Software & Systems Engineering Interplay and the SEMAT Kernel

Article · June 2018

CITATION
1

READS
211

2 authors:

Marcel Jacques Simonette
University of São Paulo
40 PUBLICATIONS **74** CITATIONS
SEE PROFILE

Edison Spina
University of São Paulo
73 PUBLICATIONS **101** CITATIONS
SEE PROFILE

The Voice of the Systems

The Journal Of The Israeli Systems Engineers

ISSUE NO.22 | June 2018

- **Software & Systems Engineering Interplay and the SEMAT Kernel** (pp. 6-20)
  Marcel Jacques Simonette and Edison Spina
- **Lateral Thinking: A Creative Method Example** (pp. 21-34)
  Avner Engel
- **Four Layers Approach for Developing System Thinking Assessment Tool** (pp. 35-46)
  Moti Frank and Sigal Koral Kordova
- **The Digital Transformation and its Impact on the Working Models of the Systems Engineers and Analysts [Abstract]** (pp. 47-48)
  Raz Heiferman

www.iltam.org

INCOSE IL האיגוד הישראלי להנדסת מערכות

ILTAM USER'S ASSOCIATION

Technion Israel Institute of Technology

BMGC Bernard M. Gordon Center

The Voice of the Systems

The Journal Of The Israeli Systems Engineers

ISSUE NO.22 | June 2018

INCOSE_IL - THE ISRAELI SOCIETY FOR SYSTEMS ENGINEERING
ILTAM - Users' Association Of Advanced Development Of Complex Systems And SoS

Address: 29 Hamered st. Tel-Aviv, Israel 6812511

Tel: +972-3-6889220 • **Fax:** +972-3-6889216

Email: iltam@iltam.org • **Site:** www.iltam.org

FOLLOW US on : facebook.com/iltamorg

GORDON CENTER FOR SYSTEMS EMGINEERING
TECHNION - ISRAEL INSTITURE OF TECHNOLOGY
www.gordon-se.technion.ac.il

Graphic design: **Ezra Siton**

Table of Content

The Voice of the System | Issue No.22 | June 2018

INTRODUCTION

The Voice of the Editor **4**
Amir Tomer

The Voice of INCOSE_IL President **5**
Ram Oron

ARTICLES

Software & Systems Engineering Interplay and the SEMAT Kernel **6**
Marcel Jacques Simonette and Edison Spina

Lateral Thinking: A Creative Method Example **21**
Avner Engel

Four Layers Approach for Developing System Thinking Assessment Tool **35**
Moti Frank and Sigal Koral Kordova

The Digital Transformation and its Impact on the Working Models of the Systems Engineers and Analysts [Abstract] **47**
Raz Heiferman

JOINING AND INFORMATION

INCOSE_IL 2019 Conference - Save the Date **49**

The program of the Systems Engineering Seminar at the Gordon Center, 19/6/2018 ..... **50**

The Hebrew section is on the opposite side of the journal
חלקו העברי של הגליון נמצא מציידו השני

Software & Systems Engineering Interplay and the SEMAT Kernel

Marcel Jacques Simontette and Edison Spina

Universidade de São Paulo, São Paulo, Brazil

marceljs@usp.br • spina@usp.br

This article is reprinted from:
RECENT ADVANCES in INFORMATION SCIENCE, Proceedings of the 4th European Conference of Computer Science (ECCS '13), (Ed. Maurice Margenstern, Kleanthis Psarris, Danimir Mandic), WSEAS Press, 2013, pp. 267-271.

Abstract

Software systems development teams, using agile practices, have a bias to focus on simpler solutions in early iterations and increments of a software system development process. Usually, requirements related to the system as a whole, as dependability, are considered requirements that are treated in the last iterations cycles. Without Systems Engineering concepts, these teams discovery too late that there is not refactoring to make up the initials choices. The interplay between Software Engineering and Systems Engineering must be increasingly present in organizations that develop software systems. System Engineering approach helps the software development team to coordinate their work processes and steer the team to the development of software systems with quality and value to the business that demands the system. This paper is about the Endeavor dimension of the SEMAT Essence kernel, and how we can work with it to reach the interplay between Systems and Software Engineering.

