Feasibility of a rapid systems engineering framework: an exploratory study

Rashmi Jain*, Anithashree Chandrasekaran, Lymari Castro and Mary VanLeer

School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ 07030, USA
Fax: +1 201 216 5541
E-mail: Rashmi.Jain@stevens.edu
E-mail: anithashree.chandrasekaran@stevens.edu
E-mail: lcastro@stevens.edu
E-mail: mvanleer@stevens.edu
*Corresponding author

Abstract: Systems Engineering (SE) has been traditionally viewed as an extremely rigorous approach for having resource intensive processes often perceived to involve bureaucratic decision making; therefore, deemed to be affordable only by large companies and government institutions. However, in recent years, the commercial industry is trying to leverage the benefits of SE by selecting the relevant aspects that apply to them and customising SE to a leaner and compressed version. This paper explores the suitability and the feasibility of rapid approaches and techniques to the existing SE processes. More specifically, this paper describes 22 techniques of rapid systems engineering (RSE) during the design and implementation processes, relevant to those processes and lessons learned. These techniques are applied to the 14 SE processes as illustrated by the SE standard – ISO 15288. An exploratory survey was developed by the authors based on the SE process activities as per ISO 15288 to evaluate application utilising a rapid approach in current projects from various industries. This paper provides conclusions to applying rapid techniques to the SE processes based on existing literature and the experiences of the projects surveyed. It concludes with a discussion of potential research projects for the evaluation of RSE.

Keywords: RSE; rapid systems engineering; systems integration; validation; verification; life cycle management; ISO 15288; risk management; COTS integration; requirements; architecture; configuration management; system design and architecture.


Biographical notes: Rashmi Jain is an Associate Professor of Systems Engineering at Stevens Institute of Technology. She has over 15 years of experience in Information Technology (IT) systems. Over the course of her career, she has been involved in leading the implementation of large and complex systems engineering and integration projects. She has done lectures internationally at Keio University, Overseas Chinese Institute of Technology (OCIT), Indian Institute of Technology – Delhi. Her teaching and research
interests include systems integration, systems architecture and design, business process reengineering and rapid systems engineering. She holds PhD and MS in Technology Management from Stevens Institute of Technology.

Anithashree Chandrasekaran is a Doctoral Candidate in the School of Systems and Enterprise at Stevens Institute of Technology. Her research interests include rapid systems development and its processes, development process reengineering, risk management and modelling, system integration and system design and architecture. Currently, she is working on developing a dynamic risk assessment model for rapid system development process. She obtained her BE in Electrical and Electronics Engineering from P.S.G. College of Technology, India. She obtained her MS in Systems Engineering from Stevens Institute of Technology. She is the President of Stevens International Council of Systems Engineering student chapter.

Lymari Castro is a Doctoral Candidate in the School of Systems and Enterprise at Stevens Institute of Technology. She received a MEng in Engineering Physics from Cornell University and a BS in Theoretical Physics from the University of Puerto Rico. She is a Lead Systems Engineer for the Department of Defense and previously worked as a Systems Engineer at Raytheon Company. Current responsibilities involve testing and integration of rapid development projects. She is a Member of the Institute of Electrical and Electronics Engineers (IEEE) and the International Council of Systems Engineering (INCOSE). Her research interests are rapid systems development and systems integration.

Mary VanLeer has over 28 years of experience in defining, designing and supporting IT solutions for high-end data centre environments. At International Game Technology, She held the position of Director, Engineering Operations, where she brought together the centralised functions of engineering and created a centre of excellence in quality management, project management and systems engineering. At Sun Microsystems, she held the position of Director, Software Engineering in both the Chief Technologist’s Office and the Enterprise Systems Software Organization. In those positions, she introduced new verification methods to test the robustness of the products’ system recovery capabilities and led numerous initiatives in the quality efforts for diagnosability, serviceability and fault management.

1 Introduction

Leadership in today’s markets is acquired and sustained through innovative products and services that appear to be available simply to the consumer at will and at a price point suitable for their use (Golovatchev and Budde, 2007; Lee et al., 2005; Onuh et al., 2007). The pressures on suppliers in healthcare, retail, communications, automotive (Sumantran, 2005) and networked services are relentless and capable of driving any particular supplier to leadership or bankruptcy within a single product life cycle. In contrast, the Industrial Revolution provided opportunities for entrenchment in transportation, construction and raw materials where time scales for change and delivery of new products to market were measured in decades. In today’s knowledge, economy time scales are measured in months (Dereli and Baykasoglu, 2008; Ekman and Angwin, 2007; Onuh et al., 2007) or more rarely in a few years, leaving companies unable to respond at this pace and merely
surviving (Acharya and Mahanty, 2008; Jamali and Keshishian, 2006). The challenge for
these companies to apply the rigour and methodology of systems engineering (SE) and
yet be able to meet the time-to-market targets, creates an opportunity to research and
introduce variations of best practices of the SE disciplines for a rapid environment.

