A Methodology for Mapping System Engineering Challenges to Recommended Approaches

Lucas Layman, Forrest Shull
Fraunhofer Center for Experimental Software Engineering
College Park, MD USA
{llayman, fshull}@fc-md.umd.edu

Paul Componation, Sue O'Brien, Dawn Sabados
University of Alabama-Huntsville
Huntsville, AL USA
{paul.componation, obriens, dawn.sabados}@uah.edu

Anne Carrigy, Richard Turner
Stevens Institute of Technology
Hoboken, NJ USA
{acarrigy, rturner}@stevens.edu

Abstract— **Our current research is focused on identifying system engineering approaches that address four key development challenges in a tightly constrained, rapid reaction environment: 1) changing and emerging requirements; 2) conflicting stakeholder priorities; 3) concurrent sustainment and development activities; and 4) integration of independently evolving components. We are building a concept map of the key elements that form a strategic bridge between development challenges and the specific methods, processes, and tools that successfully address those challenges. In this paper, we present our methodology for constructing a robust mapping that incorporates interviews, surveys, and rigorous analysis methods. We summarize the results from interviews with sponsor personnel, the results of a best practices survey of 116 professionals, and qualitative analysis of the survey responses.**

Keywords-systems engineering; software engineering; methodology; survey; qualitative analysis

I. Introduction

The sponsor organization of this research, an organization within the U.S. Department of Defense, develops systems in a challenging environment. The sponsor's teams deliver products iteratively on a tight schedule, must achieve a high standard of security and performance, and often must integrate with teams from whom they are isolated. Handling emerging requirements, multiple stakeholders, and challenging integration issues are common for these teams and improvements to their current practices are needed to improve the overall development process. Recommended solutions for these challenges often come as anecdotal evidence or from individual experiences, but such solutions cannot always be leveraged into practical application. A research methodology is needed to *systematically investigate* development challenges and their potential solutions based on industry knowledge.

Our research goals are threefold. Our first goal is to identify current industry best practices for addressing the major challenges faced by the sponsor organization. Our second goal is to identify gaps in the current state of the practice for addressing these challenges. Our final goal is to create a conceptual framework to help engineers understand the characteristics and requirements that their Methods, Processes, and Tools (MPTs) must fulfill to provide a viable and sufficient solution to their challenges. This final goal will help organizations to both strategize solutions for challenges and to assess the sufficiency of their current practices.

To meet our research objectives, we derived a methodology composed of four parts: 1) *semi-structured interviews* to identify specific "challenge areas" faced by the development organization; 2) an *industry survey* to assess current best practices; 3) *in-depth qualitative analysis* of survey responses to identify general solutions; and 4) the *conceptual mapping* of specific practices to the challenge areas they help address.

II. Step One - Identifying the Problem

Interviews were conducted with personnel at the sponsor organization to elicit the engineering challenges they face.

A. Semi-structured Interviews and Method

Interviews are a common research method for building a qualitative data set from experiential knowledge. Semi-structured interviews are guided by an interview template with directed questions pertaining to a research goal, but with some open-ended questions to elicit thoughtful responses from the interviewees [7]. Semi-structured interviews are a cost-effective way of eliciting relevant information when not all important aspects of the domain are known, as was the case with the system engineering challenges in the sponsor's environment.

Twelve semi-structured interviews were completed with thirteen total participants in the sponsor organization from March through April 2009. The participants were lead systems engineers, program managers, or systems architects within their organizations. A script was provided to guide the discussion. Interviewees were asked several questions regarding their roles and responsibilities and the sources of problems in their products. Interviewees were also asked a series of questions on how they handle requirements, configuration management, technical reviews, documentation, testing and life cycle planning. Interview results were captured in MS Word and reviewed by two interviewers for accuracy. All information identifying the interviewees was removed from the transcripts. The interviews were examined by the research team to identify commonly occurring themes.

B. Results - Sponsor Challenge Areas

The personnel interviews yielded four key challenge areas facing the development teams. These challenge areas were reviewed with the sponsor for validation.

