Managing user expectations on software projects: Lessons from the trenches

Stacie Petter *

Information Systems and Quantitative Analysis, University of Nebraska at Omaha, The Peter Kiewit Institute, 1110 S 67th Street, Omaha, NE 68116, United States

Received 26 May 2008; accepted 29 May 2008

Abstract

Prior research has demonstrated that inappropriate user expectations can have a downstream effect on the use of and user satisfaction with an information system. Although several studies have examined the importance of managing user expectations, there has been little work on how project managers can and should manage user expectations on a software project. In this research, software project managers were interviewed to discover the strategies and tactics that were successful and not-as-successful in managing user expectations. Three general strategies for managing user expectations were identified: user involvement, leadership, and trust. These findings as well as the implications for researchers and practitioners are discussed.

Keywords: Managing projects; Managing stakeholders

1. Introduction

Software projects are fraught with risks, with many risks common to nearly all projects. In an attempt to identify the risks that appear most often on software projects, Schmidt et al. [1] conducted a Delphi study with software project managers from three different countries. Overall, fifty-three risk factors were identified by these three panels, and eleven risk factors were consistent across the panels. Project managers have little to no control over most of these risk factors. Yet, three risk factors identified by each of the Delphi panels are within the software project manager’s control: failure to manage end user expectations, misunderstanding requirements, and insufficient/inappropriate staffing.

This study by Schmidt et al. was useful in identifying risks; however, the authors suggested that future research should determine how to address these risks during a software project. This research explores one of these risk factors by considering the following question:

What strategies and tactics can be utilized by software project managers to address the risk “failure to manage end user expectations” on a software project?

To answer this research question, interviews were conducted with twelve software project managers from a large IT and management consulting organization. The goal was to discover lessons learned associated with managing user expectations. Each software project manager was asked to recall an experience in which s/he managed user expectations successfully and an experience in which the management of user expectations was not-as-successful. These interviews yielded a framework of general strategies and specific tactics to be used by software project managers to address the risk of “failure to manage user expectations.”

The remainder of this paper is organized as follows. First, the literature associated with managing user expectations is examined to further motivate this research. Empirical research that has examined the effect of user expectations on project and software success is highlighted. In addition, two related research streams are discussed that justify the importance of managing expectations: expectation-confirmation theory and service quality. Section 3
describes the research study in terms of the data collection, data analysis, and results. Three general strategies for managing user expectations were identified based on the interview data: user involvement, leadership, and trust. These findings are presented in light of supporting literature and the implications for practitioners and researchers are discussed in Section 4. Section 5 offers concluding thoughts about the contributions and insights from this research.

2. Literature review

Szajna and Scammell [2] defined user expectations as “a set of beliefs held by the targeted users of an information system associated with the eventual performance of the IS and with their performance using the system” [p. 494]. Managing user expectations is defined as “the actions a project manager performs to ensure that the assumptions held by the user for a software project are realistic and consistent with the software deliverable promised by the project team” [3,4]. These expectations “must be correctly identified and constantly reinforced in order to avoid failure” [1, p. 15].

2.1. Impact of user expectations on software projects

Baccarini et al. [3] found that IT project managers identified “unrealistic expectations” as the third highest ranked project risk in a list of 27 risk factors derived from the literature. The Project Management Institute has also stated that meeting user expectations is one of the primary criteria for project success [5]. Success may be categorized as both project success (i.e., the project management process, measured by adherence to time, cost, and functionality objectives) and product success (i.e., the outcomes of the final product or software, measured by system and information quality, use, satisfaction, and net benefits) [6–8]. Failure to properly manage user expectations can affect both types of success. If users have improper assumptions about the features that will be delivered, then users may perceive that the functionality objectives associated with project success have not been met. In addition, these faulty expectations may lead to lower levels of product success as measured by perceived system quality, perceived information quality, or user satisfaction.

Mahmood et al. [9] conducted a meta-analysis on several antecedents to user satisfaction of information systems. They identified seven studies that examined the relationship between user expectations and user satisfaction and found a significant relationship (effect size = 0.458). User satisfaction is a widely-used measure of product success and has a downstream affect on other important measures of information systems success, such as use, individual net benefits, and organizational benefits of an information system (e.g., [8,10,11]). Based on this research, it seems reasonable to infer that the actions of the project manager to manage user expectations during the project can have a downstream effect on the measured effectiveness of the software.

2.2. Related research

There are two streams of literature from marketing that have direct relevance to the phenomenon of managing user expectations: expectation-confirmation theory and service quality. Both streams of literature have been used within the information systems domain and have the ability to explain the importance and relevance of managing user expectations within a software project.

2.2.1. Expectation-confirmation theory

Expectation-confirmation theory is widely cited in the marketing literature to examine and understand consumer behavior [12]. This theory suggests that individuals first develop an expectation of a product or service (or software, in this case) prior to its use. After using the product or service, the individual develops opinions regarding its performance. The performance of the product or service is compared to the individual’s expectations. These expectations are either confirmed or disconfirmed. The disconfirmation level affects the individual’s level of satisfaction, which in turn informs the consumer’s repurchase intention. Fig. 1 illustrates this framework as posited by Oliver [13].