Perceptions exist that there are only two choices to implement SE (Eisenhardt and
Tabrizi, 1995):

1. implement structured, disciplined and well planned throughout the entire life cycle
   then evaluate opportunities to compress the activities

2. create an environment that utilizes an intuition experience rather than planning.

These options provide challenges in rapidly changing environments where companies
must address the three sides of the ‘iron triangle’ (Ambler, 2008) of cost, quality (Savino
et al., 2007), and schedule with equal consideration (Kim, 2007; Rizzi and Regazzoni,
2007). Given the situation, companies should examine what is relevant for their business
environment, and then implement the appropriate SE activities with approaches that
allow them to achieve the desired targeted cycle times. In Section 2, this paper introduces
the definition of rapid systems engineering (RSE) and some common success factors
associated with RSE. In Section 3, this paper introduces research and ideas that are
relevant to this discussion. The authors have based their views of the challenges on an
examination of applied rapid techniques to the SE process defined in ISO 15288. ISO
15288 was used as the baseline process framework since it offers a comprehensive view
of the processes key to successful SE. In Figure 1, ISO 15288 is presented showing the
use of an iterative methodology for the technical processes and the infrastructure and
enabling processes, which are foundational to the project across the bottom. On either
side of the iterative diagram are the processes focused on key areas of acquisition, supply
chain management and deployment. Given this as an initial focused research on the
applicability of rapid techniques for SE, the authors selected only a set of key processes
to evaluate shown as the shaded boxes in Figure 1. This provides a starting point to
courage future research in the application of rapid techniques to these and the
remaining processes.

In section 4, the authors introduce the next level of research through the creation and
administration of a focused survey applicable to the process techniques presented in
Section 3. The survey was administered to individuals responsible for the SE processes in
both the commercial and government industry segments. The survey asked the
participants to evaluate the importance of these techniques, as they are applicable to RSE.
Based on the responses, Section 5 provides an overview of the authors’ analysis of
the data. Sections 6 and 7 provide the authors’ implications associated with this research
and concluding statements including potential research topics to expand the capabilities
of RSE.
2 What is rapid systems engineering?

While RSE is an evolving term that will mature in scope as SE is implemented in rapid environments, the authors have chosen to define RSE as: adopting methodologies, tools and techniques that can introduce rapidity into the SE process while optimising the success factors during the development of projects. The success factors are specific to the system-of-interest being developed, and depend on the type of system, its related product line and/or platform, organisation structure and processes, and market forces such as competition and window of opportunity, innovation and customers. Some of the common success factors across projects are return-on-investment (ROI) (Oke et al., 2008), cost of ownership, time to market (Muthiah and Huang, 2008), productivity (Muthiah and Huang, 2006) quality (Amuthakkannan et al., 2008) and customer satisfaction. Another way to view RSE is as a set of tools, methodologies and people and management techniques that results in a SE life cycle that reduces the time to market a product, from concept to implementation, without sacrificing the quality of products.

3 Research on rapidity of key processes

In order to identify the scope of SE processes that apply and may be described as being most relevant for implementing rapidity to any development project, the authors selected 14 processes (as illustrated in the shaded boxes in Figure 1), sub-processes and activities from ISO 15288. This section discusses the scope of these SE activities as it applies to rapid development projects. The discussion is based on extensive doctoral-level research on existing literature, provisions of the ISO 15288 standard and the experience of the authors in successfully implementing rapidity of the SE processes on several projects.
3.1 Life cycle model management

Rapidity in SE has to be addressed throughout the life cycle of a system in order to optimise its impact and benefits. A basic pre-requisite for addressing rapidity is that the life cycle processes support smaller and incremental (Conforto and Amaral, 2008; Larman, 2004; Shore and Warden, 2008) development and frequent releases. This technique was originally introduced in software development to respond to demands of shorter time scales at a lower cost while maintaining high-quality standards (Cockburn, 2006; Highsmith, 2004). Although large formal enterprise releases could run as long as 12–18 months, this does not preclude a release cycle of small iterations in the 1–2 week time frame. In the first evolution of rapid development, the initial focus is placed on breaking the project into smaller pieces, the second evolution overlaps in a phased approach and the third evolution shortens the phases into an iteration that is focused on value (Cottmeyer, 2008).

