A RASCH ANALYSIS ON THE USABILITY MODEL IN APPLYING CMM TO IMPROVE THE QUALITY OF SOFTWARE MAINTENANCE PROCESS

AL-AHMAD HANEEN HASSAN, RODZIAH ATAN

Faculty of Computer Science & Information Technology, Universiti Putra Malaysia.

AZRILAH, A.A., SAIDFUDIN, M

UTMSPACE, Universiti Teknologi Malaysia

ABSTRACT

Maintenance plays an important role in the life cycle of a software product. It is estimated that there are more than 100 billion lines of code in production in the world. As much as 80% of it is unstructured, patched and not well documented. Maintenance can alleviate these problems. IEEE and ISO have both addressed software maintenance, the first with a specific standard and the latter as a part of its standard on life cycle processes. Multiple solutions to problem of software development have been proposed such as development methodology, management model and software tools. Software maintenance suffers because of the scarcity of management model that would facilitate its evaluation. This paper described a model of a quality software maintenance process. The proposed model is based on Capability Maturity Model of the Software Engineering Institute (CMM-SEI) to evaluate and enhance the software maintenance process. The architecture of the CMM model is retained while the development process is either modified or extended to take into account the characteristics specified to the maintenance function. These characteristics were then organized into key process areas as the new CMM model.

Keywords: Software Maintenance, CMM model, Software life cycle, Software development, Maintenance evaluation, Rasch Model

INTRODUCTION

The maintenance process is often taking the greatest share of software corporate resources between 50% and 70% of the software budget is allocated to the maintenance process (Arthur, 1988; Swanson & Beath, 1989; Sharpe et al., 1991). Software maintenance has not received it proportionate share of management attention and that it has suffered from lack of planning, thus resulting typically in crisis management. Within this context, software maintenance is perceived as expensive and ineffective.

There exist management models to evaluate the quality of the maintenance process and to propose improvements. However, there is lack of similar models which take into account the characteristics specifics to the maintenance process. The literature search has not come up with diagnostic techniques to evaluate the quality of the maintenance process of a given organization, nor to identify an improvement path.

Evaluation models must support the following three management objectives:

a. At the operational level they provide a detailed analysis and evaluation of a business process and of its key process.

b. At the tactical level, they identify the strengths and weaknesses of each process as well as a progression path, should there be a decision taken within a continuous improvement process program. They also provide a map to develop an action plan to address the strengths and weaknesses within the set of organizational priorities and allocation of resources.
c. At the strategic level, they provide to the senior executives the relative positioning of their organizations within their competitive environment. Based on this evaluation, priorities are then set, which lead to the allocation of scarce corporate resources to meet the corporate objectives.

The CMM was originally developed to assist the U.S. Department of Defense (DoD) in software acquisition. Contractor performance was included as a factor in contract awards and this has become a guide or framework for software process improvement. It used to judge the maturity of the software processes of an organization and to identify the key practices required to increase the maturity of these processes. It describes the principles and practices underlying software process maturity and is intended to help software organizations improve the maturity of their software processes in terms of an evolutionary path from ad hoc, chaotic processes to mature, disciplined software processes.

The CMM is a framework that describes the key elements of an effective software process. The CMM describes an evolutionary improvement path from an ad hoc, immature process to a mature, disciplined process.

OBJECTIVES

There is lack of usability model to evaluate the maintenance process and lack of proper approach to improve maintenance process which aligned to CMM standard. This study will apply CMM model to evaluate and improve the quality of the software maintenance process. It will also introduce an enhanced usability model named Haneen Usability Model (HUM) in order to evaluate maintenance process based on CMM KPAs.

RESEARCH INSTRUMENT

The study uses survey questionnaire in gaining the users perception on how software maintenance process is being done on their respective organization. The questionnaire has four (4) KPA levels of which are initiatives, repeatability, definability, manageability, and optimisability, which was derived from ISO15504 Software Maintenance Engineering, the CMM and its Key Process Areas (KPA).

It comprises of three sections; the first is on users' demographic details, the second section is to gain the users information on their maintenance process based on ISO15504, CMM and the KPAs, while the third section is gain the users perception on the current used system. Each question has four (4) options; and each is described as in Table 1.

