

Modeling Software Process Maturity on Development Team Size

Olalekan S. Akinola
Solom202@yahoo.co.uk

Babatunde O. Akinkunmi
bo.akinkunmi@mail.ui.edu

Mutiat A. Ogunrinde
bukky2004_ui@yahoo.com

University of Ibadan, Ibadan, Nigeria

ABSTRACT

Software projects are not usually completed and delivered on time and on budget. This failure has been traced to immature development models being used by developers. Five selected reputable software development organizations' models were studied in Lagos State, Nigeria using Capability Maturity Model (CMM) interview questions provided by the Carnegie Mellon University Software Engineering Institute. For each of the Organizations, one past and one current projects' processes were studied. Observation and Interview techniques were used to gather information about the software development process used by each of the organizations. Collected data were analyzed and interpreted with simple percentages and graphs. Results show that only one organization attained (CMM) level two with 84.09% while the rest of the organizations did not. A linear regressive model of maturity levels (ML) of the organizations at level two on the development team size (DTS) shows that ML increases as DTS decreases.

Key Words: CMM, Development Team Size, Software Development, Software Project, Software Organizations.

1. INTRODUCTION

Software Project is a temporary effort to create a unique software product or service which usually includes constraints and risks regarding cost, schedule or performance outcome (PMI, 2008). Software projects continue to become more critical to the organizations that employ them. Still, software failures are frequent. According to Standish Group International (1995), about 16.2% of all software development projects deliver a final product are completed on-time and on-budget. In the larger companies, the news is even worse, only 9% of their projects come in on-time and on-budget. And, even when these projects are completed, many are no more than a mere shadow of their original specification requirements.

In large software development projects, more than 80% are excessively late and over budget (Linberg, 1999 and Glass, 1998). One approach to combat the failure rate has been technical, with organizations introducing new design methods or improving their software development process model, characterize the maturity of their models and effect of the software development team size on the organization's process maturity. Controlling and

improving the process used to develop software is a remedy to these problems (Bandinelli and Fuggetta, 1995).

A Process model is a structured collection of practices that describe the characteristics of effective process; the practices included are those proven by experience to be effective. Software Process maturity can be described as an indication of how close a software process is to being developed and complete, and capable of continuous improvement through quantitative measure and feedback. Capability Maturity Model (CMM) is a process maturity model developed to evaluate software providers. It is broadly referred to a process improvement approach that is based on a process model (Paulk et al., 1993).

The Software Engineering Institute has published a Software Capability Maturity Model (SW-CMM) that can be used to rate an organization's software process maturity. The motivation behind the SW-CMM is that a matured software development process will deliver the product on time, within budget, within requirements, and of high quality. The model is based on five levels; organizations with ad hoc processes start at Level One. To progress

to the next higher level, Level Two, an organization has to demonstrate a repeatable process. To gain a Level Three rating an organization has to demonstrate a defined process. A Level Four organization has a managed process and a Level Five organization has an optimizing software development process (Paulk et al. 1995a). This work is aimed at determining the maturity levels of some selected software development organizations and model it on their development team size.

2. RELATED WORKS

Maturity can be described as a measure to evaluate the capabilities of an organization in regards to certain discipline. it has become popular since the Capability Maturity Model (CMM) was proposed by the Software Engineering Institute at Carnegie Mellon University. The original CMM has a specific focus on the evaluation of software development processes which has been varied and extended in a number of approaches and is now applied for the evaluation of IT Infrastructure Management, Enterprise Architecture Management and Knowledge Management to name a few (Paulk et al., 1993).

CMM was originally designed to help characterize the maturity of software practices, guide a program of continuous process and workforce development, set priorities for immediate actions, and establish a culture of software development process quality excellence. CMM assumes that the quality of a software project outcome depends on the extent of suggested activities actually implemented by an organization (Humphrey, 1989). Today, the CMM has spread far beyond its original application area and is widely used by software organizations in the US and around the globe. The report from the responses of 154 experienced software project developers confirmed that software process management maturity is positively associated with project performance. This result is largely consistent with the many individual cases reported in Lipke and Rosenbaum, (1993). To software managers, this result suggests that CMM, in general, could be a useful guide to improving their current state

of software processes in order to improve project performance.

