

REQUIREMENTS FOR SYSTEMS DEVELOPMENT LIFE CYCLE MODELS FOR LARGE-SCALE DEFENSE SYSTEMS

Kadir Alpaslan DEMIR

PhD, Assistant Program Manager, Turkish Naval Research Center Command,
Istanbul, Turkey

Large-scale defense system projects are strategic for maintaining and increasing the national defense capability. Therefore, governments spend billions of dollars in the acquisition and development of large-scale defense systems. The scale of defense systems is always increasing and the costs to build them are skyrocketing. Today, defense systems are software intensive and they are either a system of systems or a part of it. Historically, the project performances observed in the development of these systems have been significantly poor when compared to other types of projects. It is obvious that the currently used systems development life cycle models are insufficient to address today's challenges of building these systems. Using a systems development life cycle model that is specifically designed for large-scale defense system developments and is effective in dealing with today's and near-future challenges will help to improve project performances. The first step in the development a large-scale defense systems development life cycle model is the identification of requirements for such a model. This paper contributes to the body of literature in the field by providing a set of requirements for system development life cycle models for large-scale defense systems. Furthermore, a research agenda is proposed.

Key words: defense systems, large-scale defense systems, characteristics of defense systems, system development models, software development models, requirements.

1. INTRODUCTION

Countries spend billions of dollars on defense spending [1]. A significant portion of this spending goes into the acquisition and development of large-scale defense systems (LSDSs). Considering the amount of resources used in the acquisition and development of these systems, surprisingly, the scientific literature on the topic is quite limited. Most of the current literature consists of books and magazine articles written by defense practitioners and reports from government agencies. The models, processes, tools, and techniques used in the development of defense systems have not improved much. For example, the Waterfall system life cycle development model [2] and the V model of system development are among the most commonly used models in defense

industry. Both of them were developed decades ago based on the needs of their time. Naturally, defense systems are evolving over time. For example, defense systems were not software intensive in the past. Today, they are. See **Table 1** for the evolution of military aircrafts in terms of software intensity.

Table 1. System functionality performed in software
Source: [3] [4]

Defense System – Military Aircrafts	Year	% Functions Performed in Software
F-4	1960	8
A-7	1964	10
F-111	1970	20
F-15	1975	35
F-16	1982	45
B-2	1990	65
F-22	2000	80
F-35 Lightning II	2012	90

As evidenced with many reports presenting the poor performance in defense projects [7], the currently used models are ineffective in dealing with today's challenges in LSDS development. While the defense industry has failed in advancing the systems development processes and models, the civilian software industry was successful in responding to evolving challenges of software development. Many variants of agile methods for software development were developed [5]. The reports indicate that agile methods have contributed to productivity in civilian software industry [5]. In this respect, Jones [6] reports that the productivity in defense software development is noticeably low. Furthermore, as the software scale increases, the rates of project cancellation increases, too, while the productivity in military software industry decreases [6]. Last but not the least, the defense community is conservative in adapting best practices from civilian industry [6].

Naturally, there have been attempts at improving the defense project performances through a series of initiatives [7]. The US Department of Defense (DoD) sponsored the Software Engineering Institute (SEI) for finding various solutions including the famous Capability Maturity Model (CMM) series. CMMs are series of models used to assess the maturity of system and software developing organizations. In addition, government defense ministries and agencies supported the development of various enterprise architecture frameworks (EAF). US DoD's Department of Defense Architecture Framework (DODAF), the British Ministry of Defence Architecture Framework (MODAF), and NATO Architecture Framework (NAF) are among such efforts. Object Management Group's Unified Profile for DODAF/MODAF (UPDM) is an attempt to combine these architecture frameworks. The purpose of these EAFs is to guide the development of defense system of systems projects.

Recently, Boehm and his colleagues developed the incremental commitment model (ICM) for software development [8].

The model is developed based on the critical success factor principles and the strengths of various other development models such as V model, spiral model, agile methods, etc. It is claimed that the model is effective in various system developments including defense system developments [9]. Furthermore, it is emphasized that the ICM milestones is compatible with US DoD acquisition milestones [9]. However, the ICM model has not been widely tested in defense industry. Therefore, the model performance is unknown.

As the expectations from systems is increasing, new system development life cycles are developed. Various agile models [5] and ICM [8] are among the examples. Various reports and studies also identify the current advantages of models and challenges observed in the implementations of these models. However, to our knowledge, the requirements for systems development life cycles models for LSDSs have not been researched in detail. This study aims to contribute to this area of the literature.

