SOFTWARE RISK ASSESSMENT MODEL

Say-Wei Foo and Arumugam Muruganantham
National University of Singapore
Email: elefoosw@nus.edu..sg

ABSTRACT

The software industry has grown rapidly with the increasing applications of information technology. Just like any development project, there is inherent risk in software development projects. Failure to develop software according to specifications or within budget, or to deliver software products on time can be costly. In this paper, a new model for the assessment of risk in software projects is proposed. The model, named Software Risk Assessment Model (SRAM), makes use of a comprehensive questionnaire. Test results show that using the risk indicator obtained from the SRAM, it is possible to predict the possible outcome of software projects with good accuracy.

Keywords: Software, Risk Assessment, Risk Management, Model

1. INTRODUCTION

The cost of software product can be very high and failure to produce the product on time, with the estimated resources and according to specifications can be costly and damaging. Risk in the context of software engineering is defined as the factors that may cause late delivery, cost overrun or low quality of a software product.

A typical software development process consists of identification of product requirements, analysis of the requirements, design of the product, coding, unit testing and system testing. In the process of development, it may be necessary to backtrack to previous stages should problems be encountered in any stage. Problems encountered in any of the stages could result in delay in delivery and cost overrun. It is important that risks are assessed before development work begins and careful attention is paid to risk management as a wrong move at a single stage of the software development process may result in overall failure of the project.

Risks are associated with certain factors or risk elements. Some of the important risk elements are complexity of the software, productive level of staff, flexibility of schedule etc.

Systematic risk analysis and management is an organized way of identifying risk factors, analyzing them, assessing their impacts on the project and mitigating them when they arise. Systematic risk analysis and management may be facilitated through the use of a risk assessment model that provides quantitative assessments of the risks involved. In this paper, a new risk assessment model is proposed. The proposed model is named Software Risk Assessment Model (SRAM).

In Section 2, two existing risk assessment techniques and their limitations are presented. The proposed SRAM model is described in Section 3. Detailed explanation of the calibration of the model and testing of the model are given in Section 4 and Section 5 respectively. The concluding remark follows in Section 6.

2. EXISTING RISK ASSESSMENT MODELS

There are a few published models that evaluate the risk of a software project. In [1], a method to evaluate risk using risk drivers is described. Risk Driver method is conceived following US Air Force's guidelines for software risk identification and abatement. The US Air force defined the major risk components as performance risk, cost risk, support risk and schedule risk. This model, however, does not have questions that bring out process related risks and is more suited for acquisition than development of software.

Another model named Software Engineering Risk Model (SERIM) [2] focuses on three risk elements: technical risk, cost risk and schedule risk. Similar to the Risk Driver model, SERIM is based on subjective probability. SERIM defines a hierarchical probability tree formulated by risk elements, risk factors and risk metrics of decision alternatives. The model, however, does not take into account of the software complexity issues, which plays an important role in determining the risk for the software project. It also does not account for issues related to requirements. Requirements contribute significant risk to the project.

3. PROPOSED SOFTWARE RISK ASSESSMENT MODEL (SRAM)

3.1 Concept

The proposed risk assessment model considers the following nine critical risk elements: complexity of software, staff involved in the project, targeted reliability, product requirements, method of estimation, method of monitoring, development process adopted, usability of software and tools used for development. A set of questions is carefully chosen for each of these elements with three choices of answers each. The answers are arranged in increasing order of risk.

For example, Software Complexity is one of the risk elements of software projects. The higher the complexity of the software, the higher is the risk. One of the questions to assess the risk associated with the complexity of software is:

Q1. What is the function of the software to be developed?
Choices:
a. Data processing software
b. Service software (Communication software)
c. System software

The risk assessor will pick one of the three choices based on the nature of the project and the actual situation. Choice a. is given a risk rating of one, Choice b. a rating of two and Choice c. a rating of three. In the above example, system software is considered the most complex among the three types of software as it has to interact with the hardware and facilitates the operation of other types of software. Hence a rating of 3 is given if the type of software to be developed is system software. Data processing software, on the other extreme, only deals with local data and hence is deemed to be the least complex of the three and hence a rating of 1 is given for this choice.

Due to the scope of this paper, the questionnaire relating to the SRAM is given in Appendix 1 without detailed explanation and justification of the questions and choices.