A model of software development maturity (the capability maturity model (CMM)) describes managerial processes that can be used to attack software development difficulties from the managerial control perspective at five maturity levels. Bradford (1996) examined performance of projects in relation to the activities at these various levels of maturity. Survey of software engineers was carried out that indicated the activities associated with the managerial control of development related positively to project performance measures. However, not each level of maturity demonstrated observable benefits, indicating that greater caution is needed in the planning and implementation of the activities. Results indicate that project performance is most related to the process engineering and organizational support activities of the CMM (Level III) but that product and process quality activities (Level IV) also have a positive relationship with project performance. On the other hand, basic project management process activities (Level II) were not significant at all. Organizations therefore need to realize that benefits may not be reached until they achieve Level III. This requires a great amount of time and money before benefits can be realized. Also, strong relations to benefits seem to tail off after Level III. Not all organizations may wish to pursue the CMM under these conditions. Bradford (1996) concluded that achieving higher levels of software development process maturity requires a long-term commitment to continuous process improvement therefore may take organizations years to achieve the next level of maturity and to realize the benefits.

Springsteen et al. (1992) reported that the Institute for Defense Analysis (IDA) performed a study for the U.S. Department of Defense. They presented quantitative and qualitative data on the SW-CMM and compared the SW-CMM to other process maturity models. IDA's review of cost estimation models found disagreement among their proprietors with respect to the effects of the SW-CMM.

Jones *et al.* (1994) based his assessment on his Checkpoint Model from Software Productivity Research with its associated database of 3000 projects. His prediction was that quality of software would peak at SW-CMM Level 3 and the productivity would peak at Level 3 and decline for Levels 4 and 5. His analysis was an extrapolation based on the very small sample of projects at the time with high SWCMM levels. For example the IBM-Houston Space Shuttle software project was assessed at Level 5, but its productivity was not high, due in large measure to its safety-critical nature.

Larry Putnam (1978) based his assessment of SW-CMM effects on his Productivity Analysis Database System with its database of 1500 projects. He assumed that there was a similarity between the Productivity Index (based on development size, effort, and time) and SW-CMM Levels. Higher levels of Process Maturity resulted in higher levels of productivity. However, the Productivity Index does not account for specific software practices such as those specified in the SW-CMM and it may not be correlated with the SW CMM Levels. For example, many of the projects his database indicated that were in Level 5 may have had a high Productivity Index because of the low complexity of their applications.

Some studies support the concept that technical project performance can be improved if team members engage in higher levels of self-control as opposed to rigid organizational control (Bailyn, 1984 & Weinberg, 1971). This is one example of how the CMM provides “what to do” but allows flexibility on the part of the team members in “how to” accomplish their tasks.

3. Research Methodology

3.1 Data Collection Strategy

The objectives of this research work were achieved by understudying some big software organizations in Nigeria as a case study. Five major software development organizations were selected based on the type of software they produced and size of their development teams. The selected samples of this research work were

located in Lagos State of Nigeria. According to Soriyan and Heeks (2004), Lagos, which is widely regarded as Nigeria’s “economic capital”, accounts for 52 software companies representing about 49 percent of the software companies in Nigeria. Of course, this number would have increased by now because new software houses are evolving in Lagos every year.

Observation and fact finding techniques were used to gather information about the software development process used by each of the organizations. Interview questions were drafted based on the standard provided by the Software Engineering Institute who developed Software capability Maturity Model (SW-CMM) was used to evaluate process maturity level of the organizations according to each level’s Key Process Areas (KPA). Oral interview was conducted for personnel of selected software organization with emphasis on the role played by the target audience of this study which are the system analyst, the solution developer/programmer and project manager in producing reliable software.

Rating and categorizing the software development organizations was done based on the response from the principal managers on their organization’s maturity level activities. The effect that development team size has on maturity of the software development process was also determined based on observation and fact finding using a statistical tools.

