A Cross-Functional Systems Project in an IS Capstone Course

ABSTRACT

Information systems (IS) practitioners must regularly work cross-functionally with business users when implementing enterprise systems. However, most IS higher education is not truly cross-functional in nature with students typically relying on instructors or even themselves to represent user requirements. To address this gap, we describe an ambitious multi-course project that paired students from an operations management class as business users with students from an undergraduate IS capstone course as systems developers to build an enterprise resource planning (ERP) application. In doing so, we attempted to emulate the critical success factors typically encountered in realistic cross-functional systems implementation projects as identified in existing literature, including top management support, team interaction, communication, project management, and training. Structural modeling of student survey data reveals moderate realization of these success factors. As an added benefit, the project outcomes also provided important feedback for our entire IS program. We combine the results with post-project student debriefings to thoroughly highlight opportunities for replicating and improving the project.

A Cross-Functional Systems Project in an IS Capstone Course

INTRODUCTION

Systems implementation in industry requires a breadth of functional skills such as systems analysis, programming, and project management combined with qualitative skills such as teaming and communication (Stratman and Roth 2002; Mabert et al. 2003; Ngai et al. 2008). Many of the functional skills can be taught within an information systems (IS) degree program, culminating in a capstone course to integrate and advance the concepts (Gupta and Wachter 1998; Steiger 2009). Given that most IS course projects rely on homogeneous, IS-only teams however (Kruck and Teer 2009), the qualitative skills can be difficult to teach.

Given this challenge, we coordinated a multi-course systems development project to emulate the realistic challenges of working cross-functionally with business users. The project paired 56 “users” from an undergraduate operations management (OM) class, who defined the business requirements of the system, with 40 “developers” from an undergraduate IS capstone class, tasked with scoping project requirements and building a new solution. The subsequent primary objectives of the research include:

1. Emulate the critical success factors of a cross-functional IS project in an academic setting
2. Assess opportunities for improvement and replication at other institutions, including enhancing IS student cross-functional interaction

These first objective is addressed through analysis of IS student survey responses regarding systems implementation success factors established in existing literature. The results reveal moderate actualization of these success factors and subsequent depiction of a realistic IS implementation. However, the students still did not fully appreciate important cross-functional factors such as teaming and communication. So, we review student debriefings and our own reflections to thoroughly support the second objective of project improvement and replication.

LITERATURE

We focus the literature review on enterprise resource planning (ERP) systems due to the richness of the literature base and similarity to the application used in our project. An ERP system serves as the foundation for a business (Antonucci et al. 2004) by comprehensively integrating data flow and linking processes across departments. ERP thus requires cross-disciplinary thinking and integration of business processes (Pellerin and Hadaya 2008; Cronan et al. 2011; Rienzo and Han 2011). With such reach, ERP implementations are not only generally long and costly (Mabert et al. 2003) but also frequently unsuccessful (Kanaracus 2010; Momoh et al. 2010).

Table 1 summarizes the most frequently cited critical success factors for ERP implementation. First, *top management support* involves leadership communication, commitment, resource allocation, and conflict resolution (Stratman and Roth 2002; Finney and Corbett 2007; Ke and Wei 2008). Next, *team interaction* must be cross-functional across technical resources and business users (Finney and Corbett 2007; Rothenberger et al. 2010), thus requiring a broad set of interpersonal skills (Hignite et al. 2002; Boyle and Strong 2006). Relatedly, *communication* within the cross-functional team is also critical (Finney and Corbett 2007; Wickramasinghe and Gunawardena 2010) as is strong *project management* skills such as project planning, task assignments, and progress monitoring (Chen et al. 2009; Dey et al. 2010). *Training* refers to developing sufficient technical skills (Schmidt et al. 2001; Stratman and Roth 2002; Finney and Corbett 2007). Finally, *change management* entails overcoming internal resistance to adapt to the new system and associated processes (Stratman and Roth 2002; Finney and Corbett 2007).

<Insert Table 1 about here>

With the breadth and complexity of these success factors, some academics consider ERP to be too complex to learn by traditional lectures alone (Davis and Comeau 2004). Subsequently, researchers highlight the need for experiential learning (Chen et al. 2011), ideally across multiple courses (Swanson and Hepner 2011). One typical approach involves ERP simulations and games, which have been shown to enhance ERP knowledge retention (Leger 2006; Cronan et al. 2011; Seethamraju 2011). Additionally, educational offerings from ERP software vendors (Antonucci et al. 2004) allow students to obtain direct, hands-on ERP experience.

INNOVATION

The above experiential approaches improve IS student technical skills and even understanding of business processes. Still, typical IS projects in practice "are often staffed by interdisciplinary teams, not necessarily in the same location, working together to solve complex tasks" (Kruck and Teer 2009, p. 326). Yet, conventional IS courses only involve IS students without business user representation (Kruck and Teer 2009), thus not imparting the cross-functional, team-based success factors identified above. We argue that the traditional IS capstone class, while a valuable tool in IS education, requires further innovation to better represent the cross-functional skills required in practice, With this need, the cross-course basis of our project thus provides an important contribution to the IS pedagogy literature by extending the "threshold" (Meyer and

Land 2003) of the typical IS capstone class to promote student comprehension of critical IS skills at a higher level of complexity and integration (Land et al. 2005; Wright and Gilmore forthcoming).

IMPLEMENTATION

The "system" of study was eOps (Jacobs 2011), an existing web-based simulation that serves as a basic ERP application to highlight interactions between core business functions. In eOps, business users produce and sell computers, managing *purchasing* of components, *production planning*, and ultimately *sales* of finished units. eOps generates a performance rating derived from accounting output of profitability relative to utilized resources. Specifically, the goal is to achieve a high rating by optimizing purchase prices and sales revenue while maintaining high manufacturing plant utilization.