This research was sponsored by the U.S. Department of Defense's Systems Engineering Research Center.

978-1-4244-5883-7/10/$26.00 ©2010 IEEE

1) Emerging ***requirements*** *and/or changing priorities***.** Requirements change frequently and the teams often develop functional prototypes for delivery in a 90 day window. The teams typically create these systems "on demand" for users who need capability right away. Teams are evaluated based on whether or not their systems are deemed useful. Useful systems may be integrated into longer term, on-going projects. Projects that do not produce useful intermediate results will fail or be abandoned as these are overtaken by those demonstrating incremental value.

2) Obtaining useful ***stakeholder*** *input and dealing with conflicting priorities.* Stakeholders are a challenge in the environment because they have difficulty formulating good requirements up front. They are not always sure what they want, and it is only through the use of the technology that they are able to articulate their needs. Development teams often have to prioritize the requirements themselves because of short delivery cycles and lack of access to end users and other teams. Organizational or institutional issues often preclude direct engagement between customers and developers.

3) Resource conflicts between developing new capabilities and ***sustaining*** *currently deployed systems.* Systems are initially developed with local functionality, performance and security in mind. Capabilities that worked "well enough" for initial release may be elevated into a larger system with widespread deployment. The intial solutions must be re-engineered to meet the more rigorous non-functional requirements needed for integration into the larger system.

4) ***Integration*** *of independently evolving components into a larger interoperable system.* System integration is a challenge when teams are developing interoperable components concurrently. Teams receive requirements pertaining to interoperability with other systems being concurrently developed, yet cannot always communicate with other teams directly. Interfaces are not well-defined and are in flux due to the nature of rapid development and a lack of suitable interface management (including communication and enforcement) among the isolated teams.

III. Step Two – Surveying the State of the Practice

After identifying the key challenge areas, we conducted an online survey of commercial and government organizations. The purpose of the survey was to meet research objectives two and three: to identify current best practices for addressing the sponsor's challenges and to identify gaps (unanswered challenges) in systems engineering practice.

A. Survey Method and Demographics

A survey is a common research method that asks questions to produce data from a sample from a specific population that can be statistically analyzed [4]. Our survey included demographic questions to identify respondents with high security, performance and communication constraints whose suggestions may be a better match for the sponsor environment. Respondents were solicited by email invitation to industry contact lists of the researchers, as well as by postings to industrial group mailing lists such as the Los Angeles chapter of INCOSE. We make no claims regarding the completeness of our survey results; the aim of this research phase was to identify candidate practices with the potential to address the sponsor's challenges. The contents of the survey can be found in Appendix A of [2]. Summaries of the demographic results of the survey are shown in Tables I and II.

TABLE I. Primary Job Functions of Survey Respondents

Which of the following best describes your primary job function?	**%**	**N**
Functional Manager	22.4%	26
Project Manager	7.8%	9
Lead Systems Engineer	18.1%	21
Other	22.4%	26
Systems Engineer – Supporting Multiple Projects	19.0%	22
Systems Engineering – Supporting a Single Project	6.9%	8
Software Engineer	0.9%	1
Engineer	2.6%	3
Total responding to the question		**116**

TABLE II. Expertise of Survey Respondents

Please indicate your years of experience in each of the following:						
	0	***1-5***	***6-10***	***11-20***	***>20***	***N***
Sys. Engineering	0.9%	28.3%	23.9%	29.2%	17.7%	113
Soft. Engineering	15.3%	19.4%	14.3%	25.5%	25.5%	98
Other Eng.	14.5%	24.1%	22.9%	15.7%	22.9%	83
Total responding to the question						**116**

As of July 15, 2009, 116 respondents had participated in the survey. The respondents represented many job functions, encompassing both management and engineering roles across multiple industries. Many respondents had a significant degree of experience in systems and software engineering.