3.2 Human resource management

In order to build a product within the shortened time frames, the organisation of the team and the utilisation of skills is critical (Butcher, 2007; Cockburn, 2006; Alssamadisy, 2008). There are three main functions recommended for a team involved in rapid development activities: the product owner or customer proxy, development team and quality assurance (Alssamadisy, 2008). Customer involvement or a customer proxy is critical to the success of the team in a rapid environment by providing continuous feedback on feature/functionality of the product under development (Shore and Warden, 2008). The development team creates the product according to customer requirements and their feedback. Quality assurance then tests the product (Leffingwell, 2007) to ensure compliance to customer needs and specifications.

Rapid techniques encourage close collaboration among team members creating a shared ownership in the project. As a team, they look for improvements to streamline the process and implement appropriate changes to ensure success. Teams should be made up of a mix of senior and junior people who are risk takers, results oriented and willing to receive feedback on design and delivery. Throughout the life cycle process, there is an evaluation of the skills and competencies necessary to complete the iteration. Within the iteration, everyone is given an opportunity to learn new technologies and become proficient in different areas; however, there may be a point when no one in the team has the particular skill set (Gero and Kannengiesser, 2004). As in any major project, planning and budgeting should allow necessary training to ensure that the skill levels are raised to meet the demands.

3.3 Project planning and project assessment

ISO 15288 presents project planning and project assessment as separate processes. For this paper, the authors combined these processes in their research and evaluation. By definition, rapid techniques employ an iterative and incremental process providing customer value. Traditional project plans would start with a waterfall approach and assume that everything was stable from the requirements input, the team members allocated, to the level of executive support. When the project was finally released, it would meet the needs of all the stakeholders. Unfortunately, these conditions are rare in
today’s development environment. Requirements change based on the customer input, environmental changes and executive support. Team members may come and go based on the skill levels and the business needs. Executive support can change based on new business challenges. These realities often create the conditions for project cost and schedule overruns.

The challenge for project managers in today’s environment is to anticipate the changes in conditions and then create plans that strive for predictability within shorter cycles (Cohn, 2006). Iterative and incremental development are not simply a review of overlapping waterfall. Instead, it is designed to focus on delivery of value added functionality by having all the necessary skill sets focused and working in parallel at all times.

At the beginning of the project, a kick-off meeting is held between the product management team and the development team to create the shared vision of the project and develop a project schedule and resource allocation plan divided into a series of iterations defining a sequence of project releases. Every new iteration starts with a planning session which reviews the total feature set, prioritises the feature set, establishes the schedule, identifies the resources, organises the work and gains agreement from the team on deliverables (Leffingwell, 2007). Daily meetings or ‘stand ups’ provide the communication mechanism and keep track of the completed progress. The end of the iteration provides the team the opportunity to demonstrate their success with completed features and a forum for retrospective, which encourages the team to reflect and make the necessary changes to the process, making the next iteration more efficient. The project manager becomes an integrator of the team’s knowledge and the project management processes throughout the project life cycle (Morandotti, 2007).

3.4 Risk management

Risk is a function of the anticipated frequency of occurrence of an undesired event, the potential severity of resulting consequences and the uncertainties associated with the frequency and severity (NASA-STD-8719. 13A, 1997). Risk is a combination of an abnormal event or failure and the consequences of that event or failure to a system’s operators, users or environment (McManus, 2001). Risk management is the practice of identifying, analysing, assessing and acting to eliminate or mitigate risk (Stein et al., 2002). The risk management paradigm shown in Figure 2 (Carr et al., 1993) provides emphasis on the iterative nature of the risk management processes.

Figure 2 Risk management paradigm (see online version for colours)

Determining an initial set of known risks and identifying relevant check points throughout the RSE project aids in faster and continuous risk management. Identifying when and where the information about a risk factor becomes critical in SE process helps in doing risk analysis more frequently and in avoiding negligence due to lack of risk analysis (Conrow and Shishido, 1997). Identifying the initial set of known risks upfront and focusing on gathering information on unknown risks increases the risk awareness and saves time throughout the process. As unknown risks emerge are identified, they can be added to knowledge management and then applied to other RSE efforts.

