

# AN EMPIRICAL EXPLORATION OF SOFTWARE DEVELOPMENT QUALITY

Harry C. Benham, Montana State University, hbenham@montana.edu

## ABSTRACT

*This study uses data collected on students enrolled in a software development course to explore the role of Creative Self-Efficacy, Playfulness, and Self-Efficacy in the quality of the students' software development process and projects using a theoretical model proposed by Chiravuri and Ambrose [4]. Instruction in the Carnegie Mellon Software Engineering Institute's Personal Software Productivity (PSP) techniques was coupled with traditional software engineering topics providing an opportunity to measure both the student's software development process and the quality of the software developed. Creative Self-Efficacy and Self-Efficacy were found to have a modest impact on the quality of the students' software product.*

**Keywords:** Software Quality, Development, Creative Self-Efficacy, Playfulness, and Self-Efficacy

## INTRODUCTION

The desirability of developing quality software is obvious. Users would experience higher levels of satisfaction with the software and software vendors would be spared the cost and embarrassment of corrective maintenance. Yet as software products become increasingly complex, it becomes increasingly difficult to deliver quality. Chiravuri and Ambrose [4] posit that psycho-social personal factors that foster creativity in software developers should be considered in conjunction with the traditional technical and business skills. Supporting Chiravuri and Ambrose, Robert Glass has identified creativity as the missing link in software design [6].

Basing their argument on social cognitive theory foundations [1, 2, 3], Chiravuri and Ambrose "argue for the inclusion of two key cognitive factors – self-efficacy and creative self-efficacy and one affective factor – playfulness, in an IS personnel's skill set" [4, pg. 200]. The essence of their research model is reproduced in Figure 1.

The purpose of this paper is to empirically explore the model proposed in Figure 1. The remainder of the paper is organized as follows. The Methodology section discusses measurement issues and data

collection procedures. The Results section provides a discussion of the empirical results obtained. The Discussion section looks at a revised model and its empirical results. Finally, the Conclusion summarizes the papers findings and stresses its limitations.

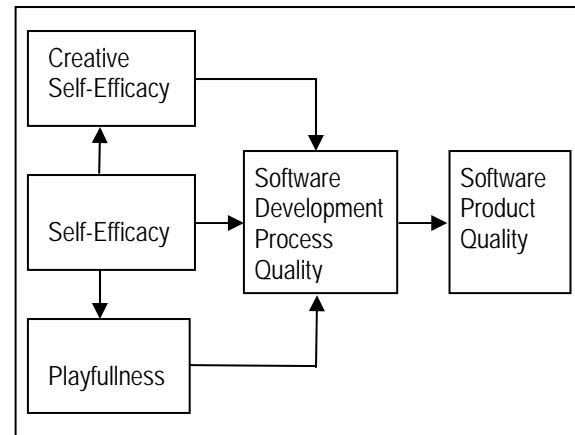


Figure 1: Theoretical Model

## METHODOLOGY

A colleague teaching a software engineering course in a computer science program provided access to his students as subjects. We submitted the necessary "Human Subjects" forms and obtained permission from his institution to proceed with the research project. My colleague's involvement in the project was limited to facilitating the data collection.

### Construct Measures

Survey instruments exist for the measurement of the cognitive and affective factors indicated by Chiravuri and Ambrose [4]. In these instruments, respondents are asked to react to item statements using a 7-point Likert-like scale with responses ranging from "very strongly disagree" to "very strongly agree." Although these are standard instruments, internal consistency of the responses must be checked before using the item responses to construct measures. With well-established multi-item survey instruments exhibiting sufficient internal consistency, the standard procedure is to average the respondents scores on each item (taking into account any reverse scored items).

In the realm of information systems, Compeau and Higgins [5] developed a computer self-efficacy survey instrument and demonstrated its psychometric properties. This measure has been used successfully in IS research for a number of years.

Creative self-efficacy is a less well know instrument in the IS literature. Tierney and Farmer [7,8] developed and validated a multi-item instrument to measure creative self-efficacy. This measure was used without modification.

Playfulness is an even older construct in IS literature than self-efficacy. The multi-item instrument was developed by Webster and Martocchio [9] and has been used in a number of studies since 1992.

Generally accepted “instruments” to measure the quality of the software development process and the quality of the software development product do not exist. However, using students in a software engineering course provided measurement opportunities not available elsewhere. The software engineering course in which the student were enrolled emphasized Carnegie Mellon University, Software Engineering Institute’s Personal Software Productivity [10] techniques. Students were required to log their errors, reflect upon their error frequencies, and take steps to modify their personal development procedures. As a regular part of their coursework, students maintained a “software development journal.” These journals were collected at the end of the semester and scored. The students’ scores on their journals were used as a measure of the quality of their software development process. A student’s score on their journal was based on the instructor’s perception of the completeness of the log entries, the depth of the student’s analysis of log entries, and the degree to which the student modified his/her development process in response to the analysis. It should be emphasized that the scoring of these student journals was totally independent of this research process.

