ELSEVIER

Information and Software Technology 44 (2002) 53–62

INFORMATION AND SOFTWARE TECHNOLOGY

www.elsevier.com/locate/infsof

Case study: factors for early prediction of software development success

J. Drew Procaccino[a,*], June M. Verner[b], Scott P. Overmyer[c], Marvin E. Darter[a]

[a]College of Business Administration, Rider University, 2183 Lawrenceville Road, Lawrenceville, NJ 08648, USA
[b]College of Information Science and Technology, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA
[c]Massey University, Private Bag 102 904, North Shore Mail Centre, Auckland, New Zealand

Received 25 June 2001; revised 31 August 2001; accepted 5 November 2001

Abstract

Project managers can make more effective and efficient project adjustments if they detect project high-risk elements early. We analyzed 42 software development projects in order to investigate some early risk factors and their effect on software project success. Developers in our organization found the most important factors for project success to be: (1) the presence of a committed sponsor and (2) the level of confidence that the customers and users have in the project manager and development team. However, several other software project factors, which are generally recognized as important, were not considered important by our respondents. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Software development success; Management support; Customer/users involvement; Requirements management; Risk management

1. Introduction

Despite extensive research into successful software development [2,8,24,26], many software development projects are still failing [15]. Research into software development is likely to continue until we have the development process under better control [8,15,16,26]. In 1994, 31% of all corporate software development projects resulted in cancellation [8]. A more recent study found that 20% of software projects failed, and that 46% experienced cost and schedule overruns or significantly reduced functionality [15]. Another study suggested that failure rates for software development projects are as high as 85% [11]. Clearly, systems development projects can present serious risks to the well being of an organization [15]. Troubled projects can, for example, cause developers to suffer long hours of unpaid overtime, loss of motivation, and burnout, leading to excessive staff turnover and its associated costs.

Recent research that focuses on factors contributing to the failure of information systems projects usually takes a practitioner point of view (such as that of an IT manager, project manager or software developer) [7,9,14,24]. Management also has a view of what constitutes a successful project and, in some cases, this view is different from that of the project manager and developers, while users may take another quite different view. (Note that when we refer to management, we are referring to corporate management; where appropriate, we will explicitly refer to a project manager.)

The major factors that contribute to the success or failure of software systems fall into seven categories: (1) management, (2) customers and users, (3) requirements, (4) estimation, and scheduling, (5) the project manager, (6) the software development process, and (7) development personnel [7,15,20,22]. Our research investigates some of the most influential success factors early in the development process from a developer's perspective. Successful software development projects are commonly cited as having met agreed upon business objectives, and having been completed on time and within budget [12,14,19,25]. Other definitions of success, include the degree to which the project achieved its goals; reliable, maintainable and met the requirements of users; user satisfaction; effective project teamwork and professional satisfaction on the part of the project manager [10,21]; and the extent to which the software is actually used [4,6]. Glass [9] noted a profound difference of opinion between managers and team members concerning software project success, and our recent research agrees with his view [23]. Management tends to consider a project a success if it is delivered on time, within budget and meets specified business objectives [25]. In contrast, developers are more likely to perceive a project to be a success

* Corresponding author. Address: Computer Information Systems, Rider University, 2083 Lawrence Road, Lawrenceville, NJ 08648, USA. Tel.: +1-609-896-5259.

E-mail addresses: jdproc@aol.com (J. Drew Procaccino), june.verner@cis.drexel.edu (J.M. Verner), s.p.overmyer@massey.ac.nz (S.P. Overmyer), medarter@optonline.net (M.E. Darter).

PII: S0950-5849(01)00217-8

whether it is 'completed or canceled' [15]. In a study of several projects, the only criteria for success that had strong agreement among all the involved parties, were 'meets user requirements, achieves purpose, meets time scale, meets budget, happy users, and meets quality' [14]. Glass [7] suggests that problems arising from expectations established at the outset of a project have a greater influence on a developer's perception of success and failure than what happens during project development. For developers, a successful project is one in which the product works the way it is supposed to work, their participation provides a technical challenge and a learning experience that develops new skills [15]. Developers also appreciate when a project's cost and schedule are comparable to normal industry standards, which should mean that excessive unpaid overtime will not be required [15]. On the other hand, developers perception of project failure can include unrealistic schedule expectations, lack of resources, and poor understanding of the project's scope at the outset [15].

Insight into early project factors likely to threaten project success will help project managers and other stakeholders to better predict the likelihood of their project's success. This will allow for corrective action early in the development process. This study is a first step towards the establishment of guidelines for project risk prioritization and mitigation in the early phases of software development. Because of the number of generic risks the project manager has to contend with, we wish to provide a severity roadmap showing the consequences of early project actions (taken or imposed). In addition, we will explore the developer's view of factors that impact project success; a successful project manager needs to be aware of success and failure factors from all stakeholder viewpoints, and to be able to respond to these perceptions, either directly or indirectly, in a timely manner [13]. We anticipate that results of this study (and other studies) will be incorporated into a project management decision support tool.

