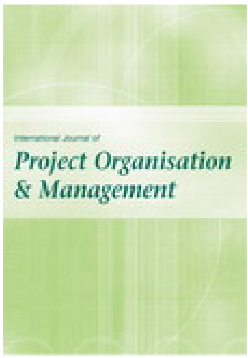
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Project Organisation and Management

ISSN online: 1740-2905 - ISSN print: 1740-2891
https://www.inderscience.com/ijpom

A complexity scoring model for evaluating complexity of software projects

Vyron Damasiotis, Panos Fitsilis

DOI: 10.1504/IJPOM.2023.10054371

Article History:

Received:	09 February 2021
Last revised:	11 July 2021
Accepted:	08 August 2021
Published online:	07 March 2023

A complexity scoring model for evaluating complexity of software projects

Vyron Damasiotis

Department of Accounting and Finance,
University of Thessaly,
Gaiopolis Campus,
41500 Larissa, Greece
Email: damasiotis@uth.gr

Panos Fitsilis*

Department of Business Administration,
University of Thessaly,
Gaiopolis Campus,
41500 Larissa, Greece
Email: fitsilis@uth.gr
*Corresponding author

Abstract: This research identifies project factors that contribute to project complexity based on the analysis of ten project management subject areas, and of one subject area which is focusing on software development process factors. Subsequently, a project complexity model based on these factors that can be used for assessing the project complexity is presented. This enables organisations to evaluate projects' complexity at early phases, providing important knowledge that may be used for project selection. The followed approach acknowledges the endogenous character of complexity in projects but instead of trying to identify complexity characteristics in project results, it focuses on the complexity of project management processes. The proposed framework can be used for highlighting the most significant complexity areas acting as an important tool for better, more efficient, and more effective project management. The final step of this research is the validation of the proposed model using case studies.

Keywords: project management; project complexity; software project complexity; complexity measurement; software development complexity; management complexity.

Reference to this paper should be made as follows: Damasiotis, V. and Fitsilis, P. (2023) 'A complexity scoring model for evaluating complexity of software projects', *Int. J. Project Organisation and Management*, Vol. 15, No. 1, pp.48–76.

Biographical notes: Vyron Damasiotis acquired his PhD in Modelling Software Project Management Complexity from Staffordshire University, UK. He holds the position of Assistant Professor at the Department of Accounting and Finance at University of Thessaly (UTH), Greece.

Panos Fitsilis is a Full Professor at the Business Administration Department of the University of Thessaly, Greece, and Academic Coordinator of the module 'Software Design' at Hellenic Open University. He has extensive project management experience with the development and deployment of large IT systems and extensive management experience in various senior management positions. His research interests include smart cities and required competences, smart factories (Industry 4.0), business information systems, social systems, educational technology, and software project management.

1 Introduction

As digital technology evolves and becomes part of every aspect of our everyday life, the demand for better software is a necessity. This leads to larger and more complex software systems in terms of functionality, development and maintenance effort needed (Janczarek and Sosnowski, 2015) and as well in terms of innovation and size (Alves et al., 2016). Software projects are considered in most cases complex and their outcome in various cases are limited or impaired since they fail to fulfil or to complete successfully the initial requirements set.

Studies such as CHAOS report (The Standish Group, 2009, 2015) and Rezvani and Khosravi (2019) indicated that among the main factors, affecting project failure or success, are those related to project management issues. A most careful analysis of the results indicates that many of these issues arise at the early stages of project design, for example during the project scope definition and the requirements elicitation stage. This implies that the basis for a successful project is set at the initial steps of project design and goes through successful and efficient project management. However, despite the progress of project management practices a project will still fail, with most of these failures to be attributed to the complexity of projects indicating that there is a direct relationship between project complexity and project failure due to high or underestimated project complexity as it was early identified by Williams (2002, 2005) and Neleman (2006).

A significant number of studies has been undertaken in recent years in order to understand, define and determine the concept of project complexity (Qazi et al., 2016; Chapman, 2016; Bakhshi et al., 2016; Nguyen et al., 2015; Lu et al., 2014; Vidal et al., 2011; Geraldi et al., 2011; Dombkins and Dombkins, 2008; Geraldi and Albrecht, 2007; Hass, 2007; Maylor et al., 2008; Vidal and Marle, 2008; Williams, 2002). They proposed various approaches in defining project complexity and determining areas that are sources of complexity but the majority of researches are limited to a conceptual approach and do not provide a practical framework for assessing or measuring complexity on projects (Poveda-Bautista et al., 2018). Attempts to provide a complete framework/model for measuring project complexity were made by Vidal et al. (2011), He et al. (2015) and Nguyen et al. (2015). However, the need for a better understanding of project complexity is needed and especially an effective measurement model as the first step in successful complexity management.

This research is aiming to provide a practical framework for assessing complexity on software projects based on project management and software development technical aspects. The increased acknowledgement of the role of project management in dealing with project complexity both in general projects or software projects is identified by many researchers. Furthermore, the ‘thinking’ of the proposed complexity model is similar and compatible to project management ‘thinking’, as it is compatible with process-based project management. Data required as input to the model are already known and available to project managers during the initial stages of project and project management planning and can be used without any further processing or modifications resulting to an easy to use and simple complexity scoring model.

The structure of this paper is as follows. In Section 2, a short introduction on software project complexity is made and various approaches in measuring software project complexity are presented. In Section 3, the theoretical framework followed in this research for modelling project complexity is presented. In Section 4, the methodological steps, the statistical analysis performed to define the model and the formation of the complexity scoring model are described, while in Section 5, the results of the model evaluation are presented. Finally, in Section 6 the conclusions and the limitations of the study are discussed.

2 Related literature

2.1 Organisational complexity

Organisational complexity, in general, is a topic that has been extensively studied during the last decades. According to Dooley (2002), “organizational complexity is defined as the amount of differentiation that exists within different elements constituting the organization”. Further, according to Reiman et al. (2015) complex organisations exhibit several key behaviours or have key characteristics such as:

a nonlinearity

b emergence behaviours

c are self-organising systems

d usually are far-from-equilibrium conditions

e there is coevolution in the various subsystems they are composed of

f contain subsystems or nested systems

g are using historical data for their future decisions.

Even though many of the above behaviours are important the primary characteristic of a complex system is considered as self-organisation (Arevalo and Espinosa, 2015).

Complexity has been studied using different perspectives according to the problem area and the required by the system behaviour. However, independently from the problem area, some common observations relate to organisational complexity (Broche and Marinescu, 2008):

- Complex systems cannot be understood by reductionism or breaking the whole down into components, since in many cases the behaviour is observed only within the totality of the system.
- Complex systems are evolving and their evolution is not deterministic.

- Change is always affecting the behaviour of a complex system.
- It is impractical to model a complex system in its totality. Instead, we are using conceptual models and views to describe their behaviour.
- Complex systems' behaviour is dependent to its initial condition.

In the next sections, we will focus on a special case of organisational structure, projects and how complexity behaviours are observed within projectised organisational structures.

2.2 Notion of project complexity

Complexity is part of our world and appears in different domains and different forms. Complex systems exist in many scientific fields and therefore different definitions of complexity have been given for each domain. Quite frequently people have difficulties in distinguishing between the term complex and complicated, considering them as synonyms (Geraldi et al., 2011). A project, even of large scale, that is self-contained, well-defined, with clear and structured tasks to their completion can be complicated but not complex. For example, the wiring of a large building (e.g., skyscraper or hospital) can be complicated but not complex, since it follows a clear implementation methodology and has a specific design. On the contrary, a complex project is a project that contains interactions of various types, structural elements with different forms and interconnections and dynamic elements that change over time (Whitty and Maylor, 2009). So, we can safely state that any project, irrelevant of its size, that is highly dependent on its environment (e.g., political, economic, legal), with stakeholders that have conflicting interests or ever-changing requirements, strategies, and decisions demands can be considered complex (Chapman, 2016).

Therefore, the differences between the terms 'complex', 'complexity' and 'complicated' are important and must be completely understood by project management scholars and practitioners in order to advance the study of project complexity and its sources. According to the Association for Project Management (APM) the complexity in a project stems from the interactions between organisations affecting the project, the interaction of various units/teams within the same organisation, the requirement for coordination between various project elements and the use of diverse project management tools, methods, and techniques (APM, 2008).

To make it clearer, this research presents a set of characteristics that make a project complicated or complex with respect to four project dimensions in Table 1.