Risk management cannot be performed without measurement. The depth (what extent of accuracy) and the width (how many metrics) of measurement determine how much effort and resources are required to perform risk analysis. In RSE, the role of risk management is to guide in continuous decision making (Schroeder, 2000). The objective is to identify the minimum information that is required to make the best decision (Barki et al., 2001; Browning et al., 2002). The required level of information to make a SE decision can be obtained by using relative or indirect measurement. By doing so, the time taken to estimate measurements will be significantly reduced and still provide adequate information to make good decisions. For example, within the process, there is a need to understand the task completion time to develop a module of software. One approach would be to expend time and effort in accurately analysing and estimating the amount of time to complete. Instead, by using relative or indirect measurements, tasks of equal size and complexity which have been completed are used as the input into the estimation process. This approach provides the estimate much faster with less detailed analysis significantly reducing the time it takes to provide exact measurements.

3.5 Configuration management

Configuration Management (CM) is the process of controlling the evolution of families of systems and provides techniques, methods and procedures to maintain a product history, to uniquely identify and locate each version of a system, and to initiate, evaluate and control change to the product during development and after release (Feiler and Dart, 1990; Mette and Hass, 2003; Stevens et al., 1998). The primary objective of CM is to ensure effective management of the evolving configuration of a system, both hardware and software, during its life cycle (INCOSE, 2006). CM effectiveness can be measured in terms of consistency, stability and traceability (Stevens et al., 1998).

Rapidity in CM can be achieved in a number of ways, from establishing configuration controls to the choice of tools and automation. The main focus of CM, which is also highly critical for RSE, is traceability. Systems need to adapt to the rapid change driven by the global dynamic marketplace, technological evolution and variety of environments (Feiler and Dart, 1990; Fricke and Schulz, 2005). Traceability of changes is critical in controlling and managing their impact on the SE life cycle (Jain et al., 2008, 2009). Two such common ways of establishing traceability are maintaining functional and physical relationships and tracking attributes of configuration items (Mette and Hass, 2003). They are also the most important among the essential activities in CM for RSE.

Relating and maintaining functions and services to subsystems and components results in better change control and configuration management. When there are changes in services and functions due to the changing customer or market needs, the established traceability between services and functions helps in identifying the right configuration items that are impacted by the changes. The study of the impact of changes based on the
CM dependency mappings assists in making informed decision about the feasibility and required level of effort, and thereby the value of the change. This technique enables and supports the successful estimation and prioritisation activities within the project. A configuration item can have any number of attributes associated with it based on the level of detail. Identifying and tracking these attributes are important but also time consuming. The one approach is to optimise the time spent on identifying and tracking the attributes. Service level objectives are attributes that provide the most valuable information and should be tracked within the CM process for RSE.

3.6 Requirements and requirements analysis

The time necessary to identify, define and analyse the stakeholder’s requirements and transform them into system requirements can be significantly reduced by introducing RSE practices in the requirements process. RSE will reduce the time to complete the requirements definition and analysis process by providing flexibility in addressing unidentified or changing stakeholder’s requirements.

Since RSE uses an iterative and incremental development approach, the multiple iterations offer rapid feedback on the progress of the project and the functionalities of the system (Callahan and Moretton, 2001). Frequent interactions with the stakeholders enable rapidity during the requirements definition and analysis process because SE can obtain early feedback from the stakeholder and manage new, unidentified or changing requirements. Continuous stakeholder feedback is an essential element of RSE (De Beer et al., 2009). When using a linear-sequential approach, the project is at risk of taking too long to complete and not meeting the stakeholder needs at the time of delivery, because the original requirements or business needs have fundamentally changed (Maner, 1997). RSE overcomes this risk by obtaining stakeholder feedback at the end of the iteration and by allowing the stakeholder to prioritise the desired functionalities or requirements at the beginning of the next iteration. This ensures that the delivered product satisfies customer needs and reduces the cost associated with rework after the product has been delivered. RSE enforces customer involvement throughout the projects’ life cycle to ensure requirements volatility is managed throughout the project. This can be accomplished by collocating teams. When collocation of the whole team is not easily achieved, then stakeholders’ representatives are collocated with the engineering team to ensure the stakeholder needs are addressed in the iteration and that any misunderstandings about the system requirements are resolved early in the project’s lifecycle (Teasley et al., 2002a).

3.7 Architectural design

In RSE, projects that utilise familiar architectures or small improvements to the architecture, less integration for new technologies or the introduction to the market with familiar technologies can move through the process quickly. Projects that require novel technologies or innovative architectures can introduce uncertainty which may require experimentation, thus creating longer development times (Meyer and Utterback, 1995). To lessen the risk created by novel technologies, rapid prototyping can be used (Andrews and Goeddel, 1994; De Beer et al., 2009). Time boxed research spikes can be used to determine architecture novelty as long as they do not distract the team from completing the project (Isham, 2008).
In addition to prototyping, selection of less complex architectures can speed the development times. Some projects have found that by trading off technology novelty for greater complexity allows for faster time to market (Tatikonda and Rosenthal, 2000). When possible, postpone architectural decisions until the ‘last responsible moment’ allowing an interleaving of development with design creating a just in time architectural process (Krunic, 2007).

