
# Evaluation of Systems Engineering Methods, Processes and Tools on Department of Defense and Intelligence Community Programs

Technical Report SERC-2009-TR-002-1

September 30, 2009

**Principal Investigator:** Dr. Richard Turner, Stevens Institute of Technology

**Team Members:**

Fraunhofer Center at the University of Maryland: Dr. Lucas Layman, Dr. Forrest Shull

Stevens Institute of Technology: Anne Carrigy

University of Alabama in Huntsville: Dr. Paul Componation, Dr. Julie Fortune, Sue O'Brien, Dr. Dawn Sabados

University of Southern California: Ed Colbert

Contract Number: H98230-08-D-0171 Task Order: MPT, DO 001

**Report No. SERC-2009-TR-002-1**

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract H98230-08-D-0171 (Task Order DO 0001, TO 001, MPT).

# ABSTRACT

This report describes the results from the first in a series of related efforts to address systems engineering shortfalls in projects characterized as quick response, network-enabled, or emergent. The objectives of this task were to 1) identify methods, practices and tools (MPTs) considered viable in the Sponsor's environment; and 2) identify gaps where no useful or viable MPTs could be identified. Based on these products and their internal expertise, the team recommends three MPTs as most likely to increase effectiveness in the Sponsor's environment: Scrum, rapid prototyping, and continuous integration. Primary gaps identified were decision management, stakeholder requirements definition, and sustainment. Further research was recommended in three areas: mitigating environmental constraints, refining the definition of the current state of systems engineering practice, and accelerating MPT adoption.

This page intentionally left blank

# TABLE OF CONTENTS

**Abstract** ..... 3

**Table of Contents** ..... 4

**Figures and Tables** ..... 6

**1 Summary** ..... 7

**2 Introduction** ..... 9
- **2.1 Definitions** ..... 9
- **2.2 Approach** ..... 10

**3 Methods, Assumptions and Procedures** ..... 11
- **3.1 Interviews** ..... 11
- **3.2 Literature Survey** ..... 11
- **3.3 Industry Survey** ..... 12

**4 Results and Conclusions** ..... 13
- **4.1 Sponsor's Environment Summary and Key Challenge Areas** ..... 13
- **4.2 Correlation of Survey with Key Challenges** ..... 15
- **4.3 Survey Analysis** ..... 16
- **4.4 Gap Analysis** ..... 17

**5 Recommendations** ..... 19
- **5.1 Recommended Methods** ..... 19
- **5.2 Recommendations for Future Research** ..... 25

**Appendix A - Survey Questions** ..... 27
- **A.1. Demographics** ..... 27
- **A.2. Your Development / Acquisition Environment** ..... 28
- **A.3. Identification of Methods, Practices and Tools** ..... 28
- **A.4. Identifying Gaps** ..... 29
- **A.5. Community of Interest (Optional)** ..... 30
- **A.6. Other Feedback** ..... 30

**Appendix B – Survey Analysis (as of May 30, 2009)** ..... 31
- **B.1. Demographics** ..... 31
- **B.2. Initial Survey Analysis** ..... 32
- **B.3. In-depth Qualitative Analysis – Identifying Common Themes among MPTs** ..... 38
- **B.4. Pareto Analysis** ..... 43
- **B.5. Qualitative analysis summary** ..... 48
- **B.6. MPT References** ..... 48

# FIGURES AND TABLES

Figure 1. Systemigram of Phase I activities ..... 7

Table 1. Sponsor's environment ..... 13

Figure 2. Environmental responses ..... 16

Table 2. MPTs mentioned most frequently ..... 18

Table 3. Candidate MPT viability criteria ..... 19

Table 4. Candidate MPTs and associated rationale ..... 20

Figure 3. Scrum applied to the MPT taxonomy ..... 21

Figure 4. Rapid prototyping applied to the MPT taxonomy ..... 22

Figure 5. Continuous integration applied to the MPT taxonomy ..... 24

Table 5. Summary of the MPTs identified for each question ..... 35

Table 5. Continued ..... 36

Table 6. Critical areas identified as needing new MPTs ..... 37

Table 7. Counts of specific problems, MPTs and MPT themes from survey responses .. 38

Figure 6. Requirements MPT themes ..... 39

Figure 7. Stakeholder MPT themes ..... 40

Figure 8. Sustainment MPT themes ..... 41

Figure 9. Integration MPT themes ..... 42

Figure 10. Stakeholder coding results ..... 44

Figure 11. Integration coding results ..... 45

Figure 12. Sustainment coding results ..... 46

Figure 13. Requirements coding results ..... 47

# 1 SUMMARY

This research task is the first in a series of related efforts to address systems engineering shortfalls in projects characterized as quick response, network-enabled, or emergent. The objectives of this task were to 1) identify methods, practices and tools (MPTs) considered viable in the Sponsor's environment; and 2) gaps where no useful or viable MPTs could be identified. Figure 1 summarizes the activities.

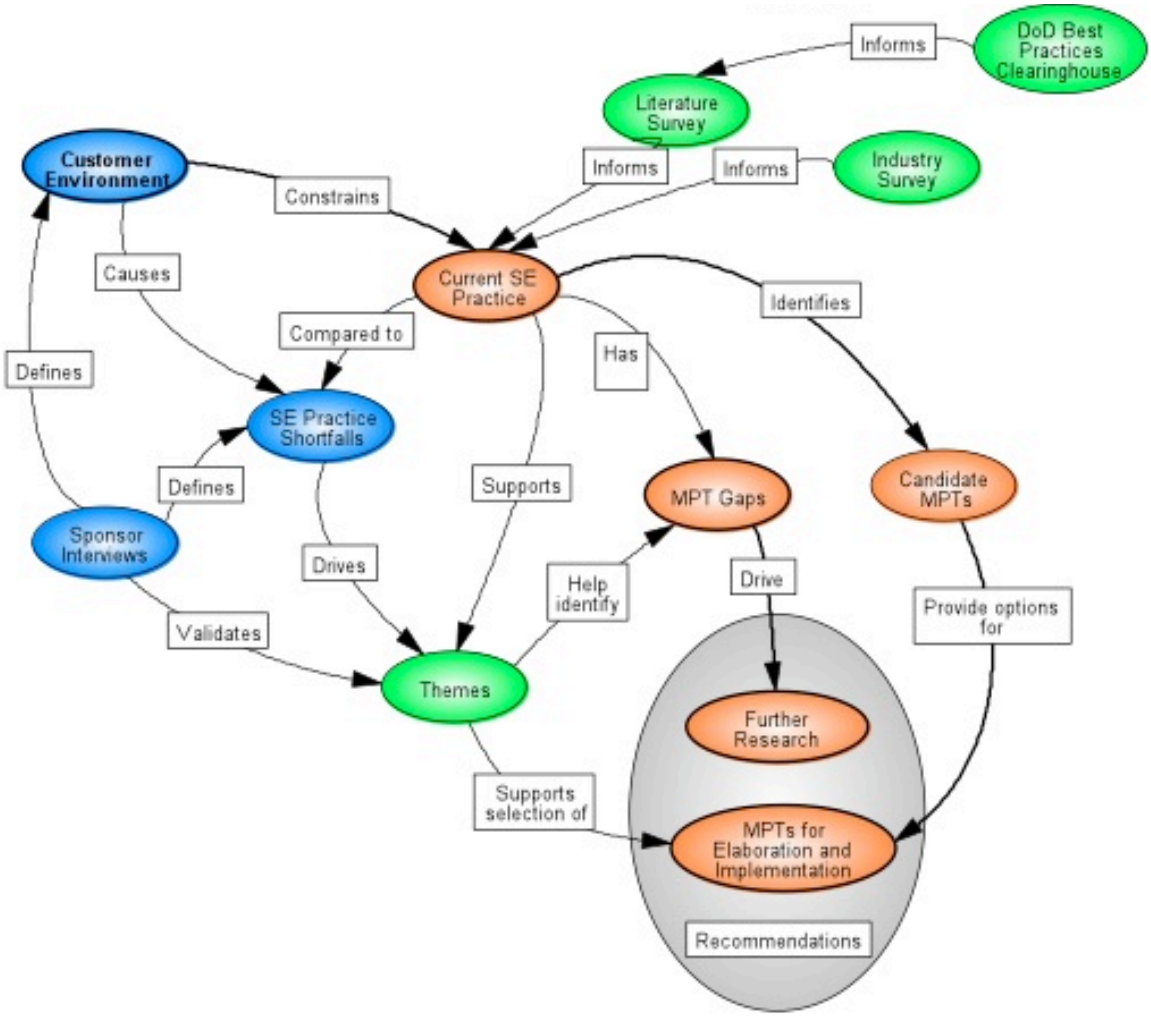


**Figure 1. Systemigram of Phase I activities**

Interviews with Sponsor development and acquisition personnel produced a description of the environment and its issues. The interviews established four key challenges:

1. Requirements - Changing requirements priorities and/or emerging requirements

2. Stakeholders – Obtaining useful stakeholder input and dealing with conflicting stakeholder requirements

3. Sustainment – Conflicts between developing new capabilities and supporting a currently deployed system

4. Integration – Integrating independently evolving components into a larger interoperable system

Literature and industry surveys were used to gather data on current systems engineering practice in similar environments. Consolidation and analysis of the data produced a set of common themes across the interview responses, a list of candidate MPTs that were recommended by organizations with environments similar to the Sponsor's, and a framework for showing the relationships among challenges, themes and MPTs. Based on these products and their internal expertise, the team recommends three MPTs as most likely to increase effectiveness in the Sponsor's environment:

- Scrum
- rapid prototyping
- continuous integration

Gaps were identified through the industry survey responses and by comparing useful, viable MPTs to the four challenge areas. Primary gaps identified through the survey were decision management and stakeholder requirements definition. The challenge area where the MPT gap was largest was Sustainment.

Further research was recommended in three areas:

- mitigating environmental constraints
- refining the definition of the current state of SE practice
- accelerating MPT adoption

# 2 INTRODUCTION

The Systems Engineering Research Center (SERC) Evaluation of Systems Engineering (SE) Methods, Processes and Tools (MPTs) research effort was initiated to provide a broad sense of the availability or absence of useful SE MPTs, particularly in a fast-paced, network-enabled, emergent development environment.

## 2.1 DEFINITIONS

An MPT is a systems engineering technique that fits into one of the following category definitions:

**Method (M)** - a collection of inter-related processes, practices and tools. A method is essentially a "recipe." It can be thought of as the application of inter-related processes, practices and tools to a class of problems with something in common.

**Process (P)** - a logical sequence of tasks intended to achieve an objective. The objective achieved may be abstract (e.g. “negotiate among multiple stakeholders”) and/or a composite of multiple individual goals (e.g. “Deliver a fixed-date, variable-scope system”). The structure of a process enables levels of aggregation to allow analysis at multiple levels of abstraction in support of decision-making.

**Practice (Pr)** - defines the "HOW" of each task. (The terms “practice,” “technique,” and “procedure” are often used interchangeably in disciplines such as systems engineering). The tasks associated with a process are performed using practices.

**Tool (T)** - automates or partially automates a specific practice or process, and thereby enhances task performance efficiency. The purpose of a tool is to facilitate the accomplishment of the “HOWs.” Some tools used to support systems engineering are model-aided.

A **useful** MPT is defined as one that is:

- *Relevant to the application environment:* applicable to some subset of systems within the target environment.
- *Repeatable:* sufficiently well defined that implementation is possible in a different context.
- *Likely to have significant impact:* can materially improve systems engineering practice in the application environment.

A **viable** MPT is successfully implementable in the target organization given appropriate and reasonable tailoring.

## 2.2 APPROACH

To focus this task, the needs and constraints of the Sponsor's specific environment were used to bound research activities. This report is based on data gathered from:

- discussions with Sponsor representatives
- interviews with the Sponsor personnel on challenges stemming from their unique agile system development environment
- MPT identification tasks performed by independent teams from open literature sources
- a survey of industry organizations thought to have similar environmental challenges

The report includes a description of the research methods, an analysis of the data collected, and recommendations of MPTs for further study and gaps that would benefit from focused research activities. The body of the document is focused on the results; detailed information is presented in appendices.

