

# Program management approach for large-scale information technology development projects: a case study

Vijay Somanchi<sup>1</sup> and Ravikiran Dwivedula<sup>2</sup>

<sup>1</sup> IBS Hyderabad, Hyderabad, India

<sup>2</sup> SKEMA Business School, Lille, France; IBS Hyderabad, Hyderabad, India

## Abstract

This case study presents the program management approach in managing large-scale information technology projects. Using the case of ACME Solutions Ltd., a large software development company, the authors first emphasize on the need to use program management over project management when managing related multiple information technology projects. Then, they explain the program lifecycle in this company and discuss the 'process' and 'people' related challenges across the program stages. Broadly, they observe that there is a lack of coordination between the stakeholders and processes of the program. Recommendations to overcome these limitations are discussed.

**Key words:** Program Management; Product Development; Information Technology; Project Coordination; Case Study

## Introduction

Our view of project management has changed in the past two decades. Projects and Programs are now positioned as vehicles to put into operation the strategy and/or used to create conditions that shape newer strategies for an organization (see Morris & Jamieson 2005 for a detailed discussion on the relation between an organization's strategy, programs, and projects). This then has two implications. The first is that project management will function alongside the other business functions of the organization. Second, that the role of project program management is better understood in the implementation of such business strategy (see Cooke-Davies, Crawford & Lechler 2009). While at the strategic level, both project portfolios and project programs are focused on prioritizing the resources for optimal business performance (Partington 2000), programs offer a better methodology to deal with day-to-day coordination of projects so as to ensure their implementation success (Thiry 2004). While there is an extensive literature that discusses managing projects, multiple projects, and large-scale projects, research using program management as a theoretical lens to understand the dynamics of interrelated multiple projects is lacking. Also, most of the studies reported are from United States and Europe, and few from emergent economies such as India. This is significant considering that, for its Eleventh Five Year Plan for the period 2007-2012, the government of India has earmarked US\$550 billion for infrastructure projects alone. Further, despite the economic meltdown and a cautious recovery, Indian software exports were valued to be around US\$ 48-50 billion for the year 2009-10. Therefore, addressing these gaps in the project-program management research, our objective for this paper is to explain how a global information technology company is using Program Management as a framework to manage its suite of products in India.

We have organized our paper thus. In the ‘Theory’ section, we will first present the operational definition of Program Management for the purposes of this study. We will also highlight the need to use programs over projects. This will be followed by the ‘Methodology’ section, where we will explain our stance on the choice of research philosophy and for using a case study. In the following section, ‘ACME’s Program Management Approach for Product Development’, we present the case study. Following a background note on the company, we will discuss the program organization structure, and the program lifecycle. Important challenges encountered across the program stages are presented with solutions to overcome them. This is followed by the Conclusion.

## **Theory**

Program management is a method that ensures maximum gains for an organization by the integration of project management activities. While there are multiple perspectives (and definitions) to understand project program management, we will adopt the definition of program as: (i) the means to implement an organization’s strategy using optimal level of resources; and (ii) projects which are grouped purposefully to achieve business benefits. Thus, we use Thiry’s definition of a program (2004) – as being *a collection of change actions (project and operational activities) purposefully grouped together to realize strategic and/or tactical benefits* – as the operational definition of program management for this paper. A similar definition by United Kingdom’s Office of Government of Commerce (OGC), in its widely accepted program management standard for *managing successful programs* (MSP), defines a program as “a temporary, flexible organization created to coordinate, direct, and oversee the implementation of a set of related projects and activities in order to deliver outcomes and benefits related to the organization’s strategic objectives. (OGC 2007: 4; cited in Shao, Turner & Müller 2009). Program management itself is defined as the action of coordinating, directing, and implementing a dossier of projects and transformation activities to achieve outcomes and realize benefits that are of strategic importance to the business.

According to another influential project management body – The Project Management Institute (PMI® 2006; p.4), a program is a *group of related projects managed in a coordinated way to obtain benefits and control not available from managing them individually*. Program management is therefore defined as *the centralized coordinated management of a program to achieve the program’s strategic benefits and objectives*.

The above definitions suggest that program management is essentially about building synergies across the projects (Blomquist & Müller, 2006), focusing on building networks across the program (Shao, Turner & Müller, 2009). It may not be as much about managing individual deliverables that contribute to the benefits of the program (PMI®, 2006, p.6).

