A Study of COTS Integration Projects: Product Characteristics, Organization, and Life Cycle Models

Katerina Megas
Computer Science, Virginia Tech.
7054 Haycock Road
Falls Church, VA 22043 USA
kmegas@vt.edu

William B. Frakes
Computer Science, Virginia Tech.
7054 Haycock Road
Falls Church, VA 22043 USA
Ph: +1-703.538.8740
wfrakes@vt.edu

Julián Urbano
University Carlos III of Madrid
Avda. Universidad, 30
28911 Leganés, Spain
jurbano@inf.uc3m.es

Gabriella Belli
Educational Research, Virginia Tech.
7054 Haycock Road
Falls Church, VA 22043 USA
gbelli@vt.edu

Reghu Anguswamy*
Computer Science, Virginia Tech.
7054 Haycock Road
Falls Church, VA 22043 USA
reghu@vt.edu

ABSTRACT

We present a descriptive and exploratory study of factors that can affect the success of COTS-based systems. Based on a review of the literature and industrial experience, the choice of life cycle model and the amount of glueware required were hypothesized as the main factors in predicting project success. In this study we examined the relationship between different life cycle models and COTS integration project success. Two life cycle models were studied: the sequential model and the iterative model. Seven subjects from six industrial organizations responded to a survey providing data on 23 COTS integration projects. While there was variability between iterative and sequential projects on a variety of organizational and product factors, little difference was found between the life cycle models on the success criteria of projects (i.e. being on time, meeting requirements and being within budget). We found that projects that met two or three of the success criteria had significantly higher scores on project characteristics (organizational plus product) than those meeting none or just one.

Categories and Subject Descriptors

[**Software and its engineering**]: Software creation and management – *COTS integration.*

General Terms

Management, Measurement, Documentation, Performance, Economics, Experimentation, Human Factors.

Keywords

COTS Integration, Life Cycle Model, Organizational Characteristics, COTS Product Characteristics.

1. INTRODUCTION

This paper focuses on the important reuse area of commercial off-the-shelf (COTS) components and their integration process. Benefits of using COTS products include shorter development times and reduced integration effort and costs. As a result, their demand by customers has increased (an observed dramatic growth in the number of COTS-based system (CBS) projects from 28% in 1997 to 60% in 2001 [1]). According to a report by the Standish Group [2], overall software project success rates rose from 16% in 1994 to 28% in 2001, but by 2011 they still were at only 34%. While these data show improvement, more improvement is needed. We surveyed the COTS literature to identify predictors of project success, defined as: "…to bring a project to completion on time, within the budget cost, and to meet the planned performance or end product goals" [3]. These predictors are summarized in Figure 1 as our initial model of factors affecting the success of COTS integration projects. We then collected data from 23 such projects and used these data to begin exploring relationships between two life cycle models (LCMs) and the three success variables. Challenges and risks for COTS based development have been studied in the past [4-8]. For this study, we gathered empirical data on life cyc1e models, project characteristics and success of several COTS integration projects. Our hope was to identify a set of factors that would predict project success, including choice of life cycle model. If a particular LCM could be shown to improve the likelihood of project success, this would change the current practice of using a mix of life cycle models.

Motivation: Tools such as COCOMO have been developed to take into account LCM. A primary premise, quoted by Barry Boehm et.al. for revision of COCOMO in COCOMO 2.0 [9], is the recognition that there was a need to be able to tailor the cost estimation tool, depending on which LCM the organization planned to implement. Therefore, one must choose the LCM that will be the most cost effective while insuring project success.

If there are measurable factors of a project that can be used to predict the likelihood of meeting project schedule, budget and requirements, the risk budget built into the cost proposal and the amount of 'total slack' built into the proposed schedule could be better predicted. The main purpose of this paper is to begin the search for an empirically-driven set of conditions that project and technical managers can evaluate during the early phases of a COTS integration project. One example might be during the procurement phase, which would characterize the project, and support the decision of which LCM will be successful and most cost-effective.

2. COTS INTEGRATION

In this section we review factors affecting COTS integration projects: software concepts, life cycle concepts, and project characteristics.