Table 1 Complicated vs. complex

Dimensions	*Complicated*	*Complex*
Organisational	Bureaucratic, process-based, many stakeholders	Uncertainty, continuous change, diverse stakeholders
Perceived	Understanding requires substantial analysis	System is not deterministic and easily understood
Factual	Data and relationships are available and well known	Data are partially available and relationships are emergent or not understandable
Interaction	Interactions are well defined and structured	Interactions are not defined and ad hoc

A significant number of complexity frameworks have been proposed during the last years, trying to capture the complexity in projects and despite the progress have been done there are still a significant work to be done (Oehmen et al., 2015). The majority of these studies are empirical, as they are based on the opinions of experts or key point project team members and stakeholders in order to identify factors which affect project complexity. The sources of information used in these researches, were projects from construction (Qazi et al., 2016; Hagan et al., 2011; Cicmil and Marshall, 2005), infrastructure (Chapman, 2016; Nguyen et al., 2015; Dunovic et al., 2014), large infrastructure (Vidal et al., 2011; Hertogh and Westerveld, 2010) engineering (Bosch-Rekveldt et al., 2011; Geraldi and Albrecht, 2007), new product development (Schuh et al., 2017) and information systems (Xia and Lee, 2005; Ribbers and Schoo, 2002) domains. There were also researches that tried to identify project complexity components that exist in every type of projects (Lu et al., 2014; Sedaghat-Seresht et al., 2012; Remington et al., 2009; Tatikonda and Rosenthal, 2000; Williams, 1999; Baccarini, 1996). Although there is no consensus on the definition of complexity among the various researchers there is a general consensus about the project aspects that affect complexity. Uncertainty is probably the most common factor which is identified as a main source of complexity in the proposed frameworks. Uncertainty is considered to be the factor that reflects the ambiguity associated with many project aspects such as data, lack of clarity, lack of structure and unpredictable behaviour among project stakeholders (Ward and Chapman, 2003). Williams (1999) discusses uncertainty in goals related to the requirements elicitation, resource limitation and task complexity. Also, the uncertainty stemming from means used to carry out the project, is acknowledged as an important dimension of project complexity (Lu et al., 2014; Xia and Lee, 2005). Williams (1999) states that uncertainty adds to project structural complexity. Xia and Lee (2005) and Baccarini (1996) identified two dimensions of structural complexity, one related to organisational issues and the other related to the technology being used. Organisational and technological factors are next to uncertainty the most commonly identified complexity factors among the researchers. The organisational factor is related to project staffing, coordination of stakeholders, contract management project planning and scheduling, organisation departments, hierarchy structure, etc. and has received great attention by researchers during the previous years (Nguyen et al., 2015; Lu et al., 2014; Vidal et al., 2011; Bosch-Rekveldt et al., 2011; Xia and Lee, 2005; Baccarini, 1996). Vidal et al. (2011) suggest that organisational complexity is the most significant source of project complexity. The technological factor refers to relationships between technology elements, the variety of technology platforms, technology novelty, newness of project technology, technology changes and has also attracted attention from other researchers (Nguyen et al., 2015; Lu et al., 2014; Vidal et al., 2011; Bosch-Rekveldt et al., 2011; Xia and Lee, 2005; Remington et al., 2009; Baccarini, 1996). Two aspects of project technology, which are the newness of technology being used in projects and the technology immaturity, are identified by PMI (2013) among the most important factors of the complexity of projects and their management. It is worth noticing that despite the number of proposed complexity frameworks the majority of them are limited to a conceptual approach and do not provide a practical framework for assessing or measuring complexity on projects.

There are two major approaches to complexity (Floricel et al., 2018; Schlindwein and Ison, 2004). The first one is called *descriptive complexity* and describes complexity as a property of a system. The second approach is called *perceived complexity* and it is described as the subjective complexity that someone experiences through the interaction with the system. Researchers such as Hagan et al. (2011) and Baccarini (1996) are considering complexity as a subjective property that can change according to the observer, an approach that imposes difficulties in understanding and dealing with a problem or situation and for this reason, it is not considered by the authors as a reliable basis for further analysis.

On the other hand, a number of researchers argue that the perception of complexity is dependent on the cognitive level (knowledge, experience, background, personality) of the people involved (Remington et al., 2009; Fioretti and Visser, 2004) and that the subjectivity in the evaluation of factors affecting project complexity is an inherent characteristic of this process (Montequín et al., 2018). According to this view, for the same project, some project managers may define the project as complex, e.g., because of the number of changes in requirements, while for some other project managers and the same project to consider that the complexity of the system is attributed to the large number of stakeholders, and finally for some others, probably more experienced project managers, to define the same project as of low complexity simply because of their experience. Furthermore, one characteristic of complexity that we should consider is the observer's perception of complexity that can change over time. This change may be due to increasing experience and/or knowledge gained over time, making a project that was initially perceived as complex, to be considered less complex, if performed repeatedly, or followed by more ambitious projects (Chapman, 2016). This research is in line with the second approach.

2.3 Complexity in software projects

Software projects are a special case of projects since the final product in many cases is not tangible, they model and implement various aspects of human interaction and behaviour and their success is heavily relying on development team coordination. As such, software project complexity relates to the complexity of the software final product and as well on the software development process (SDP) used. Several approaches of software complexity have been proposed by researchers according to the domain where they originated from.

Zuse (1990) approached software complexity from a programmer's psychological perspective and defined it, as the difficulty to analyse, maintain, test, design and modify the software. Along the same lines, Kushwaha and Misra (2006) defined software complexity as the degree of difficulty to understand and verify a system or a component. Keshavarz et al. (2011) stated that although there were different approaches for defining software complexity, most of them comply with Zuse's approach. Ribbers and Schoo (2002) in their research for complex software implementation programs, examined complexity through the prism of implementation complexity, and identified three complexity dimensions: variety, variability, and integration. Variety is defined as the different states a system can take. Variability of a system is defined as the dynamics of its elements and the interrelations between them. Finally, integration is referred to as the planned changes during the implementation program including IT systems and business processes.

Software engineers measure software complexity using various properties and code characteristics such as code size, number of software defects, development cost and time, number of control paths and frequency of operators and operands within the software.

However, the existence of larger numbers of classes, control flows, or modules within the developed software system does not necessarily imply that this software project is more complex than a similar one with smaller numbers (Ghazarian, 2015). In addition, Khan et al. (2016) in their research compared several complexity measurement models based on code characteristics and identified that different models produce different results as they capture different aspects of software code.

Other researchers approach software project complexity from various perspectives such as the perspective of project maturity level (Bolat et al., 2017), adoption of effective project management model (Aydin and Dilan, 2017), creation of an effective project management plan (Rahman et al., 2016), identification of critical project success factors (Stevenson and Starkweather, 2017; Altahtooh and Emsley, 2017) and adoption of agile development methods (Truong and Jitbaipoon, 2016). Finally, Marengo and Pagano (2020) state that assessing projects, in many ways resamples assessing a soft skill due to its nature.

The above is an indication that the study of the complexity of software projects, in general, is a multidimensional process.

Focusing on SDP, Sharma and Kushwaha (2010) and Keshavarz et al. (2011) in their study on software complexity measurement state that software complexity measures based on code are not the best practical approach in assessing software complexity, as the code of the software is produced at the later stage of software development. For this reason, they proposed a complexity framework which is based on requirements engineering documents. They argue that if we study complexity at the requirements level we can utilise software aspects such as functional and non-functional requirements, technical expertise, design constraints, number of interfaces, number, and type of inputs and outputs and number of users and locations that will be deployed by the software system, which is considered quite useful.

Methods that are able to measure such characteristics are well known parametric software estimation methods such as the use case point (UCP) (Karner, 1993), function points (FP) (Albrecht, 1979) model and constructive cost model (COCOMO II) (Boehm et al., 2000). These models and their similarities go beyond the identification of fundamental code characteristics and they take into consideration factors that relate to aspects of the software development process such as change management, requirements stability, team cohesion, team experience and training, team motivation issues, etc.

Furthermore, we need to consider models that relate to software development effort (Jiang and Naude, 2007), software development productivity (Trendowicz and Münch, 2009; Wagner and Ruhe, 2008) and software systems development outcomes (McLeod and MacDonell, 2011) since the effort required or the productivity offered by a specific SDP and SDP complexity are directly related, as higher complexity usually implies higher effort and reduced productivity.

Finally, many studies make evident that the study of the complexity of software projects should contain both project management and SDP complexity. For example, Velayudhan and Thomas (2018) identify the influence of technical complexity and technological uncertainty in project planning, Ribbers and Schoo (2002) in their model for assessing software programs implementation complexity identified that the management aspects directly affect complexity (e.g., team structure, communication, cost, and time management). Similarly, Xia and Lee (2005) and Lee and Xia (2002) stated that the complexity of the information systems development projects sources from both technological and business processes. Fitsilis et al. (2010) stated that size alone is not sufficient for measuring software project complexity, "since a large but well-structured software project with a relaxed cost and time constraints can be much less complex in comparison with a relatively small-in-size project, which has a highly integrated product design and limited budget and/or time-to-market objectives".

Project management has a major contribution to project success and its complexity can significantly affect the project result (Cooke-Davies et al., 2007). Ribbers and Schoo (2002) in their proposed framework for assessing software programs implementation complexity, identified the management aspects which affect complexity such as team structure, communication, cost and time management. Lee and Xia (2002) stated that the complexity of the information systems development projects sources from both technological and business processes. Tie and Bolluijt (2014) state that project management and project complexity management are very close related. Kermanshachi et al. (2015) acknowledging the relationship between project complexity and project management identified 37 complexity indicators and the corresponding management strategies. These should be incorporated to the project execution plan, in order to keep it within budget and schedule constraints. Regarding agile methods, they emphasise measuring the software product or the software development process mainly and only partially and/or fragmentally take into consideration project management as a separate entity. This indicates that the study of complexity of IT projects and of SDP is a multifaceted phenomenon. Regarding the top ten factors that lead to project success or project failure as described in various studies such as the 'CHAOS report' (The Standish Group 2015, 2009) and 'Why software fails' (Charette, 2005), it is obvious that most of them identified many project management aspects as the causes of failure. Issues related to proper planning, requirements management, scope management, risk management, procurement management, communication management, human resource management, executive management support, user involvement and technology related issues are referred to as success or failure factors within these researches. Furthermore, Xia and Lee (2005, p.2) stated that information systems projects "are inherently complex because they deal not only with technological issues but also with organisational factors largely beyond the project team's control".

Considering the previous approaches, it is apparent that project management and project complexity interlock and the management of the one should involve the aspects of the other, too. Therefore, the assessment of software projects complexity should take into account beyond technical software development aspects and project management aspects.

3 Theoretical framework for modelling project complexity

Damasiotis (2018) presented in a systematic way the available, in the literature, project complexity assessment frameworks that constitute the theoretical basis of this work. These frameworks will enable us to build a theoretical project complexity framework that aims at being a reference for project managers for assessing software project complexity, so that they can manage software projects more effectively.

- As it is obvious from the literature, briefly presented in the previous section and from Damasiotis (2018): complexity undoubtedly exists in every aspect of a software project in various forms and levels, it has many facets and pertains to all software project processes including project management. In addition, this complexity does not differ substantially from complexity apparent in other types of projects.