## **Need for programs over projects**

Before we begin the discussion on program management, we first need to ask *why do we need programs over projects?* In other words, we ask *why do ‘projects’ fail so that we need ‘programs’*. Research on project failures has been ongoing at least for two decades. Broadly, the authors have identified the causes to be both people and process related (Kappelman, McKeeman & Zhang 2006). Lawrence and Scanlan (2008), in their review of large-scale aerospace projects, have identified: *poor initial planning, lack of clear objectives (and deliverables); improper*

*resource allocations; poor risk management; poor change management; lack of buy-in from the stakeholders; and poor understanding of priorities* as the major reasons for projects to fail. Specifically in case of Information Technology (IT) projects, Charette (2005) states that 5% to 15% of the IT projects are abandoned before delivery. In their review of IT project failures, Cerpa and Verner (2009) identified both process-related risks such as *delivery date influencing the development process, improper risk management with risks not being re-assessed, and controlled; and delivery date decision being made without appropriate requirements information; and people-related risks such as lack of rewards, and unpleasant experience while working on the project*. Similarly, Kappelman, McKeeman, and Zhang (2006) from their survey of 138 IT project managers report that *lack of top management support, ineffective project manager, inadequate involvement of stakeholders, inadequately committed team, and lack of expertise* as major people related problems leading to project failures. The critical process-related problems were *requirements and success criteria not being documented, improper change management process, ineffective scheduling, communication break—down between the stakeholders, prioritization of resources, and lack of business driver for the project*.

These examples then support the view that ‘project management’ may be inadequate to manage the complexities associated with large-scale projects, where projects may have to be undertaken concurrently, and therefore a ‘systemic’ perspective is required. Previous studies seem to agree with this opinion. In a volatile business environment, program management is capable of providing a centralized view of all the projects in an organization. This leads to better prioritization, planning, and coordination of projects, which in turn ensures good governance of projects (Shehu & Egbu 2007; Shenu & Egbu 2010).

Williams and Parr (2006) suggested the following benefits of program management over project management:

- Effective execution of major business investment in projects
- Improved on time delivery by understanding key interdependencies between the projects and managing their interfaces
- Effective utilization of resources
- Identification and management of major project risks leading to efficient time and cost management
- Effective decision made through a business perspective, thus reducing the scope conflicts within the projects and the program
- Enhanced delivery capability by promoting shared practices among the program stakeholders
- Increased realization of benefits through integration of processes, system, people, and organizational change

The next question that is to be addressed is how should these projects be organized? Pellegrinelli (1997) and later Thiry (2004) suggest three *Configurations* – perspectives to manage projects and associated work (including programs) in groups:

- Portfolio or chunked configuration
- Strategic or Goal-Oriented configuration
- Incremental or Heartbeat configuration

When projects are grouped around a common ‘theme’, it is called a chunked or portfolio configuration. The primary objective is to achieve *efficiency* through prioritization of projects, identification of dependencies so that resources are better managed, and thus projects are better controlled.

When projects are grouped around a common ‘purpose’, it is called a strategic or goal-oriented configuration. The primary objective will be to increase the organizational effectiveness by translating business needs into actions, reduce uncertainty through iterative development, and develop a learning loop that fosters creativity.

When projects are grouped under a common ‘platform’, it is called as ‘incremental’ or a ‘heartbeat configuration’. The platform may be a process, business system, or infrastructure. It would be possible for the organization to provide an integrative framework where the short-term requests can be mapped to the long-term strategy of the company, controlled changes are affected when necessary, and continuous reassessment (of the projects’ performance) with a holistic perspective is possible.

Thus, such grouping of the projects seems to address the process- and people-related concerns that lead to failure of large-scale projects. In the following section, we will present the case study of a global information technology company, ACME Solutions Ltd. (referred to as ACME hereafter), where large-scale projects are managed as a program. The scope of this study is restricted to ACME’s operations in their Development center in Hyderabad, India.

## **Methodology**

We have used a descriptive approach to study the problem. This approach describes the phenomena as they exist and goes beyond the exploratory research since it is undertaken to describe the characteristics of the issue. Thus, it is heuristic in nature and is grounded in *Constructivist philosophy*. This approach posits that knowledge does not reflect any transcendent reality but is contingent upon conventions, human perception, and social experience. This is to say that the authors’ experience of working with the described program management approach has shaped their perceptions about these phenomena. From the epistemological perspective, the research instruments are the experiences (with Program management at ACME) of the authors themselves. Thus, it is a case of *participatory action research*.