The rest of the article is as follows. In the second section, we list the main characteristics of LSDSs. Next we identify the characteristics of LSDS projects. In the fourth section, challenges related to the development of LSDSs are discussed. The following section lists the requirements for systems development cycle models for LSDSs. These requirements are derived from LSDS characteristics and LSDS project characteristics and challenges.

2. MAIN CHARACTERISTICS OF LARGE-SCALE DEFENSE SYSTEMS

In this section, the main characteristics of LSDSs are identified and briefly discussed. While some of these characteristics are shared with civilian systems of the same size, some of the characteristics are observed in only few civilian systems. While only a portion of civilian systems are safety and mission critical, almost all defense systems are mission-critical

and most defense systems are safety-critical. Development of LSDSs are costly and challenging due to following characteristics:

1. LSDSs are large-scale.
2. LSDSs are software intensive.
3. LSDSs are safety-critical.
4. LSDSs are mission-critical.
5. LSDSs are system of systems.
6. LSDSs should be high quality.
7. LSDSs are complex.
8. LSDSs have long life cycles.

David Lorge Parnas wrote a paper [10] in 1985 before resigning from the Panel on Computing in Support of Battle Management, convened by the Strategic Defense initiative Organization (later renamed as Ballistic Missile Defense Organization). It became a controversial paper in the defense community. It outlines the “software aspects of strategic defense systems.” The paper stimulated discussions among defense community whether building trustworthy large-scale defense systems is feasible or not.

LSDSs are large-scale. The scale in LSDSs is increasing [11] as the defense needs and expectations are increasing. Development of large-scale systems has always been challenging. Historically, on average, defense systems are larger than civilian systems [6]. Thus, the defense community have experience in developing large-scale systems compared to civilian industry [6]. However, we have yet to see an upward project performance trend in LSDS developments. Jones reports that as the scale increases in military software, productivity significantly lowers [6].

LSDSs are software intensive. Today, software is the major component in any defense system [12-14]. The success of a weapon system is dependent on the success of the system software [12,13,16,17]. In 1974, the F-16A included 135 thousands of source lines of code (SLOC). In 2012, F-35 includes 24 million SLOC [13]. Software development by itself is difficult due to some inherent properties (essential difficulties) [15]. The defense context increases the challenge to a higher level. Therefore, software related problems are dominating the majority of defense

project problems. For example, as one of the major defense projects, F-35 fighter aircraft development is reported to be plagued with software related problems [16]. Many major weapon systems deliveries are delayed due to a magnitude of software and quality problems.

LSDSs are safety-critical. A safety-critical system may be defined as “a system whose failure may cause injury or death to human beings” [18]. A significant portion of defense systems are weapon systems and naturally safety-critical systems. Development of safety critical systems is hard [18, 19] and requires a safety perspective from the start supported by a rigorous system safety program. Ensuring system safety requires rigorous design, analysis, and testing, all of which contributes to high costs. A defense system cannot be used unless system safety is ensured. The warfighters should be able to use these systems without the fear of harming friendly forces or themselves.

LSDSs are mission-critical. A mission-critical system is a type of system in which the failure may result in not achieving a critical goal, significant loss in terms of money, or trust in the system [18]. In the defense context, the failure of a mission-critical system may cause a failure in a mission or a limitation in the defense capability temporarily or permanently. Development of mission-critical systems is challenging in many aspects [18].

LSDSs are a system of systems (SoS). The definition of SoS in the Defense Acquisition Guide [20] is “A system of systems (SoS) is defined as a set or arrangement of systems that results from independent systems integrated into a larger system that delivers unique capabilities”. The technological advancements in computing systems and especially networks led to the development of a system of systems [23]. “Network centric warfare” is one of the military concepts introduced to increase defense capability utilizing the system of systems approach [21]. Using SoS, the armed forces expect new capabilities that individual systems comprising the SoS cannot offer alone.

While the benefits of SoS are appealing, the development of SoS has many challenges.

LSDSs should be high quality. A defense system should be trustworthy and have high quality [22]. Ensuring complete trustworthiness especially in large-scale systems is considered unlikely to be achievable by some researchers, while others believe that it is possible [10]. In addition to trustworthiness (that includes attributes such as dependability, reliability, etc.), usability, supportability (through open architectures), maintainability, security, safety, testability, evolvability, fault tolerance, interoperability, survivability, high performance, efficiency, and effectiveness are among other qualities to be expected from LSDSs.