- There are numerous complexity typologies and frameworks.
- The study of complexity in software projects should include both project management processes and SDP processes.
- Current approaches are studying complexity taking into consideration only some project management aspects, and therefore project management complexity should be addressed in its totality by studying all different aspects of project management processes.

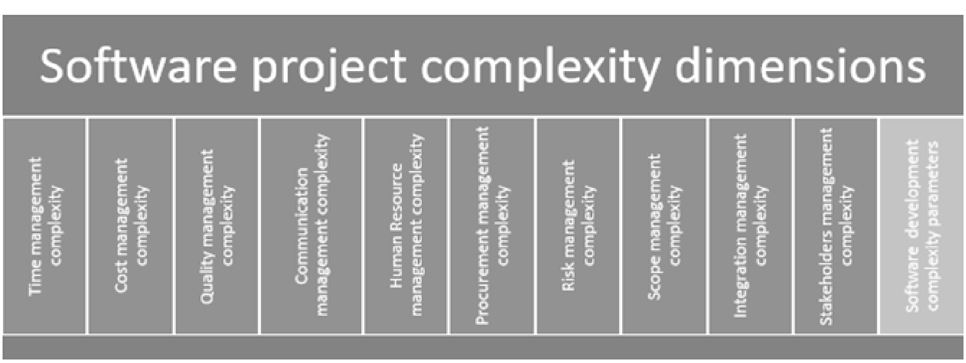
This research is attempting to resolve the above issues since:

- it defines the complexity in software projects both from the perspective of the project management and from the perspective of the software development process
- it develops a comprehensive complexity model able to measure the complexity of software projects, even at the early stages of a software project
- it considers that the perception of complexity is dependent on the cognitive level of the people involved and their subjectivity in the evaluation and approaches project complexity from this perspective.

For achieving the above, this research identifies 11 dimensions of complexity in managing software development projects. These are the ten management areas as defined by PMBOK that are a generic framework and cover management aspects of all project types and since this research is focusing on software projects these dimensions are supplement by one more dimension that cover aspects of software project development not able to be captured by a generic framework. PMBOK structure and subject areas were selected, to be the background of this work due to its popularity, the extensive coverage of project management processes and due to the fact that it is process-based (Cardona-Meza and Olivar-Tost, 2017). The result is the formation of a complexity typology that has eleven dimensions. Figure 1 presents the proposed complexity typology.

Next section describes the methodology followed to develop the complexity model and the relative complexity index.

Figure 1 Software projects complexity typology



4 Research methodology

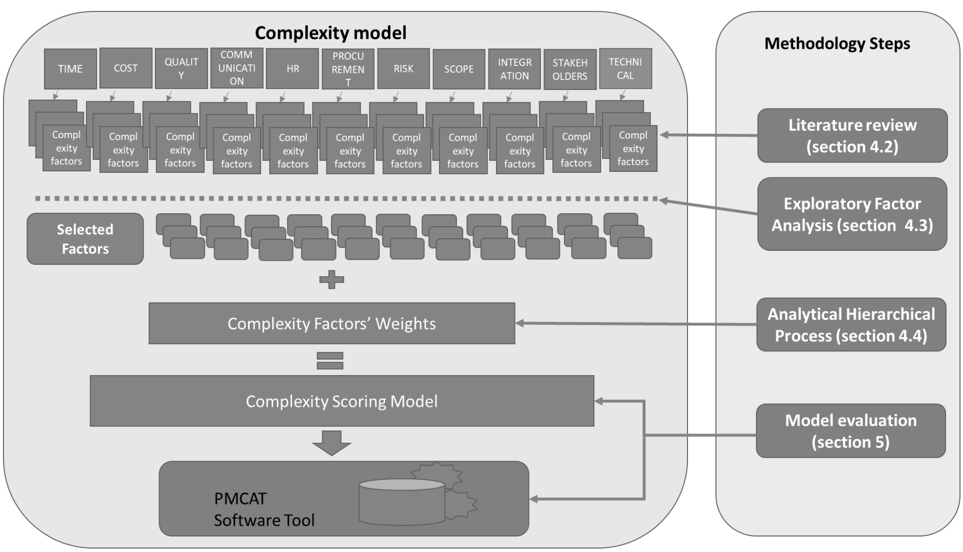
In the following section, the research methodology and steps followed in order to define the complexity model are presented.

4.1 Research design and steps

The design of this research is inductive and exploratory. The exploratory approach implies that the research is not intended to provide conclusive evidence but to help readers to have a better understanding of the problem and change direction as a result of the revelation of new data and insights (Saunders et al., 2012). The inductive approach means that starts with detailed observation of a specific matter and moves towards generalisations that are more abstract. Following an inductive approach, this research tends to develop empirical generalisations and identify preliminary relationships as it progresses. Furthermore, no hypotheses can be made at the initial stages of the research and cannot be drawn conclusions about the type and nature of the findings until the study is completed. Under that prism, this research did not start by setting up a specific hypothesis to test, but starting up with some observations regarding the factors that could affect the software development process and in continuous tries to identify existing relationships between them as research progress. This is done through literature review initially and using appropriate statistical methods in continuous. Then through empirical testing, the findings are generalised.

The main steps followed for defining the complexity model are in short presented in Figure 2, so the development process of the complexity model is elaborate.

Figure 2 Main methodology steps



As illustrated in Figure 2, the first step of this research is the identification of a set of complexity factors sourcing from each one of the 11 dimensions of the proposed complexity model (see Section 4.2). Second, as the number of the initially identified factors is expected to be high, it will be tried to be reduced to a more manageable number by grouping common factors and/or deleting some of them as statistically insignificant resulting to final set of complexity factors by applying exploratory factor analysis (EFA) (see Section 4.3). Third, weights to each one of the factors of the final set of complexity factors are assigned in order to be determined their relative contribution to total project complexity. To implement that expert judgement and a multi-criteria decision-making technique and specifically analytical hierarchical process (AHP) are used (see Section 4.4). Next, the complexity scoring model is formed. Finally, the proposed complexity model is validated through a set of case studies.

4.2 Determining software project complexity measures

To identify the initial set of complexity factors an extended literature review was conducted, using mainly e-resources such as e-databases and web search engines. Briefly, a number of electronic databases were used such as Science Direct, Emerald, IEEE Xplore, Taylor online, ACM Digital Library, Google Scholar, and general web search engines such as Google and Bing. The search strings used for finding papers relevant to the concept of each complexity area varied for each area. For example, some of the search strings used, in the area of scope management some of the searching strings that were used were 'scope management complexity', 'scope management', 'software complexity', 'software scope management', 'requirements management', 'successful scope management factors', 'requirements management performance', 'requirements elicitation', 'requirements engineering', etc. while for time management were 'time management complexity', 'time management', 'scheduling complexity', 'software time management', 'scheduling management', 'successful time management factors', 'time management performance', etc. As a result, a set of relevant papers were identified and a list of 135 complexity factors was extracted (Damasiotis, 2018; Damasiotis et al., 2017). All factors were categorised into 11 complexity areas, according to the domain they originated as described in Section 3. Obviously, the number of complexity factors identified through this literature review was quite large. The inclusion of a large number of factors in a measurement model makes the model cumbersome and unmanageable. Beyond that, it is a fact that many of these factors are interrelated and this implies that dependencies have to be further examined before concluding to a final smaller and concrete list of factors.

4.3 Complexity factor reduction using exploratory factor analysis

For reducing the number of factors, simple statistical methods such as those based on median, mean, missing variables, high correlation, and low variance can be used but these methods cannot identify underlying structured relationships between factors. Therefore, statistical methods able to achieve this goal were examined. It was decided that exploratory factor analysis (EFA) should be selected, as it is considered the most suitable method for reducing the factors and determining the structured relationships between them while retaining as much of the original information as possible (Yong and Pearce, 2013; Field, 2009; Child, 2006; DeCoster, 1998). Furthermore, there are numerous descriptions and suggestions in the literature, there are plenty of software tools available that implement it and it is a well-known method with numerous applications.

However, the main problem in this phase was the large size of the required sample to proceed with EFA, because of the large number of variables (135) to be examined and this was amplified by the nature of the sample, which should be experts of project management domain. Specifically, it would be needed more than 1,000 responses to get an adequate sample for proceeding with EFA. Considering that usual response rate in surveys are 5–25%, the initial sample size should had been enormous and impossible to be gathered by the research team. As that, the identified factors were grouped in categories according their domain of origination that allowed their processing in categories. This led to the reduction of the required sample size by ten times. As a basis for this categorisation, the eleven dimensions of the proposed model were used. In the next sections, the main points of EFA analysis are presented.

4.3.1 Preparing survey

A questionnaire divided into 11 subsections was developed, in accordance with the proposed typology. This allowed the identification of the main complexity factors within each category, and the focus on complexity sources within the various software project processes.

A list of possible respondents who had professional and/or academic experience in software projects in either the private or public sector was formed. The sources from which this list was formed were the Greek Information Society S.A, the Federation of Hellenic ICT enterprises, Technical Chambers of Greece, Greek Project Management organisations and associations, academic organisations in Greece and in UK, various business organisations in Greece and the UK whose business scope was relative to IT development, software development, and software engineering. Also, individual project managers working either in public or private organisations were included. The total number of responders exceeded the total number of 500 persons. Responders were asked to identify the relative contribution of each factor to project management complexity. In order to achieve this, they have rated each project complexity factor using a positive five-point Likert scale. The questionnaire was distributed electronically and the platform used to create and collect the responses was Google Forms.