3.2 Key Process Areas

In the CMM framework, Key Process Areas (KPAs) are used to describe each maturity level except level 1. A KPA contains several Key Practices (KPs), which are the smallest unit in CMM and need to be carried out to fulfil KPAs. The Independent Variables used for this work are the CMM maturity 18 Key Process Area (KPA) Variables listed below:

KPA1: Requirements Management

KPA2: Software Project Planning

KPA3: Software Project Tracking and Oversight

KPA4: Software Subcontract Management

KPA5: Software Quality Assurance

KPA6: Software Configuration Management
 KPA7: Organization Process Focus
 KPA8: Organization Process Definition
 KPA9: Training Program
 KPA10: Integrated Web application Management
 KPA11: Software Product Engineering
 KPA12: Inter-group Coordination
 KPA13: Peer Reviews
 KPA14: Quantitative Process Management
 KPA15: Software Quality Management
 KPA16: Defect Prevention
 KPA17: Technology Change Management
 KPA18: Process Change Management

Each of these CMM 18 key process areas (KPA) has some tasks/activities associated with it referred to as Key Practices (KPs). For instance, KPA1 has six activities, KPA2 has seven activities, KPA3 has seven activities and so on to be performed by any software development organization, as specified in the original CMM model (Paulk et al., 1993). It is from these activities that software development organizations choose among the activities for her life cycle process.

To form a maturity level, a number of KPAs are combined together for a particular level. For instance, all organizations are assumed to be at level 1, while KPAs 1 to 4 are combined together to form level 2. The number of observed and / or reported KPA activities by the organizations divided by the total tasks to be done in the KPA determines the level of process maturity for a particular organization.

In addition to the Key Process Areas measured, the software development team size comprising of the analysts, designers, programmers and

testers in each of the organizations covered in this work was also elicited from the organizations.

4. RESULTS

The data gathered from software development organizations based on the eighteen (18) key process areas of the standard CMM were presented in Table 1 for the organizations studied. The organizations studied were represented using the following symbols C1 for the first Company, C2, Second Company, C3, Third Company, C4, Fourth Company and C5 for Fifth Company.

The figures were obtained from the ratings of the administered checklist by the managers of the five organizations studied. The maturity level analyses of the organizations were done using the percentage number of positive response to each key process areas in the checklist. The data obtained from the analysis are presented in Table 2. The percentage of yes for each level was calculated using this formula:

$$ML = \frac{X}{Y} \times 100$$

where

ML is maturity level obtained, X is activity performed by the organization and Y is total activity that should be performed in the KPA category.

An organization must attain 80% ML before it will be judged to have attained the maturity level (Paulk et al, 1993).

Table 1: Key Process Areas Distribution of the Organization Studied

Key Process Areas	Total Activity For Each KPA	Yes					No					Not Often					Not Available				
		C1	C2	C3	C4	C5	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
KPA1	6	5	5	6	2	5	1	0	0	4	1	0	0	0	0	0	0	1	0	0	0
KPA2	7	7	4	5	6	7	0	1	0	1	0	0	2	1	0	0	0	0	1	0	0
KPA3	7	5	5	4	5	4	1	1	1	2	3	1	1	2	0	0	0	0	0	0	0
KPA4	8	6	5	0	5	0	2	1	1	3	8	0	1	7	0	0	0	0	0	0	0
KPA5	8	8	4	6	7	5	0	3	1	1	3	0	0	1	0	0	0	1	0	0	0
KPA6	8	6	7	0	8	0	2	0	2	0	8	0	0	6	0	0	0	0	0	0	0
KPA7	7	6	7	1	4	0	0	0	0	3	7	1	0	3	0	0	0	0	3	0	0
KPA8	6	6	3	4	2	5	0	1	0	4	2	0	1	0	0	0	0	1	2	0	0
KPA9	7	4	7	2	3	0	0	0	2	3	7	3	0	0	1	0	0	0	2	0	0
KPA10	6	4	3	1	1	6	0	0	1	5	0	2	3	0	0	0	0	0	3	0	0
KPA11	6	3	2	0	5	5	2	0	0	1	1	1	4	0	0	0	0	0	6	0	0
KPA12	7	5	4	0	5	3	1	1	0	2	4	1	2	0	0	0	0	0	7	0	0
KPA13	6	3	1	0	1	0	3	5	0	4	6	0	0	0	1	0	0	0	6	0	0
KPA14	7	0	4	0	0	0	7	1	0	7	7	0	2	0	0	0	0	0	7	0	0
KPA15	7	6	7	4	3	7	0	0	0	4	0	0	0	0	0	0	0	0	3	0	0
KPA16	7	4	7	0	2	4	2	0	0	4	3	1	0	0	1	0	0	0	7	0	0
KPA17	7	1	7	5	3	4	4	0	0	4	3	2	0	0	0	0	0	0	2	0	0
KPA18	7	4	6	3	1	1	2	1	2	5	6	1	0	0	0	0	0	0	2	0	0

Table 1 shows the data obtained from the response of the studied organisations' key principal managers based on their main focused key process areas. From Table 1, some of the organizations perform all the activities in each key process areas while some of them perform few of the activities.