Keywords: Software engineering • System engineering • Essence kernel • Software process • Socio-technical, complexity

1. Introduction

There are several organizations in which the development of software systems is not the primary business; nonetheless, these companies have a department or area that works with Information Technology (IT). When the IT department of these organizations receives a demand from some other organization business area, a team is formed to perform the software system development (SSD) endeavor. However, the IT department of this kind of organizations was not created to innovate, to create innovative software systems for each business demand; they were created to be efficient [1]. Moreover, it is a commonplace the fact that these IT departments also have to deal with demands that are related to others software systems that are already in operation, and also with business pressures to reduce costs and delivery times.

When the IT department defines a team to the SSD endeavor, it is necessary to define a way of work that allows the team members integration and cooperation. A way of work that steers the understanding of what must be implemented, the implementation itself, and the validation of the system that is delivered. The team coordination must be done with a focus on the delivery of a software system that achieves both the customer needs and expectations. This kind of mantra of SSD management can be reached by way of work that promotes a unique understanding of what must be delivered, considering both the functional and non-functional aspects of the business demand. Furthermore, the way of work must have a quick and effective reaction to the changes that are imposed to the software systems requirements; changes originated by market forces that act upon the organization business.

Some of the factors that characterize the differences between organizations of the same business sector are the qualities of their software systems. This characteristic requires that IT departments seek innovation and flexibility of their processes and software systems, which brings complexity to both the software system as the development process itself.

Systems Engineering offer practices that can enable the integration and collaboration among IT team members, which enable the development of a

single vision of the different parts of a system to be developed [2]. It is the vision of the software system as a whole that promotes the identification of the system dependability issues [3], [4] at the beginning of iterations of the work of the SSD endeavor.

Work, Team, and Way of Work are present in one of the three areas of concern that were identified in the SEMAT kernel. They take part of the things that we always deal with in Software Engineering. They are the alphas of the SEMAT kernel Endeavor context [5].

SEMAT kernel is the first answer to a "Call for Action" [6], a little step in the process of redefining the Software Engineering. This paper is about Systems Engineering through the life cycle of software systems, supporting Software Engineering practices with a socio-technical approach that promotes the integration of the people that take part of an SSD endeavor, looking for a knowledge construction about the system as a whole, since the early cycles of system integration. Barry Boehm has also highlighted the importance of Systems and Software Engineering interplay in several articles, as [7], [8] & [9]. Boehm argues that without up-front Systems Engineering and teambuilding, two common failure modes occur: a leadership scarcity of the agile team, and the bias to focus on easiest-first solutions in the early iterations; an approach to treat dependability issues, like scalability and security, as features to be incorporated in later iterations, when no amount of refactoring will compensate the early choices.

This approach contributes to enabling IT team to develop software systems that add value to the organizations, contributing to organization performance and efficiency.

2. SEMAT Kernel - The Essence

An Opportunity identified by a business area of an organization is a search for a solution to the IT department of this organization. A solution in which the people that work in the SSD process make use of Software Engineering and System Engineering practices to understand what must be done, validate this understanding, and implementing the solution. Although there are several practices to conduct this process, there is a set of essentials elements that are universal and are present in all the endeavors to develop high-quality software systems. These elements - called alphas in SEMAT kernel - deal with "things we always work with" and "things we always do" in an SSD process. These elements provide an integrated set that enables the predictability of the delivery, and the continuous improvement in the way of work, communication, and understanding of what, and how, the team is working in a moment, and what they must to do in the following project increments [5].

SEMAT kernel does not compete with existing SSD methods. It is not a new methodology to SSD. It is an answer to the need of redefining Software Engineering [10]. Essence is a kernel of universal elements that are both present in several SSD methods and that enable the measurement of the project progress and health. The kernel was first published in the SEMAT submission to OMG call: Foundation for Agile Creation and Enactment of Software Engineering [11]. **Figure 1** and **figure 2** represent the "things we always work with" and "things we always do" in an SSD process. More than a conceptual model, the kernel provides:

I. A framework for SSD teams to think about the endeavor progress and the state of their efforts.

II. A common basis to discuss, improve, compare, and share Software Engineering best practices and methods.

III. A framework to SSD teams assemble practices from different origins, continuously improving the way of work.

IV. A framework for defining metrics that are independent of practices, to evaluate both the quality of developed software and the methods used to develop it.