2.1 COTS Software Concepts

For a typical COTS product, the source code is not available to the application developers, and its future evolution is not under their control. As a result, and depending on the potential gap between requirements and COTS out-of-the-box functionality, glue code or glueware is required to compensate for missing COTS functionality [10]. This glueware is also responsible for joining together the COTS products integrated in the project, as they are not designed to interoperate with each other sometimes because of lack of compliance with standards [5]. The effort and time associated with developing this glueware has been the focus of many COTS cost estimation tools such as COCOTS. In fact, data as of 2000 suggest that the majority of the costs associated with COTS integration are the result of this glue code development [11]. Paradoxically, it is still the COTS products the ones that execute the most lines of code in these COTS-based systems, being the glue code hardly executed [12].

Gathering requirements for COTS-based systems differs from the traditional way, where requirements are defined at the beginning of the project and are the basis of the system specification and expected capabilities. In contrast, for COTS integration, it is the capabilities of the COTS products used that influence the requirements [13]. This activity is what ultimately drives the size and effort of a COTS integration project.

There is also a secondary source of effort for COTS integration projects: mismatched requirements. These are requirements identified after the project has begun, which may be defined as the expected volatility of requirements. In the COCOTS estimation model, BRAK is identified as the percentage of COTS glueware thrown away due to requirements volatility [14]. Therefore, when analyzing project success criteria (i.e. budget, schedule, and intended requirements) it appears that mismatched requirements impact all three of these critical success aspects [15]. Multiple product implementations should also lead to a richer requirements base in the product. As such, “product characteristics” are evaluated, as one kind of project attributes that may affect which life cycle model may be more likely to produce project success.

Another source of mismatched requirements is the `I’ll know it when I see it` factor. Often, sponsors are not familiar with the technology used on the project or the technical specifications that are required by the organization. As an organization matures, they are more likely to recognize the requirements earlier in the process and provide them. Thus “organizational characteristics” is the second of the project conditions that can determine which life cycle may be more likely to produce project success.

2.2 Life Cycle Model Concepts

COTS integration projects use both sequential and iterative life cycle models [16]. Sequential models, such as the waterfall model, clearly define stages in the life cycle and identify glueware requirements prior to development and integration activities. Iterative models such as the spiral model [17] involve revisiting each life cycle phase with continuing feedback.

In sequential models, defined activities are conducted in order, and controls are in place to do verification before moving to the next phase. A fundamental assumption of the waterfall model is that the requirements are knowable before implementation. The obvious drawback to this approach is the dependency on non-volatile requirements. COTS products are a response to an established set of market requirements. Therefore we would assume that, for a mature COTS product, requirements would be non-volatile and fit the waterfall model in the overall integration project. Also, as noted by Balk and Kedia in a COTS Integration case study [18], this model, if appropriate, is generally considered the most efficient way of building software and therefore, from an integrator’s perspective, should be chosen if capable of delivering project success. However, in COTS-based systems, it is the COTS products themselves who often dictate the requirements, so no assumptions can be made about the system’s future capabilities [19]. This indicates that the use of traditional sequential methods should not be adopted.

The argument for an iterative life cycle is the risk aversion benefit it offers. A key concept is to leverage prototyping and iterative construction phases to allow repeated verification of requirements. As the probability increases that a project will have volatile or mismatched requirements, the need for a life cycle to mitigate this risk becomes imperative. Although many variations of the iterative model exist, for exemplary purposes we will consider the spiral model. The spiral model, as described by Barry Boehm in this Win-Win concept, has various anchor points throughout the life cycle [20]. These are the Life Cycle Objects, Life Cycle Architecture and Initial Operational Capability. Each iteration is performed as many times as required to meet the objectives of the anchor points.

2.3 Project Characteristics

Barry Boehm has stated that the decision of how to tailor project’s processes depends on the degree of understanding of the requirements and architecture [12]. This is the essence of the problem. There is thus a need to identify characteristics that can be used to assess the degree of understanding of requirements and architecture. This paper focuses on two such aspects of a project: organization and product. A summary of factors that may affect the success of a COTS integration project is given in Figure 1. Measures of success are whether the project is finished on time, within budget, and meets the intended requirements.

Based on a review of the literature, the primary predictors of project success are the amount of glueware required to make up for the requirements that the COTS products cannot satisfy, and the life cycle model chosen. The amount of glueware required depends on mismatched requirements, a function of customer knowledge of the technology involved. An understanding of system requirements and architecture early in the development process heavily influences the choice of life cycle model. This understanding is influenced by many product and organizational factors.

