Recruiting Software Practitioners: The Importance of Self-Efficacy

Dr. Orit Hazzan
Israel Institute of Technology

Dr. Tali Seger
Ruppin College, Israel

Software organizations face challenges when trying to recruit highly competent software practitioners who can successfully participate in and contribute to a cooperative working environment. This article suggests—based on the presented research conducted in a large international communication company—that recruiting practitioners with high levels of self-efficacy can contribute to the organization on both the individual and team levels. This article also describes the research and its findings and discusses specific recommendations based on the research.

Self-efficacy is a characteristic that distinguishes between individuals according to their tendency to perceive difficult events as challenging and to what extent they feel capable of accomplishing almost any task [1]. Based on the research presented in this article, we suggest considering self-efficacy as one selection criterion for software practitioners. Recruiting practitioners with high levels of self-efficacy may contribute to the organization—not only with individuals, but also on the team and organization levels.

Our research was conducted in a large international communication company located in Israel. During Phase I, we explored for two job levels (i.e., senior versus junior) personal characteristics that can predict practitioners' orientation towards cooperative software development environments, such as Agile ones. Self-efficacy was found to be a crucial factor for practitioners in senior positions only [2]. This finding led us to focus (in Phase II) on how different levels of self-efficacy are related to software practitioners' perception of their working environment.

This article describes the research and its findings and discusses specific implications derived from the findings with respect to the recruitment processes of software practitioners. One of our primary contributions, we suggest, is the wider perspective offered on self-efficacy in the context of software organizations. Specifically, our research indicates that self-efficacy is also related to software practitioners' perception of their work environment.

Phase I: Job-Level Comparison of Agile Orientation

The first phase of the research (see [2]) examined how software practitioners' personal characteristics are related to their Agile orientation. Research participants comprised 376 software practitioners employed in two divisions of the company. In terms of job level, the sample included 228 experts and managerial-level practitioners (61 percent) and 148 junior-level practitioners (39 percent).

The research variables examined in Phase I were:

- **Agile Orientation.** Agile orientation was determined by examining practitioners' perceptions of how Agile software development takes place in practice. It was measured using eight out of the 23 items from the original version of Hazzan and Dubinsky's [3] questionnaire. The items address customer expectations, teamwork habits, and perceptions.
- **Self-efficacy.** Software practitioners' level of self-efficacy was measured using the new general self-efficacy scale, as adapted from [4].
- **Psychological Needs.** McClelland's needs theory [5] and needs survey [6] were used. The needs theory attempts to explain and predict attitude and behavior based on an individual's internal needs. Software practitioners' level of psychological needs is examined using 35 items that appeared on the original version of the needs survey, addressing the following five needs: achievement, dominance, affiliation, difference, and attitudes toward change.
- **Perceived Supervisory and Co-workers' Support.** Software practitioners' perceptions of co-workers' and supervisory support was measured using 26 of the 44 items appearing on the original questionnaire developed by Halpin and Croft [7]. This variable includes seven indicators, four of which were adopted due to their correspondence with the topic of this study (software development environments): cooperation, morale, intimacy, and manager supportiveness.

> *"Recruiting practitioners with high levels of self-efficacy may contribute to the organization—not only with individuals, but also on the team and organizational levels."*

All research variables were rated on the Likert scale (where 1 indicates a low perceived level of the measured item and 5 indicates a high perceived level)[1].

Specifically, by using the structural equation modeling analysis method [8, 9], Phase I examined relations between Agile orientation and psychological needs, self-efficacy, and perceived supervisory and co-worker support at junior and managerial-level practitioner job levels. Main findings in this phase included:

- For both job levels, perceived supervisory and co-worker support seem to be a crucial factor that mediates the relations between the individual's different psychological needs and his or her Agile orientation. Specifically, a high level of perceived support is positively related to a high level of orientation. This finding is important since it has direct implications for the design of a software development environment.
- For managerial-level practitioners only, a high level of self-efficacy is positively associated with a high level of perceived support and a high level of Agile orientation.

Further details on Phase I and explanations of the results are presented in [2].