SERC organizations involved in the data collection and analysis include The Fraunhofer Center for Experimental Software Engineering, Stevens Institute of Technology, The University of Alabama in Huntsville, and the University of Southern California.

# 3 METHODS, ASSUMPTIONS AND PROCEDURES

A number of research methods were used to understand the specific challenges faced by the Sponsor and to identify potential MPTs for addressing those challenges. First, interviews were conducted with personnel at the Sponsor organization to identify the challenges facing the development groups. Second, a literature search was to identify an initial set of candidate MPTs based on existing best practices. Third, a survey of systems engineering experts was conducted to obtain suggestions for MPTs used by industry practitioners facing challenges similar to the Sponsor.

## 3.1 INTERVIEWS

Interviews were held with Sponsor personnel at management and practitioner levels. The objectives were to obtain a deeper understanding of the challenges regarding systems engineering in the target environment, to validate the description of the environment characterization, and to determine the current state of the practice and MPTs already being applied within the Sponsor organization.

Twelve interviews were completed with thirteen total participants from March through April 2009. A script was provided to guide the discussion, but individual answers often prompted follow-up questions to gain clarity on issues of interest that the basic script would have missed. Most interviews (10 of 12, 83%) were conducted with two interviewers and a single interviewee. One interview (1 of 12, 8%) was conducted with one interviewer, and one interview (1 of 12, 8%) was conducted with two interviewers and two interviewees. A review of the data did not indicate a difference in the range or type of data collected with the different data collection approaches. Interview results were captured in MS Word and reviewed by both interviewers. The approved summary of the interviews was then coded to remove information on the specific interviewee and then posted to the data set. The complete set of responses was made available to the MPT team for analysis.

## 3.2 LITERATURE SURVEY

The literature survey involved a high level, independent review of software and systems engineering literature and other sources to find candidate MPTs. The Sponsor's literature was also reviewed to establish a baseline of the current practices, identify useful MPTs, and identify areas where MPTs need to be modified or replaced. In order to assist in capturing information in a consistent manner for the literature search, a template was developed to record the high-level information about MPTs that may be relevant for the Sponsor's environment. The template helped capture information relating to the overall relevance, quality and usability of the MPT. The MPTs deemed useful were incorporated into the list of candidate MPTs

## 3.3 Industry Survey

To provide a broader sense of the available techniques and to gather information on gaps within the systems engineering processes under the constraints of the Sponsor's environment, a survey of commercial and defense industries is being conducted. The survey is being conducted online and contains demographic questions to assess the similarity of the survey respondents with the Sponsor's environment and questions asking respondents to identify MPTs for resolving the key challenges identified. A complete copy of the survey is provided in "Appendix A - Survey Questions."

As of July 15, 2009, 116 respondents had participated in some part of the survey; the following results are based on that sample. A full listing of responding organizations and the respondents' demographics can be found in "Appendix B – Survey Analysis." The respondents represented a large cross section of job functions, encompassing both management and engineering roles across multiple industries. Respondents typically had a significant degree of experience in systems and software engineering, but typically less experience in other engineering areas.

# 4 Results and Conclusions

The research team collected and analyzed data from the sources described in Section 3. The results and conclusions as described in this section were the basis for the recommendations in Section 5.

## 4.1 Sponsor's Environment Summary and Key Challenge Areas

The summary of the Sponsor's environment developed and validated in the interviews is presented in Table 1.

**Table 1. Sponsor's environment**

| | |
|---|---|
| **Overview Of The Target Environment** | A quick-reaction environment mixed with ongoing, more traditional acquisition |
| | Service-oriented approach with many different capabilities on many different platforms, many developed independently |
| | The complexity of the problems to be solved drive complex solutions |
| | Development (from concept to use) may be weeks or months |
| | System-level systems engineering exists, but is seen as secondary and not integral to the acquisition/development cycle |
| **Requirements Handling** | Requirements are often reacting to critical real-time needs |
| | Requirements are often vague, volatile, or immature |
| **System Interdependency** | Some services may depend on other services; dependency may be critical with no identifiable work-around |
| | Some services overlap or are duplicative |
| **System Evolution** | Good-enough may be sufficient for initial use |
| | Effective services may be scaled up, deployed widely, integrated into developing and legacy systems, and require operational support |
| | Services may evolve independently or based on the evolution of other services |
| | Services may have a lifetime of weeks or years |
| | There is a reluctance to replace/upgrade fielded services due to the risk of impacting other services |
| **Governance** | Service developers are diverse, dispersed, and have little inter-developer communications; teams often compete rather than collaborate; organizational culture and restrictions exacerbate communications difficulties |
| | Common oversight and cross-developer governance are inconsistent |
| | Traditional acquisition programs often have no insight into quick-response activities and vice versa |

The systems engineering shortfalls in the Sponsor's environment seem to be two-fold. The first is the rapid development of emerging capabilities with often vague initial requirements. The second is the scaling up and wide-spread deployment of the net-centric services that have been found to be useful. The Sponsor interviews, combined with conversations with Sponsor management, were used to create a description of the Sponsor's environment and discern the key challenges facing Sponsor development teams. The four key challenge areas to be addressed with MPTs are:

1. **Requirements -** Changing requirements priorities and/or emerging requirements
2. **Stakeholders** – Obtaining useful stakeholder input and dealing with conflicting stakeholder requirements
3. **Sustainment** – Conflicts between developing new capabilities and supporting a currently deployed system
4. **Integration** – Integrating independently evolving components into a larger interoperable system

The four challenge areas relate well to the stated SE shortfalls. Challenge areas 1 and 2 are closely related to the rapid development concerns and areas 3 and 4 are indicative of the scaling and deployment shortfall.

### 4.1.1 Requirements

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 something 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.

### 4.1.2 Stakeholders

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 may preclude direct engagement between customers and developers.

### 4.1.3 Sustainment

The evolutionary nature of the solutions, the fact that "good enough" is fine for initial deployment, and that these are likely to be fairly small scale development projects suggests that the environment is amenable to iterative approaches. Systems are initially

developed with local functionality, performance and security in mind. The problem is how to best support the widespread deployment of a capability that worked "good enough" for initial release, but may not meet the more rigorous non-functional requirements needed for integration into the larger system.

### 4.1.4 Integration

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 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.

## 4.2 Correlation of Survey with Key Challenges

As shown in

Figure 2, a majority of the responses showed alignment with the environment. The individual respondent's degree of agreement with the Sponsor's environment was calculated using a weighted average of their responses on question 5. Each statement could be answered with "Strongly Agree", "Agree", "Disagree", or "Strongly Disagree", providing a basis for applying a Likert scale with 4 values. Traditionally used to make observations about the general population being surveyed, these values were used to rank individual respondents. The scale was 1, 0.5, -0.5, -1 for the response range, respectively, and scores for each respondent were summed. A symmetric scale with positive and negative values was selected to enhance spread in the data. Thus, with 6 questions, an ideal fit to the Sponsor's environment would score a total of 6 points (answering each question with Strongly Agree), an organization with no commonality would score a -6, and most organizations would be expected to fall somewhere in between. The following categories were established for scores:

- Excellent fit to Sponsor's environment > 5 (11% of responses)
- Good fit to Sponsor's environment 3 to 4.5 (34%)
- Some commonality 0 to 2.5 (45%)
- No commonality < 0 (10%)

This grouping allowed MPT investigation and gap analysis to focus on topics most directly applicable to the Sponsor.

| 5. Please indicate your level of agreement with the following statements. | | | | | |
|---|---|---|---|---|---|
| | Strongly Disagree | Disagree | Agree | Strongly Agree | Response Count |
| My product development environment has an emphasis on quick-reaction to stakeholder needs. | 3.7% (4) | 16.5% (18) | **45.0% (49)** | 34.9% (38) | 109 |
| Requirements are often unstable because of changes in the operating environment. | 1.8% (2) | 15.6% (17) | **51.4% (56)** | 31.2% (34) | 109 |
| Capabilities delivered to stakeholders are dependent on other existing deployed systems. | 2.8% (3) | 12.8% (14) | **45.0% (49)** | 39.4% (43) | 109 |
| The capabilities I am developing will be deployed in a wide range of operating environments. | 4.6% (5) | 22.9% (25) | **41.3% (45)** | 31.2% (34) | 109 |
| My product development environment often operates independently of other product development efforts within my organization. | 14.7% (16) | 28.4% (31) | **31.2% (34)** | 25.7% (28) | 109 |
| Changes in requirements for system upgrades are common once the initial capability is delivered to the stakeholder. | 2.8% (3) | 18.3% (20) | **46.8% (51)** | 32.1% (35) | 109 |
| | | | | ***answered question*** | **109** |
| | | | | ***skipped question*** | **7** |

**Figure 2. Environmental responses**

## 4.3 Survey Analysis

Two analyses of the survey data were conducted to identify candidate MPTs. In the first analysis, specific MPTs identified by the survey responses were counted for each of the four challenge areas: requirements, stakeholder challenges, sustainment, and integration. Also, critical areas needing new MPTs in agile systems engineering were also identified and counted. In the second analysis, an in-depth qualitative analysis was performed to group the MPTs into MPT "themes" that shared a common approach for addressing a challenge. The MPT themes provide a means of bridging between specific challenges and specific MPTs and supported the gap analysis of target environments. Complete details of the survey analyses may be found in "Appendix B – Survey Analysis." These analyses directly informed the recommendations in the following section.

## 4.4 Gap Analysis

The initial gap analysis was provided by the responses to question six of the industry survey. Respondents were asked to identify three or four of the ISO/IEC 12207 process areas they saw as the most in need of new or revised MPTs. As described in Appendix B, the views of the survey respondents concerning gaps were broad. However, when the analysis only considered those respondents with either good or excellent correlation to the Sponsor's environment, there were eight areas that more than 25% of the respondents identified as needing additional MPTs. These were:

- Decision Management (47%)
- Stakeholder Requirements Definition (40%)
- Measurement (28%)
- Architectural Design (28%)
- Integration (28%)
- Project Planning (26%)
- Project Assessment and Control (26%)
- Risk Management (26%)

Decision Management and Stakeholder Requirements Definition were obviously the primary gaps identified. It is interesting that nearly two-thirds of the most frequently mentioned MPTs fall in the Stakeholder and Requirements challenge areas. It is clear there are MPTs that are being used, but they are evidently not perceived as sufficient. Decision Management as a gap points to governance issues where there are very few MPTs mentioned (some examples are integrated product teams, configuration control boards, trade studies and the incremental commitment model). Missing in the conversation are supporting MPTs such as enterprise architectures and concepts of operations.

Table 2 shows the most mentioned candidate MPTs collected through the industry survey versus the key challenges for which they were suggested. A gap is apparent in the low number of MPTs applicable to the sustainment challenge.