3.2 Assessment of the overall risk level of a project

For all questions with respect to a particular risk element, choices are obtained and their numerical ratings are accumulated. The normalized value for the risk element is obtained by dividing the accumulated value by the number of questions attempted for the particular risk element. This value is given the name of risk element probability. The risk elements may be ranked according to the risk element probability. Manager can then decide to allocate resources proportional to the magnitude of risk of the risk elements.

Let the nine risk element probabilities be denoted by $\mathbf{r_1, r_2, .., r_i, .., r_9}$. As the nine risk elements have different degree of impact on different types of software project, different weights may be assigned to these elements when combining the risk element probabilities to derive at the overall risk value for the project. Let the weights assigned to the elements be denoted by $\mathbf{w_1, w_2, .., w_i, .., w_9}$. The risk level $\mathbf{R}$ of the project is then computed as $\mathbf{w_1\ r_1 + w_2\ r_2 + \ldots + w_i\ r_I + \ldots + w_9\ r_9}$.

If the maximum rating for all questions is 3 and the minimum rating is 1, then the maximum value of $\mathbf{R}$, $\mathbf{R_{max}}$, is given by $\mathbf{R_{max} = w_1\ 3 + w_2\ 3 + \ldots + w_i\ 3 + \ldots + w_9\ 3 = 3\ (w_1 + w_2 + \ldots + w_i + \ldots + w_9)}$. Similarly, the minimum $\mathbf{R}$, $\mathbf{R_{min,}}$, is given by $\mathbf{R_{min} = 1^*(w_1 + w_2 + \ldots + w_i + \ldots + w_9)}$. The overall risk level $\mathbf{R}$ may then be normalized as follows:

$$\textbf{Normalized R} = \mathbf{R_n = (R - R_{min}) / (R_{max} - R_{min})}$$

The normalized $\mathbf{R_n}$ provides the risk level of the assessed project as a fraction between 0 and 1. $\mathbf{R_n}$ for project with the lowest risk (no risk) is 0 and $\mathbf{R_n}$ for project with the highest risk is 1. We shall refer to this normalized value for a project as project risk.

3.3 Assessment of the Quality, Schedule and Cost risks of a project

The level of risk of the project in relation to quality, schedule and cost can also be assessed separately based on the risk element probabilities obtained. This is done by assigning different weights to the probabilities according to the impact of the associated risk elements on quality, schedule and cost respectively.

For example, if the following table of impact is used, The risk elements Complexity, Staff, Reliability and Requirements have highest impact on quality of the project hence the largest value of weight (say 3) is to be assigned to the risk element probabilities of these risk elements. Monitoring and Tools have the least impact on quality and hence the smallest value of weight (say 1) is to be used. Estimation, Development Process and Usability have medium impact on quality and hence the value of weight assigned to the associated risk element probabilities should be values between the two extremes (say 2). The normalized value $\mathbf{R}_n$ computed using these values of weight and the risk element probabilities then gives an indication of the level of risk of the project in relation to quality.

Table 1: Level of Impact of Risk Element on Quality, Schedule and Cost

Risk Elements	Quality	Schedule	Cost
Complexity	HIGH	HIGH	HIGH
Staff	HIGH	HIGH	HIGH
Reliability	HIGH	MEDIUM	MEDIUM
Requirements	HIGH	MEDIUM	MEDIUM
Estimation	MEDIUM	HIGH	HIGH
Monitoring	LOW	MEDIUM	LOW
Development Process	MEDIUM	MEDIUM	MEDIUM
Usability	MEDIUM	LOW	LOW
Tools	LOW	LOW	LOW

The level of risk in relation to schedule and cost can be obtained in a similar fashion.

4. CALIBRATION OF SRAM

The relative level of risk of projects may be compared using the project risk values. Alternatively, by calibrating the project risk values with the Customer Feedback Index using past projects, customer satisfaction or outcome of a project may be predicted based on the calculated project risk value. Note that the Customer Feedback Index is calculated by the customer department of the company based on data such as call throughput, system availability, number of defects and usability of the released product obtained from customers. It is an indication of the customer's satisfaction of the project and the degree of success/failure of the project. A value between 0 and 12 is obtained. A value of 12 indicates the highest level of satisfaction and a value 0 indicates the lowest level of satisfaction.

Ten projects of a multinational company are used to calibrate the SRAM. For simplicity, these projects are named P1 to P10. The projects are all related to mobile software and degree of impact of the risk elements on Quality, Schedule and Cost of the projects are similar. A common set of values of weight is used. The values are those given in Table 1 with HIGH=3, MEDIUM=2 and LOW=1.