B. Results – Recommended Methods, Processes and Tools

As previously mentioned, one objective of the survey was to identify respondents whose environment closely matched the sponsor's. A similarity score was computed based on responses to six four-point Likert scale questions. The score is a sum where the responses of ["strongly agree", "agree", "disagree", "strongly disagree"] mapped to [1, 0.5, -0.5, -1]. Respondents were also asked to identify three or four of the ISO/IEC 12207 process areas they saw as most in need of new or revised MPTs. The respondents with excellent or good similarity were used to help meet the research objective of identifying gaps in current systems engineering MPTs in environments similar to that of the sponsor. The similarity score and gap results are summarized in Tables III and IV respectively.

TABLE III. Respondents' Similarity Scores to Sponsor

Similarity score	**Score range**	**%**	**N**
Excellent	[5-6]	11.0%	12
Good	[3-4.5]	34.0%	37
Weak	[0-2.5]	45.0%	49
None	[-6-0]	10.0%	11
Total responding to similarity questions			**109**

TABLE IV. CRITICAL AREAS NEEDING NEW OR BETTER METHODS, PROCESSES AND TOOLS IDENTIFIED BY EXCELLENT/GOOD MATCHES WITH THE SPONSOR (25% CUTOFF)

Critical areas identified as needing new or better methods, processes and tools	%	N
Decision Management	46.9%	23
Stakeholder Requirements Definition	40.8%	20
Measurement	28.6%	14
Architectural Design	28.6%	14
Integration	28.6%	14
Project Planning	26.5%	13
Project Assessment and Control	26.5%	13
Risk Management	26.5%	13
Total # of excellent/good commonality respondents		**49**

Decision Management and Stakeholder Requirements Definition were the primary gaps identified. Nearly two-thirds of the most frequently mentioned MPTs fall in the Stakeholder and Requirements challenge areas. Systems engineers are using MPTs to address these challenges, but the MPTs are not perceived as sufficient. Decision Management as a gap points to governance and oversight issues in the development and acquisition process..

Four questions on the survey asked respondents to describe how their organizations address the key challenge areas of changing requirements, multiple stakeholders, sustainment, and integration/interoperability. Many responses to these questions were non-specific (e.g. "do strong requirements analysis"), while others referenced a specific approach (e.g. "modeling requirements using SysML"). The five most-frequently-mentioned specific MPTs are listed in Table V.

TABLE V. TOP FIVE MOST-FREQUENTLY-MENTIONED MPTS

MPT	Total	Req.	Stake.	Sust.	Int.
Rapid prototyping [6]	14	3	8	0	3
Continuous integration [3]	11	0	0	2	9
Iterative/incremental dev. [5]	11	5	1	0	5
Interface control doc.[9]	8	1	1	0	6
Incremental commitment model [1]	7	6	1	0	0
Total MPTs		**49**	**41**	**29**	**36**

These results provided a list of candidate MPTs for addressing the sponsor's key challenge areas. However, many of these MPTs would be difficult for the sponsor to implement given their strong security and performance constraints. The challenge of adapting these MPTs to the sponsor environment and of creating new MPTs to address current gaps in systems engineering practice is the subject of ongoing work.

IV. STEP THREE - IDENTIFYING GENERAL SOLUTIONS

We examined the survey responses to identify *how* a recommended MPT solves a given problem to determine important underlying approaches. For example, a variety of requirements challenges may be addressed through effective communication techniques, leveraging expert personnel, or modeling. We call these general approaches MPT "themes." By understanding the themes underlying common MPTs, new MPTs can be created and current MPTs adapted to operate within the sponsor environment while still retaining the aspects that make them successful in addressing the challenge areas.

A. *Qualitative analysis of survey responses*

A multi-step process of qualitative analysis was performed to identify common themes of *how* the MPTs mentioned in the survey responses addressed the various challenges. Each step involved *open coding*: the process of identifying the categories in qualitative data and the properties of those categories [8]. First, the survey responses were examined to identify unique MPTs for each challenge area. In total, more than 200 unique MPTs were identified. Many MPTs (e.g. rapid prototyping) were suggested for multiple challenge areas.