**Table 2. MPTs mentioned most frequently**

| Practice | Req. | Stake. | Sust. | Int. |
|---|---|---|---|---|
| Rapid Prototyping | ✓ | ✓ | | ✓ |
| Continuous Integration | | | ✓ | ✓ |
| Iterative / Incremental Development | ✓ | ✓ | | ✓ |
| Interface Control Document (ICD) | ✓ | ✓ | | ✓ |
| Incremental Commitment Model (ICM)[1] | ✓ | ✓ | | |
| Stakeholder Analysis | ✓ | ✓ | | |
| Sustainment Plan | | ✓ | ✓ | |
| Requirements Arbitration | | ✓ | | |
| Scrum | ✓ | ✓ | ✓ | |
| Requirements Impact Analysis | ✓ | ✓ | | |
| Separate Teams for Development & Sustainment | | | ✓ | |
| Requirements Traceability | ✓ | ✓ | | |
| System Modeling / System Modeling Language | ✓ | ✓ | | |
| Trade Studies | | ✓ | ✓ | |
| Change Impact Analysis | ✓ | | ✓ | |
| Integrated Product Team (IPT) | ✓ | | | ✓ |
| Model-Based Testing (MBT) | | ✓ | | ✓ |
| Modeling and Simulation | ✓ | | ✓ | ✓ |
| Service-Oriented Architecture (SOA) | ✓ | | | ✓ |
| User Stories | ✓ | | | |
| Stakeholder Interviews | | ✓ | | ✓ |
| Block Upgrades | ✓ | | | ✓ |
| Cost As an Independent Variable (CAIV) | ✓ | | | |
| Change Control Board | | | ✓ | ✓ |
| Customer Co-location | ✓ | | | ✓ |
| Feature-Driven Development (FDD) | ✓ | | | |
| Focus groups | | ✓ | | |
| Open Standards | ✓ | | | |
| Requirements Volatility Analysis | ✓ | | | |
| Retain Key Engineers in Sustainment | | | ✓ | |
| Reviews (small team) | | | ✓ | ✓ |
| Self-generating Documentation | | | ✓ | ✓ |
| Stakeholder Roleplaying | | ✓ | | |
| Storyboarding | ✓ | ✓ | | |
| System Architect Role | ✓ | | | ✓ |
| WinWin | ✓ | ✓ | | |
| **Total MPTs** | **23** | **18** | **11** | **15** |

# 5 RECOMMENDATIONS

The following recommendations represent the consensus of the research team. Other recommendations were considered. Succeeding tasks will refine and augment those provided.

## 5.1 RECOMMENDED METHODS

Table 3 describes the rationale for the criteria used to select viable MPT candidates.

**Table 3. Candidate MPT viability criteria**

| Criteria | Rationale |
|---|---|
| Likely to reduce schedule. | A sprint-based schedule was the primary driver at the Sponsor's organization. |
| Have some measures of effectiveness available. | The risk of implementing MPTs with no track record of effectiveness makes adoption nearly impossible for Sponsor project managers. |
| Can be applied at the level of individual projects or lower in the hierarchy. | Organizational circumstances impede the ability to standardize at higher levels. |
| Are useful for system developers (as opposed to acquirers). | Interviews indicated system development was the area of primary need. |
| The MPT can be adapted to work within the Sponsor's environment | It is useless to recommend something that cannot be adapted for the intended environment. |
| The MPT can be implemented with minimal resource overhead | Resource constraints in the Sponsor's environment make this a necessity. |

The team proposed several candidate MPTs for each challenge area. These MPTs may require tailoring to the Sponsor's environment. Each MPT was suggested based on the following inclusion factors (table identifier in parentheses):

- number of mentions in the industry survey (*I*);
- number of recommendations from research team members (*R*);

The table below contains the candidate MPTs and their inclusion factors. A list of MPTs observed in the survey may be found in Appendix B. A checkmark (✓) in a challenge area column indicates that the MPT was mentioned in response to the question pertaining to that challenge area.

The team discussed the candidates and decided to recommend three. The lack of a consensus on an MPT to address the sustainment challenge area further illustrates its recognition as a gap. The highlighted MPTs are recommended and are described in more detail in the following sections. Iterative/incremental development was not

specifically recommended because it is a broad philosophy and is included as a key component of many of the other MPTs, including Scrum.

**Table 4. Candidate MPTs and associated rationale**

| MPT | Req. | Stake. | Sust. | Int. | I | R |
|---|---|---|---|---|---|---|
| **Rapid prototyping** | ✓ | ✓ | | ✓ | 13 | 6 |
| **Continuous integration** | | | ✓ | ✓ | 11 | 5 |
| Iterative / incremental development | ✓ | ✓ | | ✓ | 11 | 5 |
| **Scrum** | ✓ | ✓ | ✓ | | 5 | 5 |
| System Model / System Modeling Language | ✓ | ✓ | ✓ | ✓ | 4 | 5 |
| User Stories | ✓ | | ✓ | | 3 | 1 |
| Feature-driven development | ✓ | | | | 2 | 1 |
| WinWin | ✓ | ✓ | | | 2 | |
| Competitive prototyping | ✓ | ✓ | | | 0 | |
| Single product owner/decision maker | ✓ | ✓ | | | 7 | 1 |
| Same-time/same-place customer meetings | ✓ | ✓ | | | 3 | |
| Service-oriented architecture | ✓ | | | ✓ | 3 | 2 |
| Cross-functional integration teams | | | | ✓ | 3 | 1 |

### 5.1.1 Scrum

Scrum is a lightweight project management methodology that only addresses project management and planning; specific technical processes are left up to the team. Scrum was originally designed for smaller teams composed of a ScrumMaster (project manager), the product owner (stakeholder representative), and the development team. Requirements are determined by input from all stakeholders and broken down into a set of (preferably) independent features that can be completed in 2-4 weeks each. The features, along with their estimated time to completion, are kept in a *product backlog*. Analysis, development and testing take place in 2-4 week iterations called *sprints*. At the beginning of each sprint, the product owner, project manager, and team determine which features can be completed during the sprint, considering priority and the estimated time required to complete a feature.

During the sprint, the project manager ensures that the team is focusing on their assigned tasks and enforces accurate reporting of the amount of time spent on each feature. The specific practices and techniques used during analysis, implementation, and testing during the sprint are at the discretion of the development team. Each day, the team gathers for a daily stand-up meeting where each member describes what he/she completed the previous day, what he/she plans to do today, and anything that is blocking the team member from completing his/her goal.

At the end of each sprint, the team updates the product backlog to reflect the time spent on each feature and the time remaining on each feature. The team and stakeholders meet to discuss problems and difficulties and to plan the next sprint accordingly.

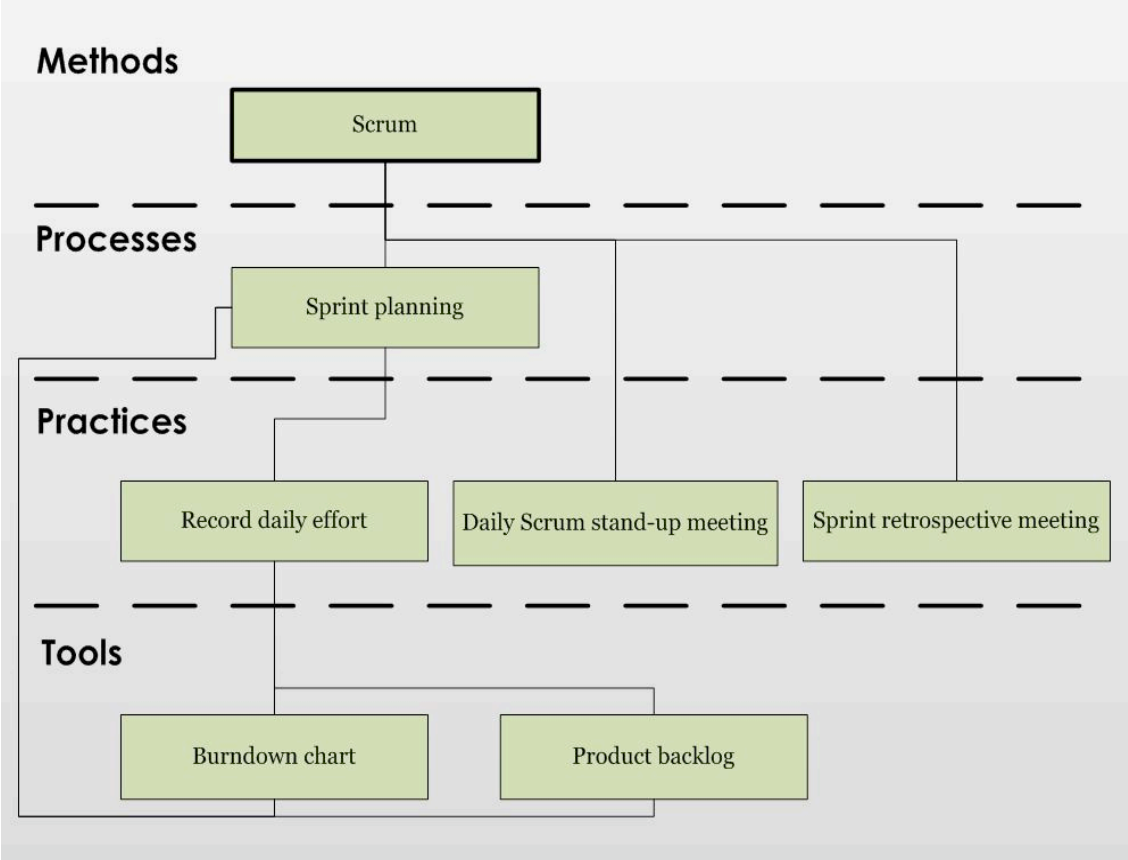


**Figure 3. Scrum applied to the MPT taxonomy**

**Processes and Practices:**

- Sprint planning - Requirements are broken down into multiple (independent) features that can be completed in 2-4 week development sprints. Each sprint should produce an incremental deliverable with functioning code and tests. Detailed feature planning and development activities take place prior to each sprint. Planning includes the development team, project manager and customer representative. Evaluation of progress and risks in a meeting with all stakeholders at the end of each sprint

- Daily Scrum stand-up meeting - Daily stand-up meetings held by the development team where individual members report on what they accomplished the previous day, what is planned for today, and any blocking problems that are preventing them from accomplishing their goals.

- Record daily effort – Developers record the effort spent on feature on daily basis. This progress is tracked through a product backlog which accumulates the features still to be developed, and burn-down charts show the remaining work. The effort tracking is used as a planning aid in the sprint planning meetings.

### 5.1.2 Rapid Prototyping

Rapid prototyping (often called Rapid Application Development when concerning software) is the development of an executable system that implements or mimics a subset of the desired functionality of a system. The working prototype is used to uncover errors, ambiguities or omissions in the system requirements, identify usability and other non-functional concerns, improve design, and improve maintainability. By rapidly producing prototypes, developers can solicit feedback quickly and early in the development process from key stakeholders. Rapid prototyping creates *minimal* systems that demonstrate or explore functional requirements, but without the architectural robustness or non-functional properties necessary for integration into the final deliverable. Rapid prototyping can also be used to create minimal systems that exhibit desired behavior for testing against the deployment system.