LSDSs are complex. National defense needs are increasing. The expectations of warfighters from defense systems is also increasing as the warfighters see the recent technological advancements in civilian applications. Satisfying the ever-increasing defense needs, expectations, and a significant amount of functionality with high quality SoS defense systems leads to complexity in defense systems. The development of complex systems poses many challenges [19].

LSDSs have long life cycles. The costs are so high and schedules are so long that replacing LSDSs in short periods is economically unsustainable. Defense systems such as ships, military aircrafts, tanks, missiles etc. are expected to be in service for at least 30-40 years. Currently, the F-35 is planned to have a 50-years long life cycle [13]. Naturally, there are upgrade programs over the years to prolong the service life in addition to overhauls and maintenance. Supportability, maintainability, and evolvability are among the quality concerns for systems having long life cycles. An important challenge results from the difference in the rate of evolution between hardware and software. Hardware is evolving much faster than software. Acquiring legacy hardware is expensive if possible. Vendors quickly adapt new manufacturing technologies to stay competitive.

3. MAIN CHARACTERISTICS OF LARGE-SCALE DEFENSE SYSTEM PROJECTS

The development of LSDSs is challenging due to the following characteristics:

1. LSDS projects are long.
2. LSDS projects are costly.
3. LSDS projects are risky.
4. LSDS projects are developed based on government regulations.
5. LSDS projects are verification and validation (V&V) oriented.

LSDS projects are long. A defense system is usually delivered in 5 to 10 years [24]. The time to develop a LSDS may take a decade or more [25]. In general, the development of defense systems is a long and expensive effort [26]. Large-scale, complexity, government acquisition procedures, the amount of required functionality, high quality expectations, slow development, the need for extensive testing, and proof of compliance with many standards are among the factors contributing to long development cycle.

LSDS projects are costly. When the systems are large-scale, complex, expected to have high quality, and the development cycle is long, the high costs are inevitable. Unless these characteristics change and affordable and effective solutions are found to these challenges, the development of LSDSs will be costly. Currently, the cost of defense systems is increasing [7] and this trend is not expected to change in near-future.

LSDS projects are risky. LSDS projects are among the type of projects that have the highest cancellation rates [6]. Based on the statistics provided by Jones, in military as the project scale goes up, the rate of success falls dramatically [6, 27]. While, only 10% of defense software with a size of 1,000 function points is facing cancellation, the rate is 33% when the size of the military software reaches 100,000 function points [6]. Function point is a measure of provided functionality. There are also many projects that are delivered with less functionality than planned and

with quality problems. According to a 2015 GAO report [28] on high risk list, “Many DOD programs are still falling short of cost, schedule, and performance expectations.” The US GAO started reporting the high risk areas in 1990. Since 1990, major weapons acquisitions are in the US GAO’s high list risk updated every two years [29]. What is more, the software and IT projects are challenged in scope management [36]. When the scope is not clear in the beginning, then many risks are introduced to the project.

LSDS projects are developed based on government regulations. The acquisition of defense systems has to go through the government defense acquisition process. The management of defense acquisitions is burdensome, inefficient and bureaucratic [30,31]. Defense projects have some noticeable differences compared with civilian norms [6]: The procurement process, the litigation problems, the adversarial relationship between DoD and contractors. More than half of the military contracts are challenged by disgruntled competitors, leading to litigation [6]. Resolving the litigations and starting the project may cause a delay of 6 to 18 months [6]. Therefore, the military projects are late even before project start. The amount of specifications and documentation produced in a defense system project is three times larger than civilian projects [32]. The production and review of documentation is a major cost element in a defense project. A significant portion of the documentation consists of reports for government project monitoring and control. In LSDS development, the contractors are required to develop the system based on many standards [18]. While compliance with these standards contributes to achieving high quality systems, they also increase the cost and time to build the system [18]. In addition, LSDS projects have a high number of stakeholders [18]. The stakeholders include armed forces, department of defenses, military personnel, government acquisition agencies, etc. Satisfying this number of stakeholders with different motivations and expectations, sometimes conflicting,

requires hard work with political and social skills.