2. Background

The Standish Group's widely cited research [25] reveals the relative importance of management support, user involvement and requirements for successful project development. In fact, these three software development factors emerge as reoccurring risks.

- *Management support*. Inadequate management practices have far reaching implications for successful software development [1]. A project needs a committed sponsor or champion throughout. This sponsor should participate in the decision-making process and encourage the same commitment from other stakeholders. If no sponsor is available at the start of a project it is important that the project manager obtains one as soon as possible.
- *Customer/users*. The expectations of customers and users are important contributors toward the perceived success or failure of a software project. Realistic expectations can reduce conflict and this in turn supports the perception of project success from both the developer and managerial standpoints [24]. Research suggests that one of the most important contributors to successful project development is end-user involvement [17]. Further, this involvement should occur in *all* phases of system development [1].
- *Requirements*. A clear understanding of the problem that assists in the production of well-defined requirements is essential to the development of a successful system [18,22]. However, we fail to use requirements management to surface (early) errors or problems [3]. Poor requirements are involved in most project failures [7], continue to be a major contributor to today's software crisis [26] and are a huge problem for IS development [18]. One major problem associated with poor requirements is the enormous increase in costs related to the fixing of errors when the errors are discovered during testing or maintenance rather than at the requirements phase (the order of magnitude has been estimated at 1:50 for requirements/testing and 1:200 for requirements/maintenance) [2,13]. Requirements omissions likely represent the most common class, and most expensive type, of error [13]. To make things worse, unrealistic customer and user expectations often arise because projects start with incomplete requirements [26].

Given these software development problems, our investigation reviews a number of software development efforts in order to shed some light on the relative importance and severity of these factors and their effect on project success.

3. Our investigation

In late 1999, we engaged in structured discussions with 21 IT professionals, including developers, technical support personal, project leaders and project managers (referred to in the body of this paper as 'respondents' or 'developers'), from a large financial institution. Our discussions focused on the issues within their organization that they identified as associated with software project success and failure. Based on these discussions, we developed a questionnaire that targeted software development practices in an attempt to discover which development practices are most important for software project success.

Our IT developers each completed two questionnaires—one related to a project he or she considered to be a successful project, the other to a failed project. We also asked developers to consider whether their management perceived the same projects to be a success. As developers were the respondents of this survey, it is the *developer*'s perception of success that we describe here; when we refer to management's view of success we are reporting the developers' perception of management's view of success. Notwithstanding the fact that management may have different perceptions

Table 1
Management support results and management and developer success ($\alpha < 0.05$)

	Question (variable)	MSUCCESS			DSUCCESS		
		No (%)	Yes (%)	Sig.	No (%)	Yes (%)	Sig.
Q1.1	The project manager was not given full authority to manage project	36%	64%	0.161	67%	33%	0.300
	The project manager was given full authority to manage project	16	84		50	50	
Q1.2	The project did not start with a committed sponsor (or champion)	30	70	0.584	73	27	0.165
	The project started with a committed sponsor (or champion)	21	79		48	52	
Q1.3	The sponsor commitment did not last right through the project	**50**	**50**	**0.003**	**87**	**13**	**0.003**
	The sponsor commitment lasted right through the project	**8**	**92**		**39**	**61**	
Q1.4	The sponsor was not involved in project decisions	22	78	0.907	58	42	0.829
	The sponsor was involved in project decisions	24	76		55	45	
Q1.5	The other stakeholders were not committed and involved	22	78	0.907	58	42	0.829
	The other stakeholders were committed and involved	24	76		55	45	

of project success from developers, the developers' perception of what management considers a successful project is also important. Such understanding can foster better communication and professional understanding between management and developers. 'Success' was determined through asking the following two questions:

Do *you* consider this project was a success? [Yes/No] (DSUCCESS)
Did the organization consider this project a success? [Yes/No] (MSUCCESS)

Our questionnaire, was arranged under the following headings: (1) Requirements, (2) Management, (3) Customers and users, (4) Estimation and scheduling, (5) Project manager, (6) Software development process, and (7) Development personnel. Although we addressed all of these topics in the questionnaire, the current research focuses on management support, requirements and customer and user involvement. While none of the questions we ask is new and the good development process practices they cover are found in any software engineering textbook, software development projects are still failing because software project managers continue to make many of the same mistakes that have plagued software development for many years. Because so many organizations are still at CMM level 1, and are wishing to improve their development processes, it is important to discover which practices are most important to success. The results of our investigation are presented in Section 4 followed by discussion in Section 5. Then we conclude in Section 6.

