

# Criteria for Method Selection and Process Design of a Technical Development Strategy

Suja Joseph-Malherbe  
CSIR

[sjoseph@csir.co.za](mailto:sjoseph@csir.co.za)  
[sjosephmalherbe@gmail.com](mailto:sjosephmalherbe@gmail.com)

Copyright © 2012 by Suja Joseph-Malherbe. Published and used by INCOSE SA with permission.

**Abstract** Building on previous work by the author, the intention of the study is to collate what is available on developing a technical development strategy framework. The technical development strategy is foreseen to include a development strategy, a delivery strategy and a testing strategy with the associated approach and methods for each strategy. For example, the testing strategy will describe the testing approach of the system development cycle and it will include methods of testing, testing objectives, time and resources required.

As there are very many methods and strategies to choose from, the rational is that there should be an evaluation framework for selecting an appropriate strategy and method. The framework encourages one to design a problem-solving approach by matching a method to task (Glass and Vessey, 1998). In engaging with such a framework, the questions of what development framework are best suited to a particular application and whether enough research information is available to aid the selection of development methods and approaches need answering. These questions were raised by Glass and Vessey (1998) almost fifteen years ago in software systems development and are still relevant today.

## Introduction

The goal of any system is to make human endeavour easier and safer (Singh and Kotzé 2003). The purpose of development methodologies is to successfully design and develop systems within specific project constraints such as time, cost and resources. A methodology refers to a framework that is used to structure, plan and control the process of developing a system. It is a collection of related processes, methods (techniques) and tools. Each methodology has its strengths and weaknesses and no single methodology is necessarily suitable for use on all projects as context specific criteria bounds the development challenge of each project uniquely and needs to be addressed as such. The key to solving a development challenge lies in matching the methodology to the project and the organization (Farrell 2007). It therefore becomes necessary to develop an evaluation framework for selecting an appropriate method or approach to facilitate the development process of a system.

The question to ask is, “Which development process is best for my project?” rather than asking “Which development process (if any) is the best?” (Kolawa 2002). This question can be suitably answered through the application of an evaluation framework for selecting an appropriate development methodology.

## Overview of Development Methods

This section introduces the main groups of development methods: waterfall, incremental and iterative. Certain basic principles, under lying assumptions, strengths and weakness for each method are highlighted. Development methods transform system requirements into products that enable a service during the Utilization stage in the system life cycle. In development one ensures that the

aspects of other stages (production, utilization, support and retirement) and their enabling systems' requirements and capabilities are considered and incorporated into the design of the system (ISO/IEC TR 24748-1 2010).

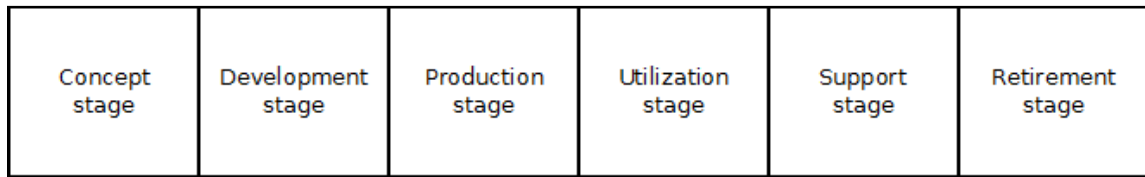


Figure 1 System life cycle model (Adapted from ISO/IEC TR 24748-1 2010)

## ***Waterfall***

This method performs the development process a single time (ISO/IEC TR 24748-1 2010). A waterfall method structures the project into distinct phases. Each phase is dependent on the previous phase and each phase requires a defined set of inputs from the previous phase. In general, the first few phases attempts to capture “what” the system must do, the next phases determines “how” the will be designed, implemented, integrated, tested and evaluated.

A derivative of traditional waterfall model is the Modified waterfall, whereby the end of one phase may overlap the beginning of another phase; this allows phases to operate in parallel with some feedback. However, the basic principle of completing the deliverables of the prior phase before the next phase is fully started still applies.

In a project that employs this method, the requirements are clearly specified at the beginning of the project and little change is expected through the system development cycle. The environment is said to be predictable and therefore the management of the project regarding tasks and their durations can be optimized. The technology utilized on the project is well known to the team. Reliance on processes and documentation is the name of the game. This is essential in major projects, to coordinate people and to gain control.

