Understanding the Dynamics of Fixed-price Programs

A Case Study Illustration and Generalization

Veerendra K Rai
Tata consultancy services
54-B, Hadapsar Industrial Estate,
Pune, MH, 411013, India.
veerendrak.rai@tcs.com

Sanjit Mehta
Tata consultancy services
54-B, Hadapsar Industrial Estate,
Pune, MH, 411013, India.
Sanjit.mehta@tcs.com

Abstract

This paper reports results of a study conducted to rescue a *fixed-price* program. The program was of strategic importance for the client as well as the vendor. The program was large and complex and was first of its kind to both client and vendor in terms of complexity and experience. The program was failing on all major fronts and existing operating models were unable to cope with it. Authors took systems approach to investigate and design solution to improve the state of the program. Problems facing the program were identified and As-Is state of the program was constructed. A system dynamics model was constructed and simulated with a set of policies to reach the desired state (To-Be state) and derive at a set of recommendations and design principles for the *fixed-price* class of programs. This study has endeavored to transcend this particular program in order to be useful for other programs in general.

Keywords: *Fixed-price projects, Systems approach, System dynamics, OCR program, Program management*

1. Introduction

Two of the world's largest financial institutions came together to form a joint venture (JV). This JV had three phases. The first phase was concerned with legal formalities. In the second phase the JV would have a combined payroll system and an increasing number of jointly available products and services. The third phase involves integration with the following objectives.

- Client and financial advisors satisfaction: ensure smooth transition and minimal disruption in an accelerated timeframe.
- Revenue retention: integrate the field and achieve goals of revenue production retention.
- Cost synergies: achieve more than $ one billion in cost synergies on or before pro-forma target dates.
- Platform integration: integrate platforms to efficiently meet the needs of the field.
- Product integration: create a joint product offering that meets the needs of financial advisors and clients.

In order to realize the above objectives, several new IT initiatives became imperative. A number of vendors from across the world are involved along with the IT organization of the client in this endeavor. A key foundation initiative among these arose from the need to expand the *Outlet Numbers* for the merged organization due to the addition in the number of *outlets*. The *Outlet Number* is a key field, which is used by more than 200 applications across the IT landscape of the new organization and is therefore fundamental to the merged organization achieving the new IT blue print.

Our organization, one of the leading IT service and business solution providers, was entrusted with the responsibility of undertaking this ***Outlet Code Restructuring (OCR) program***. This involved, identifying and remediating all impacted systems and business processes to enable addition of new outlets.

1.1 The OCR Program Scope

- Finalize application inventory
- Identify and analyze impact
- Design solution patterns
- Test strategy & planning
- Remediation of impacted applications
- Integration / system testing
- QA support
- Turnover & rollout support
- Program management
    - plan release cycles as per logical groups for implementation / roll out
    - tracking, monitoring, and reporting
    - stakeholder management

1.2 The Nature of the OCR Program

OCR program was a very large and complex, one of its kinds for the vendor relationship with the client. The peak time team size was more than 150. Never before, a program of this size and complexity was awarded by the client to the vendor on a ***Fixed-price*** basis. This program was unique and novel for the clients as well. Never before clients have gone through the experience of creating a joint venture of this dimension. Program team was spread across different continents, cities, time zones and cultures. Also, this involves working with a new set of client managers from the merged organization with whom our organization never worked before. In addition, this being a foundation program, the on-time completion was extremely critical since several other programs in the overall portfolio depended on its outputs. For us, being a preferred vendor for this customer, failure on this program could have an adverse impact on our long standing relationship and reputation with this customer.

1.3 State of the OCR Program

The OCR program was in severe risk zone when this study was conducted. The program was faltering on all of the following fronts:

- Stakeholder management
- Resource management

- Project management
- Operations management

2. Dynamics of *Fixed-price* Contracts

Fixed-price contracts involve setting a total ***Fixed-price*** for a defined product or service to be provided. Changes in scope could be accommodated, but generally at an increase in contract price (PMI, 2008). A number of variants of ***Fixed-price*** are being currently implemented in the industry. Convention wisdom holds that ***Fixed-price*** contracts are good when the risk is low (Turner and Simister, 2001). Turner and Simister (2001) argue that selection of contract type must be a function of uncertainty in the product (project deliverables) and uncertainty in the process of their delivery. A combination of high and low uncertainty in product and process dimension would call for different contract models.

From systems view ***Fixed-price***d contracts are a way to outsource project management complexities such as stakeholder management, cost and schedule management, and administrative overheads to a vendor for a **fixed price**, which is presumably reasonable. Precisely for this reason ***Fixed-price*** projects tend to favor the customer rather than the service provider. ***Fixed-price*** appears to be just an 'envelop' for a particular style of executing projects wherein nothing is *fixed* really and underlies a lot of turmoil. The main problem with ***Fixed-price*** projects is that they encourage us to do things which are not reasonable and eventually result in wastage of time and resources. For instance, greater the need for estimation accuracy the greater the need for detailed information about the project. As a result project leadership is tempted to take big requirement up front, a change management process to avoid scope creep, a big design up front, and a *sequential* software development lifecycle (Dr. Dobb's, 2007).

Full knowledge of requirements emerges as a result of discovery process wherein client interacts with project team. It takes several iterations before relatively stable set of requirements are understood. Requirements are not known upfront even to customers as assumed and required by big requirement upfront and big design upfront subsequently. Knowledge of requirements also influences accuracy of estimation.

Besides, specification, estimation and execution require sufficient experience in the domain (Cauwenberghe, 2006a). All of these require transfer of knowledge from similar projects in the domain besides other things. Because of these reasons, Cauwenberghe (2006a), suggests a set of criteria to decide whether a given project is fit for bidding under ***Fixed-price*** contract. These criteria cover all phases of ***Fixed-price*** contract process; qualifying, selling and implementing.

Change management, a strategy adopted for ***Fixed-price*** projects to manage change, does not always yield desirable results. This strategy is fast becoming a misnomer. Instead of managing change it prevents change. Requirements evolve and by preventing change in the name of preventing scope creep we resist change even if it is desirable in the overall interest of the project and thus increase the chance of project being delivered on schedule and budget (as it has not been allowed to admit change), but decrease the chance of stakeholder satisfaction. This appears to defeat the basic purpose of programs, which is not essentially about delivering product or service in time, but to create value to stakeholders. Cauwenberghe (2006b) proposes 'exchange request' in place of 'change request' to overcome the problem whereby a particular functionality is added only when another functionality requiring at least the same effort is removed first.

Change management is something that brings short term gains, but incurs disadvantages in the long run. *Short term gain at the cost of long term pain later* is a familiar pattern observed in many real life situations and has been discussed in detail in the works of (Senge, 1990) and (Weinberg, 1997). Sequential software development lifecycle induces Waterfall Model (Royce, 1970). It is impossible for any non-trivial project to proceed in strict sequential manner, as expected by Waterfall model, by freezing the current phase before proceeding to the next. In retrospect, all traditional techniques we are tempted to use on fix price projects do not seem to yield desirable result.

One reason for this could be not using the newer insights in project and program management. For instance, understanding a large program as system of systems. Rai and Swaminathan (2010) construct a program management framework taking system of systems approach. This framework treats each project in the program as a system and program as a whole as system of systems. Constructing a Portfolio management framework could be of good help in managing a collection of projects within a program (Rai *et al*, 2010). A well designed PMO (project management office) could be used to handle the complexity of managing a large and complex program (Rai, 2009).

2.1 Risk Management

Fixed-price projects could be a reward for vendors that excel in delivery excellence; failing could be a punishment as well. ***Fixed-price*** contracts are prone to risks and have low success rate. Competent risk management is one of the critical factors to enhance chances of success in ***Fixed-price***. Strategy to manage risk include risk transfer (to another party), risk mitigation, risk avoidance and risk acceptance. Several risk management standards have been developed over the years including standards by Project Management Institute (PMI, 2008), ISO standards (ISO; 2009a, 2009b). A major source of risk in project management is project costs, demand and other impacts. To avoid risk in ***Fixed-price*** projects it is worthwhile to consider if the project is in known domain and environment, project team is known, and project is of usual size vendor has worked before (Cauwenberghe, 2008a).

