*Araujo, A. R. S.; Moura, H. P.; França, C.*
Revista Gestão.Org, v. 13, Edição Especial, 2015. p. 445-455
ISSN 1679-1827
http://www.revista.ufpe.br/gestaoorg

# Complexity Within Software Development Projects: an Exploratory Overview

## Complexidade em Projetos de Desenvolvimento de Software: uma Visão Geral Exploratória

Allan R. S. Araujo[1], Hermano Perrelli de Moura[1], César França[1]

[1]Universidade Federal de Pernambuco, UFPE, UFS, Brasil

Correspondência: Allan R. S. Araujo. Endereço: Av. Prof. Moraes Rego, s/n, Cidade Universitária. CEP.: 50.670-901 Recife, Brasil. Tel.: 55 81 2126-8000 E-mail: arsa@cin.ufpe.br

Recebido: 14 de outubro de 2015 Aceito: 26 de março de 2016 Publicado: 09 de maio de 2016

**Abstract**

Currently, success is hard to achieve in software development (SD) projects. Studies have reported in which only 35% of projects might be considered as successful. In the same time, projects have been progressively more complex, what makes it even more challenging to succeed in such projects. People, in general, do not properly understand complexity phenomena besides the common sense of something is difficulty to achieve or resolve. Hence, in order to being able to cope with complexity, people need – beforehand – understand properly what it is. Therefore, this study claims that both – practitioners and researchers – should know theories, concepts and definitions for assessing and reflecting about complexity and taking decision in such environments. In addition, it might be important and useful know models and approaches for managing complex projects and navigate on complex situations. In fact, this research is an extensive and exploratory literature review based on works found in, both academic and industrial, search engines from 1990 until 2015. Then, it was necessary to identify the most remarkable ones according to the purpose of this paper by reading and categorizing them. The finding included an inclusive (not exclusive) definition of complexity/complex projects. Moreover, it was found some models and approach for supporting managers and team members working in complex project. The main limitations related to the study include the absence of a formal approach for performing the literature review (e.g. Systematic Literature Review); and the lack of empirical evidence (e.g. case studies, focus group, survey, etc.) obtained from real-world projects in order to increase reliability and robustness. These limitations open a field for future works and studies both in academia and in industry.

**Keywords:** Complexity; Project Management; Complex Projects; Information Technology; Software Development

**Resumo**

Atualmente, os índices de insucesso em projetos de software são alarmantes. Estudos mostram que apenas cerca de 35% dos projetos podem ser considerados como sucesso. Ao mesmo tempo, há uma percepção que os projetos têm se tornado gradativamente mais complexos, o que torna ainda mais desafiador obter sucesso em projetos desta natureza (complexidade). Pessoas, em geral, não entendem apropriadamente o fenômeno da complexidade além do senso comum de que pode ser algo difícil de alcançar ou resolver. Logo, visando a se tornarem aptas a lidar com a complexidade, pessoas precisam – de antemão – entender apropriadamente o que significa. Portanto, este estudo argumenta que ambos – praticantes e pesquisadores – devem conhecer teorias, conceitos e definições para identificar e refletir a respeito da complexidade para que sejam capazes de tomar decisões nestes ambientes. Além disso, é importante e útil conhecer modelos e abordagens que possam ajudar pessoas a gerenciarem projetos complexos e navegarem por situações complexas. De fato, esta pesquisa se trata de uma extensiva revisão bibliográfica exploratória (ad hoc) baseada em trabalhos encontradas em mecanismos de busca (da academia e da indústria) considerando o período de 1990 a 2015. Depois, foi necessário identificar e selecionar os trabalhos mais alinhados a este estudo através da leitura e categorização dos mesmos. Os resultados foram inclusivos (não excludentes) mostrando que as definições de complexidade e de projetos complexos podem ser combinadas. Adicionalmente, foram encontrados alguns modelos e abordagem para sustentar participantes de projetos complexos. As maiores limitações deste trabalho se encontram na falta de uma abordagem sistemática para realização da revisão bibliográfica (ex. Revisão Sistemática ou Mapeamento Sistemático); e na falta de evidências empíricas (Estudos de Caso, Grupo Focal, Enquetes) obtidas através de projetos reais para fortalecer a confiabilidade e robustez do estudo. Estas limitações abrem um vasto conjunto de possibilidades para trabalhos futuros tanto na academia quanto na indústria.

**Palavras-chave:** Complexidade; Gerenciamento de Projetos; Projetos Complexos; Tecnologia da Informação; Desenvolvimento de Software.



## 1. Introduction

Nowadays people use to claim that their projects are complex. In fact, since the World War II project are becoming progressively more complex and so on (Baccarini, 1996, p.201). In order to confirm or refuse this assumption we should rely on a broadly accepted definition for what complex or complexity is. Currently, although there are multiple definitions available, there is no single standard definition for what complexity is from the perspective of software development (SD) projects. Notice that this study has a particular interest in researching aspects of complexity within SD projects, so terms such as information technology (IT), information technology and communication (ICT), and information systems development (ISD) are taken as interchangeable for this means.

According to Greenfield (2005), it was expected that:

> "Total global demand for software will grow by an order of magnitude over the next decade, driven by new forces in the global economy like the growing role of software in social infrastructure, by new application types like business integration and medical informatics, and by new platform technologies like web services, mobile devices and smart appliances".