The students in this software engineering course also had a final project that they worked on for a little over half of the semester. The instructor scored the students’ projects using a rubric that included satisfaction of project requirements, robustness of software product, quality of the user interface, appropriateness of algorithms, and, of course, the correctness of the output. Student scores on their software development projects were used to measure software product quality. Again, scoring of these projects was unrelated to this research project.

Although not explicitly entered in the theoretical model, the student participants self-reported their cumulative grade point averages to provide a measure of ability.

### Data Collection

Data were collected in two phases. During the third week of the semester, student subjects were given the URL an interactive web form containing the creative self-efficacy, self-efficacy, and playfulness instruments. As an incentive to participate, students who responded were awarded a few participation points.

The second phase of data collection occurred in the last weeks of the semester as the software development journals and the development project were submitted and scored. Again, the scoring of these student assignments was a normal part of the evaluation process for the course. The faculty scoring these assignments had no involvement in this research project beyond providing access to the students and their scores on the assignments used to measure process and product quality.

## RESULTS

A total of 53 students completed the measurement instruments and submitted both a software development journal and term project for grading thereby establishing the sample size for this exploratory study. Table 1 presents summary statistics for the variables.

**Table 1: Summary Statistics (n=53)**

	ICR	Mean	S Dev	V Share
CSE	.83	4.17	1.13	.91
SE	.89	4.63	1.02	.86
P	.92	5.29	1.16	.88
PQ	NA	42.80	9.21	NA
Q	NA	161.20	2.35	NA

Notes: ICR: Internal Consistency Reliability  
 V Share: Square Root of Shared Variance between constructs and their measures.  
 CSE: Creative Self-Efficacy  
 SE: Self-Efficacy  
 P: Playfulness  
 PQ: Software Development Process Quality  
 Q: Software Product Quality

Table 1 indicates that all internal reliabilities (ICRs) are greater than .70, the commonly accepted standard. The V Share column in Table 1 shows the

square roots of the shared variance between the constructs and their measures. Comparison of these shared variances with the correlation coefficients displayed in Table 2 shows that the shared variances are all greater than the correlations between measures supporting the convergent and discriminant validity of the creative self-efficacy (CSE), self-efficacy (SE), and playfulness (P) measures.

**Table 2: Correlations**

	CSE	SE	P	PQ	Q
CSE	1.00				
SE	0.48	1.00			
P	0.14	0.47	1.00		
PQ	0.11	0.14	-0.21	1.00	
Q	0.49	0.38	0.52	0.08	1.00

The Chiravuri and Ambrose model in Figure 1 was estimated using partial least squares. The results of this estimation are shown in Figure 2. As predicted by Chiravuri and Ambrose, self-efficacy contributes to both creative self-efficacy and playfulness as indicated by the statistically significant beta coefficients estimated. However the results for the more interesting effects, how do self-efficacy, creative self-efficacy, and playfulness influence the quality of the software development process and ultimately the quality of the software product, were not good. None of the estimated betas were statistically significant. Further variances explained were not good: for Software Development Process Quality the  $R^2$  was 0.11 and for Software Product Quality the  $R^2$  was 0.06. Thus Figure 2 would seem to suggest that the model proposed by Chiravuri and Ambrose is not supported.

**DISCUSSION**

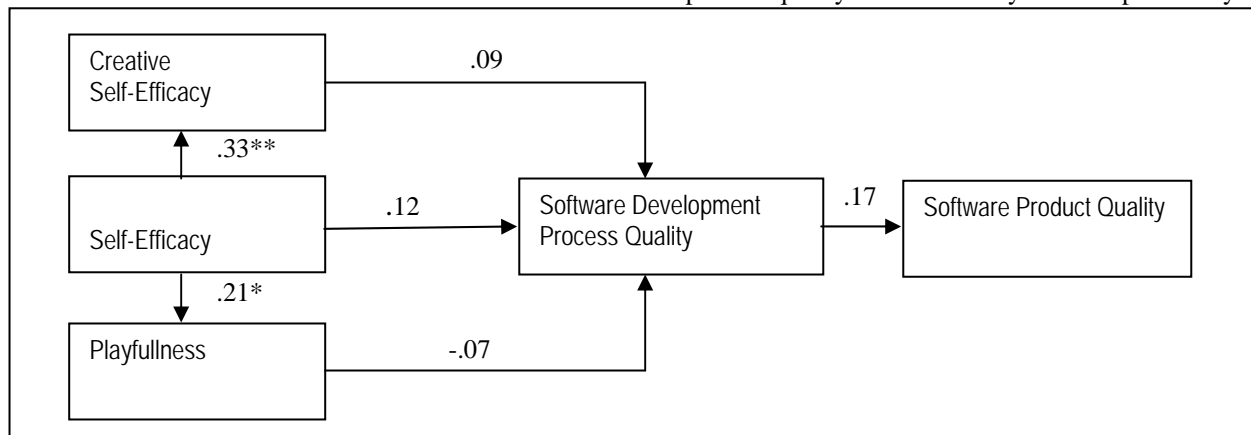
Returning to Table 2's correlations, one can see a

