Regression Analysis on Experience Based Factory Model for Software Development Process

M. Hanafiah, R. Abdullah, M. A. A. Murad and J. Din
Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.
mastura.hanafiah@daimler.com

Abstract**—Software organizations are still struggling to reuse the best of their knowledge and experiences in future projects. Often, when there are changes on employee re-assignment, a lot of time and effort are spent for knowledge transfer activities. This however may not assure that all knowledge and experiences are well transferred and shared; some could be missing or misplaced. In this study, a model has been proposed for managing knowledge and experiences based on experience factory approach to provide a more efficient and effective experience management for software development community. Experience Factory is an infrastructure that aims for reuse of products, processes and experiences gained during a system life cycle. A set of components have been identified as the predictors of the model which eventually forms the two main organizations: project organization and experience factory organization. This study further has gone through a correlational survey research to verify the relationship between the identified predictors towards the experience factory goals. Reliability analysis has been conducted to validate the measures, while correlation and regression analyses have been carried out to examine the relationship between the constructs and the goals. Results reveal that reliability of the model is high and construct validity is satisfactory. Experience factory organization is found having more positively significant towards experience goals as compared to project organization; however, there is no significant impact towards the model due to inexistence of causal relation.**

Index Terms**—Experience Factory; Regression Analysis; Correlational research; Software Development Process.**

I. INTRODUCTION

Today, software development (SD) has advanced from traditional development methodology towards more lightweight trends like agile development. While some organizations remain with conventional methods, many are moving towards agile development in order to fulfill the high and evolving user needs. Currently, software development work has becoming even more global in a way that development work is distributed among teams in different geographical boundaries. In global software development (GSD), organizations face more challenges due to the difference in culture, political, distance, education and the like [1]. To minimize the gap and to ensure that geographically distributed individuals and teams are collaborating, it is thus indispensable to have such an infrastructure to allow the exchange of knowledge and information [2].

There are many solutions that have been proposed and implemented thus far in knowledge management for SD. An earlier analysis by Hanafiah et. al [3] on the proposed KM solution for SD domain reveals that approaches such as ontology, semantic, pattern based, agent based, taxonomy and experience factory have been proposed and implemented previously. Experience based solutions have been realized by previous researchers such as by Sharma et al. [4], Ivarsonn and Gorschek [5], Ardimento et al. [6], and Maturro and Silva [7]. These solutions, however, focus on certain particular SD topics. For example, Sharma et al. focuses more on improvement models such as CMM and TQM; Ivarsonn and Gorschek focus more on best practices or lessons learned, while Maturro and Silva focus mainly on postmortem reviews. Ardimento et al., on the other hand, emphasizes on e-learning tool for identified knowledge experience packages: tool, evidence, competence and projects. Although these projects stress upon preserving experiences in development work, it seems that not many SD processes are involved; additionally, collaboration capabilities are also lacking.

Experience factory (EF) is an infrastructure introduced especially for SD domain with the aims on quality improvement by reusing of products, processes and experiences originating from system lifecycle [8]. With the promising outcome of EF as implemented by Koennecker et al. [9], Althoff et al. [10] and Chen et al. [11], we apply the EF approach to conceptualize a model for SD process, called EBF-SD (Expericence Based Factory for Software Development Process). Further verification on the model has been conducted via correlational survey research and regression analysis with the data collected from the SD expert community.

The paper is structured as follows: Section II discusses the theoretical frameworks involved in the model development; Section III discusses about the research methodology; Section IV focuses on conceptual model and hypothesis development; Section V discusses about the results and analysis; and Section VI is the conclusion of the study and some insight of future works.

II. LITERATURE REVIEW

This literature review covers the important theoretical frameworks relevant towards the model development.

A. Experience Factory

Experience Factory (EF) was introduced by Basili et al. with the goal to leveraging experience from previous projects to improve performance in terms of cost, quality and schedule [12]. Managing knowledge and experiences requires more commitment and expectations that cannot be left to individual projects; therefore EF separates the responsibilities into two distinct organizations: Project Organization and Experience

Factory, with the purpose to help project teams preparing the resources to make experience available for reuse. Figure 1 depicts the EF infrastructure with separated logical and physical organizations. The project or development organization provides the experience factory with product development data e.g., environment characteristics, data, and models currently used. The experience factory analyzes, synthesizes and packages the experiences it gains and provides repository services back to the project organization.

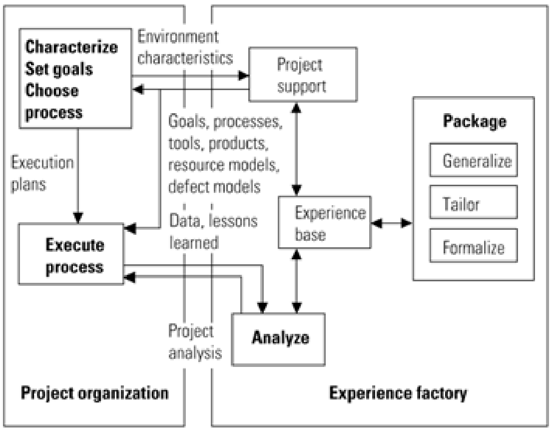

Figure 1: The Experience Factory [12]

It is the responsibility of EF organization to ensure that the activities of the experience factory and project organization are well integrated. This is shown in Figure 1 whereby the experience factory extracts the experiences from project organization and provides them back the packaged experiences that are usable for future reuse.

An organization may benefit from EF by establishing software improvement process, producing a repository for everyday practices, developing organizational internal support for substantial cost and quality performance benefits, providing a mechanism for incorporating new technologies, and supporting reuse in software development process [12].