Second, the unique MPTs were grouped into categories of MPT themes. The themes and unique MPTs grouped under them were reviewed and approved by a team of three researchers. More than 30 themes were identified, and some themes (e.g. direct stakeholder communication) appeared for multiple challenge areas. Table VI summarizes the number of MPTs and MPT themes identified for each challenge area.

TABLE VI. COUNTS OF MPTS AND MPT THEMES BY CHALLENGE AREA

Challenge area	MPTs identified	MPT themes
Requirements	89	17
Stakeholder issues	71	13
Sustainment	41	14
Integration	50	15

Finally, the survey responses were distributed among five researchers. Each researcher was given a description of the MPT themes and coded the survey responses for a challenge area using the MPT themes. This independent coding served to validate that the themes are robust and well-defined, and to independently assess the most common themes among the responses. Based on the researchers' applications of the MPTs to the survey responses, we computed the most common MPT themes among the responses as well as the level of agreement between the researchers.

The themes are a useful way of categorizing the survey responses, especially those that do not mention a specific practice. For example, while "meet the customer" is not a specific technique, it is indicative of a general approach. Clustering MPTs into themes allows further analysis to be done to identify common operational concepts.

B. *Themes for addressing the Requirements challenge area*

The coding results of MPT themes for the requirements challenge area are depicted in Fig. 1. The themes depicted in are those with the highest coding agreement between researchers in the survey responses. The horizontal axis depicts the number of respondents whose response was classified under that theme.

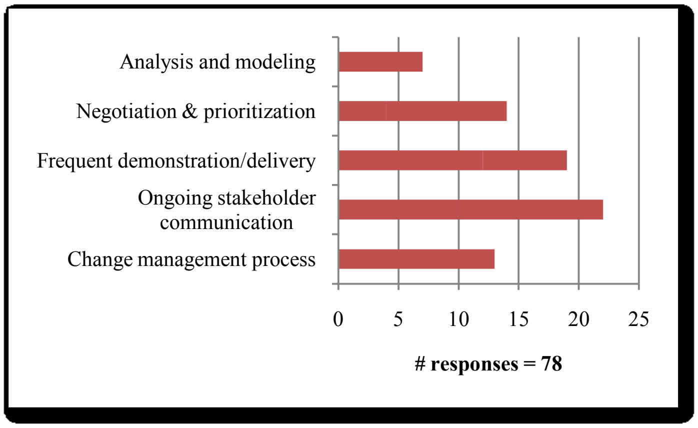

Figure 1. Most common themes for the Requirements challenge area

Analysis and modeling – Using tools, techniques and languages that can model systems and the relationships between systems. This category includes using UML to create model customer requirements and dependencies between subsystems. Often, this theme was specifically described as a way to understand the impact of changing requirements on downstream development.

Negotiation & prioritization process – Methods and techniques *specifically* for requirements priority negotiation among multiple stakeholders. This theme includes using techniques such as virtual forums and WinWin negotiation to solicit requirements and having a process for determining the business value of stakeholder requirements.

Frequent demonstration/delivery – Periodic demonstration or delivery of functionality (including prototypes) to stakeholders. Short delivery or demonstration cycles provide continuous validation of the system being built thereby reducing the risk that the system is not being built to satisfaction.

Ongoing stakeholder communication – Soliciting feedback from customers and/or stakeholders throughout the development process using either direct communication or collaborative tools. Direct communication ensures that requirements are prioritized with customer input, that ambiguous requirements can be resolved quickly, and that requirements definitions originate from actual users.

Change management process – Having a process that dictates how requirements are changed, which may include a change approval process, change tracking, traceability and tool support. Requirements change management processes promote responsibility for change and facilitate communication of changes throughout the organization.

C. *Themes for addressing the Stakeholder challenge area*

The coding results of MPT themes for the stakeholder challenge area are shown in Fig. 2. The "frequent demonstration/delivery" and "ongoing stakeholder communication" themes are the same as for the Requirements challenge area.