hint of difficulty: the measure of software development process quality does not correlate well with any of the other measures. Indeed, the correlation between the process quality measure and product quality is nearly zero. The construct measure employed here may well be flawed. The construct is designed to reflect the maturity of the developer's processes. In using data derived from students perhaps initial introduction to formal development processes, a number of other phenomenon may have been measured. For example a student who learned a great deal about his or her personal error processes and used that information to correct their personal process would be likely to score high on the software development journal. While students who did not appear to learn about and improve their personnel software processes (perhaps because they had a high quality process initially) might have been scored lower on this particular assignment.

**Revised Model**

The essence of Chiravuri and Ambrose's model is that developers with more creative self-efficacy, self-efficacy, and playfulness should ultimately produce higher quality software. Table 2 indicates that creative self-efficacy, self-efficacy, and playfulness all correlate reasonably well with the software quality measure. Thus in developing a revised model, the intervening development process quality step is deleted from the model.

Missing from the original model specification in Figure 1 is any explicit ability measure. Some individuals are more skilled software developers than others. It would follow that the more skilled developers would produce a higher quality product regardless of their self-efficacy or playfulness levels. Leaving an important determinant of software product quality out of the analysis could potentially



**Figure 2: Model Estimation Results**

\* p < 0.05    \*\* p < 0.01

mask the more subtle influences of self-efficacy and playfulness. Thus in the revised model, the only potential developer ability measure we have, the

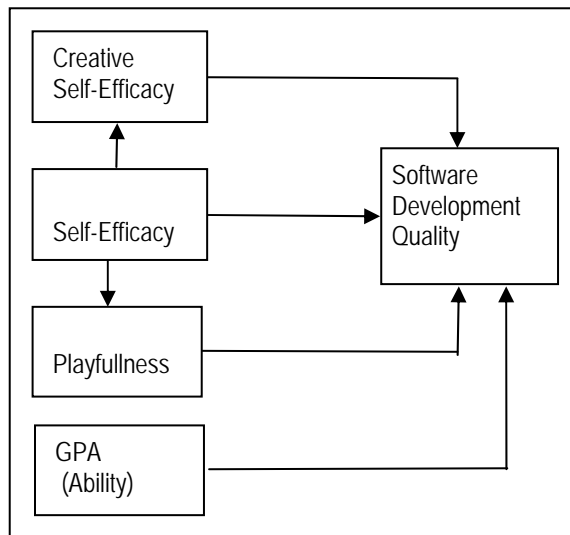


Figure 3: Revised Model

students' grade point average (GPA), is explicitly introduced into the model. Certainly GPA is an imprecise measure of a student's software development ability but it does provide an overall indicator of a student's ability and may serve as a reasonable proxy for software development ability. Figure 3 shows the revised model.

Revised Model Results

The results of estimating the revised model are much more supportive of the ideas expressed by Chiravuri and Ambrose. Figure 4 displays the estimated betas indicating the strength of the relationships between the revised model's constructs. As was the case in the original model, self-efficacy continues to have a significant, positive influence on both creative self-efficacy and playfulness. Unlike the original model's estimates, creative self-efficacy and playfulness are also found to have a positive significant influence on software product quality. Thus the central point made by Chiravuri and Ambrose has some support.

But the strongest measured determinant of software product quality was GPA, our proxy for developer ability. It makes intuitive sense that some measure of the developer's skill would be the most powerful explanatory variable for software product quality. The R<sup>2</sup> for software product quality as 0.47 indicating that GPA plus the cognitive factors of self-efficacy and creative self-efficacy and the affective factor,

playfulness, explained 47% of the variance in measured software product quality.

Therefore there seems to be some evidence supporting the argument put forth by Chiravuri and Ambrose [4] that self-efficacy, creative self-efficacy, and playfulness are factors that enable an IS person to better cope with the demands of their profession. But from a practical, employer position, it would appear that the first priority in selecting a potential software developer would be to select an individual with strong development skills. But when selecting between equally skilled individuals, self-efficacy, creative self-efficacy, and playfulness would be a differentiator.

