# Experience and Innovation Factory: Adaptation of an Experience Factory Model for a Research and Development Laboratory

*Full Paper*

**Attany Nathaly L. Araújo, Keli C.V.S. Borges, Sérgio Antônio Andrade de Freitas, Edna Dias Canedo and Wander Cleber Maria Pereira da Silva**
Faculty UnB Gama (FGA) – University of Brasília (UnB)
Mailbox 8114 – 72405-610 - Brasília, DF, Brazil
nathalyaraujoo@gmail.com, keliunb@gmail.com,
sergiofreitas@unb.br, ednacanedo@unb.br,wandercleber@unb.br

## Abstract

This paper aims to present a knowledge management proposal for a software factory organization. A software factory requires a holistic organization, in which many factors must be taken into account, such as: people management and business management. The complexity of the structure leads us to elaborate on an adapted model of Experience Factory to meets the needs of research and development laboratories. The construction of the adapted model used a bibliographical research about Experience Factory models, the characterization of the project, a mapping between the project's characteristics and the activities of the Experience Factory models. The Experience Factory models attend to the main characteristics identified for the Software Factory Laboratory (SFL): development, capacitation, training, research and innovation. Finally, we tested and analyzed the results on the proposed model on the knowledge generated by the SFL during the software development process.

**Keywords**

Software Factory, Knowledge Management, Experience Factory.

## Introduction

A Software Factory Laboratory (SFL) is an educational vehicle for universities, where the artifacts produced in the factory serve to improving learning and provide teaching materials in close collaboration with industry (Fagerholm et al. 2013). The main established needs for such project are development, capacitation and training, research and innovation, and knowledge management.

Aligned to these characteristics, the utilization of an Experience Factory approach was proposed. The concept of continuous improvement according to particular objectives and characteristics of the organization, as well as its own growing set of experiences attributed to ongoing or finished development efforts, drive this approach. The Experience Factory works with the premise that each organization has its own needs, objectives and characteristics, which must be the essential conductors of the software improvement program of the organization (Basili and McGarry 1997).

For Basili and Seaman (2002), an Experience Factory analyzes and synthetizes all kinds of experiences, including: lessons learnt, project data, and technology reports, providing repository services for the storage of the experience. An Experience Factory can be a logical and/or physical organization, but it is important that its experiences are separated and independent from the experiences conducted in the project's organization (Basili et al. 1992).

According to Basili and Caldiera (1995), the Experience Factory is the unit that supports the reutilization of experiences and collective learning, through the development, requisitions supplying and updating, and grouping of competences (experience packages) to be used by the project's organization. In this way, the purpose of this work is the study about Experience Factory models and the proposition of a model adapted to the context of the SFL. For this end, the following were done: a bibliographical research for a higher affinity with the Experience Factory theme; an structured questionnaire with members of the SFL to gather information; a context characterization of the laboratory, validated by its members; a mapping between the characteristics of the SFL and the activities of the models of Experience Factory identified in the bibliographical research and, finally, the proposition of a model adapted to meet these criteria.

The structure of this paper consists of sections. In Section 2, we have the project characterization. Section 3 addresses the theoretical reference and Section 4, the detailing of the methodology. The conclusion and future works are presented in Section 5. In the end, there are the bibliographical references that acted as a base for the development of this work.

# Project Characterization

The SFL organization is centered on the concepts of a Software Factory, adding other components in its environment. Aside from the development of the product (software), typically found in a Software Factory, it also has the objective of creating an environment that provides capacitation and research, as well as the development of tools and methodologies in the Software Engineering field, involving researchers and students. These are the units that compose the SFL:

- **Development**: The Development is a unit of the SFL that focuses on building the Software product. The SFL is adept to Scrum, the agile software development methodology.
- **Capacitation and Training**: The SFL seeks to offer the students and researchers an environment for experimentation, capacitation, and training. The very students are fit to create training material and teach other students. The majority of the capacitation is provided to the members of the SFL by the course.
- **Research and Innovation**: The fields of research in the SFL are quite broad, as it includes: patterns, methodologies, methods, models, processes, and supporting tools. Furthermore, this field has an essential role in the search for innovation. The projects seek to solve real world problems, besides the research applied to the academy
- **Knowledge Management**: the student members of the SFL possess knowledge obtained from both the graduation as well as in the SFL. To manage this knowledge is as important as the development processes executed by the SFL. Currently, there is no defined knowledge management process in the SFL.