These trends have been confirmed over the last years. Meanwhile, failure has been recurrent in that industry. Standish Group`s Chaos Report shows that only 16-37% of all projects from 1994 to 2010 were considered successful (Standish Group, 2010). In order to back up for the increasing demand for software-intensive products in general, it is important to continuously enhance our ability to successfully manage and develop increasingly complex software development projects. In this context, this paper intends to (1) review current theories, concepts and definitions about complex projects; and (2) present the existing models and approaches for managing complex projects.

Complexity studies are spread though many different fields and theories (Cooke-Davies et. al., 2007, p. 50-55):

- Life Science: Butterfly Effect (Edward Lorenz); Self-organization, Evolution and Complexity (Stuart Kauffman); Edge of Chaos (Chris Langton); Emergence (Chris Langton); Complex Responsive Processes Relating (Ralph Stacey).
- Physical Sciences: Universality (Mitch Feigenbaum); Dissipative Structures (Ilya Prigogine); Self-Organizing Systems (Herman Hacken).
- Mathematics: Strange Attractors (David Ruelle); Fractais (Bernoit Mandelbrot); Patterns and Patterning (Ian Stewart); Complex Adaptive Systems (John Holland).

It is important to warn that common distinction is made between product and project complexity. Although complexity of a product is generally associated with complexity in its development project (Williams, 1999), the main concern of this research is figuring out the aspects of project complexity (e.g. organizational, people, processes, etc.). Therefore, detailed discussions about product (e.g. software elements such as programming language, algorithms, etc.) complexity are out of scope.

This review analyzes both general and specific issues, i.e. related to the SD context. General issues are important to provide a wider and holistic view about reality. On the other hand, specific issues are essential for covering useful information about this particular context. In the Background section, basic definition of projects are provided, and the SD context is outlined. Methods section details the procedures followed in our exploratory literature review. Finally, in section Findings, we present the relevant concepts and definitions found for complexity in projects, as well as models and approaches for managing complex projects. In section Conclusion, we discuss some of the limitations for this work, and present opportunities for evolve and deepen this research

## 2. Background: The Context of Software Development Projects

According to the PMI a project is "a temporary endeavor undertaken to create a unique product, service, or result" (Project Management Institute, 2014, p.1). Considering the temporary nature indicates a project has a beginning and an end (timebox). In this context, XXI century projects are "about adding value to the organization through the implementation of breakingthrough ideas, optimizing business processes, and using ICT (information and communication technologies) as a competitive advantage" (Hass, 2008, l.439). ICT products and service are usually associated to software development projects. These endeavors may involve merge or acquisition, urgency (due to the reduced time-to-market), emergence of new technologies or methods, global competition and execution. Specifically, software development projects may include (Hass, 2008, l.294):

- Large, multilayered, geographically dispersed, and multicultural teams.
- Aggressive schedule and inflexible budget.
- Unstable, ambiguous and poorly defined requirements.
- High visibility, political charge and conflicting expectations.

Thus, software in contemporary businesses is considered as a critical success factor for competing in the global economy, which assigns strategic importance of software development projects, turning both industry and academy attentive to research and studies aiming to improve our general ability to develop reliable, valuable and profitable software projects. Software has been written since 1940 decade. The term "software engineering" came about during the 1969 NATO Software Engineering Conference, which is considered the official start of that activity as an specific profession (Naur and Randell, 1969). From that moment, much investment has been done on elaborating:

- Software development lifecycles, such as Waterfall and Iterative and Incremental, etc. (Sommmerville, 2010).
- Software maturity models, such as CMM, CMMI, MPS.BR, etc. (Crissis et. al., 2011).
- Software processes and methods, including Unified Process (Krutchen, 2003) and agile methodologies (e.g. Scrum, Extreme Programming, DSDM, FDD, Crystal) (Griffiths, 2012, p.21-53).
- Software development techniques, such as object-oriented programming and analysis, UML (Kutchen, 2003), test-driven development (TDD), continuous integration, etc. (Beck, 2004).

In 1994, the Standish Group International published the first edition of the "Chaos" report, revealing dramatic information about failure in software development projects in both private and public sector in United State. At that time, the overall studied population was around 175,000 projects spending more than US$ 250 billion per year. The total sample size was 365 respondents and represented 8,380 software applications (Standish Group, 1994). Projects were categorized and results obtained as following:

- Success: The project is completed on-time and on-budget, with all features and functions as initially specified (16%).
- Challenged: The project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified (53%).
- Failed: The project is cancelled at some point during the development cycle (31%).

Hence, in 1994 only 16% of these projects were considered successful. Later versions of the "Chaos" report have revealed the difficulty to achieve success in IT projects: 1996 (27%), 1998 (26%), 2000 (28%), 2004 (29%), 2006 (35%), 2008 (32%), 2010 (37%), 2012 (39%) (Hass, 2008, l.393). Although a significant evolution has been achieved, mostly because of the multidisciplinary interaction between fields such as software engineering and project management, success rates are still low. Even the largest companies are struggling in producing high quality and successful products and services. According to Hass (2008, l.307), for example, "US companies estimated it was spent US$ 80-145 billion per year on failed and cancelled software-intensive projects" and "in European Union is also estimated around US$ 140 billion per year wasted in failed ICT projects".