An in-depth literature review of consumer satisfaction found much support for expectation-confirmation theory [14]. The research that preceded the development of expectation-confirmation theory suggests that a strong inter-relationship exists between expectations, performance, and satisfaction. Contrast theory suggests that when expectations do not match actual performance, the difference between the expectation and performance will be exaggerated [14,15]. Generalized negativity theory suggests that a disconfirmation of expectations is actually worse than having no expectations at all [14,16]. The importance of setting appropriate expectations and meeting those expectations has been widely supported within the consumer marketing literature.

Although these theories were developed in consumer settings, they have been adapted and applied to information systems. Staples et al. [17] examined librarians pre- and post-implementation of a new cataloging software. This study revealed that individuals with higher expectations of the software pre-implementation were less satisfied with the software post-implementation than individuals with lower expectations prior to the software’s implementation. Ginzberg’s [4] work on pre-implementation expectations found that users with appropriate expectations about an information system had higher levels of use and user satisfaction than those with less realistic expectations. In his work, he realized that “realism of expectations was a better predictor of success” than other measures used in his study.

Others have found that expectations and confirmation of expectations can have downstream or longitudinal effects. Bhattacharjee [12] found the confirmation of expectations affected perceived usefulness, satisfaction, and continuance intention to use online banking software in his work. In a series of longitudinal experiments, Davis and Venkatesh
Parasuraman et al. [19] examined the literature studying the consumer’s expectations and perceptions of a service. This study confirms that it is critical to properly manage user expectations, even during early stages of the project, such as requirements analysis and design.

2.2.2. Service quality

Another stream of literature useful for understanding expectations is the concept of service quality within the marketing literature. This construct specifically examines user expectations, even during early stages of the project, as requirements analysis and design.

Expectations

Perceived Performance

Disconfirmation

Satisfaction

Fig. 1. Expectation-confirmation theory.

[18] determined that users develop their opinions and expectations during the prototyping stage and remained consistent even after they had an opportunity to use the system. This study confirms that it is critical to properly manage user expectations, even during early stages of the project, such as requirements analysis and design.

Within the information systems literature, researchers have considered service quality in terms of the information systems department [20,21]. One function of many information systems departments is to develop and create software. Software project management is not simply product development [22], but is a combination of product and service delivery to offer a solution to the user. Extrapolating the service quality literature to software project management, the software project manager is providing a service of managing software development for the users. Therefore, the stream of service quality literature would suggest that it is important for the software project manager to ensure the users have reasonable expectations during this service encounter of managing the project.

One operationalization of service quality is a measurement instrument known as SERVQUAL. This measure examines the difference between a person’s expectations and perceptions, known as the gap, and suggests where the firm is over-promising and/or under-delivering the service [23]. This gap reflects an attitude about individuals that is closely related to satisfaction [23]. The SERVQUAL instrument suggests multiple dimensions for measuring service quality: tangibles (communication, personnel, and appearance of facilities); reliability (dependability to perform services); responsiveness (promptness of service); assurance (politeness, knowledge, and trust shown during service encounter); and empathy (attention to individuals’ needs) [24].

Pitt et al. [21] acknowledged the growing reliance that organizations have on the information systems department to assist users and maintain software. They suggested that the service quality offered by the information systems department plays an important role in the perception and understanding of the effectiveness of software. Pitt et al. went on to adapt the SERVQUAL instrument from the marketing literature and apply it to an information systems context. Furthermore, they evaluated the instrument and offered reasons why service quality should be considered either a dependent or independent variable of information systems success.

Others have also evaluated service quality in the information systems literature. For example, Kettinger and Lee [20] found that the higher the SERVQUAL gap score (i.e., the difference between perceptions and actual service), the lower the level of satisfaction. Watson et al. [25] conducted a longitudinal study of service quality found that service quality is ongoing and not a short-term effort. They found that open communication with clients, developing repeatable sound processes for service, and training service personnel are some of the important activities an information systems department must perform to maintain service quality and meet expectations.

Some of the dimensions of service quality have been shown to have an impact on user satisfaction in an e-commerce environment [26]. Other studies have shown the importance of a single dimension, such as responsiveness [27], on user satisfaction. It is still uncertain which of the service quality dimensions are most relevant in a software project management context; however, the idea that the difference between expected service quality and actual service quality (as perceived by the user) impacts the user’s experience is important in that it is critical for project managers to properly manage and meet users’ expectations.

The discussions of expectation-confirmation theory and service quality provide evidence of the importance of understanding and managing user expectations throughout a software project. These theories help to explain why it is important for software project managers to be concerned with managing user expectations, yet this literature does little to explain specifically how a project manager can address this risk. This research study addresses this shortcoming.

3. Research study

To examine the research question, an inductive approach was taken to identify the strategies and tactics that can be
used by software project managers to manage user expectations on a software project. Two researchers interviewed software project managers in a large, IT and management consulting company of over 75,000 employees across nearly 50 countries. Twelve informants were selected with a range of project management experience. Most of these informants were deployed on projects within the government sector. Table 1 lists the informants along with job titles, years at that job title, and years of experience in project management.