Despite the educational value of eOps as an operations instructional tool, the system is over 10 years old, and technical improvements were needed to improve stability and functionality. The project thus centered on building a new, improved eOps application. The undergraduate OM class of 56 students served as the business users with content expertise to set the business requirements. Likewise, the undergraduate information systems (IS) class of 40 students filled the developer role of building the system to these requirements.

Phase 1 of the project entailed individual users competitively playing the existing eOps to attain deep understanding of the tool as if it were utilized in their daily work responsibilities like a typical ERP system. The developers did not participate in this phase. Phase 1 grading was based solely on the users' final performance ratings relative to one another. We held a debrief session with the users shortly after Phase 1 to review lessons learned and reinforce understanding of eOps business requirements. As part of Phase 1, users were also asked to assess potential improvements to the eOps tool. Opportunities included correcting existing problems as well as extending eOps functionality to improve both ease-of-play and learning of operations concepts. Users were then grouped into teams of four to five to coordinate these improvements and ultimately retain responsibility for the "as is" and "to be" eOps business requirements.

Phase 2 then paired OM user and IS developer teams to re-build eOps to the "to be" user requirements following tools and processes that the IS students had learned throughout their academic program. Each combined team retained about eight students, typically including three developers. The Phase 2 instructions (Appendix) led the combined user-developer teams through the systems development lifecycle (Grenci and Hull 2004), including construction of critical documents (e.g., project planning, business and technical requirements, use case diagrams, site maps, etc.), construction, and testing. Phase 2 represented 55% of the IS developer student final grade though only 15% of the OM user student final grade. This realistically mimicked the relative workload of the developers (i.e., main responsibility to build the system) versus the users (i.e., main responsibility to continue their daily jobs while also supporting the system development as an ancillary responsibility).

As depicted in the Appendix, we structured stepwise systems development project deliverables (Chen et al. 2011). We provided feedback on these deliverables as the project progressed and scheduled several drop-in help sessions throughout the semester (beyond office hours) for additional help. This approach created a "pull" orientation to fill knowledge and skill gaps (McLaren et al. 2007), letting "students learn from their mistakes" while providing "good customer support" (Fedorowicz et al. 2004). Furthermore, we generally avoided specifying expectations of the final eOps system (Chakravorty and Franza 2005) as to not force a particular solution approach. This encouraged student critical thought and ownership in the learning process (Umble et al. 2008) as "active constructor, discoverer, and transformer of their own knowledge" (Fellers 1996, p. 45).

EFEECTIVENESS

Teams presented their final projects, including a system walk-through, to the instructors at the end of the semester. We also conducted qualitative project debriefings during these presentations and later reviewed team diaries. To promote knowledge sharing, teams presented overviews of their solutions and lessons learned from the project in a class meeting. Additionally, we administered a survey to the IS students to measure the effectiveness of the project in emulating the critical success factors of a cross-functional IS project (Objective 1). Based on the literature review, the research model (Figure 1, Table 2) hypothesizes that higher levels of top management support (H1), team interaction (H2), communication (H3), project management (H4), and training (H5) each increase IS project success. Change management was omitted since the project did not continue to a formal implementation stage.

<*Insert Figure 1 about here*>

<*Insert Table 2 about here*>

The survey instrument was adapted from the industry-focused ERP literature (Stratman and Roth 2002; Nah and Delgado 2006; Nah et al. 2007) to our pedagogical context (Table 3). For instance, *team interaction*, *communication*, and *project management* were applied directly as the same constructs from the practitioner literature. *Top management support* was modified as instructor support, and *training* incorporated skills imparted in prior coursework and access to supplemental resources. Added self-report items in the survey measured *project success* and student perceptions about project learning, difficulty, and realism.

<Insert Table 3 about here>

The survey was administered to the developers at the end of the project but before grading with a response rate of 90%. Table 3 displays the average responses for the survey items, including significance from the scale medians (4). Most success factor items (*top management support, team interaction, communication, project management*, and *training*) were significant in a positive direction (i.e., greater than 4). However, TR 1 (“had skills necessary to complete the project”) was not significant, indicating that the developers felt that they were not technically prepared for this project. The team debriefings highlighted specific concerns with programming and database skills, which will be discussed later. Still, respondents indicated a relatively high level of overall *project success* (S 1-3). Combined, Table 3 suggests initial support for the first research objective. The additional items addressing level of learning (A1), expansion of thinking (A2), difficulty (A3), and realism (A4) were also significantly positive.

Partial Least Squares (PLS) Analysis

To evaluate the first research objective in greater depth, the model in Figure 1 was tested using partial least squares structural equation modeling (PLS-SEM) (Wold 1975; Hulland 1999). In comparison to covariance-based structural equation modeling (Joreskog 1978, 1993), PLS-SEM is suitable for smaller sample sizes like that involved with our project (Hair et al. 2011). PLS-

SEM applies a two-stage approach, initially evaluating the measurement model and then the structural model (Hair et al. 2011). First, the *measurement model* reviews the reliability (Nunnally and Bernstein 1978) and validity (Fornell and Larcker 1981) of the survey items and associated constructs (Table 4). Both indicator reliabilities and construct composite reliabilities exceeded recommended 0.70 levels (Hulland 1999). Relative to convergent validity, the latent variables explained more than half of each indicator variance (Fornell and Larcker 1981), though the team interaction construct was marginal at .50.

<Insert Table 4 about here>

Examining discriminant validity, the average variance extracted for each construct exceeded all squared correlations with other constructs, and item loadings for each designated construct exceeded all loadings for other constructs (Hair et al. 2011). Multicollinearity, which causes estimating difficulties due to high variable correlation (Silvey 1969), was confirmed in SPSS with VIF and tolerance for independent variables within acceptable ranges of 5 or less and greater than .2 respectively (Menard 2002; Hair et al. 2011). Indicator significance was verified for all items at the .01 level.