Table 2 Ratings for questionnaire

Rating	Remarks
1	Not sure
2	Not important or not being addressed
3	Partially beneficial or somewhat effective
4	Important
5	Critical or already in place and effective

RESULTS AND DISCUSSIONS

The respondents' demographic details are represented in five (5) characters; Xabcdee comprising of

X - first character indicates whether the respondents are locals – Saudi citizen, which is represented by letter 'L' or foreigner – non Saudi citizen, which is represented by letter 'F'.
a – indicates the respondents gender; '1' represent male and '2' a female
b – indicates the age group where '1' represents age group between 20 to 29
'2' represents age group between 30 to 39
'3' represents age group between 40 to 49
'4' represents are group between 50 to 59
'5' for age group above 60
c – to indicate level of education
d – to indicate years of experience
ee – counter

The respondents are staff from an oil company in Kingdom of Saudi Arabia where N=26. The data are tabulated and analyzed using Rasch analysis software; Winsteps v 3.6.8. The summary statistics in Table 2 shows that the Cronbach-α value is 0.96 to indicate that the instrument used is reliable to reveal the users perception towards their software maintenance process usability. In Rasch, besides providing the Cronbach-α value, it also provides another two (2) reliability check that is the Item reliability and Person reliability. The Person reliability is 0.94 which indicates that the response patterns given by respondents are as expected response. The high value too indicates that it has high possibility of response consistency in which the same team is likely to give same response for similar survey.

Table 3 Summary Statistics

TABLE 3.1 CMMI_Data A ZOU033WS.TXT Apr 10 9:10 2010
INPUT: 26 Persons 18 Items MEASURED: 26 Persons 18 Items 5 CATS 3.68.2

SUMMARY OF 26 MEASURED **Persons**

	RAW SCORE	COUNT	**MEASURE**	MODEL ERROR	INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD
MEAN	61.5	18.0	**0.74**	.33	.97	-.1	.98	-.1
S.D.	16.2	.0	**1.49**	.07	.40	1.2	.40	1.2
MAX.	86.0	18.0	3.67	.55	2.27	2.7	2.25	2.7
MIN.	37.0	18.0	-1.34	.26	.35	-2.9	.36	-2.8

REAL RMSE .35 ADJ.SD 1.45 SEPARATION 4.13 **Person RELIABILITY .94**
MODEL RMSE .33 ADJ.SD 1.46 SEPARATION 4.37 Person RELIABILITY .95
S.E. OF Person MEAN = .30

Person RAW SCORE-TO-MEASURE CORRELATION = .99
CRONBACH ALPHA (KR-20) Person RAW SCORE RELIABILITY = **0.96**

SUMMARY OF 18 MEASURED Items

	RAW SCORE	COUNT	MEASURE	MODEL ERROR	INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD
MEAN	88.8	26.0	.00	.26	.99	-.1	.98	-.1
S.D.	4.6	.0	.30	.00	.39	1.3	.36	1.2
MAX.	98.0	26.0	.51	.26	2.24	3.6	2.07	3.2
MIN.	81.0	26.0	-.61	.25	.54	-1.9	.55	-1.9

REAL RMSE .27 ADJ.SD .12 SEPARATION .44 **Item RELIABILITY .16**
MODEL RMSE .26 ADJ.SD .15 SEPARATION .60 Item RELIABILITY .26
S.E. OF Item MEAN = .07

UMEAN=.000 USCALE=1.000
Item RAW SCORE-TO-MEASURE CORRELATION = -1.00
468 DATA POINTS. LOG-LIKELIHOOD CHI-SQUARE: 1013.18 with 422 d.f. p=.0000

Item reliability's value of 0.16 indicates that the questions are not measuring what the instrument is supposedly to measure. In other words, there are possibilities that the questions cannot give true indication on users' perception towards their software maintenance process. Table 3 shows further that seven (7) items out of 18, which is highlighted by box in the table, have point measure correlation (Pt_Measure Corr.) above 0.8, indicating that these questions were non discriminatory (Bond&Fox, 2007); lack in differentiating management level among the team (Azrilah et *al*., 2008a,2008b). The questions can be improved further so that it can differentiate between a supervisor and a manager. Apart from that, there is no item which has Pt_Measure Corr. below than 0.5 indicating that no items are found to be redundant.