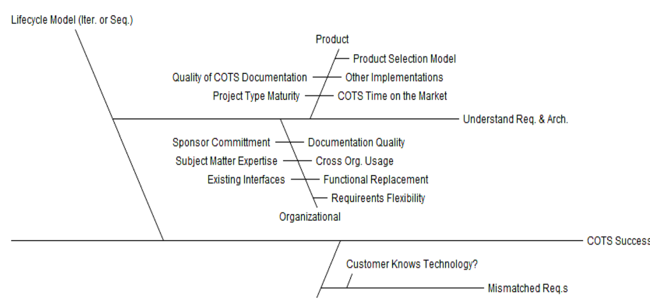

Figure 1. COTS Integration Project Success Factors

3. METHODS

Ideally, one would like to compare success measures for similar projects where sequential and iterative life cycle models were integrated and to determine which organizational and product characteristics were most closely related to success in each type of LCM. However, gaining entry into various organizations to conduct such an experimental study is fairly difficult and existing data, if any, are hard to find. In an attempt to begin studying these issues based on the opinions of experienced system integrators, a questionnaire was constructed to gather information from managers of projects where COTS products were used. Each respondent was asked to provide information about one to five projects.

3.1 Questionnaire

The questionnaire was divided into four sections asking about the following aspects for each of the projects being described: (1) project specifics, (2) organizational characteristics, (3) product characteristics, and (4) project success.

The following information was requested about project specifics: the scheduled length of the project duration in months, approximate effort of the project in man months, the business sector of the project, and the lifecycle applied to the project (two choices were provided: sequential model or iterative model).

The second section of the questionnaire dealt with the following organization's characteristics: functional replacement (how much functionality was being replaced by the COTS product and whether it was introduced in a new or existing process), cross functional requirements (the number of organizations crossed by the stakeholders group), stakeholder subject matter expertise, sponsor commitment, existing interfaces (the complexity of the environment into which the COTS product would integrate), requirements flexibility, and quality of existing environment documentation. The third section of the questionnaire dealt with product characteristics for each of the projects: time on market, market maturity, number of other implementations that existed for this product, quality of COTS documentation, and product selection method. The final section dealt with overall project results, and consisted of five questions with yes/no responses for each project described: 1. Did the project complete on time? 2. Did the project complete on budget? 3. Did the project meet the requirements? 4. Would a different lifecycle model have been preferred? 5. If an iterative LCM was used, would you expect the project to have met the cost and budget objectives had the project implemented a sequential LCM?

3.2 Respondent and Project Demographics

Seven technical and project managers working as system integrators completed the questionnaire, each providing information about project conditions and outcomes for one to five projects. Their total years of experience ranged from 10 to 35 years, with a mean of 19.1 and a standard deviation (sd) of 11.2 years. Their experience in COTS integration ranged from 1 to 15 years, with a mean of 8.1 and sd of 4.4 years.

The size of the 23 projects ranged from 2 to 1,200 man months of project effort (mean = 196, sd = 341), while their overall duration ranged from 2 to 24 months (mean = 11, sd = 7). The two most represented business sectors were security (7 projects) and financial (5 projects). One project came from each of the following sectors: consulting, consumer product, commercial aviation, defense, government, and healthcare. The remaining five projects fell into an "other" category.

Three respondents provided data on both sequential and iterative projects, for a total of 11 projects. All responses were compared for these projects, examining for potential patterns across respondents. Because none were evident, the remainder of this paper contains a discussion of all 23 projects as independent data sources.

4. RESULTS

Presented first are details about the LCMs investigated and project characteristics. We conclude with success criteria and their relationship to other variables, as defined by the objectives.

4.1 Life Cycle Model

The sequential model was the one most commonly applied, with the iterative model present in less than one quarter of the 23 projects. In follow up discussions, respondents indicated difficulty in applying an iterative life cycle model. Pricing and estimating the cost of an iterative life cycle in advance is often a challenge, requiring estimates of number of iterations needed. Additionally, once the project is underway, the management of scope and requirements is difficult. The scope of a project is typically defined early in the project and tied to the cost estimate. As the requirements evolve during iterations, project and technical managers experience challenges in managing to the original scope that was specified and estimated at project initiation. Augmenting a sequential model with techniques that mitigate the risk of missed requirements is therefore preferred.