The second PLS-SEM step involves assessment of the *structural model* and associated hypotheses. Table 5 and Figure 1 summarize the results. The R^2 model significance of 0.485 is considered moderate (Hair et al. 2011). The estimated path coefficients were tested for significance using the nonparametric bootstrapping process in PLS-SEM (Henseler et al. 2009). The paths for *top management support* and *project management* were significant, thus providing support for H1 and H4. However, the paths for *communication* (H3) and *training* (H5) were not significant. Moreover, the team interaction path (H2) was significant but in a negative direction. These results are discussed in greater detail below.

<Insert Table 5 about here>

As a final test, Q^2 was evaluated to assess the predictive relevance of the structural model. Q^2 is a cross-validated redundancy technique to appraise structural model robustness by iteratively removing parts of the data matrix and running the model (Hair et al. 2011). For the model, Q^2 scores for six of seven iterations acceptably exceed zero (ranging from -.0435 to .5470) and are considered moderate (Hair et al. 2011).

Discussion

The structural model results reveal moderate success in accomplishing the first research objective of emulating the critical success factors of a cross-functional IS project. On one hand, the students realized the importance of project management (H4) to project success, which follows training from prior IS coursework. Furthermore, students recognized top management (i.e., instructor) support (H1) as an important project enabler.

On the other hand, training (H5) was not significant, suggesting that the IS students felt that their skills were not sufficient to impact the success of the project. The instructors had originally determined that the students had appropriate technical training given previous coursework. Additionally, we not only carefully laid out deliverables with due dates (including sample documents) but also held many open help sessions beyond normal office hours (supporting H4). However, detailed review of team diaries as well as post-project debriefings revealed that many students were overwhelmed by the eOps technical requirements, particularly programming and database interactions. This aligns with the lack of significance of survey item TR 1 shown in Table 3. Developers consistently reported insufficient time to learn needed programming skills. Moreover, successful teams seemed to rely on one student who already retained superior programming skills from professional experience or personal interests.

Continuing with the structural model results, communication (H3) was not significant, and team interaction (H2) was actually significant in a negative direction. So, the IS students perceived that collaboration within the cross-functional team not only failed to support project success but actually worked against it in some respects. Again, the diaries and post-project debriefings revealed several explanations. First, the IS students questioned user abilities to effectively fulfill their role of producing functional business requirements. More prominently, the aforementioned technical challenges with the project caused many developers to abandon interaction with the users in order to concentrate on programming. In such cases, the developers viewed extended team interaction as delaying and weakening project success. This became evident during the final presentations wherein many users had not yet tested the developers' solutions and even had little knowledge of the developers' progress.

Despite the limited success in imparting communication and teaming, project debriefs and diaries revealed that most developers still seemed to grasp the high-level understanding that the project effectively mirrored the challenges of a realistic cross-functional IS implementation. In fact, they clearly recognized the importance of the project deliverables as well as interaction with the user team. Nevertheless, they felt that they had no choice but to revert to a "no time to follow the rules," "do whatever it takes" mindset to produce a solution, regardless if that solution did not reflect the overall "corporate" (i.e., user) objectives of improving eOps.

TRANSFERABILITY - PROJECT REPLICATION AND IMPROVEMENT

The moderate success from the outcomes points to the need for thorough assessment of opportunities for project improvement and replication (second research objective). We start with the technical difficulty of the project, revealed above as the most significant hindrance to promoting the cross-functional critical success factors. As former practitioners, the instructors can attest that the technical complexities of most system implementations are severely underestimated. The students also appreciated this aspect of the project as true-to-life. Despite this reality however, the eOps project was probably too technically complex given the limited timeframe of a 15 week semester, and it instigated abandonment of cross-functional teaming. A less-complicated "system" would have allowed the developers to better focus on refining skills relative to cross-functional interaction and communication. Still, a simple project may marginalize the importance and impact of the user role. As an alternate approach, conducting the project across two semesters would allow use of a realistic and complex system to emphasize both the cross-functional and technical skills needed in practice.

Student Mindsets

The technical challenges of the project also highlighted student autonomy and dependence as important considerations for project replication. Several developer teams seemed to lack creativity and assertiveness in attempting to overcome technical difficulties. For instance, while students generally agreed to a high-level of top management (instructor) support, few teams consistently took advantage of instructor availability and willingness to help. In the same vein, students were frequently tentative in exploring external help sources such as on-line tutorials and even other instructors. Relatedly, some students lapsed into a learned helplessness attitude, blaming the users, the instructors, and/or prior courses rather than accepting accountability for overcoming skill gaps. Likewise, some teams justified underperformance with one another (i.e., "we're not doing well but no one else seems to be either"), which in some cases perpetuated lack of progress and low performance.

To better guide student perceptions of performance expectations, instructors can better promote the accomplishments of higher performing teams as benchmarks for the entire class throughout the project. As another consideration, organizing specific external training sources may help alleviate student skill deficiencies but could also further foster student dependence. Ideally, we found it most effective to interact with individual teams and direct them to additional resources only when absolutely necessary. With this, we advocate structuring multiple required cross-functional team meetings with the instructors to help identify skill gaps as well as provide coaching and encouragement. We also urge an anonymous, mid-project survey as another line of communication.

Cross-Functional Team Interactions

Continuing with improvement and replication, we next discuss the challenges associated with the cross-functional student interactions. Most prominently, the developer and user sides of the teams each maintained a general self-centered view of the project, framing deliverables and workload primarily in their own terms. While students were considerate of one another, neither side seemed to understand or necessarily respect the other's workload, skill set, and time commitments. For instance, the developers became frustrated with delayed input from users, not appreciating the lower importance of the project for users compared to other course obligations like exams. On the other hand, the users did not fully grasp the technical difficulty of the project and were sometimes slow to complete deliverables. As another example, users assumed that the developers had significant business knowledge and tended to omit what they thought were seemingly obvious requirements (e.g., profit calculation). Conversely, developers grew frustrated with some reaching or cosmetic user requirements that, in some ways, originated from user lack of technical awareness.