# Literature Review

## *Software Factory*

Jafarinezhad and Ramsin (2012) define Software Factory as a Software Product Line (SPL) approach, a paradigm for systematic reuse of software products. Software Factory is a SPL for the industrialization of software development. Software Factories are actually the logical next step in the continuing evolution of software development methods and practices.

Fagerholm et al. (2013) say: "[...] a software factory can be considered as an infrastructure platform that provides and supports software engineering research, education, and entrepreneurship. As a platform, it serves multiple purposes. It is a test bed for software engineering ideas and a source for original basic scientific software development research. It is an educational vehicle for universities, where the artifacts produced in the factory serve to improving learning and provide teaching materials in close collaboration with industry".

The work developed in a Software Factory is based on the concept of collaborative learning. The term "collaborative learning" refers to an instruction method in which students at various performance levels work together in small groups toward a common goal. The students are responsible for one another's learning as well as their own. Thus, the success of one student helps other students to be successful

(Gokhale 1995). Griss and Wentsel (1994) presents the term "flexible software factory", derived from the ideas and methods used by organization re-engineering as applied to manufacturing, product development and flexible automation. We are looking at the processes, organization and tool environments for these organizations, specifically at how standard process elements and organizational design methods can be used to construct an optimal and efficient software factory.

## *Experience Factory*

Many authors, such as Basili, Caldiera and Seaman, discuss the concept of Experience Factory. With the intent of comprehending the structures of Experience Factories proposed by literatures, the following five models are presented.

Model 1, from Basili (1989), proposes the Improvement Paradigm, which consists of the idea that the improvement of processes can be reached through iterative planning, execution of plans, and reutilization of experiences between software development projects, in courses or in an organization. The paradigm explicitly recognizes the necessity of capturer and reutilizing knowledge, products and processes of previous projects. For Basili (1989), a reutilization-oriented process model must consider reutilization, learning, and feedback as key components and put all experience under the control of a base of experiences.

The implementation of the Improvement Paradigm considers two distinct and separated organizational structures: The Project Organization, whose objective is to deliver the systems requested by the clients, and the Experience Factory, whose role is to analyze and monitor the development of projects, packing experiences to be reutilized and supplying them to the project's organization.

In model 2 (based on model 1), Basili and Caldiera (1995) present the Quality Improvement Paradigm for the software industry, which is based in manufacture models but focused in the reutilization of learning and experiences through the establishment of Experience Factories. Experience Factory is the organization that supports the reutilization of experiences and collective learning, updating and supplying, as requested, groups of competences to be used by the project's organization (Basili and Caldiera 1995).

The project's organization feeds the Experience Factory with products, plans, processes and models used in its software development, and the data collected during the development and operation. The Experience Factory transforms this into reusable units and supplies them to the project's organization, together with specific monitoring, the consulting support (Basili and Caldiera 1995). The objective of the project's organization is to produce and maintain software. The Experience Factory provides direct feedback for each project, as well as objectives and models adapted from similar projects (Basili and Caldiera 1995).

The Experience Factory, acting as support team, sustains and facilitates the interaction between developers and analysts, saving and maintaining information, making them easily recoverable, and controlling and monitoring their access (Basili and Caldiera 1995). Model 3 is the "PIA Experience Factory – The PEF Model", a model which was produced by the work group called PERFECT consortium, between 1996 and 1997. The work of this group presents a model of how an Experience Factory can work in tandem with software development projects and management of an organization, to ensure that the organization gathers experiences from previous projects and that other projects make use of these Experiences (The PEF Model 1990).

The PEF model is based in three main blocks: the sponsoring organization, the software development projects, and the Experience Factory. The sponsoring organization constitutes the entire organization and possesses management-related activities. The software development project is described in three phases: pre-project, execution, and post-project. These phases are actively supported by the Experience Factory, which helps the projects use their processes in a more efficient manner (The PEF Model 1990).

