Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2009 Proceedings

Americas Conference on Information Systems (AMCIS)

2009

Integrating SDLC and ITSM to 'Servitize' Systems Development

Carol E. Pollard
Appalachian State University, pollardce@appstate.edu

Dhiraj Gupta
Progressive Insurance, dhirajgupta@windstream.net

John W. Satzinger
Missouri State University, johnsatzinger@missouristate.edu

Integrating SDLC and ITSM to 'Servitize' Systems Development

Carol E. Pollard
Appalachian State University
pollardce@appstate.edu

Dhiraj Gupta
Progressive Insurance
dhirajgupta@windstream.net

John W. Satzinger
Missouri State University
JohnSatzinger@MissouriState.edu

ABSTRACT

IT Service Management (ITSM) is generating much interest in industry as the quality and reliability of IT services are increasingly recognized as a critical factor for business success. Academic researchers have been slower to respond to industry demand for IT service management research and coursework. This paper argues that academia has an important role to play in integrating ITSM concepts and skills into traditional information systems coursework, specifically system development. The core systems analysis and design course is used to illustrate how IT service management concepts and models can be introduced into existing coursework to support the focus on IT services required by industry today, eventually reducing the growing percentage of IT budgets currently attributed to the operation and maintenance stages of 'non-servitized' systems development projects.

Keywords: ITSM; ITIL, IT service management, service science, system development, analysis and design, integration

INTRODUCTION

The business environment is changing at a dizzying pace, driven by factors such as globalization, business virtualization and outsourcing. To drive business agility, organizations need agile information technology resources including hardware, software, networks, processes and people. As a result, organizations are demanding far more from their Information Systems (IS) groups than in the past. In addition to 'better and more disciplined provisioning of IT services to ensure smooth operation' IT departments are expected to respond with greater agility in pursuing new business opportunities, to demonstrate responsible financial management and satisfy external customers through on-line systems (Johnson, Hately, Miller and Orr, 2007). They must also be able to swiftly adapt business processes to accommodate changing business conditions.

At the same time, as a means of providing more granular and flexible ways of designing, developing and deploying systems in an ever increasingly complex IT environment, systems development approaches have evolved from the traditional waterfall model to agile development and the emergence of service-oriented architecture (SOA). However, simply applying one of these new technical approaches to IT projects does not necessarily assure business flexibility since the focus of systems development remains on the project and not on the business.

For maximum business value, this paper proposes that IT service management and systems development should be integrated not only in practice, but also in the traditional academic systems development curriculum. We suggest that the systems analysis and design course would appear to be the logical place to focus on IT services because, on several levels, the finished "product" of system development is closely associated with IT services throughout the organization. Therefore, it is proposed that system development should be recast as service development. This paper first discusses the evolution of IT service management and system development methodologies, then explains the concept of SAD servitization, discusses its integration with an ITSM framework, such as ITIL, and concludes by proposing a possible way forward for introducing ITSM concepts into a university SAD course.

IT SERVICE MANAGEMENT (ITSM) AND THE IT INFRASTRUCTURE LIBRARY (ITIL)

IT service management (ITSM) is a strategy that addresses the need for IT to become more customer-focused by offering information systems under contract to customers and managing IT performance as a service. ITSM has grown out of the need to jointly address increasingly complex IT architectures and the call for more mature IT management. ITSM provides real benefits by helping IT organizations become more adaptive, flexible, cost effective, and service oriented. ITSM drives fundamental change within the IT organization, including how it manages its processes, technology assets, vendors and personnel, and how IT staff view their organizational roles. Galup, Quan, Dattero and Conger (2007) suggest that providers of IT services can no longer afford to focus primarily on technology but must shift their focus to the quality of the services they provide and foster an improved relationship with their customers.

To guide IT Service management, best practice frameworks such as the IT Infrastructure Library (ITIL) are emerging to support organizations and promote flexibility. There are many indicators of the growing awareness and adoption of ITIL worldwide. For example, there are now more than 100,000 members worldwide and 46 national chapters and numerous Local Interest Groups (LIGs) affiliated with the IT Service Management Forum. (itSMF). itSMF and vendor conferences, such IBM's semi-annual PULSE conference and Pink Elephant's annual International IT Service Management Conference and Exhibition, are enjoying record attendances and the demand for ITIL-qualified staff is increasing accompanied by an exponential rise in the number of IT Infrastructure Library (ITIL) Foundations certifications (Pollard and Cater-Steel, 2009). The ITIL framework is evolving as the most commonly used set of best practice guidelines for IT organizations considering, or actively adopting a service management strategy. ITIL focuses on a complex set of processes, functions and roles within five stages of the Service Lifecycle.