These challenges were heightened by the on-line, digital nature of the project work. Most teams overcame scheduling conflicts through asynchronous interactions such as discussions boards and net meetings. As another enabler, most teams struggled for effective cross-functional leadership as many students were too timid and inexperienced to fulfill the leadership role required to bridge the developer and user sides. For instance, there was often a lack of clarity across the two sides as to which was better equipped to lead a particular deliverable, which occasionally caused neither side to take control.

Overall, we underestimated the ability of both sides to bridge team integration challenges. We thus recommend obliging the two sides to interact on a face-to-face basis frequently and early in the project, ideally in the presence of the instructors. Likewise, we advocate highly specific user or developer ownership of deliverables. Additionally, added cross-instructor class interactions (i.e., instructor visits to the other instructor's class) and advanced team-based learning pedagogical approaches such as those suggested by Reinig et al. (2011) could enhance cross-functional learning. As a final consideration, using an MBA-level class on the user side may support a more mature, experienced benchmark for the IS students to overcome cross-functional challenges.

Project Execution

We next highlight specific execution challenges for such a complex, cross-functional project. First, instructors must set detailed deliverable expectations and an associated timeline at the project start. Likewise, deadlines must be strictly enforced. The subsequent complication is the necessity of detailed and immediate instructor feedback (i.e., within one to two days). This prevents timeline slippage, enhances team accountability, and quickly identifies skill gaps. Providing sample deliverables for each step is important, though such examples may stifle creativity as student simply replicate the given format and level of detail. As another project execution consideration, instructors should set up a combined class in the school's course management system to support communication and team work. When possible, discussions and chats in the course management system could be loosely monitored to identify critical problems that students may not elevate to instructors.

Instructors also need to encourage active diary updates on both developer and user sides, including instructor review at relatively frequent periods. This again allows for active recognition and hurdling of skill set deficiencies, passive student mindsets, and team interaction concerns described previously. Related to documentation, the combined team sizes (averaging eight students) caused some inertia in developing initial deliverable drafts as students would wait for input from all team members. To overcome this, we recommend somewhat smaller teams where possible on the user side. Instructors should also coach teams to have individuals develop initial drafts well before the due date then request feedback from the team.

One particular, and in hindsight, evident problem with our project was the timing of the actual classes. The classes met on the same day, but the start times were drastically different.

This caused a face time problem between the developer and user sides. Moreover, the time difference also meant that one class drew traditional full time students (users) while the other primarily drew part-time, working students (developers). This difference further aggravated team face time difficulties. As a final project execution concern, we encouraged teams to self-manage under-contributing members but also set clear guidelines and meaningful consequences for loafing through post-project peer evaluations linked to individual final project grades (Jassawalla et al. 2008).

IS Program Learnings

Finally, we close with important project lessons that highlighted potential improvements within our entire IS degree program. For example, we found that the developers were not necessarily used to working with external, inflexible user requirements found in typical IS implementations. In previous coursework, the developers had often fulfilled the user role, so they could change project requirements at convenience. This finding highlighted consideration of redesigning coursework in the entire IS program to abide by firmer and more realistic requirements. Such a change would also foster student autonomy in overcoming technical skill gaps discussed above.

Next, several students noted that most if not all prior IS courses had major projects that were group-oriented. Although this mirrors industry and allows students to learn skills sets from one another, overreliance on others can also allow students to avoid learning some skills. In the case of our project, many students had apparently eluded programming in prior projects, which rendered them significantly less effective on a large scale project such as ours wherein multiple coders were needed. This problem was exacerbated by a subset of developer students who already retained a wealth of IS skills based on professional experience yet were seeking a formal degree to support career progression. So, assessment of learning effectiveness of prior coursework may have been skewed by students who were already capable in the taught skill sets. Relating these two ideas, inexperienced IS students may, in some cases, have been deferring difficult project tasks in prior coursework to others without achieving sufficient learning. Again, this is an important lesson-learned for our IS program that may have gone unnoticed without the difficulty and scale of our project.

CONCLUSION

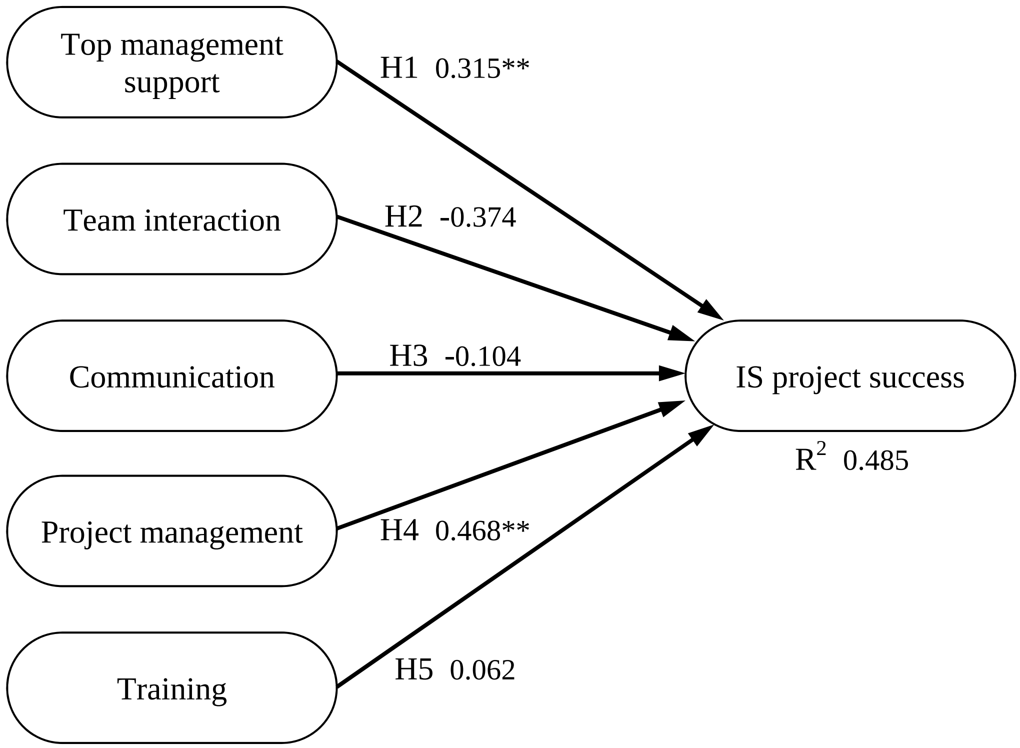
Typical IS academic projects do not simulate true cross-functional interactions that are experienced in realistic systems implementations. With this gap, this paper describes an ambitious multi-class IS project that sought to impart the challenges and critical success factors of a cross-functional IS implementation. The project provided students with a realistic replication of what they can expect as systems analysts in the professional world. In fact, we contend that the project was about as realistic as possible in a purely academic setting. As a secondary and unexpected benefit, the complexity of the project illuminated potential gaps in our IS program that would likely have gone unfound with traditional, easier projects.