In relation to model 4, Basili et al. (2001) present the "Knowledge Dust to Pearls" approach which, influenced by the Quality Improvement Paradigm (QIP), brings notions of continuous improvement and iterations as the main way of planning, execution, evaluation, and improvement of processes. This approach takes the organization to the management of knowledge and gradual improvement.

In this approach brought by Basili et al (2001), the Experience Factory organization gathers data produced by the project's organization. The data goes through an analysis phase and is synthetized in higher levels of knowledge, which are then packed in the form of experience packages. The experience packages are stored in the base of experiences and are made available for the project's organization, mainly in the form of business support supplied by the group. This model adds a short and new feedback circuit to the Project's Organization. This allows for the organization to invest less in the current moment and have faster returns (Basili et al. 2001).

Finally, in model 5, from Basili and Seaman (2002), the base for the Experience Factory Organization concept is that software development projects can improve their performance (in terms of cost, quality and planning) with the influence of the experience from past projects. Basili and Seaman (2002) separate the responsibilities of an Experience Factory into two distinct organizations: project's organization and Experience Factory. The project's organization uses experience packages to deliver software products, while the Experience Factory supports the development of software by supplying adapted experiences.

For Basili and Seaman (2002), the Experience Factory must: (a) pack experiences, analyzing, synthetizing, and creating models that represent abstractions of that experience; (b) keep a base of experiences or repository for data, models, and other means of knowledge and experiences; (c) support projects in the identification and utilization of proper experiences for each situation.

To explore the similarities and differences between the presented Experience Factory models, the Table 1 was developed to presents a detailing of the entities, activities, and utilization context of each of the presented model.

| Activity/ Entity | Model 1 (Basili 1989) | Model 2 (Basili and Caldiera 1995) | Model 3 (The PEF Model 1990) | Model 4 (Basili et al. 2001) | Model 5 (Basili and Seaman 2002) |
|---|---|---|---|---|---|
| Entity 1 | Project's Organization | Project's Organization | Sponsoring Organization | Project's Organization | Project's Organization |
| Entity 2 | Experience Factory | Experience Factory | Experience Factory | Experience Factory | Experience Factory |
| Activity 1 | Characterizing | Characterize/ Establish objectives/ Choose processes | Sponsoring Organization (objectives, resources) | Planning | Planning (needs) |
| Activity 2 | Planning | Execute process | Software Development Project | Doing | Accomplish |
| Activity 3 | Making | Analyze/Pack | Experience Engineering | Analysis (dust knowledge) | Analyze/Pack (Synthetize) |
| Activity 4 | Analysis | Storing in the Base of Experiences | Storing in the Base of Experiences | Storing in the Base of Experiences | Storing in the Base of Experiences |
| Activity 5 | Synthesis | Project Support | Project Support | Return of Pearls of Knowledge | Project Support |
| Activity 6 | Base of Experiences (formalize, customize, generalize) | - | Improvement Management | Synthesis | - |

| | | | | | |
|---|---|---|---|---|---|
| Activity 7 | Feedback and learning | - | - | Business Support | - |
| Activity 8 | - | - | - | Return of Packed Experiences | - |
| Context | Software Development | Software Industry | Software Projects | Software Industry | Software Development |

**Table 1. Detailing of activities and entities of the Experience Factory models.**

### *Knowledge Management*

Knowledge management (KM) is the process of creating, sharing, using and managing the knowledge and information of an organization. It refers to a multidisciplinary approach to achieving organizational objectives by making the best use of knowledge (Girard, 2015). Knowledge management efforts typically focus on organizational objectives such as improved performance, competitive advantage, innovation, the sharing of lessons learned, integration and continuous improvement of the organization. These efforts overlap with organizational learning and may be distinguished from that by a greater focus on the management of knowledge as a strategic asset and on encouraging the sharing of knowledge. KM is an enabler of organizational learning.

## Methodology

This work was developed from a bibliographical research about Experience Factories, interviews with SFL members, a context characterization and validation from SFL members, a mapping between the SFL context and the Experience Factory models. Finally, yet importantly, a proposal of adaptation of an Experience Factory model to the context of the SFL and a validation of the same by the laboratory members. These activities are represented in Figure 1.