According to Griffiths (2012), original software development approaches were adopted from industrial work (e.g. manufacturing) ideas, including Gantt chart, functional decomposition, and localized labor based on the concepts introduced during the Industrial Revolution in 18-19th centuries. However, in 1959, Drucker coined the term "knowledge work" to describe a new and emergent kind of labor - in comparison with traditional industrial work –with emphasis on non-routine problem solving using both convergent and divergent, and creative thinking. Knowledge workers are not only found in SD contexts; they are also engineers, teachers, doctors, scientists, lawyers and others employees. Currently, knowledge workers are the largest segment of US workforce. The two types of workers are contrasted in Table I.

| **Industrial Work** | **Knowledge Work** |
|---|---|
| Work is visible | Work is invisible |
| Work is stable | Work is changing |
| Emphasis is on running things | Emphasis is on changing things |
| More strucuture with fewer decisions | Less structure with more decisions |
| Focus on the right answers | Focus on the right questions |
| Define the task | Understand the task |
| Command and control | Give autonomy |

| Industrial Work | Knowledge Work |
|---|---|
| Strict standards | Continuous innovation |
| Focus on quantity | Focus on quality |
| Measure performance to strict standards | Continuous learn and teach |
| Minimize cost of workers for a task | Treat workers as assets, not costs |

**Table 1**: Industrial x Knowledge Work Characteristics.

**Source**: Griffiths (2012).

According to Griffiths (2012), projects failures are mostly due to the application of industrial work techniques to manage knowledge work activities. Since traditional project management relies on many characteristics of industrial work, researchers and practitioners are realizing that such projects involve more uncertain and unpredictability, they require a more flexible and adaptive approach to project management in order to address their inherent nature (Hass, 2008, l. 456).

**2.1. Complexity Theory**

According to Thomas and Mengel (2008), complexity stands for:

> "A large number of independent agents, each capable of behaving according to unique principles of interaction and relation. Organizations are adaptive in that they do not simply respond to events, but evolve or learn. Each agent is guided by its own schema or rules of behavior and by a scheme shared with other groups".

Such complex systems hold some specific properties (Hass, 2008, l.576), namely:

- Self-organization: There is a continuous reorganization in the system in order to find out the best fit to the environment instead of planning and managing based on prior command and control definition.
- Non-linearity: The interaction between agents and variable within the system is not predictable and may generate unexpected results.
- Emergence: The agents within the system interact in unpredictable ways. By understanding the patterns that emerge through these interactions, it is possible to understand the system itself.
- Interdependency: Most systems are nested within other systems. Therefore, their behavior may not be isolated from external influence.

From that, emerges the concept of "pattern of complexity" which is the minimum manageable context of complexities within a project. It focus on the behavior/pattern of a system instead of its singular characteristics. Hence, it is important to point out that the concept of "pattern of complexity" does not intend to model or explain complexity, but support project practitioners to reflect "holistically" and figure out how to cope with complex situations (Geraldi and Aldbrecht, 2007).

**3. Methods**

This work is part of a wider and deeper PhD research about complexity within SD projects. As a first step, it was intended to establish aspects of relevance, novelty; discuss similar works; and identify opportunities for future works. Having said that, the following steps were performed to conduct this preliminary study:

- Searching on engines such as Science Direct, ACM Digital Library and IEEE Explorer by using the combination of keywords (e.g. "complex", "complexity", "complex projects", "project management", "information technology" and "software development"). The search was limited to papers between 1990 and 2015 and retrieved 70 candidate papers.
- Classification based on the following questions: "Does it provide a conceptual definition about complexity or complex projects"? "Does it provide a model or abstraction for dealing, coping or understanding the phenomena of complexity in projects"? And "Does it discuss complexity/complex projects through an ICT/SD perspective"? Since a huge effort was required to read all retrieved papers, reading was focused on abstract, introduction and conclusion sections. Then, the papers were classified into OK (fully or partially address the questions) or NOK (does not address the questions). Thus, 22 (twenty-two) were selected as they were aligned with the purposes of this research.

Finally, the 22 papers were fully read to extract answers for the questions. The findings are discussed over the following sections. In fact, it was not conducted a systematic literature review (SLR) or mapping (Kitchenham, 2004) due to its required effort and purpose. Instead of providing a complete, consistent and up-to-date overview of state of art, it was intended to figure out preliminary (and up-to-date) evidences about the subject in order to support further studies. Nevertheless, this set of 22 papers may serve as a gold standard (Zhang and Babar, 2010) for a future systematic review.

## 4. Findings

### 4.1. What is a Complex Project?

Currently, there is no standard or broadly accepted definition for complex projects. In general, the term "complex" has several meanings. Oxford dictionary defines it as "consisting of many different and connected parts" or "not easy to analyze or understand; complicated or intricate". The former definition may find its roots in the general system theory (Bertalanffy, 1969), while the latter is more likely raised in practitioners' common sense, because it is commonly used to refer to a project that is big or hard to be accomplished (William, 1999, p.269).