B. Knowledge Management

As in other domains, the concepts of data, information and knowledge in SD is theorized as a hierarchy of data-information-knowledge-wisdom [13]. Data is more valuable when they are processed to become information, information is used as answers to questions, and knowledge refines information by making possible transformation that can be used as instruction. By having wisdom means an individual is having the ability to judge and evaluate the consequences of any act. Knowledge can be learned by anybody, while experience is acquired by the people that involves in a particular activity. Knowledge and experience management go through similar cycle: (i) acquire or collect, (ii) manage or engineer and restore, (iii) make available or disseminate, and (iv) find/identify or activate [6].

Knowledge management (KM) is the process of acquiring new knowledge; transform it from tacit into explicit or vice versa; storing, disseminating, evaluating it; and applying knowledge in new situation [14]. Tacit and explicit knowledge can be transformed from one form to another as described by Nonaka and Takeuchi [15]. It goes through 4 stages of conversion: from tacit to tacit knowledge (socialization); from tacit to explicit knowledge (externalization); from explicit to explicit knowledge (combination); and from explicit to tacit knowledge (internalization) (see Figure 2). The four ways of knowledge conversion allows knowledge transfer from one person to another, from raw form into a more meaningful information which sometimes combine with other knowledge to make it even more meaningful.

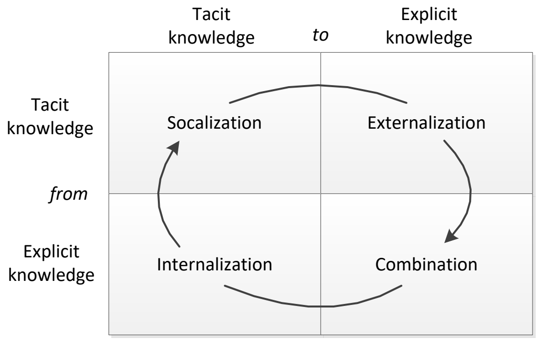

Figure 2: Four Models of Knowledge Creation [15]

Generally, knowledge transfer lifecycle goes through the following [16]: 1) identify and evaluate the knowledge; 2) validate and document the knowledge; 3) publish and share the knowledge; 4) transfer and apply the knowledge; 5) learn and capture the knowledge. Piktialis and Greenes further elaborate that transferring knowledge to different generations may require different approaches [16]. For example, for Generation X (Gen-X) (born 1965-1979), they adapt easily towards both formal and informal learning, and they learn based on experiences and mistakes. Generation Y (Gen-Y)/ Millennials (born 1980-1995) are more technically savvy, value diversity, have more global perspective and expect lots of feedback and communication; and thus, the creation of KM system should provide such capabilities to continuously draw the interest from the community.

C. Software Development Process

Software development involves that many activities from the inception until the software is in operation. Some organizations are still using traditional methodologies like waterfall model or iterative and incremental approach while some has moved towards agile or lean approach. In either case, understanding the phases and approaches of the lifecycle model is crucial for any software practitioners so that software is developed in a defined and controllable manner.

Main phases in software engineering typically involve the following [37]:

(i) Software Requirement (RE): the elicitation, analysis, specification, and validation of software requirements as well as the management of requirements during the whole life cycle of the software product.
(ii) Software Design (DE): the analysis of requirements that describe the software internal structures resulting into software architecture with subcomponents and interfaces defined.
(iii) Software Construction (CO): the detailed creation of working software through a combination of coding, verification, unit testing, integration testing, and debugging.
(iv) Software Testing (TE): the dynamic verification of a program by executing a predefined set of test cases against the expected behaviors.

(v) Software Maintenance (MA): continuous tasks and activities required to support the software.

It is important to have the common understanding among the stakeholders in order to develop the product right in a right way and to avoid any kind of misunderstanding. In many cases, best practices in SD process are recognized as important to be shared across organization so that knowledge and experience are reused effectively [5, 17].

D. Community of Practice

Communities of Practice (CoPs) are the one of most effective organizational forms for sharing and transferring knowledge between people with a common profession, practice area or domain" [16]. CoP may consist of many different roles and can evolve over time [18]. These roles come from different working group; however, they have one common goal to achieve together to make a project successful. These professionals have to work together, to collaborate and cooperate among each other to achieve own organizational goal; and eventually they develop a unique perspective, a common sense of identity with a common body of knowledge, approaches and practices [19].

CoPs that are empowered with ICT such as internet and communication capabilities allow more collaboration and more effective knowledge sharing. This would also allow people and information can be accessed anytime, allow more effective problem solving and decision making with the different skillset and expertise, cost effective and people are willing to interact and are less inhibited [20]. The growth of internet and Web 2.0 has led to various powerful technologies as well that can develop and enhance CoPs such as Wiki, Social Networking, Forums, Blogs, Learning Management and Content Management [21].

E. Technology and Infrastructure

KM implementation requires technologies to facilitate the knowledge creation, storage, sharing and access. EF activities can be supported by Web 2.0 features by using the available technologies such as Wikis, blogs, and social networking to allow knowledge capture and transfer, collaboration and workplace learning. Features that are available in the technologies such as syndication, search, visualization, personalization, recommendation, rating and commenting are able to support EF activities of retrieving, analyzing, formalizing, generalizing, adapting, and discarding experiences [22].

Automation in EF can be facilitated by engaging software agents. Multi Agent Systems (MAS) can be employed to solve more complex problems by employing a collection of agents that are collaborated in a given domain [23]. These agents usually have a small knowledge base with a specified intelligence that collaborates with other agents to ensure the consistent and coherent knowledge based, as well as facilitate the communication and coordination among the agents.

Establishing EF infrastructure would require substantial amount of investment. Cloud computing offers infrastructure and storage services to facilitate storing of knowledge, by offering on-demand services with high availability, reliability, elasticity and scalability; therefore, users can access via network services anytime and anywhere with pay-per-usage basis [1]. This suggests zero upfront infrastructure investment, ready infrastructure, more efficient resource usage, usage-based costing and a real potential for lessening the processing [24]. Cloud deployments also increase the speed and efficacy in which data can be accessed and analyzed as it grows, delivering greater access to broader audience [25].

