(12)

AD-A272 383

AD

AD-E402 486

Technical Report ARPAD-TR-93003

THE READINESS GROWTH MODEL: A QUANTITATIVE ANALYSIS OF SOFTWARE RISK

David R. Castellano
Paul E. Janusz
Sharyn L. Tolochko

October 1993

U.S. ARMY ARMAMENT RESEARCH, DEVELOPMENT AND ENGINEERING CENTER

Product Assurance and Test Directorate

Picatinny Arsenal, New Jersey

US ARMY
ARMAMENT MUNITIONS & CHEMICAL COMMAND
ARMAMENT RDE CENTER

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operation and reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE October 1993	3. REPORT TYPE AND DATES COVERED Final/1990-1993	
4. TITLE AND SUBTITLE THE READINESS GROWTH MODEL: A QUANTITATIVE ANALYSIS OF SOFTWARE RISK			5. FUNDING NUMBERS
6. AUTHOR(S) David R. Castellano, Paul E. Janusz, and Sharyn L. Tolochko			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESSES(S) ARDEC, PA&TD Technology Division (SMCAR-QAH) Picatinny Arsenal, NJ 07806-5000			8. PERFORMING ORGANIZATION REPORT NUMBER Technical Report ARPAD-TR-93003
9.SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(S) ARDEC, IMD STINFO Br. (SMCAR-IMI-I) Picatinny Arsenal, NJ 07806-5000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) The readiness growth model (RGM) developed in 1990 is a methodology used to quantitatively assess software maturity and, in turn, software readiness for test. It is comprised of data from many low level metrics, and provides a "bottom line" assessment which managers can use to track software progress throughout the lifecycle. RGM can be used to assess the quality of both the process and the product, can identify problematic areas, and can determine associated risk. All of this information is summarized quickly by means of its graphical output and metric data. RGM has been successfully implemented on various U.S. Army Armament Research, Development and Engineeirng Center systems, supporting executive software/system readiness and fielding decisions. It has received high level endorsement from the program managers, and the Army and Office of the Secretary of Defense leadership that have applied it to date.			
14. SUBJECT TERMS Software maturity Readiness for test Software metrics			15. NUMBER OF PAGES 25 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR

NSN 7540-01 280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

ACKNOWLEDGEMENT

The readiness growth model (RGM) was developed by utilizing the vast experiences and innovative ideas of the Technology Division. The RGM's success to date can be attributed to the dedicated and persistent work of these people:

Michael Bucknor
David Castellano
Michael Cernek
James Ferree
Joseph Gombos
Paul Janusz
Wayne Lee
Ed Loniewski
Patricia Lyon
John Mardo
Geza Pap
Albert Stanbury
Sharyn Tolochko
Stanley Truong
Allison Willis

CONTENTS

INTRODUCTION

This report examines several software metrics which can be used to determine and monitor software maturity and readiness for test. It also discusses some of the experiences gained through use of these metrics on Army weapon systems, and how this work led to the development of the readiness growth model (RGM).

BACKGROUND

The U. S. Army Armament Research, Development, and Engineering Center (ARDEC), at Picatinny Arsenal, New Jersey, performs research, development, and engineering on direct fire, close combat systems ranging from bayonets to tank cannons, and indirect fire support systems such as artillery, mortars, ammunition, mines, countermines, and demolitions. The software quality engineering (SQE) Branch of the Product Assurance and Test Directorate (PA&TD) is responsible for the assessment of the software embedded within these systems. These software quality assessments are achieved through the process of independent verification and validation (IV&V) by performing the following:

- Checking to see if all the software requirements are being met
- Checking that each of the software metrics monitored are within acceptable limits
- Checking that there are no adverse trends denoted by the metrics
- Determining if past errors have been corrected
- Making sure corrective actions are in place when new errors are discovered
- Identifying the risks associated with proceeding to test have been identified
- Having some degree of certainty that the software is mature

One of the problems previously encountered in software development and still faced today is that the lack of software maturity has become the leading cause of system fielding slips. Program Executive Officers (PEOs) and Program Managers (PMs) were not independently measuring the quality of software. Users were not adequately defining software functional requirements, and the test and evaluation community lacked a focus on software in system level tests. This emphasized the already apparent need for a management level tool to support executive software/system readiness and fielding decisions.