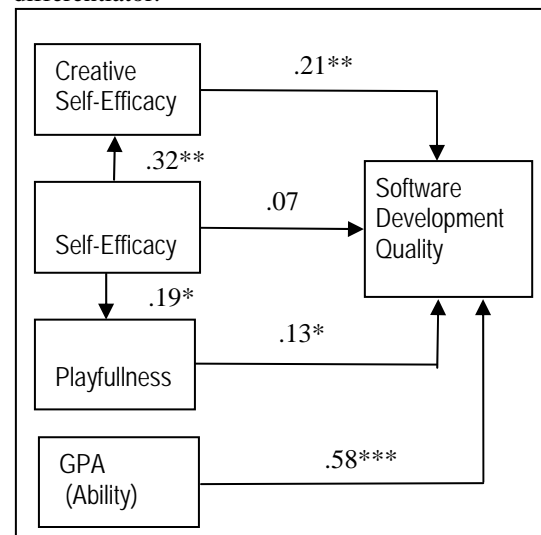


Figure 4: Revised Model Results

\* p < .05, \*\* p < .01, \*\*\* p < .001

CONCLUSION

This study undertook to empirically explore the theoretical model proposed by Chiravuri and Ambrose linking self-efficacy, creative self-efficacy, and playfulness with both the quality of the software development process and the quality of the software development product. Using software engineering students as subjects, measures for all constructs were proposed. Self-efficacy, creative self-efficacy, and playfulness were all measured using their standard instruments. Student scores on a "software development journal" assignment were used as a measure of software development process quality. In retrospect, the validity of this measure is suspect. Finally, software product quality was measured by the student's score on a software development project.

The original model, using the now suspect measure, did not reveal any support for the Chiravuri and Ambrose theory. However, the revised model, dropping the suspect software development quality measure, provided some support for the role of creative self-efficacy and playfulness in influencing the quality of the final software product.

While this study sought to empirically explore the Chiravuri and Ambrose model, it did so in a highly artificial situation. Students in a class/learning environment may differ significantly from practicing professions. The quality of a software product is generally thought to be a multi-dimensional concept. In this case, a software engineering faculty member's judgment as to the appropriate score for that project conveniently provided a single quantitative measure.

It is difficult to generalize from the highly artificial "classroom" environment used in this study. In retrospect, it appears that the proposed software development process quality measure either failed to adequately capture the theoretical construct proposed by Chiravuri and Ambrose, or the quality of the software development process has no impact on product quality. Given that process quality has been verifiably linked to product quality in a number of other domains, the most likely conclusion is that the measure adopted here was flawed.

Additional empirical investigation of Chiravuri and Ambrose's model is needed before a definitive judgment can be made. However, the evidence reported here suggests that those further investigations would be worthwhile. Development of robust measures for both software development process quality and software product quality will be challenging.

## REFERENCES

1. Bandura, A. (1977). Self-Efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological Review*, 84(2): 191-215.
2. Bandura, A (1986). *Self-Efficacy, Social Foundations for Thought and Action: A Social Cognitive Theory*, Englewood Cliffs, NJ: Prentice-Hall.
3. Bandura, A. (1996). *Self-Efficacy: The Exercise of Control*, New York, NY: W.H. Freeman.
4. Chiravuri, A, & Ambrose, P.J. (2007). Exploring the Role of Self-Efficacy, Playfulness, and Creative Self-Efficacy in Information Systems Development. *Issues in Information Systems*, 8(2): 200-206.
5. Compeau, D.R. & Higgins, C.A. (1995). Computer Self-Efficacy: Development of a measure and initial test. *MIS Quarterly*, 19(2): 189-211.
6. Glass, Robert L. (2006). *Software Creativity 2.0*, New York, NY: developer.\* Books.
7. Tierney, P. & Farmer, S.M. (2002). Creative Self-Efficacy: Its Potential antecedents and Relationship to Creative Performance. *Academy of Management Journal*, 45(6): 1137-1148.
8. Tierney, P. & Farmer, S.M. (2004). The Pygmalion Process and Employee Creativity. *Journal of Management*, 30(3): 413-432.
9. Webster, J & Martocchio, J.J. (1992). Microcomputer Playfulness: Development of a Measure with Workplace Implications. *MIS Quarterly*, 19(2): 189-211.
10. \_\_\_\_\_. Software Engineering Institute, Carnegie Mellon University. <http://www.sei.cmu.edu/tsp/psp>.