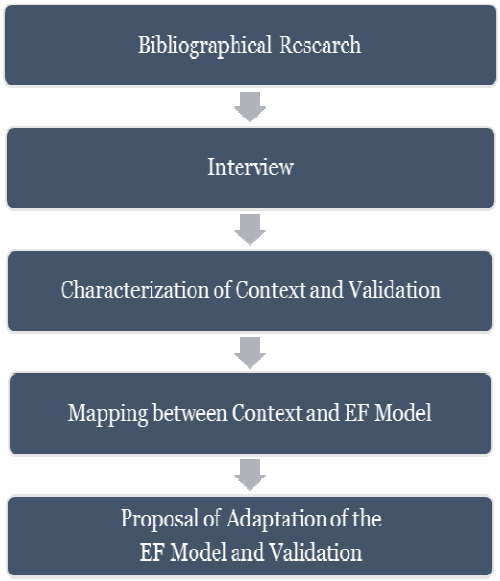


**Figure 1. Methodology Activities (Kitchenham, 2007), (Petersen, 2008).**

### *Bibliographical Research*

For the elaboration of this work, the first activity was the Bibliographical Research, executed through manual search in scientific electronic bases about the Experience Factory theme. As a result of this

research, some different approaches related to the models of Experience Factories were identified. These are detailed in Section 3.

### *The Interview*

The second activity was the interview. According to Gil (2002), it is an exploratory study that seeks to address realities that aren't well known to the researcher, or to offer an approximate view of the problem being researched. The interview was used like a structured questionnaire done with SFL researchers as well as students of the laboratory team, that aims to collect SFL members' perception. allowing for the collection of data to move on to the next step of the methodology.

The questionnaire presents to respondents a scenario in which described the experience factory itens and asked she/he to rate on a scale from 1 to 5 (Likert) perceived development for 9 indicators: 1)Planning, 2)Research and Training, 3)Execution, 4)Software Product, 5)Analysis, 6)Initial feedback, 7)Experience Base, 8)Packing and 9)Final feedback. Also it provides demographic information: sex, age, if previously worked and time at the SFL.

### *Characterization of Context and Validation*

The third activity was the characterization of the SFL context based on the information obtained with the informal interview. Posteriorly, member of the SFL validated the characterization of the context. The characterization of the SFL context can be found in Section 2.

### *Mapping between SFL context and Experience Factory Models*

Figure 2 shows the relationship between characteristics of the SFL context and the Experience Factory models already presented. Each characteristic is associated to the most similar model:

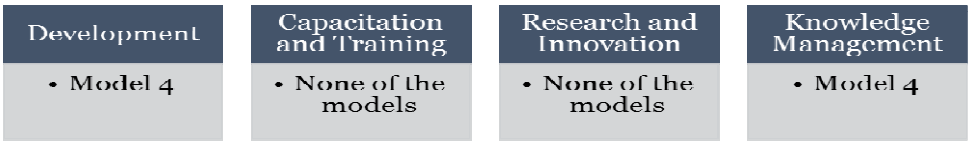


**Figure 2. Mapping between SFL context and Experience Factory Models (Kitchenham, 2007), (Petersen, 2008).**

#### Development x Experience Factory Models

As shown in Table 1, Experience Factory models 1 and 5 possess contexts related to Software Development. Models 2 and 4 are based on Software Industry, while model 3 has Software Projects as it main scope.

Given that Development is a typical activity of any Software Factory, it's possible to say that all models are suited to this characteristic of the SFL. Model 4, however, has the additional advantage of possessing a short feedback circuit for the Project's Organization, which brings faster returns (Basili et al. 2001). This faster feedback has higher compatibility with the Scrum methodology, employed in the laboratory, making it so model 4 is the best-suited model in this characteristic.

#### Knowledge Management x Experience Factory Models

All the Experience Factory models presented in Section 3 meet the Knowledge Management activity through the Base of Experiences activity. The Base of Experiences incorporates historic information from all the projects in such a way that they may be useful to future projects (Basili 1989).

The authors of model 4, however, explicitly recognize that the presented approach leads the organization to gradually reach the management of knowledge (Basili et al. 2001). Therefore, model 4 is the most adhering to this SFL characteristic. The Capacitation and Training and Research and Innovation characteristics can't be mapped in any of the Experience Factory models.