According to Geraldi and Aldbrecht (2007, p.33), technical studies usually define complexity by characteristics such as a the number of elements or variables, heterogeneity of these elements, and the variety of these relationships; variety of goals, perspectives, cultures, etc.; and by its behavior, such as non-linearity, emergence, positive feedback loops, self-regulation, irreversibility, unpredictability. Others prefer defining a complex phenomenon by its difficulty; uncertainty; dynamism; uniqueness; lack of clarity, or low degree of definition of goal, scope and methods.

In summary, on one hand, some authors prefer defining a complex project in terms of its components and characteristics. On other hand, there are researchers and practitioners define a complex system (e.g. software development projects) by its behavior (Geraldi and Aldbrecht, 2007, p.33). These two approaches are consistently reflected when translated to project complexity (Table II). Turner and Cochrane (1993), for example, presents a definition associated with the idea of uncertainty. They classified projects by two parameters: "How well defined the goals are" and "How well defined the methods are". Therefore, they identify four distinct profiles of projects and suggest different management approaches for each of them. On the other hand, Baccarini (1996) proposes a definition of project complexity in terms of differentiation (the number of varied elements) and interdependency (the degree of connectivity among these elements).

Baccarini's measures apply to several project dimensions – such as organizational and technological complexities - as following:

- Organizational – by differentiation
    - Vertical: depth of organizational hierarchical structure, i.e. number of levels.
    - Horizontal:
        - Organizational units: The number of formal organizational units, i.e. departments and groups.
        - Task structure: refers to the division of tasks, including division of labour and personal specialization.
- Organizational – by interdependency: There are three types: pooled (independent), sequential and reciprocal (highest level of complexity).
- Technological – by differentiation: refers to the variety of some aspect of a task, including number and diversity of inputs and/or outputs; Number of separate and different actions or tasks to produce the final product of a project; and number of specialties.
- Technological – by interdependency: It may include dependencies between different technologies, inputs/outputs, etc. There is also three types: pooled, sequential and reciprocal.

| | **Conceptual Approach** | |
|---|---|---|
| | *Uncertainty of the phenomenon* | *Structure of the phenomenon* |
| Complexity | "not easy to analyze or understand; complicated or intricate" (Oxford Dictionary)<br>"difficulty; uncertainty; dynamism; uniqueness; lack of clarity, or low degree of definition of goal, scope and methods" (Geraldi and Aldbrecht, 2007) | "consisting of many different and connected parts" (Oxford Dictionary)<br>"the number of elements or variables, heterogeneity of these elements, and the variety of these relationships" (Geraldi and Aldbrecht, 2007) |
| Project Complexity | "How well defined the goals are" and "How well defined the methods are" (Turner and Cochrane, 1993) | differentiation (the number of varied elements) and interdependency (the degree of connectivity among these elements) (Baccarini, 1996) |

**Table 2**: Summarization of Concepts.

**Source**: Authors (2015).

Following this same rationale, Williams (1999) presented a definition of complexity based on the original works of Baccarini (1996); and Turner and Cochrane (1993), summarizing his idea as pictured in Image 1.

Revista Gestão.Org, v. 13, Edição Especial, 2015. p. 445-455
ISSN 1679-1827
http://www.revista.ufpe.br/gestaoorg

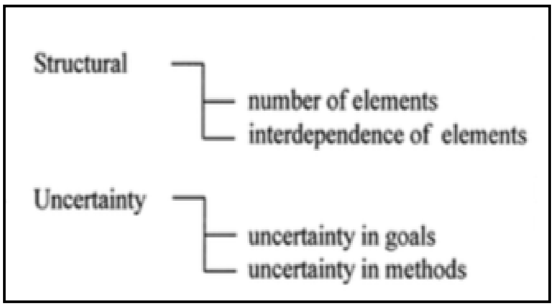


**Image 1**: Complexity Definition.

**Source**: Willliams (1999).

Lately, Geraldi and Aldbrecht (2007) introduced the concepts of complexity of faith, fact and interaction by drawing upon the works of Baccarini (1996) and Turner and Cochrane (1993), as following:

- **Complexity of Faith**: It is similar to the definition proposed by Turner and Cochrane (1993) which is associated with the idea of high uncertainty within a project. Typical instances of this kind of complexity are innovative project in which lack definition of goals (to guide decision-making) and methods (to support management and development). On one hand, simple project are those in which goals (e.g. requirements) and methods (ex. technologies) are well defined. On the other hand, complex projects are those in which both goals and methods are far from certainty. Once there is much uncertainty, traditional approaches that rely on preliminary definitions and steady progress use to fail. Otherwise, adaptive approach (based on trial-and-error and learning by doing) may be necessary to support the dynamic environment. Therefore, scope change and rework are not exception but expected (see Image 2).

