for achieving, sustaining and maximizing business success. Six Sigma is uniquely driven by close understanding of customer needs, disciplined use of facts, data, and statistical analysis, and diligent attention to managing, improving, and reinventing business processes." (12) Where the initial emphasis was on pin pointing a problem and improving it with enthusiasm and determination, the *s*ystem was now expanded to include its *design*. Catch the problem before it happens.

Design for Six Sigma (DFSS) presents it own steps: **I**dentify the opportunity for improvement, **D**efine the requirements, **D**evelop the concept, **O**ptimize the design and **V**erify it. **IDDOV** is designed to attack new processes and improvement. (8) This includes areas such as tightening

tolerances, reducing variations (as in a transaction processes), and employing robust design techniques. The goal is to generate fewer problems in the first place.

A SUMMARY

In some ways, Six Sigma represents several of the Systems Ten Commandments done well, while the SDLC represents all the commandments. This is not a negative comment. Maintenance projects have long been a staple of IT project portfolios. The history of the SDLC demonstrates, if nothing else, that analysts have spent almost half a decade pursing better and more effective processes. To have several parts greatly improved on the one hand is not a negation of the other. It is a contribution, the success of which can be taken seriously. The end- user is extremely important and cost management mandatory.

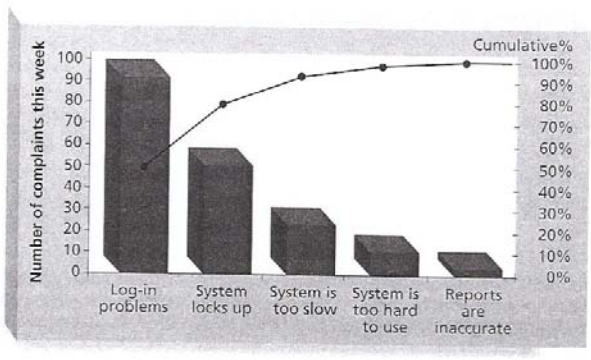


Figure 8-1. Sample Pareto Diagram

However, one should not confuse better engineering with better information systems analysis and design. It can be dangerous to simply transfer engineering considerations to Information Technology (IT) portfolio management. One might, as an example, consider outcomes when formal Project Management tools are simply transferred to an IT problem, without careful consideration. The diagram on the left shows how a Pareto Diagram might be used to resolve a problem with a system that is not performing well. The problem of inaccurate reports is marginally recognized: "the company should focus on making it easier to log in to the system to improve quality, since the majority of the complaints fall under that category. The company should also address why the system locks up. Because [the Figure] shows that the problem of inaccurate reports is barely mentioned, the project manager should investigate who made this complaint before spending a lot of time on addressing that potentially critical problem with the system." (15)

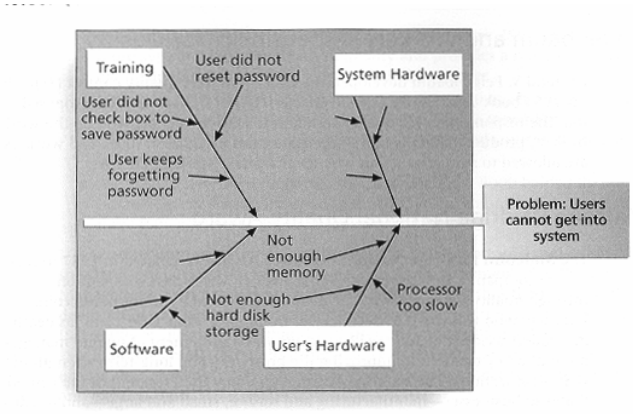


Figure 8-6. Sample Fishbone or Ishikawa Diagram

The presentation then moves to a recreation of this same scenario with a Fishbone diagram. The incorrect management reports are not included. For an information system professional, an incorrect management report is not a matter of "who made the complaint". Rather it is one of corporate information management and of central concern for any IT project. If the reports are not accurate, a better engineered system will not improve business practices. If the inaccurate reports are corrected, the level of engineering can be ascertained to improve performance. There is a difference for an information system professional.