The interviews consisted of semi-structured questions related to challenges in managing user expectations, developing solutions to address the problem, and the results of the actions taken. Each software project manager was asked to recall two situations in which they faced challenges in managing user expectations: one situation in which the software project manager was proud of the end result and the other situation, in which looking back via hindsight, was not as pleased with the result.1 Data on twenty-four different situations in which the software project manager had to manage user expectations was obtained via the questions asked in the interview. After exhausting the questions, informants were offered the opportunity to provide other insights regarding managing user expectations. The questions used for the interviews are listed in Appendix A.

All interviews were conducted via telephone. Two researchers were present at each interview and took turns asking questions. Each researcher recorded personal notes and tape-recorded each interview to use for the analysis. Each interview was transcribed and yielded over 150 pages of text. Anonymity and confidentiality was assured to all informants.

### Table 1

<table>
<thead>
<tr>
<th>Name</th>
<th>Job title</th>
<th>Time in position (years)</th>
<th>Project management experience (years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abigail</td>
<td>Consultant</td>
<td>2.5</td>
<td>3</td>
</tr>
<tr>
<td>Ryan</td>
<td>Consultant</td>
<td>3.5</td>
<td>3</td>
</tr>
<tr>
<td>Emily</td>
<td>Manager</td>
<td>.5</td>
<td>3</td>
</tr>
<tr>
<td>Anthony</td>
<td>Manager</td>
<td>.5</td>
<td>3</td>
</tr>
<tr>
<td>Hannah</td>
<td>Manager</td>
<td>1.5</td>
<td>3</td>
</tr>
<tr>
<td>Joseph</td>
<td>Manager</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Christopher</td>
<td>Manager</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Jacob</td>
<td>Manager</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Joshua</td>
<td>Manager</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Matthew</td>
<td>Manager</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Ethan</td>
<td>Senior Manager</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Andrew</td>
<td>Senior Manager</td>
<td>1.5</td>
<td>20</td>
</tr>
</tbody>
</table>

1 All names have been changed to ensure confidentiality.

### 3.1. Data analysis

The transcripts were read several times to gather a full understanding of the strategies that each project manager used to address managing user expectations.2 After ensuring a full understanding of the transcripts, two partially ordered meta-matrix displays were developed to conduct cross-case analysis using word processing software [28]. One partially ordered meta-matrix display focused on the successful attempts to manage user expectations; the second partially ordered meta-matrix display contained the not-as-successful attempts to manage user expectations. Within each display, the following information was captured: the timing of the story within their career (if mentioned in their description); a description of the user group; the situation encountered in which expectations needed to be managed; the solution the project manager employed to address the situation; the result or outcome of the solution; and any insights the informant noted about the situation. By developing these displays, it was possible to compare the tactics used for managing user expectations in both the successful and less successful situations. Each unique tactic was identified and was placed in a column as either a successful or less successful tactic to address the problem of managing user expectations. Other tactics were identified based on the self-reflection of a project manager who realized “in hindsight, this is what I should have done.” Some software project managers used a single tactic to deal with the problem, while others may have incorporated multiple tactics in their solution. Appendix B shows an excerpt of the partially ordered meta-matrix displays and identifies how tactics emerged from the data.

These tactics were then grouped into logical categories. After identifying both the tactics and the categories, each of the transcripts was re-examined to keep in mind the hermeneutic whole of the data [29]. The tactics and strategies identified in the analysis were examined in the context of the entire transcript from each informant. Both the successful and not-as-successful situations were considered. It was confirmed that when an individual had a successful outcome in managing user expectations, one or more of the successful tactics were employed. When a project manager had a not-as-successful outcome in managing user expectations, one or more of the less successful tactics were used. In addition, for these not-as-successful situations, the successful tactics were not discussed by the project managers as being strategies they tried on the project.

### 3.2. Results

Three broad strategies to manage user expectations emerged from the data analysis. Table 2 lists the strategies and tactics identified in the research study. Each of the

---

1 These questions on managing user expectations were asked as part of a larger study on knowledge reuse. Only the relevant data and analysis associated with managing user expectations is reported in this research.

2 All data analysis was conducted solely by the author of this manuscript.
strategies is discussed in detail with relevant quotes from the interviews below.

3.2.1. User involvement

The most common theme that occurred within these interviews was the importance of working interactively with the users. Many software project managers recalled spending a lot of time talking with the users to understand their needs for the software. More specifically, the “talking” that occurred with the user focused more on listening to the user’s concerns and asking questions.

It was an effort in learning how to listen... I let him talk about what he worked on, what his area was, what his experiences were, why he was the person and the only person that knew it. (Jacob – Successful Outcome)

Many of the situations involved users facing a new process for doing their jobs as well as a new technology. Many users were resistant to this change and were hesitant to buy-in to the project. Users needed to understand that “we are all in this together” to be put at ease.