4. Results

We received data from 21 developers, reflecting 42 software development (rather than maintenance) projects. While in normal research terms this number of developers and projects is not large, it is a significant number when we consider that they are all from the same organization and that this research reports an organizational case study. The average number of full-time developers working on the projects was eleven (minimum of one and maximum 80) with an average of five contractors and/or consultants (minimum zero and maximum 10).

We now comment on the results under the following headings: Management Support, Customer/Users, and Requirements. This is followed by sections on prediction of success and failure, where we consider developers' views of project success and their perceptions of management's view of project success.

We used Chi-square tests to determine which of the results are significant with either MSUCCESS or DSUCCESS. These results are shown in Tables 1–3. Overall, while our developers thought that management viewed 77% of the projects successful, developers themselves only considered 44% of the same projects to be a success.

4.1. Management support

We began with a number of assumptions regarding management support including projects are more likely to succeed if:

- project managers are given full authority to manage their project,
- the project has a committed sponsor(s),
- the sponsor supports the project throughout the development process.

We asked five questions related to management support. Table 1 depicts responses to Questions 1.1–1.5. Significant relationships are shown in bold.

Q1.3 Did this [sponsor] commitment last right through the project? This was the only question with answers significant with both MSUCCESS and DSUCCESS. Our responses showed that a project would be more likely to succeed if it has sponsor commitment that lasts throughout the project. Respondents' perceptions of project success, and their perception of management's view of project success, decreased when the sponsor/champion commitment did not last throughout the project. Overall, 62% of

our projects had sponsor commitment from beginning to end. For projects with ongoing commitment, developers thought that 92% were considered successful by management, though only 61% were perceived as successful by the developers themselves. Not surprisingly, there is a significant correlation (0.010) between Q1.2 (the project had a committed sponsor) and Q1.3 (The sponsor's commitment lasted right through the project). According to developers, management believed that 95% of those projects that *both* begun with a committed sponsor *and* had commitment that lasted throughout the project were a success. However, only 64% of developers considered these same projects to be a success.

We have a small, but none the less interesting, group of projects that began without a sponsor, but managed to acquire one later. If a committed project sponsor was acquired late, the project was still likely to be considered a success by developers. When a project began with a committed sponsor, but that commitment did not last through the project, developers considered a relatively small percentage of such projects to be a success (14%); developers thought management viewed 29% of these same projects a success. It was better to begin without a committed sponsor and then acquire one later, as developers considered a larger percentage of such projects to be a success and they perceived that management did as well (50 and 75%, respectively).

4.2. Customer/users

The assumptions behind our customer/user questions are that projects will succeed if they have:

- customer/users who are highly involved and who stay involved throughout the project,
- customer/users who contribute to project estimates,
- customer/users with realistic expectations,
- a low level of customer/user turnover.

We asked seven questions related to customer/users. Table 2 illustrates the relationship of the answers to the customer/user questions with both success variables. Significant relationships are again shown in bold.

Q2.1 What was the level of involvement of customer/users? As customer/user involvement increases, so does the chance of a project being viewed as a success by developers and developers thought that management agreed with them on this point. Across all levels of customer/user involvement, developers thought management categorized a higher percentage of projects as successful than did the developers themselves. Overall, 36% of our projects had high customer/user involvement, 24% had a reasonable level, 12% had some involvement, 26% had little involvement and only 2% had no customer/user involvement. Fig. 1 shows the relationship between respondents' own views and their perception of management's views of success across the varying levels of customer/user involvement.

Q2.2 Did the involved customer/users stay right through the project? Developers are more likely to view, and they believe that management is also more likely to view, a project as successful if customers and users stay involved throughout a project: their perceptions of success decreased noticeably when this did not occur. Overall, 74% of projects had their customer/users' attention throughout the project. For those projects where the involved customer/users did *not* stay throughout the project, developers thought that management viewed 46% a success, though only 9% were considered a success by developers themselves.

Q2.3. What level of confidence did the customer/user have in the project manager/team members? The higher the level of confidence that customers/users have in the project manager and development team, the more likely developers are to consider the project a success. Forty-one percent of projects had high or very high customer/user confidence in the project team, 33% reported an average level, and 26% reported either low or very low. If we consider only those projects reported with 'high' or 'very high' levels of customer/user confidence in the project manager/team, developers thought that management considered 92% of these projects as a success although only 77% were considered a success by the developers themselves. For those projects reported to have an 'average' level of confidence, developers thought that management still considered 69% successful, though respondents' own view of success for these projects fell to 29%. When viewing projects with only a 'low' or 'very low' level of confidence, developers thought that management considered 77% of these projects a success, though respondents' own views of the success rate dropped markedly to 44%.

Q2.4 Were the customer/users involved in making schedule estimates? Both developers' perception of success and their perception of management's view of success increase when customer/users are involved in making schedule estimates. One has to question why this is so. Is this seen as a way to increase customer and user involvement and commitment, or are we cynical in suggesting that it might have something to do with the ability to blame the customer if the schedules estimates are not met?