The fact that self-efficacy is frequently researched in organizational behavior studies—and that it appears in our Phase

I research as a variable that dominates perceived support and Agile orientation only among managerial-level practitioners—led us to examine its role in software practitioners' perceptions of their work environment. Such an examination suggests that self-efficacy may also be indicative of practitioners' perception of climate (expressed by their perceived support). In other words, along with acting as an indicator on the individual level, self-efficacy is an indicator that is related to the team and organizational levels. The following sections elaborate on self-efficacy and describe the second phase of the research, where self-efficacy played a central role.

Self-Efficacy

The concept of self-efficacy has received increased attention in organizational research over the past two decades [10]. Bandura defines self-efficacy as the "belief in one's capabilities to organize and execute the courses of action required to produce given attainments" [1]. Thus, the higher one's self-efficacy is, the more likely that he or she engages and persists in task-related behavior. Research showed that self-efficacy positively predicts job attitudes [11], training proficiency [12], and job performance [13], and acts as a buffer to improve the negative effects of work stressors on employees' psychological well-being [14].

Self-efficacy has also gained some attention in the software engineering research, where it is addressed mainly with respect to competence. For example, Paul J. Ambrose proposed that in order to obtain a holistic assessment of competence, it is essential to evaluate developer perceptions and beliefs on what they can achieve—since these beliefs can impact their performance, regardless of the skills the developer possesses [15]. For this purpose, Ambrose developed a measure of developer self-efficacy to assess a critical facet of developer competence. In [16], the self-efficacy model is used in the context of knowledge sharing. The authors concluded that a software manager (or other managers) can easily look at the inputs and outcomes of the model and see where he or she could positively affect tacit knowledge sharing. The authors of [17] investigated factors associated with software developers' intention to reuse software assets. They found that technological-level (infrastructure) and individual-level (reuse-related experience and self-efficacy) factors were major determinants of the developers' behavior.

	LOW Self-Efficacy (≤ 3.75) (N = 55)		HIGH Self-Efficacy (≥ 4.75) (N = 80)				
	Mean	SD	Mean	SD	Wilks' Lambda	Beta[4]	*F*
Agile Culture Perceptions	3.20	.45	3.51	.62	.93	.33	10.34**
Cooperation	3.17	.51	3.47	.62	.94	.14	8.33**
Morale	3.40	.66	3.86	.73	.91	-.07	13.84***
Intimacy	3.24	.54	3.66	.65	.90	.09	15.42***
Manager Supportiveness	3.54	.44	3.88	.69	.93	.22	10.38**
Achievement	3.89	.45	4.48	.39	.67	.37	65.21***
Dominance	3.10	.51	3.71	.55	.76	.45	42.84***
Affiliation	3.95	.34	4.36	.42	.79	-.01	35.26***
Difference	3.93	.42	4.18	.56	.95	.16	7.66**
Attitudes Toward Change	3.80	.61	4.39	.51	.78	.40	37.17**

Wilks' Lambda[5] (1,133) = .52   Eigenvalue[6] = .91   Canonical Correlation[7] = .69
Chi Square (10) = 83.02, p ≤ .001

**p < .01; ***p < .001   SD=Standard Deviation

Table 1: *Summary of Canonical Discriminant Functions and Standardized Canonical Discriminant Function Coefficients*

Phase II: Self-Efficacy in Software Practitioners' Profile

Phase II of the research focused on the role of self-efficacy in practitioners' perceptions of their work environment. Specifically, we wanted to determine whether or not it was possible to distinguish between practitioners with different (high and low) levels of self-efficacy using the variables included in our research. If it is possible, we could argue that software organizations may benefit from the recruitment of practitioners with a high level of self-efficacy, not only on the individual level (based on the general studies on self-efficacy), but also on the team/organization level (based on the results of Phase I of our research).