Table 4 Item Measure Table

TABLE 13.1 CMMI_Data A ZOU661WS.TXT Apr 12 10:28 2010
INPUT: 26 Persons 18 Items MEASURED: 26 Persons 18 Items 5 CATS 3.68.2

Person: REAL SEP.: 4.13 REL.: .94 ... Item: REAL SEP.: .44 REL.: .16

Item STATISTICS: MEASURE ORDER

ENTRY NUMBER	TOTAL SCORE	COUNT	MEASURE	MODEL S.E.	INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD	PT-MEASURE CORR.	PT-MEASURE EXP.	EXACT MATCH OBS%	EXACT MATCH EXP%	Item
7	81	26	.51	.26	.54	-1.9	.55	-1.9	.86	.78	65.4	49.3	**A1_Q7 PROTOTYPE**
10	82	26	.44	.26	.65	-1.4	.64	-1.4	.79	.78	73.1	49.1	**A2_Q1 PREVENTIVE PLAN**
11	83	26	.38	.25	.94	-.1	.83	-.6	.84	.78	57.7	48.7	A2_Q2 SYSTEM CONFIG.
12	84	26	.31	.25	.95	-.1	.95	-.1	.81	.77	38.5	47.5	A2_Q3 SMP SIZE PROCEDURE
1	86	26	.18	.25	1.16	.7	1.27	1.0	.66	.77	46.2	46.5	A1_Q1 SOFTWARE MANAGER
9	86	26	.18	.25	.89	-.3	.82	-.6	.85	.77	42.3	46.5	A1_Q9 DESIGN ERROR STATS
2	87	26	.12	.25	.59	-1.7	.60	-1.7	.83	.76	53.8	46.3	A1_Q2 SQA REPORT
4	88	26	.06	.25	2.24	3.6	2.07	3.2	.68	.76	26.9	45.6	A1_Q4 ASSESS DESIGN/CODE
6	89	26	-.01	.25	1.16	.7	1.08	.4	.83	.75	46.2	47.0	A1_Q6 TOOLS&TECHNIQUE
13	89	26	-.01	.25	.70	-1.2	.76	-.9	.76	.75	53.8	47.0	A2_Q4 SUB-CON COMPETENCY
3	90	26	-.07	.25	1.42	1.5	1.36	1.3	.74	.75	46.2	46.9	A1_Q3 INT.DESIGN REVIEW
15	90	26	-.07	.25	1.07	.3	.96	-.1	.70	.75	57.7	46.9	A2_Q6 FLEXIBLE SYSTEM
8	91	26	-.14	.26	.74	-1.0	.89	-.3	.77	.75	42.3	46.8	A1_Q8 TEAMWORK
5	93	26	-.27	.26	1.43	1.5	1.45	1.6	.67	.74	23.1	46.1	**A1_Q5 OBSERVE PLAN**
14	93	26	-.27	.26	.76	-.9	.74	-1.0	.79	.74	57.7	46.1	A2_Q5 AUTO-DATA ANALYSIS
17	94	26	-.34	.26	.82	-.6	.74	-1.0	.81	.73	61.5	46.7	A2_Q8 PROJECT PLAN
18	95	26	-.40	.26	.84	-.5	.87	-.4	.66	.73	46.2	47.5	A2_Q9 PROJECT MGMT.
16	98	26	-.61	.26	.91	-.2	1.06	.3	.68	.71	57.7	49.8	**A2_Q7 CORRECTIVE ACTION**
MEAN	88.8	26.0	.00	.26	.99	-.1	.98	-.1			49.8	47.3	
S.D.	4.6	.0	.30	.00	.39	1.3	.36	1.2			12.3	1.2	

The next information we are looking for in Table 2 is the overall users' perception reflected by the Person Measure Mean; μ_{PERSON}= +0.74*logit* (P[θ]=0.8905) (Azrilah et al., 2008a). This implies that generally the user observed good software maintenance practice. The perceived most observed user or the highest person L142410, is at +1.41*logit* with the most unsatisfied user or the lowest L111102 is at -0.63*logit*. This can be deciphered from Table 2. The most difficult item or the most uncommon practice is A1_Q7 Prototype at +0.51*logit* while the most common practice is item A2_Q7 Corrective Action at -0.61*logit*.
Figure 1 shows the PIDM: tabulates the users' location in a very clear graphical presentation which is easy to read and easier to understand (Zamalia et *al*., 2010). There are 21 foreigners against 5 locals in this organization. In general, the users' separation, G=4.13 is good value that indicates that there is enough differentiation among users observed practices to separate them into distinct practices level (Azrilah et al., 2008b). The practices levels are described from very poor practices which is located at the bottom most of the PIDM in Figure 1, to the most excellent practices located at the top in the PIDM.