V. And the most important, a way to help SSD teams to understand where they are, what they should do next, and where they need to improve.

2.1. Endeavor alphas

Alphas are representations of the essential elements that must be monitored to guide the success of SSD process. Each alpha has a set of states, and each state has a checklist that identifies whether the state has been met. Alphas must be seen together, not in an approach that individually considers each alpha [5].

The kernel alphas that belong to the Endeavor area of concern are Work, Way of Work, and Team. Observing the Endeavor kernel dimension at **Figure** 1, Team performs and plan the Work, and that the Team applies Way of Work, which guides the Work (Fig. 3).

2.2. Endeavor kernel dimension and the inherent software complexity

IT team members are professionals that are under pressure both to be efficient and to produce software systems that meet stakeholders need; they must address the identified Opportunity. There are different ways of work that these professionals can use to fulfill their mission successfully. However, in this endeavor, they must not forget that the Work to be performed depends on understanding the requirements of the problematic situation that appears as an Opportunity.Whichever the Way of Work applied by the Team, there is an issue present in all SSD endeavors: the identification of the problematic situation, the Opportunity, and the Requirements that such problematic situation brings. This identification process is primarily an activity related to people talking to people [12], [13], [14].

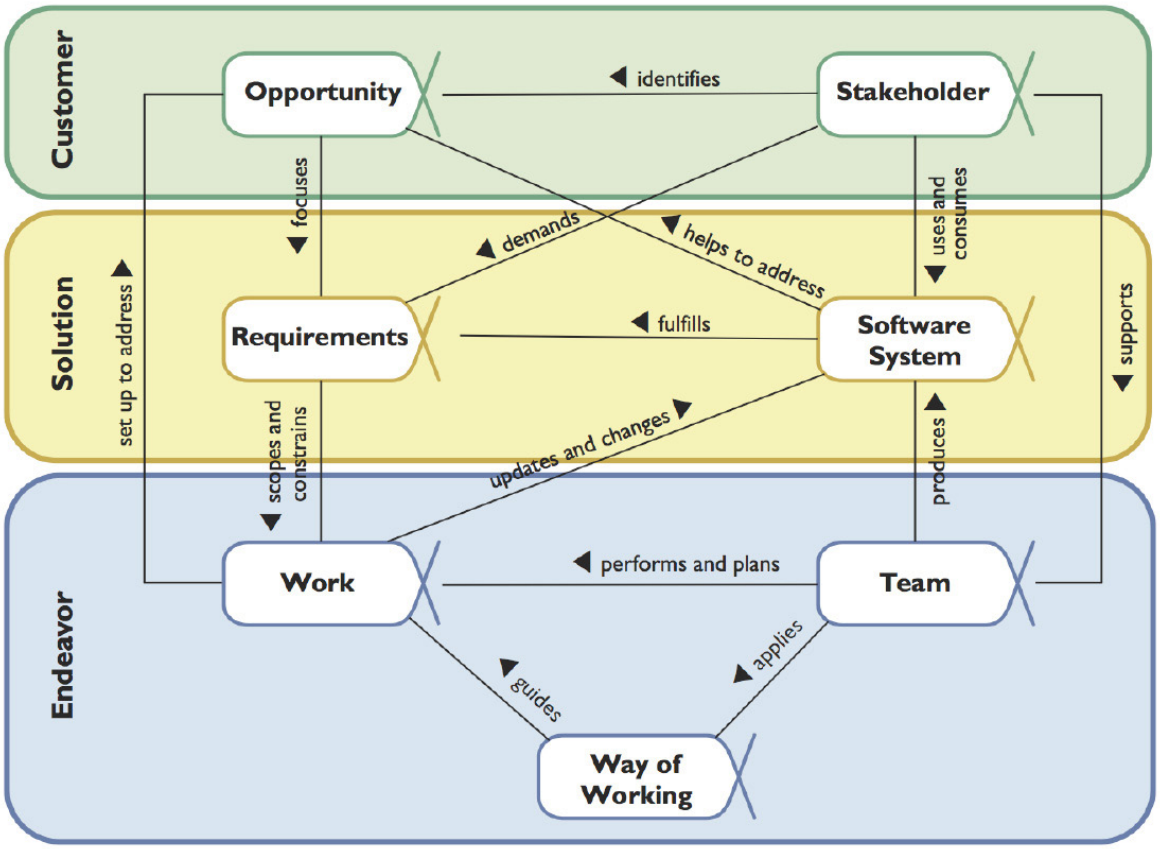

Figure 1. "Things we always work with" in SSD process.