The authors were engaged in a systematic process of self-reflection which involved – planning a change, acting & observing the process and consequences of change, reflecting on these processes and consequences, and re-planning when trying to understand a phenomena (Denzin & Lincoln 2005). This led to a descriptive case study. Although, case studies are useful for theory building in areas where existing theory is limited (Eisenhardt, 1989) and theories can be proposed using a single case study (Lee & Baskerville 2003; Yin 2003), this is not the purpose of this paper as the intention is only to provide a rich description of the case (Bogdan & Biklen, 1992).

## ACME's program management approach for product development

### *Background*

ACME is a software product development company headquartered in the United States. It has evolved into a global corporation over a period of two decades, and employs more than 20,000 associates across the globe. It specializes in developing and marketing Enterprise Software Products for thousands of small, medium, and large companies, which are organized into different verticals such as Banking, Finance, Insurance, Manufacturing, Supply Chain Management, Process Manufacturing, and Human Capital Management. Each vertical develops its suite of products using *Standard Development Kits (SDKs)* that are highly customized to the client's needs. For example, ACME has specialized and well-developed software for Human Capital Management needs that comes with packaged functions such as Payroll, Benefits, Recruitment, Training and so on, so that customers can install and run the software as an 'out-of-the-box solution' for their business. ACME also provides the facility of purchasing and using independent standalone components (independent functions such as Payroll or Recruitment) if required so that the customer need not buy the entire package and only purchase components that they require. In this way, ACME meets the customer requirements without relying on third-party solutions.

In such a project-organization, the key is to ensure coordination between the different types of project teams that are engaged in the development of either a complete enterprise solutions package or the standalone components of that enterprise solutions package. These project teams need to ensure that the product they develop is capable of working as a standalone but can also function with other modules. For example, while Payroll can work as an independent entity it should also work with other products such as Benefits Management, Training, or Career Management. In some cases, they also need to ensure that their software also works with third-party software, if the customer so chooses. In order to meet this organizational requirement, each component has a different development team. This ensures that while each component is independent, they can also be integrated with another component or a product as required by the customer. This then means that at a given point of time, components or products may be in different stages of development. However, when releasing these components or products as a *Suite*, the different development teams need to ensure that the components and products are fully integrated such that the customer need not wait for a specific component to be released. In order to manage these complex 'streams' of work, ACME has adopted program management as a framework to control these related multiple projects.

### *Program organization structure*

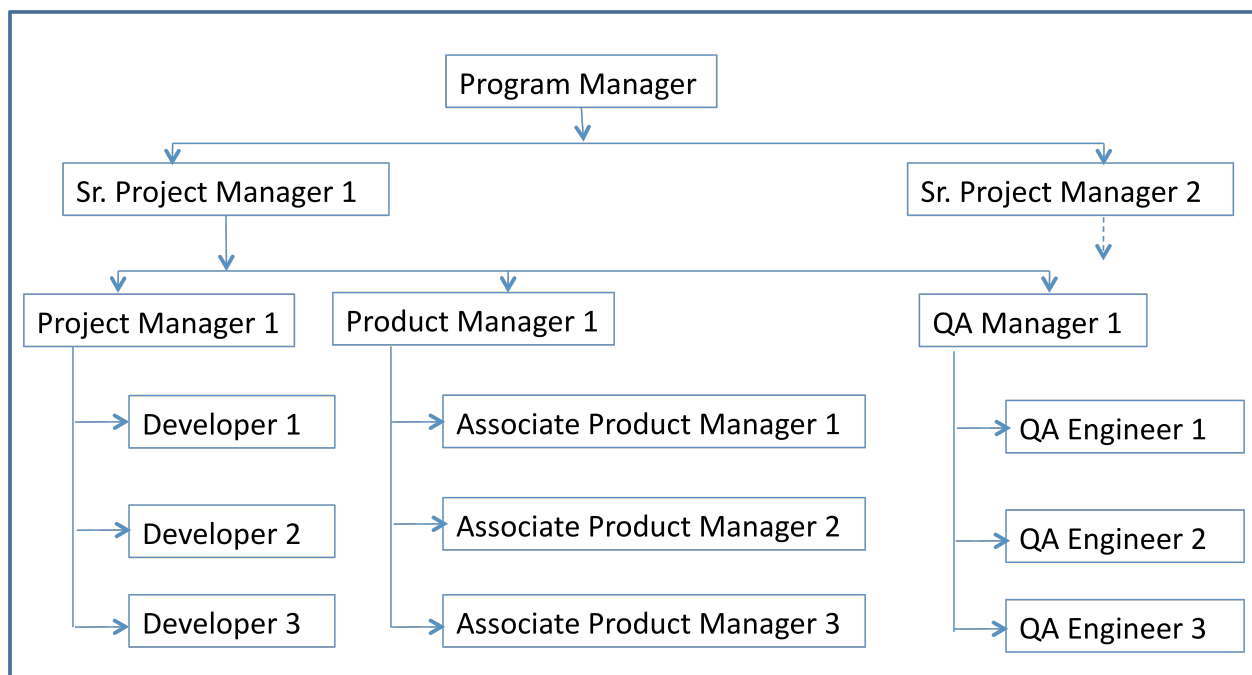
As mentioned earlier, ACME's project management approach requires products/components to be developed independently, but to be released simultaneously. Therefore, ACME uses the 'Waterfall model'<sup>1</sup> to develop these software. An apparent limitation of this approach is that a