3.8 Implementation

In RSE, implementation focuses on developing systems in short time scales, at low cost, while maintaining high standards of quality (Woi Hin, 2006). RSE achieves rapidity in the implementation/development process by allowing the customer to identify and prioritise the functionalities that add more value to the system at the start of the iteration. The basic premise of this model is that it is easier to add capability to a small well-developed product than to build a complete application at once. For customers, managers and developers, this method can offer a potential payoff in minimum defect injection and leakage rates (Constantine, 2001). A technique used originally in the software development process is Test Driven Development. It encourages the development and implementation processes to be driven in a structured methodology using simple designs and continuous delivery of highly reliable functionality (Beck, 2003).

3.9 Integration

Systems integration is an important element of SE, which involves the integration of hardware, software, products, services, business processes and human beings (Grady, 1994; Jain, 2007). The systems integration process is a set of activities that transforms the stakeholder’s requirements into an operational system by unifying the process components and product components into a whole while ensuring compliance to the specified levels of component operations and interoperability (Jain et al., 2010). The scope and objectives of systems integration should be clearly identified to ensure that new systems and components are able to work with the existing systems and components.

Within the process, integration requirements are gathered and analysed along with other requirements ensuring that everything supports the integration aspects of the system. This process supports traceability of requirements and verification that the requirements can be tested or qualified. Prior to the design phase, systems integration evaluations provide guidance to the design team(s) of any constraints and/or challenges for integration that may be determined through the analysis of modelling and simulation. During the design of the system, there are various assumptions about the systems that can have an impact on the integration aspects of the process. Utilising experts to validate the design decisions provides the checks and balances to reduce the possibility of integration issues later forcing a redesign when changes are both time consuming and costly. These approaches can save cost and effort on rework later by reducing the probability of redesign for developed or procured interfaces due to integration defects.

RSE promotes continuous integration whereby components and modules are physically or virtually integrated and tested as they are being developed. As the components and subsystems evolve over a period of time, changes are made to the overall system, where integration of these components or changes is necessary at a system level. A best practice is to perform integration (physically or virtually) as these
new developments or changes are made. Continuous integration increases quality of the product and reduces the development risks (Duvall et al., 2007; Holck and Jorgensen, 2003). A well-integrated system will increase the quality throughout the life cycle.

### 3.10 Verification and validation

Verification and validation (V&V) is a collection of analysis and testing activities across the full life cycle and complements the efforts of other quality-engineering functions (Wallace and Fujii, 1989). Verification establishes the truth of correspondence between a product/system and its specification. The activities in this V&V sub-process verify and validate if the system is being built right and the right system is being built to address the customer needs. The purpose of verification is to confirm that the specified design requirements are fulfilled by the system (ISO/IEC – 15288, 2008). Validation establishes the fitness of a product/system for its operational mission based on operational or field testing. The purpose of the validation is to provide objective evidence that the services provided by a system when in use comply with stakeholders’ requirements (ISO/IEC – 15288, 2008). This process performs a comparative assessment and confirms that the stakeholders’ requirements are correctly defined. All V&V activities should be traced back to the system requirements (Jain et al., 2008).

A systematic approach using different testing methods to predict the defects and performance is proposed by Amuthakkannan et al. (2008) to improve the quality of real-time software by identifying the defects in software product and improving the development processes. Qualification requirements address the needs to qualify the system as being designed right, the right system and an acceptable system (Beude, 2000).

The four elements of the qualification requirements (Beude, 2000) are:

1. observance – how the estimates (qualification data) for each input/output and system-wide requirements will be obtained, that is, test analysis and simulation, inspection or demonstration
2. verification plan – how the qualification data will be used to determine that the real system conforms to the design that is developed
3. validation plan – how the qualification data will be used to determine that the real system complies with the originating requirements
4. acceptance plan – how the qualification data will be used to determine that the real system is acceptable to the stakeholders.