It can be deduced from Table 1 that C1 focused on major activities of KPA2, KPA6, KPA7,

KPA8 and KPA15, C2 focused on major activities of KPA6, KPA7, KPA9, KPA16, KPA17 and KPA18, C3 focused on some of the activities of KPA1, KPA2, KPA5 and KPA17, C4 focused on some of the activities of KPA2, KPA3, KPA4, KPA5, KPA6 while C5 focused on some of the activities KPA1, KPA2, KPA8, KPA10, KPA11 and KPA15. None of the organizations focused on the activities of the KPA13 and KPA14.

Table 2: Organizations' Aggregate Maturity Levels

Maturity Level	Key Process Areas for each level	Total Activity for each Level	No of Yes for each Level					% Yes for each Level				
			C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
1	Initial Level	-	-	-	-	-	-	-	-	-	-	-
2	KPA1, KPA2, KPA3, KPA4, KPA5, KPA6	44	37	30	15	33	21	84.09	68.18	34.09	75.00	47.72
3	KPA7, KPA8, KPA9, KPA10, KPA11, KPA12, KPA13	45	31	27	8	21	19	68.88	60.00	17.77	46.66	42.22
4	KPA14, KPA15	14	6	11	4	3	7	42.85	78.57	22.57	21.42	50.00
5	KPA16, KPA17, KPA18	21	9	20	8	6	9	42.85	95.23	38.09	28.57	42.85

Table 2 shows the aggregate analysis of the organizations' maturity levels using the data from the key process areas in Table 1. It can be inferred that all the organizations attained level 1. At level 2, total activity of the key process areas is 44; only C1 attained level 2 with 84.09% while the rest of the organizations did not with the following values: C2 has 68.18%, C3 has 34.09%, C4 has 75.00% and C5 has 47.72%. This implies that if the companies C2 and C4 can improve on the remaining key process areas for this level it will attain 80% in order to attain the capability maturity level 2.

C2 attained level five (5) but according to capability maturity model, a company must fulfill all the key process areas in the previous level before moving to the next level (Paulk et. al. 1993). This company has 68.18% in level 2 key process area, 60.00% in level 3 key process area and 78.57% in level 4 key process areas. The company majorly focuses on the managerial processes rather than technical processes in their key process areas in at all levels.

Taking Maturity Level 2 as a bottom line level for the organizations, table 3 shows the development team size and the Maturity Level 2 of the organizations studied.

Table 3 Development Team Size and % Maturity Level of the organisations

ORGANISATIONS	DEVELOPMENT SIZE	% MATURITY
C1	6	84.09
C2	5	68.18
C3	12	34.08
C4	5	77.00
C5	10	47.72

Figure 2 shows the organizations' maturity at level 2 and their development team sizes.