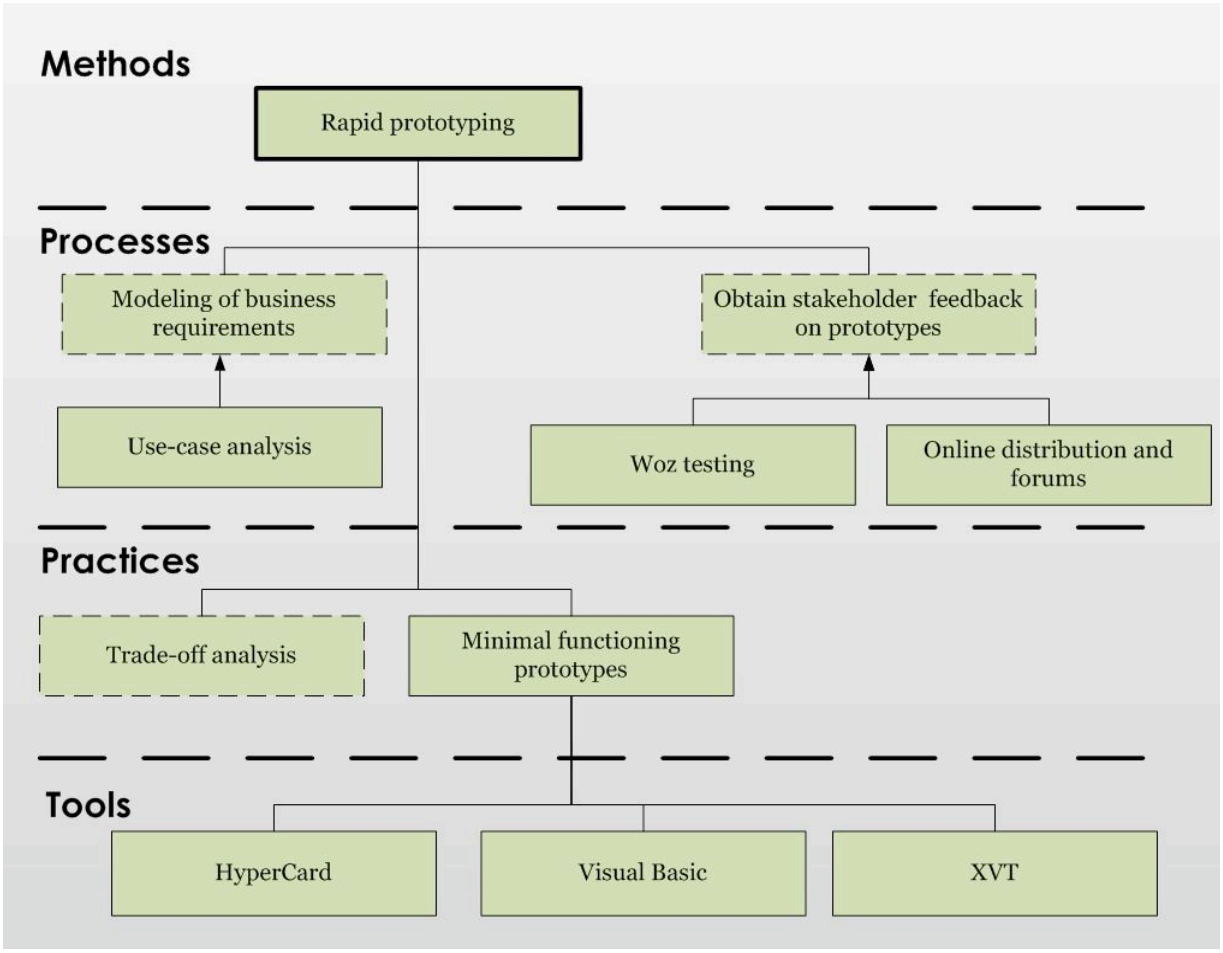


**Figure 4. Rapid prototyping applied to the MPT taxonomy**

**Processes and Practices:**

- Modeling of business requirements – A representation of the customer's business requirements that serve as the basis for developing prototypes. These may be modeled using use cases, user stories, or high level functional requirements.
- Obtain stakeholder feedback on prototypes - Functioning prototypes are demonstrated to and evaluated by multiple stakeholders to inform changes in requirements, architecture, and scope. Feedback may be obtained following online distribution in forums or through conferencing, or by more formal evaluations.
- Minimal functioning prototypes – These are created to explore specific elements of the desired system, such as a particular requirement, user interface, or architecture decision
- Trade-off analysis - Rapid prototyping requires a careful balancing of the tradeoffs: A quickly developed prototype can effectively solicit stakeholder feedback in a timely fashion, but should not be integrated with the final system due to quality concerns; A more robust prototype requires more effort for development, but incurs the risk that the features demonstrated do not answer stakeholder needs.

### 5.1.3 Continuous integration

Continuous integration (CI) is a software development process where members of a team integrate their work frequently; usually each person integrates at least daily - leading to multiple integrations per day. The term 'continuous integration' originated as one of the original twelve eXtreme Programming development practices. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible. Bad builds may occur from time to time, and a CI implementation that includes rollback features can reduce down time when an integration fails. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. The result of doing this is that there is a stable piece of software that works properly and contains few bugs.

Although many organizations do regular builds on a timed schedule, such as once each night, this is not an instance of CI. The point of CI is to find problems as soon as possible. Scheduled builds mean that bugs lie undetected for the entire duration between builds and it takes longer to find and remove the bugs.

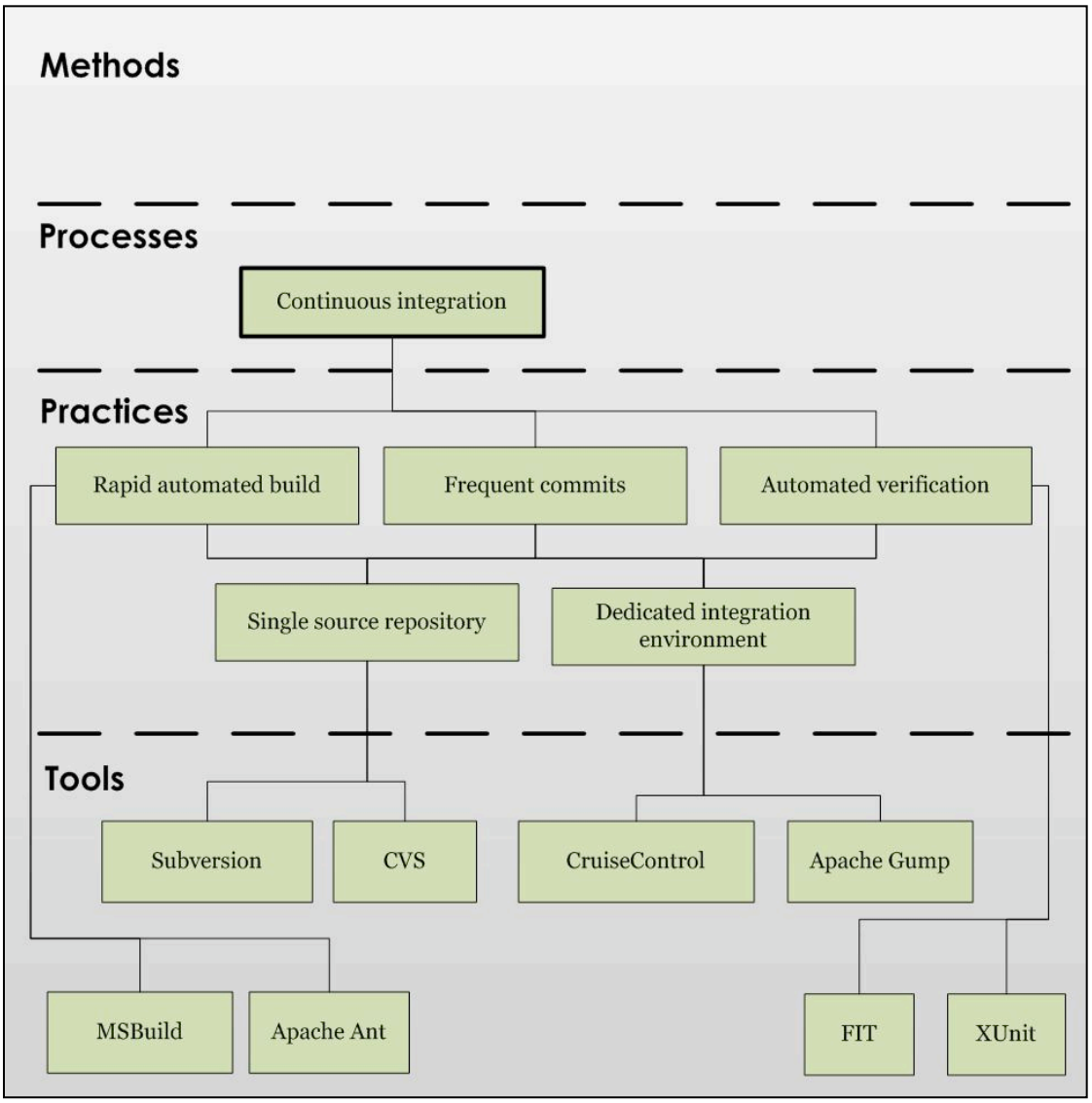


**Figure 5. Continuous integration applied to the MPT taxonomy**

**Practices:**

- Rapid automated build - The build process is automated using build scripts and should be quick (10 minutes or less) to provide rapid feedback.
- Automated verification - The build process is paired with the execution of an automated test suite (a regression suite) that indicates whether a committed change has caused a test failure in the overall product.
- Frequent commits - Most effective at detecting integration errors when all developers commit their changes regularly, preferably multiple times per day. By doing this frequently, developers quickly find out if there's a conflict between two developers. Because so little has happened between builds it is easier to identify and resolve.

Frequent commits encourage developers to break down their work into small chunks which helps track progress and provides a sense of accomplishment.

- Dedicated integration environment – Every commit triggers a build on an integration machine based on the mainline code (a final check to ensure there are no conflicts with changes from another developer). This practice represents infrastructure that supports other processes and practices.

- Single source repository - Product development is tracked through a single, centralized source repository. This repository should include everything needed to do a build (test scripts, properties files, database schema, install scripts, and third party libraries). All developers should have access to the repository to observe changes made to the system in real-time if necessary. This practice represents infrastructure that supports other processes and practices. Although many teams use repositories, a common mistake is not putting everything in the same repository.

## 5.2 Recommendations for Future Research

This section presents the team's recommendations for research in succeeding tasks. This includes, but does not specifically call out the research underway in the MPT Extension Research Task (RT-9). These broadly stated recommendations will be refined with more specific activities in the MPT Extension Final Report due December 15, 2009.

### 5.2.1 Mitigate Environmental Constraints

As discussed in Section 4.1, the Sponsor's environment places significant constraints on the engineering methods, practices and tools that may be effectively applied. The following recommendations address the identification and creation of MPTs to better fit existing constraints as well as the study of enabling technologies that reduce the impact of those constraints on MPTs.

- Enabling Strategic Activities - Investigation into creative ways to build, evaluate, use and evolve the concept of operation, enterprise architecture and technical architecture artifacts in constrained environments.

- Communications and Integration - Investigation into new ways to handle communication, integration and interface control in compartmented development environments.

- Brownfield Development - Investigation into rapid capability development in systems with complex legacy components.

### 5.2.2 Refine the Definition of the Current State of Systems Engineering Practice

These activities will broaden and deepen the current description of the state of the practice:

- Enhance the MPT definition framework through follow-up interviews with industry survey respondents.
- Refine the description and understanding of the Sponsor's environment.
- Conduct a literature survey more focused on specific MPT groups and identified gaps.
- Conduct interviews and workshops with more practitioners from industry environments, particularly those that may be considered surrogates for the Sponsor's environment.

### 5.2.3 Accelerate MPT Adoption

These activities will better enable the Sponsor to implement recommended MPTs:

- Develop implementation packages for recommended MPTs tailored to the Sponsor's environment.
- Recommend additional MPTs for Sponsor implementation.

# APPENDIX A - SURVEY QUESTIONS

The following are the questions asked on the survey. An asterisk (*) indicates a required answer.

## A.1. DEMOGRAPHICS

*1. Your company or organization (e.g. Coca-Cola, US Navy)

2. Your division, group or sub-organization (e.g. Coca-Cola Europe Group, NAVAIR) [Optional]

*3. Which of the following best describes your primary job function?

Functional Manager

Project Manager

Lead Systems Engineer

Systems Engineer – Supporting Multiple Projects

Systems Engineer – Supporting a Single Project

Software Engineer

Engineer

Other (please specify)

*4. Please indicate your years of experience in each of the following

| | 0 | 1-5 | 6-10 | 11-20 | >20 |
|---|---|---|---|---|---|
| Systems Engineering | | | | | |
| Software Engineering | | | | | |
| Other Engineering Field (Electrical, Mechanical, aerospace, etc.) | | | | | |

## A.2. Your Development / Acquisition Environment

We would like to know more about the environment where you applied agile systems engineering. This could be your present project or a project you worked on in the past.

*1. Please indicate your level of agreement with the following statements (Strongly Disagree, Disagree, Agree, Strongly Agree):

My product development environment has an emphasis on quick-reaction to stakeholder needs.

Requirements are often unstable because of changes in the operating environment.

Capabilities delivered to stakeholders are dependent on other existing deployed systems.

The capabilities I am developing will be deployed in a wide range of operating environments.