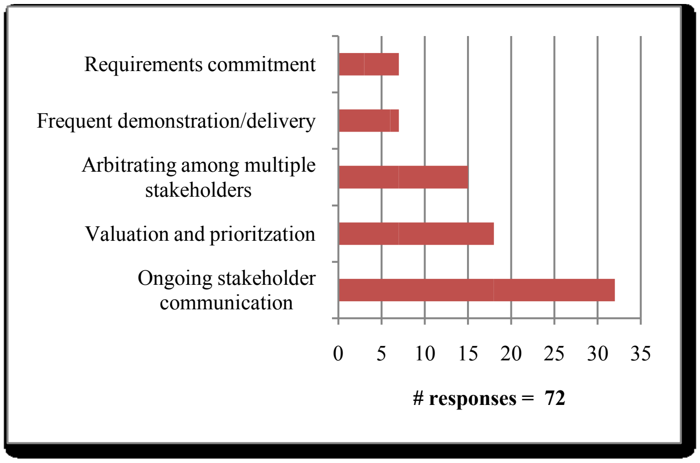

Figure 2. Most common themes for the Stakeholder challenge area

Requirements commitment – Committing to a set of requirements (requirements freeze) and creating a project plan and schedule. By committing to a set of requirements, teams do not become paralyzed by excessive iteration over requirements definitions or prioritization. Freezing requirements after an initial demonstration or establishing a baseline are mechanisms around which a more concrete development schedule can be created.

Arbitrating among multiple stakeholders – Involving multiple stakeholders in requirements prioritization and a defined process for deciding on requirements priority. This theme is similar to "negotiation and prioritization process" in the previous section, but with an emphasis on an arbitration strategy for deciding final priorities between conflicting stakeholder requirements. In addition to WinWin negotiations, QFD matrices and simply having an empowered decision maker were mentioned.

Valuation and prioritization – Processes and techniques specifically for assigning a business value to a requirement in relation to other requirements

D. *Themes for addressing the Sustainment challenge area*

The coding results of MPT themes for the stakeholder challenge area are shown in Fig. 3.

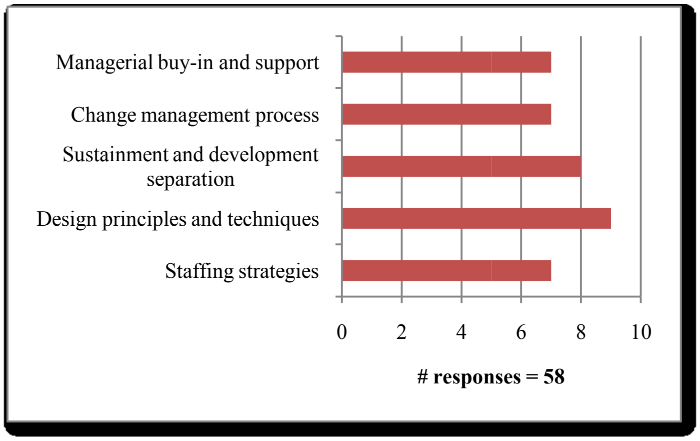

Figure 3. Most common themes for the Sustainment challenge area

Managerial buy-in and support – Managerial buy-in and support for development activities and decisions made during the process. This theme involves management understanding the costs of maintenance/sustainment activities and ensuring that these costs are not ignored when planning development activities. Management support is critical for stakeholders to understand and to respect the cost of sustainment activities.

Change management process – Having a process that governs how project artifacts (including requirements and code) are changed. Change management may include an approval process, change tracking, traceability and tool support. Change control boards and reviews by the engineering team help assure the correctness of changes and an assess change impact. Configuration management creates separate source code branches for co-development and creates rollbacks in case of error.

Sustainment and development separation – New development activities and maintenance activities are handled by separate teams. The separation of responsibility enables the development team to devote their time to new capabilities at the cost of additional manpower and communication overhead.