## ***Incremental***

In this method, the user needs are determined, the system requirements defined and then the rest of development is performed in a sequence of builds (ISO/IEC TR 24748-1 2010). In other words, the design, implementation and testing phases are executed more than once. At each build part of the planned capabilities are incorporated until the system is completely developed. The project evolves in accordance with a pre-conceived plan at each increment. This approach allows for a phased delivery of the system to the client. Even in this case, the compliance to the pre-conceived plan is required throughout the development.

## ***Iterative***

This method also develops a system in builds. However, in utilizing this method it is understood that the user is not fully understood and all the requirements cannot be defined up front (ISO/IEC TR 24748-1 2010). As a result, user needs and system requirements are partially defined up front and then are refined as the project progresses in each build.

This method also known as evolutionary puts emphasis on creating a series of prototypes for evaluation until the objectives are accomplished and the system is ready for final release. A well-known iterative method is the Spiral method. A number of iterations are implemented to better define requirements and design by assessing risk and observing simulation, modelling and verification progress (PM Solutions 2003). This method relies on the evolution of prototypes to help define requirements and design. Toward the end of the cycle, the prototype becomes operational

and it is used to do detail design, testing, integration and delivery. The key characteristic of the Spiral model is the risk management at regular intervals in the system development cycle.

Agile methods are also a subset of iterative methods (Williams 2007). In Agile methods, each iteration spans requirements analysis, design, implementation and testing. These methods focus on minimizing risk by developing small parts of the system. The purpose of short iterations is to allow feedback from the earlier iterations which aid in refining requirements. These changes are then adapted in the next iteration. The feedback from the customer on adapting a requirement is based on observation of the evolving system. This method is time driven rather than requirements driven (PM Solutions 2003). The requirements phase is completed as in any other method although the implementation of the method is focused on delivering the functionality as soon as possible. Agile methods are not a single approach to development rather they are a family of development processes (SCRUM, Crystal, Feature Driven Development and Extreme Programming) (Benito et al. 2010).

Table 1 highlights certain assumptions and basic principles of the methods (Farrell 2007; Lemétayer 2010; PM Solutions 2003; Boehm 1988; Sorensen). Table 2 highlights some strengths and weaknesses of the methods (Farrell 2007; Lemétayer 2010; PM Solutions 2003; Sorensen; CMS 2008).

Table 1 Assumptions and Basic Principles of Methods

Method	Assumptions	Basic principles
Waterfall	<ul style="list-style-type: none"> <li>▪ All requirements are well understood and documented at the beginning.</li> <li>▪ Change will be negligible through system development cycle.</li> </ul>	<ul style="list-style-type: none"> <li>▪ A lot of effort is put in the up-front planning of the project.</li> <li>▪ Compliance to the plan drives the whole project.</li> </ul>
Incremental	<ul style="list-style-type: none"> <li>▪ May be utilized with a complete set of requirements or with less defined objectives.</li> <li>▪ Well-defined interfaces are required.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Follows a pre-conceived plan developed at the beginning of the project.</li> <li>▪ The development phases are executed more than once.</li> <li>▪ Allows for phased delivery to the client.</li> <li>▪ Compliance to the plan is a high priority.</li> </ul>
Iterative	<ul style="list-style-type: none"> <li>▪ Requirements for the system are not well understood.</li> <li>▪ Requirements are changing rapidly.</li> <li>▪ Customers will have to commit to the whole development process (regarding their involvement)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Development of small and ever-growing portions of a system to assist in uncovering important issues that may have an impact on the successful operation of the system.</li> <li>▪ Validation of the system is based on the adequacy of the developed system to address the problem and not on the compliance to pre-conceived requirements.</li> </ul>

Table 2 Strengths and Weaknesses of Methods

Method	Strengths	Weaknesses
Waterfall	<ul style="list-style-type: none"> <li>▪ Allows for co-ordination of large teams.</li> <li>▪ The schedule is visible to all stakeholders.</li> <li>▪ Progress can be reported to stakeholders based on the up-front planning.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Cannot readily respond to emerging technical and functional requirements.</li> <li>▪ Any change required due to missed or unclear requirements would lead to a change in the budget and timeline.</li> <li>▪ Danger of discrepancy between what the client expected and what was delivered due to the lead time between full requirements definition at the beginning of the project to delivery and the lack of client participation during that period.</li> </ul>
Incremental	<ul style="list-style-type: none"> <li>▪ Early functionality of the system is made visible.</li> <li>▪ Progress can be reported to stakeholders based on the up-front planning.</li> <li>▪ Assists in reducing risk, if it is too risky to develop the whole system at one go (such as following the Waterfall method)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Formal reviews and audits may be difficult to implement on increments than on a complete system.</li> <li>▪ The tendency to leave address the difficult problem to the future to demonstrate success in early increments.</li> </ul>
Iterative	<ul style="list-style-type: none"> <li>▪ Ability to accept change to requirements throughout the system development cycle.</li> <li>▪ Allows for feedback from stakeholders to change the requirements.</li> <li>▪ Control costs and risk through prototyping as a technique.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Will require strong project management process in place to control and manage the changes to prevent “scope creep”.</li> <li>▪ There may be a danger that systems that utilize prototyping as a technique have performance issues as the development does not focus on any non-functional requirements.</li> </ul>