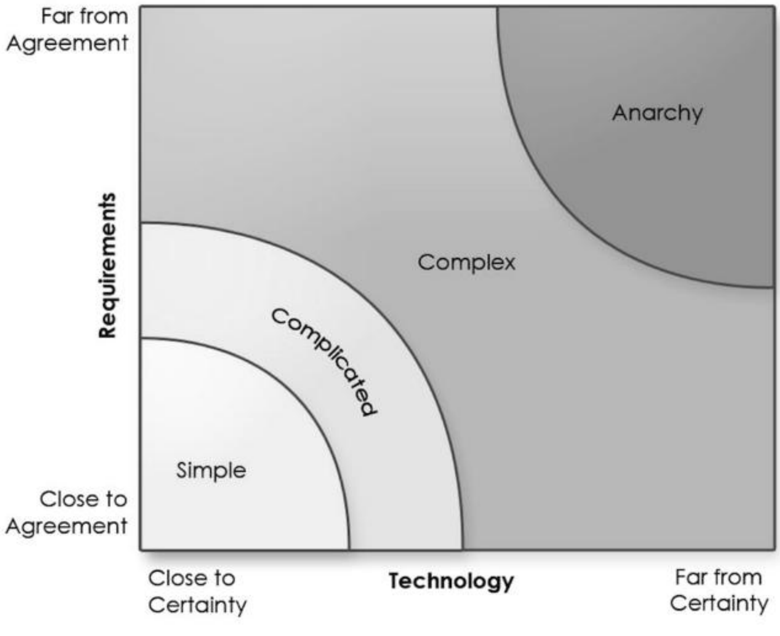


**Image 2**: Complexity Spectrum.

**Source**: Griffiths (2012).

- **Complexity of Fact**: It refers to the kind of complexity related to dealing with a huge amount of interdependent information; and is similar to the concept introduced by Baccarini (1996) – named structural complexity – considering the number of elements (differentiation) and their interdependence within an environment (e.g. ICT and SD project). In this context, practitioners are not able to figure out all elements and their relationship. Hence, decision-making should be performed without understanding all necessary information. Mistakes are not exception but expected. Additionally, during a project complexity of fact becomes higher since the number of elements (and their relationship) increase (Geraldi and Aldbrecht, 2007, p. 36). For instance, an ICT project aimed in developing a software use to begin with a small set of features. As the work progresses the number of features and their coupling increase (complexity of fact).
- **Complexity of Interaction**: Besides complexities of faith and fact, it is possible to identify complexity

of interaction that is concerned with the interfaces between locations, team, organizational units, political groups, etc. This kind of complexity influences both complexity of faith and fact (Geraldi and Aldbrecht, 2007, p. 53). Modern SD projects usually includes large, multilayered, geographically dispersed and multicultural teams; and also high visibility, political charge and conflicting expectations. Thus, these projects use to have high complexity of interaction.

In addition to the definition of these concepts, Geraldi and Aldbrecht`s (2007) work also investigates which projects dimensions relate to each kind of complexity and their intensity on each project phase.

**4.2. Models and Approaches for Complex Projects**

Current sub-section intends to discuss different models and approaches for managing complex projects. Traditional project management approach is (Saynich, 2010, p.5):

> "(...) based mainly on a mechanical, monocausal, nondynamic, linear structure and a discrete view of human nature and societies and their perceptions, knowledge, and actions. It works on the basis of reductionist thinking and on the Cartesian/ Newtonian concept of causality (the mechanistic science). Traditional project management cannot solve these widespread profound challenges (…)".

21st century business - including SD projects - are all complex systems (Hass, 2008, l. 604) and require a management approach that might help teams and organizations to overcome – or at least cope with – complexity. In this context, emerges the concept of sense-making that "represents a viewpoint that sees organizations not as fixed objective entities, clearly delimited by organizational charts and management hierarchies, but as variable and multiple representations of reality" (Táxen and Lillieskold, 2008). Thus, a traditional management approach (one-size-fits-all) is not enough to realm the management of complex projects. Sense-making argues that multiple narratives are necessary to support project management and engage stakeholders.

Xia and Lee (2005, p.56) introduced one of first attempts to create models or approaches for managing complex projects (in SD context). A "Conceptual Framework for Information Systems Development Projects (ISDP) Complexity" was based on several empirical studies (e.g. using focus group, interviews, etc.) and identified two dimension for handling complexity.

The first one is concerned on the structural or dynamic aspect of the project. Structural characteristic refers to the number of components and their interdependence within a system and is in line with the concept presented by Baccarini (1996): as this number increase and these relationships become more coupled, it is harder to predict project results and outcomes.

Dynamic characteristic refers to the complexity related to the changes in the components of a project and is in line with the definition presented by Turner and Cochrane (1993): as goals (e.g. requirements) and methods (e.g. processes and technologies) are far from certainty, changes increase and it is difficult to predict project results and outcomes. The second dimension is concerned on the organizational or technological aspect of the project. Organizational aspects include culture, processes, maturity, top management support, strategy, etc. Technological aspects include hardware, software, network, team knowledge (skills, experience).

| | | |
|---|---|---|
| Organizational | Structural Organizational Complexity (SORG) | Dynamic Organizational Complexity (DORG) |
| Technological | Structural IT Complexity (SIT) | Dynamic IT Complexity (DIT) |
| | Structural | Dynamic |

**Image 3**: Summarization of Conceptual Framework for ISDP Complexity.

**Source**: Xia and Lee (2005).

Moreover, Xia and Lee (2005, p.56-60) identified a set of project characteristics and performed a factor analysis in order to relate each one to the classes of complexity above (SORG, DORG, SIT and DIT).