RSE builds in verification activities in an iterative method throughout the life cycle of the project to confirm the conformity and identify gaps between system specifications and the final systems (Forte, 1997; Larman, 2004). By doing continuous verification activities, errors and defects are found in a timely manner and contained in the initial stages thus reducing the cost to fix (Boehm and Basili, 2001; Crispin and Gregory, 2009; Gilb and Graham, 1993; Leffingwell, 2007). Risk-based decisions on acceptable deviation and effort of rework can be made at the right time and can be followed through iteratively. Figure 3 presents a sample iterative and adaptive testing process using a design structure matrix.
In RSE, validation is performed through continuous and frequent prototyping (McConnell, 1996) with customer involvement early in the life cycle. This helps in timely validation that the developed or to be procured system is exactly what the customers want (Buxton, 2007; Grimm, 2004; Noorani, 2006). The general challenge of miscommunication of a conceptualised system between customers and engineers is well addressed. Prototypes also help in customers getting familiarised with the system as it evolves (Venkateswaran and Son, 2004). In RSE, identifying constraints and expected results would enable rapidity in validation for confirming the system to achieve its goals within the intended operational environment. Inadequate input from the design activities or use of the wrong verification and validation methods can cause unnecessary rework and retest (Levardy and Browning, 2005). The optimal information flow means that the right verification and validation activities deliver the right information using the right input data from the design process at the right time and place in a most effective and efficient manner.

3.11 Transition

The purpose of the RSE transition process is to establish a capability to provide services specified by stakeholder’s requirements in the operational environment. The activities conducted during this process verify the system’s readiness for deployment as well as the readiness of enabling systems, for example, operating systems, support systems, operator training systems, user training system, etc.

To introduce rapidity in the transition process, RSE relies on an iterative approach, continuous testing, integration and customer involvement and feedback through iterative or rapid prototyping. Deployment and operational requirements are continuously revised as the project progresses. RSE techniques ensure changes in deployment requirements are effectively managed. Rapid prototyping is used to obtain early systems validation and accelerated product development (Andrews and Goeddel, 1994). The most obvious benefit of this technique is the ability to evaluate requirements for applicability and unanticipated errors. By using iterative prototypes, RSE can monitor the rate of change of deployment and operational requirements and modify the system under development to satisfy the new requirements. The customer needs and desires may change when they begin operating the system and gain a deeper understanding of how it will support their...
mission (Boehm, 2000). Thus, requirements tend to emerge with continued use and mission understanding rather than be pre-specifiable. RSE techniques reduce total project costs because the delivered product tends to be more stable since most of the errors have been already discovered and fixed in early iterations of the system.

4 Research methodology

Based on the literature research, the authors have included and discussed within the scope of this paper, 14 SE processes from the ISO 15288 standard and 2 relevant rapid implementation techniques for each SE process evaluated. To enable a more focused research activity, some processes were combined, such as human resources and life cycle management, thus reducing the total number of process focus areas from 14 to 11. The 11 process combinations and techniques are described in Table 1. Based on this information, the authors created a survey to evaluate the applicability of these techniques in RSE projects. The survey was administered to individuals within both the commercial and defence sectors where respondents were asked to rate each technique statement on a scale of 1–5 (1 being highly disagree and 5 being highly agree) with regard to enabling rapidity within the SE life cycle. The survey consists of 22 questions along with an introduction on the meaning of RSE and a few questions on demographics of the respondents. One survey question was a false statement – a rapid SE technique which would not have relevance in application of rapidity. The authors’ expectation was that all respondents would disagree with the statement.

Table 1 Techniques by process

<table>
<thead>
<tr>
<th>Process</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human resources and life cycle management</td>
<td>Create a team with a combination of disciplines and creative/innovative flexibility</td>
</tr>
<tr>
<td></td>
<td>Plan for an increased number of release cycles that include customer approved value-added functionality</td>
</tr>
<tr>
<td>Project planning and project assessment</td>
<td>Develop project plans and update them continuously throughout the life cycle</td>
</tr>
<tr>
<td></td>
<td>At inception, plan for scope and project definition changes throughout the project life cycle</td>
</tr>
<tr>
<td>Risk management</td>
<td>Determine the initial set of known risks and identification of relevant checkpoints throughout the project life cycle</td>
</tr>
<tr>
<td></td>
<td>Use relative and derived metrics instead of direct metrics</td>
</tr>
<tr>
<td>Configuration management</td>
<td>Relate and maintain functions and service to subsystems and components</td>
</tr>
<tr>
<td></td>
<td>Define service level objectives that provide the minimum set of attributes that need to be tracked and maintained</td>
</tr>
<tr>
<td>Requirements definition and requirements analysis</td>
<td>Provide mechanisms for initial and continuous collaboration from the customer and stakeholders</td>
</tr>
<tr>
<td></td>
<td>Use iterative prototypes to manage project dynamism</td>
</tr>
</tbody>
</table>
Table 1  Techniques by process (continued)