## Selecting a Development Methodology

The choice of a suitable development methodology is based on technical considerations, team considerations, project characteristics as well as the business and organizational environment in which the project operates in (Lemétayer 2010; CMS 2008).

Boehm and Turner (2003) developed a tool to determine whether an Agile or a plan-driven approach is more appropriate for a project. The assessment is based on five critical factors which are measured on a scale from pure plan-driven to pure Agile (see Figure 2). The factors are: need for personnel supervision, project criticality, project size, control culture and requirements stability.

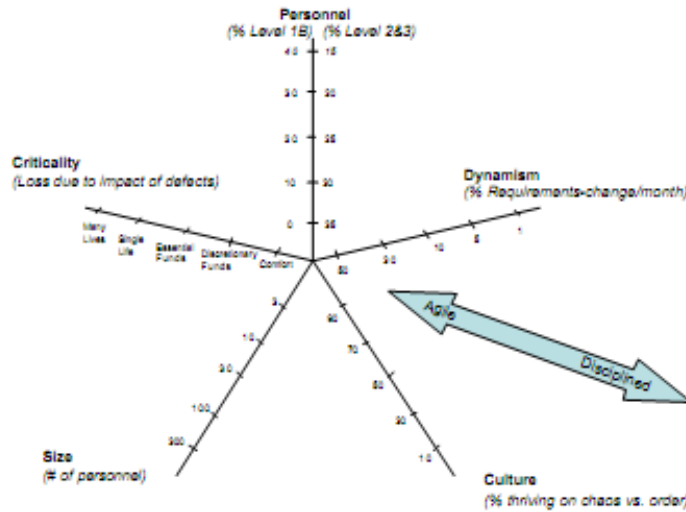


Figure 2 Dimensions Affecting Method Selection (Boehm and Turner 2003)

From the figure above, one can see that the following conditions encourage the use of plan-driven approach:

- When a high-level of personnel supervision is required.
- Highly critical projects that require controls and rigidity.
- When coordination is required with more people on the development team.
- When development happens in well-established companies with a control culture, that is, people are more comfortable when their role in the project is clearly defined.
- There is minimal requirements change during development.

The opposite of the above stated conditions will encourage the use of Agile approaches. In addition to Boehm and Turner’s work, numerous other factors have been mentioned in literature that has an influence on the use of an approach versus another. In the software development environment, a study was conducted by Lemétayer (2010) to identify critical factors to choose a development methodology. Broadly speaking, some contingency factors relate to the project environment and others to the project itself.

Table 3 List of a Few Contingency Factors

Compliance and governance	Culture of the team
Corporate culture	Customer involvement
Market uncertainty	Technological uncertainty
Nature of the contract	Stability of requirements
National culture	Co-location of the team members
Organization size	Team maturity
Management support of an approach	Project duration and cost

The study concludes on two factors that are important in selecting a methodology for a project and these are organizational culture and the empowerment of the project team. Organizational

culture contains variables relating to the project environment such as level of entrepreneurship and the methodology supported by management. The “empowerment of the project team” factor relates to the characteristics of the project such as project uncertainty and technological uncertainty.

In another study the matching of organizational types to a development methodology was shown (Farrell 2007). The organizational type pertains to the organization developing the system and one would also extend it to the organization that hosts the development as the hosting organization may impose some of the processes they follow. The basic organizational types are entrepreneurial, machine, professional, diversified, innovative, missionary and political. These categories can be described in brief as follows:

- “Entrepreneurial: This is a simple and informal organizational type. It is flexible and has very little hierarchy. These organizations have simple and dynamic environments.
- Machine: These organizations are more complex and formal. They are hierarchical in nature and are characterized by a centralized bureaucracy and established, formal processes. These organizations have a strong division of labour where employees are generally divided into functional groupings. The environments are simple and stable and the organization is more mature and resistant to strategic change. The organization is generally efficient, reliable, precise and consistent.
- Professional: This organization is bureaucratic and decentralized and has minimal hierarchy. There are wide spans of control and the organization is complex and stable. These organizations are comprised of groups of individual professionals working autonomously. The overall strategy is stable but the specific details are often changing. These organizations are focused on democracy and personal autonomy.
- Diversified: This organization is characterized by autonomous market divisions under a central administration. These organizations offer diverse products and services and are generally larger and more mature.
- Innovative: This organization is fluid and organic and is generally based on a decentralized structure. The organization is usually comprised of functional experts and relies on mutual coordination. These are commonly a matrix structure in a complex and dynamic environment which undergoes frequent change.
- Missionary: This organization is based around a strong system of values and beliefs and is based on a sense of mission. They often have charismatic leadership and manage the organization through selection and indoctrination of members. The organization is basically decentralized but has powerful centralized controls.
- Political: This is not so much an organization but a characteristic that can appear in different organizations. There are issues with self-interested power, conflict, and political games. The political aspects are usually overlaid on a conventional organization.” (Farrell 2007)

Farrell (2007) used the following characteristics to determine the organizational type:

- Bureaucratic or non-bureaucratic.
- Functional structure, project structure or matrix structure (combination of functional and project organizational structures).
- Low complexity projects or high complexity projects.
- Low work effort projects or high work effort projects.
- Custom work or large volume work.
- Unchanging or changing environments of requirements.
- Large or small organization.

- Flexibility of the organization.

The table below proposed by Farrell (2007) can be used at a very basic level to determine the organizational type.

Table 4 Organizational Characteristics and Types (Farrell 2007)

	<b>Entrepreneurial</b>	<b>Innovative</b>	<b>Machine</b>	<b>Diversified</b>	<b>Professional</b>
<b>Bureaucracy</b>	Non-bureaucratic	Non-bureaucratic	Bureaucratic	Non-bureaucratic or Bureaucratic	Bureaucratic
<b>Structure</b>	Matrix (with less hierarchy)	Matrix	Functional	Functional	Functional
<b>Complexity</b>	High	High	Low or High	Low or High	High
<b>Environment</b>	Simple & dynamic		Formal		Decentralized
<b>Work effort</b>	High	High	Low or High	Low or High	Low or High
<b>Work type</b>	Custom	Custom	Large volume	Custom or large volume	Custom
<b>Change</b>	High	High	Low	Low or High	Low or High
<b>Size</b>	Small/Medium	Small/Medium	Medium/Large	Medium/Large	Medium
<b>Flexibility</b>	High	High	Low	High	Low

The table below shows the organizational types versus methodologies. It gives general guidelines on which methodology may be suitable for which organizational type.

Table 5 Methodologies and Organizational Structures (Farrell 2007)

	<b>Entrepreneurial</b>	<b>Innovative</b>	<b>Machine</b>	<b>Diversified</b>	<b>Professional</b>
<b>Waterfall</b>	No	No	Yes	Yes	Yes
<b>Incremental</b>	No	No	Yes	Yes	Yes
<b>Iterative</b>					
<b>Spiral</b>	Yes	Yes	Yes	Yes	Yes
<b>Agile</b>	Yes	Yes	No	No	No

Lemétayer (2010) suggests that factors such as organizational culture, project team culture and the role the customer plays in a project contributes greatly to which development methodology to use. Further research is necessary to clarify the influence of these factors on the fit of a methodology to a project.

Elaborating on the role customers play in a project and organizational culture, in general defence-related projects follow a highly structured, top-down, step-by-step process, based on an assumption that an end state is known. However, the traditional approach to acquiring systems is

not completely appropriate as systems are becoming more software-intensive – a study has shown that software in military systems now account for eighty percent regarding its functionality (Benito et al. 2010). The traditional acquisition process may not be suitable for software-intensive systems development as the development cycle is too long and the problem is compounded by the practice of fixing detailed requirements at the beginning of a project. In addition, by the time the solution is fielded the technology may be outdated or required functionality no longer relevant.

The Defense Science Board (2009) recommended a new acquisition process to support rapid information technology development cycles and software-dominated acquisitions. They argue that the problems that face Information Technology (IT) acquisition are similar to those that the acquisition of major systems experience as these systems contain high content of embedded IT. They go on to say that conventional acquisition process is too long and cumbersome to fit the needs of the many systems that require continuous changes and upgrades (due to supportability of hardware and the short half-life of commercial IT to name a few). The changes to the acquisition process include:

- Early (prior to milestone build decision) and continual user involvement.
- Multiple rapidly executed releases of capability.
- Early and successive prototyping to support an evolutionary approach.
- Modular, open-systems methodology to support the rapid adaptation to changing circumstances.