A promising new approach to mitigating risks is based on theories of decision making under uncertainty that won 2002 Nobel Prize in economics (Flyvbjerg, 2006). Risk management has come to occupy a central place in project management and has become a distinguished theme and idiom in the project management domain. "Arguably, with the exception of Risk Management no new principles of cost, design, or schedule control have been developed since Earned Value, Configuration Management, Value Engineering, Precedence Scheduling and Resource allocation in the mid-1960s (Morris, 1994, p. 217). Learning from experience has always been part of project management. 'Project Retrospectives' (Kerth, 2001) provides useful format to learn from experience. If project team has developed knowledge on handling new risks this knowledge could be recorded and preserved to be used later.

2.2 Agile Approach for *Fixed-price* Projects

Can agile methodology be used on ***fixed-price*** projects where scope and price are fixed upfront? The answer is 'No' if agile methodology is viewed as an approach for handling rapidly changing requirements. However, there is something else at stake in ***fixed-price*** projects and it is not the changing requirements. Efficiency is the key in ***fixed-price*** projects for rapid development. Time is money for the vendor in ***fixed-price*** projects and speed and quality of delivery are the primary drivers for successful execution of ***fixed-price*** projects. Cockburn (2004) remarks, he did not come to agile principles for the need of *handling rapidly changing requirements*, but through the need for *efficiency*. Agile approach was predominantly viewed as solution to the former and not to the latter. It does not, however, mean agile approach could be used on all ***fixed-price*** projects. Projects need to be analyzed for Agile-suitability before appropriate agile technique could be used.

2.3 Lean Development

Lean approach to development and management of projects is yet another approach that could come to the rescue of ***fixed-price*** projects. It not only helps in coping with changing requirements, but also in eliminating waste and facilitating rapid development- the attributes desperately needed for ***fixed-price*** contracts. Besides, lean thinking approach recommends seeing the whole. Poppendieck (2003) discusses lean software development methodology based on principles of lean thinking and observes that majority of software project management theories are based on theory of disaggregation, which promotes silo orientation and hence lack of emergence of big picture.

3. Commissioning of the Study

This study was commissioned with the mandate to review and study OCR program and provide, recommendations for improvements, which will help rescue the program and pull it out of the risk zone. A consultant (one of the authors of the paper) was engaged for the study.

3.1 Engagement Approach and Methodology

Consultant deployed **4D** methodology consisting of the following stages (Murthy, 1994). **Discover,** wherein discussions were held with concerned stakeholders; **diagnose** wherein key emergent themes were identified; **design & deploy** wherein recommendations were made for improvement. Experience from delivery excellence initiatives taken earlier were also pooled in to benefit from earlier learning.

4. Execution of the Methodology

4.1 The Discover Phase

Under discover phase, a discussion with leadership team was held to seek first-hand information on the issues and challenges facing the program. The semi-structured interview with the leadership centered on:

- Stakeholder management
- Resource management *(Experience and capabilities; Attribution of business analysts; Distribution of program team across locations)*
- Project management *(Effort estimation and tracking; Risk and issue management; Role of PMO)*
- Operations management *(Staffing)*

Table 1 below contains the main findings.

Table 1: Observation on the OCR program

S. No.	***Category***	***Observation***
1.	Stakeholder Management	• Initial lack of sufficient ownership and commitment from the client program management team
2.	Stakeholder Management	• Not all client application managers bought-in to vendor's remediation approach and some of them were averse to vendor taking complete ownership of this initiative.
3.	Resource Management: *Experience & Capabilities*	• Quality of onsite coordination was a concern
4.	Resource Management: *Experience & Capabilities*	• Limited number of vendor Subject Matter Experts (SMEs) in the OCR program team leading to high dependence on the SMEs from Non-OCR projects in the vendor organization.
5.	Resource Management: *Experience & Capabilities*	• Lack of application knowledge within the vendor's team on more than 50% of the applications in the inventory
6.	Resource Management: *Experience & Capabilities*	• Lack of sufficient experience in execution of large critical Fixed Price (FP) projects within vendor organization's project managers and project leads in the OCR program
7.	Resource Management: *Attrition of Business Analysts*	• Team staffed with a high proportion of trainees leading to low throughput • Low level of technical capabilities of the team
8.	Resource Management: *Distribution of program team across locations*	• Offshore team distributed across two cities in the country in which vendor organization is located and this was an overhead and led to inefficiencies in the process • Mainframe quality analyst being out of project location was a concern
9.	Project Management related: *Scope Management*	• Tendency to accept work from client not part of the *Task-order* responsibilities. This unduly increases the scope of the work.
10.	Project Management: *Effort Estimation and Tracking*	• Remediation effort and complexity underestimated • There is a risk of running out of project cost
11.	Project Management Related *Risk and Issue Management*	• No formal risk and issue management process was in place, though issues were being discussed in the daily calls
12.	Project Management Related *Role of PMO*	• PMO was in a reactive role, capturing data and preparing reports in addition to performing some operational work • Lot of subjective judgment in assigning the Red-Amber- Green (RAG) status of the projects and predicting the completion of milestones.
13.	Operations Management *Staffing*	• Delays in staffing the open positions was affecting the project timelines • There were 70 open positions in the program. Resource management Group was on this task but a more coordinated effort with a task force was required owing to the gravity of the problem.

4.2 The Diagnose Phase: Establishing Etiology of the State of OCR Program

Based on the observations made in the 'discover' phase as reported in table 1. This table tells the current state of affairs of the system and describes the mess the system is in. Gharajedaghi (2006) defines mess as a system of interrelated problems and elucidates its formulation and representation. The mess can also be modeled as causal loop diagrams (Sterman, 2001).

4.3 Empirical Validation of Diagnosis using System Dynamics

The purpose of System Dynamics Simulation is to provide an empirical structure to capture the observations using a system dynamics model and validate the diagnosis through simulation of the model. The model is constructed based on observations captured by the author using the influence diagram. The model is simulated using assumed data and proportional relationships which are consistent with the relationships among entities shown in the influence diagram, which in turn are verified through existing literature or by the author's own experience gained while conducting the study.

The results obtained from the simulation are divided into two phases - a) the as-is state of the program, which represents the state, the program is in, at the time of conducting the study & b) the to-be state of the program, which represents the expected state of the program if the recommendations suggested by the author are implemented over time. Rai and Mehta (2012) outline a systems thinking based methodology for assessing the As-Is state of a service engagement so that policy intervention could be made to ensure service engagement reaches the desired (To-Be) state.

4.3.1 System Dynamics Model Conceptualization

4.3.1.1 Scope

This section describes the system dynamics model for the OCR program. The design pattern described in this paper can be extended to model fixed-price programs in any given context. A fixed-price program initiates at the estimation level. The client provides scope and time for project in a request for proposal (RFP) document. The vendor estimates effort required to fulfill client requirements. Fixed-price quoted for the project is a sum of fixed costs, effort costs and profit margin. The objective of the vendor is to execute the program such that it fulfills client objectives at optimal costs. We use this model to identify and demonstrate policies to fulfill client and vendor's objectives for the OCR program. These policies are generalized to formulate guidelines for execution of fixed-price programs.

4.3.1.2 Model description

The core structure for the model is the service delivery process. Service delivery process consists of activities required to provision services defined in the contract. Figure 1 outlines the core activities in OCR program.

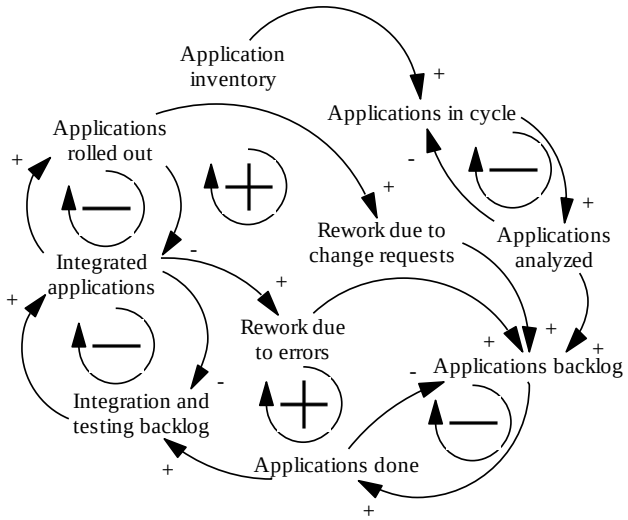

Figure 1. Core service delivery structure

Service delivery starts with an inventory of applications to be remediated. Applications are remediated in work cycles wherein each cycle consists of a set of applications. Applications are analyzed to determine work to be done and their impact on the overall system (other applications). Applications accumulate in a backlog for remediation. Remediated applications accumulate for integration and testing. Erroneous work items are added to the backlog while others are rolled-out. In this context, roll-out implies deployment and demonstrations to customer to gather feedback. The client may require certain changes (change requests), i.e. addition, deletion and/or modification to the work items which are re-added to applications backlog.