Q2.6 Did the customer/users have realistic expectations? Developers are more likely to consider the project a success if the customers/users have realistic expectations and their perceptions of success decrease noticeably when the customer/users do not have realistic expectations. Developers thought that 92% of projects having customer/users with realistic expectations were, according to developers, considered a success by management, while only 75% of these same projects were considered a success by developers themselves. When considering only those projects where the customer/users did *not* have realistic expectations, developers thought that

Table 2
Customer/user results and management and developer success ($\alpha < 0.05$)

	Question (variable)	MSUCCESS			DSUCCESS		
		No (%)	Yes (%)	Sig.	No (%)	Yes (%)	Sig.
Q2.1	The level of involvement of customer/users was average/below average	**44**	**56**	**0.011**	**94**	**6**	**0.000**
	The level of involvement of customer/users was above average	**9**	**91**		**32**	**68**	
Q2.2	Customer/users did not stay right through the project	**55**	**45**	**0.003**	**91**	**9**	**0.007**
	Customer/users stayed right through the project	**11**	**89**		**43**	**57**	
Q2.3	The level of confidence the customer/user have in the project manager/team members was average/below average	30	70	0.191	**79**	**21**	**0.000**
	The level of confidence the customer/user have in the project manager/team members was above average	13	87		**24**	**76**	
Q2.4	Customer/users were not involved in schedule estimates	**38**	**62**	**0.007**	64	36	0.202
	Customer/users were involved in schedule estimates	**0**	**100**		44	56	
Q2.5	The level of customer/users staff turnover you have to contend with was average/below average	33	67	0.661	67	33	0.702
	The level of customer/users staff turnover you have to contend with was above average	22	78		55	45	
Q2.6	Customer/users did not have realistic expectations	31	69	0.130	**68**	**32**	**0.013**
	Customer/users had realistic expectations	8	92		**25**	**75**	
Q2.7	You did not run into problems due to the large number of customer/users involved	26	74	0.426	61	39	0.238
	You ran into problems due to the large number of customer/users involved	13	88		40	60	

69% were considered a success by management, while only 32% were considered a success by the developers themselves.

4.3. Requirements

We expected that those projects viewed as successful would:

- begin with complete and accurate requirements,
- have gathered requirements using a specific methodology,
- have a well-defined scope,
- have consistent requirements throughout the project.

We asked a total of nine questions related to requirements. Six questions exhibited no statistically significant relationship with *either* the developer's perception of management's view of success (MSUCCESS) or their own view of success (DSUCCESS) (see Table 3). Significant relationships are again shown in bold.

Q3.2a Were the requirements complete and accurate at the start of the project? We were unable to perform Chi-square tests with this variable because no project began with complete and accurate requirements. While it is possible to have a successful project that does not start with complete requirements, it is difficult do so if a waterfall lifecycle is used. It is interesting to note that only four projects used prototyping, two within a JAD approach (none used iterative development). Developers thought that all four projects were considered a success by management while three projects were considered a success by developers who complained about a 75% overrun for one of the prototyped projects.

Q3.2b If requirements were not complete and accurate at the start of the project, were they completed adequately? Developers thought that management viewed 94% of projects that had completed their requirements to be a success, while developers themselves considered only 67% of the same projects to be a success. While developers thought management considered 64% of projects that *never* had complete and accurate requirements to be a success, they, themselves, considered only 23% of those same projects to be a success.

Q3.5 Did the customer/users make adequate time available for requirements gathering? Developers thought management considered 79% of projects that had customer/users who made adequate time available for requirements gathering successful, while they themselves considered only 60% of these same projects to be a success. While they thought that management considered 71% of projects that did *not* have customer/users who made adequate time to be a success, developers themselves considered only 13% of these projects to be a success.

4.4. Prediction of success or failure

Next, we conducted the multivariate analysis to explore the impact that the variables generated from the management support, customer/users and requirements questions have on perceptions of project success. Since the success variables are coded as zero for 'unsuccessful' and one for 'successful', we were unable to use ordinary least squares regression, which relies on a continuous dependent variable. Instead, we conducted a logistic regression analysis. We also recoded variable 2.1 (What was the level of involvement of customer/users?), from its original five-point scale to a two-point scale (above average level of involvement = 1; not above average level of involvement = 0), to avoid using dummy variables. It should be noted that there are several prediction equations that could be developed based on our data. We have focused on the best all around equation by developing one that was both a strong predictor of project failure *and* a good predictor of project success.

4.4.1. Prediction of success from the developer viewpoint

Three variables entered our logistic regression equation and together, they explained 49% of the variance in developer's view of project success (DSUCCESS, the dependent variable; $R^2 = 0.494$). See Table 4, corresponds to management support (Q1.3), customer/users (Q2.6) and requirements (Q3.2b) variables from the logistic regression. Parameter estimates for project success are all significant at the 10% level. Our logistic equation correctly predicts 85% of projects overall (86% of the unsuccessful projects and 82% of the successful projects).