The emergence of the capstone class was an important development in IS pedagogy, yet we assert that our cross-functional project approach represents the next generation of capstone course design. As insinuated above, the project generated an extremely heavy workload for the instructors, more so than we ever expected. Still, we recommend the project for dedicated educators who are looking to push pedagogical boundaries and enhance IS student learning.

Table 1: Summary of ERP success factor literature

Citation	Top Mgt Support	Team Interaction	Communication	Project Mgt	Training	Change Mgt	Other Example Critical Success Factors	Methods
Akkermans and van Helden (2002)	X	X	X	X			Integration, clear goals	Case
Al-Mudimigh et al. (2001)	X		X	X	X	X	Business case	Lit review
Dezdar (2011)	X		X		X			Survey
Dezdar and Sulaiman (2009)	X	X	X	X	X	X	Process re-engineering, minimal customization, vision, culture	Lit review (content analysis)
Finney and Corbett (2007)	X	X	X		X	X	Software, strategy, consultants	Lit review
Huang et al. (2004)	X	X	X		X		Processes, supplier support, data	Lit review, case
Li and Seddon (2009)	X	X	X	X	X	X	Process reengineering, ERP selection	Content analysis (user presentations)
Mabert et al. (2003)	X	X	X	X	X	X	Performance measures, minimal customization	Case studies, survey
Momoh et al. (2010)	X				X	X	Customization, business requirements, data quality	Lit review
Nah and Delgado (2006)	X	X	X	X		X	Business plan	Case study
Nah et al. (2007)	X	X	X	X			Culture	Survey
Ngai et al. (2008)	X	X	X	X	X	X	Software, strategy, processes	Lit review
Noudoostbeni et al. (2010)	X	X	X	X			Goals, integration, decisions	Case study
Plant and Willcocks (2007)	X	X	X	X	X	X	Goals, integration, expectations	Case study
Somers and Nelson (2001)	X	X	X	X	X	X	Integration, goals expectations	Meta-analysis
Stratman and Roth (2002)	X		X	X	X	X	Strategy, business process skills	Survey, interviews
Wickramasinghe and Gunawardena (2010)	X	X	X	X		X	Strategy, control	Survey, lit review

Figure1: Research model and hypotheses



Notes: ****p < .01,* ** *p < .05;* **p < .1 (one-tailed test)*

Table 2: Hypotheses

	Hypothesis
H1	Higher levels of **top management support** increase **IS project success**
H2	Higher levels of **team interaction** increase **IS project success**
H3	Higher levels of **communication** increase **IS project success**
H4	Higher levels of **project management** increase **IS project success**
H5	Higher levels of **training** increase **IS project success**

Table 3: Mean and significance of survey items

Construct		Survey Item	Mean (p-value)
Top Mgt Support[a]	TM 1	Instructors clearly defined Phase 2 project objectives	5.49 (.000)
	TM 2	Instructors committed to project	5.84 (.000)
Team Interaction[a]	TI 1	User and developer sides worked well together	4.92 (.001)
	TI 2	Had open dialogue with the users during project	5.14 (.000)
Communication[a]	C 1	Easy to communicate within entire project team	4.89 (.006)
	C 2	Members used right communication media (discussion boards, e-mail, face-to-face, etc.)	4.78 (.021)
Project Mgt[a]	PM 1	Followed documented project plan	5.68 (.000)
	PM 2	Specific project tasks clearly assigned to members	5.77 (.000)
Training[a]	TR 1	Had skills necessary to successfully complete project	4.57 (.107)
	TR 2	Could successfully obtain answers to technical questions (e.g., class, other classes, Internet, etc.)	4.86 (.007)
Project Success[a]	S 1	Our final eOps submission strong	5.19 (.000)
	S 2	Our final submission likely better than most others	5.08 (.001)
	S 3	Will receive a good grade on project	5.94 (.000)
Other Student Feedback[b]	A 1	Learned a lot from project	6.19 (.000)
	A 2	Project expanded thinking and skills	6.03 (.000)
	A 3	Project more difficult than projects in other classes	6.44 (.000)
	A 4	Project was realistic	5.17 (.001)

Notes: [a] *scales from 1 (strongly disagree) through 4 (neither agree nor disagree) to 7 (strongly agree)*
[b] *scales from 1 (highly uncertain) through 4 (somewhat certain) to 7 (highly certain)*
p-value represents significance from value of 4 (median)

Table 4: Measurement model results

Construct	Item	Factor Loadings	Composite Reliability	Ave Variance Extracted
Top Mgt Support	TM 1	0.964	0.901	0.820
	TM 2	0.844		
Team Interaction	TI 1	0.957	0.914	0.843
	TI 2	0.877		
Communication	C 1	0.938	0.907	0.830
	C 2	0.883		
Project Mgt	PM 1	0.952	0.927	0.865
	PM 2	0.908		
Training	TR 1	0.958	0.950	0.905
	TR 2	0.944		
Project Success	S 1	0.744	0.904	0.760
	S 2	0.957		
	S 3	0.900		