My product development environment often operates independently of other product development efforts within my organization.

Changes in requirements for system upgrades are common once the initial capability is delivered to the stakeholder.

## A.3. Identification of Methods, Practices and Tools

These questions help us to build a database of systems engineering methods and tools that have been successfully applied in these non-traditional development environments.

*1. Are changing requirements priorities or emerging requirements a challenge in your product development environment? (Yes, No)

2. Are there any approaches you would recommend to other teams for dealing with this problem?

*3. Is getting useful stakeholder input, or dealing with conflicting stakeholder requirements, a challenge in your product development environment? (Yes, No)

4. Are there any approaches you would recommend to other teams for dealing with this problem?

* 5. Is resolving conflicts between developing new capabilities and supporting the currently released system a challenge in your product development environment? (Yes, No)

6. Are there any approaches you would recommend to other teams for dealing with this problem?

*7. Is integrating independently evolving components into a larger system a challenge in your product development environment? (Yes, No)

8. Are there any approaches you would recommend to other teams for dealing with this problem?

*9. Are there any approaches you tried to address the above problems that were not effective? (Yes, No)

10. If so, please describe the unsuccessful approaches.

## A.4. Identifying Gaps

While we are interested in identifying things that work and things that don't work, we are also interested in where there seems to be nothing available. Please help us identify places where research and new approaches would be welcome.

1. Please identify the 3 or 4 areas where you believe the most critical need for new approaches are found.

| | |
|---|---|
| Acquisition | Measurement |
| Supply | Stakeholder Requirements Definition |
| Life Cycle Model Management | Requirements Analysis |
| Infrastructure Management | Architectural Design |
| Project Portfolio Management | Implementation |
| Human Resource Management | Integration |
| Quality Management | Verification |
| Project Planning | Transition |
| Project Assessment and Control | Validation |
| Decision Management | Operation |
| Risk Management | Maintenance |
| Configuration Management | Disposal |
| Information Management | |

2. Please let us know about any specific needs you have discovered for better methods, practices and tools.

## A.5. Community of Interest (Optional)

1. If you would like a copy of the findings report and a list of responding organizations, please enter your email address here.

2. We are establishing a community of interest around agile systems engineering. Would you like to be added to the list of interested parties? (You will need to enter your email address above) (Yes, No)

## A.6. Other Feedback

A chance for you to comment on the survey or the concept of agile systems engineering.

1. Please add any comments here.

# APPENDIX B – SURVEY ANALYSIS (AS OF MAY 30, 2009)

## B.1. DEMOGRAPHICS

**Organizations responding**

AB Volvo
Advanced Micro Devices
AMD
Army Acquisition
AT&T
Binion Enterprises, LLC
Boeing
California DOT
Cobham
DCMA
Delft University of Technology
Delta
DoD
DRS Technologies
FMC Technologies
Harris Corporation
International Game Technology
Intuit
Jacobs Technology Inc.
Jet Propulsion Lab
Kongsberg Defence & Aerospace
Kongsberg Maritime AS
Lockheed Martin
Micronic Laser Systems
National Security Space Office
Nokia Corporation
Nokia Siemens Networks
Northrop Grumman Corporation
NXP
Océ-Technologies B.V.
Paradigm Optics
Performance-Based Earned Value
Philips
Philips Research
Planmeca Oy
Pragma Systems Corporation
QinetiQ North America
Rijkswaterstaat
Rockwell Collins
Rottne Industri AB, Sweden
SAAB
SAIC
Sandia National Laboratories
Scania Commercial Vehicles
Sioux Embedded Systems B.V.
Sun Microsystems
Syntell AB
TopCoder
Topic Embedded Systems
UAHuntsville
Univa UD
University of Southern California
US Army
US Navy
Volvo
Volvo Construction Equipment, Sweden
Walt Disney Imagineering

| 3. Which of the following best describes your primary job function? | | | |
|---|---|---|---|
| | | Response Percent | Response Count |
| **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 Engineer – Supporting a Single Project | | 6.9% | 8 |
| Software Engineer | | 0.9% | 1 |
| Engineer | | 2.6% | 3 |
| | Other (please specify) | | 32 |
| | ***answered question*** | | **116** |
| | ***skipped question*** | | **0** |

| 4. Please indicate your years of experience in each of the following | | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1-5 | 6-10 | 11-20 | >20 | Response Count |
| Systems Engineering | 0.9% (1) | 28.3% (32) | 23.9% (27) | **29.2% (33)** | 17.7% (20) | 113 |
| Software Engineering | 15.3% (15) | 19.4% (19) | 14.3% (14) | **25.5% (25)** | **25.5% (25)** | 98 |
| Other Engineering Field (Electrical, Mechanical, aerospace, etc.) | 14.5% (12) | **24.1% (20)** | 22.9% (19) | 15.7% (13) | 22.9% (19) | 83 |
| | | | | ***answered question*** | | **116** |
| | | | | ***skipped question*** | | **0** |

## B.2. Initial Survey Analysis

Five questions were developed to help identify both suitable candidate MPTs, and MPTs that proved to be unsuccessful when implemented. The research team selected a subset of the MPTs as most important for further analysis based on the following criteria: multiple respondents had suggested the MPT; the respondent who did recommend the MPT seemed a particularly good match to the Sponsor's environment; and, the MPT description was sufficiently concrete and specific implying a high likelihood of success

transitioning it into the Sponsor's environment. The most promising MPTs identified for each question are summarized below, with Table 5 providing a complete summary of the MPTs identified for each question.

### B.2.1. Question 1: "Are changing requirements priorities or emerging requirements a challenge in your product development environment?"

Ninty-four respondents (91% of our sample) agreed that this was an issue in their environment. The most promising MPTs identified under this question are: Incremental Commitment Model (ICM)[1], Iterative/Incremental Development, Rapid Prototyping, Scrum[2], System modeling / System Modeling Language, Feature Driven Development (FDD)[3], Requirements Impact Analysis, and Requirements Traceability. When the respondents did not name specific MPTs, initial analysis of commonalities in the responses indicated that the following general themes or principles were most often mentioned:

Balancing agility and discipline: Respondents emphasized that neither an entirely agile nor an entirely disciplined approach was likely to produce the desired outcomes.

"Build safety in": Although agile approaches often advise against doing more design up-front than absolutely necessary, the respondents emphasized that safety and reliability are properties that have to be addressed early in the lifecycle. It is very difficult (if not impossible) to add them later to an unsafe, unreliable system.

Contracting for flexibility: Respondents described the general goal of leaving some flexibility in the contracting process to deal with emerging requirements.

Modular design: Modular systems are generally easier to adapt or build upon later in the lifecycle as new priorities emerge.

Requirements elicitation techniques: Some classes of emerging requirements can be avoided by doing a more thorough job of eliciting requirements from the user or customer in the first place. In some cases, these techniques can help the user better understand his/her own needs or work process. Such techniques could take the form of customer interviews; exercises such as "laddering" or card sorting that ask users to work with concepts, or observation of the customer while they perform their role.

Stakeholder involvement: This was by far the most important recurring concept, mentioned in some way by at least 14% of respondents. The central idea was to make sure that the right stakeholders are involved in designing the system as early as possible, are given the opportunity to respond to new features as they are implemented, etc.

### B.2.2. Question 2: "Is getting useful stakeholder input, or dealing with conflicting stakeholder requirements, a challenge in your product development environment?"

Ninety-three respondents (90% of our sample) agreed that this was an issue in their environment. The most promising MPTs identified under this question are: Rapid Prototyping, Requirements Arbitration, Stakeholder Analysis, Focus Group, Requirements Impact Analysis, and Requirements Traceability. When the respondents did not name specific MPTs, initial analysis of commonalities in the responses indicated that the following general themes or principles were most often mentioned:

Continuous validation of requirements: Respondents advocated treating the requirements documentation as a living document and revisiting it periodically to make sure that it still reflects the needs of the system users.

Education and training in agile: Agile can help bring stakeholders on board. The respondent advocates starting with small, pilot projects to show the benefits, before scaling up.

Stakeholder involvement: Again, finding an effective mechanism for stakeholder involvement was far and away the most often-mentioned theme of respondents' answers. Approximately 18% of responses mentioned this in one way or another.

### B.2.3. Question 3: "Is resolving conflicts between developing new capabilities and supporting the currently released system a challenge in your product development environment?"

Seventy-eight respondents (76% of our sample) agreed that this was an issue in their environment. The most promising MPTs identified under this question are: Sustainment Planning, Configuration Management (CM), Separate Teams for Development and Sustainment, Continuous Integration, and User Stories. When the respondents did not name specific MPTs, initial analysis of commonalities in the responses indicated that the following general themes or principles were most often mentioned:

Modular design: Modular systems are generally easier to adapt or build upon later during sustainment.

Stakeholder involvement: Respondents emphasized the importance of following the "voice of the customer" in making these tradeoffs.

### B.2.4. Question 4: "Is integrating independently evolving components into a larger system a challenge in your product development environment?"

Eighty respondents (78% of our sample) agreed that this was an issue in their environment. The most promising MPTs identified under this question are: Continuous Integration, Interface Control Document (ICD), Iterative/Incremental Development, Rapid Prototyping, and Model-Based Testing (MBT)/ Model-Driven Engineering. When the respondents did not name specific MPTs, initial analysis of commonalities in the responses indicated that the following principle was most often mentioned:

Modular design: Modular systems are generally easier to adapt or build upon later. By minimizing the size of the interface (i.e. the information that has to be known in order for another system to integrate with it), it becomes easier to re-combine systems after development in unforeseen ways.

### B.2.5. Question 5: "Are there any approaches you tried to address the above problems that you found were not effective?"

The set of items that respondents listed included: Configuration Management (CM), CMMI, freezing the code, freezing the requirements, Integrated Project Schedules (IPS), lack of control of interfaces / Interface Control Documents (ICDs), lack of Integrated Product Teams (IPTs) or wrong expertise on an IPT, lack of stakeholder analysis, lack of stakeholder involvement, lack of cross-functional teams, long iteration cycles, software patching, system modeling (too simplistic), top-down development, V-model, and Waterfall model.

**Table 5. Summary of the MPTs identified for each question**

| Practice | Total* | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|---|
| Rapid Prototyping | 14 | 3 | 8 | 0 | 3 |
| Continuous Integration | 11 | 0 | 0 | 2 | 9 |
| Iterative / Incremental Development | 11 | 5 | 1 | 0 | 5 |
| Interface Control Document (ICD) | 8 | 1 | 1 | 0 | 6 |
| Incremental Commitment Model (ICM)[1] | 7 | 6 | 1 | 0 | 0 |
| Stakeholder Analysis | 6 | 1 | 5 | 0 | 0 |
| Sustainment Plan | 6 | 0 | 1 | 5 | 0 |
| Requirements Arbitration | 6 | 0 | 6 | 0 | 0 |
| Configuration Management (CM) | 5 | 1 | 0 | 4 | 0 |
| Scrum[2] | 5 | 3 | 1 | 1 | 0 |
| Requirements Impact Analysis | 5 | 3 | 2 | 0 | 0 |
| Separate Teams for Development & Sustainment | 5 | 0 | 0 | 5 | 0 |
| Requirements Traceability | 4 | 2 | 2 | 0 | 0 |
| System Modeling / System Modeling Language | 4 | 3 | 1 | 0 | 0 |