4.2 Project Characteristics

4.2.1 Organizational Characteristics

Table 1 gives results for organizational characteristics for sequential and iterative projects, and for the total data set.

Functional Replacement: The COTS integration objective for 61% of the projects was to replace extensive functionality in a new process. This was the most complex of four response options involving limited or extensive functionality being replaced in a new or existing process. Identifying all requirements up front will be least likely when one must extensively replace existing functionality and use a new process. Under these circumstances, it will be more difficult to gather complete requirements than for the opposite case, with limited replacement of functionality and an existing process. Although most projects in this category applied a sequential life cycle, the percentage of each LCM was similar across all the categories of functional replacement.

Cross Functional Requirements: The largest percentage of projects (44%) involved solutions for the needs of multiple organizations with differing missions. As the industry trend is moving away from stovepipe solutions towards service oriented architectures, it can be expected that the number of implementations that will be deployed across organizations will increase. The complexity is further enhanced, however, when these organizations have differing missions. In such cases, it is sometimes discovered during the requirements gathering phase that requirements are conflicting as well as different for different organizations. None of the implementations that involved only a single organization used the iterative model, which is to be expected, given the difficulties of implementing an iterative LCM.

Stakeholder Subject Matter Expertise: As might be expected, four out of five iterative projects, but only one-third of the sequential projects, had no stakeholder expertise. One would expect that with little or no stakeholder expertise, it would be more difficult to identify and capture requirements.

Sponsor Commitment: When a project's sponsor or champion is at a higher level in an organization, one expects the focus and assignment of resources to the project to also be at a higher level, thereby facilitating the requirements gathering process. All of the sequential projects had a champion (two-thirds at the executive level and one-third at the managerial level). Two of the iterative projects had no sponsor.

Table 1. Organizational Characteristics for 23 COTS Projects

	Sequential N = 18	Iterative N = 5	Total N = 23
Functional replacement			
extensive functionality, new process	11 (61%)	3(60%)	14(61%)
extensive functionality, existing process	3(17%)	1(20%)	4(17%)
limited functionality, new process	3(17%)	1(20%)	4(17%)
limited functionality, existing process	1(6%)	0(0%)	1(4%)
Cross functional requirements			
multiple organizations; different missions	7(39%)	3(60%)	10(44%)
multiple organizations with same mission	4(22%)	2(40%)	6(26%)
single organization	7(39%)	0(0%)	7(30%)
Stakeholder subject matter expertise			
no expertise	6(33%)	4(80%)	10(44%)
some expertise	7(39%)	0(0%)	7(30%)
significant expertise	5(28%)	1(20%)	6(26%)
Sponsor commitment			
no champion	0(0%)	2(40%)	2(9%)
champion at management level	6(33%)	1(20%)	7(30%)
champion at executive level	12(67%)	2(40%)	14(61%)
Existing Interfaces			
11 or more	1(6%)	2(40%)	3(14%)
between 6 and 10	5(29%)	1(20%)	6(27%)
5 or fewer	11 (65%)	2(40%)	13(59%)
Requirements flexibility			
no flexibility	6(33%)	0(0%)	6(26%)
some flexibility	11 (61%)	4(80%)	15(65%)
product "as is"	1(6%)	1(20%)	2(9%)
Quality of existing documentation			
no documentation	7(39%)	3(60%)	10(44%)
some documentation	10(56%)	2(40%)	12(52%)
documented in detail	1(6%)	0(0%)	1(4%)

Existing Interfaces: As the number of interfaces increases, one might guess that the number of requirements will increase and therefore also the difficulty in assessing and identifying all requirements early in the project life cycle. An iterative process may facilitate discovery of the various interface requirements in a progressive manner. Although the range for number of interfaces was almost identical for both types of projects, iterative projects had, on average, a higher mean number of interfaces than sequential projects (9 versus 5).

Requirements Flexibility: With a sequential life cycle and flexibility in an organization, negotiations may be possible if missed requirements are discovered at a later stage, thereby keeping the project on track. The most likely scenario for success is taking the product "as is" because there is no potential for missed requirements. Only one iterative and one sequential project fell into this category. As the ability to negotiate decreases, and there is no flexibility due to legal or contractual implications, one would expect the iterative life cycle to be more popular because it allows for revisiting and refining requirements during the life cycle, thereby ensuring that all must-have requirements are captured and met. However, no iterative project fell into this category, while one-third of the sequential projects did. Most of the sequential projects (61%) and four out of five iterative projects fell into the middle category, where all requirements are desired, but a level of understanding exists that some requirements may need to be modified.