4.4.2. Prediction of success from developers views of management

One variable (Ql.3b, 'If requirements were not complete and accurate, at the start of the project, were they completed adequately?') entered our logistic regression equation and it explained 31% of the variance in developers reported view of management's opinion (MSUCCESS, the dependent variable; $R^2 = 0.305$). See Table 5. The parameter estimate for this factor is significant at the 10% level. Our logistic equation correctly predicts 77% of projects overall (78% of the unsuccessful projects and 77% of the successful projects and 77% of the successful projects).

From everyone's perspective, the commitment of the sponsor throughout the project is critical to project success. But this is not true for MSUCCESS which only has requirements as a parameter. From the developer perspective, the next most important factor for success is that requirements are completed adequately at some stage and that customers/users have realistic expectations. The above results, will help project managers to predict the likely success or otherwise of their projects, and to be able to do so with reasonable accuracy. It is somewhat surprising that we are able to predict the success (or otherwise), of software development project with this degree of accuracy from so few project variables.

We also developed exploratory path analysis diagrams in order to provide graphic depictions of some of the factors leading to developer's perception of success (DSUCCESS). We did not chart management's perception of success because it was acquired secondhand (through developers). Path coefficients are standardized Beta weights and we have only included Betas that were greater than 1.

Fig. 2 corresponds to management support, customer/users and requirements, and the variables shown in Tables 1–3 that were significant with developer's view of success (DSUCCESS) based on Chi-square tests, as well as those variables that were significant with each of these variables, but yet made some theoretical contribution to DSUCCESS. 'Other influencers' represent unexplained variances, which is derived from the square root of $(1 - R^2)$ for each dependent variable (the dependent variables include Q3.5, Q2.1, Q3.2b, Q2.2, Q2.3, Q2.6 and DSUCCESS).

Our proposed model includes the following assumptions and variables:

- Whether sponsor commitment lasted throughout the project (Q1.3) has a causal effect on both whether customer/users made adequate time available for requirements gathering (Q3.5), and the level of involvement of customer/users (Q2.1).
- Whether customer/users made adequate time available for requirements gathering (Q3.5), whether the project scope was well defined (Q3.3) and whether customer/users were involved in making schedule estimates (Q2.4) have a causal effect on whether requirements were completed adequately at the beginning of the project (Q3.2b).
- The level of customer/users involvement (Q2.1), whether customer/users were involved in making schedule estimates (Q2.4) and whether developers encountered problems due to a large number of customer/users (Q2.7) have a causal effect on whether the involved customer/users stayed throughout the project (Q2.2).
- Whether requirements were gathered using a particular method (Q3.l), whether there was a single central repository for requirements (Q3.6) and whether requirements resulted in well-defined software deliverables (Q3.7) have a causal effect on the level of confidence that customer/user had in the project manager/team members (Q2.3).
- The level of confidence that customer/user had in the project manager/team members (Q2.3) has a causal effect on whether the customers had realistic expectations (Q2.6).
- Whether the customers had realistic expectations (Q2.6), whether the involved customer/users stayed throughout the project (Q2.2) and whether requirements were completed adequately at the beginning of the project (Q3.2b) have a causal effect on developer's perception of success (DSUCCESS).

Table 3
Requirements results and management and developers success ($\alpha < 0.05$)

	Question (variable)	MSUCCESS			DSUCCESS		
		No (%)	Yes (%)	Sig.	No (%)	Yes (%)	Sig.
Q3.1	Requirements were not gathered using particular method	24	76	0.907	64	36	0.295
	Requirements were gathered using particular method	22	78		47	53	
Q3.2a	Requirements were not complete and accurate at project start	24	76	n/a	58	42	n/a
	Requirements were complete and accurate at project start	0	0		0	0	
Q3.2b	Requirements were not complete and accurate at start and they were not completed adequately	**36**	**64**	**0.031**	**77**	**23**	**0.005**
	Requirements were not complete and accurate at start, but they were completed adequately	**6**	**94**		**33**	**67**	
Q3.3	The scope of the project was not well defined	33	67	0.120	70	30	0.073
	The scope of the project was well defined	12	88		41	59	
Q3.4	The project scope did not change during the project	33	67	0.342	58	42	0.944
	The project scope changed during the project	19	81		57	43	
Q3.5	The customer/users did not make adequate time available for requirements gathering	29	71	0.588	**87**	**13**	**0.004**
	The customer/users made adequate time available for requirements gathering	21	79		**40**	**60**	
Q3.6	There was not a single central repository for requirements	13	87	0.167	71	29	0.150
	There was a single central repository for requirements	32	68		48	52	
Q3.7	The requirements did not result in well-defined software deliverables	15	85	0.184	67	33	0.218
	The requirements resulted in well-defined software deliverables	33	67		47	53	
Q3.8	The size of project did not negatively affect requirements elicitation	30	70	0.383	55	45	0.855
	The size of project negatively affected requirements elicitation	18	82		58	42	

When considering only the significant paths (Q2.4 → Q2.2 → DSUCCESS), this model explains about 25% of the variance in DSUCCESS ($R^2 = 0.252$). It should be noted that our model resulted in a fairly large number of non-significant paths because of our relatively small sample size (42). A thorough path analysis would require at least 200 cases. However, as previously noted, our proposed model is exploratory in nature. Although limited, our model does serve to illustrate some of the important aspects of the development process for this particular organization. In Section 6, we outline some of our findings that do not necessarily agree with conventional wisdom of software development.