III. METHODOLOGY

Figure 3 depicts the methodology of the whole research. It starts from literature review analysis, the formulation of conceptual model, the development of the prototype and the evaluation of the prototype. This paper focuses on the correlation survey research as part of the conceptual model formulation (marked as grey colored in Figure 3).

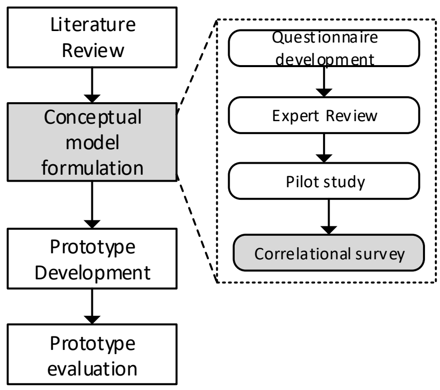

Figure 3: Research Methodology

The formulation of the conceptual model goes through a series of steps including the development of the questionnaire, the expert review analysis, the pilot study of the questionnaire followed by the actual correlational survey. Several hypotheses are derived and they are further tested with Pearson correlation.

The participants of the survey are software practitioners in software organizations which consist of software engineers, project managers, business analysts and consultants. The instrument used in this correlational survey is self-developed questionnaire items. 76 items are constructed with 4 point Likert scale (1- strongly disagree, 2-disagree, 3-agree, 4-strongly agree): SD (7 items), CoP (8 items), KM (23 items), Tech_Infra (20) and EF_Goals (18 items). 80 questionnaires have been distributed via convenience sampling to 4 software companies and in return we received valid 54 responses (67.5% response rate).

Before regression and correlational analysis are carried out, reliability and validity analysis is conducted to ensure overall consistency of a measure. High reliability analysis indicates similar results can be obtained if the study is conducted in consistent settings. The reliability of the responses is analyzed using Rasch analysis [26], a psychometric model for analyzing categorical data, as a function of the trade-off between person ability versus item difficulty. Analysis with Rasch model gives preliminary insights on the model development whether the model being built constitutes the right components, and whether the items' and persons' measures fit the model. Boone et al. asserts that researchers in all fields can improve the reliability assessment of their instruments by using Rasch techniques to evaluate reliability [27].

Validity analysis in construction testing is a continuous activity that should be carried out for each and every use. Validity is defined as "the degree to which evidence and theory support the interpretation of test scores for proposed uses of test" [28]. Construct validity threats may come from two scenarios: construct underrepresentation and construct contamination [29]. Construct under-representation means imperfectness of the test in which the constructs might miss some important measures even though the researchers do include the features according to their definition, while construct contamination is the existence of unrelated sub-dimensions that are irrelevant to the focal constructs and they produce reliable variances in test scores but irrelevant to the constructs. These two threats could result invalidly low scores for difficult items and invalidly high scores for easy items. It is therefore important to ensure the fitness of the data obtained before it is used for further analysis.

To test the hypotheses, Pearson correlation is used. Correlational research is a quantitative method to examine two or more variables whether there is any relationship between them. Correlation coefficients measure the strength of association between two variables. The Pearson product moment correlation coefficient value is calculated using the following equation [30]:

$$r = \frac{\sum xy}{\sqrt{\sum x^2 \sum y^2}} \quad (1)$$

where $x = x_i - x$, x_i is the x value for observation i, x is the mean x value, and $y = y_i - y$, y_i is the y value for observation i, and y is the mean y value. When Pearson's r is close to 1, there is a strong relationship between the two variables, and when Pearson's r is close to 0, there is a weak relationship between the two variables.

Additionally, we also look into the relationship between the predicators (independent variables) and the response (dependent variable) by analyzing it with multiple linear regression (MLR). MLR is a statistical technique that aims to predict a variable of interest from several other variables [31]. MLR is used to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data. MLR is denoted by:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_{k-1} x_{i,k-1} + \epsilon_i \quad (2)$$

To determine if the model is useful, we would like to perform a hypothesis test as follows:

- H_0: $\beta_1 = \beta_2 \ldots \beta_k = 0$
- H_A: At least one $\beta_i \neq 0$ (for $i = 1,..k$)

However, MLR and hypothesis testing with multiple variables can be complicated and time consuming. Another alternative is to use joint test, which is based on a statistic that has an F distribution when H_0 is true, by using the overall *F*-test and *P*-value reported in the analysis of variance (ANOVA) table. F distribution is denoted by:

$$F = \frac{SSR/k}{SSE/(n-k-1)} \quad (3)$$

where SSR is the regression sum of squares (SST – SSE) and SSE is the residual sum of errors.

The details of the result are discussed in Section V.

IV. Conceptual Model and Hypotheses Development

The development of the model is mainly based on the literature review discussed in the previous section. The model is composed a set of relevant components that may contribute to its efficiency and effectiveness. Figure 4 below depicts the proposed correlational model. Based on the success model of Experience Factory, the two organizations are incorporated: Project Organization (Proj_Org) and Experience Factory Organization (EF_Org).