To address these concerns, ARDEC SQE developed a methodology incorporating a baseline set of metrics and indicators covering both process and product requirements, quality, and management information into one overall assessment known hereinafter as the readiness growth model (RGM). RGM is an analysis used to quantitatively assess software maturity, and in turn, software readiness for test. It can be monitored throughout the software life cycle as a single, comprehensive approach designed to determine software maturity at strategic points within the development cycle: critical design review (CDR), formal qualification test (FQT), technical test (TT), and operational test (OT). Each strategic point reflects the test results from the previous development, test, and evaluation activities. This method was based on the Department of Defense Standard 2167A (DOD-STD-2167A, dtd. 29 Feb 88) waterfall methodology and can be adapted to spiral, object-oriented, and other development processes. With this method comes flexibility in application and calculation, and the model can be tailored specifically to reflect the software development within a single project/system, or it can be designed heuristically for acquisition planning and source selection purposes.

SYSTEM CHARACTERISTICS

RGM was developed by determining the system characteristics which have a measurable impact upon software maturity. The cumulative RGM makes use of requirements traceability, stability (of requirements/design/code), test coverage, test success, fault profile analyses, and maturity indicators; all of which provide significant information as to the progress and maturation of the software development. These main areas are then subdivided into 26 quantitatively assessed categories which contribute weighting factors ranging from 1 to 10 to a total RGM score of 100. Point allocations were assigned to each characteristic and weighted in such a manner that if the system software merited a perfect rating for each characteristic, then the RGM to enter OT would equal 100 (relating to 100% readiness for test).

A listing of each measurable characteristic which contributes to the total RGM score and provides their corresponding point allocation is provided in table 1. The point allocations for each metric were assigned based upon the criticality of that metric in predicting or contributing to a mature software product. The point allocations are also structured to reflect that the software can, however, be weak in one area and still receive a score indicative of a high state of test readiness; it is a cumulative effect since there is no one single factor upon which the maturity analysis relies. Therefore, the RGM identifies and highlights areas that require attention or improvement.

DETAILED DISCUSSION

The following discussion expands upon each of the 26 RGM categories and the metrics/characteristics on which they are based.

Requirements--Allocation, Trace

The validity of any software design is heavily based on proper flow down of system requirements. System requirements must be properly interpreted into lower-level software requirements to form the basis for algorithm design. These detailed, lower-level requirements are allocated down to the design of the software components and units where the algorithms are developed, and code is written to ultimately satisfy the system level requirements. The adequacy of the lower-level requirements definition, as well as the completeness of the flow down/allocation, are key factors indicative of a solid design foundation. Tracing entails the provision of evidence that a lower level design/test criteria have foundation in a parent document. This section evaluates and assesses the validity of the parent-child relationship.

Stability--Requirements, Design, Code

Requirements stability is a measure of the degree and frequency to which detailed software requirements are changed. Stability is reflected by a lack of change or minimization of change to these requirements, as well as by a conditional acceptance or full approval of the requirements baseline by the customer. It is important to measure and understand changes to requirements in order to assess the end-quality of the products coming out of its development process. Software requirements which are changing as a result of errors/omissions by the producer are cause for concern. Requirements which are changing as a result of scope changes by the customer are indicative of poor a priori planning. In either case, a requirements baseline which is not stable will delay maturation of the software.

Stable requirements will consequently reflect a stable design and eventually lead to stable code. Design stability is a measure of the degree of change in the design, for example, the way in which the computer software configuration items (CSCI's) are divided to address requirements or even readjustments of further divisions into computer software components (CSC's) and units (CSU's). Likewise, code stability is assessed by tracking the number of actual code changes made in a given period of time. Changes resulting from errors/omissions are cause for concern and are indicative of a poor design process. Changes due to fluctuations in the detailed requirements are more serious and incur higher costs for the long-term exemplifying wasted development resources and producing schedule slips.