Discriminant function analysis[2] was used to explore the research hypothesis—that is, the ability to differentiate between software practitioners according to their level of self-efficacy by relying on the other Phase I variables. In general, this analysis method enables the determination of which variables discriminate between two or more groups. The basic idea underlying discriminant function analysis is to establish whether groups differ with respect to the mean value of the variables, and then to use these variables to predict group membership (e.g., of new cases). If the mean values of the variables are significantly different in different groups, then we can say that these variables *discriminate* between the groups.

Specifically, given two groups, the discriminant function analysis selects from the set of the research variables, a subset of variables that significantly distinguish between the two groups (significance level of at least **$p < .05$**). In the case of a single continuous variable, a **t** test is used to determine whether or not a variable discriminates between groups; in the case of nominal variables, the chi-square ($\chi 2$) test is used and for ordinal variables, and a Mann-Whitney test[3] is employed. In such cases, the **F** value for each variable indicates its statistical significance in the discrimination between groups; that is, **F** is a

Table 2: *Classification Results for Self-Efficacy Levels*

	Predicted Level		
	LOW (≤ 3.75)	HIGH (≥ 4.75)	Total
LOW (≤ 3.75)	**47 (85.5%)**	8 (14.5%)	55 (100%)
HIGH (≥ 4.75)	11 (13.7%)	**69 (86.3%)**	80 (100%)

measure of the extent to which a variable makes a unique contribution to the prediction of group membership.

To strengthen the results of this research approach, a sub-sample of the 376 practitioners was examined. It included only those practitioners who could be clearly characterized with either low levels of self-efficacy (lower than 3.75, 55 practitioners) or high levels of self-efficacy (higher than 4.75, 80 practitioners). Figure 1 presents the means of the research variables for these practitioners.

Table 1 (see previous page) presents the **F** values of our research variables, calculated to distinguish between software practitioners with high or low levels of self-efficacy. It also shows that all research variables used in Phase I are included in this set of variables; in other words, they all contribute to the discrimination between the two groups of software practitioners. Figure 1 reflects this claim by illustrating that for each research variable, its mean for participants with low self-efficacy is lower from its means for participants with high self-efficacy. The logical conclusion from this data is that all these variables contribute to the discrimination between the two groups.

Specifically, as Figure 1 shows, practitioners with high self-efficacy tend to have a greater Agile orientation than do low self-efficacy practitioners; in addition, high self-efficacy practitioners are:

- More cooperative.
- Have a greater sense of morale working with their team members.
- Feel that their personal relationships with co-workers are closer.
- Get better managerial support.
- Report higher needs in achievement, dominance, affiliation, and difference.
- Have better attitudes towards change.

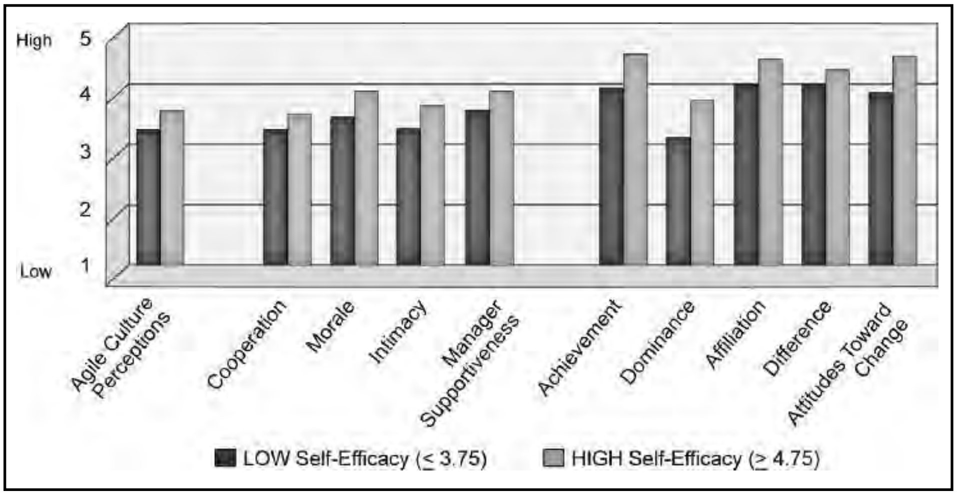
A close examination of each of these variables clearly justifies their relative value with respect to practitioners with low and high levels of self-efficacy.