Table 5: Hypotheses results

	Hypothesis	Path Coefficient	Hypothesis Result
H1	Higher levels of **Top Management Support** increase **Project Success**	0.315**	Moderate support
H2	Higher levels of **Team Interaction** increase **Project Success**	-0.374	No support
H3	Higher levels of **Communication** increase **Project Success**	-0.104	No support
H4	Higher levels of **Project Management** increase **Project Success**	0.468**	Moderate support
H5	Higher levels of **Training** increase **Project Success**	0.062	No support

Notes: $^{***}p < .01$, $^{**}p < .05$; $^{*}p < .1$ *(one-tailed test)*

APPENDIX – Phase 2 Project Instructions (Developers)

Project Requirements

1. Use CSIS 3600/IS 2060/IS 3060; Systems Analysis & Design style
2. Use any software to simplify your work load: Visual Studio 2010, MS Word, Excel, Access, SQL Server, Visio, Project Manager 2010 or later. This is not an inclusive or exclusive list.

Due Date: December 1st 2011. Late projects will not be accepted.

Objective

eOps has served as a valued learning tool for students worldwide. However, improvements are needed to enhance the simulation. Student **users** (Mgt 3200) will help **developers** (IS 4880 student teams) build a **To Be eOps simulation** based on new technology**.**

Your new simulation will at least match As Is eOps functionality with new technology. Ideally, your new eOps will **significantly improve** functionality and usability. Improvements might be derived from your own eOps experiences, brainstorming, and course concepts.

As Is eOps Functionality

- Interactive sales, manufacturing, and procurement functionality
- Dynamic pricing for purchasing (parts) and sales (finished computers)
- Performance tracking (e.g., balance sheet, operating statement, inventory, and events)

Examples of Possible Improvements

- Advanced performance rewards/penalties (i.e., backorders, inventory, plant utilization, etc.)
- Advanced purchasing options, advanced sales options
- Multiple user options

Project Details

The heart of this course is a semester-long group project, in which each group will define the user requirements, document, design, and implement the eOps application. Each group will work with the users of the application to define the user requirements and functionality of the application. The users in this case are students registered in the operation management (MGT 3200) course. Since the SDLC process in this case will be highly interactive involving users and project managers, you may have to have to undergo much iteration of the various deliverables of the SDLC to create a successfully working prototype.

Project teams may consist of 3 developers and 3-5 users. In addition, there will be one team lead from the developers' side and one team lead from the users' side. The users and developers will work together to develop some of the deliverables. The developers team will primarily be formed based on the results on the first brainstorming exercise and in-class discussion. Ensure that at least one person on your team has solid programming skills. One member of the team should set up a team web page on which you will post the results of team assignments. Ensure the names and email addresses of the team members are at the top of the page and post it to a server. Organize the page so the instructors can quickly find your assignments.

A Cross-Functional Systems Project in an IS Capstone Course

Your project should be fully implementable by the end of the semester. Each group will present the product to the users and instructors. Groups will compete with one another.

Deliverables, Milestones, Diary

Users and developers team leads will be responsible for assembling the deliverables below. Additionally, team leads will each maintain a detailed **weekly diary** of project progress. Users and developers will get together for minimum 2 sets of **mandatory testing** of the application.

Deliverable	Description	Due Date	Users	Developers
Business Proposal **Business Case**	One page description of application functionality, technology to be used, and how application will be developed Cost-benefit analysis of alternatives (e.g., keep As Is, improve As Is, or develop from scratch).	9/15/11	X	X
Task Planning (Gantt Charts) **Task Planning (Risk Analysis)** **Project Charter**	MS Project schedule used to manage resources and track deliverables Table with possible project implementation risks, including a risk versus probability matrix. Contract between team members regarding the plan for the project.	9/22/11	X	X
Requirements - As Is System (Document)	Documents functional, database, back end, and front end requirements of the current (As Is) system.	10/6/11	X	X
Requirements - As Is System (UML)	Diagrammatic representation of the functionalities of the As Is system.	10/6/11		X
Requirements – To Be System (UML, E-R)	Written and diagrammatic representation of the functionalities and data expected in the To Be system.	10/6/11	X	X
Design (site map, story book)	Site map lists web pages of To Be system, including links to one other. Story board is a series of rough sketches describing each web page.	10/20/11		X
Testing	Users test application and offer feedback on fixes and improvements	11/3/11	X	X
Testing **User Documentation**	Additional user testing Step by step instructions on operating the application.	11/17/11	X	X
Finished Project, Documentation	All application development and documentation completed, submitted.	12/1/11	X	X

Grading

Each combined user/developer team will submit **one completed simulation with documentation** (describing use, functions, and help/FAQs) by **Dec 1, 2011**. Users and developers are **equally responsible** for the submitted simulation, which will be graded on functionality and quality. Grading is competitive and will be based on **functionality and quality**. **Creative, original** projects are preferred.

Your finished project should be fully implementable. Each team will present their completed project to the instructors and other teams. Late projects may not be accepted.

Individual student participation on the project will be evaluated by user and developer **peers**. The instructor may **significantly reduce the final project grade** for those receiving poor peer evaluations.

Description of Functionality, Quality	Grade
Simulation is fully functional and significantly improves upon the As Is eOps (i.e., "wow" factors). Documentation is thorough and professional.	A
Simulation is fully functional and offers moderate improvements over the As Is eOps. Documentation is mostly thorough and professional.	B
Simulation is mostly functional and/or essentially mirrors the As Is eOps. Documentation may be basic with some gaps and/or organization issues.	C
Simulation is not completely functional. Documentation may be minimal.	D
No submission or submission has significantly functionality issues.	F

REFERENCES

Akkermans, H. and van Helden, K. (2002). Vicious and Virtuous Cycles in ERP Implementation: A Case Study of Interrelations between Critical Success Factors. *European Journal of Information Systems*, *11*(1), 35-46.

Al-Mudimigh, A., Zairi, M. and Al-Mashari, M. (2001). ERP Software Implementation: An Integrative Framework. *European Journal of Information Systems*, *10*(4), 216-226.