We developed a dialogue with the user group so they would understand this is something we were going to do with them and not to them. (Ethan – Successful Outcome)

Another successful tactic was to let the users make choices about trade-offs between the budget, schedule, and functionality. The users asked Anthony for extra functionality very close to the release date. He knew that adding resources to the project would get the job done, but it would cost the users more money and it may be lacking in quality. Rather than absorbing the costs and attempting to add the functionality, Anthony offered another alternative – extending the schedule.

We gave a ballpark estimate in terms of schedule versus cost. For example, the implementation would be two weeks later, but the costs would remain on target. Once we gave those estimated numbers to the client, they made the ultimate decision (Successful Outcome).

For those working with large groups within a project, one tactic that was particularly helpful was breaking a large group into smaller groups to allow individuals to speak about their needs and then report the findings back to the larger group. Users also wanted to feel as if they were part of the effort, so by openly giving credit to users for ideas also helped make them an active participant in the project.

Another tactic was the importance of building positive momentum toward the project, and continuing it throughout the development phase by keeping the users involved. As implementation progressed, the importance of training, help desks, and other support functions became important to ensure that users were comfortable and felt involved as their work process changed. Ongoing communication throughout the project with the users was paramount. It was important that throughout the project, the users were kept informed of the progress of the project and what it was they were really getting.

Table 2
A framework for managing user expectations

<table>
<thead>
<tr>
<th>General strategy</th>
<th>Successful tactics</th>
<th>Less successful tactics</th>
</tr>
</thead>
<tbody>
<tr>
<td>User involvement</td>
<td>Listening to users</td>
<td>Not communicating with users on the state of the project</td>
</tr>
<tr>
<td></td>
<td>Asking questions</td>
<td>Planning to “outlast” a difficult user rather than working with them</td>
</tr>
<tr>
<td></td>
<td>Understanding users’ concerns regarding change and help them to feel at ease</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Working with users (not at them or to them)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Letting users make tough choices about budget, schedule, and/or functionality</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creating small groups for large projects to allow all to be heard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Giving credit to specific users for ideas within the group throughout the project</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Keeping users involved and updated throughout the project</td>
<td></td>
</tr>
<tr>
<td>Leadership</td>
<td>Ensuring a strong project champion is sharing the vision</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Articulating a clear vision of the project</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Motivating the project team to get the project done</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Educating users on the value and benefits of the system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Obtaining buy-in from the primary (and/or most vocal) stakeholders and work outward</td>
<td></td>
</tr>
<tr>
<td>Trust</td>
<td>Sharing good news and bad news throughout the project</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Providing specific times for deliverables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failing to realize the importance of a strong project sponsor or vision for the project</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failing to explain the purpose/value of the system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Failing to control scope creep</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Following others if there is suspicion that their actions are misguided</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hiding the true state of the project from the users</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Fake it until you make it” (i.e., failing to disclose a lack of knowledge in an area)</td>
<td></td>
</tr>
</tbody>
</table>
The hurdle, as in all software development projects, is getting users involved in the various stages of the lifecycle of that project. First is ensuring that you correctly documented their requirements. Once you’ve completed the design, you go back to them and say, “This is what we discussed, this is what it looks like, this is what it feels like, is that an appropriate solution for you?” Once you have that buy-in, then you complete development. You go back to the users again before system testing and show them the solution: what it is going to do and what the value is to them. You get the users’ buy-in once again. Then, you go into user acceptance testing. Essentially you’ve brought them through the whole system development life cycle with you (Andrew – Successful Outcome).

In the situations where there was a failure in managing user expectations, there was a lack of user involvement. This occurred because the project manager adopted one of the less successful tactics. In one situation, the users did not receive regular updates on the software, the features, and the implementation plan. The resulting software had too many features that users did not understand and was a radical departure from their current process. Not surprisingly, the users were less than satisfied. Upon reflecting on this project, the software project manager commented:

A lot of times, you say here’s our prototype, and in six months we’ll give you this product. Well, the reality is the prototype is about forty percent right. You should walk them through the change every time or on a regular basis. For a six month development process, every two weeks you say ‘Here is where we are and what it looks like. By the way, here are some notes about the changes taking place.’ Everyone is on the same page and the client does not feel like they are getting something different than what they expected. (Ryan – Not-as-Successful Outcome)

Ryan realized that not communicating with the users and involving them throughout the project’s lifecycle was a key reason for the project’s demise.

Another example of a situation in which user involvement was lacking occurred when Joshua encountered a particular user that only wanted things done his way.

We attempted to take a passive-aggressive role. We thought we would outlast this person. (Joshua – Not-as-Successful Outcome)

This project then suffered from many missteps. The project dragged on months longer than necessary and the relationship between the user and project manager suffered. Upon reflecting on this situation, Joshua realized he needed to be firmer with the user and explain the ramifications of the user’s choices. Had he educated the user on his options and the consequences of each choice and let the user decide the best option, Joshua may have had a better chance of successfully meeting this user’s expectations.