The ITIL stages are: service strategy, service design, service transition, service operations and continual service improvement. Early versions of ITIL focused on service delivery and service support. However, the latest version, ITILV3, released in July 2007, introduced a five-stage "service lifecycle". Recently, Enterprise Management Associates, a Denver, Colorado systems and network consulting firm reported results of a survey of 364 systems and network managers across the U.S. (Conger, Winniford, Erickson-Harris, 2008). Of these, 60% reported they were planning to use of were already using IT service management principles in their companies, and 80% of those using ITSM, 72% were familiar with or were already using the ITIL framework. Similar percentages were reported in a worldwide study of 488 IT executives conducted by IDC, a global provider of market intelligence, advisory services, and events for the IT, telecommunications, and consumer technology markets (Broussard, 2008).

Despite the usefulness of ITIL best practice guidelines prescribed for processes, functions and roles, most service management efforts have focused on the final stages of the service lifecycle to better manage IT services already in operation, such as service desk, incident management and problem management. Gupta (2009) refers to this as the concept of "build and then manage" – "one group builds IT services in isolation and throws them over the proverbial "wall" to another group who struggles to support them" (p1). We propose that to create really meaningful communication between the business and IT and achieve the flexibility needed, there must be consistent interactions from the beginning of systems development, when IT investments are first being discussed, through prioritization of projects, to design, development, deployment, delivery and support during operations. Without the proper processes in place, from beginning to end, to enable clear and consistent communication between IT and the business, initiatives are likely to continue to fail or at the very least reduce the delivery of IT value to the business.

EVOLUTION OF SYSTEM DEVELOPMENT METHODOLOGIES

Information systems development methodologies have continuously evolved in terms of structure and flexibility. Avison and Fitzgerald (2003) divide the relatively short history of development methodologies into four eras. The Pre-Methodology Era in the 1960s and 1970s included few formal techniques or models. The Early Methodology Era from the late 1970s to early 1980s included the waterfall SDLC, limited modeling techniques, and minimal focus on the business. The Methodology Era from the mid 1980s to late 1990s included structured techniques, information engineering, prototyping, early OO, participative development, and enterprise systems.

Despite many difficulties implementing and managing these methodologies, there was considerable focus on the business and strategic needs that information systems supported and on the application and technological architectures required to support the information systems.

The Post Methodology Era starting in the late 1990s involved a serious reappraisal of system development methodologies (Avison and Fitzgerald, 2003). Web development, component architectures, iterative development, and OO modeling expressed with the unified modeling language (UML) called for serious changes (Batra and Satzinger 2006; Satzinger, Batra and Topi, 2007). The Rational Unified Process (RUP) (Kruchten, 2004) became a potential standard that might have formalized many newer system development concepts. The RUP eliminated the lifecycle support stage, in effect, throwing support over the wall. At the same time, more radical, highly agile methodologies emerged that broke with the traditional information systems development focus on the organization and its business processes. Extreme Programming, Crystal, and Scrum (Larman, 2004) represent popular development methodologies with software engineers and increasingly with information systems developers.

The problem with the Post Methodology Era is the danger that newer highly agile methodologies focus more on software development than on organizational information systems. Additionally, highly agile methodologies, including agile approaches to the Rational Unified Process (Larman, 2004), provide minimal guidelines for the traditional support stage of the SDLC at a time when information systems are regarded as key IT services with extensive support requirements.

Recently, some researchers have begun to return to the view that information systems support a larger work system and must be designed in the context of available and appropriate user operations over time (Alter, 2006). Focusing on IT service management as a core discipline within system development supports this broader view work systems. Ultimately, viewing the information system within a work system as a service aligns information system development with IT service management. Actively adding IT service management concepts and models to the system development curriculum becomes desirable and even necessary.

TOWARD A 'SERVITIZED' SYSTEM DEVELOPMENT METHODOLOGY

The first requirement for a service management view of systems analysis and design is to propose the focus of systems development is the *service*, not the *product*. Then to extend this view, the focus is not on the system development project, but rather, on the broader aspect of developing and managing an IT service that adds value to the business. Instead of *developing an IT product*, the goal of analysis and design becomes *aligning the business, system development, and operations teams to develop solutions* that enable IT to link the infrastructure directly to the individual business processes that IT supports.