---

<sup>1</sup> The Waterfall Model is a sequential software development process that includes different stages such as *Requirement Analysis, Design, Implementation, Verification, and Maintenance*. This model maintains that the progress of the project should be 'sequential' passing through all of these phases, and one can proceed to a later phase only after the scope of the preceding phase is completed and perfected.

delay in the development of one or more components or products will necessitate revising the schedules for the rest of the components or products until all of them are in sync.

In order to overcome such risks, ACME has a program manager who manages all the *threads* (multiple tasks or processes) of the projects that are being undertaken as a part of that program. For example, for the ‘Human Capital Management’ program, the program manager is in charge of Payroll, Benefits, Recruitment, Career Management, Appraisal System, and Training; all of these components being managed as ‘projects’. Under the program manager there is a senior project manager who is in charge of each (project) stream. The senior project manager is assisted by three leads or roles – project manager, quality assurance manager, and a product manager. While the project manager heads the project implementation, the product manager handles the requirements management. Figure 1 shows the program organization structure at ACME. A detailed description of the roles in this organization is discussed next.



**Figure 1. ACME program organization structure**

**Program manager:** The program manager is responsible for the overall planning, execution, and control of the program. The program manager coordinates the projects in their various stages of individual projects such that they meet the common release date (of the program).

**Senior project manager:** The senior project manager oversees all the aspects of one project. The associated project manager, product manager, and Quality Assurance (QA) managers report to him. A senior project manager reports to the program manager. The senior project manager is the main person responsible for the delivery of the project related to his vertical on time.

**Product manager:** It is the job of the product manager and his team to interact, collect, assimilate, verify, and review the technical feasibility of the requirements of the customer and create the Functional Design Document and High Level Design Documents which are then passed on to the corresponding project manager. It is also the job of the product manager to determine what features will go into the current release and what can wait or need to be deferred.

**Project manager:** The project manager reviews the High Level Design Documents after doing the feasibility analysis. Each project manager has a team of developers reporting to them. These developers are entrusted with the responsibility of implementing the design using the SDKs.

**Quality assurance (QA) manager:** The QA manager is responsible for the testing and validation of the software. The QA manager and their team of QA engineers are in-charge of the Unit testing, Integration testing, System Testing, Stress Testing and End-user testing to ensure that the software meets the industry standards or even surpasses it.

**Developers:** Developers are programmers who are mainly responsible for the development of the code based on the design provided to them. It is their responsibility to ensure that they adhere to the functional design that has been provided to them by the design team. So developers have to have the right technical skills to convert the functional design into a working piece of code using the SDKs at their disposal.

**Associate product managers:** They are responsible for the design and feasibility of the sub-components of a product. They report to the product managers in charge of product design. Associate product managers determine the feasibility of a certain functionality being implemented in the current product. They do detailed analysis and submit their updates to the product managers.

**Quality assurance (QA) engineers:** QA engineers do the quality assurance for the product. With various tools at their disposal, they take up unit testing, system testing, load testing and so on to minimize the bugs in the product. If any bug is found, QA testers reject the code and the developers responsible for the code will have to ensure that they fix the issues.