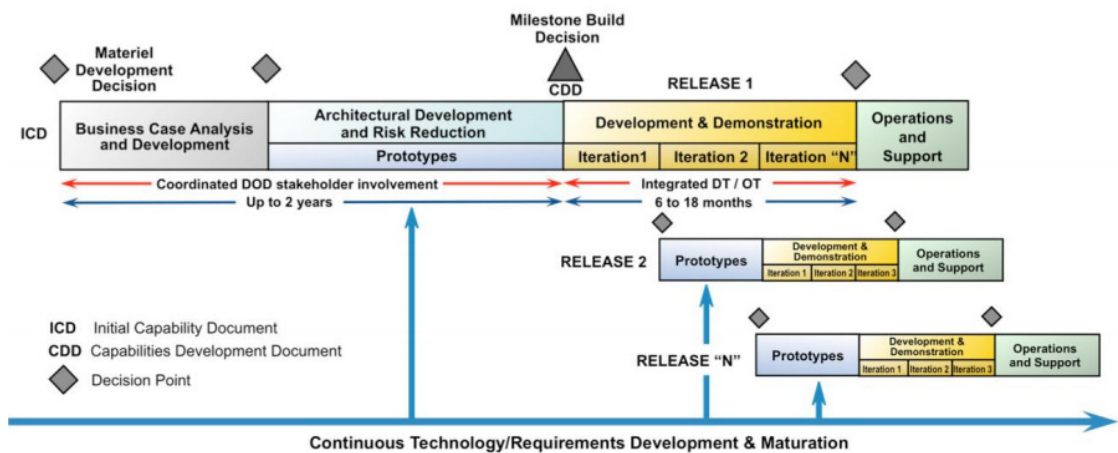


Figure 3 Recommended Changes to the Acquisition Process for Information Technology (Defense Science Board 2009)

Benito et al (2010) in their report recommend a new acquisition process based on Agile software development practices for rapid acquisition. The following distinction is made between Agile acquisition and Rapid Acquisition.

---

**Agile acquisition** is defined as “actions taken by a capable, experienced government management team to adjust program parameters throughout the life-cycle to respond to changes in the program’s environment and its customers’ expectations.” (Benito et al. 2010)

---

**Rapid Acquisition** is defined as “the actions taken by a government management team to acquire a capability in the shortest time possible within or outside of the government acquisition process (definition, funding, procurement, development, testing and fielding).” (Benito et al. 2010).

---

Based on the above definitions, the Agile acquisition emphasis is on reacting to change in a project’s environment and customers’ expectations and Rapid acquisition focuses on fielding a



solution as quickly as possible. With Rapid acquisition, the intention is to field a solution from a number of days to up to two years (see Figure 4) by reducing uncertainty early in the development process. This is achieved through earlier development prior to systems acquisition and increased prototyping.

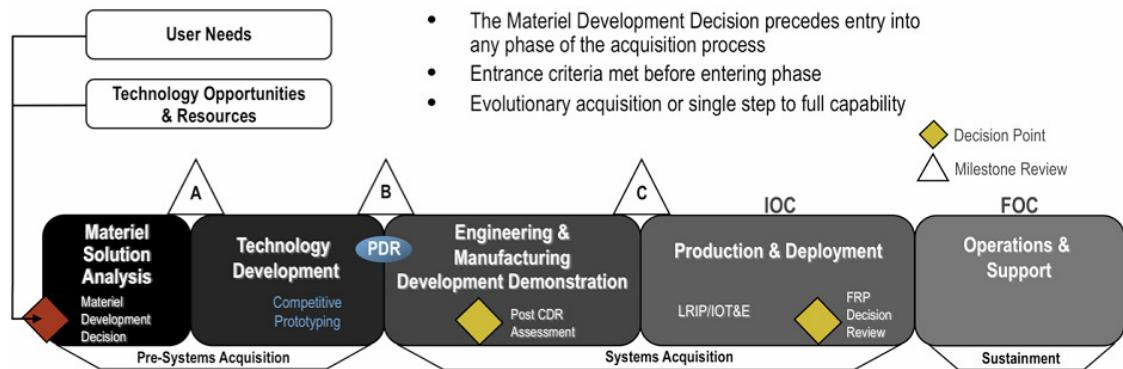


Figure 4 New Acquisition Process (Benito et al. 2010)