The result is IT helps manage the execution of the business. Therefore, service-based systems analysis and design focuses on providing services that directly and indirectly support specific business activities. All system development activities should be carried out in the context of providing IT services that are necessary to support end-to-end business services.

Table 1 illustrates the similarities and differences between the product focus of the SDLC and the service focus of IT service management. Contrasting the foci of these approaches suggests that combining the best of each of the two approaches will result in an integrated development methodology greater than the sum of the two separate approaches. Both systems and service development will be enhanced, with concepts of one approach complementing the other. Systems development will better serve the business needs through the addition of the service management concepts. Service management will be facilitated by the development of technology that enables end-to-end business services that add greater value to the business.

Most system development methodologies discussed in analysis and design courses use a variation of a traditional system development lifecycle (SDLC). Most courses discuss the need for an iterative and agile approach to the project and may include the use of a newer lifecycle such as the Rational Unified Process. Some now use the Unified Modeling Language (UML) for requirements modeling and design (see Batra and Satzinger, 2006; Satzinger, et al., 2007).

Despite a brief overview of broader information system issues, most courses focus on the "product" as the software. Additionally, most courses briefly discuss but do not design for or emphasize system support or the "service" aspect of systems development, which is where IT service management requirements come into play (Ward-Dutton, 2007).

Unlike systems development, where the focus is on the concepts of *project* or *technology* (product), ITSM focuses on the concepts of *service* and *process* in an effort to be customer-focused rather than IT-focused. This is where the differences in the two approaches are at their greatest and where the mindset or skills of the developer need to change. However, the new service lifecycle focus of ITILV3 suggests some evolving synergies between the systems development lifecycle stages and those of the new ITILV3 service lifecycle.

Product Focus (SDLC)	**Service Focus (ITSM)**
Planning	
Negotiate scope based on function	Negotiate scope based on end-to-end business process
Internally (IT) focused	Customer focused
IT jargon	Business jargon
Requirements Modeling	
"Over the wall" mindset	Stakeholder involvement
Technology insights	Business metrics
Automate function once and move on	Automate service once, reuse service in different ways
Focus on inputs and outputs	Focus on business needs and process
Design	
Capture logic of business function	Model business rules and external relationships
Focus on IT artifact	Focus on end-to-end business services
Construction	
Create a software product	Increase focus on value-added portions of applications
Buy, build or lease	Buy, build, lease and INTEGRATE
Deployment	
Technology driven	Minimum impact on business services
Test technology	Test service environment
Train on technology	Train in business service/process
Support	
Maintain hardware/software/networks	Continual service improvement

Table 1 –Product vs. Service Focus in SDLC

Table 2, shows how the stages of the ITILV3 service lifecycle loosely correspond to those of the systems development lifecycle. Those familiar with the two approaches will also realize the focus of the stages of the two approaches are quite different.

Life Cycle	**ITILV3 (service focus)**	**Systems Development (product focus)**
S	• Service Strategy	• Project Planning
T	• Service Design	• Requirements Modeling • Design • Implementation (Construction)
A	• Service Transition	• Conversion (Deployment)
G	• Service Operations	• Support
E	• Continual Service Improvement	

Table 2 – A comparison of the SDLC and ITIL

For example, ITILV3 takes a broad focus on the integration of IT with the business to maximize business-IT alignment and add business value. In contrast, project planning in the SDLC takes a much narrower 'project' or 'product' focus concerned primarily with technological efficiency and 'elegance'. Similarly, while requirements modeling, design and implementation center around IT architecture and software design, Service Design also provides guidance on the development of IT policies and documents for the design of innovative IT service solutions and processes. And, whereas conversion or deployment of a

system focuses on the operability of the technology, Service Transition focuses on the broader, long-term change management role and release practices to ensure minimal risks and maximum benefits of ongoing services.

Finally, support in the SDLC typically involves operating and maintaining the hardware and software, in contrast to Service Operations that explains and details delivery and control activities to achieve business excellence on a day-to-day basis and Continual Service Improvement that involves identifying and introducing service management improvements and issues surrounding service retirement. Understanding these differences highlights the opportunity to capitalize on the similarities and moderate the differences between the two approaches to emphasize an end-to-end business service focus from the very beginning of a systems development project in a servitized approach to systems development, while creating a high quality product.