Test Coverage--Depth, Breadth

Test coverage is a significant portion of the RGM analysis. It is basically a measure of test program completeness starting from the lowest level CSU and working in a bottom-up test strategy until system software requirements are validated.

Depth of test coverage relates to the lower-level CSU/CSC tests. It is a measure of the planned for completeness of tests to execute all paths through the software logic algorithm. Path testing, as it is often referred to, is essential for early error detection; deficiencies in path testing will lead to discovery of errors at higher levels of test, thereby contributing to the late maturation of the software design. Significant negative impact to cost, schedule, and risk can result from improper and/or incomplete path testing.

Breadth of test coverage relates to the actual validation of software requirements at all levels, from pre-formal Qualification Tests (e.g., informal testing) through the CSCI level up to the system-bench (integrated CSCI level testing) and system-field levels (hardware and software platform integration). Validation testing, as it is often referred to, also employs a bottom-up strategy which is essential for quantifying the degree of compliance to all software requirements. Informal test data are used to determine the software readiness for FQT. Once FQT is conducted on each CSCI and at an integrated level, the test coverage and results are used in determining readiness for TT, which is a system development test conducted by an independent agency prior to customer evaluation. The TT results are, in turn, used in determining the OT (customer/user run test and evaluation) readiness. Regression testing results are used in the readiness for TT and OT RGMs and reflect the percentage of failed tests that were corrected and tested back during FQT and TT from the CSU level. A high percentage of regression testing will imply better breadth of test results and will prevent problems from migrating into higher level tests.

Tests Passed--Depth, Breadth

To correspond with the test coverage (depth and breadth) results, a determination must be made as to the percentage of these tests which successfully passed the stated criteria. Failed tests usually require code, design, or even requirements rework, which severely affects the maturity rate. The combination of these three test categories, depth of test coverage, breadth of test coverage, and test passed (depth and breadth), supply a significant portion of the RGM analysis.

Fault Profiles

Fault profiles are comprised of three analyses each contributing to the RGM analysis: predicted faults, fault rate, and problem change reports (PCRs).

1. The Rome Air Development Center (RADC) metric, also known as the predictive reliability measure, provides a measure of software reliability using an application metric, software development metric, and a design metric with lines of code to predict the number of faults expected to occur during the development of a software system. The predicted number of faults can be determined and then be compared to the actual number of faults encountered in development to calculate percent compliance and its impact on the maturity of the software.

2. Fault rate also contributes to the fault profile section by providing a measurement of the numbers of faults that occur in a given time (e.g., weekly, biweekly, monthly). The objective of this measurement is to reduce faults to a minimum prior to entering test. Higher fault rates are usually demonstrated at the onset of development with a gradual, but constant, progression toward zero.

3. PCRs provide input to fault assessment through the tracking of open PCRs. The number of open PCRs represents problems that are yet to be resolved. This number should approach zero as the total number of PCRs level out at some upper boundary. By tracking the progression of open PCRs, a number for closed PCRs can be calculated and used as a measure of goodness contributing to the overall software maturity.

The combination of the predicted faults, fault rates, and PCR tracking encompass a broad range of fault analysis and therefore provide more accuracy to the fault profile aspect of the software development/maturity assessment.

Process Indicators--Ada Implementation, Code Complexity

Ada implementation and code complexity are two factors that contribute to the overall software maturity assessment. The Ada indicator reflects the degree to which Ada, as a requirement for implementation language or program design language, has been implemented into the lines of code. The higher the percentage of implementation: the more structured the design, thereby increasing supportability and maintainability of the software. The code complexity indicator measures the complexity of low-level software modules. A minimally complex CSCI will contribute a higher score to the maturity assessment since less complex code is easier to test during basis path testing (re: depth of testing) and is also easier to maintain. Work is currently ongoing to bring the Software Engineering Institute's (SEI) maturity level assessment (level 1 to 5) into the model.