5. Discussion and conclusions

This study leads us to a number of conclusions about project success from the perspective of developers and their perception of management and project success. We found that developers are very critical of the projects that they worked on and considered fewer projects successful (44%) than they thought management did (77%). We also found various aspects of development that we assumed would be important to developer's perception of success, but proved to lack statistical significance (based on our 42 cases). The following is a summary of the more interesting results related to the *developers perceptions* of management support, customer/users and requirements.

Management support:

- It is *not* necessary for project success for the project manager to be given full authority to manage the project.
- Starting a project with a committed sponsor is important.
- A project sponsor who drops out of this role in the project has a more detrimental effect on perceptions of project success than starting without a sponsor and picking up one later. With no sponsor commitment, developers perceive little chance of project success.
- Projects can be perceived as successful without the sponsor's involvement in project decision-making. This helps define the role of project sponsor as political, rather than production support.

Customer/users:

- The higher the level of confidence that customers/users have in the project manager and development team, the more likely developers are to consider the project a success.
- The involvement of customers/users in making schedule estimates does *not* increase the likelihood that developers will consider the project a success.

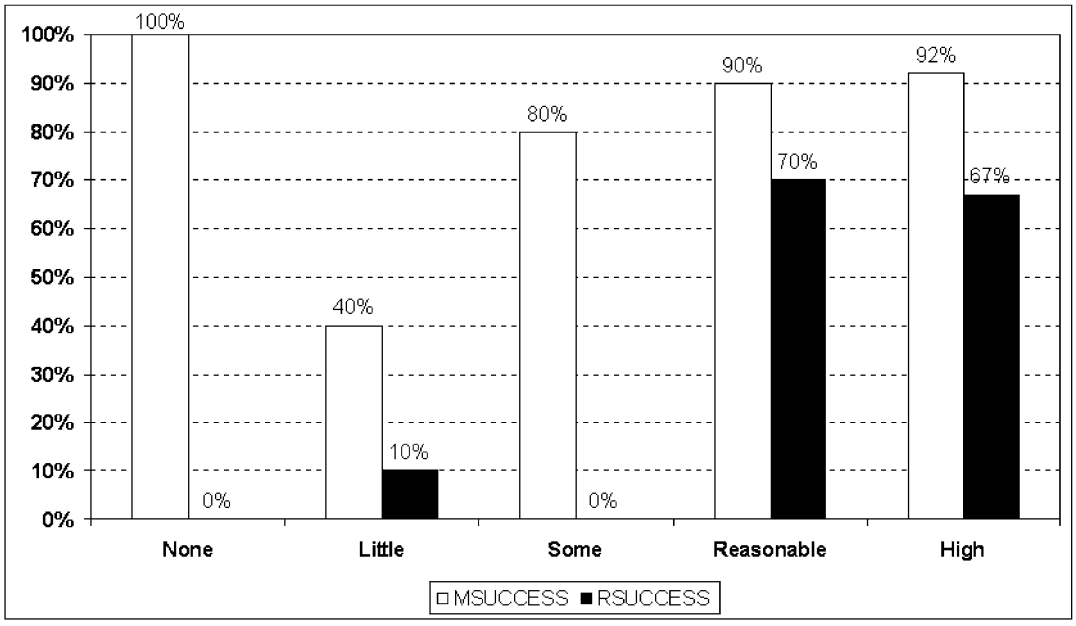

Fig. 1. Percent of projects considered a success by level of customer/user involvement.

Requirements:

- Using a requirements gathering methodology did *not* appear to increase the perception of project success.
- Changing the scope of a project during development does *not* change the perception of project success.

5.1. Manager's and developer's views of success

As noted earlier, 44% of the projects studied were deemed successful by the involved developers. This number is far lower than the 77% reported by these same developers when asked to evaluate management's perception of success for the same projects.