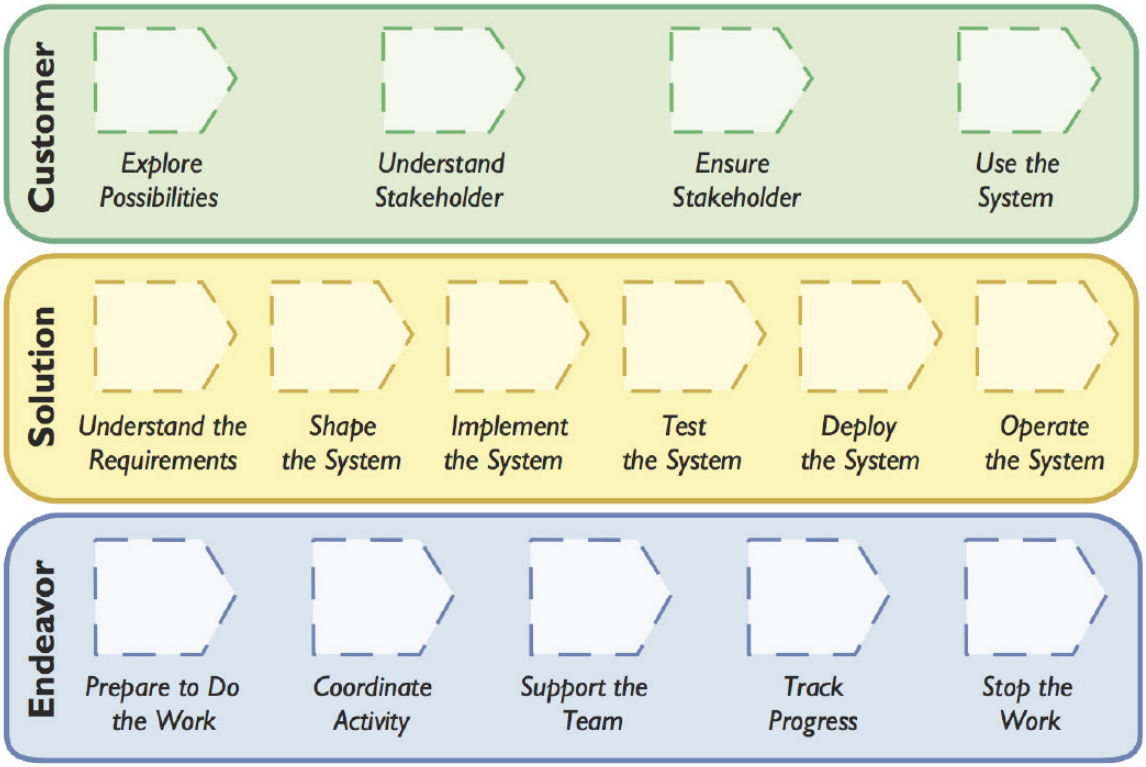

Figure 2. "Things we always do" in an SSD process.

Team member's deal with information and knowledge obtained through interactions with Stakeholders to understand the Opportunity and Requirements. In this process, and especially in the implementation of the software system, these team members need to work in a concentrated way to deal with the underlying complexity of the SSD process [15], [16]. This complexity has also been highlighted by Frederick Brooks in his classic 1987 article "No Silver Bullet: Essences and Accidents of Software Engineering" [17], in which he points out that due to a large number of distinct elements that are present in the SSD process, the development of this type of system is an activity that is more complex than any other type of man-made construction. As evidence of the number of elements that exist in the SSD process, Clarke & O`Connor [18] argue that there are 40 factors and 170 sub-factors that are present in SSD context.