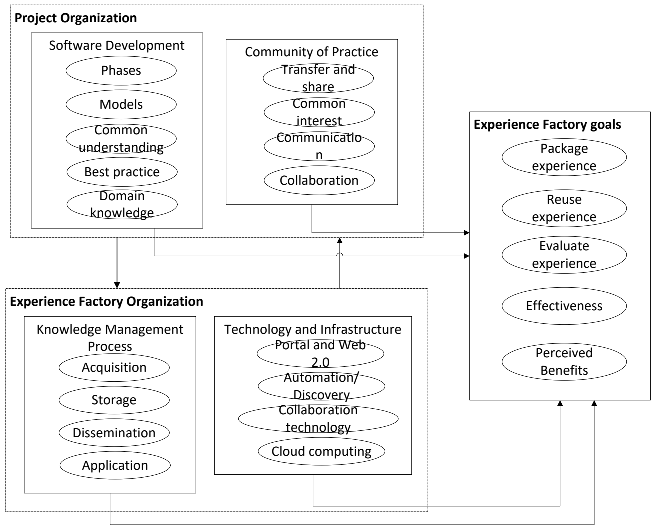

Figure 4: EBF-SD Correlational Model

In the context of software development, Proj_Org consists of the following elements: Software Development Process (SD) and Community of Practice (CoP), while EF_Org contains Knowledge Management Process (KM) and the appropriate Technology and Infrastructure (Tech_Infra). Experience Factory Goals (EF_Goals) are the expectations that we want to achieve in this model.

The following points out the key elements for each of the identified constructs.

Software Development Process (SD):

- development phases
- lifecycle models
- domain knowledge
- best practices
- common understanding

Community of Practice (CoP):

- knowledge transfer and sharing
- common interests among CoP members
- communication and collaboration

Knowledge Management Process (KM):

- knowledge acquisition, capture or creation
- knowledge storage and mapping
- knowledge dissemination
- knowledge application and reuse

Technology and Infrastructure (Tech_Infra):

- use of portal and Web 2.0
- knowledge automation and discovery
- collaboration technology
- cloud computing

Experience Factory goals (EF_Goals):

- package experience
- reuse experience
- evaluate experience
- effectiveness and efficiency
- perceived benefits of the EF approach

With the above-mentioned construct and items, we posit the following hypotheses.

H1: Software development process is positively related to experience factory goals.

H2: Community of practice is positively related to experience factory goals.

H3: Knowledge management is positively related to experience factory goals.

H4: Technology and infrastructure is positively related to experience factory goals.

H5: Project organization is positively related with experience factory organization.

The hypotheses' testing is discussed in the following sections.

V. Results and Discussion

A. Reliability and validity analysis

Rasch reliability analysis shows the measures of person ability vis-à-vis item difficulty in one scale. It also able to distinguish misfit responses in order to achieve the data be fitted in the model before the data can be used for further study. Figure 5 shows that person reliability is excellent (0.95) and Cronbach's alpha value is excellent (0.96).

```
     SUMMARY OF 54 MEASURED PERSONS
-------------------------------------------------------------------------------
|          RAW                          MODEL         INFIT         OUTFIT    |
|          SCORE      COUNT    MEASURE  ERROR      MNSQ   ZSTD   MNSQ   ZSTD  |
|-----------------------------------------------------------------------------|
| MEAN     254.0       76.0        .89    .30       .95    -.3    .94    -.4  |
| S.D.      23.5         .1       1.76    .06       .54    3.0    .57    2.7  |
| MAX.     300.0       76.0       4.82    .52      2.66    8.2   2.74    6.7  |
| MIN.     216.0       75.0      -2.36    .24       .10   -5.6    .07   -5.1  |
|-----------------------------------------------------------------------------|
| REAL RMSE    .32  ADJ.SD     1.73  SEPARATION  5.34  PERSON RELIABILITY  .97 |
|MODEL RMSE    .30  ADJ.SD     1.73  SEPARATION  5.71  PERSON RELIABILITY  .97 |
| S.E. OF PERSON MEAN = .24                                                   |
-------------------------------------------------------------------------------
       VALID RESPONSES:  99.9%
PERSON RAW SCORE-TO-MEASURE CORRELATION = .99 (approximate due to missing data)
CRONBACH ALPHA (KR-20) PERSON RAW SCORE RELIABILITY = .98 (approximate due to
missing data)

     SUMMARY OF 76 MEASURED ITEMS
-------------------------------------------------------------------------------
|          RAW                          MODEL         INFIT         OUTFIT    |
|          SCORE      COUNT    MEASURE  ERROR      MNSQ   ZSTD   MNSQ   ZSTD  |
|-----------------------------------------------------------------------------|
| MEAN     180.5       54.0        .00    .34       .99    -.1    .94    -.2  |
| S.D.       6.2         .1       1.12    .03       .22    1.0    .34    1.1  |
| MAX.     197.0       54.0       2.64    .40      1.86    3.1   2.15    3.7  |
| MIN.     165.0       53.0      -1.42    .25       .60   -2.5    .44   -2.1  |
|-----------------------------------------------------------------------------|
| REAL RMSE    .35  ADJ.SD     1.06  SEPARATION  3.00  ITEM   RELIABILITY  .90 |
|MODEL RMSE    .34  ADJ.SD     1.07  SEPARATION  3.12  ITEM   RELIABILITY  .91 |
| S.E. OF ITEM MEAN = .13                                                     |
-------------------------------------------------------------------------------
```

Figure 5: Summary Statistics

Spread of responses is 0.89 and it is much closed to 1. Outfit and infit mean square are 0.94 and 0.95 respectively, and they are very close to the expected value of 1. Z-standardized value is -0.3, and this is also close to value 0 and within the normality range: $-2 < Z < +2$.

Item reliability is also excellent with reliability value of 0.91 and the mean square values are also very close to 1. Items' Z-standardized value is -.1, it is expected to be at norm and within the normality range. This indicates overall fit to the Rasch model.

Person-item distribution map (PIDM) (Figure 6) shows that items at the higher scale are harder to agree with while items at the lower scale are easier to agree with. There are some persons that are item free at the higher and lower scale; correspondingly, they are easier to agree with all items and harder to agree with all items. Person mean (+0.89) is higher than item mean (0.00) indicating the items' difficulty is within the persons' ability.

In Rasch analysis, negative correlation gives the perception there could be something wrong with the item or person.