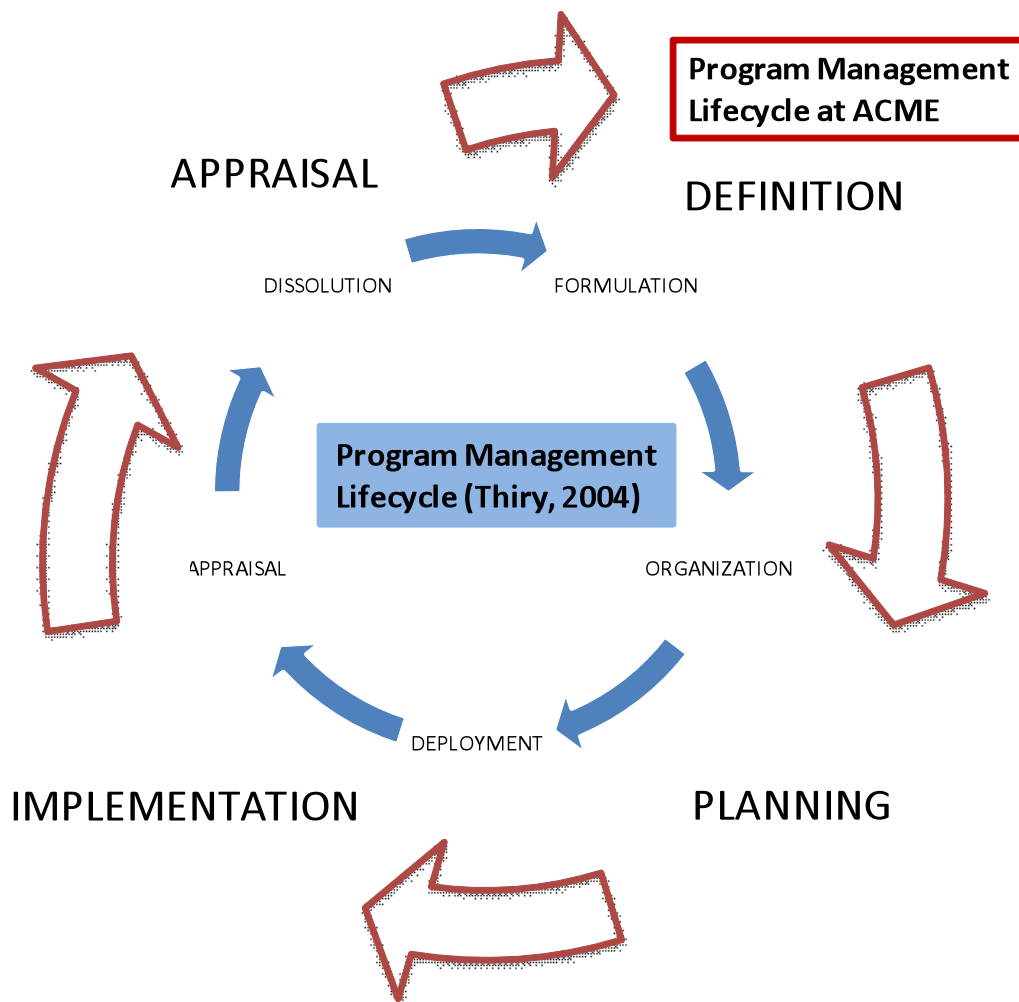
### *Managing the program stages*

The project-organization also discussed above poses certain inherent challenges. For example, since ACME tries to release the entire suit of products as a single package, a schedule overrun of one product can delay the release date of the entire program. Resource management is another critical issue. The scope of the different projects within the program will vary. Improper scope management will lead to under utilization of resources, which in turn will put pressure on the other projects that require these resources. Human factors such as unavailability of experienced personnel and employee attrition are also major problems. Issues such as these that occurred across the program's lifecycle at ACME are discussed below.

ACME's program lifecycle is composed of 4 stages – *definition, planning, implementation, and appraisal*. Conceptually, it corresponds to Thiry's program management lifecycle (Thiry, 2004) as shown in Figure 2.

*Definition stage*

In this stage, the purpose of the program is clarified. In the case of ACME, the fit between the various projects within the program and their scope is defined at this stage. Most importantly, the program manager identifies the various projects that are to be performed under one program and, if necessary, distributes the projects under various programs. Perhaps the most important activity of this stage is identification of the right people and resources to roll out the program. Allocation of resources was done using a 'bottom-up' approach where resources were allocated to a project, according to its size, and not for the entire program. Any ambiguities at this stage will lead to scope creep.



**Figure 2. Program stages at ACME**

An important area of concern at this stage is to identify the key functionalities or activities for the entire program before organizing them into projects. Identifying the key people and resources required to take the program forward is also critical at this stage. If this exercise is not done properly, it will lead to problems in terms of determining project scope and consequently the



scope of the program. The outcome of this exercise will largely influence the success and failure of the program.

### *Planning stage*

Development of the 'Business Case', which justifies the business rationale of the program, is the key activity in this stage. At ACME, the business case is defined by capturing the market conditions and also the customer requirements. Every product (of the program) is determined such that it is aligned with the most generic business practices and processes. Identification of the right resources and technologies for the implementation of each project is done at this stage. For example, for the development of a self-service application (web-based application that any user can easily use) might require a developer who is comfortable with Java, JSP, and J2EE platforms.