## *Proposal of the Adapted Model of Experience Factory*

This section seeks to show the proposed model for utilization in the SFL context, elaborated from the previous mapping. Figure 3 brings the Experience and Innovation Factory Model adapted from model 4 of the Experience Factory.

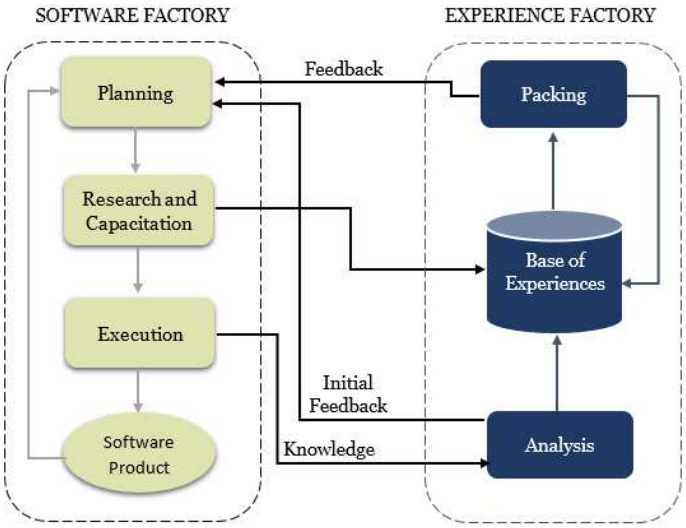


**Figure 3. Adapted Experience and Innovation Factory Model.**

It's possible to establish a parallel between the activities of the adapted model and the activities of model 4. For this reason, Table 2 was elaborated.

| Adapted Model | Model 4 of Experience Factory (Basili et al. 2001) |
|---|---|
| Planning | Planning |
| Research and Capacitation | - |
| Execution | Doing |
| Software Production | - |
| Analysis | Analysis (Dust knowledge) |
| Storing in Base of Experiences | Storing in Base of Experiences |
| Initial Feedback | Return of Pearls of Knowledge |
| Packing | Synthesis |
| - | Business Support |
| Feedback | Return of Packed Experiences |

**Table 2. Parallel between adapted model and Experience Factory Model.**

It can be seen in table 2 that the Research and Capacitation and Software Production activities were included in the adapted model, while the Business Support one was removed. The detailing of the activities of the adapted model is presented ahead:

- **Planning**: the planning activity includes the planning of a software development project to be executed by the SFL, as well as the planning of Sprints existing in that project. When starting a new project, the SFL must perform the following tasks: (a) Initial analysis of the client's demands; (b) Analysis of required capacitation, and (c) Analysis of related research.

- **Research and Capacitation**: the research and capacitation activity seeks to identify, in the base of experiences, the stored knowledge that was deemed necessary in the previous activity. If this knowledge already exists in the base of experiences, they must be used by the members of the SFL as a method of capacitation for the execution of the project. If they still don't exist in the base of experiences, sources, courses and others, or SFL members possessing such knowledge must be found to supply this need. Additionally, the activity should promote actions that can capacitate the others for the execution of the project.
- **Execution**: the Execution activity includes all the activities inherent to the SFL's software development, so that a software product is obtained by the end of the project. For this end, the Execution activity repeats after each of the project's Sprints, according to the established planning. In this activity, artifacts that constitute the knowledge of the SFL are generated, including scientific knowledge originating from researches and publications, as well as the experience extracted at the end of the project. These are called "dust knowledge". During the Execution activity, the training and capacitation of SFL members also takes place.
- **Software Product**: the Execution activity generates the element called Software Product. After every Sprint, a software increment is obtained, forming a software product by the end of the project. This software product is delivered to the client, who will validate it. If any corrections are necessary, these will be planned in the Experience Factory Planning activity. This way, a new execution will take place and, posteriorly, the delivery of a new software product.
- **Analysis**: the Analysis activity uses the knowledge and experiences generated by the execution activity. This simpler and untreated knowledge forms the element called "dust". Certain knowledge elements are selected, according to the experience extracted, to be stored in the base of experiences. The stored knowledge is called a mini-pearl.
- **Storing in the Base of Experiences**: repository of SFL knowledge and experiences, where the mini-pearls and pearls of knowledge are stored. The base of experiences is accessed through the Research and Capacitation, Analysis, and Packing of Experience Factory activities.
- **Initial Feedback**: the initial feedback is passed along from the Experience Factory organization to the Software Factory organization after each Sprint, so that the Planning activity of the next Sprint can be executed taking into account the necessary improvements identified in the end of the Sprint.
- **Packing**: activity that transforms the mini-pearls, already stored in the base of experiences, into pearls of knowledge. This activity is done in the end of the project. After the packing, these pearls are stored in the base of experiences and returned to the organization so that they can be reutilized through Feedback.
- **Feedback**: the feedback is passed along from the Experience Factory organization to the Software Factory organization at the end of each project. This way, the next project will be planned considering the necessary improvements that were identified.