SDLC – ITSM INTEGRATION

In the current IT environment where system development and ITSM are often managed separately, we propose the most effective way to bridge the gap is to focus on system requirements, breaking them down into functional and non-functional requirements. Traditional methodologies as well as RUP and newer agile methodologies recognize this distinction, often using the acronym FURPS, for functional, usability, reliability, performance, security, and supportability (Larman, 2004). The problem is the whole IT industry, including the education system, focuses on the functional requirements at the expense of the others.

A servitized system development methodology calls for a holistic approach to the design and construction of IT systems in order to ensure that the finished "product" of system development does not only satisfy the business functional requirements but also helps manage the execution of the business itself in an effective and efficient manner. In this holistic approach, service management activities, that have traditionally focused on the final stages of the service lifecycle, must be pulled forward and integrated with the SDLC in the form of well articulated non-functional requirements (the Reliability, Performance, Security and Supportability requirements). Similarly, system development, which has traditionally focused almost exclusively on functional requirements, must add specific service management deliverables based on the non-functional requirements into the appropriate stage of the SDLC as shown in Table 3. Business, System Development and Operations teams must work together to produce these deliverables.

SDLC Stage	Service Management Activities / Deliverables
Planning	• Business Impact Analysis
Requirements Modeling	• Service Level Requirements
Design	• Performance Modeling • Component Failure Impact Analysis (CFIA) • Security Mechanisms • Measurement Methods & Metrics
Construction	• Instrumentation • Service Monitors • Disaster Recovery Plan
Deployment	• Operational Testing • Capacity Plan • Operations Guide
Support	• Workflow support • Service Level Agreement/Operations Level Agreement • Service Support Training

Table 3 – SDLC – ITSM Integration through Service Management Deliverables

In the requirements modeling (analysis) stage most system development methodologies focus almost exclusively on functional requirements at the expense of non-functional requirements. Consequently, most SAD courses just pay lip service to non-functional requirements. This leads to IT systems that meet functional requirements but perform poorly – too many unplanned outages and poor transaction response times. In a servitized approach, equal attention must be devoted to operational and non-functional requirements. Performance, Reliability, Security and Supportability requirements must be thoroughly analyzed and clearly documented in the form of Service Level Requirements. A comprehensive Business Impact

analysis must be conducted during the requirement stage to identify critical business functions – business functions critical to the existence of the business itself. This analysis should then be used to derive continuity requirements for the critical business functions automated by the IT system being developed. We propose that SAD courses add the definition of Service Level Requirements and Business Impact Analysis techniques to the existing syllabus.

The design stage of the SDLC focuses on the overall architecture of the IT system and functional specifications for its various components. In order to ensure effective and efficient operations, the system design must meet all non-functional requirements as well. Therefore, it is not sufficient to just develop functional design and specifications. The system design must be assessed for adequate transaction response times at the design stage thru Performance Modeling. The reliability aspects of the system design must be validated at the design stage thru comprehensive Component Failure Impact Analysis (CFIA), more commonly known as Failure Modes and Effects Analysis (FMEA). Adequate redundancy and appropriate fault tolerance mechanisms must be designed to meet reliability and continuity requirements. Security mechanisms for authentication, authorization and confidentiality must be specified to protect sensitive business and customer information. Finally, appropriate measurement metrics and methods must be introduced into the system design to ensure that the execution of the business thru the IT system can be objectively assessed.

In order to prepare IT professionals for Service Development, the SAD courses must add discussion of Performance Modeling, CFIA (or FMEA), Security Mechanisms and Measurement Methods and Metrics to the existing curriculum. With regards to system performance, the course work should include identification of key performance scenarios, modeling techniques to assess the design for the performance scenarios, performance benchmarking techniques for critical design components and stress testing techniques for performance scenarios. With regards to system reliability, SAD students must be taught the CFIA (or FMEA) technique to analyze a particular hardware/software configuration to evaluate and predict the impact of failures and exposing Single Points of Failure (SPOF) in the design.