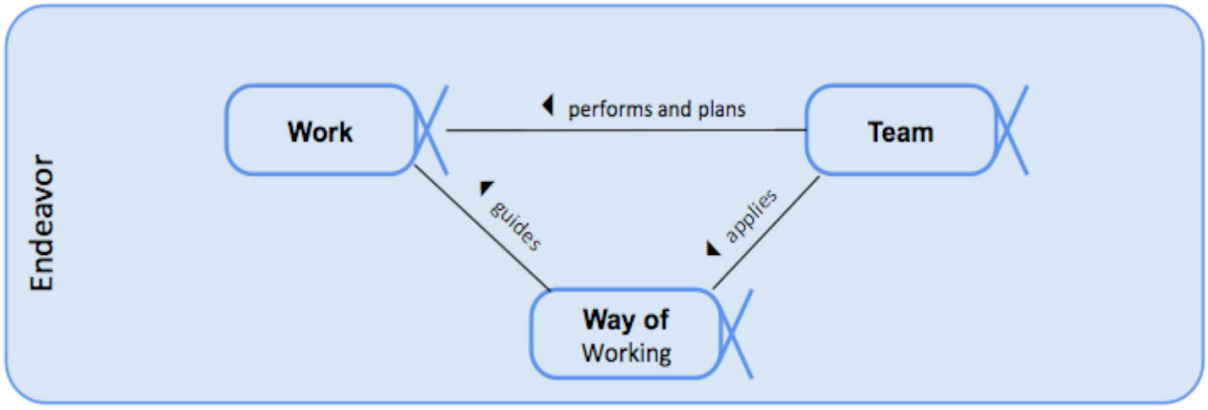

Figure 3: The alphas of the kernel Endeavor area of concern and their interrelationships.

2.3. Endeavor dimension and its relationships

Figure 4 shows all the relationships of the SEMAT kernel alphas that belongs to the Endeavor area of concern:

I. Work has relationships with Requirements - Opportunity - Software System.

II. Team has relationships with Software System - Stakeholder.

III. Way of work has relationships only with alphas of the Endeavor dimension.

Nevertheless, it is an essential link between Team and Work and is under the direct influence of the factors mentioned by Clarke & O `Connor [18].

Despite the apparent simplicity of the relations between the alphas shown in **Figure 4**, there is a challenge in these relations. A challenge related to the cooperation among Team members, and between these people and stakeholders. This relationship is essential to IT team realize a work that adds value to the organization.

3. Systems Engineering

The Cartesian way of dealing with a problem is to divide it into smaller and simpler parts, as much as possible; it is the most successful technique used by Engineering, and this approach enables humanity to reach the current state of technology. However, even with the successes of this approach, there are problems in which this approach cannot succeed; System Engineering uses System Thinking to deal with these problems. Hitchins [2] argues that System Thinking caught the attention of engineers when they realized that the Cartesian approach has difficulties to deal with systems that include people. According to Peter Senge [19], Systems Thinking is a discipline to see a system in its totality, a kind of framework to view the system components interrelationship more than to view the components, to view the system patterns of change more than to view system static images.

Hitchins [2] states that Systems Engineering uses Systems Thinking to understand the nature of the problems. This approach looks for practical experiences and interactions with the problem, trying to understand it, and proposes solutions that may not solve it, but that can provide a response that improves the problem understanding, allowing the development of a solution which is the best at the moment.