3.2.2. Leadership

There are two types of leadership that need to be exhibited during a software project to properly manage user expectations: a project champion for the users and a project manager/leader for the team. The project champion helps to manage expectations by promoting the vision for the project, educating users on the value and benefits of the software, by rallying the “troops” around a common purpose, and explaining to the users how they can assist in the effort. Many project managers realized the need for a representative from the user group that understood the value of the software and could educate other users. They needed someone that understood the benefits of the system and could focus the users on the positive changes that would be forthcoming. There was a need for at least one person within the user group that had a clear vision of the software to articulate the requirements and priorities of the system.

The software project managers offered several “lessons from the trenches” regarding what makes a good project champion. One project manager considered the type of champion she would want on a project.

You need a person who is well-respected within the organization. Someone influential. Someone with connections. Someone with really good people skills who can help people understand that we are not threatening them. (Hannah – Not-as-Successful Outcome)

Hannah, as well as other software project managers, realized the importance of having a champion to put users at ease during a time of change. Several informants also discussed the importance of having a project champion that can encourage the users to put in the time required to develop requirements and review deliverables.

Other ideal qualities in a project champion included strong, clear goals for the project. One project described by Ryan appeared to be “set up for failure” due to aggressive timelines, inexperienced developers on the project team, and high visibility in the organization. However, the project champions had very clear goals for the project and prioritized their requests to allow the project team to focus on the most critical functionality. This strong leadership from the user group was certainly helpful in ensuring this high-risk project could be successful.

When strong presence of a project champion was absent, problems tended to occur. Hannah recalled a situation in which the project sponsor asked her to run meetings with the users; however, the users did not respect “outsiders.” Upon reflection, she stated:

Maybe I could have been more assertive about her leading the meetings and her needing to be a leader. (Hannah – Not-as-Successful Outcome)

This project was eventually tabled because it did not move forward. Hannah realized that she may have been able to make a difference on this project had she encouraged the project sponsor to be a leader among the user community.
Users also need to understand the value of and purpose of the system that is being developed. After a project failed, Andrew realized the importance of users understanding the value of the system. He realized that the project sponsor should share the value of the software with the users or Andrew could have taken this leadership role himself. Andrew realized that it is critical for a leader to tell the users:

‘This is the true value of the project. If we don’t do this, we will not perform. We will not meet our goals. We need to get behind this.’ (Andrew – Not-as-Successful Outcome)

He realized that:

Ultimately it’s the sponsorship. If the sponsorship believes in the value message, then the users follow through.

The second type of leadership critical to managing user expectations is a strong software project manager. The software project manager not only is responsible for making sure his/her team is getting work done effectively, but also for leading the users along the right path. Matthew entered a project that was under-scope and with only one resource on the project, an inexperienced developer. Sheer willpower was the reason for success for this project. Matthew taught the developer at night so she could work on the project during the day. He motivated his team by not asking his team member to do more than he was willing to do.

As the project manager, it is important to put oneself in the “user’s shoes.” This provides credibility when tough decisions have to be made about the project. Joshua found that by educating the primary stakeholder within the user group and gaining her buy-in on the project, he was then able to move outward to other stakeholders to get their buy-in on the project and their assistance throughout the requirements analysis phase of the project.

Project managers that used one or more of the less successful tactics were unable to achieve the strategy of Leadership. For example, Ethan’s failure to control scope creep led to an unsuccessful effort at managing user expectations.

We were committed to deliver a system that did A, B, and C. Over the lifecycle, they decided to modify it to be D, E, and F. Because good documentation and process wasn’t followed, at the end of the day, everything was saying that the system should be A, B, and C. We weren’t able to achieve D, E, and F. So there was a lot of confusion as to the overall process and what should be delivered. (Not-as-Successful Outcome)

Strong management controls to manage the scope of the project is crucial. A strong leader must be willing to tell the users something they do not want to hear.

Another less successful tactic is failing to trust one’s own judgment and following someone’s advice or guidelines when there is evidence they are incorrect. Matthew was pulled into a project that was in a state of disarray. The senior manager wanted to be the sole source of contact with the users and did not want Matthew to speak directly with the users. Matthew knew something was wrong with this situation, but he trusted that the senior manager was “doing the right thing.” This created a multitude of problems on the project and led to dissatisfied users. Matthew realized that he could have done something in this situation. He could have spoken with the senior manager’s supervisors to resolve the problem.

3.2.3. Trust

As in any situation where personal relationships are important, maintaining trust is a large component of managing user expectations.

If your client trusts you, they know you’re not trying to get something over on them. I believe that me and the client have to be friends. It’s always give and take. The client’s going to give, you’re going to give, you’re going to take, and he’s going to take. If you’re not in business with your client, then you’re going to be up the creek without a paddle. (Matthew)

At times, a software project manager may walk into a less than desirable situation. One project manager recounted a situation in which the users selected a technology prior to specifying the requirements. This meant that the software project manager had to force a solution due to a mismatch between the technology and requirements. The project manager shared the problems with the users. Ultimately, the users were happy with the outcome after a lot of hard work. While some software project managers may feel the need to just make the solution work and hide these problems from the users, this project manager offered a different approach.