DEVELOPING A READINESS GROWTH MODEL AND PLAN

Since it represents a continuously growing assessment, a system's RGM can be monitored throughout the software development. A system's RGM changes as the life cycle progresses since more test data becomes available, past errors are being corrected, and requirements are becoming more clearly defined. The spreadsheet shown in figure 1 can be filled in monthly to track historical data and to plot the data on a program milestone curve.

The optimal RGM values represent the rating that a system would achieve at each phase if everything was being performed correctly and in accordance with the development plan up to that point in time. For example, if everything was going as planned at CDR, the system would be rated at the optimum score for CDR, in this case, 20.5. If everything continued to be performed correctly up to FQT level A/B, the system would be rated at the optimum score at FQT level A/B, in this case, 55.5. Likewise, the optimum value at FQT level C would be 85. Then, using these optimal values and the system schedule acquired from the statement of work (SOW) or software development plan (SDP), a readiness growth plan or curve can be constructed. An insight into a system's growth plan, showing its planned RGM values at CDR, FQT, TT, and OT is provided in figure 1.

The system's updated RGM values can also be graphed over time and compared to its corresponding growth plan. By doing so, it is possible to detect various trends in the RGM values. If a system's current RGM values are above the growth curve, this is typically indicative of a system ahead of schedule, while one that is below the growth curve is indicative of one which is behind schedule. However, a further analysis of the metric data should be performed to gain a proper detailed assessment of the project's status.

SYSTEM TRENDS

Marked improvements in the RGM values can be noted over time and can also show a project falling behind schedule. For example, in figure 2, it can be seen that CSCI X RGM values are for the most part above the growth plan, indicating that the project is ahead of schedule. However, notice that the RGM values become constant. This could indicate several possible scenarios. At this point in time, problems could have been detected as a result of testing, a change in requirements, or a change in design, therefore slowing the project development. The individual metric data could then be examined to determine the source of the problem. In contrast, looking at CSCI Y, the RGM values indicate that the project was behind schedule, and when the contractor's schedule allowed for it, or when additional personnel were assigned to the project, the contractor made up the schedule difference. Again, the individual metric data should be examined to determine the problem.

SYSTEM READINESS

In order to provide a level of readiness, the RGM values can be used with GO/NO GO thresholds or risk zones established at each readiness decision point. To do this, the optimum scores for entering CDR, FQT, TT, and OT are determined by placing break points (table 1) in the RGM and totaling the score for the characteristic information necessary to make a readiness decision. The scores are as follows: 20.5 for CDR, 55.5 for FQT, 85.0 for TT, and 100 for OT. Therefore, the risk zones shown (fig. 1), identify the minimum acceptable percentage for entering each decision point.

Ideally, a system's RGM should be above the risk zone. This would indicate a full GO position for the software to enter the next phase of the development. Values in the risk zone indicate that problems exist that may hinder testing. These problems must be addressed during the decision process before entering a formal test. Values below the risk zone suggests that the software is not mature enough to proceed to the next phase of the development. For values in or below the risk zone, specific details on the problems encountered can be gathered by analyzing the individual metric data. The associated risks can then be identified and managed, and trends can be predicted to determine future readiness for test.

RGM USERS

The RGM can be used by several organizations and is a mediator between the metrics data and the decision maker. The PM, materiel developer, and software developer can use it to manage their software development and use it to assess their project status, determine where the project stands, where the project should be, and where the project is going. Problematic areas can be identified early, and the proper corrective actions can be taken. The independent assessor can use the RGM to assess the quality of both the process and the product, can identify problematic areas, and can determine associated risk. The technical and operational testers can use the RGM to determine readiness for testing. All of this information is summarized quickly by means of the graphical output and metric data.

FLEXIBILITY OF THE APPROACH

A list of the metrics chosen for a system which have a measurable impact upon software maturity is given in table 1. These metrics and their corresponding point allocations are tailored for each system depending upon the unique characteristics of the software development and system. Also, as described here, the metrics are displayed by the various phases of the life cycle. In this case a waterfall software development approach is used. The RGM methodology can be tailored accordingly for other software development approaches such as rapid prototyping, spiral model, etc.