The overall project risk is obtained by taking the average of the risk values calculated for Quality, Schedule and Cost. These values and the Customer Feedback Index of all 10 projects are tabulated in Table 2.

Table 2 Values of Project Risk and Customer Feedback Index

Project	Quality	Schedule	Cost	Overall	Customer Feedback Index
P1	0.633	0.628	0.629	0.63	3
P2	0.130	0.735	0.727	0.53	5
P3	0.228	0.209	0.217	0.22	10
P4	0.217	0.197	0.205	0.21	11
P5	0.570	0.571	0.569	0.57	4
P6	0.452	0.466	0.470	0.46	6
P7	0.505	0.519	0.514	0.51	7
P8	0.376	0.399	0.396	0.39	8
P9	0.388	0.384	0.399	0.39	7
P10	0.248	0.244	0.248	0.25	10

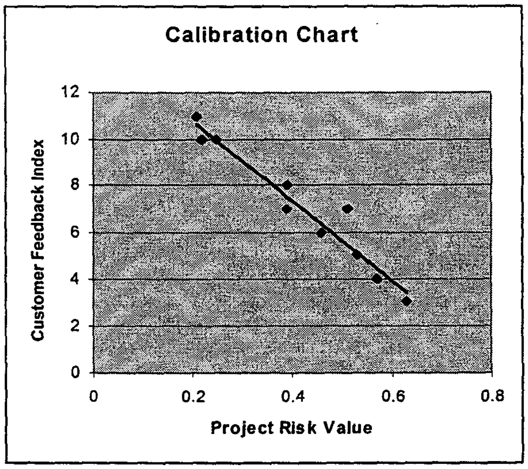

Figure 1: Customer Feedback Index vs. Project Risk Value

The Customer Feedback Index is plotted against the corresponding value of project risk for all 10 projects and is shown in Figure 1. A regression line is drawn through all the ten points. This line then serves as a calibration chart between overall project risk and Customer Feedback Index.

From data of the past software projects in the company, the Customer Feedback Indices for unaccepted projects are found to be below 4. The

Customer Feedback Indices for projects completed with extended schedule, low quality or cost overrun, are between 4 and 8. The Customer Feedback Indices for projects completed in time, within budget and according to specifications are between 8 and 12.

From Figure 1, it can be seen that there is a close correlation between the project risk value and the Customer Feedback Index.

If the risk level of a project assessed by SRAM is above 0.6, the corresponding Customer Feedback Index is below 4. This indicates that it a high-risk project. In this case, it is strongly advised to discontinue or not to undertake the project.

If the risk level of a project assessed by SRAM is between 0.36 and 0.6, the corresponding Customer Feedback Index is between 8 and 4. It is necessary to reduce the likelihood and impact of the risk elements through immediate risk containment procedures. The project manager shall identify the weak areas that require urgent attention and resources.

If the risk level of a project assessed by SRAM is below 0.36, the corresponding Customer Feedback Index is above 8, the project is a low risk project and the chances of success are high.

Table 3: Values of Project Risk

Project	Quality	Schedule	Cost	Overall
P11	0.444	0.433	0.435	0.437
P12	0.201	0.216	0.219	0.212
P13	0.485	0.488	0.484	0.486
P14	0.34	0.328	0.322	0.33
P15	0.285	0.282	0.283	0.283
P16	0.210	0.208	0.204	0.207
P17	0.168	0.161	0.159	0.163
P18	0.255	0.266	0.264	0.262
P19	0.364	0.373	0.366	0.368
P20	0.377	0.393	0.380	0.383

5. TESTING OF THE MODEL

The accuracy of prediction of the SRAM is evaluated by applying the SRAM to ten other projects of similar nature. The results are tabulated in Table 3.

The Predicted Customer Feedback Index is then obtained from the calibration curve of Figure 1 for each of the project using the calculated overall project risk. These are tabulated together with the actual Customer Feedback Indices in Table 4.