Design principles and techniques – Technical principles that guide the design or architecture of the system, such as design patterns, modularization and separation of concerns. Respondents suggested using design patterns, object de-coupling, a modular architecture and strong interface specification to create a system where components can be changed with minimal impact to the rest of the system.

Staffing strategies – Assigning particular development roles or specialists to a project, staff training, or personnel allocation. Several respondents suggested that the same team who develops the product also be responsible for its maintenance, or using "apprentices" during new development who take over sustainment activities after release. Staffing strategies also include having a separate system architect to oversee development and integration of separate components.

E. *Themes for the Integration challenge area*

The coding results of MPT themes for the stakeholder challenge area are shown in Fig. 4. The "design principles and techniques theme is the same as in the previous section.

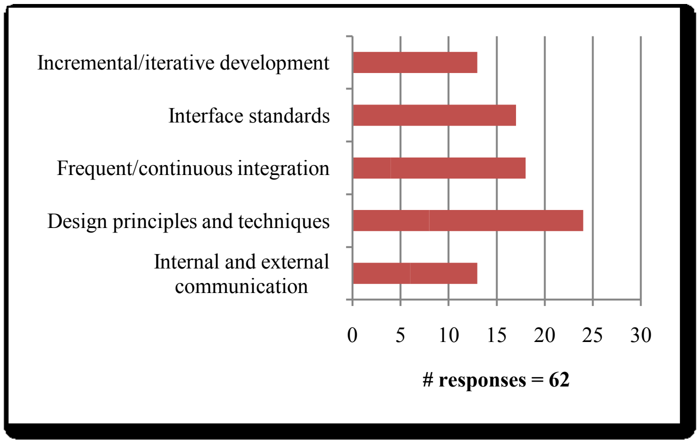

Figure 4. Most common themes for the Integration challenge area

Internal and external communication – Communication process to make technical information available within the development organization and to coordinate internal and supplier teams. Awareness of the development progress of external, dependent teams can be used to prioritize work items. Communication between internal teams facilitates better coordination and breaks through organizational stovepipes to keep developers aware of changes made to dependent components. A communication process ensures that change notifications are both timely and relevant.

Interface standards – A focus on the design of system and software interfaces, including APIs. Also includes processes for changing interfaces and disseminating interface standards/definitions to development teams. Enforcing strong interface standards and interface definitions across the system allows the system to be developed concurrently by multiple teams. Timely change notification regarding interface specifications keeps rework to a minimum.

Frequent / continuous integration – Regularly integrating individuals' code or sub-components into a larger system. Regular integration enforces interface standards and periodically verifies interoperability rather than struggling with large issues prior to delivery. Continuous integration is enabled by automated testing and an automated build process.

Incremental / iterative development – The development process occurs in short cycles rather than as one large process with a single deliverable. The impact on integration and interoperability is similar to frequent/continuous integration but includes attributes of delivery, including validation and customer feedback. Incremental delivery also provides intermediate components for dependent teams to test against.

V. Step Four - Mapping Concepts to Practices

In order to utilize the MPT themes for future scientific inquiry and to direct industry adoption of MPT themes, we have created MPT bridge diagrams. An MPT bridge diagram is comprised of three key elements: 1) the MPT theme; 2) the conceptual elements of the theme; and 3) a taxonomy of methods, processes and tools related to the theme. Fig. 5 depicts the bridge diagram for the "Frequent demonstration/delivery" theme and its associated MPTs.

The conceptual elements of an MPT are the constituent parts which define the theme. In Fig. 5, a time-boxed development process, delivering an intermediate product to stakeholders, evaluating the deliverable, and incorporating evaluation feedback into development are the essential elements of the theme. All four of these elements must be addressed or instantiated by a concrete practice in order to perform successful "frequent demonstration or delivery." The taxonomy on the right hand side lists the MPTs recommended by respondents under this theme and links them to the conceptual element(s) they instantiate.