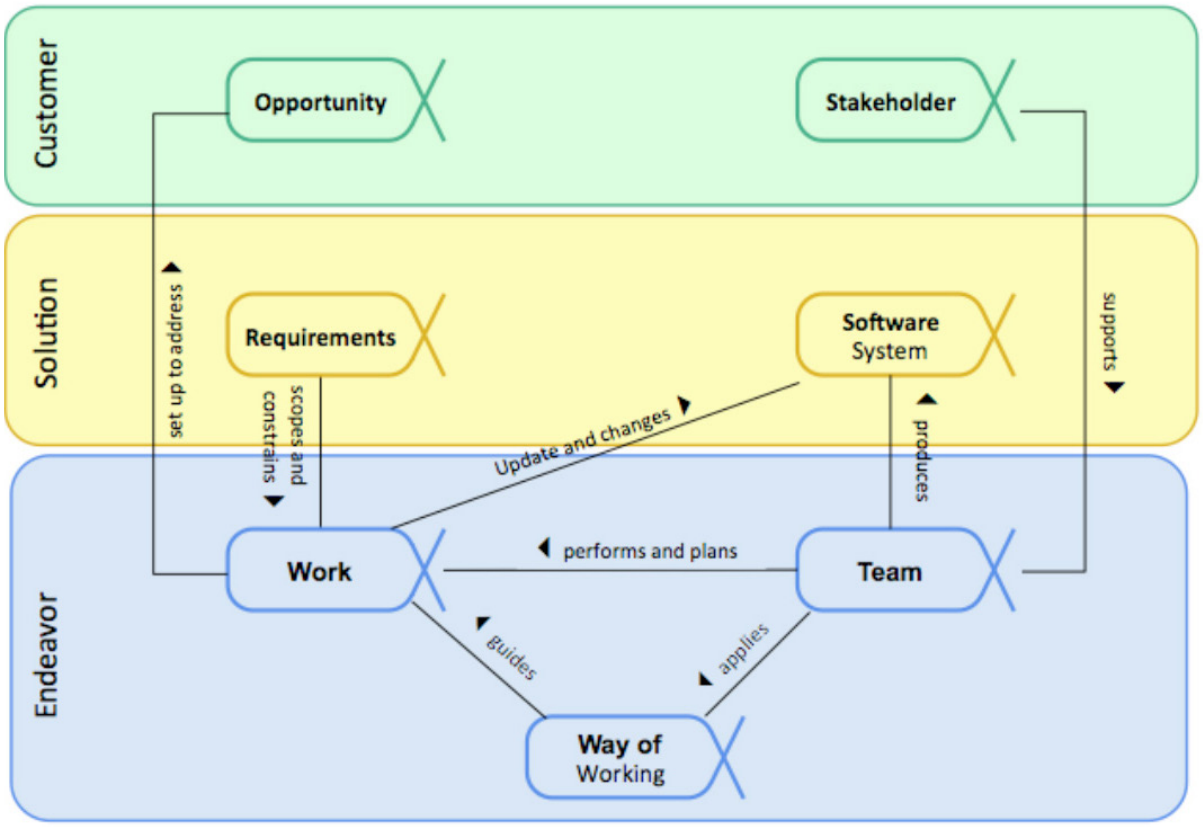

Figure. 4 - Alphas of kernel Endeavor dimension and their relationship with other kernel alphas.

Using this Systems Engineering approach, that is more qualitative than quantitative, and that make use of experience, IT team members can develop a Way of Work that enables them to deal with SSD problems. To implement thi1s.7 approach IT team can use the Lean Startup cycle [20] in the contrary course:

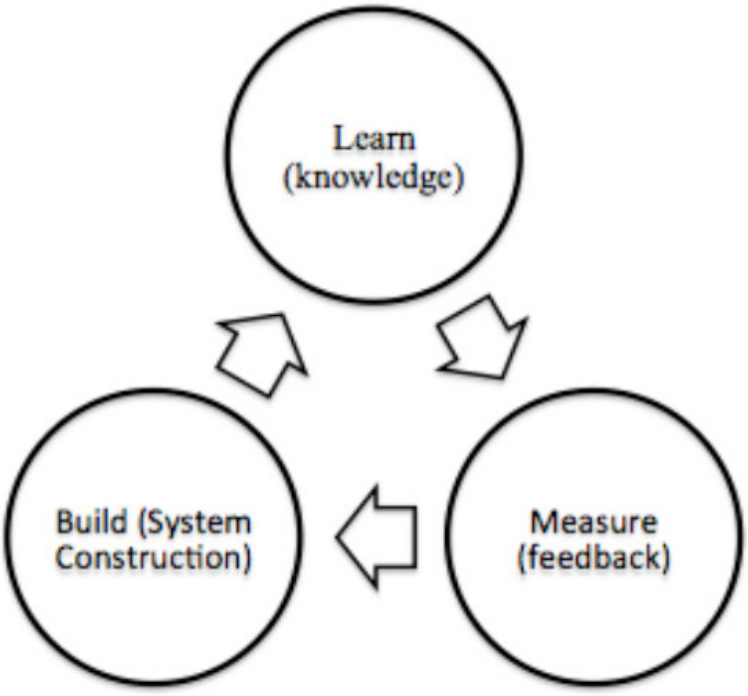

Figure 5 - The Lean Startup Cycle on the Contrary Course

The construction of a minimal version of the software system is the implementation of a response to an Opportunity that is the best possible solution at the moment. The relationship between the Team and the Stakeholders to build this minimal version of the software system leads to an evolution of the knowledge about the Opportunity, a learning experience that allows the evaluation and evolution of the software system, and a new version of the software system can be developed. This approach gives rise to an evolutionary spiral of knowledge and systems in which each turn adds more value to the organization.