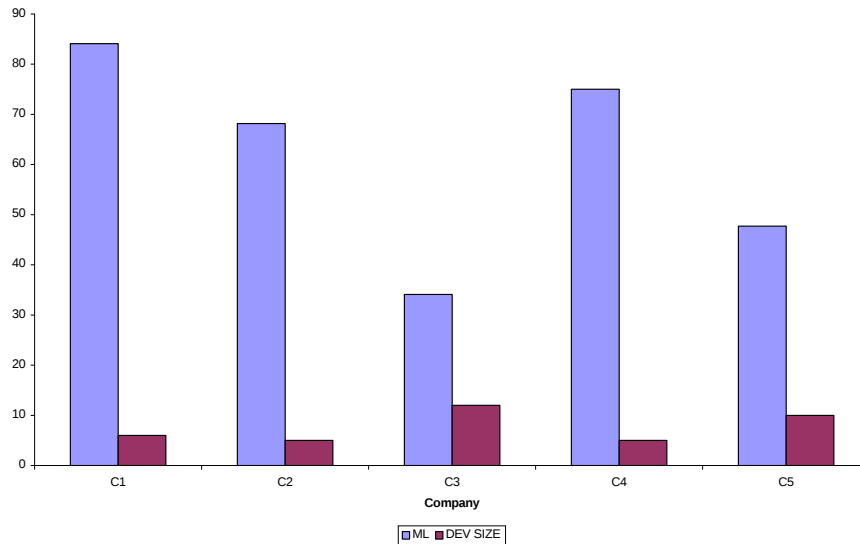


Figure 1. Maturity Levels and Development Team sizes of the organizations at Level 2

4.1. Modeling the Maturity Level on the Development Team Size

A statistical Pearson correlation analysis was initially performed on the results obtained in Table 3. The test gives a -0.926 correlation between the development size and Level 2 Maturity attained by the organizations studied. The correlation was significant at 0.05 significance level (2-tailed).

The result led us to compute a linear regression model for the data. A Linear Regression model analysis estimates the coefficients of the linear equation, involving one or more independent variables that best predict the value of the dependent variable. Maturity level two was used as the dependent variable while the development team size served as the independent variable for the regression analysis. Results obtained from the regression model analysis are presented in Table 4.

Table 4: Summary Regression Model Analysis Results

Unstandardized Development Team size (DTS) coefficient	Constant (C)	R-Square	Significance Level
- 6.008	107.875	0.857 (85.7%)	0.003

The regression model obtained is thus:

$$\text{ML} = 107.875 - 6.008\text{DTS}$$

Where

ML means Maturity Level and
DTS, Development Team Size

Figure 2 shows the chart of Level two maturity levels against the development team size for the organizations studied.

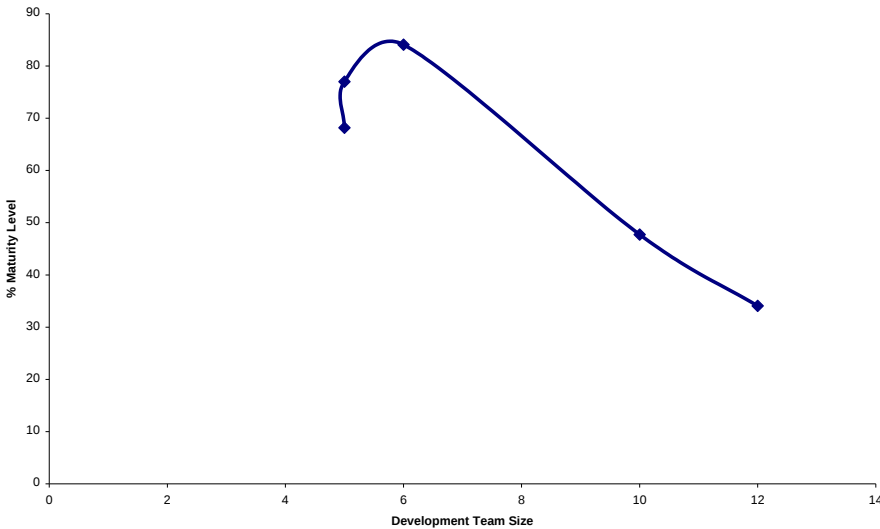


Figure 2: Percentage Maturity Level Two against Development Team size

5. Discussion of Results

Capability Maturity Model (CMM) is a recognized tool for performing *software process* maturity and capability evaluation in software organizations. This work reveals that the organizations studied implemented Capacity Maturity Model, though they were not focusing on the activities of the processes at each Level sequentially as required. They focused more on managerial processes than technical ones. Majority of the Software Development Organizations has their processes at the Capability Maturity Model – Level One. They normally base the success on their current software project on the past successful one. However, balancing the managerial and the technical activities of the process at levels of CMM will make the software development organization improve their software development process while documenting the software development processes will give them more room for constant reviewing and standardization.

This result is in line with the findings of Soriyan and Heeks (2004), which reveals that that Nigerian software industry showed significant use of formal methods but with a strong tendency to rely on in-house-

developed methods rather than industry standards.