<table>
<thead>
<tr>
<th>Process</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural design</td>
<td>Use of architectural modelling to analyse critical technical issues and evaluate alternatives</td>
</tr>
<tr>
<td></td>
<td>Utilise prototypes to allow the best architectures to evolve iteratively</td>
</tr>
<tr>
<td>Implementation</td>
<td>Selection of methods, tools and techniques that eliminate waste</td>
</tr>
<tr>
<td></td>
<td>Use an incremental development approach to break-up large functions into smaller manageable pieces focusing on core and highly prioritised functions initially</td>
</tr>
<tr>
<td>Integration</td>
<td>Place focus on integration early in the life cycle to evaluate and analyse designs which may create integration issues</td>
</tr>
<tr>
<td></td>
<td>Continuously integrating and testing components and modules physically or virtually to support traceability will result in cost, time and risk containment</td>
</tr>
<tr>
<td>Verification</td>
<td>Perform integration activities throughout the project to confirm conformity between system specifications and the final system</td>
</tr>
<tr>
<td></td>
<td>Use an iterative verification process to identify gaps between the system specifications and the final system functionality early in the life cycle.</td>
</tr>
<tr>
<td>Transition</td>
<td>Ensure the readiness of the system and the readiness of the enabling systems, for example, operating system, support system, operator training, user training, etc.</td>
</tr>
<tr>
<td></td>
<td>Identify and include deployment requirements in the requirements phase</td>
</tr>
<tr>
<td>Validation</td>
<td>Use frequent prototypes with customers to validate the system will meet the objectives</td>
</tr>
<tr>
<td></td>
<td>Identify constraints and expected results to confirm the system can achieve its intended use in its intended operational environment</td>
</tr>
</tbody>
</table>

5 Research analysis and survey results

The survey was administered to 22 respondents representing both the commercial and defence sectors. The respondents experience ranged from 9 to 31 years creation and implementation of systems in their industry. The average experience across the respondents was 20 years with an average gap (SD) of 8 years from the mean. Of the 22 techniques, the respondents agreed that 85% were applicable to RSE as seen in Figure 4. As a second level of decomposition, the total responses were then categorised by three process areas:

1 infrastructure processes which include human resource and life cycle management

2 enabling processes which include project planning, project assessment, risk management and configuration management
3 technical processes which include requirements definition, requirements analysis, architectural design, implementation, integration, verification, transition and validation.

Figure 5 presents the data in terms of percentages of strongly agree/agree, neutral and disagree/strongly disagree against the three process areas. The data presents the respondents belief that the techniques identified for the technical processes as most applicable while the enabling processes were believed to be least applicable when implementing rapidity into the SE process. The survey results indicated a clear agreement of 21 out of 22 techniques when applied in rapid environments supporting ISO 15288.

The authors then calculated mean and SD using weighted averages (−2 to 2 for Strongly Disagree to Strongly Agree) to determine any significant differences between the individual techniques. The weighted means and SD can be reviewed in Figure 6. It illustrates the respondents’ level of agreement to the techniques where the closer to the centre the higher the agreement (indicated by a high Mean score) to the technique – the smaller was the difference in their level of agreement (indicated by a small SD) across all respondents. This is an extremely important indication of two things:

1 the respondents’ support for the selection of the relevant SE processes and techniques
2 that the SE processes and techniques that were rated as relevant for achieving rapidity had very strong and consistent support across the respondents.

**Figure 4** Percentage of agreement of techniques applicable to RSE (see online version for colours)

**Figure 5** Percentage of agreement within processes (see online version for colours)
The findings show six rapid techniques representing four SE processes to have a weighted means of 6.2 and above, therefore being the most relevant. Those processes and techniques are found in Table 2. The respondents clearly identified upfront and early focus on systems integration issues, continuous and build in verification activities throughout the life cycle of the project and an iterative process in the integration and verification activities to be very suitable and relevant for implementing rapidity in SE.

There were two SE processes (integration and verification), where the responses to two RSE techniques:

1. continuously integrate and test components and modules physically or virtually to support traceability will result in cost, time and risk containment
2. perform verification activities throughout the life cycle of the project to confirm conformity between the system specification and the final system, varied the most with the highest SD.

This could be a reflection of the diversity of the industry domains represented in this exploratory study. The perception of integration and verification activities can be very different in the defence and non-defence (commercial) domains. Moreover, since these SE activities are extremely critical in the life cycle considerations, respondents tended to have strong views on them, thereby leading to significant differences.