Then, Geraldi and Aldbrecht (2007, p.36) also presented – besides the definitions of complexity of faith, fact and interaction already discussed in the previous section – an empirical study that led to the following remarkable findings:

- 10 (ten) characteristics of complexity were identified and grouped by their intensity. These characteristics were translated to widely accepted concepts in project management field, composing the "pattern of complexity":

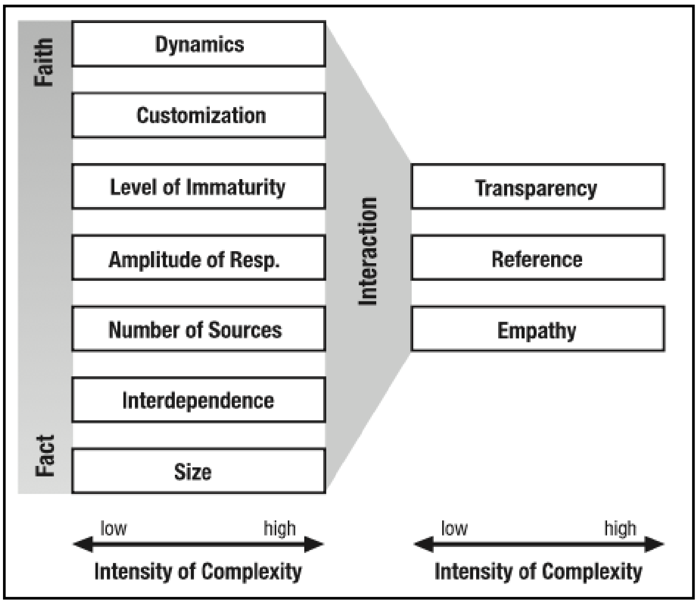


**Image 4**: Characteristics of Complexity (Faith, Fact and Interaction).

**Source**: Geraldi and Aldbretch (2007).

- A questionnaire survey answered by project managers, and analyzed through statistical calculations, pointed out that:
  - Observers/practitioners perceive complexity differently. Considering the complexity of faith, people with more experience within a project domain (goals or methods) will perceive less complexity of faith in comparison with those without experience.
  - Normally, complexity of faith is stronger in first phases of a project and decrease during the project execution. Otherwise, complexity of fact and interaction use to increase as the work progresses.
  - Considering their intensities, it is possible to argue that complexity interaction is stronger than complexity of fact. It is also possible to argue that complexity of faith is weaker than complexity of fact.

Later, Hass (2008) introduced her model for handling project complexity in ICT projects. Under the name of "Project Complexity Model", it offers a framework for identifying and diagnosing the elements of complexity on a particular project so that the project team can make appropriate management decisions.

To asses complexity, several dimensions were identified: time/cost; team size; team composition and performance; urgency and flexibility of cost, time and scope; clarity of problem, opportunity and solution; requirements volatility and risk; strategic importance, political implications and multiple stakeholders; level of organizational change; level of commercial change; risks, dependencies and external constraints; and level of IT complexity. For each of these dimensions, three different complexity categories were identified: independent, moderately complex, and highly complex. In addition, for each dimension and complexity category, it was defined a set of parameters in order to support project classification.

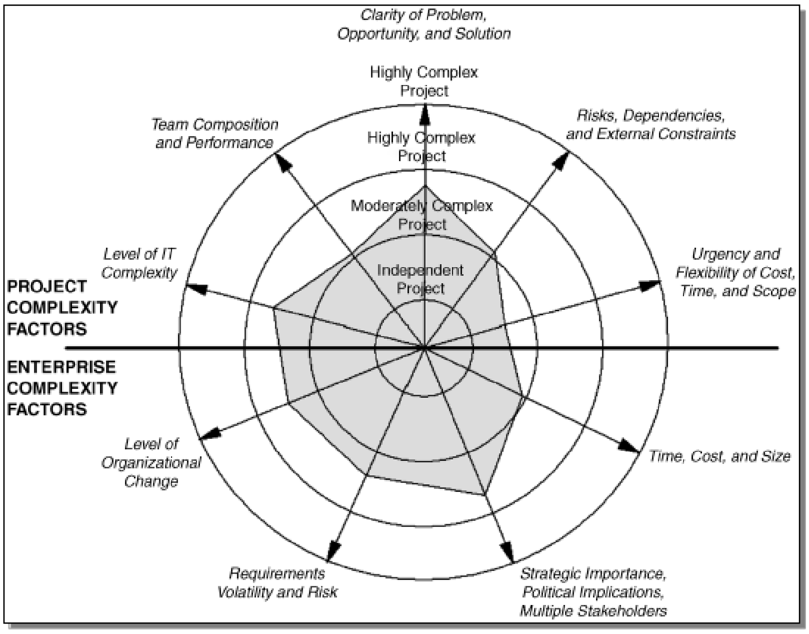


**Image 5**: Project Complexity Model – Diagnosis.

**Source**: Hass (2008).

The "Project Complexity Model" framework presented by Hass allows team members to categorize one project into several complexity dimensions and figure out its overall complexity figure. In addition, it is also possible to identify recommendations for guiding the team for taking better decisions. For instance, when choosing the project phases and life cycle:

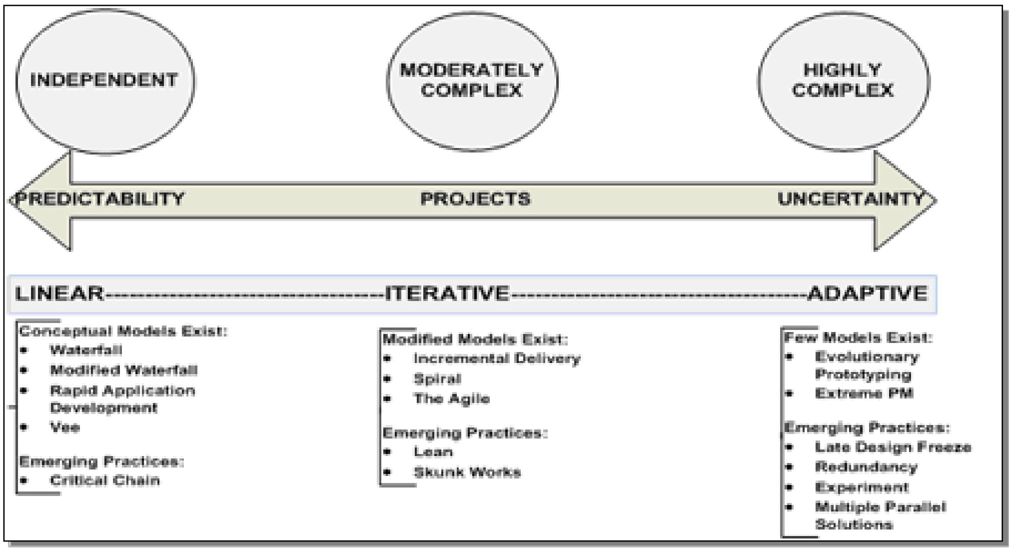


**Image 6**: Project Complexity Model – Recommendation for Life Cycles.

**Source**: Hass (2008).

In recent years, new works have been published trying to propose new approaches for complex projects. Piccinini and Gregory (2013) studied the nature of complexity in project and programs (set of related projects) through a systematic literature review. By discussing aspects of complexity strictly under the perspective of Xia and Lee (2005), they did not proposed a concrete approach (e.g. a framework or guidelines) to support decision-making in complex projects. In the same time, Whitney et. al. (2013) investigated the root cause for failure in ICT projects considering complexity as a viewpoint. Although they argue that new approaches are needed for supporting management of complex projects, their work did not propose any concrete (e.g. a process, framework or guidelines) approach for that. Both works were not as extensive and detailed as the ones proposed by Xia and Lee (2005) and Hass (2008). Further considerations about the finding are discussed in the in the following section.

## 5. Conclusion

Business are complex systems nested within an even larger complex system, namely the global economy (Hass, 2008, l.600) and SD projects are only small instances of those complex systems. Currently, software-intensive products play an important role in society and economy. At the same time, studies about complexity have been brought to the light in project management world, gaining importance both in industry and in academia.

In this context, this study describes an exploratory literature review focused on investigating the more relevant contributions within the field SD projects. First, it was necessary to set up the background related to the SD context and discuss characteristics about software development projects. Then, this paper went deeper on the definitions of what a complex project is, and what complexity within a project is. Thus, several definitions were detailed and discussed:

- Uncertain Complexity, according to the concept presented by Turner and Cochrane (1993);
- Structural Complexity, according to the concept presented by Baccarinni (1996);
- Complexity of Faith, Fact and Interaction, as introduced by Geraldi and Aldbretch (2007).

Currently, there is no broadly accepted or standard definition for what is a complex project, even on restrict contexts such as SD projects. Hence, there are multiple inclusive (not exclusive) definitions. In fact, all these concepts exhibit similarities and may be useful for practitioners for reflection when working in complex projects. This work was not intended to perform comparisons and recommend one chosen definition. Thus, the major contribution achieved through the discussion of concepts is a basis for supporting future studies (for researchers) and pragmatic reflection (for practitioners) within project management world. On the other hand, these works still do not represent a concrete framework, guideline, tool, process or method to support decision-making and application in real-world projects.

Later, the discussions focused on the models and/or approaches for complex projects. Actually, there are few extensive and detailed works in literature for supporting decision-making or a process instantiation for a project, including the works of Xia and Lee (2004 and 2005) and Hass (2008) as the most remarkable ones. From the academic perspective, Xia and Lee (2005) were responsible for the “Conceptual Framework for Information Systems Development Projects (ISDP) Complexity”. Following very reliable methods, Xia and Lee (2005) identified complexity dimensions (structural or dynamic - technological or organizational), characteristics (e.g. size, interdependence, level of maturity, number of sources, etc.) and hypothesis (e.g. managers or team members perceive complexity differently according to their previous experience). Although it delivered relevant contributions to project management science, it was not possible to identify explicit recommendations that might be used as guidelines in real-world projects.

On the other hand, from a practical point of view, Hass (2008) proposed the “Project Complexity Model” as a framework for assessing and supporting decision-making within projects. Many contributions were achieved, including the identification of dimensions (e.g. team composition and performance), categories (independent, moderately, highly complex) and guidelines (e.g. choosing the appropriate life cycle for a project). However, this work was published as a book (focused on practitioners), not as a scientific research, lacking the required rigor.

Considering the importance of understanding and coping with inherent (and increasing) complexity within ICT projects, the development of new models and approaches for handling complexity in project management world still needs to be explored over the next years in order to provide maturity and increase success rates in ICT/SD industry..