As far as managing the client, the approach we took was just being brutally honest with them and, typically that worked pretty well. (Joseph –Successful Outcome)

Users should hear about any problems on the project from the software project manager rather than someone else. Given the skeptical nature that some users have about software projects or any type of project that creates a significant impact on their work, communication with the users about the good and bad aspects of the project is important. One software project manager offered the following advice:

Make sure your users hear issues from you first before they hear it elsewhere. The key is communication. You can create the worst system in the world, but if the user is happy with it, then you’ve got a chance of salvaging something. (Joshua –Successful Outcome)

Even simple things, such as informing the users that they will see a prototype “in one week” rather than “soon,” was a tactic one project manager advocated to establish trust
Projects did go awry when the software project manager neglected to inform the client of problems on the project. Christopher joined a project that was already underscoped. The users kept requesting more functionality for a project with too short of a timeline and too small of a budget. He was unwilling to talk to the client about the situation and believed that he should hide the problems from the user. Ultimately the project was cancelled due to other problems at the firm; however, this unwillingness to disclose information about the project weakened the trust and relationship between this project manager and the users.

Jacob recalled one of his first projects where he had little understanding of the business domain and the technical aspects of the system. He did not feel comfortable asking for help and did not want to be perceived as incompetent. He remembered thinking that “confidence is competence” and that at the end of the day he would simply “fake it until you make it.” Finally, a more senior manager stepped in and realized how much he was struggling with the project and brought in a third-party to help with the situation. By this point, the software required re-work, costing both time and money. Once the user realizes that the project manager has been “faking it” in terms of his/her knowledge, it can be nearly impossible to rebuild that trust.

4. Discussion

This research addresses one of the most common risks faced by software project managers, managing user expectations. Software project managers can actively manage and affect user expectations using the three strategies for successfully managing user expectations identified in this research: User Involvement, Leadership, and Trust. This research is consistent with other studies on the topic of user expectations and user participation in software development. Using the theoretical base of Participative Decision Making, Ives and Olson [30] suggested that user participation can improve both system quality and user acceptance. The results of this research were consistent with this finding from earlier research given the comments made by project managers regarding the need to make the users feel a part of the process. The entire stream of user participation and user involvement literature has shown the importance of allowing the users to be a part of the software development process. This work provides some specific strategies that can be followed (or avoided) by a project manager to achieve this goal.

Several of the scenarios described by the informants reinforced the importance of having a strong project sponsor. The literature has already stated the importance of having a project sponsor or champion to keep the group focused on the vision of the project [31]. This work confirms that research. For example, in Beath’s [32] research on supporting the IT champion, she noted that project champions (or sponsors) value information and political support from the IS department. Beath [32] found that one of the best method for the IS department to support IT champions was to provide estimates for project costs and time to evaluate technology or persuade others. This finding was confirmed in the tactic “Letting users make tough choices about budget, schedule, and/or functionality” in that the users were given the information to evaluate their alternatives rather than having a decision mandated by the project manager. Beath’s work also found that sponsors highly value the support and leadership help provided by the IS department [32]. This research supports the need for a close relationship between the project sponsor and project manager to best influence and impact user expectations.

Other research has also discussed the importance of sharing both good and bad news on a project. Studies have tried to explain why auditors and members of a project team are sometimes less likely to share bad news (i.e., the mum effect) [33,34]. To maintain trust on the project, the project manager must be willing to share the good news and the bad news associated with the project to maintain trust with the users.

The service quality literature has also discussed the role of empathy, which is the attentiveness to an individual’s needs during a service encounter [23]. The tactic of “understanding users’ concerns regarding change and help them to feel at ease” clearly exemplifies this dimension of service quality. Empathy was found to be significantly related to user satisfaction in an e-commerce context [26], suggesting the importance of this dimension and tactic to help ensure downstream success of the project. Watson et al. [25] stated the importance of open and direct communication with clients for information systems departments wishing to improve service quality. This is consistent with the tactic of “Keeping users involved and updated throughout the project.”

Many of these specific tactics are supported directly from various literature streams associated with project management as documented above; however, several have not been suggested as specific methods to manage user expectations. Giving credit to specific users for ideas, obtaining buy-in from the primary stakeholders and working outward, or providing specific times for deliverables are tactics that do not appear to be specifically stated within the literature. These tactics are consistent with the literature on user involvement and participation, leadership, trust, and service quality.

4.1. Implications for practice

This research identified three broad strategies and many tactics to manage user expectations. This paper provides a mini playbook of tactics that can be used by software project managers to counter this very common and very real risk that occurs on software projects.
An additional implication for practice is that the user expectations of a project may have a stronger effect on the outcome than the actual quality or functionality of the software. It is easy in project management to focus on the project related success criteria of budgets, schedules, and quality [6]; however, several of the situations described by the informants had failures in one or more of the traditional project success criteria, but the users were still pleased with the outcome. Understanding the users and their needs may also be more important than other product-related success factors such as the number of features or system quality. Managing perceptions and ensuring the software meets these perceptions can be more important than perfecting the software itself.

Furthermore, managing user expectations by using the strategies and tactics employed in this paper, several other risks could be addressed (at least partially). Risks, such as failure to gain user commitment, lack of frozen requirements, or changing scope/objectives [1], are addressed to a certain extent by using the strategies outlined in this work. Using these strategies to address several risks at once could help a project manager to more successfully manage software projects.