The following service delivery parameters are required as input.

- Number of applications to be remediated at the start of project.
- Modularity – Applications are grouped logically based on functionalities and inter-dependencies. Modularity is the average number of applications in a logical group.
- Duration of the project.

We have defined three service delivery activities – Impact analysis, Remediation and Integration and testing which correspond to OCR service delivery process. Activities are executed by resources (employees). We have defined four resource roles in our model – Consultant, Senior, Junior and Trainee. These roles correspond to generic employee types which are part of any service engagement, including OCR program. Consultants identify client's requirements and objectives, provide solutions and manage relationships between client and vendor. Senior resources are associate consultants and subject matter experts with requisite knowledge of client applications and environment required to execute service delivery process. Junior resources with experience greater than one year, execute core work items (Remediation and integration). Trainees are non-experienced employees recently inducted into the organization. Cost per resource increases as role progresses from trainee to consultant. Effort required per application is estimated for each activity and resource role. It is to be noted that the set of activities and resources can be changed w.r.t. context for any service engagement.

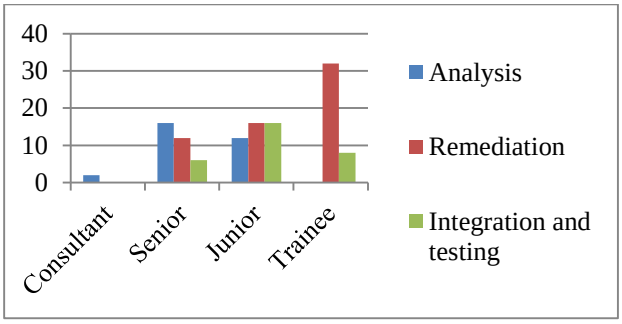

Figure 2. Effort estimation per application per activity

Resources are spread across locations. We define two location types in the model – Onsite and offsite. The metric, onsite-offsite ratio is input for each activity and resource role to determine resource distribution across locations. Fixed price quote is computed based on work volume and effort estimate.

Applications are rolled-out in cycles. Cycle time is provided as input. A longer cycle time would create a large backlog of applications which requires more time to resolve. It delays the feedback derived from application roll-out. It can be inferred from figure 3 that shorter cycles would improve customer's knowledge of requirements at a higher rate. This results in lesser change requests over time as compared to bulk change requests, if cycle time is longer.

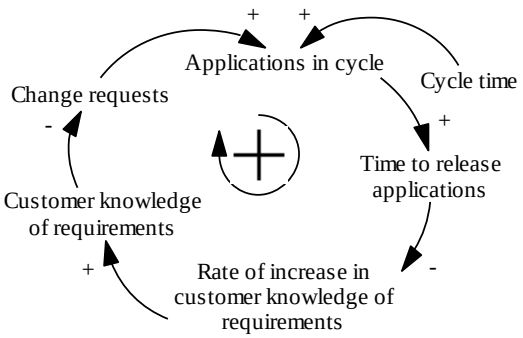

Figure 3. Effect of cycle time on clarity of requirements

Applications move from one activity to the other based on productivity at each step. One of the factors affecting productivity is resource distribution. Resource distribution at start of project is provided as input. Resources are distributed across activities at each time step based on the backlog for each activity.

*Resources in team [Role, Location, Activity] = Resources in team [Role, Location] * (Effort required for activity [Role, Location, Activity] / (Total effort required))* (1)

Effort required for an activity is the product of application backlog and effort per application per activity. Role, location and activity represent arrays consisting of elements described earlier.

Productivity is defined as the number of applications completed per unit time for each activity. If all resources would be of same type, then productivity for an activity would be computed as,

Ideal productivity [Activity] = Effort required per application [Activity] / Total resources in team [Activity] (2)

However, resource types are different and therefore we use the following procedure to compute productivity for each activity.

- Compute total resources across roles and location for each activity.
- Compute a desired distribution (of total resources) based on ratio derived from effort required per application for each activity.
  *Desired resource distribution [Role, Location, Activity] = Total resources [Activity] * (Effort required per application [Role, Location, Activity] / Total effort required per application [Activity])* (3)

- A base set of resources for each activity is computed from resources available for that activity. The base set is the largest possible number of resources across roles and locations which conform to the desired resource distribution. For this set, the productivity is equal to,
  Base set productivity [Activity] = Number of resources in base set [Activity] / Total effort (person-days) required per application [Activity] (4)

- Excess resources (difference between resources in team and base set) also have a marginal effect (positive or negative) on productivity for the activity. The deviation of excess resources from ideal distribution is computed. It would have surplus of some resource role and locations and deficit of others. A marginal effect co-efficient (user input) reflects the importance of a resource role in an activity. For example, decrease in number of consultants for impact analysis cannot be compensated through increase in number of trainees (Marginal effect = 0). If the sum of marginal effects (deficit as negative and surplus as positive) is positive, then it is added to productivity.
  *Marginal change in productivity [Role, Location, Activity] = Marginal distribution of resources [Role, Location, Activity] * Marginal effect co-efficient [Role, Location, Activity]* (5)

- Net productivity is base set productivity after incorporating the effect of marginal change in productivity.
  *Productivity [Activity] = (Base set productivity + (Sum (Marginal change for all roles and location [Activity]) / Total effort (person-days) required per application [Activity])) * Resource productivity per day* (6)

For the 'Application analysis' activity, customer's (or client's) knowledge of requirements is required to outline work items for each application. Lack of knowledge would reduce productivity of analysis by a factor of,

*Reduction in productivity due to customer knowledge of requirements [Analysis] = (1 - Customer knowledge of requirements ^ 5) * EXP (-0.5 * Customer knowledge of requirements) * Maximum impact of customer knowledge of requirements on productivity* (7)

Lack of knowledge of requirements increases number of change requests. Knowledge of requirements improves as applications are rolled-out. Shorter cycles rapidly increase customer's knowledge of requirements thus reducing change request volume.

Resources work for a fraction of day. Therefore, the ideal resource productivity per day is less than 1 (Equation 6). We input ideal resource productivity per day as 0.375 (9 hours per day).

Each activity accumulates work completed in respective level variables. Analyzed applications are accumulated in 'Analyzed applications'. Remediated applications are stored in 'Applications done' and 'Integrated applications' stores applications post integration and testing. Flow controls are defined for each level variable. Flow control implies the number of application groups that should be accumulated at an activity before advancing to next activity. In a sequential process, the flow would be equal to the number of applications in cycle, i.e. the next step starts only after the previous one is complete. In a parallel flow, work items advance to next step on completion of current activity. This results in parallel execution of activities, and shorter release cycles.

In a given cycle, the desired level of productivity is computed from the number of applications in backlog and the cycle time due. Negative discrepancy in productivity would result in more time required (than available) to clear the backlog. This creates schedule pressure. We model schedule pressure as a negative exponential function (Equation 8) since the 'Min' (minimum of two arguments) function would always return a zero or negative value for 'Productivity discrepancy fraction'. It is to be noted that positive 'Productivity discrepancy fraction' results in no schedule pressure.

*Schedule pressure [Activity] = (1 - EXP (5 * MIN (Productivity discrepancy fraction [Activity], 0)))* (8)

Schedule pressure causes resources to increase effort by working overtime, which increases resource productivity but results in exhaustion of resources. Exhaustion subsides by itself over a period of time.