Table 4: Comparison of Computed and Actual Customer Feedback Indices

Project	Predicted Customer Feedback Index	Actual Customer Feedback Index	*Variation
P11	6.8	7	+0.2/6.8 = +2.94%
P12	10.8	10	-0.8/Q.108 = -7.41%
P13	5.8	6	+0.2/5.8 = +3.45%
P14	8.5	9	+0.5/8.5 = +5.88%
P15	9.4	9	-0.4/9.4 = -4.26%
P16	10.8	10	-0.8/Q.108 = -7.41%
P17	11.4	11	-0.4/Q.114 = -3.51%
P18	9.7	10	+0.3/9.7 = +3.09%
P19	7.9	8	+0.1/7.9 = +1.27%
P20	7.6	7	-0.6/7.6 = -7.9%

Variation listed in the table is defined as the percentage of deviation of the actual Customer Feedback Index varies from the Predicted Customer Feedback Index.

The mean value of Variation is 0.9% and the standard deviation is 5.0%. Results show that the project risk value is able to predict reasonably well the Customer Feedback Index.

6. CONCLUSION

A new model for the assessment of risk of software projects, the Software Risk Assessment Model (SRAM), is described. The model focuses on nine risk elements and makes use of a set of questionnaire relating to these risk elements.

The risk levels of individual risk element may be computed. The model also allows levels of risk associated with Quality, Schedule and Cost of a project to be calculated separately.

The model is tested using past projects of known outcome. It is concluded that the model is able to assess the level of risk of a software project and predict the possible outcome of the project with reasonable accuracy.

However it is to be noted that the success of a software product not only depends on the development risk but also on the marketing risk. The SRAM only addresses the development risk and does not assess the marketing risk.

REFERENCES

[1] Roger S. Pressman, A Manager's Guide to Software Engineering, McGraw-Hill 1993.

[2] Dale, Walter Karolak, Software Engineering Risk Management, IEEE Computer Society Press 1996.

[3] Alex Down, Machael Coleman and Peter Absolon, Risk Management for Software Projects, McGraw-Hill Book Company 1994.

[4] Marian Myerson, Risk Management Processes for Software Engineering Models, Artech House, Boston, 1996.

[5] Kim Caputa, CMM Implementation Guide, Addison - Wesley 1998.

[6] Software Engineering Guide, Prentice Hall 1996.

[7] Scott E. Donaldson and Stanley G. Siegel, "Cultivating Successful Software Development", Prentice Hall 1997, Pg 55-60.

[8] Dina Berkeley, Robert De Hoog and Patric Humphreys, Software Development Project Management, Ellis Horwood Limited, Pg 23-34.

[9] E.M. Bennatan, "On Time, Within Budget", McGraw-Hill Book Company 1995, Pg 68-74.

[10] E.M. Bennatan, Software Project Management, McGraw-Hill Book Company, Pg 13-24.

[11] Barry W. Boehm, TRW Defence System Group, A Spiral Model of Software Development and Enhancement", Software Management, IEEE Computer Society Press, 1997.

[12] Ronald P. Higuera, Software Risk Management, Technical Report, Software Engineering Institute, Carnegie Mellon University, Pg 9-18.

[13] Roger L. Van Scoy, Software Development Risk: Opportunity, Not Problem, Technical Report, Software Engineering Institute, Carnegie Mellon University, Pg 11-Q.15

[14] L. Hyatt and L. Rosenberg, A Software Quality Model Metrics For Risk Assessment, Presented at European Space Agency Software Assurance Symposium, Utah, April, 1996.

[15] Post Mortem Reports - Singapore Software Centre, Motorola.

APPENDIX 1: QUESTIONNAIRE FOR SRAM

I. Complexity of Software

Q.1 What is the function of the software to be developed?
a. Data processing software
b. Service software (Communication software)
c. System software
Q.2 What is the memory limitation?
a. Allowed the use of any amount of memory
b. Some additional memory allowed
c. Restricted to a specific amount of memory
Q.3 What is the complexity of I/O devices?
a. Standard I/O devices
b. Medium complexity I/O devices (e.g. scanner)
c. High complexity I/O devices (e.g. microphone)
Q.4 What is the time requirements/limits for the functions?
a. No function requires time limits
b. Certain functions require time limits
c. Many functions require time limits
Q.5 What is the hardware (or platform) dependency?
a. Platform independent (e.g. Java programs)
b. Run on some platforms only (e.g. GNU C programs)
c. Highly platform dependent (e.g. Tandem C programs)
Q.6 What type of control operations is used in the programs?
a. Straight line code, few nested structured programming operations
b. Simple nesting, some inter module control, simple message passing
c. Multiple resource scheduling, dynamically changing priorities
Q.7 What type of computational operations is used in the programs?
a. Evaluation of simple expressions
b. Use of standard mathematics and statistical routines, basic matrix/vectors
c. Difficult and unstructured numerical analysis / stochastic data
Q.8 What type of device dependent operations is used in the programs?
a. Simple read / write statements with simple formats
b. I/O procession includes device selection / status checking
c. Device time dependent / micro programmed operations
Q.9 What type of data management operations is used in the programs?
a. Simple array in main memory, simple database queries, updates
b. Multiple input, single output, simple structural changes
c. Highly coupled, dynamic relational and object structures
Q.10 What type of user interface management operations is used in the programs?
a. Simple input forms / reports generations
b. Use of widget sets
c. Complex multimedia, virtual reality