4.2. Implications for research

There are limitations to this research. First, this study focused on a narrow group of informants, software project managers working for a large IT consulting organization. This means that the strategies and tactics may or may not apply to managing user expectations when the software is being developed internally within an organization. The literature on user involvement and participation does provide support that this strategy does hold in other contexts, but future research could explore this point. Second, the informants mainly discussed software projects for governmental organizations. Again, it is likely that these results are generalizable to other types of organizations, but additional research would need to evaluate these results. A third limitation is that this study did not examine the downstream effects of managing user expectations. There was some evidence that lack of managing user expectations did result in downstream effects such as the user satisfaction and realized benefits of the system; however, this was not explicitly measured or examined in this research. Another potential limitation is the experience level of the informants. Many of the informants had only three years of experience in a software project management role. Others had five to seven years of project management experience, and only one informant had extensive experience (i.e., 20 years of project management experience). Although many of the informants are still learning the art of project management, for those that are considered novices, each project may be more salient, possibly enriching their discussion of managing user expectations. Yet, it is possible that a more experienced sample of informants may suggest a different set of tactics. A final limitation is the small sample size used for this study. While there was some overlap in the tactics that were identified among the informants, theoretical saturation was not obtained. Therefore, this list of tactics and strategies should not be viewed as a comprehensive list, but as a starting point for additional research on the topic.

Future research should address these limitations and expand upon the results of this research. The IS literature as well as marketing theories illustrate the importance of managing user expectations to ensure future use and satisfaction with an information system. The IS literature has suggested some strategies to improve the management user expectations; however, to date, this research has revolved around the idea of user involvement and participation. This research introduces two other facets of managing user expectations, leadership and trust.

The results of this research coupled with the literature associated with expectation-confirmation theory, service quality, and user participation suggests that a research model such as the one depicted in Fig. 2 may be appropriate.

This model suggests that there are three components to managing user expectations: the actions of the project manager, the influence of these actions on the user’s perceptions, and the impact of these perceptions on the net benefits of the software. The first component, the three strategies recommended in this paper (i.e., user involvement, leadership, and trust), are the actions that the project manager can take to influence the user’s expectations about the project and software. As the project manager engages in the three strategies to manage user expectations, several dimensions of service quality, such as responsiveness, assurance, and empathy, would be achieved. Training software project managers to use these strategies throughout the project, not as a short-term effort but as a long-term approach, should impact user expectations of the project [25].

The second component of the model is a series of variables and relationships associated with the user’s perceptions about the project. The managed expectations of the user is a function of both the actions of the project manager and the a priori expectations the user has about the project. These managed expectations impact the user’s perception of the perceived performance of the project and either confirms or disconfirms the user’s expectations of the project. This relationship between the managed expectations, the perceived performance of the project, and disconfirmation is consistent with expectation-confirmation theory [13]. The service quality literature also discusses the gap between expectations and performance [19], and the disconfirmation construct is a measure of this gap. Theory and research suggests that the level of disconfirmation of expectations impacts user satisfaction with the software [12,13,35].

The third component of the model considers the eventual benefits of the software for the individual or organization. It is known that user satisfaction affects the net benefits of the software [11,36,37]. These net benefits may be measured at the individual unit of analysis (such as improving job satis-
faction, decision-making ability, productivity) or at the organizational unit of analysis (such as increased profitability, market share, competitiveness) [8]. The choice of measure for net benefits would be chosen by the researcher based on the goals for a given system or study.

This model integrates concepts related to managing user expectations and illustrates the inter-relationships between the software project manager, the users, and the resulting success of the software. To study this model in the future, it would be important to gather data from these multiple stakeholders to yield more insight on this risk and the downstream impact of measuring user expectations.

5. Conclusion

Studies in the IS field have illustrated the importance of properly managing expectations and the role this has on user satisfaction and software use; however, the literature has failed to explain specifically how to manage user expectations. This work addresses this void within the literature. This study identifies three general strategies and several specific tactics that software project managers can employ to counter this important risk on software projects.

In the user expectations literature, much of the advice provided by researchers surrounded the idea of user involvement or participation. The data support this finding. This research does broaden the understanding of user involvement and participation by providing some specific tactics that can be used by software project managers to encourage user involvement. In addition to user involvement, this research found that leadership and trust are also critical components to managing user expectations. This research also provides a contribution to future research by providing a research model that can further explore the relationship between managing user expectations, user perceptions, and software success. The tactics identified by this research are not overly complex, yet if software project managers truly understood and followed this advice, managing user expectations may not be such a high ranking risk in IT projects [1,3]. Much of the traditional research on software project management neglects to ask project managers about their perspectives and ideas on how to manage projects [38]. Therefore, advice from those “in the trenches” that face the difficult task of managing user expectations on projects on a daily basis is offered via this research. Their lessons remind researchers and practitioners alike of the importance of the user involvement, leadership, and trust on software projects.