**Table 5. Continued**

| Practice | Total* | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|---|
| Trade Studies | 4 | 0 | 3 | 1 | 0 |
| Change Impact Analysis | 4 | 1 | 0 | 3 | 0 |
| Integrated Product Team (IPT) | 3 | 2 | 0 | 0 | 1 |
| Model-Based Testing (MBT)[4] | 3 | 0 | 1 | 0 | 2 |
| Modeling and Simulation | 3 | 1 | 0 | 1 | 1 |
| SOA | 3 | 1 | 0 | 0 | 2 |
| User Stories | 3 | 1 | 0 | 2 | 0 |
| Stakeholder Interviews | 3 | 0 | 2 | 0 | 1 |
| Block Upgrades | 2 | 1 | 0 | 0 | 1 |
| Cost As an Independent Variable (CAIV) | 2 | 2 | 0 | 0 | 0 |
| Change Control Board | 2 | 0 | 0 | 1 | 1 |
| Customer Co-location | 2 | 1 | 0 | 0 | 1 |
| Feature-Driven Development (FDD)[3] | 2 | 2 | 0 | 0 | 0 |
| Focus groups | 2 | 0 | 2 | 0 | 0 |
| Lean | 2 | 2 | 0 | 0 | 0 |
| Open Standards | 2 | 2 | 0 | 0 | 0 |
| Requirements Volatility Analysis | 2 | 2 | 0 | 0 | 0 |
| Retain Key Engineers in Sustainment | 2 | 0 | 0 | 2 | 0 |
| Reviews (small team) | 2 | 0 | 0 | 1 | 1 |
| Self-generating Documentation | 2 | 0 | 0 | 1 | 1 |
| Stakeholder Roleplaying | 2 | 0 | 2 | 0 | 0 |
| Storyboarding | 2 | 1 | 1 | 0 | 0 |
| System Architect Role | 2 | 1 | 0 | 0 | 1 |
| WinWin | 2 | 1 | 1 | 0 | 0 |
| [63 practices] | 1 | | | | |
| **Total** | | **49** | **41** | **29** | **36** |

* Total may include practices cited by an individual in response to several questions (e.g. Tom suggests Scrum for Q3 Sustainment and Q4 Integration)

NOTES TO TABLE 5:

***Note 1:*** The Incremental Commitment Model (ICM) was also identified through the independent investigation of available MPTs. The ICM organizes systems engineering and acquisition processes in ways that better accommodate the different strengths and difficulties of hardware, software, and human factors engineering approaches.

***Note 2:*** Scrum was also identified through the independent investigation of available MPTs. Scrum adopts an empirical approach – accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly and respond to emerging requirements.

***Note 3:*** Feature Driven Development (FDD) was also identified through the independent investigation of available MPTs. FDD is a model-driven, iterative and incremental software development process with the purpose of delivering tangible, working software repeatedly in a timely manner.

***Note 4:*** Model Based Testing (MBT) was also identified through the independent investigation of available MPTs. MBT is a model based approach for the automatic generation of efficient test procedures/vectors using models of system requirements and specified functionality.

## B.2.6. Question 6: Critical Areas in Need of New MPTs

The respondents were asked to identify the three or four areas they believed to be the most critical areas in need of new MPTs. Of the 102 respondents for this question, the top five areas identified were: decision management; stakeholder requirements definition; requirements analysis; architecture design; and, life cycle model management. The least critical areas of those identified are: supply, transition, and disposal. Table 6 illustrates the results.

**Table 6. Critical areas identified as needing new MPTs**

| Answer Options | Response Percent | Response Count |
|---|---|---|
| Decision Management | 36.3% | 37 |
| Stakeholder Requirements Definition | 35.3% | 36 |
| Requirements Analysis | 32.4% | 33 |
| Architectural Design | 30.4% | 31 |
| Life Cycle Model Management | 27.5% | 28 |
| Risk Management | 24.5% | 25 |
| Integration | 23.5% | 24 |
| Acquisition | 19.6% | 20 |
| Project Planning | 18.6% | 19 |
| Project Assessment and Control | 18.6% | 19 |
| Verification | 17.6% | 18 |
| Measurement | 16.7% | 17 |
| Configuration Management | 15.7% | 16 |
| Validation | 15.7% | 16 |
| Project Portfolio Management | 11.8% | 12 |
| Implementation | 10.8% | 11 |
| Quality Management | 9.8% | 10 |
| Information Management | 8.8% | 9 |
| Infrastructure Management | 6.9% | 7 |
| Human Resource Management | 5.9% | 6 |
| Operation | 3.9% | 4 |
| Maintenance | 3.9% | 4 |
| Disposal | 2.0% | 2 |
| Supply | 1.0% | 1 |
| Transition | 1.0% | 1 |
| answered question | | 102 |
| skipped question | | 14 |

The team performed an in-depth qualitative analysis of the MPT survey responses of 97 engineers who responded to the survey as of May 30th, 2009. The responses were qualitatively analyzed to identify specific MPTs recommended by the survey respondents and the specific problems these MPTs addressed. The purpose of the analysis was twofold. First, validate the initial analysis of the survey responses presented in the previous sections. Second, identify the most common strategies (common themes among MPTs) for dealing with the aforementioned challenges. By identifying common strategies and themes, the team was able to identify separate MPTs that both address the Sponsor's challenges and are feasible in the Sponsor's operating environment.

## B.3. In-depth Qualitative Analysis – Identifying Common Themes among MPTs

Recommended MPTs and specific problems were identified that were associated with four challenge areas:

1. Requirements - changing priorities and emerging requirements
2. Stakeholder issues - obtaining useful stakeholder input and resolving conflicting stakeholder requirements
3. Sustainment - resolving conflicts between developing new capabilities and providing on-going support for operational systems
4. Integration - integrating independently evolving components into a larger system

To help identify candidate MPTs, the team grouped the MPTs for each challenge area into "themes" based on how an MPT addresses the challenge. For example, some MPTs are testing techniques while others describe how to assign personnel or how to design the system. The results of this initial analysis are shown in Table 7.

**Table 7. Counts of specific problems, MPTs and MPT themes from survey responses**

| Challenge area | Specific problems identified | MPTs identified* | MPT themes |
|---|---|---|---|
| Requirements | 22 | 89 | 17 |
| Stakeholder issues | 14 | 71 | 13 |
| Sustainment | 9 | 41 | 14 |
| Integration | 12 | 50 | 15 |

* Many survey responses listed non-specific MPTs, such as "good systems engineering" and "disciplined process" and these are included in the totals.

Unique MPTs identified for each challenge area were counted, though many MPTs were non-specific, such as "good systems engineering" and "disciplined process," and thus some overlap occurred. Some MPTs (e.g. Scrum) were suggested for addressing multiple challenge areas.

After creating a list of the MPT themes for each challenge area, the survey responses were distributed among five researchers. Each researcher was given a description of the MPT themes and was assigned to apply them to the survey responses for at least one challenge area. The researchers worked independently, and their application of the themes to the responses serves two purposes: 1) to validate that the themes are robust and well-defined; and 2) to independently assess the most common themes among the responses. Based on the researchers' application 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.

If more than one researcher applied the same MPT theme to the same response, those researchers "overlapped" in their reviews. If only one researcher applied the MPT theme to a response, they did not overlap. This section presents the five most-agreed-upon MPT themes applied in each of the challenge areas. A theme with high overlap suggests that the theme was well-defined and that the survey response was sufficiently specific in describing an MPT. A theme with high frequency of occurrence but low overlap suggests that the theme is poorly described or the response is subject to interpretation; such highly-occurring themes warrant further exploration.

### B.3.1. MPT themes for Requirements challenges

*"Are changing requirements priorities or emerging requirements a challenge in your product development environment?"*

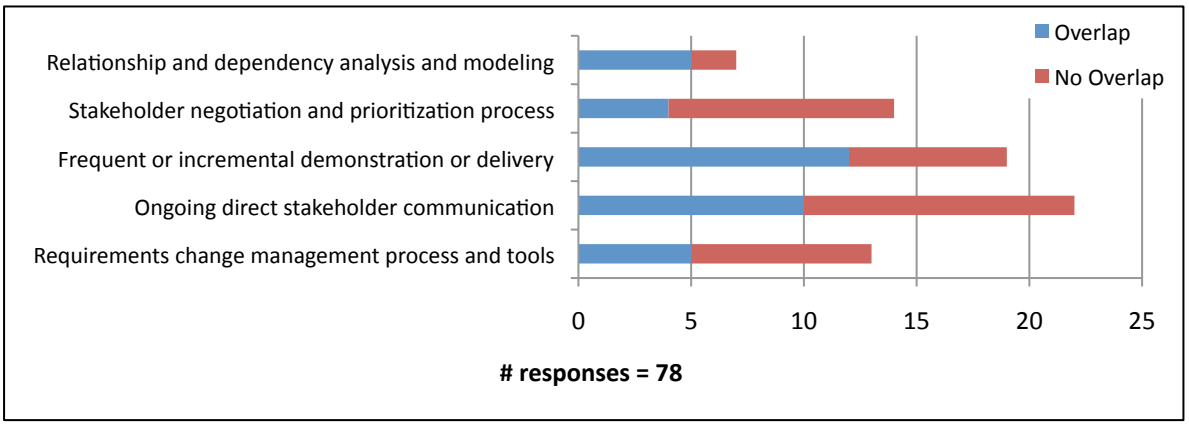


**Figure 6. Requirements MPT themes**

Relationship and dependency analysis and modeling – Using tools, techniques and languages that can model systems (including requirements, code and architecture) 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.

Requirements change management processes and tools – 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.

Ongoing direct stakeholder communication – Soliciting feedback from customers and/or stakeholders using either direct communication or collaborative tools throughout the development process. 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.

Stakeholder negotiation and 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 or incremental demonstration or delivery – Periodically demonstrating or delivering 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. Furthermore, incremental demonstration improves the quality of customer feedback by making them aware of development progress.

## B.3.2. MPT themes for Stakeholder challenges

*"Is getting useful stakeholder input, or dealing with conflicting stakeholder requirements, a challenge in your product development environment?"*

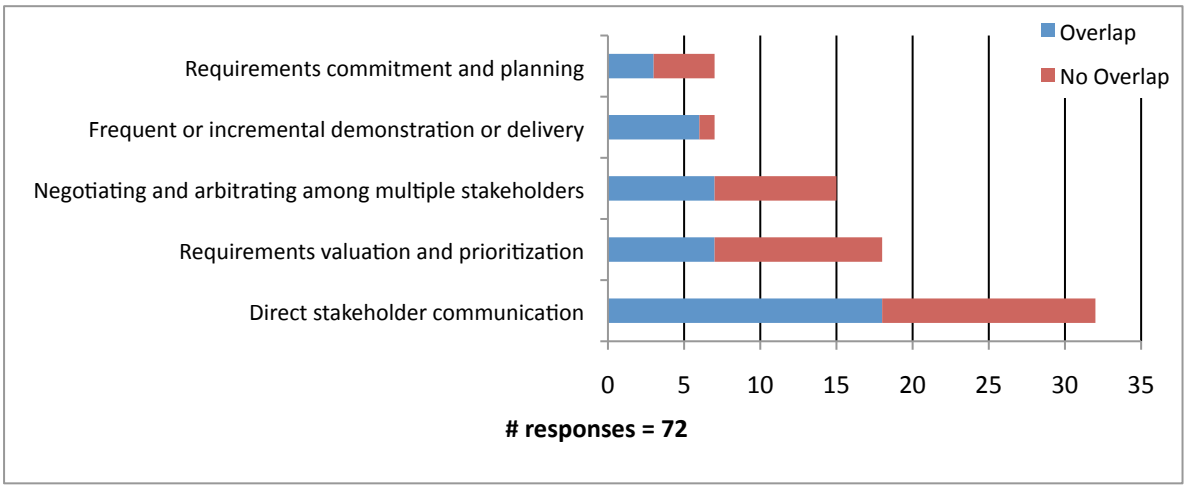


**Figure 7. Stakeholder MPT themes**

Requirements commitment and planning – 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.

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

Negotiating and arbitrating among multiple stakeholders – Involving multiple stakeholders in requirements prioritization and a defined process for deciding on requirements priority. This theme is similar to "stakeholder negotiation and prioritization process" in Section B.3.1 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.

Direct stakeholder communication – Same as in Section B.3.1.

Frequent or incremental demonstration or delivery – Same as in Section B.3.1.