4.3.2 Data source, adequacy and reliability analysis

In total, 102 valid responses received. Out of 102 responders, 89.2% were men and 11.8% were women. Concerning their educational level, 41.2% had a PhD, 36.3% had an MSc degree and 22.5% were university graduates. In relation to their work background, 55.9% were working in industry/businesses, 20.6% were coming from academia and 23.5% had experience in both academic and business domains. Further, 58.9% of the respondents had experience steaming from work at the private sector, 20.6% from the public sector, while the remaining 20.5% was related to work experience from both the private and public sectors. In addition, 11.9% had working experience below 5 years, 18.8% had experience between 6–12 years, 39.6% had experience between 13–20 years, and 29.7% had more than 20 years of experience. Lastly, 62.9% of the responders were involved in projects with a budget below €300,000, 19.1% in projects with a budget between €300,000 and €1,000,000 and 18% in projects higher than €1,000,000.

For measuring sample adequacy Kaiser-Meyer-Olkin (KMO) measure was used (Kaiser, 1970). The KMO value varies between 0 and 1. Accepted values should be greater than 0.5. The calculated KMO values for all categories of our data were between 0.763 and 0.858, which was considered as very good (Hutcheson and Sofroniou, 1999).

In order to verify the reliability of the scale used in the questionnaire, Cronbach's α test (Cronbach, 1951) was used, for each one of the 11 complexity categories/sections. The results indicate that the scale used was reliable as the calculated Cronbach's α was in all cases well above 0.8 (Field, 2009). Specifically, the lower value was 0.814 and the higher value was 0.897.

4.3.3 Factor extraction

For initial factor extraction, CFA was selected since this research is focusing on reducing the number of complexity factors (variables), by revealing the underlying complexity components that are not profound and can be assessed using these individual variables.

However, before proceeding to factor extraction, the factorability of the data had to be examined. As that Bartlett's test of sphericity (Snedecor and Cochran, 1989) was calculated. To proceed with extraction and factor analysis, the value of the significance of Bartlett's test should be lower than 0.001, which is something that holds for all our cases. Furthermore, the communalities of variables were examined. Communalities, with values, below 0.3 or 0.4, usually indicate that the variable does not fit well with the other variables and it should be considered the elimination of these variables before proceeding (Pallant, 2011; Field, 2009). However, as the factor analysis is an exploratory tool the above control is not mandatory. What should be examined is how this variable is loaded on the factor matrix. In this case, it was decided to eliminate all variables with communalities below 0.3, resulting in the elimination of one variable from the cost management area and one variable from the scope management area. This decision was supported also by the fact that these variables had loadings lower than 0.4 on all factors on their corresponding rotated factor matrix. After variables deletion, the analysis was performed again without these variables.

For deciding which factors should be extracted Kaiser criterion was used. Kaiser criterion is the most commonly applied, but when it is used with CFA extra caution is needed, since only common variance between variables is used and as such, factors with eigenvalue lower than 1 may need to be retained, as they account for significant variance otherwise under extraction of factors may occur (Beavers et al., 2013). Like that, it was additionally used the scree plot approach (Cattell, 1996), which is a graphical representation of each eigenvalue (in Y-axis) with the corresponding factor (in X-axis). Kaiser criterion suggests retaining all factors with an eigenvalue greater than one. Scree plot suggests keeping all factors before the factor that the plot becomes an almost straight line. Both criteria gave similar results in most cases, except for the cases of communication and scope complexity areas. In these two cases Costelo and Osborne (2005) suggestion it was followed. They suggest to keep the number of factors that give the 'best' results in terms of few cross-loadings, adequate factor loading, and factor number, and in our case these were indicated by Kaiser criterion.

Finally, varimax rotation was applied, which is the most common orthogonal rotation method used, as the direct solution does not provide an easy or sufficient interpretable solution (Beavers et al., 2013; Snedecor and Cochran, 1989). Furthermore, all loadings with a value below 0.4 were suppressed for facilitating the interpretation of the results.

The outcome of EFA was a list of a 35 complexity factors divided in 11 dimensions as can be seen in Table 2.

Table 2 EFA results

Complexity areas/dimensions	*Complexity factors code*	*Complexity factors name*
Time	TM1	The density of project activities
	TM2	Project activities resource constraints
	TM3	The density of project schedule
	TM4	Protracted project /activities duration
	TM5	Organisation time management immaturity
Cost	CM1	Organisation cost management immaturity
	CM2	Complicated financial structure and processes
	CM3	Long project duration
Quality	QM1	Inadequacies in quality management design
	QM2	Organisation quality management immaturity
	QM3	Rigorous quality control procedures
Communication	COM1	Organisation communication management immaturity
	COM2	Communication constraints due to project structure and staffing
	COM3	The density of project communication
Human resources	HRM1	Project team cohesion
	HRM2	Organisation HR management immaturity
	HRM3	HR management constraints due to team structure
	HRM4	Project team size and skill diversity
Procurement	PM1	The density of procurement process
	PM2	Organisation procurement management immaturity
	PM3	External barriers in project procurement process
Risk	RM1	Organisation risk management immaturity
	RM2	Project risk density
Scope	SM1	The density of project requirements
	SM2	Quality of requirements
	SM3	Organisations scope management immaturity
Integration	IM1	Integration constraints due to project characteristics
	IM2	Organisation integration management immaturity
	IM3	The density of deliverables
Stakeholders	STM1	The density of stakeholders' management
	STM2	Organisation stakeholders' management immaturity
Software development (technical) factors	SD1	Organisation technological immaturity
	SD2	Product development constraints
	SD3	Product quality requirements
	SD4	Software size

4.3.4 Evaluation of survey results

An indicative measure for evaluating how the resulted model fits with the data was to examine the second half of 'correlation reproduction matrix' called 'residual'. In order to have an acceptable model, less than 50% of the variables should have values greater than 0.05 in this matrix (Field, 2009). In all cases, the values in corresponding residual matrices that were greater than 0.05, was well below the 50% varying from 4% to 24% which is a positive indication for the fitness of the model (Field, 2009).

The analysis resulted in a list of 35 complexity factors/measures (Table 2). Each complexity factor has a number of constituent variables as a result of the EFA. In Table 3, an example of the factors extracted for the 'time management complexity area' with their constituent variables are presented. The variables presented in the second column of the table were grouped as displayed and according to our analysis revealed the common underlying factors that are presented in the first column. A full list of all the complexity factors identified with their constituent variables can be found at (Damasiotis, 2018; Damasiotis et al., 2017).

Table 3 EFA results for time management complexity area

	Extracted factors	*Variables*
1	The density of project activities	The number of project activities.
		The number of critical activities.
		Variance in project activities duration.
		Many dependencies between activities.
2	Project activities resource constraints	The number of activities with overlapping resource requirements (shared activities).
		The number of activities that require a high variety of resource types.
		Low availability of project resources.
		The number of activities that require highly specialised resource types.
3	The density of project schedule	The number of project activities executed in parallel.
		The number of intermediate deliverables should be delivered.
		The high project deliverable density (ratio, number of deliverables/project duration).
4	Protracted project/activities duration	The number of long project activities.
		The long project duration.
5	Organisation time management immaturity	Insufficient time management experience within the project time management team.
		Lack/shortage of tools for planning and monitoring project schedule.

4.4 Assigning weights to identified complexity factors

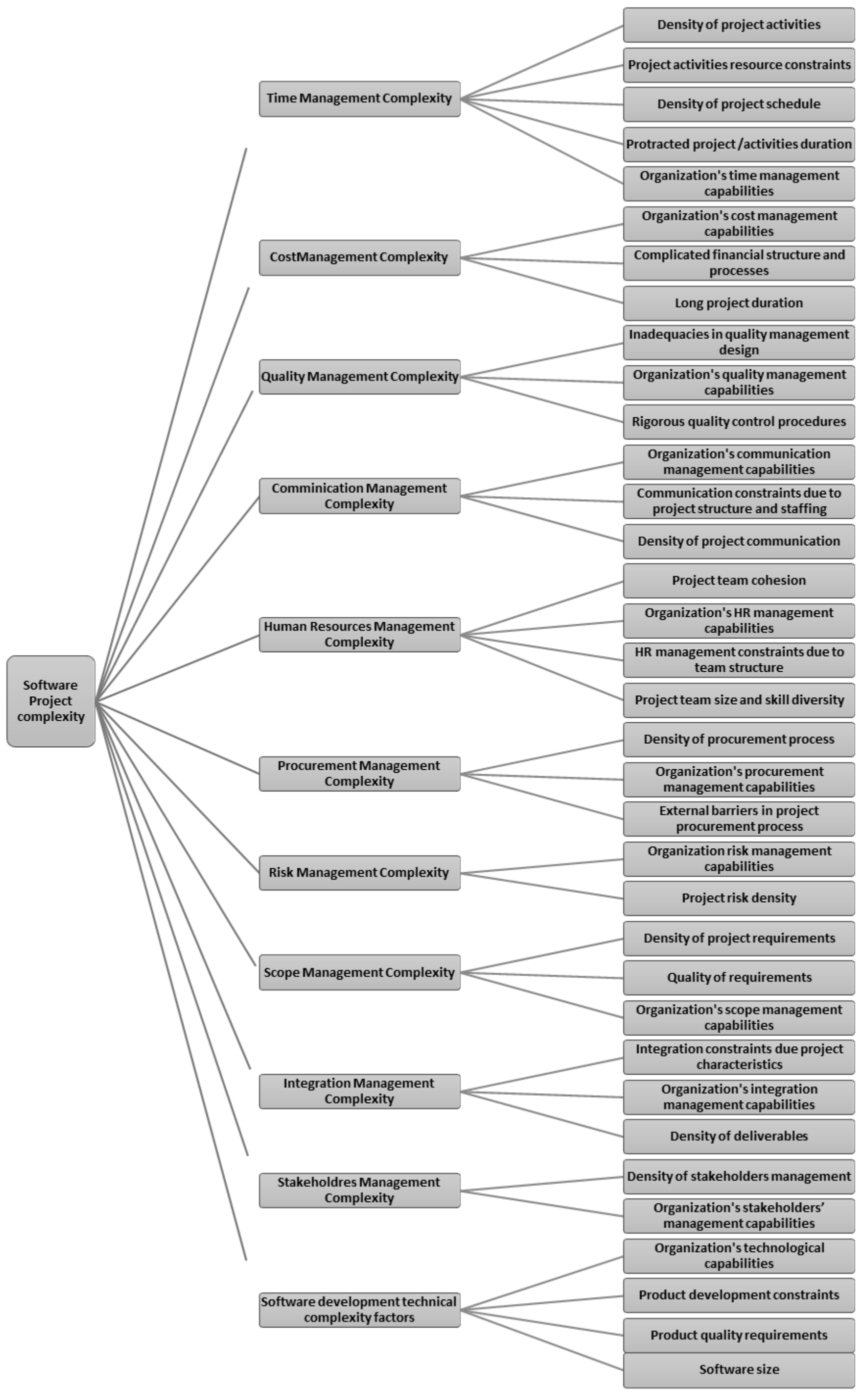
Having identified several factors affecting, the complexity of software projects, the next step was to identify the relative importance of each factor in relation to total project complexity. To achieve this, weights needed to be assigned to each complexity factor using an appropriate multicriteria method. According to Vidal et al. (2011), the selection of the appropriate multi-criteria method is not trivial and it is a multi-criteria problem itself. In their research, they identified a set of requirements that a multi-criteria method should meet in order to be used for evaluating project complexity. They used multi-criteria analysis to prove that among the various multi-criteria decision methods the AHP (Saaty, 1980) is likely to be the most suitable for complexity evaluation. Their outcome is also supported by the numerous applications of AHP, from researchers who considered it as the most appropriate and user-friendly tool in a number of different contexts such as in project management, in software tool selection, technology selection, etc. (Vaidya and Kumar, 2006; Al-Harbi, 2001; Ahmad and Laplante, 2006; Alhazmi and McCaffer, 2000; Daim et al., 2012; Patanakul et al., 2007; Hongyan, 2010; Lin et al., 2008).