Table 2 (see previous page) shows that the canonical discriminant function, constructed by the discriminant function analysis by using the other research variables, classifies correctly high-percentage of the practitioners 85.9 percent[8] according to their level of self-efficacy. Specifically, the function classifies 85.5 percent of the low self-efficacy participants and 86.3 percent of the practitioners with high level of self-efficacy. Visually, the numbers in **bold** print indicate correct classification of 47 out of the 55 practitioners with low self-efficacy and of 69 out of the 80 practitioners with a high level of self-efficacy.

> *"... self-efficacy can also be used as an indicator of whether or not a practitioner perceives his or her environment as being supportive—a perception that is positively correlated with development environments that encourage teamwork and cooperation."*

Based on the integration of the presented results, we suggest that since the research variables both help in distinguishing between practitioners with high and low levels of self-efficacy and are relevant for software development environments, it is appropriate to use self-efficacy as an indicator for an individual's perception of his or her development environment.

Figure 1: *Means of Research Variables for Practitioners with High and Low Levels of Self-Efficacy*



Recommendations

We recommend using self-efficacy beyond its current role as an indicator of practitioners' performance. Specifically, our research indicates that self-efficacy can also be used as an indicator of whether or not a practitioner perceives his or her environment as being supportive—a perception that is positively correlated with development environments that encourage teamwork and cooperation, such as Agile software development. Based on this finding we can suggest, for example, that recruiting practitioners with high levels of self-efficacy may contribute to the organization on the individual level by recruiting practitioners with high achievements and strong beliefs in their performance, but also on the team and organizational level. Accordingly, we propose that recruiting software practitioners with high levels of self-efficacy—an easy-to-measure individual characteristic utilizing available questionnaires—may foster the formation of a supportive work climate. Needless to say that when making such decisions, each company should check very carefully its full set of considerations and characteristics and decide on the importance attributed to each factor in its recruitment processes.◆

References

1. Bandura, Albert. *Self Efficacy: The Exercise of Control.* New York: W.H. Freeman, 1997.
2. Seger, Tali, Orit Hazzan, and Ronen Bar-Nahor. *Agile Orientation and Psychological Needs, Self-Efficacy, and Perceived Support: A Two Job-Level Comparison.* Proc. of the Agile 2008 Conference. Toronto, Canada. 4-8 Aug. 2008: 3-14.
3. Hazzan, Orit, and Yael Dubinsky. *Clashes between culture and software development methods: The case of the Israeli hi-tech industry and Extreme Programming.* Proc. of the Agile 2005 Conference. Denver. 24-29 July 2005: 59-69.
4. Chen, Gilad, Stanley M. Gully, and Dov Eden. "General self-efficacy and self-esteem: toward theoretical and empirical distinction between correlated self-evaluations." *Journal of Organizational Behavior* 25.3 (2004): 375-395.
5. McClelland, David C. 1961. *The*

Achieving Society. Princeton, N.J.: D. Van Nostrand, 1961.