## Results and Discussion

To evaluate the proposal of an experience factory we create a questionnaire of 9 questions. Each question represents the experience factory activities: Planning, Packing, Execution and so on. The questionnaire measures the laboratory members' perception about their experience on use of the model. Each question is a Likert of 5 values based on a proposed statement. An example of statement and options are: "The activity Planning includes the planning of a software development project that will be executed by SFL, as well as the sprints planning. Mark according to the degree of contribution for the Planning activity to the development of work in SFL: 1) This activity does not contribute in any way. 2) This activity contributes little. 3) This activity contributes reasonably. 4) This activity contributes significantly. 5) This activity contributes greatly."

The questionnaire was answered by all the 15 members (n=15) of the laboratory. The Cronbach Alpha factor (Cronbach, 1951) used to analyze the questionnaire's reliability indicates a value 0.73. Table 3 shows the Means and Standard Deviation (SD) of the results.

| Item | Mean Value | SD |
|---|---|---|
| Planning | 4.67 | 0.22 |

| | | |
|---|---|---|
| Research and Training | 4.07 | 0.73 |
| Execution | 4.73 | 0.60 |
| Software Product | 4.47 | 0.52 |
| Analysis | 4.13 | 0.38 |
| Initial feedback | 4.47 | 0.38 |
| Experience Base | 4.27 | 0.60 |
| Packing | 4.20 | 0.83 |
| Final feedback | 4.27 | 0.60 |

**Table 3. Results per item.**

Table 3 shows a high perception linking between the Experience Factory Model and the individual items. Also points out a great impact on the Planning (4.67) and the Execution (4.73). This indicates that a SFL will improve initial phases of a project (Planning) and its development phase (Execution). A less impact was detected on Research and Training (4.07), and Analysis (4.13). The value for the item Research was expected as undergraduate students do not have a direct relation between Research and Software Laboratory. By the other hand, the value for Analysis (4.13) was not expected. Our conclusion is that the member are not dedicating much time on the analysis phase.

We also conducted a two-tailed Pearson Correlation study among all the items of the questionnaire (including the demographic items). The result indicates:

- A correlation of the most recent members in the laboratory with Execution (-0,75, sig 2-tailed = 0.001), with Planning (-0,55, sig 2-tailed = 0.032) and with Feedback (-0,47, sig 2-tailed = 0.079). This indicates that the Experience Factory use influences the central phases of a software development cycles for the most recent members in the laboratory. The same did not happen with the older members of the laboratory, which presented a similarity percent less than 15% in the best case in all variables. One hypothesis for this is the existence of the natural inertia of those who already have a habitual software development model and will oppose to the implementation of a new model.
- A strong perception of the more experienced members (who have already done internship outside the laboratory) that the Planning (0.71, sig 2-tailed = 0.003), Execution (0.69, sig 2-tailed = 0.004), Final Feedback (0.61, sig 2-tailed=0.015) e Software Product (0.56, sig 2-tailed=0.031) were influenced by the experience factory model. One explanation for this is that these members have already had contact with these elements in a business or professional environment and therefore have a better perception of these elements in the experience factory model.
- A correlation between the Planning and Software Product (0.85, sig 2-tailed = 0) and Final feedback (0.72, sig 2-tailed = 0.002). This indicates that the central question of software development process, which is the software product, is a highly correlated element, in the experience factory model, with the initials and finals elements (Planning e Final Feedback). This shows a users perception that experience factory acts in the central phases of knowledge management.
- Finally, was identified a correlation between Research and Training and Experience Base (0.58, sig 2-tailed =0.0023). This indicates the perception about the importance that has a experience base in relation to staff training and research in the development methodologies area.