Prioritisation of complexity factors in the herein research rely on the opinion of experts. According to Daim et al. (2012), AHP suits better with expert judgement since it is a method that tries to reveal the consensus among a group of experts on a specific subject. Furthermore, it allows the integration of the quantitative and the qualitative aspects of decision making, which makes it suitable to be used in complex contexts (Saaty, 1980). Using AHP at this stage of the research enhances further the validity of the decision taken. In the next section a short description of AHP process followed is presented.

4.4.1 Forming questionnaire

In AHP, the existence of a large number of elements increases the number of comparisons resulting in a very arduous process for responders (Daim et al., 2012) with huge number of pairwise comparisons. In addition, Simpson and Cochran (1987) state that AHP methodology can be better applied, when 2 to 15 alternatives exist, otherwise they suggested reducing the number of alternatives. In line with the above suggestion and considering that the final list contains 35 factors, it was decided to keep the categorisation schema of the complexity factors that had been used in the previous research step and to transform the complexity categories/dimensions into criteria, and the complexity factors to alternatives resulting in having from 2 to 5 alternatives per criterion. In this way, the above suggestions were fulfilled and the importance of each complexity factor within each area was evaluated. Another advantage of this solution was the possibility to calculate the complexity of each management area, beyond the calculation of total project complexity. It is important to notice that in this research all model complexity categories/dimensions were assumed to be of equal importance in terms of total project complexity and only the complexity factors within each category, corresponding to the lower level of AHP hierarchy were quantified using expert judgement (see Section 4.4.4). The weighting of complexity categories can be included in later model modifications. Figure 3 presents the AHP hierarchy used. All stages of the AHP process were supported by a software tool, the 'Expert Choice 2000' (http://expertchoice.com) which is a tool that automates the AHP process. Experts had the opportunity to express their opinion using either a verbal scale, from equal to the extreme, or a numerical scale from 1 to 9. The verbal scale was transformed into numeric as follows: equal = 1, moderate = 3, strong = 5, very strong = 7, and extreme = 9. The intermediate values of 2, 4 and 6 were used to refine the answers. This approach allows the decision maker initially to capture his vague preference and then systematically to sort them into a prioritised sequence (Daim et al., 2012).

Figure 3 AHP hierarchy



4.4.2 Selecting survey panel

A group, consisting of 17 experts, was formed in order to evaluate the factors. Their expertise was related to software development and the domain of project management. They all had at least five years' experience in managing software development projects in the public and/or the private sector. Furthermore, six members of the group had more than ten years' academic experience in the domain of project management. Concerning their educational level, 10 of them hold a PhD and the rest an MSc.

4.4.3 Data collection

The survey was carried out through personal interviews with the responders. In the beginning, a brief description of the scope and the aims of the survey were given followed by a presentation of the alternatives and questionnaire. Then, experts were asked to provide their answers. The data collected through 'Expert Choice 2000' tool that allows the creation of different 'users'/respondents for a specific questionnaire and then automatically aggregates their responses to a single output following the AHP methodology based on geometric means. Furthermore, the use of the software tool allowed the immediate calculation of the consistency of the responses giving the opportunity to responders to re-evaluate, if they wanted to do so, their answers. In all cases of this study, the inconsistency level was below 0.1.

4.4.4 Weighting process results

The calculated weights for each complexity factor are presented in Table 4. Higher values indicate higher contribution of this factor to project complexity.

Table 4 Complexity factors weights

TM1	*TM2*	*TM3*	*TM4*	*TM5*	*CM1*	*CM2*	*CM3*	*QM1*	*QM2*	*QM3*	
0.193	0.281	0.214	0.087	0.225	0.313	0.451	0.235	0.272	0.288	0.44	
COM1	*COM2*	*COM3*	*HRM1*	*HRM2*	*HRM3*	*HRM4*	*PM1*	*PM2*	*PM3*	*RM1*	*RM2*
0.382	0.252	0.366	0.342	0.208	0.184	0.265	0.281	0.412	0.307	0.398	0.602
SM1	*SM2*	*SM3*	*IM1*	*IM2*	*IM3*	*STM1*	*STM2*	*SD1*	*SD2*	*SD3*	*SD4*
0.312	0.433	0.255	0.401	0.293	0.306	0.484	0.516	0.203	0.2	0.29	0.307

4.5 Overview of the proposed complexity model

The proposed model consists of eleven complexity dimensions. Ten out of eleven complexity dimensions are based on PMBOK's (PMI, 2013) project management knowledge areas and the last one is related to technical factors of SPD process. For assessing the complexity of a software project, all 35 complexity factors need to be assessed. Each factor is assessed using a linear scale that ranges between 0 and 10, where 0 stands for no contribution or no applicability or no significance of this factor to the project complexity, while 10 stands for extremely high contribution or applicability or significance of this factor to the project complexity. Based on the above, the complexity model has the form of a questionnaire (see Figure 4).

Finally, the complexity level of the overall project is measured in values between 0 and 10 with higher values indicating higher complexity. Specifically, a value of 0 indicates 'No project complexity' while a value of 10 indicates 'Extremely high project complexity'.

Overall project complexity (OPC) is calculated by the use of the following formula:

$$OPC = \sum_{j=1}^{35} CFV_j * CFGW_j$$

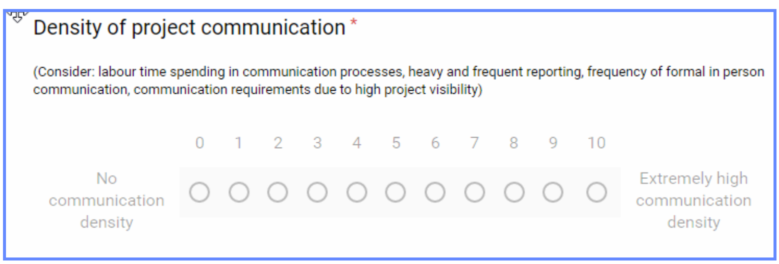
where

$CFGW_j$ is the complexity factor global weight of j factor

CFV_j is the complexity factor value of j factor.

Even thought, in this research it has been calculated the relative weights of each complexity factor giving to project managers a head starts, the weight of each factor can be adjusted in order to meet the differences and the variations of different project categories. Therefore, a project manager may create specific project profiles using custom weights, giving to the proposed model adaptability to different circumstances.

Figure 4 Example of question (see online version for colours)



4 Model evaluation

4.1 Case study design

Model evaluation was conducted by applying the model to a number of well selected software projects. These five software development projects were selected, projects that were aiming at delivering diverse software products and implemented during the three last years in Greece and other EU member states. Profiles of these projects are presented in short in Table 5.

The complexity of project 1 is sourcing firstly, from the requirement for using a specific technology platform. This added significant technical restrictions in software development. Secondly, the project was heavily bureaucratic and the project organisation structure was burdened by this. Thirdly, due to the nature of the project, the probability for requirements changes caused by legislation changes imposed either from national or from EU requests was high. Finally, the geographical distribution of the project team added another level of complexity to this project.

The complexity of project 2 according to its project manager is sourcing from the low cohesion of the project team and the applicable to the project legislation. Specifically, the project team beyond software developers included several other types of specialties such as topographers, urban planners, lawyers, and notaries, which not all of them were full time dedicated to the project. Their work, for several of them, was different and they had a role complementary to the project. This was something that affected their commitment towards the project. On the other hand, there were too many local regulations that contradicting in many cases the general legislation, resulting in messy and complicated legislation for cadastre and city plan. Furthermore, this legislation was prone to frequent changes, affecting the requirements specification. Furthermore, the lack of data led to manual data entry, which however it was not a trivial process, as the owner of the data (central government or municipality or other) was not known.