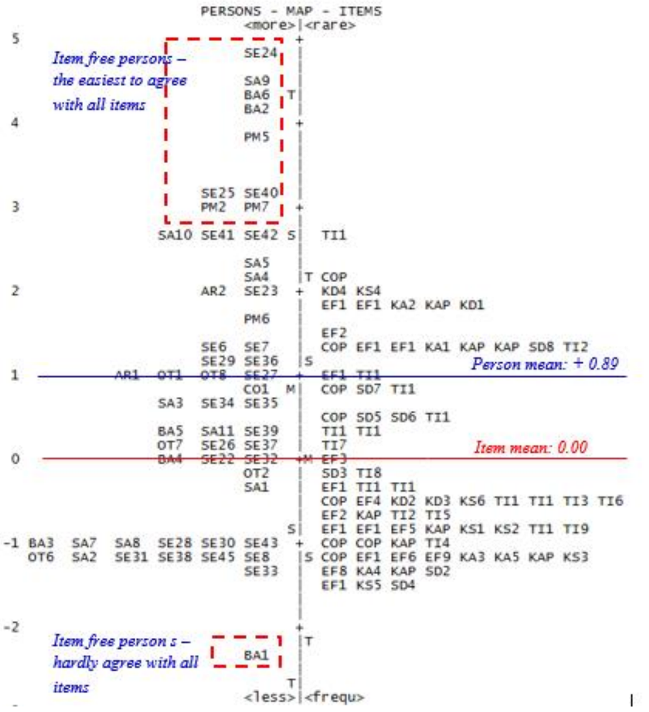

Figure 6: Person Item Distribution Map

Item misfit order table (Table 1) shows that all items are having positive point measure correlation values. However, there is an item having Z-standardize values above the acceptable range ($-2 < Z < +2$) which is item TI15. Item TI15 (*I think that mobile technology (SMS/MMS) is important for knowledge distribution*) could draw different opinions from the practitioners because the mentioned method (e.g. SMS, MMS) for knowledge distribution could be somewhat conventional as compared to more modern platform nowadays like social network and collaboration tool which are more relevant. This question could be rephrased to *I think that mobile technology (SMS/MMS) is still relevant for knowledge distribution*.

Table 1
Item Misfit Order

MEASURI	S.E.	MNSQ	ZSTD	MNSQ	ZSTD	CORR.	EXP.	OBS%	EXP%	ITEM
-0.45	0.25	1.86	3.10	2.15	3.70	0.28	0.64	61.10	66.30	TI15
-1.31	0.33	1.65	2.80	1.60	1.90	0.32	0.61	63.00	76.80	SD2
-0.90	0.28	1.53	1.80	1.72	2.30	0.36	0.61	70.40	75.50	TI16
0.30	0.34	1.29	1.30	1.47	1.60	0.39	0.58	79.60	79.20	TI1
-1.14	0.34	1.39	1.70	1.75	2.20	0.40	0.61	74.10	78.80	EF6
-0.27	0.33	1.35	1.50	1.24	1.00	0.44	0.62	70.40	77.50	TI10
-0.52	0.28	1.38	1.90	1.98	2.80	0.46	0.63	66.70	69.20	COP5
0.00	0.38	1.15	0.60	1.10	0.40	0.46	0.55	81.50	83.90	EF3
-0.50	0.34	1.28	1.20	1.17	0.70	0.49	0.61	72.20	79.30	EF4
0.90	0.34	1.21	1.30	1.54	1.50	0.50	0.60	68.50	76.60	TI11
-0.48	0.31	1.24	1.10	1.37	1.40	0.51	0.62	70.40	75.80	KS6
2.18	0.38	1.25	1.20	1.16	0.50	0.51	0.60	75.90	82.30	COP8
-0.27	0.36	1.21	0.90	1.20	0.70	0.52	0.59	75.90	81.20	TI13
-0.43	0.35	1.12	0.60	1.16	0.60	0.53	0.60	77.80	79.90	KD2
1.90	0.37	1.12	0.60	1.22	0.70	0.55	0.61	77.80	81.10	EF11
1.76	0.36	1.17	0.90	1.05	0.30	0.55	0.61	75.90	80.60	KAP2
-0.87	0.36	1.08	0.40	0.97	0.00	0.55	0.60	85.20	81.10	TI9
0.55	0.34	1.01	0.10	1.30	0.90	0.56	0.58	77.80	76.20	SD6

Person misfit table (Table 2) shows that there are 3 persons having negative point measure correlation: SE25, SE24 and PM2. SE25 and SE24 are software engineers while PM2 is a project manager. These persons also belong to the group of item free persons at the higher scale of PIDM map, who are easily agree with all items. For any kind of reasons, these persons appear to be problematic in answering the survey according to Rasch analysis. Removing of misfitting person could improve the result probably in a minor scale, even though person and item reliability are excellent at the moment.