II. Staff involved in the project

Q.1 What is the average experience of a member of the staff with the company?
a. More than 5 years
b. About 3 years
c. Less than a year
Q.2 What is the average coding experience of a member of the staff?
a. Exceptional
b. Average, at least one expert
c. Beginners
Q.3 What is the motivation level of staff?
a. High, enjoys working, no complaints
b. Moderate, some complaints
c. Low, a lot of complaints
Q.4 What is the level of knowledge of staff on the application domain?
a. Have worked on the whole process several times

b. Have worked on several portions of the system
c. Have reading knowledge of the application
Q.5 What is the average number of lines of code (C equivalent) produced per person per day?
a. 30 lines
b. 50 lines
c. More than 70 lines
Q.6 What is the variety or mix of software disciplines / experiences?
a. Good mix of all software disciplines
b. Some disciplines are inappropriately represented
c. Some disciplines are not represented
Q.7 What is the experience level of software managers?
a. Have similar management experience
b. Mix of new managers and experienced managers
c. New managers with software engineering knowledge
Q.8 Does each member of the staff have career plan and goal setting?
a. All members have a goal and career plan reviewed regularly
b. Some members may have career plan but not followed closely
c. No career plan and goal setting
Q.9 What is the level of harmony among the staff?
a. Harmonious and good teamwork
b. Some unhealthy arguments at time
c. No compromises even in the fundamentals
Q.10 Is every employee clear of his/her roles and responsibilities?
a. Clearly understands the roles and responsibilities
b. Fairly understands the roles and responsibilities
c. Confuses about the roles and responsibilities
Q.11 Is there a proper rewarding mechanism?
a. A good performance appraisal and rewarding scheme is in place
b. Adhoc performance appraisal and rewarding system
c. No performance appraisal and rewarding system

III. Targeted Reliability

Q.1 Are there error handling conditions throughout the program?
a. Error handling conditions for every possible instances
b. Error handling conditions for some possible instances
c. No error handling condition within the program
Q.2 How are error conditions handled?
a. Processing continues for any error condition
b. Processing continues for some error conditions
c. Processing discontinues upon any error condition
Q.3 Are error tolerance conditions defined for input and output?
a. All error tolerance conditions are defined
b. Some error tolerance conditions are defined
c. No error tolerance condition is defined
Q.4 Are inputs checked for validity before processing?
a. All inputs are checked
b. Some inputs are checked
c. No inputs are checked
Q.5 Are hardware faults detected and processed in the software?
a. All hardware faults are detected and processed
b. Some hardware faults are detected and processed
c. No hardware faults are detected and processed
Q.6 Is the use of global data types minimized in the software?
a. Few or no global data types are used
b. Some global data types are used
c. Global data types are heavily used
Q.7 Are defect data collected during software integration?
a. All defect data are collected
b. Some defect data are collected
c. No defect data are collected
Q.8 Are defect data logged-in and closed-out prior to the delivery?
a. All defect data are logged in and closed out
b. Some defect data are logged in and closed out
c. No defect data are logged in and closed out
Q.9 Are all the requirements tested?
a. All the requirements are tested
b. Some of the requirements are tested
c. No requirements are tested
Q.10 Is stress testing performed (for the changes in codes)?
a. Stress testing is performed on all software
b. Stress testing is performed for some software
c. No stress testing is performed
Q.11 Who performs the system testing?
a. Independent test team (other than the developers)
b. Independent test team and developers share the testing
c. Developers themselves do the testing
Q.12 Does your company have similar past experience in developing this type of software?
a. Has track record producing similar products
b. Some experience in similar products
c. No past experience