The SDLC itself could indeed use some converts and enthusiasm for identifying and executing projects that make an impact on the corporation. The necessity for top-down leadership has always been noted (identify the primary-user), end-user concerns have been constantly increased (you can complete the best project in the world, but if the end-user doesn't use it, you have nothing), and cost estimation and management have been emphasized (IT students are instructed in financial principles). However, Six Sigma, as a subset of the SDLC, forces one to look at these areas with renewed interest. If Six Sigma teaches nothing else, it is that IT scholarship needs to continue to examine the SDLC and that analysts need to keep a focus on all aspects of the SDLC, or new religions will continue to find anchor.

BIBLIOGRAPHY

1. Boehm, B. W., Egyed, A. Kwan, J. Port, D. Shah, A. & Madachy, R. J. "Using the WinWin Spiral Model: A Case Study". IEEE Computer *31*.7 (1998): 33-44.
2. Boehm, B. W., Elwell, J. F., Pyster, A. B., Stuckle, E. D., & Williams, R. D. "The TRW Software Productivity System". Proceedings, 6th International Conference on Software Engineering (1982): 148-156.
3. Boehm, B. W., Egyed, A. Kwan, J. Port, D. Shah, A. & Madachy, R. J. "Using the WinWin Spiral Model: A Case Study". IEEE Computer 31.7 (1998): 33-44.
4. Boehm, B. W., & Bose, P. "A Collaborative Spiral Software Process Model Based on Theory W." Proceedings, 3rd International Conference on the Software Process, Applying the Software Process (1994).
5. Boehm, Barry W. and Rony Ross. "Theory-W Software Project Management: Principals and Examples." IEE Transactions on Software Engineering 15.7 (1989): 902-916.
6. Boehm, B. W., Egyed, A. Kwan, J. Port, D. Shah, A. & Madachy, R. J. "Using Developing Multimedia Applications with the WinWin Spiral Model." Proceedings, ESEC/FSE (1997).
7. Boehm, Barry W. "Guidelines for Verifying and Validation Software Requirements and Design Specifications" (1979). http://www.informatik.uni-bremen.de/uniform/gdpa/def/def_v/V_MODEL.htm.
8. Chowdhury, Subir. Design for Six Sigma. The Revolutionary Process for Achieving Extraordinary Profits. Trade Publishing, 2002.
9. The Federal Government Co-ordination and Advisory Agency for IT in the Federal Administration (KBSt). "Das V-Modell - Chance zum maschinellen Workflow-Management". (2003) http://www.kbst.bund.de/Themen-und-Projekte/-,232/V-Modell.htm.
10. Gross, Paul F. Systems Analysis and Design for Management. 1976.
11. Harry, Mikel and Richard Schroeder. Six Sigma: The Breakthrough Management Strategy Revolutionizing the World's Top Corporations. New York: Doubleday, 2000.
12. Pande, Peter S., Robert P. Neuman and Roland R. Cavanagh. The Six Sigma Way: How GE, Motorola, and Other Top Companies Are Honing Their Performance.New York: McGraw-Hill, 2000.
13. Projekt WEIT. "Weiterentwicklung des Entwicklungsstandards fur IT-Systeme des Bundes auf Basis des V-Modell-97." (2003). http://www.v-modell200x.de
14. Royce, Winston, "Managing the Development of Large Software Systems."

15. Schwalbe, Kathy. Information Rechnology Project Management. Boston, Course Technology, 2004.
16. Spillner, A. “The W-Model” Strengthening the Bond Between Devlopment and Test.” XDD3572filelistname1.pdf obtained from http://www.stickyminds.com/sitewide.asp?ObjectID=357&Function=DETAILBROWS&ObjectType=ART.
17. Vorgehensmodell. “Einfuhrung in das V-Modell.” (1997). 2ENIF.pdf obtained from http://www.informatik.uni-bremen.de/uniform/VModel_PS_PDF/Deutsch/VM_pdf.zip McCracken, Jackson, 1982
18. Vorgehensmodell. “Regelungstril Vorbemerkungen.” (1997). Ovorb_z.pdf obtained from zip file GD250 V Model Part 1. http://www.informatik.uni-bremen.de/uniform/gdpa/z-pscrip.htm.
19. Vorgehensmodell. “Handbuchsammlung Berucksichtigung objektorientierter Sprachen.” (1997). OOS.pdf obtained from http://www.informatik.uni-bremen.de/uniform/VModel_PS_PDF/Detusch/Part3_pdf.zip.
20. Waterworth, Jana., “An Elucidation on the Systems Development Life cycle,” (http:// www.fgcu.edu/rboggs)