Antonucci, Y. L., Corbitt, G., Stewart, G. and Harris, A. L. (2004). Enterprise Systems Education: Where Are We? Where Are We Going? *Journal of Information Systems Education*, *15*(3), 227-234.

Boyle, T. A. and Strong, S. E. (2006). Skill Requirements of ERP Graduates. *Journal of Information Systems Education*, *17*(4), 403-412.

Chakravorty, S. S. and Franza, R. M. (2005). Enhancing Cross-Functional Decision Making: A Simulation Approach. *Decision Sciences Journal of Innovative Education*, *3*(2), 331-337.

Chen, C. C., Law, C. and Yang, S. C. (2009). Managing ERP Implementation Failure: A Project Management Perspective. *IEEE Transactions on Engineering Management*, *56*(1), 157-170.

Chen, K., Razi, M. and Rienzo, T. (2011). Intrinsic Factors for Continued ERP Learning: A Precursor to Interdisciplinary ERP Curriculum Design. *Decision Sciences Journal of Innovative Education*, *9*(2), 149-176.

Cronan, T. P., Douglas, D. E., Alnuaimi, O. and Schmidt, P. J. (2011). Decision Making in an Integrated Business Process Context: Learning Using an ERP Simulation Game. *Decision Sciences Journal of Innovative Education*, *9*(2), 227-234.

Davis, C. H. and Comeau, J. (2004). Enterprise Integration in Business Education: Design and Outcomes of a Capstone ERP-Based Undergraduate E-Business Management Course. *Journal of Information Systems Education*, *15*(3), 287-299.

Dey, P. K., Clegg, B. T. and Bennett, D. J. (2010). Managing Enterprise Resource Planning Projects. *Business Process Management Journal*, *16*(2), 282-296.

Dezdar, S. (2011). The Influence of Organizational Factors on Successful ERP Implementation. *Management Decision*, *49*(6), 911-926.

Dezdar, S. and Sulaiman, A. (2009). Successful Enterprise Resource Planning Implementation: Taxonomy of Critical Factors. *Industrial Management + Data Systems*, *109*(8), 1037-1052.

Fedorowicz, J., Gelinas, U. J., Usoff, C. and Hachey, G. (2004). Twelve Tips for Successfully Integrating Enterprise Systems across the Curriculum. *Journal of Information Systems Education*, *15*(3), 235-244.

Fellers, J. W. (1996). People Skills: Using the Cooperative Learning Model to Teach Students "People Skills." *Interfaces*, *25*(2), 42-49.

Finney, S. and Corbett, M. (2007). ERP Implementation: A Compilation and Analysis of Critical Success Factors. *Business Process Management Journal*, *13*(3), 329-347.

Fornell, C. and Larcker, D. F. (1981). Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. *Journal of Marketing Research*, *18*(1), 39-50.

Grenci, R. T. and Hull, B. Z. (2004). New Dog, Old Tricks: ERP and the Systems Development Life Cycle. *Journal of Information Systems Education*, *15*(3), 277-382.

Gupta, J. N. D. and Wachter, R. M. (1998). A Capstone Course in the Information Systems Curriculum. *International Journal of Information Management*, *18*(6), 427-441.

Hair, J. F., Ringle, C. M. and Sarstedt, M. (2011). PLS-SEM: Indeed a Silver Bullet. *Journal of Marketing Theory and Practice*, *19*(2), 139-151.

Henseler, J., Ringle, C. M. and Sinkovics, R. R. (2009). The Use of Partial Least Squares Path Modeling in International Marketing. in R. R. Sinkovics and P. N. Ghauri (Eds.), *New Challenges to International Marketing (Advances in International Marketing, Volume 20)*: Emerald Group Publishing Limited.

Hignite, M. A., Margavio, T. M. and Chin, J. M. (2002). Assessing the Conflict Resolution Profiles of Emerging Information Systems Professionals. *Journal of Information Systems Education*, *13*(4), 315-324.

Huang, S.-M., Hung, Y.-C., Chen, H.-G. and Ku, C.-Y. (2004). Transplanting the Best Practice for Implementation of an ERP System: A Structured Inductive Study of an International Company. *The Journal of Computer Information Systems*, *44*(4), 101-110.

Hulland, J. (1999). Use of Partial Least Squares (PLS) in Strategic Management Research: A Review of Four Recent Studies. *Strategic Management Journal*, *20*(2), 195-204.

Jacobs, F. R. (2011). The eOps Game. Retrieved Dec 22, 2011 from http://bl-bus-jacobsvm.ads.iu.edu/ebus/Index.htm.

Jassawalla, A. R., Malshe, A. and Sashittal, H. (2008). Student Perceptions of Social Loafing in Undergraduate Business Classroom Teams. *Decision Sciences Journal of Innovative Education*, *6*(2), 403-426.

Joreskog, K. G. (1978). Structural Analysis of Covariance and Correlation Matrices. *Psychometrika*, *43*(4), 443-477.

Joreskog, K. G. (1993). Testing Structural Equation Models. in K. A. Bollen and J. S. Long (Eds.), *Testing Structural Equation Models*, Newbury, CA: Sage.

Kanaracus, C. (2010). Biggest ERP Failures of 2010. InfoWorld, Retrieved Dec 15, 2011 from www.infoworld.com/d/applications/biggest-erp-failures-2010-741.

Ke, W. and Wei, K. K. (2008). Organizational Culture and Leadership in ERP Implementation. *Decision Support Systems*, *45*(2), 208-218.

Kruck, S. E. and Teer, F. P. (2009). Interdisciplinary Student Teams Projects: A Case Study. *Journal of Information Systems Education*, *20*(3), 325-330.

Land, R., Cousin, G., Meyer, J. H. F. and Davies, P. (2005). Threshold Concepts and Troublesome Knowledge (3): Implications for Course Design and Evaluation. in C. Rust (Eds.), *Improving Student Learning Diversity and Inclusivity*, Oxford, England: Oxford Centre for Staff Learning and Development.