LSDS projects are verification and validation (V&V) oriented. The LSDS projects are strategic due to their contribution to defense capability. Also, the development of LSDS is costly; LSDSs are mission and safety-critical; LSDS are expected to be high quality; the number of stakeholder involved is high. These and other factors result in the necessity of a verification and validation oriented acquisition and development process. At project milestones and various phases of the project, the contractors have to show that the system under development is valid and verified. This is achieved by the reviews [33] such as conceptual design reviews, preliminary design reviews, critical design reviews. Unless these reviews are satisfactory in these milestones, the development cannot progress.

4. REQUIREMENTS FOR LARGE-SCALE DEFENSE SYSTEMS DEVELOPMENT LIFE CYCLE MODELS

Based on the analysis of LSDS and project characteristics, we identify a set of high-level requirements. These requirements are listed in Table 2. Note that these are high-level and further development of low-level requirements are also essential. For example, the requirements such as “The SDLCM shall support good project management practices.” should be refined. Project management success is important for projects success [34]. The authoritative reference document in project management is the Project Management Body of Knowledge [35] (PMBOK) developed by Project Management Institute (PMI). Latest PMBOK includes 10 knowledge areas (KA):

1. Project Integration Management
2. Project Scope Management
3. Project Time Management
4. Project Cost Management
5. Project Quality Management
6. Project Human Resource Management

7. Project Communications Management
8. Project Risk Management
9. Project Procurement Management
10. Project Stakeholders Management

How to support these 10 KAs in the life-cycle development model needs further research. Another requirement “The SDLCM shall be compatible with government acquisition policies.” should also be detailed. The government acquisition policies are different for different countries. However, most countries adapt the policies and practices of the US government since, US is the leading and major producer and consumer of defense systems and software [6]. Naturally, other countries try to benefit from these experiences. Furthermore, it may be possible to develop a defense acquisition framework compatible

with many national government acquisitions. Such a framework may help multi-national defense acquisitions. Development of this framework may be a good research topic.

One of the most challenging requirements may be “The SDLCM shall be simple and easy to implement.” Considering the multi-aspect nature of LSDSs development, development of a simple and easy to implement models will not be easy. However, it is important to note that not all requirements may be implemented in a systems development life cycle model. These requirements should be seen as a direction for the optimal design. The models able to support most of these requirements will be more successful in satisfying the challenging needs in LSDSs.

Table 2. Requirements for a systems development life cycle model for large-scale defense systems developments

Requirement No.	Requirement	Waterfall Model Support
Requirement 1	The SDLCM shall be scalable.	Partially Supported
Requirement 2	The SDLCM shall support the development of system of systems.	Not Supported
Requirement 3	The SDLCM shall be compatible with government acquisition policies.	Supported
Requirement 4	The SDLCM shall support the development of complex systems.	Supported
Requirement 5	The SDLCM shall be software oriented.	Partially Supported
Requirement 6	The SDLCM shall be quality oriented.	Partially Supported
Requirement 7	The SDLCM shall have a verification and validation perspective.	Supported
Requirement 8	The SDLCM shall include safety and security perspective.	Not Supported
Requirement 9	The SDLCM shall support total continuous risk management.	Not Supported
Requirement 10	The SDLCM shall support concurrent engineering.	Not Supported
Requirement 11	The SDLCM shall support project management areas.	Partially Supported
Requirement 12	The SDLCM shall support defense enterprise architectures.	Not Supported
Requirement 13	The SDLCM shall support stakeholder involvement.	Partially Supported
Requirement 14	The SDLCM shall support architecture oriented development.	Not Supported
Requirement 15	The SDLCM shall emphasize supportability through open architectures.	Not Supported
Requirement 16	The SDLCM shall emphasize trustworthiness, maintainability, and evolvability.	Not Supported
Requirement 17	The SDLCM shall support automated documentation.	Not Supported
Requirement 18	The SDLCM shall support practices for high productivity.	Not Supported
Requirement 19	The SDLCM shall support test optimization and ease of testing.	Not Supported
Requirement 20	The SDLCM shall support tailoring/customization based on needs.	Not Supported
Requirement 21	The SDLCM shall be able to handle requirements change.	Not Supported
Requirement 22	The SDLCM shall support evolutionary development.	Not Supported
Requirement 23	The SDLCM shall be simple in nature and easy to implement.	Supported

To examine the applicability of the requirements, one of the most commonly known models is applied. The first formal description of the Waterfall model is described by Royce in 1970 [2]. In 1985, the US DoD adapted this model in a military standard (DOD-STD-2167A) for software development. Therefore, it has found use in defense projects. Its sequential approach is compatible with the milestones in the defense acquisition framework. This model is a document intensive model, therefore, it also aligns with the defense acquisition with heavy

documentation. The Waterfall model follows a sequential process of a series of design activities. These activities are requirements identification, system design, system implementation, and verification in its simplest form. The model is presented in **Figure 1**. The readers are referred to an abundant literature on the strengths, weaknesses, and applicability of this model. In **Table 2**, the last column indicates whether the Waterfall model supports the requirement or not.