Currently, the risk zones are established using several factors, starting with the chosen metric point allocations. The risk zone threshold values and their ranges (i.e., the placement and width of the risk zone) are based on engineering judgment and past system's software characteristics versus test performance. Although this approach is subjective in nature, the threshold ranges are being updated with experience, and are validated using the empirical data available from the various projects and applications used to date. Again, these values can be tailored to each individual system. However, in order for the RGM methodology to become truly effective in determining the software readiness, the PM must accept the risk zones as being his targets and thresholds for his system.

ARDEC has established corresponding goals and requirements which constitute acceptable values and limits for each of the measurable characteristics. Determining a point allocation for a measurable characteristic is also subjective in nature. Typically, the characteristic is rated by determining its significant contribution to software maturity. For example, requirements analysis and testing are the most important characteristics used to validate system performance. Again, the scoring approach taken can be tailored to each system. Accordingly, the RGM requires constant calculation and updates with the most recent available data, even if already calculated for one of the previous decision points. The RGM is usually updated monthly but can be updated more frequently during key periods, such as updating weekly when approaching a readiness decision.

SUCCESSFUL APPLICATIONS TO DATE

To date, the RGM has been successfully implemented on six ARDEC weapon systems, ranging in size from 8 thousand lines of code (KLOC) to 220 KLOC, each written in Ada. It is currently being applied to 14 others written in a variety of languages from Ada to Assembly and C. With respect to the RGM, each of these systems received full support and endorsement from their respective PMs and PEOs. These PMs used the RGM to help manage their software development and track their progress against their growth curve. On two systems, the RGM charts noted adverse trends as previously discussed. Further investigation of the metric data and discussions with the software developer revealed problems linked to the software development process. Accordingly, the necessary corrective actions were identified and implemented early, therefore avoiding major milestone failures. The RGM has also gained acceptance by various organizations throughout the Department of the Army (DA). Operational testers are using the model as a readiness indicator for software. The RGM is being considered as a standard methodology for all major subordinate commands (MSCs), and potentially across DA and the Department of Defense (DOD). The production and logistics component of the Office of the Secretary of Defense (OSD) has adopted the RGM in support of Defense Acquisition Board (DAB) reviews for software.

COST

Since the RGM uses several software metrics, the costs associated with the metric data collection may be of concern. It has been ARDEC's experience to date that if the software developer has a mature software development process, the costs associated with collecting these metrics is minimal. This stems mainly from the fact that the software developer has already been collecting the required metric data as part of their normal operations. An initial estimate of the costs associated with this data collection is approximately one quarter of a man year per year. Also, most data are currently collected manually. Substantial cost reductions can occur through the use of automated tools. ARDEC is working to collect more information on the cost of implementing these metrics by monitoring these costs on its current programs. However, if the software developer does not have a mature software development process and has not previously collected these metrics, the associated costs will be substantially higher, estimated at 20 to 30 times higher. These types of developers generally introduce very high risks on the program as a whole.

INITIAL REACTIONS TO RGM AND LESSONS LEARNED

Initial reactions to the RGM have been highly favorable and successful. Management appreciates the RGM because, in one easy to understand chart, they can get an indication of how their software stands, how mature it is, and if it's ready for testing and/or fielding. Since the RGM is highly flexible, it can be, and should be, tailored to each system. This way the PM, and other users of the RGM, can concentrate on the software issues which are of concern and of value to them. Through use of the software metrics, management is more aware of potential problems, in both the process and the product, since they now have a quantifiable means of monitoring these issues. Monthly updates of the metrics and charts keep the PM informed.