IV. Product Requirements

Q.1 Are all the software requirements identified and/or documented?
a. All requirements are identified and documented
b. Some requirements are identified and/or documented
c. No requirements are identified and documented
Q.2 Is customer involved in the definition of requirements?
a. Customer is heavily involved
b. Customer is partially involved
c. Customer is not involved
Q.3 Does the customer approve all requirements?
a. All requirements are approved by the customer
b. Some of the requirements need customer approval
c. The customer does not approve all requirements
Q.4 Are ambiguous requirements verified through the prototype?
a. Ambiguous requirements are verified through the prototype
b. Some Ambiguous requirements are verified
c. Ambiguous requirements are not verified
Q.5 Are requirements categorized as essential, nice-to-have etc.?
a. All requirements are categorized and prioritized
b. Some requirements are categorized

c. Requirements are not categorized
Q.6 Are there any differences between the customer requirements and software requirements (used by the developers)?
a. No differences i.e. customer requirements are intact
b. Some refinements on requirements
c. Many refinements on requirements
Q.7 Are software requirements frozen before the subsequent phase?
a. All requirements are frozen before the next phase
b. Some requirements are expected to be changed
c. Many requirements are expected to be changed
Q.8 Are software requirements traceable to code?
a. All requirements are traceable to code.
b. Some requirements are traceable to code
c. No requirements are traceable to code
Q.9 Are software requirements traceable to test procedures?
a. All software requirements are traceable to test procedures
b. Some software requirements are traceable to test procedures
c. No software requirements are traceable to test procedures
Q.10 Are all the open action items closed prior to delivery to the customer?
a. All the open actions items are addressed and implemented
b. Some open actions items are addressed and implemented
c. No actions items are addressed and implemented

V. Method of Estimation

Q.1 What is the estimation method used?
a. Bottom-up
b. Analogy, Top-down
c. Other techniques
Q.2 Is there any model used to compute the cost of the project?
a. A suitable cost model is used
b. A model is used for partial cost estimation
c. No cost model is used
Q.3 Is estimation based on past software productivity metrics?
a. Based on the past software productivity metrics (similar projects)
b. Partially based on the past software productivity metrics
c. Not based on the past software productivity metrics
Q.4 Are schedule estimates based on the past software projects?
a. Based on similar past software project metrics
b. Partially based on similar past software project metrics
c. Not based on similar past software project metrics
Q.5 How often are estimates revised?
a. Estimates are updated monthly or on more frequent basis
b. Estimates are updated at the end of the phases
c. Estimates are never updated
Q.6 How accurate are the past schedule estimates compared to actual schedule?
a. Varies within (+/-)5% range of actual schedule
b. Varies within (+/-)50% range of the actual schedule
c. Varies more than (+/-)100% of the actual schedule
Q.7 What is the level of participation of developers in the estimation?
a. All developers participated
b. Some developers participated
c. Only managers prepare the estimate
Q.8 What are the resources allocated for the estimation?
a. All required resources are allocated
b. Some resources are allocated but not sufficient
c. No resources are allocated

VI. Method of Monitoring

Q.1 Are there distinct milestones for each major software effort?
a. Distinct milestones for each development phase
b. Not enough milestones
c. No milestones
Q.2 Is a detailed Work Breakdown Structure (WBS) used to track and report cost and budget for each phase of the software development?
a. Cost structure or WBS exists and properly tracked and reported
b. Cost structure or WBS exist, but not tracked or reported
c. No cost structure or WBS
Q.3 Is there a monitoring system?
a. A monitoring system tracks cost, schedule and earned value
b. A monitoring system exists but does not track schedule and earned value
c. No monitoring system is used
Q.4 How often are the project progress reports created?
a. Weekly
b. Monthly
c. No project report created
Q.5 How often are cost, schedule and earned value reports updated?
a. Monthly or more frequently
b. Updated less regularly
c. Never updated
Q.6 How frequent is the problem/action log updated?
a. Weekly
b. Updated less frequently than a week
c. Problem log or action log system does not exist
Q.7 How often are the records for technical problems updated?
a. Weekly or more frequently
b. Longer than a weekly basis
c. No records are kept
Q.8 How is the schedule plan developed?
a. Bottom up, all members are involved
b. Bottom up, some important members are not involved
c. Top down, developed by one man
Q.9 What is the span of control?
a. Each superior supervises not more than three subordinates
b. Superior supervises between four and six subordinates
c. Superior supervises more than six subordinates