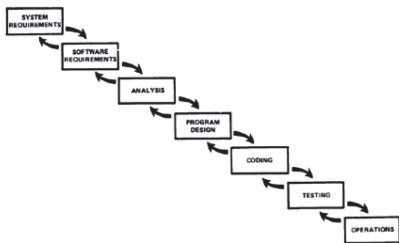


Fig. no. 1. The waterfall model of systems development life cycle [Taken from [2]]

5. CONCLUSIONS AND FUTURE WORK

Today, the project performances observed in the development of LSDSs cry out for immediate and effective solutions to a magnitude of problems encountered during development. It is obvious that we need better systems development life-cycle models that can address the specific challenges of LSDS developments. While having better systems development life-cycle models may not solve all the problems such as government acquisition problems, the ever-increasing defense systems scale and complexity, it may solve some of the problems and lessens the adverse effects of some other problems. Noticing this clear need, we conducted research on the first step of development of a systems development life cycle models for LSDSs. The first step is the identification of a set of requirements for systems development life cycle models for LSDS developments. As a result, this study is one of the first steps in a research agenda. The goal of this research is the development of a system development model for large-scale defense systems.

The research agenda consists of the following steps:

1. Identification of requirements for a LSDS development model.
2. Identification and categorization of current LSDS characteristics.
3. Identification and categorization of current LSDS project development characteristics.
4. Identification and categorization of LSDS development challenges of today.
5. Investigation of best practices in LSDS developments

6. Identification of processes consisting of best practices that can effectively address and overcome the challenges.

7. Coherent formation of processes to be used in the LSDS development model.

8. Development of the LSDS development model capable of addressing today's and near-future's needs.

9. Conducting pilot studies and industrial experiments.

Note the difference in the model development strategy between ICM and the development steps proposed here. While ICM mainly builds upon the strengths of previous models and best practices, the strategy employed in this research agenda starts with the identification of the characteristics, needs and challenges of current LSDS developments. The strategy employed in the development of ICM is valid and effective. However, the strategy in this research agenda is ideal, which is starting with the requirements specific to LSDS life cycle development models.

While this list of requirements is comprehensive, it may not be complete. Note that the determination of completeness in this area is not easy. Therefore, this list should be considered a starting point in this research area.

ACKNOWLEDGMENTS AND DISCLAIMERS

The authors take full responsibility for the contents and scientific correctness of the paper. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any affiliated organization or government.

REFERENCES

- [1] Stockholm International Peace Research Institute. Military Expenditure Database, 2015.
- [2] Royce, W. W. (1970) Managing the development of large software systems: Concepts and techniques. In Proceedings of IEEE WESTCON. (Los Angeles, CA), 1-9.
- [3] Nielsen, P. D. (2015) Software Engineering and the Persistent Pursuit of Software Quality. *The Journal of Defense Software Engineering*. May-June, 4-9.