Exhaustion = ∫((((Resource productivity – Ideal resource productivity) / Ideal resource productivity)

/ Exhaustion accumulation time) - Min (Exhaustion, (1 / Time to relieve exhaustion))) (9)

In the model, we limit the overtime to a maximum value of 'Ideal resource productivity' (9 hours per day). Therefore, resource productivity cannot exceed twice the ideal resource productivity. Here, 'Exhaustion accumulation time' implies the number of days a resource is able to work at maximum productivity before being fully exhausted. 'Time to relieve exhaustion' is the number of days required to mitigate exhaustion. Resource exhaustion proportionately decreases overall productivity of resources (burn-out). Therefore, an optimal schedule pressure (and cycle time) is favorable to maximize average resource productivity per day (Oorschot et al). A detailed discussion on schedule pressure in software projects is found in Rai and Mahanty (2001). Figure 4 shows the dynamics of schedule pressure. Higher schedule pressure introduces errors in work. These errors are identified during testing and added to backlog as re-work. Margins of error (maximum values) are defined for all activities – 'Analysis', 'Remediation' and 'Integration and testing'. The actual margin of error for these activities is defined as,

*Margin for error [Activity] = Schedule pressure [Activity] * Maximum margin for error [Activity]* (10)

Errors caused during analysis phase result in incorrect requirements capture. These manifest as rework due to change requests during roll-out of applications. Errors introduced during remediation are intercepted during the integration phase. However, due to a propensity for error (Equation 10) at the integration stage, not all errors are identified. Some errors remain unidentified and are rolled-out with applications. This reduces the quality of deliverables. Therefore, schedule pressure has a positive impact on productivity through overtime but negative impact on the quality of deliverables as well as increased volume of rework.

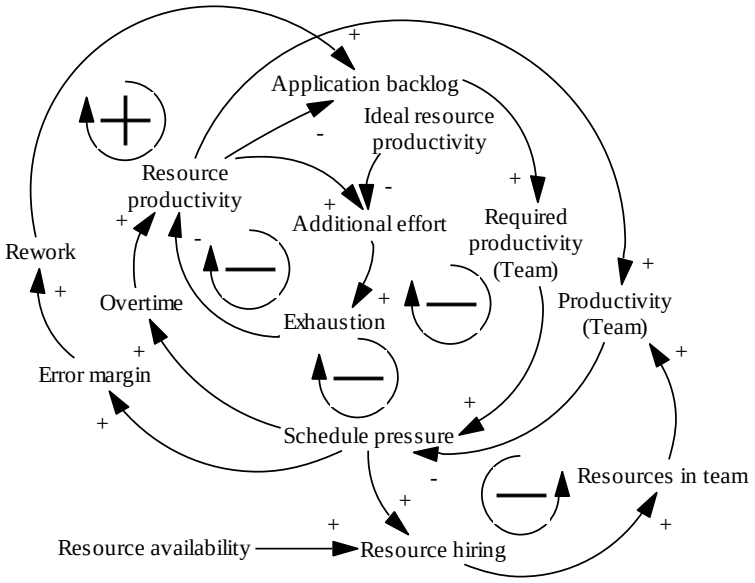

Figure 4. Schedule pressure and resource hiring dynamics

Resources are hired to increase effort available during cycles, thereby increasing productivity and mitigating schedule pressure. Hiring is based on the required distribution of resources across roles and locations, provided as input by the user. The number of resources to be hired for each activity, irrespective of roles and location is determined by the discrepancy in productivity.

*Resources to be hired [Activity] = Resources in team [Activity] * (-1 * MIN (Productivity discrepancy fraction [Activity], 0))* (11)

The desired distribution is super-imposed on the number of resources to be hired.

*Resource distribution for hiring [Role, Location, Activity] = Resources to be hired [Activity] * Effort allocation [Role, Location, Activity]* (12)

Resource requirement is the difference between resource distribution for hiring and resources available. Hiring is subject to availability of resources and a fixed amount of time required to on-board resources. The time required to on-board resources also includes the reaction time of the PMO. A reactive PMO would lack forward visibility and cause longer delays in filling open positions. 'Resource availability' is defined as the number of resources of a particular role and location suited to the engagement available within a user defined period of time ('Resource availability duration').

Resources hired [Role, Location] = ZIDZ (MIN (Resources in pipeline [Role, Location], ZIDZ (Resource availability [Role, Location], Resource availability duration [Role, Location])), Resource on-boarding time) (13)

Here, 'ZIDZ' is a division function which returns zero, if denominator is zero. 'Resources in pipeline' is an accumulator variable which maintains the count of resources required for each combination of role and location across all activities. Hired resources are added to team. 'Resource attrition' modeled in the system reduces the number of resources over time.

Resources learn through experience over time. We model work experience for each activity as a function of the work volume completed for each activity. Resource addition and attrition affects competency as well. New resources are inexperienced and therefore reduce competency of the team. Resource attrition drains existent competency. The net competency for a team is therefore the product of work experience and average team experience as elucidated in figure 5.

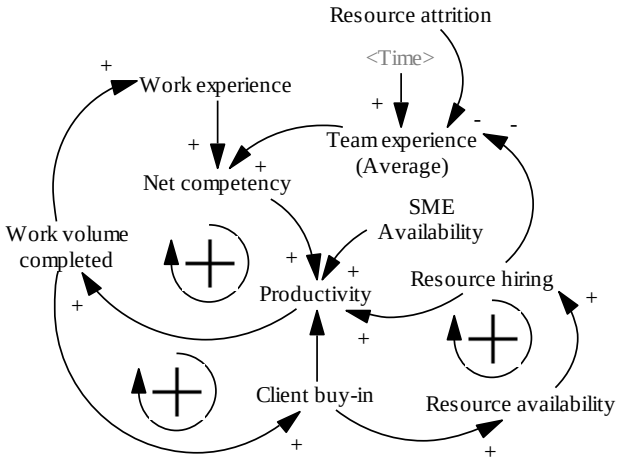

Figure 5. Competency and client buy-in dynamics

Discrepancy in team competency is addressed by subject matter experts (SME). SME may be from the vendor team or client team. Lack of SME in vendor team is compensated by client SME. This is subject to client's ownership and commitment (client buy-in) to the project. Client buy-in also improves hiring flexibility. During resource hiring freeze or unavailability of resources, client intervention would lead to resource sanction from management layer or access to client's resources. Application remediation generates value for the client. Client buy-in increases over time as work is accomplished, thereby increasing value derived from the project. This constitutes positive feedback loops through reduced competency discrepancy and increased resource hiring flexibility. Client buy-in also increases through regular meetings, status reporting and addressing client concerns. These factors are not considered in this model. It can be inferred that shorter release cycles would improve client buy-in at earlier stages of the project. This positively impacts overall project performance.

The cost model computes costs for the project. Fixed price quote is computed for the project based on effort required for the project and a pre-defined margin. Fixed price may be altered during the course of the project, if change requests are billed. A parameter 'Percentage changes billed' moderates the number of change requests that are actually billed. Over the course of project execution, resource costs accumulate. Vendor profit decreases gradually till it reaches a constant value at the end of project. A parameter 'Paid overtime' factors in overtime costs, if any, as set by the user. Resource hiring is constrained by surplus budget available. This feature may be activated or deactivated by the user through 'Cost constrained hiring' parameter. Surplus budget at a given point in time is computed as defined in equation 14. A causal diagram for cost dynamics is shown in figure 6.

*Surplus budget = MAX (Fixed price * (1 – (1 – Margin compromise factor) * Margin) – Total resource costs – Resource costs per unit time * Expected completion time, 0)* (14)

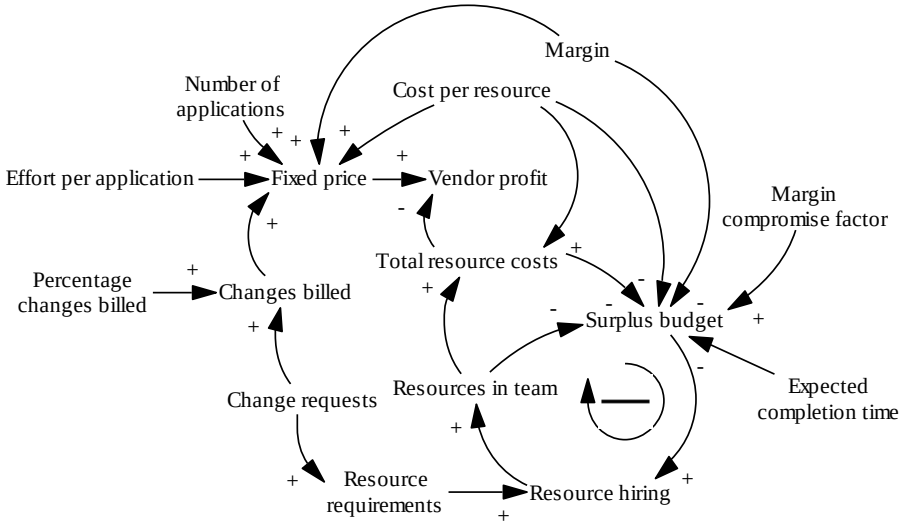

Figure 6. Cost dynamics

4.3.1.3 As-Is State Simulation

The system dynamics model for OCR program as described above is simulated to gain insight into program behavior over time. Simulation further provides empirical validation of diagnosis. The as-is state model simulation represents the current state of the program. Based on observations available from diagnosis, we configure the following data parameters (Table 2). These data parameters represent facts of engagement and therefore remain constant during the configuration of to-be state.