## B.3.3. MPT themes for Sustainment challenges

*"Is resolving conflicts between developing new capabilities and supporting the currently released system a challenge in your product development environment?"*

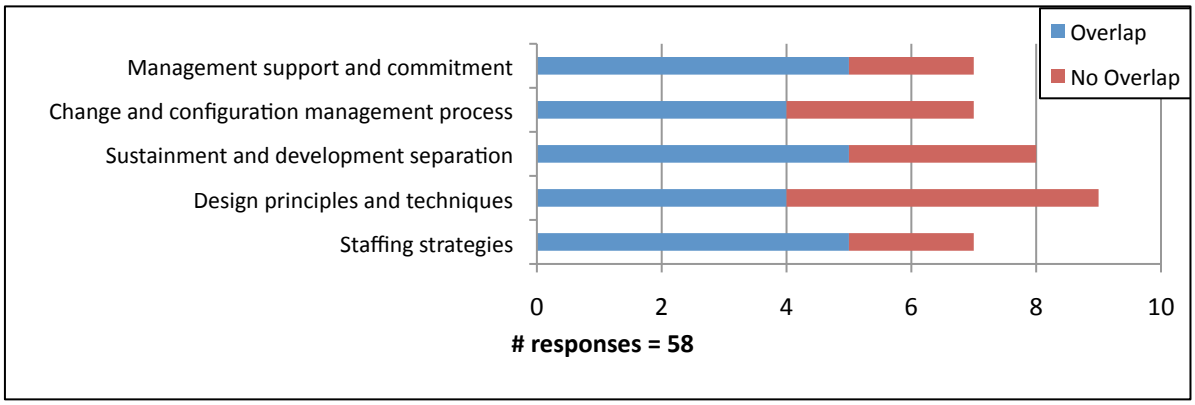


**Figure 8. Sustainment MPT themes**

Change and configuration management process – Having a process that dictates how project artifacts (including requirements and code) are changed, which 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 assessment of their impact. Configuration management creates separate source code branches for co-development and creates rollbacks in case of error.

Management support and commitment – Management buy-in and support for development activities and decisions made during the process. This theme primarily involves making sure that management understands the costs of

maintenance/sustainment activities and that these costs cannot be ignored when planning new development activities. Management support is critical for stakeholders to understand and to respect the cost of sustainment activities.

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.

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 added manpower and additional communication overhead.

## B.3.4. MPT themes for Integration challenges

*"Is integrating independently evolving components into a larger system a challenge in your product development environment?"*

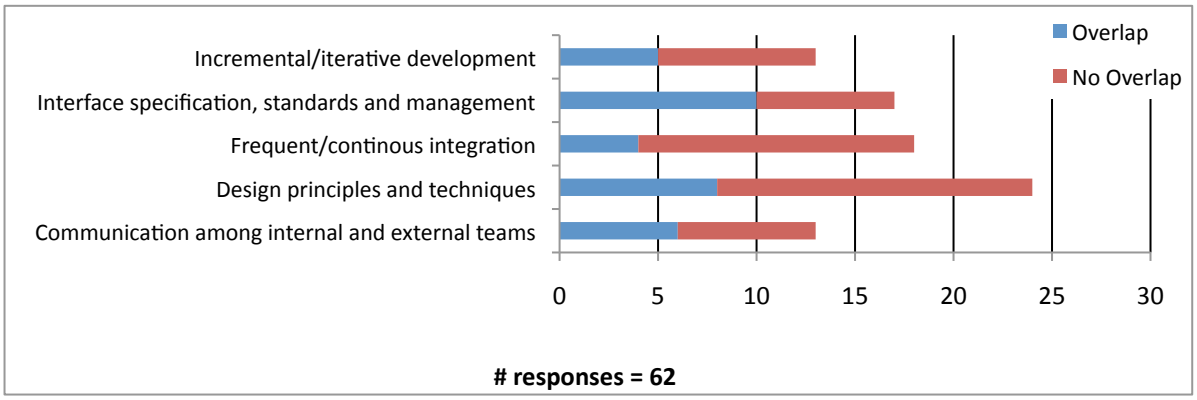


**Figure 9. Integration MPT themes**

Communication among internal and external teams – 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 specification, standards and management – 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 individually-developed code or sub-components into a larger system. Regularly integrating components enforces interface standards and periodically verifies interoperability rather than surface interoperability issues in one batch prior to delivery. Continuous integration is supported by automated testing and an automated build process.

Incremental / iterative development – Development cycles (including planning, implementation and verification) occur in increments rather than as one large 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.

Design principles and techniques – Same as in Section B.3.3.

## B.4. PARETO ANALYSIS

A Pareto analysis is an organizational methodology to look at possible causes of a problem. It follows the idea that 80% of the problems are caused by a few key factors (20%). It organizes the information in such a way that it can be quickly seen what factors are important as well as what factors can initially not be addressed. This analysis did not show the list of factors for each category that summed to 80% but rather a ranking of factors to see where initial efforts might be placed.

The information supplied in the distributed coding analysis was further evaluated. The category of stakeholder had three raters that were allowed up to 5 sub-categories for the answers supplied in the survey to be linked to. The sub-categories were:

- direct stakeholder communication
- requirements valuation and prioritization
- negotiating and arbitrating among multiple stakeholders
- requirements elicitation and definition
- frequent or incremental demonstration or delivery
- architecture and interface specification, standards, and management
- requirements commitment and planning
- validation activities
- early analysis of non-functional requirements and constraints
- requirements change management process and tools

- communication techniques for non-technical stakeholders
- relationship and dependency analysis and modeling
- prototypes and mockups

The coding results in which all three raters agreed on a sub-category are presented in Figure 10. Stakeholder coding results. For each sub-category, the number of times that all three raters agreed on that particular sub-category was divided by the total number of times all three raters agreed. For example, there were 19 instances where all three rates agreed. Five of those 19 were in the sub-category of direct stakeholder communication yielding 26% of the total. The results show that direct stakeholder communication was most frequently agreed upon rater coding followed by requirements valuation and prioritization. This suggests a possible list of themes to address first in the category of stakeholder.

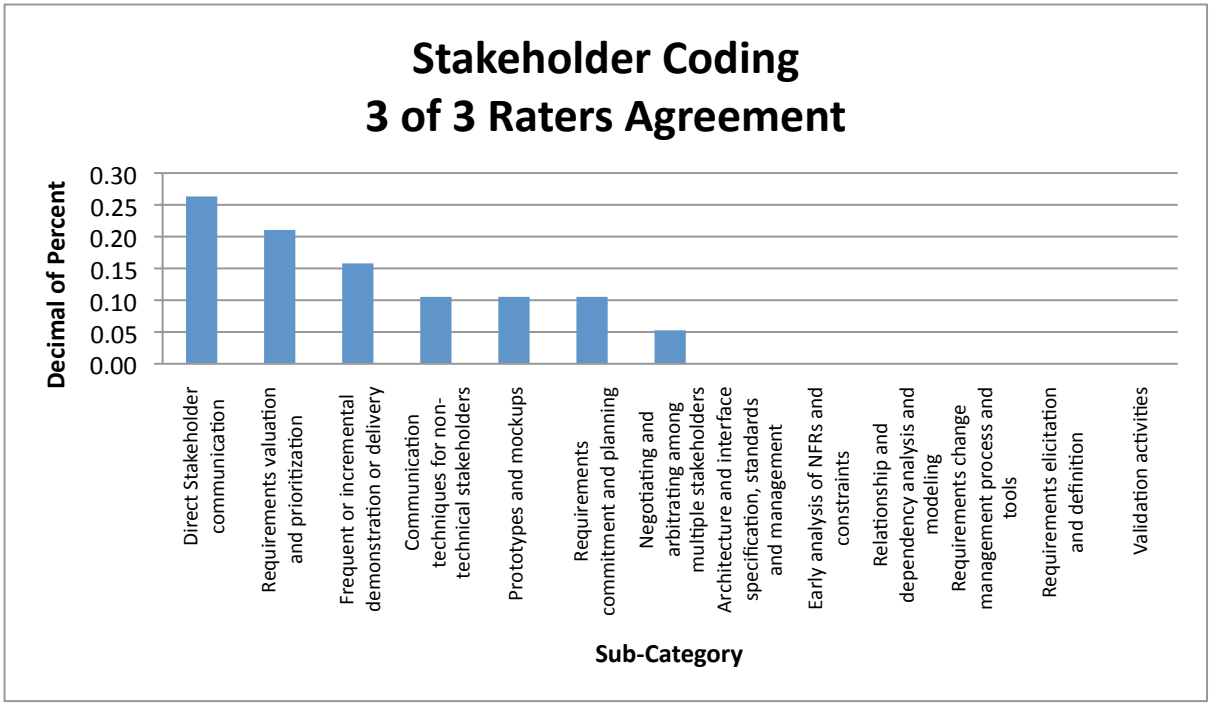


**Figure 10. Stakeholder coding results**

- This type of analysis was repeated for the reaming four categories but each of these had only two rates performing the coding therefore this counted the number of times of agreement. Integration contained the sub-categories of:
- communication among internal and external teams
- decision rules and process ownership
- design principles and techniques
- frequent/continuous integration
- prototypes and mockups
- relationship and dependency analysis and modeling

- code functionality ownership
- interface specifications, standards, and management
- incremental/iterative development
- testing techniques
- reviews
- change and configuration management process
- staffing strategies
- self-generation documentation
- scheduling and prioritizing integration tasks with development

The results for integration are presented in Figure 11. Integration coding results. The results show the top sub-category to be interface specifications, standards, and management followed by design principles and techniques.

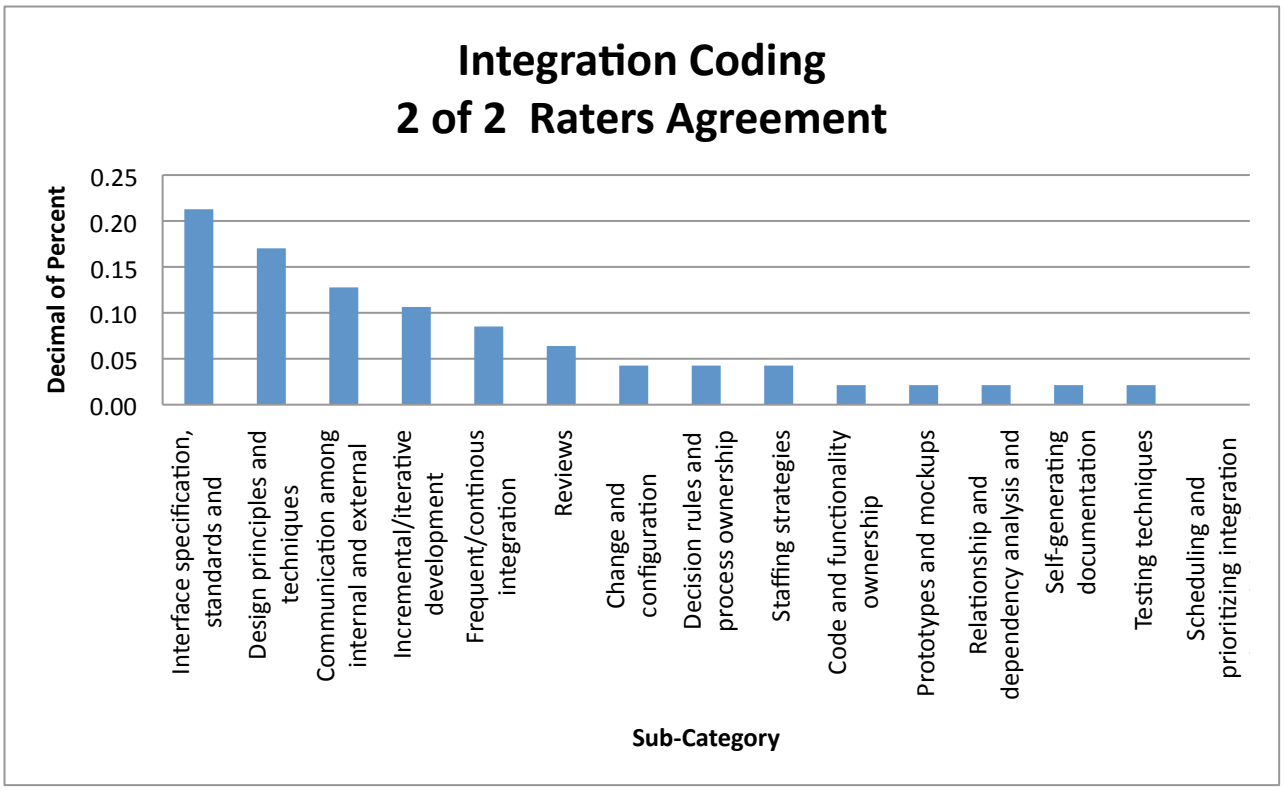


**Figure 11. Integration coding results**

Sustainment contained the sub-categories of:

- scheduling and prioritizing maintenance tasks with development
- staffing strategies
- design principles and techniques
- sustainment and development separation
- decision rules and process ownership

- tracking defect data
- change and configuration management process
- management support and commitment
- direct customer and stakeholder feedback
- frequent/continuous integration
- testing techniques
- relationship and dependency analysis and modeling
- self-generating documentation
- reviews

The results for sustainment are presented in Figure 12. Sustainment coding results. The results show the top sub-category to be a tie between management support and commitment, staffing strategies, and sustainment and development separation.