Table 2
Person Misfit Order

MEASURE	Model S.E.	INFIT MNSQ	INFIT ZSTD	OUTFIT MNSQ	OUTFIT ZSTD	PT-MEASURE CORR.	PT-MEASURE EXP.	EXACT MATCH OBS%	EXACT MATCH EXP%	PERSON
4.82	0.52	1.02	0.20	1.49	0.90	-0.16	0.06	94.70	94.70	SE24
3.25	0.30	0.91	-0.50	0.80	-0.80	-0.09	0.13	78.90	80.30	SE25
2.92	0.28	0.82	-1.30	0.76	-1.30	-0.04	0.15	77.60	75.70	PM2
2.63	0.26	1.26	2.10	1.16	1.00	0.05	0.18	67.10	71.60	SA10
2.99	0.28	1.09	0.60	0.97	-0.10	0.05	0.15	80.30	76.80	PM7
1.94	0.24	1.88	8.20	1.83	6.70	0.06	0.24	63.20	64.60	AR2
2.70	0.27	0.84	-1.30	0.84	-1.00	0.06	0.17	77.60	72.60	SE42
3.89	0.36	1.10	0.50	1.33	0.90	0.07	0.09	88.20	88.10	PM5
4.37	0.43	1.07	0.30	1.19	0.50	0.07	0.07	92.10	92.10	BA6
1.59	0.24	1.20	2.30	1.23	2.20	0.11	0.28	64.50	63.40	PM6
4.57	0.47	0.97	0.10	0.68	-0.50	0.13	0.07	93.40	93.40	SA9
1.06	0.25	1.53	4.30	1.50	3.30	0.18	0.35	52.60	66.60	OT1
1.36	0.24	1.16	1.70	1.18	1.60	0.18	0.31	73.70	63.90	SE7
0.40	0.27	1.06	0.40	1.02	0.20	0.19	0.44	75.00	75.60	SE39
1.30	0.24	1.27	2.70	1.20	1.70	0.24	0.32	57.90	64.30	SE6
-0.06	0.29	2.55	5.30	2.74	4.90	0.25	0.50	55.30	81.40	SE22
0.18	0.28	0.88	-0.60	0.98	0.00	0.26	0.47	80.30	78.60	SE37
1.00	0.25	1.39	3.20	1.49	3.10	0.27	0.36	61.80	67.40	AR1
2.00	0.24	0.76	-3.00	0.72	-3.00	0.27	0.24	75.00	65.00	SE23

By further removing the misfitting persons, the unidimensionality is examined. Unidimensionality in Rasch is the key component of content validity. It refers to how well the items fit the constructs and if there exists second dimension.

Figure 7 below shows the unidimensionality testing using Rasch factor analysis. In dimensionality analysis, the variance explained by the first contrast in the residuals indicates whether there could be another dimension exists. For unexplained variance for first to fifth contrast, value of more than 15% is poor, 10-15% is fair, 5-10% is good, 3-5% is very good and less than 3% is excellent [32].

```
     Table of STANDARDIZED RESIDUAL variance (in Eigenvalue units)
                                                -- Empirical --    Modeled
Total raw variance in observations      =       130.7 100.0%        100.0%
  Raw variance explained by measures    =        54.7  41.8%         41.1%
    Raw variance explained by persons   =        28.8  22.0%         21.6%
    Raw Variance explained by items     =        25.9  19.8%         19.5%
  Raw unexplained variance (total)      =        76.0  58.2% 100.0%   58.9%
    Unexplned variance in 1st contrast  =         6.4   4.9%   8.4%
    Unexplned variance in 2nd contrast  =         5.4   4.1%   7.1%
    Unexplned variance in 3rd contrast  =         5.3   4.0%   6.9%
    Unexplned variance in 4th contrast  =         4.7   3.6%   6.2%
    Unexplned variance in 5th contrast  =         4.3   3.3%   5.7%
```

Figure 7: Unidimensionality

In this analysis, the unexplained variance for the first contrast is very good (4.9%), and the eigenvalue unit shows the strength of around 7 items. Raw variance explained by measure is 41.8%, giving a strong measurement dimension [33]. The criteria of unidimensionality is that over 40% of the variance should be attributable to the first dimension with an eigenvalue of less than 2.0 and the variance for the first contrast is less than 5% of the total unexplained variance [38]. Even though the eigenvalue is more than 2, the plot however looks quite random vertically (Figure 8), and so it can be concluded that there is no visible secondary dimension.

B. *Correlation and Regression Analysis*

With the elimination of the three misfitting persons, we proceed with correlation and regression analysis. Table 3 and 4 show the correlation test results which are produced by SPSS software. As can be seen, each of the identified components (SD, CoP, KM, Tech_Infra) that makes up the model is positively and significantly related towards EF_Goals.

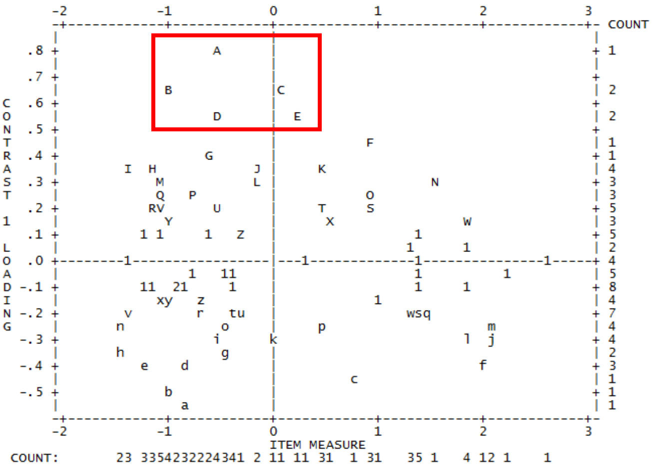

Figure 8: Standardized Residual Contrast

Table 3
Correlation between Components

	SD	CoP	KM	Tech_ Infra	EF_ Goals
SD	1				
CoP	.530**	1			
KM	.675**	.719**	1		
Tech_Infra	.655**	.688**	.749**	1	
EF_Goals	.628**	.671**	.823**	.802**	1

**Correlation is significant at the 0.01 level (2-tailed).

We are also interested to see overall relationship between Proj_Org and EF_Org. Expectedly, Proj_Org and EF_Org are also positively and significantly correlated between each other as shown in Table 4.

Table 4
Correlation between Components

	Proj_Org	EF_Org
Proj_Org	1	
EF_Org	.839**	1

**Correlation is significant at the 0.01 level (2-tailed).

Table 5 below shows the summary of descriptive statistics, correlation and regression analysis. KM, Tech_Infra and CoP have significant positive regression weight, confirming the positive relationship towards the achievement of EF_Goals.