Group 1: Not observed software maintenance practice; (foreign, n=5, 100%) do not observe software maintenance practice. All of them are male, two (2) of which are in the age group of 30 to 39, another two (2) between 40 to 49 of age, and one (1) is between 20 to 19 years old. Those users between the ages of 30 to 49 are located above than the junior user with less than 5 working experience. This probably explain why the junior use hardly practice software maintenance however problematic on the senior users where three (3) among them have more than 10 years of working experience in the organization, still do not observe good software maintenance practice.

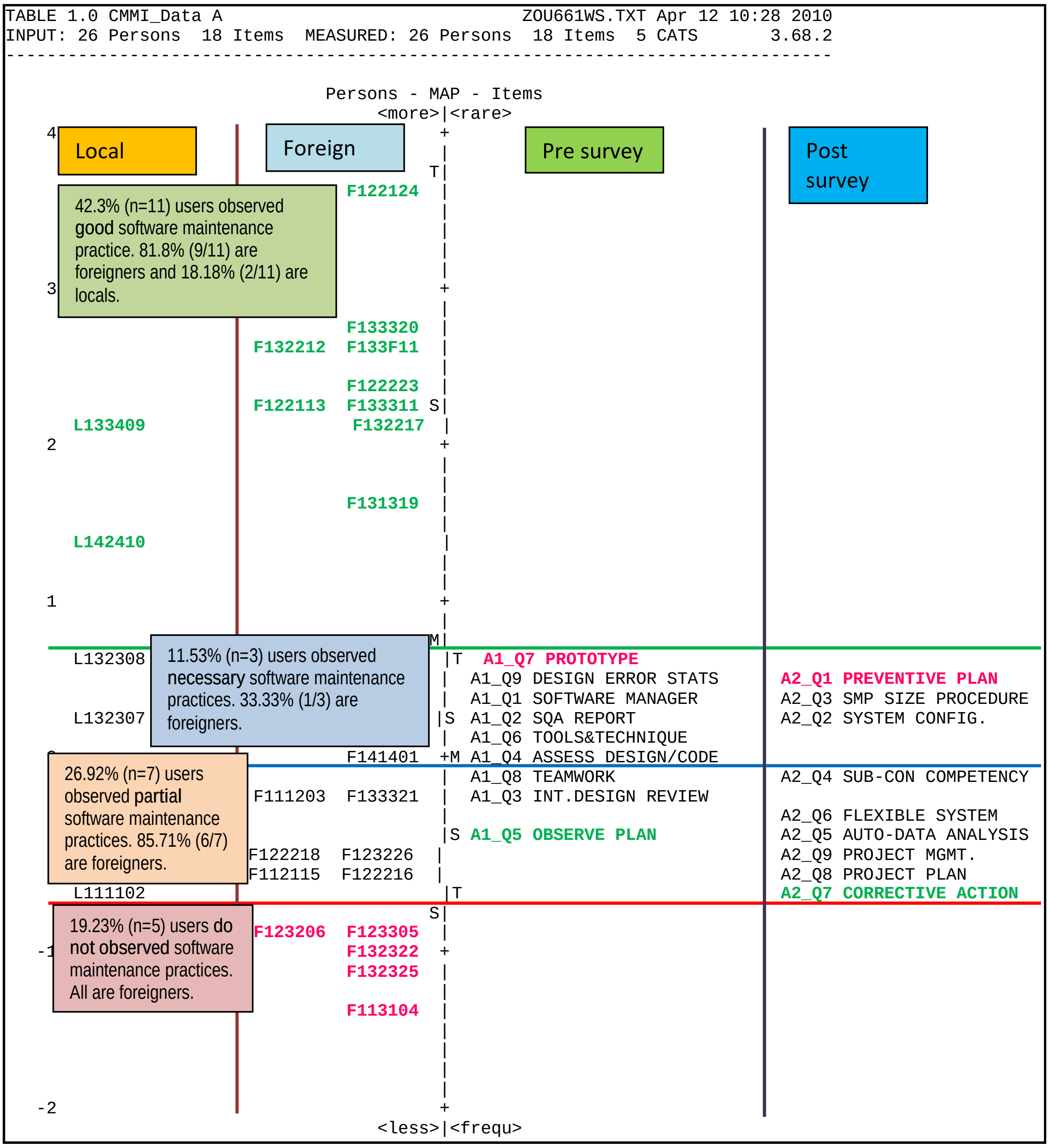

Figure 10 Person Item Distribution Map – Software Maintenance Practices

Group 2: Partially observed software maintenance practice; (foreign, n=6, 85.71%; local, n=1, 14.28%) observed partial software maintenance practice. 28.57% (n=2 out of 7) observe documented plan for developing and improving its software process, and apply internal standard review in their software maintenance. The project is planned and managed in accordance with the maintenance process and apply corrective action plan in achieving quality software. They also anticipated automated analyzing tool in analyzing complex software maintenance requirements and allowing a flexible system. Another 71.43% (n=5 out of 7) only implement corrective action plan, execute project accordingly as planned and managed in accordance to the software maintenance process. However, they do not participate as a team in the software maintenance process and contractors are not awarded based on their performance.

Group 3: Observed necessary software maintenance process; (foreign,n=1, 33.33%; local,n=2,66.67%) Software maintenance is important to them and they apply it where necessary. The most difficult task to them is to develop prototyping methods in designing the critical performance elements of the software, and defect prevention activities are not planned properly.

Group 4: Good Software Maintenance Practice; (foreign,n=9, 81.81%; local,n=2, 18.18%) apply good software maintenance practice.

CONCLUSION

Rasch Model provides a sound platform of measurement equivalent to natural science which matches the SI Unit measurement criteria where it behaves as an instrument of measurement with a defined unit and therefore replicable . It is also quantifiable since it's linear (Wright & Linacre, 1989).

The logit ruler has been developed with purpose to measure the level of software maintenance practice within an organization. It can define the software maintenance practice profile based on their degree of practices according to CMM, KRAs and ISO15504.