Acknowledgements

The author would like to thank the reviewers and participants of the Second International Research Workshop on Information Technology Project Management for their helpful comments about this research. Specifically, many thanks are extended to Blaize Reich for her valuable assistance in improving this manuscript for publication in this journal. Finally, the author wishes to thank Adriane Randolph for her assistance in research design, data collection and analysis. Without her assistance, this research would not have been possible.

Appendix A. Interview protocol

Can you think of an IT project where you overcame problems in managing the expectations of users where looking back via hindsight, you are quite proud of the results?

- Will you please share with us the project and the situation?
- What were the consequences of the decision in terms of the project, your organization, and to you personally?
- Have you found yourself trying to solve other problems with managing user expectations in the same manner as this project? Has the result been positive or negative? How so?
## Table B1
Data excerpt

<table>
<thead>
<tr>
<th>Informant</th>
<th>Situation</th>
<th>Solution</th>
<th>Result</th>
<th>Management insights</th>
<th>Tactics identified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacob</td>
<td>User group was highly resistant to change. Needed to gather information and trust from one particular person that was highly resistant to change</td>
<td>Listened to user, asked lot of questions. Persuaded that the more information the user could share with the project team, the more they could help him by making his job easier. Gave user credit along the way for successes</td>
<td>Obtained necessary information from the user and accurately captured the process and built system around process. Important information came from this troublesome user. Project was a small functionality increase; however, the approach to handling the user was a risk mitigation strategy to get his buy-in early on</td>
<td>Focused on being open to person and personality, not pre-judging. Same gender may have helped. Team lead (female) had problems with user (male) in the past</td>
<td>Listening to users Asking questions Giving credit to specific users for ideas within the group Educating users on the value and benefits of the system</td>
</tr>
<tr>
<td>Emily</td>
<td>Used JAD session to obtain requirements from five different groups with different needs and problems. Needed to figure out how to address all problems within a single system</td>
<td>Separated people by division and had mini-focus groups to hear needs individually. Let each person get their needs “off their chest” and then brought them together to determine the final solution</td>
<td>JAD session seemed successful</td>
<td>Make sure everyone is heard and has a voice in the process. Involve users from each segment of the business to get their input and ideas</td>
<td>Creating small groups for large projects to allow all to be heard</td>
</tr>
<tr>
<td>Joshua</td>
<td>User group with paper-based process with no experience with automating processes and reports. Didn’t really know what they wanted; didn’t know how to think about moving toward an automated system</td>
<td>Meetings to educate the users and prototyping. Educated primary stakeholder to gain her buy-in; extended the process outward to obtain more and more buy-in</td>
<td>Good end product that was informative and helped the users perform their jobs more effectively</td>
<td>Importance of educating the user and obtaining buy-in. Communicate often. Users should hear about problems from the project manager rather than someone else</td>
<td>Educating users on the value and benefits of the system Obtaining buy-in from the primary (and/or most vocal) stakeholders and work outward Sharing good news and bad news throughout the project</td>
</tr>
<tr>
<td>Christopher</td>
<td>Ongoing maintenance and work on a website with multiple deliverables</td>
<td>Defining timeline for the users. Did not just say “we’ll get it done soon” – provided a specific timeline for the work.</td>
<td>Developed maintenance schedule. Happy client and successful project</td>
<td>Define everything up front. Clear up confusion early on</td>
<td>Providing specific times for deliverables Keeping users involved and updated throughout the project</td>
</tr>
<tr>
<td>Ryan</td>
<td>Executives wanted a leading-edge product for internal use, but with potential resale value. Had an aggressive timeline with high stakes and junior resources on the project. Requirements remained consistent throughout project due to aggressive timeline</td>
<td>Knowing and understanding the users that ensured solid requirement gathering. Clear vision for the project with everyone in agreement</td>
<td>Successful software development. Quality of code may not have been the best, but the system did what was expected. Well-understood expectations given the constraints for the project Everyone willing to work hard to get the job done</td>
<td>Put self in the shoes of the user to make a usable system. Know your users</td>
<td>Understanding users’ concerns regarding change and help them feel at ease Ensuring a strong project champion is sharing the vision Articulating a clear vision of the project Letting users make tough choices about budget, schedule, and/or functionality Motivating the project team to get the project done</td>
</tr>
</tbody>
</table>
In the situation just mentioned, how did you attempt to solve the problem?

• When you were trying to determine a course of action, did you try to recall past experiences from projects?

Can you think of an IT project where you had difficulty managing the expectations of users where looking back via hindsight, you would have done things differently?

• Will you please share with us the project and the situation?
• What were the consequences of the decision in terms of the project, your organization, and to you personally?
• What would you have done differently?

In the situation just mentioned, how did you attempt to solve the problem?

• When you were trying to determine a course of action, did you try to recall past experiences from projects?

Are there any other things related to your experience in managing an IT project that you would care to share with us?

Appendix B. Data analysis

Table B1 is an excerpt from the partial ordered matrix display used for the successful outcomes in managing user expectations. An additional column was added to this table to illustrate how a specific tactic was identified from the data. Not all of the informants’ information are provided in this display, but selected mini-cases were chosen to illustrate how successful tactics were identified from the data.

References