SD however has lower regression weight and it is near to 0. Note that negative weight regression (opposite in sign from its correlation with the criterion) could indicate that the variables do not contribute towards EF_Goals even though they are moderately correlated. This can lead to multicollinearity when the weight is closer to negative [34]. Multicollinearity has no impact to the overall regression model and associated statistics such as R^2, F ratios and p values, and generally has no impact on the prediction made using the overall model [35]. In this study, multicollinearity is not an issue because the model is a not causal relationship model; the individual effects of individual variables are not a concern. A more causal relation towards EF goals and its effectiveness is more important. This however will be evaluated in future study.

Table 5
Descriptive, Correlation and Regression Analysis

Variable	Mean	Std. Dev	Correlation with EF_Goals	Multiple Regression b	Multiple Regression β	Collinearity Statistics Tolerance	Collinearity Statistics VIF
SD	20.902	2.274	0.628**	0.05	0.018	0.493	2.029
CoP	26.922	2.862	0.671**	0.078	0.036	0.432	2.316
KM	76.039	7.451	0.823**	0.401	0.482	0.318	3.148
Tech_Infra	65.02	6.055	0.802**	0.414	0.404	0.359	2.785

** Correlation is significant at the 0.01 level (2-tailed)

In further analysis on collinearity diagnostics, variance inflation factors (VIF) values indicate that the variables are moderately correlated. VIF between 5 and 10 indicates high correlation that may be problematic [36]. Notably, VIF values shown for SD is 2.029, not sufficiently enough to be overly concerned about. Also, note that the regression does not prove any casual relations from the predictors on EF goals; nevertheless, such casual relations are likely found intuitively. If they do exist, improving KM and technological aspects will make EF goals more achievable likely.

The multiple regression model with all four variables produce R2 = .756 (Table 6), F(4, 46) = 35.703, p < 0.001 as shown in the analysis of variance (Table 7). Therefore, the model is accepted with about 75.6% variance.

Table 6
Regression Output - Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.870[a]	.756	.735	3.191

a. Predictors: (Constant), Tech_Infra, CoP, SD, KM

Table 7
ANOVA[a]

	Model	Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1454.276	4	363.569	35.703	.000[b]
	Residual	468.430	46	10.183		
	Total	1922.706	50			

a. Dependent Variable; EF_Goals
b. Predictors: (Constant), Tech_Infra, CoP, SD, KM

VI. Conclusion and Future Works

The importance of knowledge and experience management for software development has led to the development of the model which is based on the successful experience factory framework. Several components are analyzed based on the literature that would potentially form an acceptable model. The identified components are divided into two organizations: Project Organization (Proj_Org) (SD and CoP) and Experience Factory Organization (EF_Org) (KM and Tech_Infra). The model formulation has gone through correlational research to seek the relationship among the components where the identified components are the variables and the EF goal is the outcome. Reliability analysis has been performed prior to the correlational and regression analysis. Reliability is excellent but three persons are found misfit. The invalid responses are eliminated so that a more reliable data can be used for further analysis. Correlational analysis reveals that there is a significant positively relationship between the variables towards EF goals and regression analysis indicates that the model is accepted with 75.6% variance. In the future, the model prototype will be developed and will be used as the instrument to validate the model (post-evaluation). In the post-evaluation, the model will be validated using structured equation modelling to analyze the causal relationship between the proposed model and its effectiveness and efficiency goals. Efficiency and effectiveness of the model will be further evaluated based on Jennex and Olfman success model for knowledge management [39] which focuses on system quality, knowledge quality, service quality, intent to use/perceive benefit, user satisfaction and net benefits.

References

[1] E. H. M. Huzita, G. C. L. Leal, R. Balancieri, T. F. C. Tait, E. Cardoza, E., R. R. de Moura Penteado, and R.L.Vivian, "Knowledge and contextual information management in global software development perspectives," in *IEEE Seventh Int. Conf. on Challenges and Global Software Engineering Workshops*, 2012, pp. 43-48.

[2] A. P. Chaves, I. Steinmacher, V. Vieira, and E. H. M. Huzita, "A context conceptual model for a distributed software development environment," in *Int. Conf. on Software Engineering and Knowledge Engineering (SEKE10)*, 2010, pp. 437-442.

[3] M. Hanafiah, R. Abdullah, J. Din, and M. A. A. Murad, "Towards developing collaborative experience based factory model for software development process in cloud computing environment," *International Review on Computers and Software (I.RE.CO.S.)*, vol. 10, no. 3, pp. 340-350, Mar. 2015.

[4] N. Sharma, K. Singh, and D.P. Goyal, "Experience base approach to software process improvement: comparative analysis and design of improved model advanced," in *Second Int. Conf. on Computing and Communication Technologies (ACCT)*, 2012, pp. 30-35.

[5] M. Ivarsson, and T. Gorschek, "Tool support for disseminating and improving development practices", *Software Quality Journal*, vol. 20, no. 1, pp. 173-199, Mar. 2011.

[6] P. Ardimento, M. Cimitile, and G. Visaggio, "Distributed software development with knowledge experience packages," in *On the Move to Meaningful Internet Systems: OTM 2013 Workshops: Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program*, Y. T. Demey and H. Panetto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 263–273.

[7] G. Matturro and A. Silva, "ReBEC: a method for capturing experience during software development projects," in *Knowledge Engineering and Management by the Masses: 17th International Conference*, P. Cimiano and H. S. Pinto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 524–533.

[8] V. R. Basili, G. Caldiera, and H. D. Rombach, *Experience Factory, Encyclopedia of Software Engineering.* New York: John Wiley & Sons, 1994.

[9] A. Koennecker, R. Jeffery, and G. Low, "Implementing an experience factory based on existing organizational knowledge," in *Proc. of the Australian Software Engineering Conference*, 2000, pp. 53-62.