It is a noble innovation where the ability 'ruler' can transform ordinal data into measurable scale (Mok & Wright, 2004). It's graphical output is great which gives better clarity for quick and easy decision making (Azrilah et al., 2008b). Rasch enable the measurement of the software maintenance practices within the organization and establish the areas in which they excel and or need enhancement to meet software quality. They need to focus in developing the prototyping methods and to have defects preventive plan.

REFERENCES

Azrilah A.A., Saidfudin, M, Zaharim, A, Azlinah M , Ghulman, H.A. & Sohaimi, Z., 2008c, Evaluation of Information Professionals Competency Face Validity Test using Rasch Model in *Mathematics and Computer Science Engineering: New Aspects of Engineering Education*, a series of reference books and textbooks published by WSEAS Press, July 2008, pp. 396-403.

Azrilah AA, Saidfudin, M, Sohaimi Z, Ghulman, H.A., Mohamed, A & N. Arshad, 2008a, Development of Rasch-based descriptive scale in profiling information professionals' competency, in conference proceeding *ITSIM 2008 Information Technology International Symposium* , Vol.1, Aug 2008, pp. 1-8th. K.Lumpur, Malaysia.

Azrilah AA, Saidfudin, M, NoorHabibah Arshad, Zaharim, A, Azlinah M , Ghulman, H.A. & Sohaimi Z., 2008b, Application of Rasch Model in validating the construct of measurement instrument in *International Journal of Education and Information Technologies*, Issue 2, Vol.2 published by NAUN Press, 2008, pp. 105 -112.

B. D. Wright and J. M. Linacre, "Observations are always ordinal; measurements, however, must be interval," *Archives of Physical Medicine and Rehabilitation,* vol. 70, pp. 857-860, 1989.

M. Mok and B. Wright, "Chap 1: Overview of Rasch Model Families," in *Intoduction to Rasch Measurement: Theory, Models, and Applications* Minnesota: Jam Press, 2004, pp. 1-24.

P. D. Leedy and J. E. Ormrod, *Practical research planning and design*, 8th ed. Upper Saddle River, New Jersey: Pearson Prentice Hall, 2005.

T. G. Bond and C. M. Fox, *Applying the Rasch Model: fundamental measurement in the Human Sciences*, 2nd ed. Mahwah, New Jersey: Lawrence Erlbaum, 2007.

Zamalia Mahmud, Mohd Saidfudin, Azrilah Abdul Aziz; Students Performance Evaluation for Learning Outcomes Measurement: SPELOM Model in *Recent Advances in Management, Marketing, Finances: Recent Advances in Electrical Engineering*, Mastorakis, N., Mladenov, V., Zaharim, A., Bulucea, C.A., a series of reference books and textbooks published by WSEAS Press, March 2010, pp.172-178