Through ARDEC's application of the RGM on several weapon systems, several lessons learned have been identified:

o Some of the metrics initially chosen to characterize the software may not be implemented in such a fashion that truly represents the status of the software. Because the RGM methodology has the advantage of being flexible, these metrics and their corresponding methodology can be tailored and adapted accordingly so that this situation is corrected. Consequently, the benefits and shortfalls of each metric are discovered. Over time and continued use, these individual metrics will eventually be validated.

o If used as a stand alone indicator, each metric provides valuable information and is quite useful. However, in some instances when the metrics are consolidated into the RGM, some of the utility of the metrics may be lost due to the consolidation of all the data into the final RGM figure. Also, while using the software metrics purely as indicators has certain advantages, the original intent and a more effective use of the metrics would be to integrate the metrics into the actual software development process, not just the product. By doing so, the user can adapt a proactive metrics process, as opposed to a reactive one.

o To date, the RGM has been applied to a wide variety of weapon system developments, each having its unique features and experiences. Because of such, a database should be established to track past system performance. Having these data would help establish the metrics and thresholds, risk zones, growth curves, and confidence factors.

o Most importantly, the PM must own the planned RGM curve.

o Data collection for the RGM methodology is currently performed using manual methods, which can be tedious, laborious, and even subjective. Data collection can, and should, be improved through the use of automated tools. A cost accounting system should also be established to monitor the costs associated with collecting the required RGM data.

CONCLUSIONS

The readiness growth model provides an integrated approach to software assessment, and is an effective risk management tool. Using a comprehensive software metrics program, it focuses management attention on software development progress and can be used to support executive software/system readiness and fielding decisions. It has been successfully implemented on various ARDEC systems and has received high level endorsement from the program managers and the Army and OSD leadership which have applied it to date.

Table 1. Readiness growth model (RGM) software characteristics

CHARACTERISTIC	POINT ALLOCATION	OPTIMUM
REQ'MTS ALLOCATION/TRACE	10	
REQ'MTS STABILITY	8	
DESIGN STABILITY	2.5	20.5
CODE STABILITY	2.5	
TEST COVERAGE (DEPTH)		
CSC	2	
CSU	1	
TEST PASSED (DEPTH)		
CSC	2	
CSU	1	
PQT TEST COVERAGE (BREADTH)		
% OF REQ'MTS TESTED/CSCI	6	
TEST PASSED (BREADTH)		
% TESTS PASSED/CSCI	6	
FAULT PROFILE		
RADC - Predictive Reliability	3	
FAULT RATE	3	
% PCR'S OPEN	5	
ADA IMPLEMENTATION	1.5	
CODE COMPLEXITY	2	55.5
FQT TEST COVERAGE (BREADTH)		
% REQ'MTS TESTED/CSCI	6	
SYSTEM-BENCH	4	
SYSTEM-FIELD	2	
TEST PASSED (BREADTH)		
% REQ'MTS TESTED/CSCI	6	
SYSTEM-BENCH	4	
SYSTEM-FIELD	2	
REGRESSION TESTING		
FQT CSCI	3	
SYSTEM-BENCH	2.5	85.0
TEST COVERAGE (BREADTH)		
TECHNICAL TEST	5	
TEST PASSED (BREADTH)		
% PASSED	7	
REGRESSION TESTING	3	100