Table 5 Case study projects overview

Project ref	*Project scope*	*Project finance*	*Project client*	*Project budget*	*Project duration*
Project 1	MIS for monitoring project financed by EU	Public	Ministry of Finance of EU member state	~€1,000,000	11 months
Project 2	GIS for motoring and management the cadastre and city plan of a major Greek city	Public (national and EU funds)	Municipality of Greece	~€500,000	8 months
Project 3	Decision support system (DSS) for effective water management in household and urban level	Public (national and EU funds)	Municipal organisations in Greece	~€3,350,000	36 months
Project 4	Decision support system (DSS) for personalised management of HPV related diseases	Public (national and EU funds)	Greek industries and academic institutions	~€790,000	27 months
Project 5	IS for supporting students and companies to allocate and propose vocational placement and graduate job positions	Public (national and EU funds)	Greek university	~€120,000	16 months

The complexity of project 3 was sourcing mainly from the heterogeneity and the different types of project stakeholders. In addition, the number of different types of specialties within the project team was another factor that negatively affected the cohesion of the team. In a project with the above characteristics conflict of interests could easily arise between project stakeholders, which in turn could negatively affect the project progress. Another source of complexity, in this project, was the high level of interdependences between the project processes as a delay or failure in one of them could significantly delay or fail other processes and hence jeopardise the success of the entire project. Further, the geographical distribution of the team, the lack of data related to technical aspects of the project, the lack of commitment by some stakeholders, and the unstable market conditions influenced the complexity of the project.

The sources of complexity in project 4 were similar to those of project 3 despite the fact that this was a different type of project. Specifically, project 4 had also a significant level of heterogeneity in its stakeholder's team and the project team cohesion was low. Project 4 had a high level of interdependencies between its processes as software development was highly dependent on the development of a series of biological or other models that were developed concurrently. These dependencies according to the project manager had affected the requirements elicitation process for the software being developed. Other sources of complexity were the possible lack of data, deficiencies in data quality, and the lack of commitment from some stakeholders that delayed their involvement in the project.

Project 5, although initially, does not seem to be complex mainly due to its small size and scope, according to its manager, included a significant level of complexity for the following reasons. Firstly, the development team did not have significant experience in team working. Secondly, the project was heavily bureaucratic in its financial, procurement, and various administrative processes and with many legal constraints. Thirdly, there were cash flow delays. Fourthly, there was a significantly difficulty in eliciting requirements, especially for those requirements that concern non-functional requirements. The two main types of stakeholders involved in this stage, students and employers did not have the necessary experience in expressing the requirements with clarity, accuracy and completeness. The above indicates that the project had a substantial number of dependencies from its environment either internal or external, and significant degree of uncertainty that could affect its complexity. Therefore, and considering that this research is focusing on perceived type of complexity, which implicates that different levels of complexity can be identified by different organisations due to their different capabilities, it was decided to include this project in our case study, in order to examine if the level of complexity perceived from the project manager is verified by the model.

5.2 Case study results

A questionnaire based on proposed complexity model, as presented in Section 4.5 was given to project managers. They asked to assess each complexity factor in the questionnaire considering the conditions of the specific project they had managed. In almost all questions higher values indicated higher contribution or significance of these factors to project complexity except from factors HRM1 and SM2, which are in reverse significance order, meaning that higher values indicate lower contribution or significance of these factors to project complexity. This was a presentation issue, decided mainly for better question understanding. Next, the formula described in Section 4.5 was applied and the complexity of each project was calculated. The results can be seen in Table 6. Finally, it was asked from project managers to assess the overall project complexity they perceived during project execution in a scale from 0 to 10. By that way it could be compared the two values in order to examine the validity of proposed complexity model.

Before proceeding to the analysis of the results, it should be noted that an error margin of ±15% or 1.5 unit with respect to total complexity scale was considered acceptable. The error margin defines the accepted difference between the value of complexity level calculated by the model and the value of perceived complexity determined by project managers. The value set was considered appropriate for the following reasons: Firstly, the project manager's evaluation was based on subjective evaluation of perceived complexity that from its very nature it is less accurate. Secondly, project managers were asked to evaluate project complexity using a linear scale with integer values, e.g., 1, 2, 3, etc., while model calculation allows the use of real number values in outcome. Obviously, an error margin of ±5% already exists because of the rounding from real numbers to integer numbers (e.g., all numbers from 4.5 to 5.4 are rounded up to 5, if rounded with no decimals). Further, most software measurement methods used consider acceptable an error margin of ±15% (e.g., function points, use case points).

Table 6 Case study – results of model validation process

Project name	*Project complexity as calculated by the proposed model*	*Perceived project complexity value as determined by the project's managers*	*Difference between the two complexity values (considering the margin of error)*
Project 1	5.78	7	12.2% (or 1.22 units)
Project 2	3.70	6	23.0% (or 2.3 units)
Project 3	5.72	6	2.8% (or 0.28 units)
Project 4	5.68	7	13.2% (or 1.32 units)
Project 5	5.17	6	8.3% (or 0.83 units)

The results of the case study as can be seen in Table 6, was encouraging about model validity. In 4 of 5 cases examined, the difference between the level of complexity calculated by the model and the level of perceived complexity experienced by project managers was less than 15% while in one case it was 23%, with average deviation of 11.9%. In the second project, the difference between the two values is 23%, which was outside the defined error margin, although it cannot be considered too big. In order to investigate the causes for this discrepancy, the basic projects characteristics were re-examined as they described in the project chart document. In that documented it was noted that two of the main project constraints and risks existed, were the significant lack of digital data and the variety in legislation that lead to disordered legal foundation of the city plan. Especially the second, broke one of the basic project design assumptions that was a stable city plan and a solid legislation. This resulted in complicated requirements and software design, which probably affected the whole project process.

In the proposed model, two factors had been identified, aiming to capture these situations. The first is included in the integration subject area named 'IM1: integration constraints due to project characteristics' and the second can be found in technical aspects of software development area, named 'SD2: product development constraints' that can encompass situations like this. The first factor was assessed quite high, while the second factor was assessed very low by the project manager, meaning that probably there was a misunderstanding in the factor semantics or a failure in assessment by the project manager. A higher assessment of this value would improve results, although it would not eliminate the difference between calculated and perceived complexity. This may indicate a problem in factors weighting, that needs to be further examined, but the other results weaken this case. Another reason could be that the density of this problem overshadowed the whole project process and affected the judgement of the project manager. However, the extensive experience of the project manager weakens this explanation. Thus, accurate estimates of the real causes cannot be safely extracted from the current case study results and, as such, further examination is needed by applying the model to more projects having similar problems in order to clarify, if the model underestimates these situations or the problem must be identified elsewhere.

Another point that this case study indicates, is the validity of the approach that project complexity is subjective and dependent on the cognitive level of the observer. Project complexity was evaluated at level 6 or 7 in all projects examined, despite their differences. For example, projects 3 and 5 had both been evaluated with complexity level 6 despite their huge differences in duration, budget, number and type of stakeholders, geographical distribution and type of software being developed. This does not mean in general that a relatively small project with strict constraints cannot be more complex that a larger one with more relaxed constraints. However, this is not this case, as can be concluded by the study of project charter. The encouraging point was that the model was able to capture this subjective evaluation of complexity by project managers while simultaneously it managed to capture and to indicate the difference in complexity levels between projects as can be seen from the results.

6 Conclusions – limitations

This research proposes a model aiming in assessing software projects complexity based on of project management subject areas and of technical software development process factors. The proposed model provides a link between project complexity and project management and identifies project complexity from a new approach. Provides a complete complexity typology, determines sources of project complexity and identifies 35 complexity factors that contribute to project complexity that can easily be understood by project managers and assessed at early project stages. Its compatibility with PMBOK management areas, allows project managers to evaluate project complexity in a manner that is familiar to the way they approach project execution.

The analysis of the final set of factors indicates that there are three factors that affect almost all complexity dimensions. First, is the ‘organisation’s management capability’, which is defined as the capability of a project organisation to perform the various project management and technical tasks of software development efficiently and effectively. Second is the ‘density of various project processes’, which is referred to the number, variance, frequency and interdependencies of project elements and third, the ‘existence of various constraints’, which is referred to the various constraints exists in project management and software development processes. The rest factors identified are more specific to aspects of each complexity dimension. Considering that in current literature the majority of researches approach project complexity through the aspects of uncertainty, interdependency, number and variety of elements, technical factors, etc., it is encouraging that the identified complexity factors of this research have actually embedded these factors although the approach to project complexity followed here is different.

The findings of this research highlight the significance of the human factor and of tacit knowledge and as well the maturity management of an organisation. The approach followed in model design, accepts the fact that there is variation of perception of complexity between different organisations and in different points of time, so it allows organisations to evaluate complexity of projects accordingly.

This research considers complexity as something tangible and not abstract, as variable that can be measured and a model for measuring it is proposed. The model can be used to assess complexity in practical terms and allows the evaluation of project complexity not only as an entity but also in partial through the eleven complexity areas. By that, organisations can determine management areas that are of higher complexity in comparison with other areas and to focus their efforts on these areas.

The defined measures are quantitative and allow users to express their objective evaluation. Its structure is simple implying that it can be easily understood by users. Finally, it is easy to calculate as not any special mathematical knowledge and skills are required to perform the calculations.

It may be argued that the scales used to assess the complexity factors are not fully quantitative, as they are not defined by specific boundaries for each choice. Because of that, the scale can be characterised as semi-quantitative, allowing a level of subjectivity on the answers. However, as has been discussed in Section 2.2, complexity in projects is subjective. Different users with different experiences, knowledge levels, backgrounds, and personalities may have different perceptions of complexity. Furthermore, different organisations with different characteristics in size, domain knowledge, human resources, experience, etc., will evaluate the complexity of a specific project differently because of all these different characteristics. However, when each one of these organisations needs to evaluate the complexity of two or more projects that they are interested in undertaking, it will evaluate each complexity factor proposed in this model using the same subjective criteria for all projects. This will allow each organisation to compare the expected complexity of projects in order to make the most suitable selection using the same subjective approach. The fact that different organisations will probably evaluate the complexity of a project differently is not significant for the organisation itself. Thus, the model proposed in this research should grasp this subjectivity, and the structure of the scale used to assess the complexity factors allows that.