Leger, P.-M. (2006). Using a Simulation Game Approach to Teach Enterprise Resource Planning Concepts. *Journal of Information Systems Education*, *17*(4), 441-447.

Li, A. Z. and Seddon, P. B. (2009). Understanding How Project Critical Success Factors Affect Organizational Benefits from Enterprise Systems. *Business Process Management Journal*, *15*(5), 716-743.

Mabert, V. A., Soni, A. and Venkataramanan, M. A. (2003). Enterprise Resource Planning: Managing the Implementation Process. *European Journal of Operational Research*, *146*(2), 302-314.

McLaren, T. S., Vuong, D. C. H. and Grant, K. (2007). Do You Know What You Don't Know? Critical Reflection and Concept Mapping in an Information Systems Strategy Course. *Communications of the Association for Information Systems*, *20*, 892-908.

Menard, S. W. (2002). *Applied Logistic Regression Analysis, Volume 106*. Thousand Oaks, CA: Sage Publications.

Meyer, J. H. F. and Land, R. (2003). Threshold Concepts and Troublesome Knowledge: Linkages to Ways of Thinking and Practising within the Disciplines. ETL Project: Occasional Report No. 4, Retrieved Dec 19, 2011 from www.etl.tla.ed.ac.uk/docs/ETLreport4.pdf.

Momoh, A., Roy, R. and Shehab, E. (2010). Challenges in Enterprise Resource Planning Implementation: State-of-the-Art. *Business Process Management Journal*, *16*(4), 537-565.

Nah, F. F.-H. and Delgado, S. (2006). Critical Success Factors for Enterprise Resource Planning Implementation and Upgrade. *The Journal of Computer Information Systems*, *46*(5), 99-113.

Nah, F. F.-H., Islam, Z. and Tan, M. (2007). Empirical Assessment of Factors Influencing Success of Enterprise Resource Planning Implementations. *Journal of Database Management*, *18*(4), 26-50.

Ngai, E. W. T., Law, C. C. H. and Wat, F. K. T. (2008). Examining the Critical Success Factors in the Adoption of Enterprise Resource Planning. *Computers in Industry*, *59*(6), 548-564.

Noudoostbeni, A., Ismail, N. A., Jenatabadi, H. S. and Yasin, N. M. (2010). An Effective End-User Knowledge Concern Training Method in Enterprise Resource Planning (ERP) Based on Critical Factors (CFs) in Malaysian SMEs. *International Journal of Business and Management*, *5*(7), 63-76.

Nunnally, J. C. and Bernstein, I. H. (1978). *Psychometric Theory*. New York: McGraw-Hill.

Pellerin, R. and Hadaya, P. (2008). Proposing a New Framework and an Innovative Approach to Teaching Reengineering and ERP Implementation Concepts. *Journal of Information Systems Education*, *19*(1), 65-73.

Plant, R. and Willcocks, L. (2007). Critical Success Factors in International ERP Implementations: A Case Research Approach. *The Journal of Computer Information Systems*, *47*(3), 60-70.

Reinig, B. A., Horowitz, I. and Whittenburg, G. E. (2011). The Effect of Team-Based Learning on Student Attitudes and Satisfaction. *Decision Sciences Journal of Innovative Education*, *9*(1), 27-47.

Rienzo, T. and Han, B. (2011). Does ERP Hands-on Experience Help Students Learning Business Process Concepts? *Decision Sciences Journal of Innovative Education*, *9*(2), 177-207.

Rothenberger, M. A., Srite, M. and Jones-Graham, K. (2010). The Impact of Project Team Attributes on ERP System Implementations. *Information Technology & People*, *23*(1), 80-109.

Schmidt, R., Lyytinen, K., Keil, M. and Cule, P. (2001). Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems*, *17*(4), 5-36.

Seethamraju, R. (2011). Enhancing Student Learning of Enterprise Integration and Business Process Orientation through an ERP Business Simulation Game. *Journal of Information Systems Education*, *22*(1), 19-29.

Silvey, S. D. (1969). Multicollinearity and Imprecise Estimation. *Journal of Royal Statistical Society. Series B (Methodological), 31*(3), 539-552.

Somers, T. M. and Nelson, K. (2001). The Impact of Critical Success Factors across the Stages of Enterprise Resource Planning Implementations. Proceedings of the 34th Hawaii International Conference on System Sciences.

Steiger, D. M. (2009). Enhancing Knowledge Integration: An Information System Capstone Project. *Journal of Information Systems Education*, *20*(1), 17-23.

Stratman, J. K. and Roth, A. V. (2002). Enterprise Resource Planning (ERP) Competence Constructs: Two-Stage Multi-Item Scale Development and Validation. *Decision Sciences*, *33*(4), 601-628.

Swanson, Z. and Hepner, M. (2011). Knowledge Management ERP Curriculum Design/Mapping (Theory and Development Tools). *Decision Sciences Journal of Innovative Education*, *9*(2), 209-226.

Umble, E. J., Umble, M. and Artz, K. (2008). Enhancing Undergraduates' Capabilities through Team-Based Competitions: The Edward Jones Challenge. *Decision Sciences Journal of Innovative Education*, *6*(1), 1-27.

Wickramasinghe, V. and Gunawardena, V. (2010). Critical Elements That Discriminate between Successful and Unsuccessful ERP Implementations in Sri Lanka. *Journal of Enterprise Information Management*, *23*(4), 466-485.

Wold, H. (1975). Path Models with Latent Variables: The Nipals Approach. in H. M. Blalock Jr, A. Aganbegian, F. M. Borodkin, R. Boudon and V. Capecchi (Eds.), *Quantitative Sociology*, New York: Semina.

Wright, A. L. and Gilmore, A. (forthcoming). Threshold Concepts and Conceptions: Student Learning in Introductory Management Courses. *Journal of Management Education*, 1-22.