An important concern at this stage is the differing perspectives of the technical and the non-technical team members with respect to the objectives of the projects and the program. For example, if the technical architects are not familiar with the business objectives, and are not involved during the customer requirement analysis, the final software might prove to be unwieldy or difficult to use by the end-users. Another challenge is the availability of resources. So much so, that the design engineers may have to consider technology that is popular and easily available when required. As with the definition stage, scope creep is another major challenge in the planning stage. In our experience, scope creep led to several redesigns and necessitated revisiting the functionalities. This led to extra effort in terms of person-days and increased cost.

### *Implementation stage*

The implementation stage includes the actual execution of the planned actions of the program. Thus, once the scope and functionality is determined, the implementation strategy is drawn out. Given the nature of the product suite, different functionalities require different technologies based on the kind of usage. For example, for an internal application that does not require a high number of users to access, a desktop client is sufficient; whereas, if multiple users will be accessing the application, then a web-based interface would be required. These two types of usage require different approaches to building the application.

Another important activity is the Product Review that is done from time to time to ensure that the product development teams are aligned to the initial framework of the design. Any design issues that were not considered are reviewed several times and a decision is taken based on the scope of the additional work involved.

Overlapping functionalities is another challenge that teams often face. In such scenarios, the functionalities have to be jointly developed so that each team gets to cooperate with the other team that partly owns the functionality.

It is very difficult to envisage a finished product in software without a model or concept. End-users have to be given a 'taste' of the new application, called the 'Conference Room Pilot' or CRP. Once the CRP is complete, it is quite possible that it would have to be revisited because the end-users want to add more functionality or richness to the interfaces. In addition, limitations of

the new development tools or SDK might surface as these SDKs might not support certain functionalities.

Product and development reviews can bring out issues that could not be identified in the initial design phases, and these have to be managed. Such unforeseen issues can cause both major and minor interruptions in the progress and have to be resolved before moving forward. If not, workarounds have to be devised to ensure that the program gets back on track.

The other area of concern is the human factor. The SDKs need to be continuously upgraded to ensure ease of use by the end-users and also to be in sync with the latest technologies available. This is sometimes a steep learning curve for development teams and results in project slippages due to the extra time taken to get familiarized with the tool. Employee attrition is the other major human resource challenge. Employee turnaround at a critical stage of implementation can upset the schedule of the entire program. So employee management becomes critical for the program to stay on schedule.

Apart from retaining the employees, their engagement also is critical. Product redesigns lead to frequent changes in the schedule. Developers and design architects have to revisit the issues and re-estimate the effort. Sometimes they end up changing and discarding the earlier efforts. This leads to loss of morale and interest in the project. It is the role of the program manager to ensure that the employees are motivated throughout the program and the momentum is maintained. The team members have to be made aware of the exact scope of ownership of their area of expertise and if required need to be trained so that the program is not compromised.

### *Appraisal stage*

Program appraisal is a continuous process that is performed at the end of every stage. Though the final appraisal is based on the released, well-working product, various milestones and performance standards would have been defined (in the planning stage) to continuously monitor and control the program.

In our experience, tracking the performance of a project in terms of the completion of certain scope at the end of a stage is not always accurate in software development projects. Even if the stage is complete, there is every possibility that the stage is revisited until the client's expectations are fully met. For example, in the implementation stage, software program coding is an important milestone. An accurate assessment of the project's (and program's) performance is only possible after a code freeze.

The other challenge at this program stage is that of measuring employee's performance. Given that software development is teamwork, identifying the individual's contribution to the team effort is difficult. So much so, that the team members perceive their performance ratings to be biased, which in turn leads to employee dissatisfaction.

### *No formal closure*

In the case of ACME, there is no formal program closure. This is because software development is an ongoing process. Even as a version (of the software) is released, the teams are engaged in the development of the next version of the software. No software is perfect in terms of defects

and bugs. In addition, some of the functionalities may be too complex to be included in the earlier versions, and so have to be added as enhancements over a period of time. Apart from the technical factors, the teams also track the changing business requirements of their customers. New functionalities to be included in the software are reviewed and, if required, modified to meet customer expectations. Thus reviewing, fixing the bugs, re-evaluating, enhancing, optimizing and streamlining the software are ongoing processes that keep the teams working on the projects continuously. In short, it is a cyclical process that keeps the program always on. Hence there is no clear-cut termination or closure of the programs once the software is released into the market.