The construction stage must include deliverables to help manage the delivery of services once the IT system is in production. Appropriate measurement metrics and methods must be implemented thru instrumentation to gather and externalize performance, availability, security and exception data. Service monitors must be constructed to detect service degradation and take corrective action based on the data generated by instrumentation. Adequate backup and recovery mechanisms must be put in place and a comprehensive disaster recovery plan must be developed to meet continuity requirements. Consequently, SAD courses must include discussions and student exercises on measurement methods for IT systems, Statistical Process Control concepts, monitoring techniques such as Synthetic Transaction monitors and disaster recovery and continuous availability concepts such as backup - recovery techniques and geographically dispersed systems.

Traditionally the testing stage within SDLC is limited to functional validation. Accordingly, the SAD courses and most other literature on software testing focus mostly on functional testing tools and techniques. In order to ensure that the business function is not only performed as per the specifications but also performed well, it is proposed that the testing stage in SDLC be recast as Service Validation. The scope of testing discussion in SAD courses must be expanded to include Operational Testing – testing for performance, resilience, security and exception handling capabilities of the system. The system must be tested under projected production volume to validate the performance of the system and to collect data regarding the resource requirements of the system. A comprehensive Capacity Plan for IT resources must then be developed based on this data. The system must be subjected to component failures (based on the CFIA / FMEA developed in design stage), negative test conditions, unauthorized access and security attacks to validate its resiliency, security and exception handling.

Similarly, the scope of the deployment stage must be expanded to include transition activities to ensure seamless operation of the business function. Support Workflows must be developed to handle service requests from customers and exceptions that the system might encounter during operations. Operations Guide, detailing day-to-day operational processes for the IT system, must be developed. Operations staff must then be trained to support the execution of the business function. Comprehensive Service Level Agreements (SLA) and Operational Level Agreements must be put in place to set expectations for the delivery of services. SAD students must be familiarized with transition activities other than just the physical act of deploying the IT system and rolling it out to users.

SDLC - ITSM integration provides a holistic, systematic and proactive approach to Service Development. This approach promotes the early involvement of IT Operations in the design and development of IT services and allows them to influence the supportability and manageability aspects of the IT services. It eliminates confrontation and fosters collaboration between the two silos of IT - Application Development and IT Operation.

INTRODUCING SERVITIZATION INTO EXISTING COURSES

ITSM skills in university graduates are very much in demand but are still in short supply. To this end, companies such as IBM (IBM, 2009) and Hewlett-Packard (HP, 2003) have established academic initiatives to foster the ITSM offerings in the classroom. As a result, information systems educators, who have already acknowledged and started to address the need to study IT service management from a research perspective, are beginning to recognize the pressing need to incorporate ITSM into the IT and IS curricula (Cater-Steel and Toleman, 2007; Conger, 2009; Galup, et al., 2007; Rai and Sambamurthy, 2006). This has resulted in the development of some new ITSM courses and programs of study.

To support this increased academic interest in the IT service management discipline, itSMF USA established an annual academic forum in 2007 and the numbers of academics attending and presenting has increased steadily over the past three years. Despite these positive steps forward, currently little is being done to introduce service management concepts into existing courses to facilitate and hasten the transfer of ITSM skills into the marketplace.

From discussions at these meetings and from a review of current offerings at universities and community colleges, it is evident that currently IS curriculum is still paying little attention to ITSM. Even the proposed 2008 IS curriculum revisions makes no mention of service management (Topi, Valacich, Nunamaker and Sipior, 2007). In particular, there is a continuing strong focus on systems operation and little on system support when presenting the systems development lifecycle in IS education.

For example, the typical systems development course consists of methodologies and modeling techniques (UML, data flow diagrams, entity-relationship diagrams) to enable the planning, requirements definition, design, construction, deployment and support of a computerized information system. In delivering these skills, the majority of courses still maintain a technology or product focus rather than a service focus. To achieve a balance between functional (FURPS) requirements that seem to fall principally within the SDLC domain and non-functional requirements (URPS) emphasized in ITSM.

We propose that the SAD courses integrate ITSM with System Development by focusing the classroom discussion, activities and assignments on relevant ITSM topics and activities suggested in the sections above within the appropriate SDLC stage to create a balanced approach that combines the service focus with the product focus. For example, while teaching the Planning and Requirements Definition stage, focus must be shifted to "**how** a system will run" once "**what** is being automated" has been adequately discussed. Business Impact Analysis and Service Level Requirement definition activities must be introduced at this time.