If the developers have accurately reflected management's conclusions, the difference in viewpoint with respect to project success can be attributable to the different perspectives, motivation, and responsibilities typically associated with the respective roles of management and developers on projects. For example, considering Q2.4, while management may view customer input into schedule estimates as politically important (or expedient), developers might view this input as professionally irrelevant or even intrusive, since the typical customer input may not be based on technical factors. Another example of varied perspective is reflected in Q3.5. Wherein the developers are very concerned with sufficient user access for requirements purposes, while they did not think that managers viewed this as important at all. Since requirements gathering is the responsibility of the developers, it is no surprise that they consider customer availability to be key. Developers thought that management on the other hand, is concerned with the final product, and considers requirements as a deliverable from the developers, not from the customer. This difference in both perspective and responsibility potentially explains this effect. One particularly interesting result was that of Q3.2a, in which developers thought that management was unconcerned with having accurate requirements at the beginning of a project, while developers themselves were very concerned.

Table 4
Logistic equation for prediction of developer success (DSUCCESS)

	Variable	Parameter estimate	Standard error	Probability
Q2.6	Did the customer/users have realistic expectations?	1.671	0.985	0.090
Q3.2b	If requirements were not complete and accurate, were they completed adequately?	1.527	0.838	0.069
Q1.3	Did sponsor commitment last right through the project?	2.117	0.991	0.033
	Constant	−2.867	0.985	0.004

Table 5
Logistic equation for prediction of management success (MSUCCESS)

	Variable	Parameter estimate	Standard error	Probability
Q1.3	Did sponsor commitment last right through the project?	2.442	0.911	0.007
	Constant	0.000	0.535	1.000

Alternatively, if the developers' perceptions of management were not always adequate surrogates for management responses, then some of our conclusions should be considered in a different light. For the most part, we believe that the developers have reasonably reflected management thinking. However, there are some anomalies (such as an apparent lack of concern about customer/user expectations) that merit further investigation.

6. Lessons learned

One of the most compelling reasons to measure the perceptions of those involved with software development is that participant perceptions of project success, consistent or not with reality, can dramatically affect the health of a project manager, a project team, and thus the health of the project. There is a vast difference between developers' perceptions of project success factors, and their perceptions of how management view the same factors.

What do developers consider is important to management's view of a successful project? Developers perceive that management takes a politically oriented view. It is important to management to keep customers and users, rather than developers, happy, and to be able to pass (a share of) the blame to someone else if the schedule is not met.

What do developers consider will lead the development team to consider a project successful? Because developers take a mainly inward-looking view of project success concentrating on the things that affect them and their ability to do their job properly, a successful project (and project manager) has:

(a) Customers/users who make adequate time available for them (a project sponsor will help here),
(b) a project scope that is well-defined [this will follow on from (a)], and
(c) carefully managed customer/user expectations.

With the current level of IT shortages, management needs to be aware of what keeps developers happy in order to avoid staff turnover and loss of motivation. These efforts must include effective management of the customer/users/developers relationship.

Further questions that have emerged from this analysis include:

- Can the results of a survey such as this, done in a single organization, be generalized to the broader population?

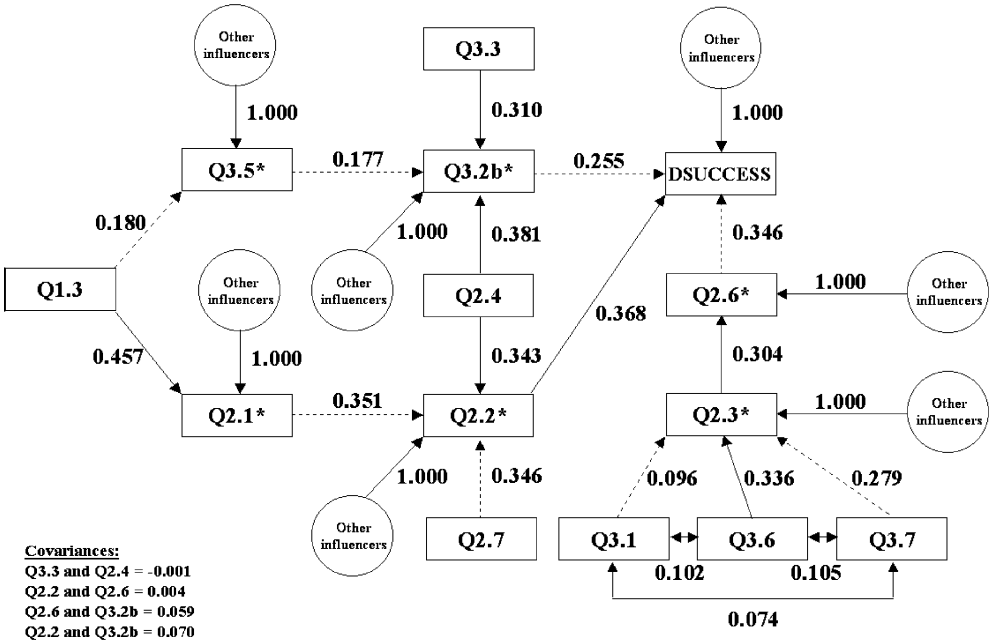

Fig. 2. Overall path analysis for significant variables and DSUCCESS.