### *Overcoming challenges*

As discussed above, there are several ‘project management’ and ‘people’ challenges with this program management approach. Over a period of time, ACME has developed well-defined processes to overcome these challenges. These processes are continuously perfected to meet the changing demands of the program. Some of the processes are described below:

#### *Continuous review and coordination within projects and program*

Each team spends several person-hours each week reviewing progress. This is done both at team and project levels. There are several advantages that are associated with this process. Firstly, everyone in the team is aware of the current stage of the project or program at their own level. If not at a detailed level, at least at an overall level, each member is aware of the role and progress of their peers at the team level. This especially addresses the issue of employee attrition, where a team member is able to step into the role of a peer if needed.

This coordination between the team members is also extended to the program level, where each team is aware of their responsibilities, and dependencies on the other team. This interaction also helps teams discuss any overlapping issues that need the attention of both teams. Also, there is some degree of flexibility within the program stages to perform the project activities. For example, if the QA team has to test a certain functionality, say ‘Func A’, on a given date but is unable to develop the functionality, it can decide to work on another functionality, ‘Func B’ that is ready to be tested while work progresses on ‘Func A’.

Also, work on the various projects is paced such that the critical activities, and the expected benefits to be realized from the program are distributed throughout the program cycle. This is done to spread the technical, people, and financial risks associated with program (through its cycle).

#### *Resource management – training new resources to step in*

Even as the development teams are working on the projects, new hires and new members – even as they are being trained – are asked to assist the teams at various levels such as project and program documentation, writing technical manuals, code reviews, testing, etc. This helps in having a pool of resources who can step into the program with some level of awareness if required at any stage.

### *Continuous interaction with stakeholders at critical stages*

At some critical stages, stakeholders and customers are brought in to have a ‘sneak-preview’ as to how the product is shaping up. Their feedback is taken and used as inputs. This gives an outsiders’ perspective on the product, which is very important for the program.

### *Partners’ inputs*

Just as customers are given previews, partners are also involved in the testing phase. This is because it is the partners who recommend ACME’s products to their clients. Partners are likely to have an idea about what customers would like to have from ACME’s solutions. So the partners’ contribution at the testing stage is considered invaluable for the programs.

### *Motivating the team members to keep up the momentum*

As with any program, it is important that the motivation levels of the teams are kept at high levels to ensure that the momentum is maintained on the projects. Hence, the program managers, project managers and senior project managers collaborate with human resources from time to time to conduct workshops, training and team-building activities. This ensures better cohesion among the team members.

### *Continuous review of processes – refining the process*

Though the processes are well established, each process is reviewed from time to time so as to fine tune them in case they are proving to be bottlenecks instead of being enablers. This helps in proactively managing and reengineering any redundant processes.

### *Continuous review of technology to improve the usability of the software*

To ensure that technology acts as a tool for faster development, SDKs are updated by incorporating easy-to-use functionalities so that developers need not spend too much effort in learning to use the technology after each upgrade.

## **Conclusions**

In theory and practice, project management today is regarded as a strategic function. While the relation between the organization’s strategy – programs and portfolios – and projects has been well documented through studies based in the developed countries, more research is warranted from the emerging economies. This article describes the program management approach of a global information technology company.

This case study brings to the fore challenges that are specific to coordination of: (i) processes, and (ii) people, when managing a program. We have observed the following patterns with respect to these challenges across the program's lifecycle:

1. As the degree of uncertainty is higher in the initial stages of the program (design and planning), setting the objectives for the program by successfully managing the scope, and the coordination between the technical and non-technical project team members is critical.
2. In the program implementation stage, managing unsystematic risks becomes a critical factor in the successful execution of the multiple projects and so the program. This problem is more evident if the program scope is not well defined in the planning stages. While this leads to rework (and hence schedule and budget overruns), it also affects the morale of the team members.
3. In the program appraisal stage, recognizing the individual contribution of the team member vis-à-vis the team effort is a challenge.

Thus, the coordination between the stakeholders and processes of the program need be closely monitored across the program's lifecycle. This then necessitates continuous interaction between the various stakeholders of the program so that: (i) the program processes can be aligned to the changes in the internal or external environment in which the program operates, and (ii) all the program resources are optimally utilized.