The bridge diagram serves three purposes. First, reading from left to right, an organization adopting an approach to solving a challenge (i.e. adopting an MPT theme) can identify which MPTs are available to instantiate the conceptual elements of that theme. Second, going from right to left, an organization can assess whether or not their existing practices match the conceptual elements for a theme. If not, then there is some risk the theme is not properly instantiated and perhaps wasteful. Finally, the diagrams can help illustrate gaps between the conceptual elements of a theme and the state of the practice, i.e. where new MPTs need to be created or adapted in order to address a particular challenge area met by a theme.

VI. Conclusion and Future Work

Our work with the bridge diagrams has demonstrated them to be an effective way of communicating the results of our research to the sponsor, and of understanding effective system engineering MPTs that can fit within the sponsor's context. We have found that these diagrams give us a useful way of understanding the key system engineering *themes* that have to be present in order to address practical problems; once we have a checklist of these themes to look for we can move on to a more detailed discussion of whether the right MPTs have been chosen for effectively addressing those themes.

We caution that the work that has been done to date is aimed at this particular environment, and not meant to be generally applicable at this time. Ongoing work is taking the existing bridge diagrams and continuing to refine them against additional experiential information, allowing us to reflect what is known about the practical application of these MPTs from as wide a set of contexts as possible. The set of themes for which we have no concrete or effective MPTs will be the basis for future research work aimed at addressing these gaps.

Acknowledgment

The authors would like to thank Barry Boehm, Lori Clarke, Cihan Dagli, Jo Ann Lane and Lee Osterweil for their many discussions and feedback on this work.

References

[1] B. Boehm and J. Lane, "Using the Incremental Commitment Model to integrate system acquisition, system engineering, and software engineering," *CrossTalk*, vol. 20, issue 10, pp.4-9, Oct. 2007.

[2] A. Carrigy, E. Colbert, P. Componation, J. Fortune, L. Layman, S. O'Brien, D. Sabados, F. Shull, R. Turtner, "Evaluation of Systems Engineering Methods, Processes and tools on Department of Defense and Intelligence Community Programs," SERC-2009-TR-002, http://www.sercuarc.org/fileadmin/sercuarc/SERC_Publications/SERC_2009_MPT_TR_002.pdf, September 30, 2009.

[3] P. Duvall, S. Matyas, and A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk*, Addison-Wesley, Reading, MA, 2007.

[4] F. J. Fowler Jr., *Survey Research Methods*, Third Ed., Sage Publications, Thousand Oaks, CA, 2002.

[5] C. Larman and V. Basili, "Iterative and Incremental Devleopment: A Brief History," *Computer*, vol. 36, no. 6, pp. 47-56, June 2003.

[6] S. McConnell, *Rapid Devleopment: Taming Wild Software Schedules*, Microsoft Press, Redmond, WA, 1996.

[7] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," IEEE Transactions on Software Engineering, vol. 25, no. 4, pp. 557-572, July/Aug. 1999.

[8] A. L. Strauss and J. M. Corbin, *Basics of Qualitative Research: Techniques and Procedures of Developing Grounded Theory*, Second Ed., Sage Publications, Thousand Oaks, CA, 1998.

[9] C. S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*, Wiley-Interscience, Hoboken, NJ, 2005.