Table 2.Data parameters for OCR simulation model

Data parameter	Value
Number of applications	180
Project duration	720 days
Effort requirements (per application)	Refer Figure 2
Initial resource distribution	Refer Figure 7
Initial competency	50% of required level (For all activities)
Margin	15%
Overtime paid	False
Resource availability (Flexibility to hire resources)	Very Low
Modularity of applications (group size)	5 applications
SME Availability	25% of required SME (Low)
Initial client commitment and knowledge of requirements	Low
Resource attrition	10% per year

Figure 7. Initial resource distribution

In addition to data parameters described above, the as-is state of OCR engagement has parameters which are configured based on context summarized in table 1. Table 3 outlines the configuration of the as-is state of OCR model. The values for these parameters are derived from the diagnosis of OCR engagement in the as-is state.

Table 3.Configuration parameters for OCR as-is state simulation

Configuration parameter	Value	Reason
Percentage changes billed	15%	Undue increase in scope of work without accommodating fixed-price or schedule change.
Proactivity	Low	Reactive PMO implies low foresight into work pipeline and resource needs.
Cost constrained hiring	True	Resource hiring is subject to budget availability.
Margin compromise factor	100%	In a worst case scenario, the margin may be compromised. However, budget availability constrains hiring of new resources in team.
Under estimation	10%	Project costs underestimated by 10%. Lack of experience in executing large scale fixed price projects.
Application groups worked on per cycle	3	Tendency to assume large amount of work upfront.
Flow control values	3,1,1	Waterfall type model wherein requirements are analyzed upfront. Remediation and integration are executed per group.
Cycle time	60 days	Longer cycle time delays feedback.

Table 2 and 3 cumulatively cover a section of diagnoses for the OCR program. Parameters pertaining to communication, co-ordination and reporting have not been covered in the model. The model is simulated for a period of 1500 days. The graphs below outline the predicted behavior of OCR program for its entire duration.

Based on the above defined configurations, we compute a fixed price quote for the project at $11.01 million. Change requests billed over time increase the price quote to $11.82 million by the end of simulation. The project exceeds this budget and incurs a loss of $800,000. The project exceeds defined timeline by 215 days (Figure 8).

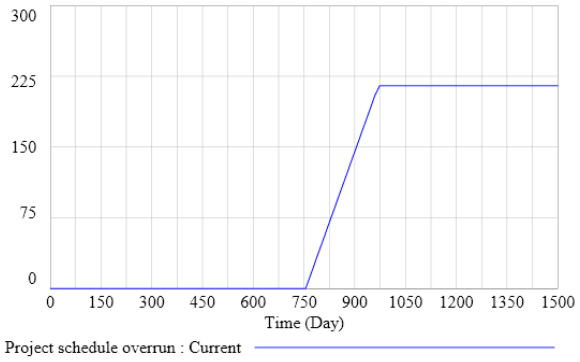

Figure 8. Project schedule overrun

One of the causes for schedule overrun is lower productivity. Figure 9 shows the deviation in productivity from the ideal. Productivity for a particular activity is calculated as applications completed (for that activity) per unit time. Low productivity is a result of higher proportion of junior as compared to senior team members. As described above productivity is computed as a sum of base productivity and marginal productivity. Base productivity refers to the maximum subset of available resources which meets the desired distribution for the activity. Additional resources may contribute marginally to the productivity, if the sum of marginal effect of all resources is positive. Figure 10 shows the number of resources in team to the sum of base and marginal number of resources actually used for work. The gap

highlights less than 100% resource utilization. This is because the absence of senior employees for certain activities is not compensated through junior employees and trainees.

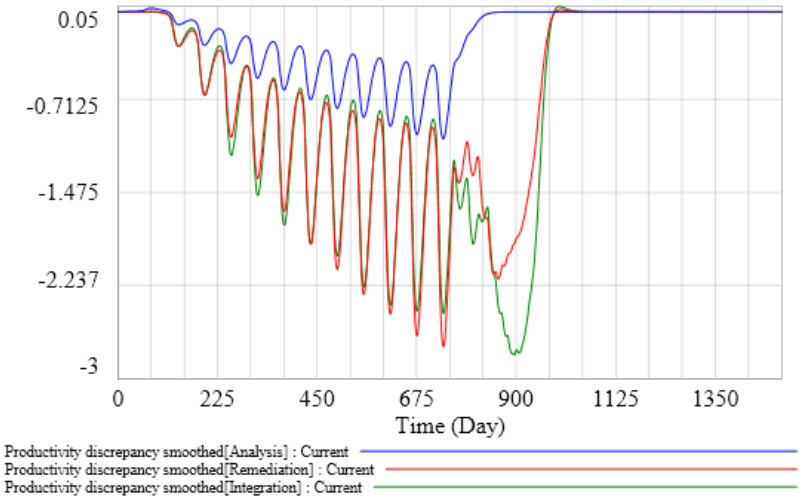

Figure 9. Productivity discrepancy (Applications per activity per unit time)

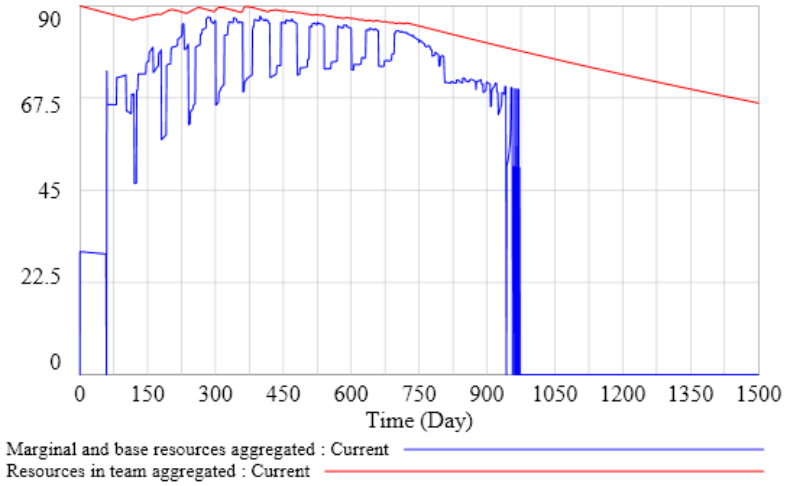

Figure 10. Productive set (Base and marginal) of resources vs. resources in team

Resource attrition with low availability for hiring reduces the number of resources in team over time (Figure 10). Resource attrition reduces the rate of improvement in team competency through experience (Figure 11). This combined with scarcity of SME significantly impacts productivity. Client commitment to project ideally reduces overheads due to lack of competency. However, since applications are not organized based on their potential for value generation therefore client buy-in increases almost linearly over time. The linear behavior of client buy-in (Figure 12) also reduces hiring flexibility and delays improvement in productivity.