The complexity of people interactions in these cycles of Learn (knowledge) -> Measure (feedback) -> Build (System Construction), and the technology used by this people, make up a kind of patchwork of technology and people that give rise to a Sociotechnical System [21] that kernel Endeavor dimension must take into account.

Usually, the Team applies a Way of Work that focuses on developing the technical part of the software system, concentrating on the software issue, and the Team assumes that the requirements reflect the business need. This technical approach results in a software system that works as the Team thinks

that it must work, and that do not take into account the stakeholders culture and experience.

4. The Lean Startup Cycle on the Contrary Course

When following an approach like the evolutionary spiral of knowledge developed by doing the Lean Startup cycle on the contrary course (Fig. 5), Team members have the opportunity to use System Thinking framework of System Engineering to learn through the knowledge and experience of the stakeholders since the early iterations. This approach makes it possible to have, in the sequence of iterations, software system versions more effective and resilient. At each cycle, since the early ones, the Team can incorporate more treatments for dependability requirements, which arises from the interrelationships that exist not only among the technical components of the software system but also between these components and the people that interact with the software system.

5. Conclusion and Future Work

It is impossible to predict the dependability issues that a software system will demand. It is not possible to leave to the last iterations cycles the evaluation of these issues; when the Team technical debt [22] about dependability issues is already at a point in which the refactoring is not possible to be done to meet the identified need [7].

The different fashions of Software Engineering methods and practices do not deal effectively with the inherent complexity of SSD because they are based on code development, without considering the inter-relationship between:

I. Technical components (hardware & software).

II. Issues related to the organization or the business process.

III. Issues related to the relationship between the team members.

IV. Issues related to the relationship between team members and stakeholders.

V. The real intentions of the users using the software system.

The effective interplay between Systems Engineering and Software Engineering is a critical condition to enable IT department to add value to the organization business, and no longer be seen as a cost of operation for organizations in which the software development is not the core business. The System Engineering systemic approach enables the IT team to develop an understanding of what has to be developed, considering the various inter-relationships, and inspire a socio-technical environment in which the IT team objective is more than build a software system, is to add value to the organization business.

The authors of this paper are conducting researches about extensions both to SEMAT kernel alphas and to the alpha states checklists, to offer an answer that considers the inherent complexity of SSD process, in which dependability has a strong presence. ■

References:

1. Govindarajan, V., Trimble, C.: The Other Side of Innovation - Solving the Execution Challenge. Harvard Business School Publishing, Boston, 2010.
2. Hitchins D.K.: Systems Engineering: A 21st Century Systems Methodology. John Wiley & Sons, Chichester, 2008.
3. Laprie, J.C., Dependable computing AND Fault tolerance. Concepts and terminology. In: Proceedings of 15th IEEE International Symposium on Fault- Tolerant Computing. IEEE, Ann Arbor, MI, pp. 2-11, 1985.
4. Avizienis, A., Laprie, J.-C., Randell, B., Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing 1 (1), 11-33, 2004.
5. Jacobson I., Ng, P.-W., McMahon, P. E., Spence, I., Lidman, S.: The Essence of Software Engineering - Applying the SEMAT Kernel. Addison-Wesley Professional, New Jersey (2013)
6. Software Engineering Method and Theory - SEMAT - Call for Action, http://semat.org/?page_id=2
7. Boehm, B., Some Future Trends and Implications for Systems and Software Engineering Processes, Systems Engineering, vol. 9, No. 1, 2006, pp 1-19.
8. Boehm, B., A view of 20th and 21st century software engineering. In Proceedings of the 28th international conference on Software engineering (ICSE '06). ACM, New York, NY, USA, 12-29, 2006.
9. Boehm, B., Some Future Software Engineering Opportunities and Challenges, in Nanz, S. (ed.), The future of software engineering. Heidelberg Berlin: Springer, 2011.
10. Jacobson, I., Ng P.W., McMahon, P., Spence, I., & Lidman, S. (2012). The Essence of Software Engineering: The SEMAT Kernel. Queue 10, 10, pages 40 (October 2012), 12 pages. doi: 10.1145/2381996.2389616.
11. OMG - Foundation for Agile Creation and Encactment of Software Engineering RFP, http://www.omg.org/cgi-bin/doc?ad/2011-6-26