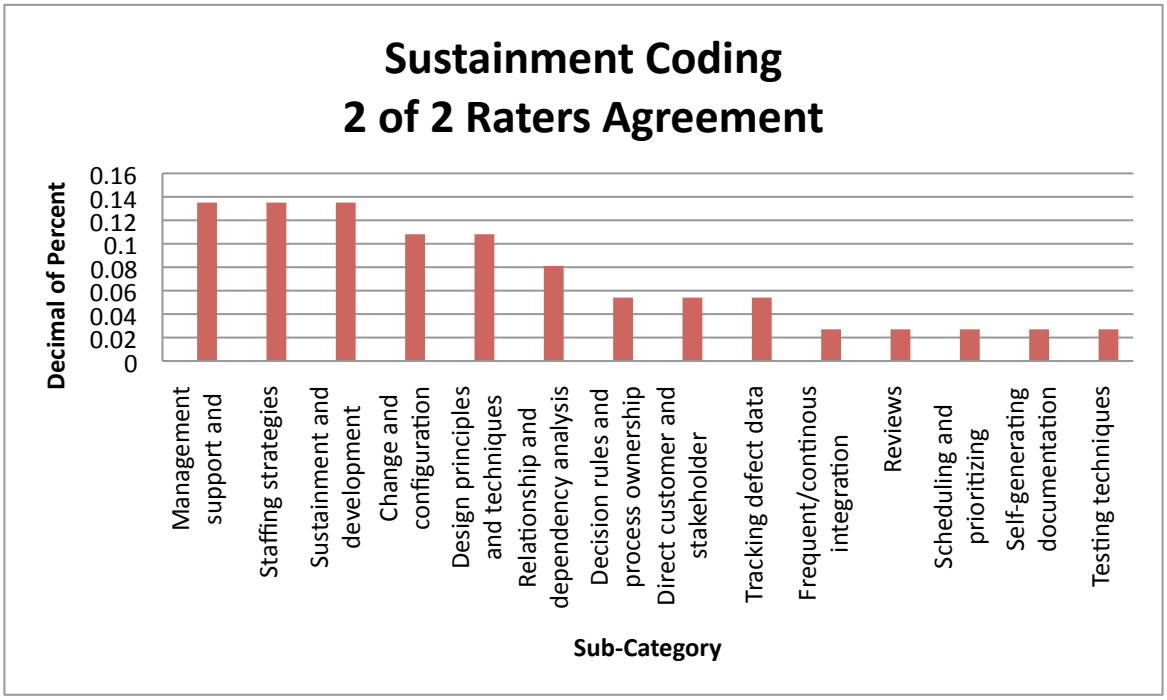


**Figure 12. Sustainment coding results**

Requirements contained the sub-categories of:

- requirements change management process and tools
- ongoing direct stakeholder communication
- requirements elicitation and definition
- open or industry standard tools and methods
- frequent or incremental demonstration or delivery

- lifecycle process & methodologies
- stakeholder negotiation and prioritization process
- domain and project knowledge management
- relationship and dependency analysis and modeling
- validation activities
- early analysis of non-functional requirements and constraints
- reviews
- commitment to a plan and deliverables
- risk management
- skilled team
- requirements reuse

The results for sustainment are presented in Figure 13. Requirements coding results. The results show the top sub-category to be frequent or incremental demonstration or delivery followed by ongoing direct stakeholder communication.

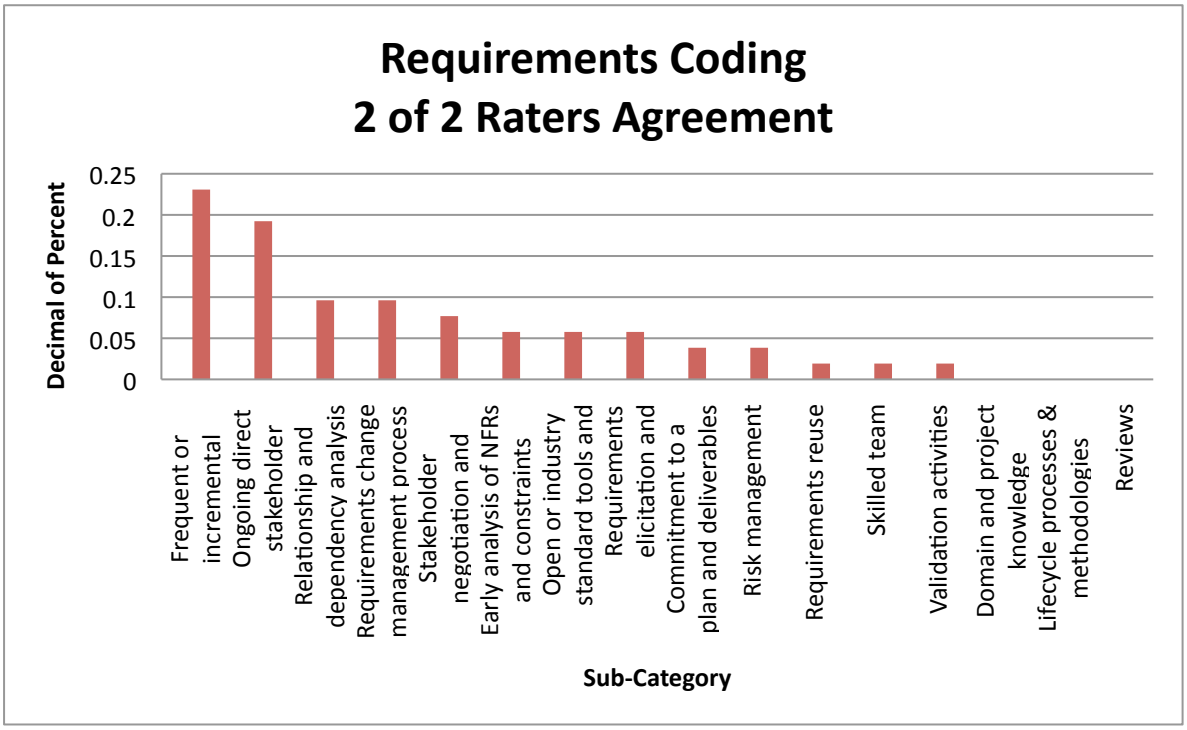


**Figure 13. Requirements coding results**

This analysis presents a suggested list of sub-categories that may be used for further research. It should be noted that in the list there is some overlap in the sub-categories:

- direct stakeholder communication
- requirements valuation and prioritization
- interface specifications, standards, and management

- design principles and techniques
- management support and commitment
- staffing strategies
- sustainment and development separation
- frequent or incremental demonstration or delivery
- ongoing direct stakeholder communication

## B.5. QUALITATIVE ANALYSIS SUMMARY

It is interesting to note that some themes discussed above are contradictory or at the very least difficult to imagine implemented in the same environment. For example, under the issue of "sustainment," some respondents advocated staffing strategies that allowed maintainers to get experience with a system during development, while other respondents advocated a strict separation of sustainment and development teams in order to better manage resources.

Similarly, many of the themes are inter-related. In particular, there is a considerable amount of entanglement between requirements and stakeholder issues. Many of the respondents who mentioned one of the issues described here also reported many of the other issues across both of these categories.

## B.6. MPT REFERENCES

The following are collected references used by the team in the survey analysis task. This is not a complete bibliography on the MPTs; however, it provides pointers to additional information the team used to gain initial understanding of MPTs mentioned by respondents.

Beck, K., *Extreme Programming Explained*, Addison-Wesley, Boston, 2004.

Boehm, B. and J. Lane (2007). "Using the Incremental Commitment Model to Integrate System Acquisition, Systems Engineering, and Software Engineering," CrossTalk, October 2007.

Boehm, B., J. Lane, and A. Pyster (2007). "Final Report: Integrating Systems and Software Engineering Project,"
http://csse.usc.edu/events/2008/ARR/pages/material.html

Boehm, B., and R. Turner (2003). *Balancing Agility and Discipline: A Guide for the Perplexed*, Addison-Wesley, Boston, 2003.

Cohn, Mike "Rolling Lookahead Planning,"
http://www.mountaingoatsoftware.com/articles/41-rolling-lookahead-planning as of May 13, 2009

http://agileproductdesign.com/useful_papers/rite_method.pdf

http://en.wikipedia.org/wiki/Configuration_management, retrieved on 2009-06-18.

http://en.wikipedia.org/wiki/Feature_Driven_Development, as of May 13, 2009

http://en.wikipedia.org/wiki/Feature_Driven_Development, as of May 13, 2009

http://en.wikipedia.org/wiki/Focus_group, retrieved on 2009-06-17.

http://en.wikipedia.org/wiki/Interface_Control_Document, retrieved on 2009-06-18.

http://en.wikipedia.org/wiki/Iterative_design as of May 13, 2009.

http://en.wikipedia.org/wiki/Iterative_design as of May 13, 2009.

http://en.wikipedia.org/wiki/Iterative_development, as of May 13, 2009

http://en.wikipedia.org/wiki/Iterative_development, as of May 13, 2009

http://en.wikipedia.org/wiki/Requirements_Traceability, retrieved on 2009-06-17.

http://en.wikipedia.org/wiki/Scrum_(development) , as of May 13, 2009

http://en.wikipedia.org/wiki/Stakeholder_analysis, retrieved on 2009-06-17.

http://en.wikipedia.org/wiki/Systems_Modeling_Language, retrieved on 2009-06-17.

http://en.wikipedia.org/wiki/User_story, retrieved on 2009-06-18.

http://en.wikipedia.org/wiki/Wizard_of_Oz_experiment, retrieved on 2009-09-20.

http://www.goldpractices.com/practices/mbt/index.php

http://www.martinfowler.com/articles/continuousIntegration.html, retrieved on 2009-06-18.

http://www.sysmlforum.com/FAQ.htm, retrieved on 2009-06-17.

M. C. Medlock, D. Wixon, M Federoff. Rapid Iterative Testing and Evaluation: RITE. Powerpoint ppt. http://www.slideshare.net/macieklipiec/rapid-iterative-testing-and-evaluation

M. C. Medlock, D. Wixon, M. Terrano, R. Romero, B. Fulton (2002). “Using the RITE method to improve products; a definition and a case study.” Usability Professionals Association, Orlando FL July 2002.
http://www.microsoft.com/downloads/details.aspx?FamilyID=3b882eb1-5f06-41d9-baba-d39ad13bc3ff&displaylang=en

Williams, Laurie and Robert Kessler, *Pair Programming Illuminated*, Addison Wesley Professional, Boston, 2002.