As shown in figure 4, recommended changes to the traditional acquisition process include the following:

- “All program/projects will proceed through a formal acquisition process entry point, the Materiel Development Decision (MDD). Programs will no longer immediately proceed to Milestone B.
- Programs requiring technology development will conduct competitive prototyping at the system or sub-system level, when appropriate; to ensure that technologies have been demonstrated in a relevant environment and as a result key risks have been dealt with before programs/projects are initiated.
- Where consistent with the strategy for the Technology Development Phase, preliminary designs will be prepared to ensure that requirements are well understood and cost estimates well informed.
- The Engineering and Manufacturing Phase has been redesigned to place additional emphasis on systems engineering and manufacturing readiness.
- Configuration Steering Boards have been established to ensure that requirements changes/creep, a traditional contributor to increased cost and extended schedules, are not casually approved." (Benito et al. 2010)

## Conclusion

Glass and Vessy (1998) in their paper presented the following quote by Bo Sanden, “Rather than arguing about which design method is best, we should take an eclectic view and use any combination of approaches that yields important results in a given situation.”

Adopting a methodology to execute a project is not easy and clear-cut. There are very many factors that influence making a choice. It is encouraged that in selecting or designing an approach to solve a problem, the process be managed as a separate task or project prior to using or implementing it. The approach will need to be communicated to stakeholders (both internal and external) to ensure project success. Clearly, the organizational cultures of the project team and customer have a major influence on the best fit of an approach to a project. Further study is required to understand the relationship between organizational culture and the choice of development methodology.

## References

- Benito, R., Casagni, M., Mayfield, K. and Northern, C., "Handbook for Implementing Agile in Department of Defense Information Technology Acquisition." MITRE Technical Report, December 2010. <http://www.mitre.org>
- Boehm, B. and Turner, R., "Observations on Balancing Discipline and Agility." *Proceedings of the Agile Development Conference*, 2003.
- Boehm, B., "A Spiral Model of Software Development and Enhancement.", IEEE Computer, 1988, 21(5), pp 61 – 72.
- CMS (Centers for Medicare and Medicaid Services). "Selecting a Development Approach." Department of Health and Human Services, USA, March 2008. <http://www.cms.hhs.gov>
- Defense Science Board, "Department of Defense – Policies and Procedures for the Acquisition of Information Technology" March 2009.
- Farrell, A., "Selecting a Software Development Methodology based on Organizational Characteristics." Athabasca University, October 2007.
- Glass, R. and Vessey, I., "Strong vs. Weak Approaches to Systems Development", Communications of the ACM, Vol. 41, No. 4, April 1998.
- International Organization for Standardization., "Systems and Software Engineering – Life Cycle Management- Part 1: Guide for Life Cycle Management." ISO/IEC TR 24748-1, 2010.
- Kolawa, Adam, "Which Development Method is right for your Project?", Stickyminds.com, Jan 2002. <http://www.stickyminds.com>
- Kotzé, P. and Singh, S., "An Overview of Systems Design and Development Methodologies with Regard to the Involvement of Users and Other Stakeholders." Proceedings of South African Institute for Computer Scientists and Information Technologists, 2003, pp 37-47.
- Lemétayer, J., "Identifying the Critical Factors in Software Development Methodology Fit." Victoria University of Wellington, 2010. <http://researcharchive.vuw.ac.nz/>
- PM Solutions., "Selecting a Software Development Life Cycle (SDLC) Methodology" White Paper, 2003. <http://www.smc-i.com>
- Sorensen, R. "A Comparison of Software Development Methodologies." <http://ece.arizona.edu>
- Williams, L., "A Survey of Agile Development Methodologies." 2007. <http://agile.csc.ncsu.edu>

## Biography

Suja received her B.Sc. degree in Electrical Engineering from the University of the Witwatersrand and completed her research-based Masters (M.Eng) degree in Electrical and Electronic Engineering from the University of Johannesburg. She joined CSIR - Defence, Peace, Safety and Security (DPSS) on completion of the undergraduate degree. She was initially part of the project team involved in the application area of modelling and simulation that developed various solutions which ultimately provided decision support capabilities to the South African Air Force (SAAF). Currently she is a senior engineer at the CSIR offices in Stellenbosch, where she practices as systems engineer in the area of power systems (power management and power sources) for the dismantled soldier. She also serves as the INCOSE SA Membership Officer. Her interests are in complexity and systems engineering, with a focus on requirements analysis and the interaction between development teams and stakeholders.