[10] K.-D. Althoff, A. Hanft, and M. Schaaf, "Case factory - maintaining experience to learn," in *Advances in Case-Based Reasoning: 8th European Conference, ECCBR 2006*, T. R. Roth-Berghofer, M. H. Göker, and H. A. Güvenir, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 429–442.

[11] Z. Chen, D. Port, Y. Chen, and B. Boehm, "Evolving an experience base for software process research," in *Unifying the Software Process Spectrum: International Software Process Workshop, SPW 2005*, M. Li, B. Boehm, and L. J. Osterweil, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 433–448.

[12] V. R. Basili, and G. Caldiera, "Improve software quality by reusing knowledge and experience," *Sloan Management Review*, pp. 55-64, Oct. 1995.

[13] R. Ackoff, "From data to wisdom," *Journal of Applied Systems Analysis*, vol. 16, pp. 3-9, 1989.

[14] K. Schneider, *Experience and Knowledge Management in Software Engineering*, Berlin: Springer-Verlag, 2009.

[15] I. Nonaka, and H. Takeuchi, *The Knowledge-Creating Company*, Oxford University Press, New York, NY, 1995.

[16] D. Piktialis, and K. A. Greenes, *Bridging the Gaps: How to Transfer in Today's Multigeneration Workforce*, USA: The Conference Board, 2008.

[17] M.-I. Sanchez-Segura, F. Medina-Dominguez, A. de Amescua, and A. Mora-Soto, "Improving the efficiency of use of software engineering

practices using product patterns," *Information Sciences*, vol. 180, no. 14, pp. 2721–2742, Jul. 2010.

[18] M. Paasivaara, and C. Lassenius, "Communities of practice in a large distributed agile software development organization – case Ericsson," *Journal of Information and Software Technology*, vol. 56, no. 12, pp. 1556–1577, Dec. 2014.

[19] E. Wenger, R. McDermott, and W. Snyder, *Cultivating Communities of Practice: A Guide to Managing Knowledge*. USA: Harvard Business School Press, 2002.

[20] I. Hamburg, S. Engert, and A. Petschenka, "Communities of practice and Web 2.0 to support learning in SMEs." in *6th Romanian Educational Network (RoEduNet) International Conference*, Craiova, Romania, 2007, pp. 152-155.

[21] P. Penfold, "Virtual communities of practice: collaborative learning and knowledge management," in *Third International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 482-485.

[22] E. Ras, J. Rech, and S. Weber, "Knowledge services for experience factories," in *WM2009: 5th Conference on Professional Knowledge Management*, 2009, pp. 232-241.

[23] T.S Dillon, E. Chang, and P. Wongthongtham, "Ontology-based software engineering – software engineering 2.0," in *19th Australian Conference on Software Engineering*, 2008, pp. 13-23.

[24] P. Yara, R. Ramachandran, G. Balasubramanian, K. Muthuswamy, and D. Chandrasekar, "Global software development with cloud platforms," in *Software Engineering Approaches for Offshore and Outsourced Development: Third International Conference, SEAFOOD 2009*, O. Gotel, M. Joseph, and B. Meyer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 81–95.

[25] T. Pringle, S. Mukherjee, and T. Baer, "Thriving in the age of big data analytics and self-service," Ovum Consulting, retrieved at http://docplayer.net/8746101-Thriving-in-the-age-of-big-data-analytics-and-self-service.html, 2015.

[26] G. Rasch, *Probabilistic models for some intelligence and attainment tests*, Copenhagen: DanmarksPpaedagogoske Institut, 1960.

[27] W. J. Boone, J. William, J. R., Staver and M. S. Yale, *Rasch Analysis in the Human Sciences*, Springer Science Business Media Dordrecht, 2014.

[28] American Educational Research Association, American Psychological Association, National Council on Measurement in Education, Joint Committee on Standards for Educational, & Psychological Testing (US), *Standards for educational and psychological testing*. Washington, DC: American Educational Research Association, 2014.

[29] P. Baghaei, "The Rasch Model as a Construct Validation Tool", *Rasch Measurement Transactions*, vol. 22, no.1, pp. 1145-1146, 2008.

[30] G. Hall, "Pearsons correlation coefficient," *In other words*, vol. 1, p. 9, 2015.

[31] R. G. van den Berg, *Linear Regression in SPSS - A Simple Example* retrieved at https://www.spss-tutorials.com/linear-regression-in-spss-example/, 2016.

[32] W. P. Fisher, "Rating Scale Instrument Quality Criteria", *Rasch Measurement Transactions*, vol. 21, no. 1, pp.1095, 2007.

[33] K. J Conrad, K. M Conrad, M. L Dennis, B. B. Riley, and R. Funk, "Validation of the HIV Scale to the Rasch Measurement Model, GAIN Methods Report 1.1," *HIV Rasch Report 1.1*, 2009.

[34] R. G. van den Berg, "Multiple Regression – What Is It?" retrieved at https://www.spss-tutorials.com/multiple-regression-what-is-it/, 2015.

[35] T. Baguley, *Serious stats: A guide to advanced statistics for the behavioral sciences*. Basingstoke: Palgrave, 2012.

[36] E. Martz, "Enough Is Enough! Handling Multicollinearity in Regression Analysis," retrieved at http://blog.minitab.com/blog /understanding-statistics/handling-multicollinearity-in-regression-analysis, 2013.

[37] P. Bourque, and R. Fairley, *Guide to the Software Engineering Body of Knowledge: SWEBOK (R)): 3.0*, Los Alamitos, CA, USA: IEEE Computer Society Press, 2014.

[38] J. M. Linacre, "A User's Guide to Winsteps: Rasch-Model Computer Programs," retrieved at http://www.winsteps.com/winman/index.htm, 2016.

[39] M. E. Jennex, and L. Olfman. "A model of knowledge management success," *Journal of Knowledge Management*, vol. 2, no. 3, pp. 51-68, 2006.