VII. Development process adopted

Q.1 Is software management planning document used for the project?
a. Used and adhered closely

b. Used but not followed closely
c. Not used
Q.2 Are software configuration management functions performed?
a. All software configuration management functions are performed
b. Some software configuration management functions are performed
c. No software configuration management functions are performed
Q.3 Does communication exist between different organizations supporting the development of the software project?
a. Very good communication
b. Reasonable communication
c. Poor communication
Q.4 Are software developers trained in the development methodology?
a. All are trained
b. Some are trained / all trained in some portion of methodology
c. None is trained in the development methodology
Q.5 How closely is the software development methodology followed?
a. It is closely followed by all
b. It is followed by some
c. It is not followed

Q.6 Are software quality functions performed?
a. All of the software quality functions are performed.
b. Some of the software quality functions are performed
c. No software quality functions are performed
Q.7 Is the current development methodology suitable for the project?
a. It is tailored to the process
b. It is a fixed process
c. No process is followed
Q.8 Does the development methodology addresses requirements, design, code reviews / walk-through / inspections?
a. Yes, addresses all
b. Not all of the above
c. Does not address any of the above
Q.9 Does the development methodology require test plans and/or test procedures for all software functions?
a. It requires test plans and/or test procedures for all software functions
b. It requires test plans and/or test procedures for some software functions
c. It does not require test plans and/or test procedures
Q.10 Does the development methodology require documentation?
a. Development methodology requires documentation
b. Development methodology requires partial documentation
c. Development methodology does not require documentation
Q.11 Is regression testing performed?
a. Regression testing is performed for all subsystems
b. Regression testing is performed for some subsystems
c. No regression testing is performed
Q.12 Is there a documented organizational structure in place?
a. Documented organizational structure and operations are in place
b. No documented organizational structure but a clear line of authority is in place
c. There is no documented organizational structure
Q.13 Is the organizational structure stable?
a. No changes in the organizational structure
b. Some organizational changes but not frequent
c. Frequent changes

VIII. Usability of software

Q.1 Will a user manual be written for the software product?
a. User manual will be developed, tested, and delivered with the product
b. User manual will not be verified against the software functions
c. No user manual will be provided for the software product
Q.2 Are there help functions for input or output screen?
a. Help functions are provided for input or output function
b. Some help functions are provided for input or output function
c. No help functions are provided
Q.3 Is the user involved in reviewing prototypes or earlier version of the software?
a. User is involved and feedback is solicited from the user
b. Some feedback is solicited from the user
c. User is not involved
Q.4 Is the user interface designed to industry standards or to standards familiar to the user?
a. Industry standards or standards familiar to the user are followed
b. Some aspects of standards are followed
c. No standards are followed
Q.5 Are user response times identified?
a. All user response times are identified
b. Some user response times are identified
c. No user response times are identified
Q.6 Is the design evaluated to minimize keystrokes and data entry?
a. Entire design is evaluated to minimize keystrokes and data entry
b. Some considerations are made to minimize keystrokes and data entry
c. No design consideration is made to minimize keystrokes and data entry

IX. Tools used for development

Q.1 Are software developers trained to use tools for development?
a. All are trained to use tools
b. Some are trained to use tools
c. No one is trained to use tools or no tool is available
Q.2 Are automated software tools used for testing?
a. Automated tools are used in testing and they are adequate
b. Automated tools are used in testing but they are not adequate
c. No automated tools are used in testing

Q.3 Are automated tools used for code generation (i.e. screen painters)?
a. Adequate automated tools are used for code generation
b. Some automated tools are used but not adequate
c. No automated tools are used
Q.4 Are automated tools used for test procedure generation?
a. Adequate automated tools are used
b. Some automated tools are used but not adequate
c. No automated tools are used
Q.5 Are tools used for configuration management functions?
a. Adequate configuration management tools are used
b. Some configuration management tools are used
c. No configuration management tools are used
Q.6 Are automated tools used for regression testing?
a. Adequate automated tools are used
b. Some automated tools are used, but not adequate
c. No automated tools are used
Q.7 Are automated tools used for re-engineering?
a. Adequate automated tools are used
b. Some automated tools are used, but not adequate
c. No automated tools are used

Q.8 How stable is the compiler/linker/debugger?
a. Stable with few or no problems
b. Somewhat stable, but with some known problems
c. Not stable, with a lot of problems
Q.9 Are required tools readily available to developers when needed?
a. All tools are available
b. Some tools are available
c. No tools are available

End of Questionnaire