The brief answer to this question is 'yes'. When 67 other developers (from a number of different organizations) answered the same questionnaire, the results were very similar to those above [25].

- How consistent are management's own perceptions of project success and developer's views of those perceptions in both this, and other organizations? If they are consistent, then remedy is straightforward. If not, management must do a better job of understanding and possibly manipulating developers' perceptions to increase productivity and job satisfaction.
- Are customer's perceptions of project success consistent with both management's and developer's perceptions? If not, then perhaps recommendations can be made to achieve a better working relationship between customers/users and developers, so that views may be more consistent.
- How influential are specific perceptual variables in predicting or changing the course of unsuccessful projects? What are the critical success factors? We have suggested that on-going sponsor support may result in a number of positive actions such as users making sufficient time available for requirements gathering. Additional data, combined with revised path analysis, should help to provide some answers to these questions.

In future research, we plan to more fully investigate some of the detailed factors that comprise success as perceived by developers. We anticipate that this work will build on the studies of Steve McConnell [15], Bob Glass [9], Kurt Linberg [14], the Standish Group [25], and Tom DeMarco and Timothy Lister [5].

References

[1] K. Amoako-Gyampah, K.B. White, When is user involvement not user involvement, Information Strategy: The Executive's Journal 13 (4) (1997) 40–45.
[2] B.W. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cliffs, 1981.
[3] C. Clavadetscher, User involvement: key to success, IEEE Software 15 (2) (1998) 30–32.
[4] F.D. Davis, Perceived usefulness, perceived ease of use and user acceptance of information technology, MIS Quarterly 13 (3) (1989) 319–339.
[5] T. DeMarco, T. Lister, Peopleware: Productive Projects and Teams, second ed, Dorset House Publishing Co, New York, 1999.
[6] D. Gefen, D. Straub, The Relative importance of perceived ease-of-use in IS adoption: a study of e-commerce adoption, JAIS 1 (8) (2000) 1–30.
[7] R.L. Glass, Software Runaways, Prentice-Hall, New Jersey, 1998.
[8] R.L. Glass, How not to prepare for a consulting assignment and other ugly consultancy truths, Communications of The ACM 41 (12) (1998) 11–13.
[9] R.L. Glass, Evolving a new theory of project success, Communications of The ACM 42 (11) (1999) 17.
[10] H. Nicole, Understanding the link between IT project manager skills and project success, research in progress, Proceedings of SIGCPR Conference, Evanston, IL, 2000, pp. 192–195.
[11] T. Hoffman, Study: 85% of IT departments fail to meet business needs, Computerworld 33 (41) (1999) 24.
[12] C. Jones, Patterns of large software systems: failure and success, IEEE Computer 28 (3) (1995) 86–87.
[13] D. Leffingwell, D. Widrig, Managing Software Requirements: A Unified Approach, Addison-Wesley, Reading, MA, 2000.
[14] K.R. Linberg, Software developer perceptions about software project failure: a study, The Journal of Systems and Software 49 (2/3) (1999) 177–192.
[15] S. McConnell, Rapid Development, Microsoft Press, Redmond, 1996.
[16] A.J. Nolan, Learning from success, IEEE Software 16 (1) (1999) 97–105.
[17] M. Paulk, B. Curtis, M. Chrissis, C. Webster, Capability maturity model for software, Technical Report, CMU/SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon, 1993.
[18] E.J. Pedhazur, Multiple Regression in Behavioral Research: Explanation and Prediction, second ed, Holt (Rinehart and Winston), New York, 1982.
[19] K. Jeffrey, Pinto, D.P. Slevin, Project success: definitions and measurement techniques, Project Management Journal 19 (1) (1988) 67–72.
[20] R. Pressman, Software Engineering: A Practitioners Approach, McGraw-Hill, New York, 1996.
[21] R. Pressman, Fear of trying: the plight of rookie project managers, IEEE Software 15 (1) (1998) 50–51 see also page 54.
[22] J. Drew Procaccino, J.M. Verner, Early risk factors for software development, Proceedings of the 12th European Software Control and Metrics Conference, London, April 2001, pp. 107–116.
[23] J. Drew Procaccino, J.M. Verner, Practitioner's perception of project success: a pilot study, IEEE International Journal of Computer and Engineering Management (2001) in press.
[24] K.D. Schenk, N.P. Vitalari, S.K. Davis, Differences between novice and expert systems analysts: what do we know and what do we do?, Journal of Management Information Systems 15 (1) (1998) 9–51.
[25] Standish Group, CHAOS, http://www.pm2go.com/sample research/chaos 1994 2.asp (Accessed March 5, 2001).
[26] J.M. Verner, S.P. Overmyer, K.W. McCain, In the 25 years since the mythical man-month what have we learned about project management?, Information and Software Technology 41 (14) (1999) 1021–1026.