A limitation of this research stems from the fact that the findings are to some extent country-specific since only 10% of data collected through surveys were from countries outside Greece. Another limitation is the assumption that all 11 complexity areas are of equal weights. However, these limitations can be lifted in future research., As for the first, results obtained from future international surveys, following the same methodology, can be compared with the results of this survey. As for the other limitation, the proposed model allows the adjustment of assigned weights easily with weight values that may occur from future research or with values that are more suitable to different types of projects.

In summary, the proposed complexity model is compatible with the way most modern projects are managed, granular as allows project complexity to be measured either at the level of a project or per complexity area, user-friendly as allows calculations to be made quickly and easily, independent of software development methods due to its focus on project management and technical aspects of the software development process that are independent of the development model and allows early management of project complexity as management data are available at early project phases.

References

Ahmad, N. and Laplante, P.A. (2006) 'Software project management tools: making a practical decision using AHP', *2006 30th Annual IEEE/NASA Software Engineering Workshop*, IEEE, Columbia, MD, pp.76–84.

Albrecht, A.J. (1979) 'Measuring application development productivity', *Proc. of IBM Applications Development Symposium*, October, pp.83–89.

Al-Harbi, K. (2001) 'Application of the AHP in project management', *International Journal of Project Management*, Vol. 19, No. 1, pp.19–27.

Alhazmi, T. and McCaffer, R. (2000) 'Project procurement system selection model', *Journal of Construction Engineering and Management*, Vol. 26, No. 3, pp.176–184.

Altahtooh, U.A. and Emsley, M.W. (2017) 'An introduction to project end theory in project management', *International Journal of Information Technology Project Management*, Vol. 8, No. 3, pp.69–81.

Alves, N.S. et al. (2016) 'Identification and management of technical debt: a systematic mapping study', *Information and Software Technology*, Vol. 70, No. C, pp.100–121.

APM (2008) *APM Competence Framework*, Association for Project Management, Bucks, UK.

Arevalo, E. and Espinosa, A. (2015) 'Theoretical approaches to managing complexity in organizations: a comparative analysis', *Estudios Gerenciales*, Vol. 31, No. 134, pp.20–29.

Aydin, M.N. and Dilan, E. (2017) 'Project management method adoption: a service industry case study', *International Journal of Information Technology Project Management*, Vol. 8, No. 2, pp.17–33.

Baccarini, D. (1996) 'The concept of project complexity – a review', *International Journal of Project Management*, Vol. 14, No. 4, pp.201–204.

Bakhshi, J., Ireland, V. and Gorod, A., (2016) 'Clarifying the project complexity construct: past, present and future', *International Journal of Project Management*, Vol. 34, No. 7, pp.1199–1213.

Beavers, A.S. et al. (2013) 'Practical considerations for using exploratory factor analysis in educational research', *Practical Assessment, Research & Evaluation*, Vol. 15, No. 6, pp.1–13.

Boehm, B. et al. (2000) *Software Cost Estimation with COCOMO II*, Prentice Hall, Englewood Cliffs.

Bolat, B., Kuşdemir, A., Uslu, I.C. and Temur, G. (2017) 'An assessment for IT project maturity levels', *International Journal of Information Technology Project Management*, Vol. 8, No. 2, pp.1–16.

Bosch-Rekveldt, M. et al. (2011) 'Grasping project complexity in large engineering projects: the TOE (technical, organizational and environmental) framework', *International Journal of Project Management*, Vol. 29, No. 6, pp.728–739.

Broche, G. and Marinescu, P. (2008) *Deschideri spre lumea complexit ă ìii*, Ed. Universită ìii din Bucureèti.

Cardona-Meza, L.S. and Olivar-Tost, G. (2017) 'Modeling and simulation of project management through the PMBOK® standard using complex networks', *Complexity*, Vol. 2017, Article ID 4791635, p.12.

Cattell, R.B. (1996) 'The scree test for the number of factors', *Multivariate Behavioral Research*, Vol. 1, No. 2, pp.245–276.

Chapman, R. (2016) 'A framework for examining the dimensions and characteristics of complexity inherent within rail megaprojects', *International Journal of Project Management*, Vol. 34, No. 6, pp.937–956.

Charette, R.N. (2005) 'Why software fail', *IEEE Spectrum*, Vol. 42, No. 9, pp.42–49.

Child, D. (2006) *The Essentials of Factor Analysis*, 3rd ed., Continuum International Publishing Group, New York, NY.

Cicmil, S. and Marshall, D. (2005) 'Insights into collaboration at the project level: complexity, social interaction and procurement mechanisms', *Building Research & Information*, Vol. 33, No. 6, pp.523–535.

Cooke-Davies, T., Cicmil, S., Crawford, L. and Richardson, K., (2007) '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, pp.50–61.

Costelo, A.B. and Osborne, J. (2005) 'Best practices in exploratory factor analysis: four recommendations for getting the most from your analysis', *Practical Assessment, Research & Evaluation*, Vol. 10, No. 7, pp.1–9.

Cronbach, L. (1951) 'Coefficient alpha and the internal structure of tests', *Psychometrika*, Vol. 16, No. 3, pp.297–334.

Daim, T.U. et al. (2012) 'Exploring the communication breakdown in global virtual teams', *International Journal of Project Management*, Vol. 30, No. 2, pp.199–212.

Damasiotis, V. (2018) *Modelling Software Project Management Complexity – An Assessment Model*, Staffordshire University, Stoke on Trent.

Damasiotis, V., Fitsilis, P. and Considine, P.O.K.J. (2017) 'Analysis of software project complexity factors', *Proceedings of the 2017 International Conference on Management Engineering, Software Engineering and Service Sciences*, ACM, Wuhan, China, pp.54–58.

DeCoster, J. (1998) *Overview of Factor Analysis* [online] http://www.stat-help.com/factor.pdf (accessed 8 June 2016).

Dombkins, D. and Dombkins, P. (2008) *Contracts for Complex Programs: A Renaissance of Process*, Booksurge Publishing, Charleston.

Dooley, K. (2002) 'Organizational complexity', in Warner, M. (Ed.): *International Encyclopedia of Business and Management*, pp.5013–5022, Thompson Learning, London.

Dunovic, I.B., Radujkovic, M. and Skreb, K.A. (2014) 'Towards a new model of complexity – the case of large infrastructure projects', *Procedia – Social and Behavioral Sciences*, Vol. 119, pp.730–738.

Field, A. (2009) *Discovering Statistics Using SPSS: Introducing Statistical Method*, 3rd ed., Sage Publications, Thousand Oaks, CA.

Fioretti, G. and Visser, B., (2004) 'A cognitive interpretation of organizational', *E:CO Special Double Issue*, Vol. 6, Nos. 1–2, pp.11–23.

Fitsilis, P., Kameas, A. and Anthopoulos, L. (2010) 'Classification of software projects' complexity', *Information Systems Development*, pp.149–159, Springer New York, New York, NY.

Floricel, S., Piperca, S. and Tee, R., (2018) 'Strategies for managing the structural and dynamic consequences of project complexity', *Complexity*, Vol. 2018, Article ID 3190251, p.17.

Geraldi, J. and Albrecht, G. (2007) 'On faith, fact, and interaction projects', *Project Management Journal*, Vol. 38, No. 1, pp.32–43.

Geraldi, J., Maylor, H. and Williams, T. (2011) 'Now, let's make it really complex (complicated): a systematic review of the complexities of projects', *International Journal of Operations & Production Management*, Vol. 31, No. 9, pp.966–990.

Ghazarian, A. (2015) 'A theory of software complexity', *IEEE/ACM 4th SEMAT Workshop on a General Theory of Software Engineering*, pp.29–32.

Hagan, G., Bower, D. and Smith, N. (2011) 'Managing complex projects in multi-project environments', in Egbu, C. and Lou, E. (Eds.): *Procs 27th Annual ARCOM Conference*, Association of Researchers in Construction Management, Bristol, UK, pp.787–796.

Hass, K. (2007) 'Introducing the project complexity model. A new approach to diagnosing and managing projects. Part 1 of 2', *PMWorld Today*, Vol. 9, No. 7, pp.pp–1-8.

He, Q., Luo, L., Hu, Y. and Chan, A.P. (2015) 'Measuring the complexity of mega construction projects in China – a fuzzy analytic network process analysis', *International Journal of Project Management*, Vol. 33, No. 3, pp.549–563.

Hertogh, M. and Westerveld, E. (2010) 'Playing with complexity – management and organization of large infrastructure projects [online] http://hdl.handle.net/1765/18456 (accessed 11 October 2016).

Hongyan, L. (2010) 'The analysis of project risk management based on AHP evaluation', *2010 2nd IEEE International Conference on Information and Financial Engineering*, IEEE, Chongqing, pp.576–579.

Hutcheson, G. and Sofroniou, N. (1999) *The Multivariate Social Scientist*, Sage, London.

Janczarek, P. and Sosnowski, J. (2015) 'Managing complex software projects', *Information Systems in Management*, Vol. 4, No. 3, pp.171–182.

Jiang, Z. and Naude, P. (2007) 'An examination of the factors influencing software development effort', *International Journal of Computer, Information, and Systems Science, and Engineering*, Vol. 1, No. 3, pp.182–191.

Kaiser, H.F. (1970) 'A second generation little jiffy', *Psychometrika*, Vol. 35, pp.401–415.

Karner, G. (1993) *Metrics for Objectory*, Diploma thesis, No. LiTHIDA-Ex-9344:21, University of Linköping, Sweden.