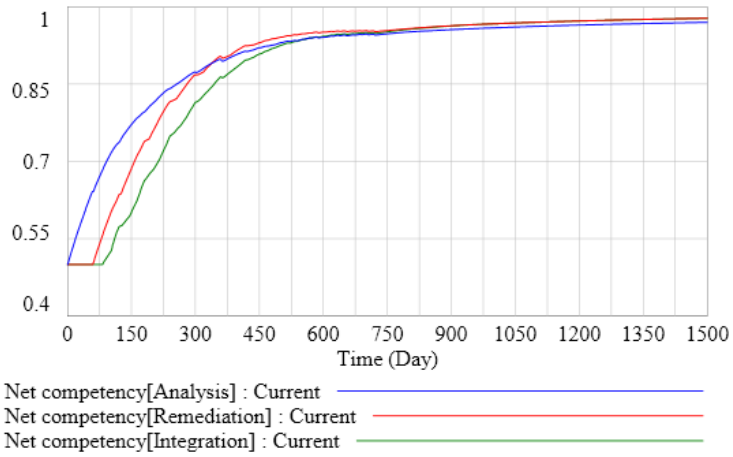

Figure 12. Team competency across activities

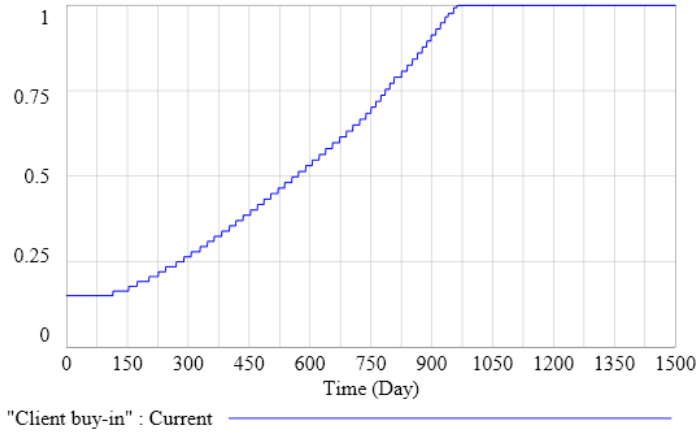

Figure 12. Client buy-in

Discrepancy in productivity generates schedule pressure (Figure 13). Due to longer cycle time, schedule pressure takes time to develop. This delays the individual productivity spurts through extra working hours. Schedule pressure remains nearly constant till end of project. This increases the team's propensity for error causing re-work (17% of total work) and error-prone releases. Re-work is measured as the equivalent number of applications worked upon in addition to the application inventory. Effort invested in re-work is the product of application equivalent re-work and effort required per application. Error-proneness of releases is also aggregated as application equivalents. For the OCR program, 1.827 application equivalents are error-prone. Mapping these errors to individual applications is out-of-scope for this model. We can state that, from an aggregate perspective, ~2% of work done is error-prone.

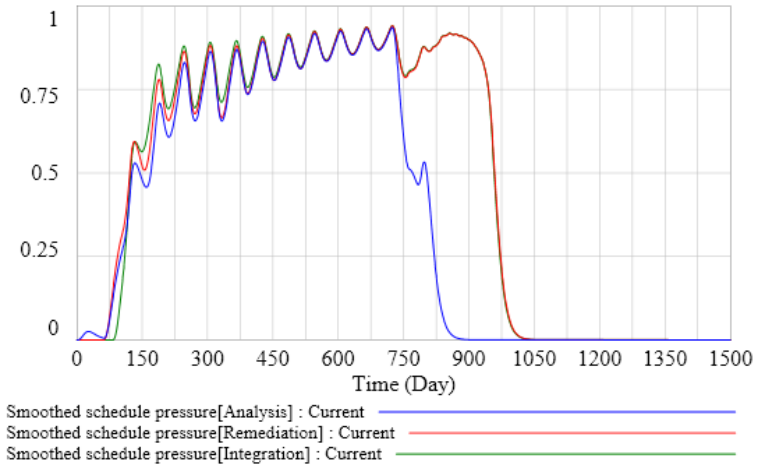

Figure 13. Schedule pressure

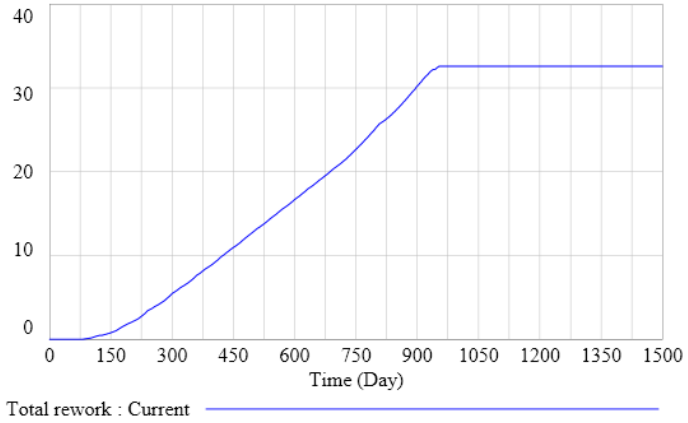

Figure 14. Total rework effort (as application equivalents)

In the current context, the client's knowledge of requirements is uncertain. Requirements are discovered (added, modified or deleted) through frequent roll-outs. This leads to change requests. Longer cycle time delays application roll-outs resulting in delayed discovery of requirements and more change requests. Figure 15 shows the improvement in client's knowledge of requirements over time. Despite improvement at earlier stages, the total number of change requests resolved by the team amounts to 80 application equivalents. Change requests costs $5.403 million, of which $0.81 million is billed and the rest depletes the budget of the project (41%).

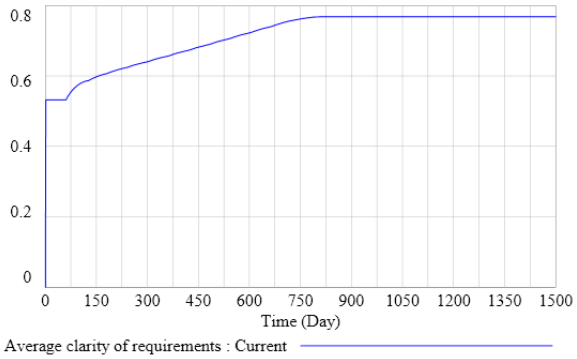

Figure 15. Client's clarity of requirements

It can be inferred from simulation results that unfavorable context, inefficient project management and governance and incorrect estimation led to the current state of the OCR program. Simulation results predict that if the program continues in current state, it would incur losses of 6.8%, exceed timeline by 30% and 2% of total applications would be error-prone

5. Design & deployment phase

5.1 Solution Design

Solution design consisted of a set of recommendations and design of lean PMO for the program. The set of recommendation was tailored to the specific problem areas of the program. Design of PMO is given in the next section.

5.2 Expected To-Be State of the Program

This section exploits the OCR system dynamics model to predict expected behavior of the project if recommendations are implemented. The model to-be state is configured to reflect fixed-price program values outlined in Table 4. Table 5 shows the to-be state configuration alongside as-is state configuration for the OCR program.

Table 5.To-Be state configuration

Configuration parameter	As-Is value	To-Be value	Reason
Percentage changes billed	15%	25%	Change in scope should be accommodated. However, accommodation should entail billing a portion of change requests as well schedule extension. This would incentivize the vendor to implement changes and fulfill client's objectives from the project.
Proactivity	Low	Medium	Increase level of proactivity through meetings, processes and reports.
Cost constrained hiring	True	False	Cost constrained hiring proves to be counter-productive. 'Time is money' idiom encourages improvement in productivity at the expense of cost.
Margin compromise factor	100%	100%	Compromise on the margin for scheduled completion of project.
Under estimation	10%	0%	Project estimation should be accurate as it is non-negotiable later under normal circumstances.
Application groups worked on per cycle	3	2	Reduce number of applications per cycle and cycle time.
Flow control values	3,1,1	1,1,1	Applications are remediated frequently on a per group basis.
Cycle time	60 days	40 days	Shorter cycle times facilitate quick feedback.
Application ordering for remediation based on potential for value generation	NA	True	Applications (or requirements) with a higher potential for value generation are addressed first. This improves commitment from client and lesser requirements may be forgone at later stages if they negatively impact project costs and schedule.

The context of the program remains same as outlined in table 2. The model is simulated for to-be state configuration of parameters. The initial fixed price for the project is set at $12.23 million which escalates to $13.5 million after changes are billed. The project exceeds timeline by 39 days as compared to 215 days in the as-is state. One of the primary factors responsible for improvement in the to-be state is – 'Client buy-in'. Organizing applications based on

value generation potential increases client confidence in the project at the earlier stages of project (Figure 16). Value-based remediation of applications promotes a lean approach as any requirements deemed unnecessary later may be excluded from the contract.

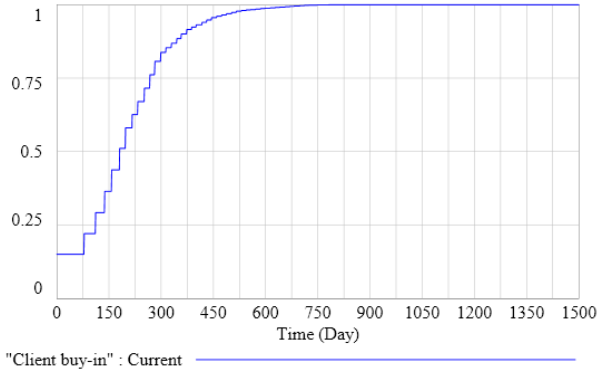

Figure 16. Client buy-in (To-Be state)

Client commitment to the project reduces overheads due to lack of competency, thereby improving overall productivity. Client commitment complements proactive project management. This facilitates rapid hiring of required resources. It is to be noted that the hiring of resources based on project requirements ensures efficient utilization of resources. Figure 17 shows the gap between available resources and productive resources which is lower as compared to as-is state for the duration of project.