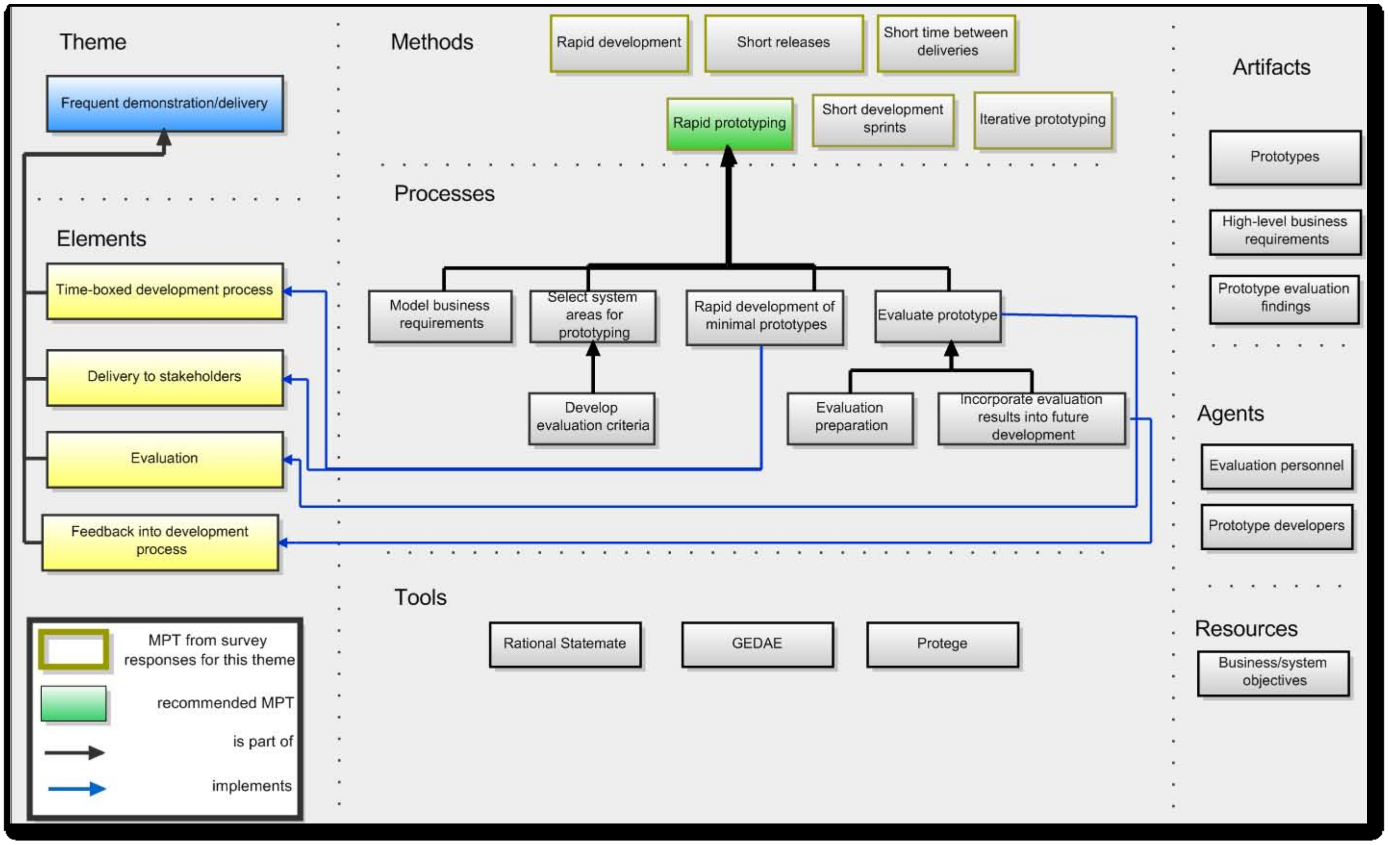
Theme
Frequent demonstration/delivery
Elements
Time-boxed development process
Delivery to stakeholders
Evaluation
Feedback into development process
Methods
Rapid development
Short releases
Short time between deliveries
Rapid prototyping
Short development sprints
Iterative prototyping
Processes
Model business requirements
Select system areas for prototyping
Rapid development of minimal prototypes
Evaluate prototype
Develop evaluation criteria
Evaluation preparation
Incorporate evaluation results into future development
Tools
Rational Statemate
GEDAE
Protege
Artifacts
Prototypes
High-level business requirements
Prototype evaluation findings
Agents
Evaluation personnel
Prototype developers
Resources
Business/system objectives
MPT from survey responses for this theme
recommended MPT
is part of
implements

Figure 5. Bridge diagram linking the theme of "Frequent demonstration/delivery" to specific recommended MPTs