Kermanshachi, S., Dao, B., Shane, J. and Anderson, S. (2015) 'An empirical study into identifying project complexity', *Procedia Engineering*, Vol. 145, pp.603–610.

Keshavarz, G., Modiri, N. and Pedram, M. (2011) 'Metric for early measurement of software complexity', *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 3, No. 6, pp.2482–2490.

Khan, A.A., Mahmood, A., Amralla, S.M. and Mirza, T.H. (2016) 'Comparison of software complexity metrics', *International Journal of Computing and Network Technology*, Vol. 4, No. 1, pp.19–26.

Kushwaha, D.S. and Misra, A.K. (2006) 'A complexity measure based on information contained in software', *Proceedings of the 5th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems*, WSEAS, Madrid, Spain, pp.187–195.

Lee, G. and Xia, W., (2002) 'Development of a measure to assess the complexity of information systems development projects', *ICIS 2002 Proceedings*, Paper 8.

Lin, C., Wang, W. and Yu, W. (2008) 'Improving AHP for construction with an adaptive AHP approach', *Automation in Construction*, Vol. 17, No. 2, pp.180–187.

Lu, Y. et al. (2014) 'Measurement model of project compelxity for large-scale projects from task and organisation perspective', *International Journal of Project Management*, Vol. 33, No. 3, pp.610–622.

Marengo, A., and Pagano, A. (2020) 'Innovative ways to assess soft-skills: the in-basket game online experience', *European Conference on e-Learning*, October, pp.325–XVII, Academic Conferences International Limited.

Maylor, H., Vidgen, R. and Carver, S. (2008) 'Managerial complexity in project based operations: a grounded model and its implications for practice', *Project Management Journal*, Vol. 39, No. S1, pp.S15–S26.

McLeod, L. and MacDonell, S.G. (2011) 'Factors that affect software systems development project outcomes: a survey of research', *ACM Computing Surveys (CSUR)*, Vol. 43, No. 4, pp.0360–0300.

Montequín, V.R., Balsera, J.V., Fernández, S.M.C. and Fernández, F.O. (2018) 'Exploring project complexity through project failure factors: analysis of cluster patterns using self-organizing maps', *Complexity*, Vol. 2018, Article ID 9496731, p.17.

Neleman, J. (2006) 'Shell gaat diep', *FEM Bus.*, Vol. 9, No. 4, pp.30–34.

Nguyen, A.T., Nguyen, L.D., Long, L. and Dang, C.N. (2015) 'Quantifying the complexity of transportation projects using the fuzzy analytic hierarchy process', *International Journal of Project Management*, Vol. 33, No. 6, pp.1364–1376.

Oehmen, J., Thuesen, C., Ruiz, P. and Geraldi, J. (2015) 'Complexity management for projects, programmes, and portfolios. An engineering systems perspective', *PMI Global Congress 2015 – EMEA*, Project Management Institute, Newtown Square, London, England.

Pallant, J. (2011) *SPSS Survival Manual: A Step by Step Guide to Data Analysis Using SPSS*, 4th ed., Allen & Unwin, Crows Nest, Australia.

Patanakul, P., Milosevic, D. and Anderson, T. (2007) 'A decision support model for project manager assignments', *IEEE Transactions on Engineering Management*, Vol. 54, No. 3, pp.548–564.

PMI (2013) *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, 5th ed., Project Management Institute, Inc., s.l.

Poveda-Bautista, R., Diego-Mas, J.-A. and Leon-Medina, D., (2018). Measuring the Project Management Complexity: The Case of Information Technology Projects. *Complexity,* 2018(Article ID 6058480), p. 19.

Qazi, A., Quigley, J., Dickson, A. and Kirytopoulos, K. (2016) 'Project complexity and risk management (ProCRiM): towards modelling project complexity driven paths in construction projects', *International Journal of Project Management*, Vol. 34, No. 7, pp.1183–1198.

Rahman, N., Wittman, A. and Thabet, S. (2016) 'Managing an engineering project', *International Journal of Information Technology Project Management*, Vol. 7, No. 1, pp.1–17.

Reiman, T., Rollenhagen, C., Pietikäinen, E. and Heikkilä, J. (2015) 'Principles of adaptive management in complex safety – critical organizations', *Safety Science*, Vol. 71, Part B, pp.80–92.

Remington, K., Zollin, R. and Turner, R. (2009) 'A model of project complexity: distinguishing dimensions of complexity from severity', *The 9th International Research Network of Project Management Conference (IRNOP-IX)*, IRNOP, Berlin.

Rezvani, A. and Khosravi, P. (2019) 'Identification of failure factors in large scale complex projects: an integrative framework and review of emerging themes', *International Journal of Project Organisation and Management*, Vol. 11, No. 1, pp.1–21.

Ribbers, M. and Schoo, K.C., (2002) 'Designing complex software implementation programs', *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pp.3391–3401.

Saaty, T. (1980) *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, McGraw-Hill, New York, NY.

Saunders, M., Lewis, P. and Thornhill, A., (2012) *Research Methods for Business Students*, 6th ed., Pearson Education Limited, s.l.

Schlindwein, S.L. and Ison, R. (2004) 'Human knowing and perceived complexity: implications for systems practice', *Emergence: Complexity & Organization*, Vol. 6, No. 3, pp.27–32.

Schuh, G., Rebentisch, E., Riesener, M. and Mattern, C. (2017) 'Method for the evaluation and adaptation of new product development project complexity', *Procedia CIRP*, Vol. 60, pp.338–343.

Sedaghat-Seresht, A., Fazli, S. and Mozaffari, M. M., (2012). Using DEMATEL Method to Modeling Project Complexity Dimensions. *Journal of Basic and Applied scientific Research*, 2(11), pp.11211-11217.

Sharma, A. and Kushwaha, D. (2010) 'A complexity measure based on requirement engineering document', *Journal of Computer Science and Engineering*, Vol. 1, No. 1, pp.112–117.

Simpson, G.W. and Cochran, J.K. (1987) 'An analytical approach to programming construction projects', *Civil Engineering Systems*, Vol. 4, No. 4, pp.185–190.

Snedecor, G.W. and Cochran, W.G. (1989) *Statistical Methods*, 8th ed., Iowa State University Press, Iowa.

Stevenson, D. and Starkweather, A. (2017) 'IT project success: the evaluation of 142 success factors by IT PM professionals', *International Journal of Information Technology Project Management*, Vol. 8, No. 3, pp.1–21.

Tatikonda, M.V. and Rosenthal, S.R. (2000) 'Technology novelty, project complexity, and product development project execution success: a deeper look at task uncertainty in product innovation', *IEEE Transactions on Engineering Management*, Vol. 47, No. 1, pp.74–87.

The Standish Group (2009) *CHAOS Summary 2009: The 10 Laws of CHAOS* [online] https://www.classes.cs.uchicago.edu/archive/2014/fall/51210-1/required.reading/Standish.Group.Chaos.2009.pdf (accessed 23 October 2016).

The Standish Group (2015). *CHAOS Report*, The Standish Group, West Yarmouth.

Tie, B.N. and Bolluijt, J. (2014) 'Measuring project complexity', *2014 9th International Conference on System of Systems Engineering (SOSE)*, IEEE, Adelade, Australia, pp.248–253.

Trendowicz, A. and Münch, J. (2009) 'Chapter 6 Factors influencing software development productivity – state-of-the-art and industrial experiences', *Advances in Computers*, Vol. 77, pp.185–241, Elsevier.

Truong, D. and Jitbaipoon, T. (2016) 'How can agile methodologies be used to enhance the success of information technology projects?', *International Journal of Information Technology Project Management*, Vol. 7, No. 2, pp.1–16.

Vaidya, O.S. and Kumar, S. (2006) 'Analytic hierarchy process: an overview of applications', *European Journal of Operational Research*, Vol. 169, No. 1, pp.1–29.

Velayudhan, D.P. and Thomas, S. (2018) 'Role of technological uncertainty, technical complexity, intuition and reflexivity in project planning – a study on software development projects', *International Journal of Project Organisation and Management*, Vol. 10, No. 1, pp.82–92.

Vidal, L. and Marle, F. (2008) 'Understanding project complexity: implications on project management', *Kybernetes*, Vol. 37, No. 8, pp.1094–1110.

Vidal, L.A., Marle, F. and Bocquet, J. (2011) 'Using a Delphi process and the analytic hierarchy process (AHP) to evaluate the complexity of projects', *Expert Systems with Applications*, Vol. 38, No. 5, pp.5388–5405.

Wagner, S. and Ruhe, M. (2008) *A Structured Review of Productivity Factors in Software Development*, TUM – Institut fur Informatik, Munchen.

Ward, S. and Chapman, C. (2003) 'Transforming project risk management into project uncertainty management', *International Journal of Project Management*, Vol. 21, No. 2, pp.97–105.

Whitty, S.J. and Maylor, H. (2009) 'And then came complex project management (revised)', *International Journal of Project Management*, Vol. 27, No. 3, pp.304–310.

Williams, T. (2002) *Modelling Complex Projects*, John Wiley & Sons, London.

Williams, T.M. (1999) 'The need for new paradigms for complex projects', *International Journal of Project Management*, Vol. 17, No. 5, pp.269–273.

Williams, T.M. (2005) 'Assessing and moving on from the dominant project management dicourse in the light of project overruns', *IEEE Transactions on Engineering Management*, Vol. 52, No. 4, pp.497–508.

Xia, W. and Lee, G. (2005) 'Complexity of information systems development projects: conceptualization and measurement development', *Journal of Management Information Systems*, Vol. 22, No. 1, pp.45–83.

Yong, A.G. and Pearce, S. (2013) 'A beginner's guide to factor analysis: focusing on exploratory factor analysis', *Tutorials in Quantitative Methods for Psychology*, Vol. 9, No. 2, pp.79–94.

Zuse, H. (1990) *Software Complexity Measures and Models*, de Gruyter and Co., New York, NY.