- [4] USAF (1992) "Bold Strike" Executive Software Course.
- [5] Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9), 833-859.
- [6] Jones, C. (2000). *Software assessments, benchmarks, and best practices*. Addison-Wesley Longman Publishing Co., Inc., 2000.
- [7] U.S. Government Accountability Office (2015). *Defense Acquisitions - Assessments of Selected Weapon Programs*, Rep. No: GAO-15-342SP. March 2015.
- [8] Boehm, B., Lane, J. A., Koolmanojwong, S., & Turner, R. (2014). *The incremental commitment spiral model: Principles and practices for successful systems and software*. Addison-Wesley Professional.
- [9] Boehm, B. & Lane, J. A. (2007) Using the incremental commitment model to integrate system acquisition, systems engineering, and software engineering. *The Journal of Defense Software Engineering*, 2007, 19(10), 4-9.
- [10] Parnas, D. L. (1985) Software aspects of strategic defense systems. *Communications of the ACM*, 28(12), 1326-1335.
- [11] Northrop, L. (2013) Does scale really matter? Ultra-large-scale systems seven years after the study. In Proc. of 2013 35th Int. Conf. on Software Eng. (ICSE), 18-26 May 2013, San Francisco, CA, USA, pp. 857-857.
- [12] Hagen, C.; Sorenson, J.; Hurt, S. & Wall, D. (2012) Software: The Brains behind U.S. Defense Systems, A.T. Kearney Inc.
- [13] Hagen, C. & Sorensen, J. (2013) Delivering Military Systems Affordably, Defense AT&L, March-April.
- [14] Nelson, M., Clark, J. & Spurlock, M. A. (1999) Curing the software requirements and cost estimating blues, *PM Magazine*, November-December, pp. 54-60.
- [15] Brooks, F.P. (1995). *The mythical man-month*.
- [16] Shalal-Esa, A. (2012) Pentagon focused on resolving F-35 software issues. Online News from Reuters, 30 March 2012.
- [17] Demir, K. A. (2005). Analysis of TLcharts for weapon systems software development, Master's Thesis, Naval Postgraduate School, Monterey, CA, USA. December 2005.
- [18] Demir, K. A. (2009). Challenges of weapon systems software development. *Journal of Naval Science and Engineering*, 5(3).
- [19] Drusinsky, D., Shing, M. T., & Demir, K. (2005). Test-time, Run-time, and Simulation-time Temporal Assertions in RSP. In Rapid System Prototyping, 2005. (RSP 2005). The 16th IEEE International Workshop on (pp. 105-110).
- [20] Defense Acquisition Guidebook (2013).
- [21] Alberts, D. S.; Garstka, J. J. & Stein, F. P. (2000) *Network Centric Warfare: Developing and Leveraging Information Superiority*. (C3I/Command Control Research Program) Washington DC. 2000.
- [22] Demir, K. A. (2006). Meeting Nonfunctional Requirements through Software Architecture: A Weapon System Example. In Proceedings of the First Turkish Software Architecture Design Conference, TSAD 2006, Istanbul, Turkey, pp. 148-157, 20-21 November 2006.
- [23] Cebrowski, A. K., & Garstka, J. J. (1998). Network-centric warfare: Its origin and future. In US Naval Institute Proceedings 124(1). pp. 28-35.
- [24] Goldin, L.; Matalon-Beck, M. & Lapid-Maoz, J. (2010) Reuse of requirements reduces time to market. In Proc. of 2010 IEEE International Conference on Software Science, Technology and Engineering, pp. 55-60. Herzlia, Israel.
- [25] Garrett, R. K.; Anderson, S.; Baron, N. T. & Moreland, J. D. (2011) Managing the interstitials, a system of systems framework suited for the ballistic missile defense system. *Systems Engineering*, 14(1), 87-109.
- [26] Demir, K. A. (2015). Multi-View Software Architecture Design: Case Study of a Mission-Critical Defense System. *Computer and Information Science*, 8(4).
- [27] Humphrey, W. S. (2005) Why big software projects fail: The 12 key questions. *The Journal of Defense Software Engineering*, March. 25-29.
- [28] U.S. Government Accountability Office, High Risk Series, Report No: GAO-15-290, Feb. 2015.
- [29] U.S. Government Accountability Office, High Risk Series, Report No: GAO-09-271, Jan. 2009.
- [30] U.S. Government Accountability Office, Major Management Challenges and Program Risks - DoD, GAO/OCG-99-4, Jan. 1999.
- [31] U.S. Government Accountability Office, Acquisition Reform - DOD Should Streamline Its Decision-Making Process for Weapon Systems to Reduce Inefficiencies, Feb. 2015, Report No: GAO-15-192.
- [32] Jones, C. (2002) Defense Software Development in Evolution, *The Journal of Defense Software Engineering*, November. 26-29.
- [33] Blanchard, B. J. & Fabrycky, W. J. (1998) *Systems Engineering and Analysis*, 3rd Edition, Prentice Hall International Series in Industrial & Systems Engineering. ISBN: 0131350471.
- [34] Demir, K. A. (2008). Measurement of software project management effectiveness. Doctoral Dissertation, Naval Postgraduate School, Monterey, CA, USA. December 2008.
- [35] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guide), Fifth Edition, 2013.
- [36] Demir, K. A. (2009). A Survey on Challenges of Software Project Management. In *Software Engineering Research and Practice* (pp. 579-585).