# Conclusion

Pursuing the improvement of the offered services and, consequently, the improvement of the competitiveness, the organizations seek models that can support them in this challenge. The Experience Factory context fits in this context, bringing improvement paradigms associated to knowledge management.

The result from the analysis of the questionnaire indicates a perception that a SFL will have benefits from using an Experience Factory. Also points out a strong correlation among Planning, Software Product and Final feedback.

For the execution of the research project and development in a research and development laboratory, the Experience Factory models seen in this work were adapted to find a final model that could attend to the main characteristics identified for the laboratory in question: development, capacitation and training, research and innovation, and knowledge management.

The model was adapted in order to provide the aspects of capacitation, training, research, and innovation, as well as the intrinsic development and knowledge management, to the SFL environment. As a future work to verify the efficiency of this adapted model, the cyclical execution of the activities proposed in the SFL adapted model is deemed necessary. This allows for a complete analysis of the model from the results obtained after each cycle, possibly resulting in new adaptations to the proposed model.

## REFERENCES


Basili, V. R. 1989. "Software development: a paradigm for the future," Proceedings of the 13th Annual International Computer Software and Applications Conference, pp. 471–485 (doi: 10.1109/CMPSAC.1989.65127).

Basili, V. R., and Caldiera, G. 1995. "Improve Software Quality by Reusing Knowledge and Experience," Sloan Management Review, pp. 55–64 (available at http://www.cs.umd.edu/users/basili/publications/journals/J55.pdf).

Basili, V., and McGarry, F. 1997. "The Experience Factory: How to Build and Run One," Software Engineering, 1997., Proceedings of the 1997 (19th) International Conference on, pp. 643–644 (doi: 10.1109/ICSE.1997.610446).

Basili, V. R., and Seaman, C. 2002. "The Experience Factory Organization," IEEE Software (2:June), pp. 30–31.

Basili, V., Caldiera, G., McGarry, F., Pajerski, R., Page, G., and Waligora, S. 1992. "The software engineering laboratory - an operational software experience factory," International Conference on Software Engineering, pp. 370–381 (doi: 10.1109/ICSE.1992.753514).

Basili, V., Costa, P., Lindvall, M., Mendonca, M., Seaman, C., Tesoriero, R., and Zelkowitz, M. 2001. "An experience management system for a software engineering research organization," Proceedings 26th Annual NASA Goddard Software Engineering Workshop (KMWorld), pp. 29–35 (doi: 10.1109/SEW.2001.992652).

Cronbach, L. J. 1951. "*Coefficient alpha and the internal structure of tests*". Psychometrika, 16(3), pp. 297-330.

Fagerholm, F., Oza, N., and Münch, J. 2013. "A Platform for Teaching Applied Distributed Software Development," pp. 1–5.

Gil, A. C. 2002. "Como Elaborar Projetos de Pesquisa Como Elaborar Projetos de Pesquisa" (doi: 10.1111/j.1438-8677.1994.tb00406.x).

Girard, John P.; Girard, JoAnn L. 2015. "Defining knowledge management: Toward an applied compendium" . Online Journal of Applied Knowledge Management. 3 (1): 14.

Gokhale, A. A. 1995. "Collaborative Learning Enhances Critical Thinking," Journal of Technology Education (7:1), pp. 22–30 (doi: 10.1007/978-1-4419-1428-6_910).

Griss, M. L., and Wentzel, K. D. 1994. "Hybrid domain-specific kits for a flexible software factory," Proceedings of the 1994 ACM symposium on on Applied Computing, SAC (doi: 10.1145/326619.326658).

Jafarinezhad, O., and Ramsin, R. 2012. "Towards a process factory for developing situational requirements engineering processes," 27th Annual ACM Symposium on Applied Computing, SAC 2012, pp. 1089–1090 (doi: 10.1145/2245276.2231946).

Perfect Consortium. 1990. "Handbook PIA Experience Factory - The PEF Model -," Esprit project (available at http://www.tarrani.net/mike/docs/PIAPEF.pdf).

Kitchenham, B. 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering, Technical Report. Keele University.UK.

Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. 2008. Systematic mapping studies in software engineering, ACM. New York.