Quality of Existing Environments Documentation: The lack of documentation for the existing environment will affect the integrators' ability to identify requirements up front in the life cycle, while complete documentation should facilitate success. Only one sequential project had complete documentation, while over one-third (39%) had no documentation, and over one-half (56%) had only some documentation. Two of the iterative projects had some documentation and three had none.

4.2.2 Product Characteristics

Here we list the product characteristics and the justification for those being drivers of the choice of life cycle model. For this analysis, we again examined the 23 projects, each of which incorporated a single COTS product. Table 2 provides the results for product characteristics. As before, higher scores imply greater stability, which should be related to a greater likelihood for success, and lower scores imply greater complexity.

Table 2. Product Characteristics for 23 COTS Projects

	Sequential N = 18	Iterative N = 5	Total N = 23
Time on market (yrs)			
5 or fewer	13(72%)	2(40%)	15(65%)
between 6 and 10	4(22%)	2(40%)	6(26%)
11 or more	1(6%)	1(20%)	2(9%)
Market maturity (yrs)			
10 or fewer	12(67%)	2(40%)	14(61%)
between 11 and 20	4(22%)	2(40%)	6(26%)
21 or more	2(11%)	1(20%)	3(13%)
No of implementations			
25 or fewer	7(50%)	3(60%)	10(53%)
between 26 and 100	3(21%)	2(40%)	5(26%)
101 or more	4(29%)	0(0%)	4(21%)
Quality of COTS documentation			
no existing documentation	0(0%)	2(40%)	2(9%)
some documentation	12(67%)	3(60%)	15(65%)
a lot of documentation	6(33%)	0(0%)	6(26%)
Product selection			
based on market materials	7(39%)	2(40%)	9(39%)
based on results of lab prototype evaluation	6(33%)	2(40%)	8(35%)
based on results of a field prototype	5(28%)	1(20%)	6(26%)

COTS Time on the Market: One would expect that an iterative model would be used more frequently with immature COTS products due to its risk avoidance features. Iterative projects incorporating COTS products did have a lower maximum time on market (12 years) than the sequential projects incorporating COTS components (17 years). However, iterative projects had longer mean and median time on the market than the sequential projects.

Market Maturity: The average maturity of the COTS product marketplace was a little over 12 years with a standard deviation of 7 years, indicating a wide variance of market maturity. The range for the number of years of existence for the COTS product space was 3 to 25 years for the sequential and 5 to 25 years for the iterative projects.

Number of Implementations: Although the average number of implementations that existed for the COTS product was almost the same for both types of projects (around 12), the median for sequential projects (122) was almost five times that of the iterative projects (26). As the number of implementations increases, it is expected that a better guess of requirements can be made earlier in the life cycle, with a sequential model being given preference, as seems to be the case here.

Quality of COTS Documentation: The more documentation that exists, the more can be shared with stakeholder organizations and facilitate requirements gathering. This should better support a sequential life cycle that is heavily documentation-oriented. While only one-third of the sequential projects had a lot of documentation, the remaining projects had at least some. In contrast, none of the iterative projects had a lot of documentation, three had some documentation and two had none at all.

Product Selection Method: Selection methods based on experience with the product beyond a survey of marketing materials should provide a better position to identify requirements. Over 60% of the products were selected based on results of either lab prototype evaluation or a field prototype. However, both project types used all three product selection methods, including those based on market materials, almost equally.

4.3 Project Success

Respondents were asked if each of the 23 projects was completed on time, completed on budget, and whether it met the intended requirements. Possible answers were “yes” or “no”. Based on these three success criteria, it was found that: 13 (57%) of the projects were completed on time, 13 were completed on budget (with 10 successful on both counts), and 18 (78%) met requirements. Nine projects (39%) met all three of the success criteria and only four projects (17%) met none of them. When asked if a different life cycle model would have been preferred, the answer was “yes” for all projects meeting none of the success criteria and “no” for the projects meeting at least one.