6. McClelland, David C. *Human Motivation*. New York: Cambridge University Press, 1987.
7. Halpin, Andrew W., and Don B. Croft. 1963. *The Organizational Climate of Schools*. Chicago: Midwest Administration Center of the University of Chicago, 1963.
8. Kline, Rex B. *Principles and Practice of Structural Equation Modeling*. New York: The Guilford Press, 1998.
9. Mueller, Ralph O. *Basic Principles of Structural Equation Modeling*. New York: Springer-Verlag, 1996.
10. Chen, Gilad, and Paul D. Bliese. "The Role of Different Levels of Leadership in Predicting Self and Collective Efficacy: Evidence for Discontinuity." *Journal of Applied Psychology* 87.3 (2002): 549-556.
11. Saks, Alan M. "Longitudinal field investigation of the moderating and mediating effects of self-efficacy on the relationship between training and newcomer adjustment." *Journal of Applied Psychology* 80.2 (1995): 211-225.
12. Martocchio, Joseph J., and Timothy A. Judge. "Relationship between conscientiousness and learning in employee training: mediating influences of self-deception and self-efficacy." *Journal of Applied Psychology* 82.5 (1997): 764-73.
13. Stajkovic, Alexander D., and Fred Luthans. "Self-efficacy and work-related performance: A meta-analysis." *Psychological Bulletin* 124.2 (1998): 240-261.
14. Jex, Steve M., and Paul D. Bliese. "Efficacy beliefs as a moderator of the impact of work-related stressors: A multi-level study." *Journal of Applied Psychology* 84 (1999): 349-361.
15. Ambrose, Paul J. "Metacognition and software developer competency: construct development and empirical validation." *Issues in Information Systems* 6.2 (2003): 273-279.
16. Endres, Megan L., et al. "Tacit knowledge sharing, self-efficacy theory, and application to the Open Source community." *Journal of Knowledge Management* 11.3 (2007): 92-103.
17. Mellarkod, Vidhya, et al. "A multi-level analysis of factors affecting software developers' intention to reuse software assets: An empirical investigation." *Information & Management* 44.7 (2007): 613-625.

Notes

1. In addition, two background variables—years of experience and age—were collected; they were not, however, included in this analysis.
2. Our description of the discriminant function analysis is partially based on <www.statsoft.com/textbook/discriminant-function-analysis>.
3. For more on Mann–Whitney and chi-square testing, see <http://en.wikipedia.org/wiki/Mann-whitney> and <http://en.wikipedia.org/wiki/Chi-square_test>, respectively.
4. The betas are the coefficients of the unstandardized discriminant function. Each subject's discriminant score is computed by entering his or her variable values (raw data) for each of the variables in the equation. The betas are used to construct the actual prediction equation, which can then be used to classify new cases. In our case, see Table 2.
5. Wilks' Lambda is the proportion of total variance in the discriminant scores not explained by differences among groups. A lambda of 1.00 occurs when observed group means are equal. A small lambda indicates that group means appear to differ. Here, the Lambda of 0.52 has a significant value (Sig. < 0.001); thus, the group means appear to differ.
6. Eigenvalue: A large eigenvalue, as present in our case, indicates a high proportion of explained variance in the predicted variable (in our case, level of self-efficacy).
7. The canonical correlation is the multiple correlation between the discriminant scores and the levels of the dependent variable. A high correlation indicates a function that discriminates well. The present correlation of 0.69 is not very high (1.00 is perfect).
8. Determined by adding the number of employees at high and low levels (47+69) divided by total practitioners (55+80).

Software Defense Application

There has been the long-held idea that individual success increases from self-efficacy: one's belief in their ability to complete a project. Success in any project and in any organization—including in the DoD—can be buoyed by high levels of self-efficacy. Though one can argue that self-efficacy by itself is sufficient for the creation of a successful team, our findings emphasize that the way in which the environment is perceived by individuals should be considered as well. Through understanding these different perceptions, and recruiting practitioners who have high levels of self-efficacy, individuals, teams, organizations, and their products, all improve.

About the Authors

Orit Hazzan, Ph.D., is an associate professor at the Department of Education in Technology and Science of the Technion – Israel Institute of Technology. In 2004, she co-authored "Human Aspects of Software Engineering" with the late Jim Tomayko. Her second book (co-authored with Yael Dubinsky), "Agile Software Engineering," was published in 2008. Hazzan is a consultant for several software projects in the Israeli software industry.

Technion
Israel Institute of Technology
Department of Education in
Technology and Science
Haifa 32000
Israel
E-mail: oritha@tx.technion.ac.il

Tali Seger, Ph.D., is a senior lecturer in the Ruppin Academic Center and the head of Human Resource and Organizational Development in their MBA program. From 1998-2004, Seger was an organizational consultant and worked as a human resources manager at several Israeli companies. She received her doctorate in business administration from Haifa University in 2006.

Ruppin Academic Center
Department of Business
Administration
Emek Hefer 40250
Israel
E-mail: talis@ruppin.ac.il