12. Gause, D.C., Weinberg, G.M.,: Exploring Requirements: Quality Before Design. Dorset House Publishing Co., New York (1989)

13. Holtzblatt, K., Beyer, H.R.: Requirements Gathering: The Human Factor. Communications of the ACM. vol. 38, no. 5, 30-32, (1995)

14. Maiden, N.: Trust Me, I'm an Analyst. IEEE Software. vol. 27, no. 1, 46-47 (2010)

15. Demarco, T., Lister, T.R.: Peopleware: productive projects and teams. 2nd Ed. Dorset House Pub. Co, New York (1999)

16. Poppendieck, M., Poppendieck, T.: Lean software development: an agile toolkit. Addison-Wesley, New Jersey (2003)

17. Brooks, F.P.: No Silver Bullet: Essences and Accidents of Software Engineering. IEEE Computer. v. 20, n. 4, 10-19 (1987)

18. Clarke, P., O'Connor, R.V.: The situational factors that affect the software development process: Towards a comprehensive reference framework. Information and Software Technology. vol. 54, no. 5, 433-447 (2012)

19. Senge, P.M.: The Fifth Discipline: The Art & Practice of the Learning Organization. Currency Doubleday, New York (2006)

20. Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Kindle Edition: Crown Business (2011)

21. Baxter, G., Sommerville, I.: Socio-technical Systems: From Design Methods to Systems Engineering. Interacting with Computers. vol. 23, 1, 4-17 (2011)

22. Kruchten, P., Nord, R.L., Ozkaya, I.: Technical Debt: From Metaphor to Theory and Practice. IEEE Software. vol.29, 6, 18-21 (2012)ה

About the authors

Marcel J. Simonette received the B.S. degree in 1991, the M.Sc. degree in 2010, and the Ph.D. degree in 2015. All titles were obtained in Escola Politécnica (Engineering School) of University of São Paulo (EP-USP), São Paulo, Brazil. Currently, he is a researcher at Knowledge Engineering Laboratory (KNOMA) - Department of Computer and Digital Systems Engineering (PCS) of EP-USP. Also, he is a collaborator on Study Center of Society and Technology (CEST- Centro de Estudos Sociedade e Tecnologia in Portuguese). He has experience with European Commission Projects experience as a team member in BELIEF (FP6), BELIEF 2 (FP7), and VertbrALCUE (Alfa3). Dr. Simonette research areas are Software Engineering, Requirements Engineering, System Engineering, and Sociotechnical Systems.

Edison Spina received the Ph.D. degree in Electronic Engineering at Escola Politécnica (Engineering School) of University of São Paulo (EP-USP). M.Sc. degree in 1990 and B.S. degree in 1981, both from EP-USP. He is Assistant Professor - Department of Computer Engineering and Digital Systems at EP-USP - since 1988. He is Freelancer Advisor in Electromagnetic Interference and Compatibility since 1987 and Auditor Leader ISO 9000 since 1995. From 1986 to 2000 he was Director of ANTARES ELETRÔNICA. Also, he was Project Manager at the Foundation for Engineering Technological Development - FDTE, and Electronic Systems Safety Analyst at LCA and GAS labs of the Department of Computer Engineering and Digital Systems at EP-USP. He has European Commission Projects experience as Task Leader in INSTINC Project (FP6), and as Brazilian team coordinator for BELIEF (FP6), BELIEF 2 (FP7), VertbrALCUE (Alfa3), eMundus, and SMART2 (Erasmus Mundus).

Prof. Dr. Spina is member of the KNOMA - Laboratory of Knowledge Engineering, Academic Coordinator of the Study Center Society and Technology of USP, Member of the Committee for International Relations of the EP-USP, and Council Member of the Committee of Sciences and Technology of OAB (Brazilian Bar Association).