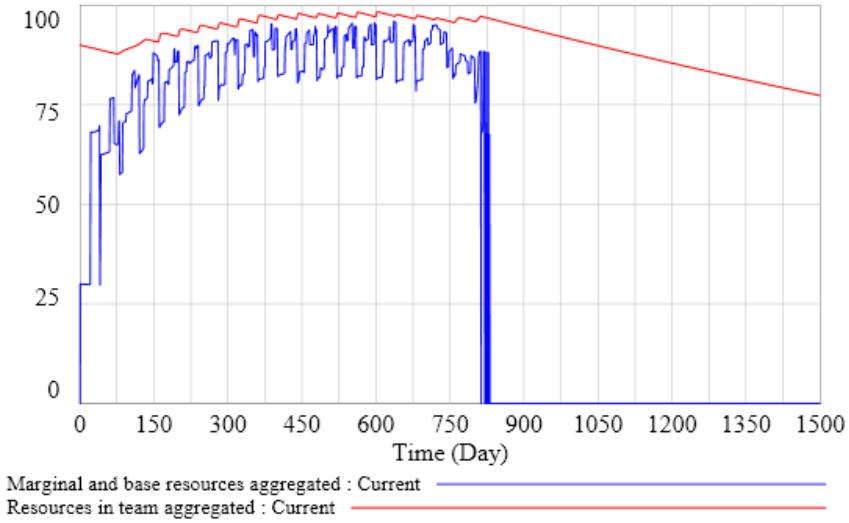

Figure 17. Productive set (Base and marginal) of resources vs. resources in team (To-Be state)

In addition to reduction in overheads, conformance of resources to requirements in terms of number, role and location reduces productivity discrepancy (Figure 18) as compared to the as-is state. Shorter cycle time creates immediate schedule pressure (Figure 19) requiring employees to invest extra effort. This increases the average productivity of an individual thereby improving overall productivity. It is to be noted that overtime constraints defined in the model maintains resource productivity at an average level of 10.8 hours per day and prevents exhaustion level from reaching breaking points.

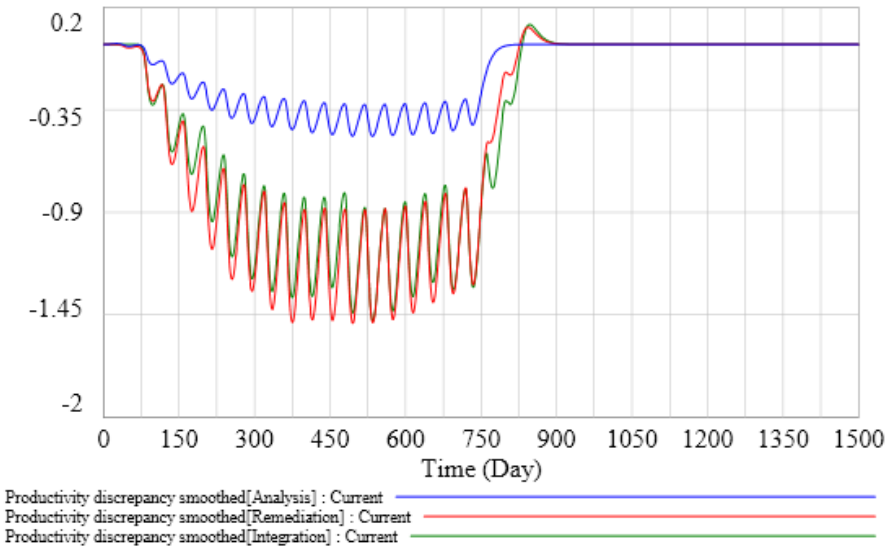

Figure 18. Productivity discrepancy (To-Be state)

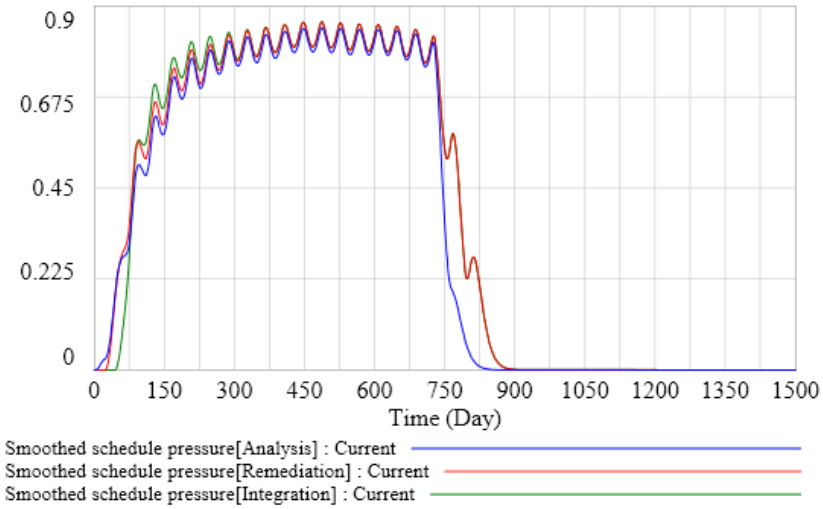

Figure 19. Schedule pressure (To-Be state)

Increase in schedule pressure causes a higher propensity for errors. Therefore, the total rework remains unchanged at 30 application equivalents (17% of total work). The overall error-proneness reduces by 12% due to scheduled execution of the project but still remains significantly high at 1.6 application equivalents. These constitute the side-effects of shorter release cycles and high schedule pressure.

Shorter release cycles improve the rate at which client discovers new requirements. This reduces the total number of change requests by 9% to 73 application equivalents. It is to be noted that the initial level of uncertainty w.r.t. project requirements is a major contributor to the total number of change requests. Shorter release cycles can decrease the number of change requests only by a fraction. Other change requests may be accommodated by the vendor, billed or exchanged with other requirements of similar nature in terms of time and effort.

To-Be state flow controls improves the competency of the team at an increased pace (Figure 20). This is because all resource roles are simultaneously working on different activities within the process. Competency reduces error-proneness in the face of high schedule pressure. This is another reason that the significantly higher schedule pressure in to-be state results in same amount of rework as in as-is state.

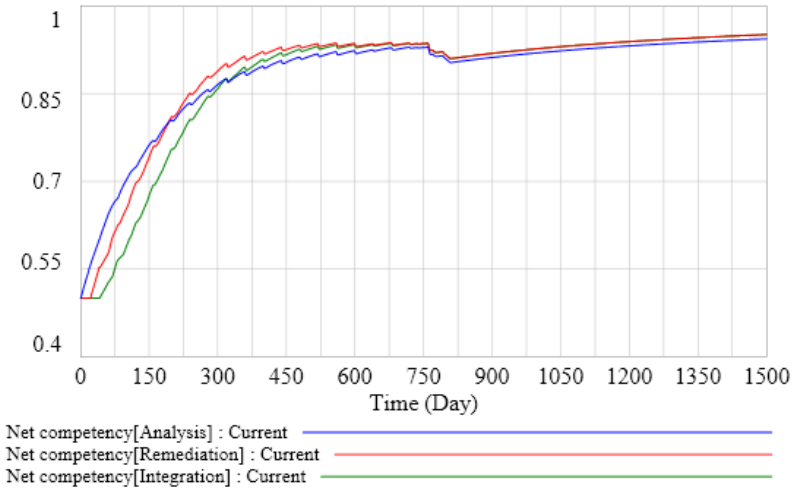

Figure 20. Activity specific team competency (To-Be state)

Accurate estimation of project costs and resources are critical to successful execution of fixed price projects. Adequate number of resources at start of project moderates schedule pressure and accurate cost estimation moderates cost pressure. Change costs and schedule extensions are accommodated for a portion of change requests. This extends timelines and reduces cost pressure on the vendor. However, this may not be possible for all fixed price projects. We simulated the model for a scenario, where change requests are not billed and schedule extensions are not permitted for change requests. The model predicted a schedule overrun of 105 days (14.5%), profit of $750,000 (6.13%) and 1.78 error-prone application equivalents. These results validate the effectiveness of to-be state configuration in accommodating all changes within a fixed price while sustaining moderate negative impact as compared to the current status of OCR program.