## References

- Blomquist, R., Mueller, R. 2006, *Middle Managers in Program and Portfolio Management Practices, Roles and Responsibilities*, Project Management Institute, Newtown Square, PA.
- Bogdan, R. & Biklen, S. 1992, *Qualitative Research for Education*, 2nd ed., Allyn & Bacon, Boston.
- Cerpa, N. & Verner, J.M. 2009, Why did your project fail?, *Communications of the ACM*, vol. 52, no. 12, 132-134.
- Charette, R. 2005, Why software fails, *IEEE Spectrum*, vol. 42, no. 3, 42-49.
- Cooke-Davies, T.J., Crawford, L. & Lechler, T.G. 2009, Project management systems: moving project management from an operational to a strategic discipline, *Project Management Journal*, vol. 40, no.1, 110-123.
- Denzin, N.K. & Lincoln, Y.S. 2005, *The Sage Handbook of Qualitative Research*, 3rd ed., Sage Publications, Thousand Oaks, CA.
- Eisenhardt, K.M. 1989, Building theories from case study research, *Academy of Management Review*, vol. 14, no. 4, 532-550.
- Kappelman, L.A., McKeeman, R. & Zhang, L. 2006, Early warning signs of IT project failure: the dominant dozen, *Information Systems Management*, vol. 23, no. 4, 31-36.
- Lawrence, P. & Scanlan, J. 2008, 'Planning in the dark: why major engineering projects fail to achieve key goals, *Technology Analysis and Strategic Management*, vol.19, no. 4, 509-525.
- Lee, A. & Baskerville, R. 2003, Generalizing generalizability in information systems research, *Information Systems Research*, vol. 14, no. 3, 221-243.
- Morris, P., & Jamieson, A. 2005, Moving from corporate strategy to project strategy, *Project Management Journal*, vol. 36, no. 4, 5-18.
- OGC 2007, *Managing Successful Programmes (MSP)*, The Office of Government Commerce, Norwich, UK.
- Partington, D. 2000, Implementing strategy through programmes of projects, in R. Turner & S. Simister (eds), Gower Publishing, Aldershot, UK.
- Pellegrinelli, S. 1997, Programme management: Organising project-based change, *International Journal of Project Management*, vol.15, no. 3, 141-149.
- PMI 2006, *The Standard for Program Management*, Project Management Institute, Newtown Square, PA.

- Shao, J., Turner, R. & Müller, R. 2009, The program manager's leadership style and program success: a literature review and research outline, *Proceedings of IRNOP IX Research Conference*, Berlin, 11-14<sup>th</sup> October.
- Shehu, Z. & Egbu, C. 2007, The skills and competencies of programme managers, *Proceedings of RICS-COBRA 2007 Annual Conference*, 6-7 September 2007, Atlanta, Georgia.
- Shehu, Z. & Egbu, C. 2010, The nature of programme management and how they differ from project management, <http://zshehu.com/Papers/Programme%20Management%20-%20058%20-%20Revised.doc> (Accessed 25 March 2010)
- Thiry, M. 2004, Program management: a strategic decision management process, in P.W.G. Morris & J.K. Pinto (eds), *The Wiley Project Management Resource Book*, John Wiley & Sons, New York.
- Williams, D. & Parr, T. 2006, *Enterprise Program Management: Delivering Value*, Palgrave Macmillan, New York, NY.
- Yin, R.K. 2003, *Case Study Research: Design and Methods*, 3<sup>rd</sup> ed., Sage Publications, Thousand Oaks, CA.

#### **About the authors:**

**Vijay Somanchi** has a Master's degree in Computer Science from Texas Tech University and is currently doing his Executive MBA from IBS, Hyderabad. He has more than 10 years of experience in the IT industry, during which time he has worked in various roles such as Project Lead, Senior Consultant and Project Manager. Vijay is currently working as a Development Manager for an IT company in Hyderabad, India. His experience includes being part of two large ERP implementation projects as a Senior Consultant and Project Manager.

Email: [vijay.kks@gmail.com](mailto:vijay.kks@gmail.com)

**Ravikiran Dwivedula** has a PhD in Strategy, Program & Project Management from SKEMA Business School (former ESC Lille School of Management, France), and an MBA from Osmania University, India. He is an Assistant Professor of Human Resource Management with IBS-Hyderabad, India. He is also an Adjunct faculty and a member of Lille School of Management Research Group (LSMRC) at SKEMA Business School, France. His teaching and research interests are in the area of Project Human Resource Management, specifically on topics such as employee motivation, organizational and occupational commitment, and project success. He has widely published in his field. He is also a reviewer for the *Project Management Journal* and *International Journal of Project Management*. He is on the editorial board of the *Journal of Project, Program, & Portfolio Management*.

Email: [dvsravikr@ibsindia.org](mailto:dvsravikr@ibsindia.org)