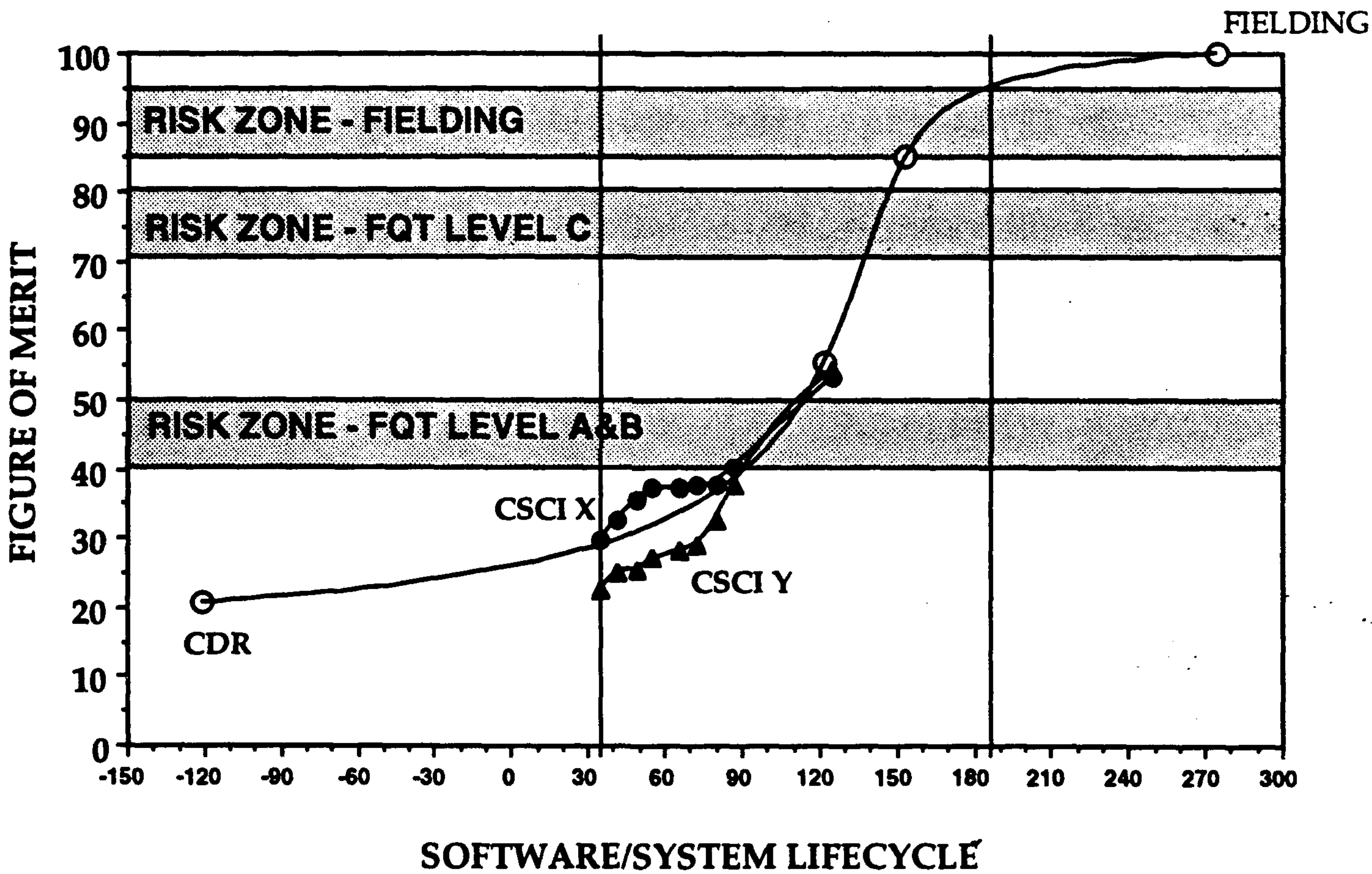

Figure 1. Example readiness growth model

BIBLIOGRAPHY

1. Science Applications International Corporation, "Methodology for Software Reliability Prediction," Final Technical Report RADC-TR-87-171, Vols I and II, Rome Air Development Center, Griffiss Air Force Base, NY, Nov 87.

2. Software Metrics in Test and Evaluation , Draft Version, Initial Staffing, U.S. Army Armament Research, Development, and Engineering Center (ARDEC),June 1992.

GLOSSARY

ARDEC	U.S. Army Armament Research, Development and Engineering Center
CDR	Critical design review
CSC	Computer software components
CSCI	Computer software configuration items
CSU	Computer software units
DA	Department of the Army
DAB	Defense Acquisition Board
DOD	Department of Defense
FQT	Formal qualification test
IV&V	Independent verification and validation
KLOC	Thousand lines of code
MSC	Major subordinate command
OSD	Office of the Secretary of Defense
OT	Operational test
PA&TD	Product Assurance and Test Directorate
PCR	Problem change reports
PEO	Program Executive Officer
PM	Program Manager
RADC	Rome Air Development Center
RGM	Readiness growth model
SDP	Software development plan
SEI	Software Engineering Institute
SOW	Statement of work
SQE	Software quality engineering
TT	Technical test