4.3.1 Life Cycle Model and Success Rate

Figure 2 shows that there was little difference between median scores on the three success criteria for the two LCMs. A composite score was created by counting how many of the three criteria each project met. On average, the 18 sequential projects met 1.9 criteria and the five iterative projects met 2.0 criteria, a trivial difference. We then considered a project “successful” if it met at least two of the success criteria and a “failure” if it met none or only one of the criteria. Using this definition, 12 of the 18 sequential projects (67%) and four of the five iterative projects would be considered successful.

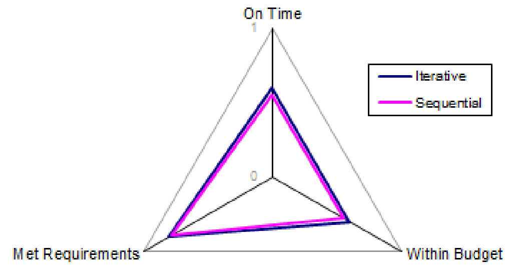

Figure 2. Median Success Proportions by LCM

4.3.2 Project Characteristics and Success Rate

Figure 3 summarizes the relationship between success and project characteristics, which include both product and organizational variables. The success group is defined as those projects that met two or three of the success criteria (on-time, on-budget, and meets requirements). Projects in the failure group met zero or just one of the success criteria. As can be seen, success was associated with three organizational characteristics (simpler cross functional requirements, fewer existing interfaces, and greater detail in existing documentation) and with three product characteristics (greater market maturity, more implementations, and greater likelihood of being based on a realistic prototype). Note also that none of these project characteristics showed a higher value in failed projects than in succeeded projects.

It is expected that the fewer interfaces a COTS product has to interact with, the more likely the project is to succeed. That interaction is usually performed with the aid of glue ware, which is the only piece of software developed from scratch. Clearly, having quality documentation helps in this matter. The market maturity and the number of other implementations of the COTS products seem to be related to project success. Indeed, one would expect a COTS product to be more reliable if it belongs to a established market, besides expecting it to fulfill more requirements that may be needed in the integration project. Developers use to avoid formal selection procedures, and make their decisions based on the familiarity with the product [5].

Figure 4 shows with boxplots that projects that met two or three of the success criteria had statistically significant higher scores on project characteristics (organizational plus product) than those meeting none or just one.

5. CONCLUSIONS

This paper presented a descriptive and exploratory study of factors that can affect the success of COTS integration projects. We examined the effect of two life cycle models (iterative and sequential) on COTS integration project success. Seven subjects from six industrial organizations responded to a survey providing data on 23 COTS integration projects. While there was variability between iterative and sequential projects on a variety of organizational and product factors, little difference was found between the life cycle models on the success criteria of projects being on time, meeting requirements, and being within budget. It has been shown that both iterative and sequential models have advantages for COTS integration projects, but they also have drawbacks, as the data gathered here shows. This indicates that traditional software development may not be suitable for COTS integration projects, as has been noted before [19].

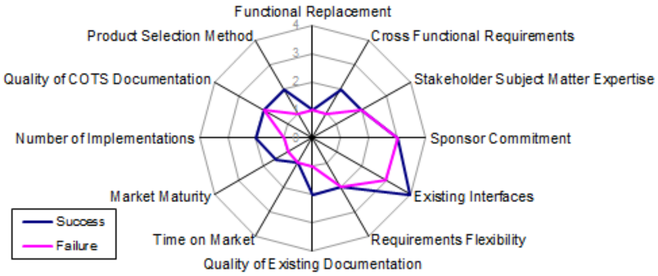

Figure 3. Differential Success by Project Characteristics

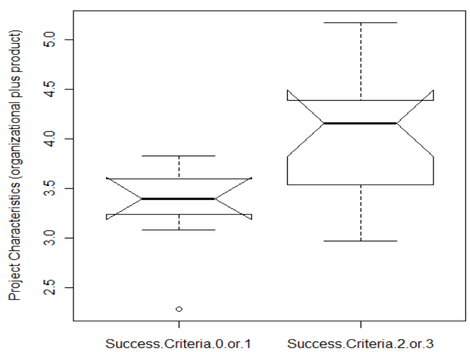

Figure 4. Project Characteristics by Project Success

We found that 39% of the projects met all of the success criteria, 30% met two of the three, 13% met only one, and 17% met none of the success criteria. Success was associated with three organizational characteristics: simpler cross functional requirements, fewer existing interfaces, and greater detail in existing documentation. Success was also associated with three product characteristics: greater market maturity, more implementations, and greater likelihood of being based on a realistic prototype. Projects that met two or three of the success criteria had significantly higher scores on project characteristics (organizational plus product) than those meeting none or just one.