The correlation coefficient obtained in this study (-0.926) suggests that there is a very strong negative correlation between the Maturity Level Two and Development Team Size for the organizations studied in this work. Figure two gives a strong indication for this result. As the Maturity level two increases, the development team sizes of the organizations decreases.

The regression model also gives a negative trend prediction for the maturity levels. Goodness of fit statistics (R^2) gives the level at which we can accept our model. For instance, If the R^2 associated with a variable is large, say 80% or more, that means that the variable is a good predictor of the dependent variable. The R^2 obtained in this work (85.7%) led us to accept the predictive regression model. The regression model suggests that large software development team size may not determine a good software process maturity level for a software development organization. What matters most is for the organization to follow a strict software development process procedure in order to achieve quality on its products and attain a good maturity level no matter the size of its development team.

6. Conclusion

The result revealed that the Nigerian software industry is very deficient in so many areas. This includes virtually all the Key Process Areas (KPA) in the SEI Maturity Evaluation Questionnaire model. The appraisal also revealed that the software process of the Nigerian software industry studied is at the maturity level 1, which is the very base level.

Moreover, the effect of the software development team size on process maturity is found to be on a negative trend. The results show that the effect cannot be determined in one industrial survey. We intend to extend this study further to cover more software organizations in order to substantiate the results.

REFERENCES

- Bailyn, L. (1984). *Autonomy in the Industrial R&D Lab*, MIT Press, Cambridge, MA.
- Bandinelli, S. and Fuggetta, A. (1995). Modeling and improving an industrial software process, *IEEE Transaction on Software Engineering*, 21 (5), pp. 440–454.
- Bradford K. Clark (August 1996). Effect of Software Process Maturity on Software Development Effort, Version 1.0, *Qualifying Exam Report*, University of Southern California, Los Angeles.
- Glass, R.L. (1998). *Software Runaways*, Prentice-Hall, New York, NY.
- Humphrey, W.S. (1989). Managing the Software Process, Software Engineering Institute, *The SEI Series in Software Engineering*, Addison-Wesley, New York, NY.
- Jones, R., Bate, R., Garcia, S., Armitage, J., Cusick, K., Kuhn, D., Minnich, I., Pierson, H., Powell, T. and Reichner, A. (1994). *A Systems Engineering Capability Maturity Model, Version 1.0*, CMU/SEI-94-HB-04, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa.
- Linberg, K.R. (1999). Software development perceptions about software project failure: a case, *The Journal of Systems and Software*, 49 (12), pp. 177–192.
- Lipke W.H. and Rosenbaum, S. (1993). Software improvements in an international company, in: *Proceedings of the 15th International Conference on Software Engineering*, pp. 212–220.
- Paulk, C. Mark, Weber, V. Charles, Garcia, Suzanne M., Chrissis, Mary Beth and Bush Marilyn (1993). Key Practices of the Capability Maturity Model SM, Version 1.1
- Paulk, M.C., Curtis, B., Chrissis, M.B. and C.V. Weber (1993). *Capability Maturity Model for Software, Version 1.1*, CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa.
- Paulk, M.C., Weber, C.V., Garcia, S.M., Chrissis, M.B. and Bush, M. (1995a), *Key Practices of the Capability Maturity Model, Version 1.1*, CMU/SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa.
- PMI. (2008). *A Guide to the Project Management Body of Knowledge (PMBOK GUIDE)* (4TH ed.). Newton Sq., PA 19073, USA:Project Management Institute.
- Putnam, L.H. (1978). A General Empirical Solution to the Macro Software Sizing and Estimating Problem, *IEEE Transactions on Software Engineering*, pp.345-361.
- Soriyan Abimbola & Richard Heeks (2004). *A Profile of Nigeria's Software Industry*. Development Informatics Working Paper No 21, Institute for Development Policy and Management, University of Manchester, 2004.
- Springsteen, B., Brykczynski, B., Fife, D., Meeson, R and Norris, J. (1992). *Policy Assessment for the Software Process Maturity Model*, IDA D-1202, Institute for Defense Analyses, Alexandria, Va., 1992.
- The Standish Group International. The Chaos Report;
www.standishgroup.com/sample_research/PDFpages/Chaos1994.pdf). Downloaded in March 2011.
- Weinberg, G. (1971). *The Psychology of Computer Programming*, Van Nostrand Reinhold, NY.