DISTRIBUTION LIST

Commander
Armament Research, Development and Engineering Center
U. S. Army Armament, Munitions, and Chemical Command
ATTN: SMCAR-IMI-I (3)
SMCAR-QAH-A, M. Bucknor
D. Castellano (2)
M. Cernek
M. Connell Jr.
M. Cox
D. Edmiston
J. Ferree
J. Gombos
P. Janusz (2)
J. Korec
W. Lee
C. Lopez
P. Lyon
G. Pap
R. Payne
A. Stanbury
S. Tolochko (2)
L. Torres
S. Truong
SMCAR-QAH-P, E. Loniewski
SFAE-AR-TMA
Picatinny Arsenal, NJ 07806-5000

Commander
U. S. Army Armament, Munitions, and Chemical Command
ATTN: AMSMC-GCL (D)
Picatinny Arsenal, NJ 07806-5000

Administrator
Defense Technical Information Center
ATTN: Accessions Division (12)
Cameron Station
Alexandria, VA 22304-6145

Director
U. S. Army Materiel Systems Analysis Activity
ATTN: AMXSY-MP
Aberdeen Proving Ground, MD 21005-5066

Commander
Chemical/Biological Defense Agency
U. S. Army Armament, Munitions, and Chemical Command
ATTN: AMSCB-CII, Library
Aberdeen Proving Ground, MD 21010-5423

Commander
U.S. Army Edgewood Research, Development and Engineering Center
ATTN: SCBRD-RTT (Aerodynamics Technical Team)
Aberdeen Proving Ground, MD 21010-5423

Director
U.S. Army Research Laboratory
ATTN: SRL-OP-CI-B, Technical Library
Aberdeen Proving Ground, MD 21005-5066

Chief
Benet Weapons Laboratory, CCAC
Armament Research, Development and Engineering Center
U. S. Army Armament, Munitions, and Chemical Command
ATTN: SMCAR-CCB-TL
Watervliet, NY 12189-5000

Director
U. S. Army TRADOC Analysis Command-WSMR
ATTN: ATAA-SL
White Sands Missile Range, NM 88002

HQ AMC
ATTN: AMCCE-QE, J. Byers
5001 Eisenhower Ave.
Alexandria, VA 22333-0001

Commander
USA CECOM
ATTN: AMSEL-ED
Ft. Monmouth, NJ 07703-5000

Commander
USA MICOM
ATTN: AMSMI-QA, E. Dalrymple
Redstone Arsenal, AL 35898-5000

Commander
USA TACOM
ATTN: AMSTA-QA
Warren, MI 48397-5000

Commander
USA TROSCOM-BRDEC
ATTN: STRBE-TQ, L. Makowsky
Ft. Belvoir, VA 22060-5606

Commander
USA TECOM-WSMR
ATTN: STEWS-TE, M. Wygant
White Sands, NM 99002-5070

Commander
USA AVSCOM
ATTN: AMSAV-Q
St. Louis, MO 63120-1798

Commander
USA AMCCOM/CRDEC
ATTN: AMSMC-QAV-A
Edgewood, MD 21010-5423

Deputy Under Secretary of the Army (Operations Research)
ATTN: SAUS-OR
Washington, DC 20310

Commander
USA OPTEC
ATTN: CSTE-PM
Alexandria, VA 22302-1458

Program Executive Officer, Armaments
ATTN: SFAE-AR
Picatinny Arsenal, NJ 07806-5000

OASD (P&L) IEQ
ATTN: C. Wingfield
5203 Leesburg Pike
Suite 1403
Falls Church, VA 22041-3466

NAWCWPS
ATTN: Tom Rozanski
Code P03931
Pt. Mugu, CA 93042

DSMC
FD-8M
ATTN: J. Dobbins
Ft. Belvoir, VA 22060

NUWC Division Newport
ATTN: J. McGarry
Code 2233
Bldg. 1171/2
Newport, RI 02841

Commanding Officer
ATTN: G. Robertson
Bldg. C-60, Code 924
NCCOSC RDTE Div
53560 Hull Street
San Diego, CA 92152-5800