### 5.1. Limitation and Future Works

This work represents the result of an extensive exploratory literature review using an ad hoc method. In the future, a systematic literature review (SLR) or mapping (Kitchenham, 2004) may be performed in order to achieve richer information about the state of art on management of complex project. By using a formal and robust method, it is possible to define a framework for assessing complexity and then provide a set of recommendations as guidelines.

In addition, the development of software is one area in which lack empirical evaluation in order to provide more confident evidences about in which contexts the claims are true (Cartaxo et. al. 2013, p.129). Hence, it is necessary to proceed studies in real-world projects so that empirical data might be exposed and confirm or refuse theories. Some examples of empirical studies may include case studies, surveys, focus group, ethnography, etc. (Shull et. al., 2008, p.289).

## References

BACCARINI, D. **The Concept of Project Complexity** – A Review. International Journal of Project Management. Vol. 14, No. 4, pp. 201-204, 1996.

BECK, K. **Extreme Programming Explained**: Embrace the Change. 2nd edition. USA: Addison-Wesley, 2004.

CARTAXO, B.; ARAUJO, A.; BARRETO, S.A.; SOARES, S.C.B. **The Impact of Scrum on Customer Satisfaction**: An Empirical Study. 27th Brazilian Symposium on Software Engineering. p. 129-136, 2013.

COOKE-DAVES, T.; CICMIL, S.; CRAWFORD, L; RICHARDSON, K. **We`re not in Kansas Anymore**, Toto: Mapping the Strange Landscape of Complexity Theory, and its Relationship to Project Management. Project Management Journal. Vol. 38, No. 2, p. 50-61, 2007.

CRISSIS, M.B.; KONRAD, M.; SHRUM, S. **CMMI for Development**: Guidelines for Process Integration and Product Improvement. 3rd edition. USA: Addison-Wesley, 2011.

GERALDI, J.; ALDBRECHT, G. On Faith, Fact, and Interaction in Projects. **Project Management Journal**. v. 38, n. 1, p. 32-43, 2007.

GREENFIELD, J. **Moving to Software Factories**. 2005. Available at: http://blogs.msdn.com/b/jackgr/archive/2005/12/09/502314.aspx. Accessed on Aug 15th, 2015.

GRIFFITHS, M. **PMI-ACP Exam Preparation**. 1st edition. USA: RMC Publications, 2012.

HASS, B. K. **Managing Complex Projects**: A New Model. 1st edition (Kindle version). USA: Management Concepts, 2008. Locations: 200-1020.

KITCHENHAM, B.A.; DYBA, T.; JORGENSEN, M. Evidence-Based Software Engineering. **Proceedings** of the 26th International Conference on Software Engineering. p. 273-281, 2004.

KRUTCHEN, P. **The Rational Unified Process**: An Introduction. 3rd edition. Boston, USA: Addison-Wesley, 2003.

NAUR, P.; RANDELL, B. **Software Engineering**: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO, 1969.

PICCININI, E.; GREGORY, R. The Nature of Complexity in IS Projects and Programmes. **Proceedings** of the 21st European Conference on Information Systems. 2013.

SAYNISH, M. Mastering Complexity and Changes in Projects, Economy, and Society via Project Management Second Order (PM-2). **Project Management Journal**, v. 41, n. 5, p. 4–20, 2010.

SHULL, F.; SINGER, J.; SJOBERG, D.I.K. **Guide to Advanced Empirical Software Engineering**. 1st edition. London, England: Spring, 2008.

SOMMERVILLE, I. **Software Engineering**. 9th edition. England: Addison-Wesley, 2010.

STANDISH GROUP INTERNACIONAL. The Chaos Report. 1994-2010. Technical Report.

TAXÉN, L.; LILLIESKOLD, J. Images as action instruments in complex projects. **International Journal of Project Management**. v. 26, n. 5, p. 527–536, 2008.

THOMAS, J.; MENGEL, T. Preparing project managers to deal with complexity. **Advanced project management education**. v. 26, n. 3, p. 304–315, 2008.

TURNER, J.R.; COCHRANE, R.A. Goals-and-Methods Matrix: Coping with Projects with Ill Defined Goals and/or Methods of Achieving Them. **International Journal of Project Management**, v. 11, p. 93-102, 1993.

XIA, W.; LEE, G. Grasping the complexity of IS Development Projects. **Communications of ACM**, v. 47, n. 5, p. 69-74, 2004.

XIA, W.; Lee, G. Complexity of Information Systems Development Projects: Conceptualization and Measurement Development. **Journal of Management Information Systems**. v. 22, n. 1, p. 45-83, 2005.

WHITNEY, K.M.; DANIELS, C.B. The Root Cause of Failure in Complex IT Projects: Complexity Itself. **Procedia Computer Science**. v. 20, p. 325-330, 2013.

WILLIAMS, T. Paradigms for Complex Projects. **International Journal of Project Management**. v. 17, n. 5, pp. 269-273, 1999.

ZHANG, H.; BABAR, M.A. On searching relevant studies in software engineering. **Proceedings** of the 14th international conference on Evaluation and Assessment in Software Engineering. ESEM’2010, pp. 111-120, British Computer Society Swinton, UK.