Similarly, as part of the Design stage discussion, Performance and Capacity Modeling as well as CFIA (or FMEA) techniques must be introduced following the discussion of hardware, software and database architectures to demonstrate how infrastructure and application components inter-connect to influence the run-time characteristics of an IT system and ultimately impact the user experience, business processes and service quality. The discussion on testing should focus equally on functional acceptance and verification of performance, capacity and availability design specifications. Testing techniques to test the performance, resilience, security and exception handling capabilities of the system must be introduced as part of the testing discussion.

While teaching the Deployment stage, transition activities other than just the physical act of deploying the IT system and rolling it out to users must be adequately discussed. End user training and Help Desk training and startup activities must be discussed as a pre-requisite to actual rollout.

Taking this approach will move the focus away from a systems development course that develops an elegant hardware/software solution to one that emphasizes aligning business, development and operations teams to develop solutions that enable IT to link infrastructure directly to the individual business processes it supports. This approach will also help students move from a IT System paradigm to a IT Service model and provide them practical skills in Service Management. The result will be IT that helps manage the execution of the business.

A servitized systems development course will provide IT that supports services that directly and indirectly support particular business activities. All development and integration efforts will be carried out in the context of these services and activities primarily will be guided by the contributions that these services make to business activity. Table 4 summarizes a number of suggested changes in topic focus for incorporation into systems development offerings.

Functional SDLC Requirements	Non-Functional ITSM Requirements
Planning	
Tasks to be automated	Numbers – Users, locations, functional units, transactions, peak volumes, response time, hours of availability
Stakeholders	Organizational impact and criticality
Project team	Auditability
Project Management Office	Compliance
Requirements Modeling	
Details of **what** is being automated	Details of **how** it will run – *Service Level Requirements*
Fact Finding focused on technology	Fact Finding focused on business impact – *Business Impact Analysis*
Design	
SW Architecture	All numbers from planning refined
	Release planning
HW Architecture	HW Architecture inter-connections with respect to business services – *CFIA (or FMEA)*
DB Architecture	Number of Query types
Plans for testing	Model capacity and performance
Plans for cutover	Forward schedule of change
Construction	
Translate to HW/SW/DB environment	Query and DB Optimization
Test	'Ops Run' Books (*Operations Guide*)
DB Trans Tests	Change Control
Stress Test	Release Control
Deployment	
Acceptance Test	Testing AND Training
Live Test	Verification of Availability and Capacity – *Operational Testing*
	Auditability test
	Help Desk Training & Startup
	Signoffs
Continual Improvement	
	Measure business processes
	Measure IT services to increase efficiency, effectiveness and minimize costs
	Measure IT Architecture to increase efficiency, effectiveness and minimize cost
	Manage improvements

Table 4 – Servitizing the Systems Development Course

CONCLUSIONS

Combining systems development best practices with an effective IT service management framework such as ITILV3, will enable organizations to:

- Clarify business needs for the development team, to ensure fulfillment of agreed upon service level commitments by IT operations.

- Simplify and accelerate the process for 'going live' with new and updated IT systems and services.
- Reduce complexity and time-to-resolution of service-level breaches that occur in systems development

Central to well-governed IT organizations is alignment, control, accountability and vision. To achieve this level of maturity, greater integration must exist between systems development and IT operations. It is only through these combined efforts that IT can offer true innovation and differentiation. Servitized systems development that couples IT service management with business-focused development principles, is the pathway to this much sought after goal of achieving greater alignment between business and IT.

This is not a call to recast the systems development course into a service development course in one radical step. Such a change would be a daunting task, even for those familiar with service management concepts, and would also be controversial in the information systems academic community. Rather, this is a call for educators who are comfortable introducing the elementary concepts of service management into their current systems development course to begin doing so through a 'linking service with product' module based on the tables provided in this paper. It is also a call for textbook authors to integrate ITSM and service concepts into introductory and advanced analysis and design texts. Additionally, information systems curriculum developers and standards organizations should explore possible synergies by taking a service development view of what is currently known as system development.

We propose that the best way to create the necessary mindset and skills to achieve this goal is through the incorporation of best practices of servitized systems development into current systems analysis and design courses in universities and community colleges. This will pave the way for a new generation of systems developers to emerge, equipped to better understand business needs and integrate them with the goals of the IT department and act as a further step toward answering the elusive question of IT value and that IT does matter.

References - available on request.