The survey respondents only somewhat agreed (neutral) to the use of relative and derived metrics instead of exact and direct metrics, for reducing the time and resources spent on risk management. This could be explained by the need for the projects to have accuracy in risk management through more direct and exact metrics. Most projects
following RSE perform detailed management of issues arising out of oversight in risk planning and management. Also in some of the RSE projects, the nature of the project requires close to 100% accuracy in determining the risks. In projects where human lives depend on the product risks, performing a relative metric risk assessment will not be an option. Hence, this statement cannot be generalised for all RSE projects.

Table 3 presents the four rapid SE techniques associated with the three SE processes that rated the least relevant in their application to the SE life cycle. The one that created the biggest surprise was the technique to plan for scope and project definition changes. This question was worded in the negative to create additional thought in the answers. Instead, a majority of the respondents rated it as not relevant with 13 responding as expected and 9 responding either neutral or negative. Based on the weighted means, the two techniques that had the lowest means and the lowest SD were

1. plan for an increased number of release cycles that include customer approved value-added functionality
2. use relative and derived metrics instead of direct metrics.

Table 2  Most relevant techniques for rapidity based on the weighted means of the survey

<table>
<thead>
<tr>
<th>Process</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements definition and requirements analysis</td>
<td>Provide mechanisms for initial and continuous collaboration from the customer and the stakeholders.</td>
</tr>
<tr>
<td>Implementation process</td>
<td>Use an incremental development approach to break up large functions into smaller manageable pieces focusing on core and highly prioritised functions initially</td>
</tr>
<tr>
<td></td>
<td>Continuously integrate and test components and modules physically or virtually to support traceability will result in cost, time and risk containment</td>
</tr>
<tr>
<td>Verification process</td>
<td>Continuously integrate components and modules as they are developed</td>
</tr>
<tr>
<td></td>
<td>Use an iterative verification process to identify gaps between the system specifications and the final system functionality early in the life cycle</td>
</tr>
</tbody>
</table>

Table 3  Least relevant techniques applied to rapid systems engineering processes

<table>
<thead>
<tr>
<th>Process</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human resources and life cycle management</td>
<td>Plan for an increased number of release cycles that include customer approved value-added functionality</td>
</tr>
<tr>
<td>Quality management and project management</td>
<td>Develop project plans and update them continuously throughout the life cycle</td>
</tr>
<tr>
<td></td>
<td>At inception, plan for scope and project definition changes throughout the project life cycle</td>
</tr>
<tr>
<td>Risk management</td>
<td>Use relative and derived metrics instead of direct metrics</td>
</tr>
</tbody>
</table>
6 Discussions and implications

Some of the survey participants indicated that they found it difficult to answer the questions in terms of RSE since they were not familiar with the concept. They expressed it would have been easier for them if the questions were expressed in terms of reducing time to market, for example, some respondents indicated that the practices proposed by RSE are good SE practices that should be implemented in any SE project, therefore they agreed with most of the questions. The majority of the respondents commented that although they agree with the fact that the scope of the project should change throughout the life cycle of the project to allow rapidity, they did not like the question because it was not compatible with the SE processes used in their working environments.

7 Conclusion

While this exploratory study indicates a good beginning towards application of some time-tested SE processes in a rapid environment, there is caution in drawing conclusions from its limited scope. The strength of the paper comes from its empirical basis in credible research and an attempt to design a framework for application of rapidity in SE processes. It also provides a common repository of techniques supporting the SE processes within the ISO 15288 standard. Many of the outcomes in this survey could be skewed due to interpretation of the term RSE, structure of the questions and also the familiarity of the SE processes as described in the 14 that were covered in this study. Our findings generally indicate that activities related to technical processes are viewed as relevant techniques for implementing rapidity. Some aspects of the management processes as enablers were almost unanimously viewed with skepticism in assigning priority in fast-paced and resource critical rapid environments and were also considered as overheads and not essential components of rapidity. These provide an initial guidance on areas of SE where rapidity can be implemented and those where skepticism in general on the benefits of these SE processes for rapid development will have to be first removed. Further work needs to be done to initiate some larger studies addressing more specific questions on the role of SE in rapid environments.

Areas of future research

Based on the initial analysis, the following areas provide additional research opportunities:

- development and administration of a more refined survey that crosses multiple industry and project boundaries with domains focused on RSE opportunities and consequences
- systems analysis techniques that support RSE
• consequences of design decisions on deployment (certain design decisions may enable rapidity through the early phases of the life cycle, yet introduce an increase of cost in the deployment and sustaining processes)
• supportability, the cost of ownership long term, training and implications of technology changes required to sustain a RSE process.

References
Feasibility of rapid systems engineering framework


Feasibility of rapid systems engineering framework