It has been shown that COTS integration, while introducing many advantages, also has many problems related to safety, reliability, and security, among others. These problems need to be addressed. Another issue not addressed has to do with maintainability of COTS-based systems. As there is no control over the evolution of the COTS products used, maintainability represents a great challenge because upgrades are frequently not compatible, products become obsolete and they end up unsupported by vendors, which can easily make the system unusable.

It has been shown that this post-development cost can easily exceed expectations, so empirical studies in this matter are needed. Supporting previous research, we identified some key factors that seem to affect the success of COTS integration projects. They are related mainly to COTS product selection and architecture. On the other hand, some other factors supposed to affect these kinds of projects do not seem to do so, according to our data. Although this could be because of the small size of our data sample, more empirical research is needed. Also, the small sample size is a threat to the external validity.

REFERENCES

[1] Boehm, B., et al., *Not all CBS are created equally: COTS-intensive project types.* COTS-Based Software Systems, 2003: p. 36-50.

[2] Group, T.S. *CHAOS Report - Standish Group Website*. 2012 [cited 2012 September 12]; Available from: http://www.standishgroup.com.

[3] Dvir, D., et al., *An empirical analysis of the relationship between project planning and project success.* International Journal of Project Management, 2003. **21**(2): p. 89-95.

[4] Li, J., et al., *An empirical study of variations in COTS-based software development processes in the Norwegian IT industry.* Empirical Software Engineering, 2006. **11**(3): p. 433-461.

[5] Torchiano, M. and Morisio, M., *Overlooked aspects of COTS-based development.* Software, IEEE, 2004. **21**(2): p. 88-93.

[6] Chen, W., et al., *An empirical study on software development with open source components in the chinese software industry.* Software Process: Improvement and Practice, 2008. **13**(1): p. 89-100.

[7] Li, J., et al., *Development with off-the-shelf components: 10 facts.* Software, IEEE, 2009. **26**(2): p. 80-87.

[8] Cechich, A. and Piattini, M., *Early detection of COTS component functional suitability.* Information and Software Technology, 2007. **49**(2): p. 108-121.

[9] Boehm, B., et al., *Cost models for future software life cycle processes: COCOMO 2.0.* Annals of Software Engineering, 1995. **1**(1): p. 57-94.

[10] Davis, L., et al. *Toward formalizing service integration glue code*. 2005: IEEE.

[11] Abts, C., et al. *COCOTS: A COTS software integration lifecycle cost model-model overview and preliminary data collection findings*. in *ESCOM 2000*. 2000. Munich, Germany: Citeseer.

[12] Basili, V.R. and Boehm, B., *COTS-based systems top 10 list.* Computer, 2001. **34**(5): p. 91-95.

[13] Stavridou, V. *COTS, integration and critical systems*. 1997: IET.

[14] Abts, C.M. and Boehm, B.W. *COTS Software Integration Cost Modeling Study*. University of Southern California Technical Report 1997 [cited 30602 94-C]; 1095].

[15] Voas, J.M., *The Challenges of Using COTS Software in Component-Based Development.* Computer, 1998. **31**(6): p. 44-45.

[16] Morisio, M., et al., *Investigating and improving a COTS-based software development*, in *Proceedings of the 22nd international conference on Software engineering*. 2000, ACM: Limerick, Ireland. p. 32-41.

[17] Boehm, B., *Escaping the software tar pit: model clashes and how to avoid them.* ACM SIGSOFT Software Engineering Notes, 1999. **24**(1): p. 36-48.

[18] Balk, L.D. and Kedia, A. *PPT: a COTS integration case study*. in *Proceedings of the International Conference on Software Engineering, ICSE 2000*. 2000.

[19] Boehm, B. and Abts, C., *COTS integration: Plug and Pray?* Computer, 1999. **32**(1): p. 135-138.

[20] Boehm, B., et al., *Using the WinWin spiral model: a case study.* Computer, 1998. **31**(7): p. 33-44.