5.3 Recommendations for the OCR program

We derived the following inferences from configuration and simulation of the to-be state for OCR program.

- **Value based analysis and lean approach –** In the context of OCR program all applications do not yield the same value for the client when remediated. We recommend ordering the applications in terms of value they would yield when remediated. The primary prerequisite for value based analysis is application knowledge. It requires extra effort for consultant and senior associates. However, the benefits exceed the costs. Value based analysis generates client commitment at earlier stages of program. This positively affects productivity, resource acquisition and onsite co-ordination. Higher productivity and lower overheads ensures that program meets its objectives.

- **Agile methodology –** Literature provides instances of complementarity between agile methodology and fixed price approach. We modeled agile methodology within the OCR program. Reduction of cycle time by 35% (60 days to 40 days) yields higher productivity and lesser change requests (by 9%). Reducing the cycle time further creates higher schedule pressure, exhaustion and rework than desired. Through optimization we resolved 40 days with 2 application groups as the optimal cycle.

- **Proactive program management –** Figure 21 shows a schematic for PMO for the OCR program.

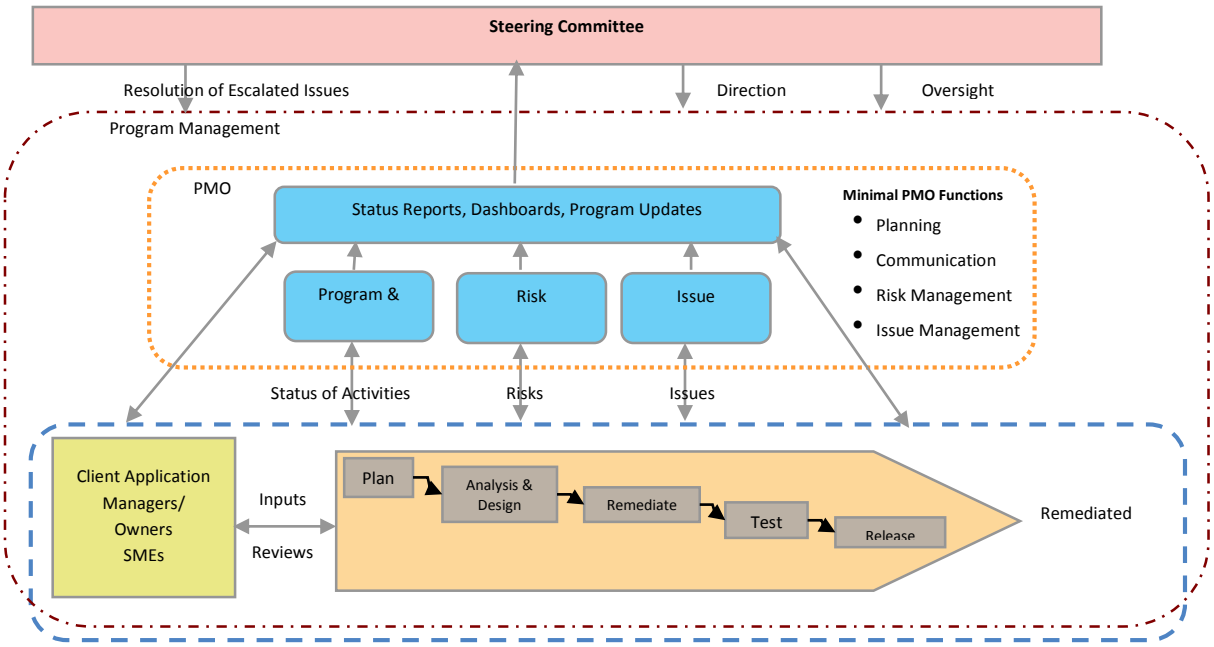

Figure 21. OCR Program PMO schema

As shown in the schema, continuous and consistent communication and co-ordination is required to execute the program within defined constraints. Fixed price programs require learning at early stages so that large amount of rework is avoided at later stages. Predictive intelligence affects the program in a number of ways – a) Resource procurement, b) Risk management, c) Client commitment, d) Requirements clarification, e) Competency management. We model resource management as an effect of proactivity in the model. This results in early procurement of right resources (number, role and location) within the program at the right stage. This reduces discrepancy in productivity and facilitates program execution within defined constraints.

- **Negotiable pricing model –** Contrary to what the name says, in a fixed-price program nothing is really fixed. Therefore, contractual provisions are required to implement models in cases of uncertainty. In the context of OCR program, we implemented billing and schedule extensions for a portion of change requests. This creates schedule and price flexibility for the vendor, an incentive for managing change as opposed to obstructing change. Change obstruction is of little use to the client as the delivery won't conform to requirements. Therefore flexibility in the operating and pricing model creates an optimal environment for the client as well as the vendor.

- **Fixed price estimate –** Vendors occasionally under-quote fixed-price to win contracts. This practice however, creates schedule and cost pressure since the start of program. Therefore, experience in terms of critical fixed price project execution is required to correctly estimate costs and quotations. In addition to correct estimation, cost constrained program decisions prove to be counter-productive. We switch on cost-constrained hiring in the as-is state and switch it off in the to-be state. Cost-constrained hiring reduces influx of resources because it may exceed costs. However, this reduces productivity as well and the resource costs for stretched timeline exceeds resource hiring costs. We therefore, infer and recommend that a fixed-price program should focus on generating value – through timely execution, quality of deliverables, conformance of delivery to requirements and imbibing program values. Cost optimality is the outcome rather than the goal itself.

6. Program Status and Conclusion

6.1 Program Status

The program took a period of 4 months to move from red to amber to green state after overcoming initial hiccups. Systems approach has given mechanisms to understand program state comprehensively and helped identify dominant themes and problem areas responsible for taking the program into deep risk zone. It helped understand that program

state was an emergent property resulting from the interaction of a number of problems in the program domain and no problem was alone and isolated.

Solution development has taken care of the fact that every program has its own characteristics and values and strategy definition must be tailored to these in order for the program to deliver values to its stakeholders. Besides, this study has addressed specific problem areas and made appropriate recommendations pertaining to each of these areas, namely, stakeholder management, resource management, project management and operations management. Systems approach recommends that all aspects of a system must be looked into before understanding of the system under consideration as a whole could be arrived at. Solution development has also included design of lean, minimalist PMO centered around maintaining and updating program plan, reporting program and project status, communication management and most importantly risk and issue management.

6.2 Conclusion

One of the fundamental learning from this case study is that in the fixed price paradigm – 'Everything must go right'. Any deviation from project schedule creates a ripple effect which would threaten the entire program. The primary example for this statement is – 'Resource on-boarding'. Delay in resource on-boarding as shown in (As-Is state simulation), reduces productivity with repercussions in terms of schedule pressure, error propensity, rework, change requests and costs.

Fixed price programs inherently create conflicting expectations for the client and vendor. A client expects fixed price without restrictions on scope whereas for vendor change in scope is dreaded. Therefore, for everything to go right, project requirements and expectations must be established with a high degree of certainty. In case of uncertain environments, we demonstrate through our simulation experiments, the importance of establishing a negotiable pricing model, which would accommodate change in scope through billing and/or schedule extension. This would create an optimal outcome for the client as well as the vendor.

We also observed that levying cost based constraints on fixed price projects is counter-productive. Optimal execution cost is one of the favorable outcomes as opposed to being the target itself. Learning through feedback is critical in a fixed price project (as demonstrated in the to-be state configuration). Therefore, techniques such as agile approach, lean development could be appropriately used to better manage these projects.

Although, this study and model pertains to a specific program, methodology followed here could be used on any other program with appropriate contextualization. ***Fixed-price*** contracts contain a business model within and are here to stay unless a better model is proposed that suits the interest of clients as well as vendors. Furthermore, ***Fixed-price*** contracts challenge the discipline of project management and encourage researchers and practitioners to discover systems, structures and processes that could help manage these projects better and enrich project management as an area of research.

7. References

Cauwenberghe, P.V. (2006a). *Agile **Fixed-price** Projects part 1: "The Price Is Right"*. Retrieved November 26th, 2009, from http://www.softdevarticles.com/modules/weblinks/singlelink.php?lid=228

Cauwenberghe, P.V. (2006b). *Agile **Fixed-price** Projects part 2: "Do you want agility with that?"* Retrieved 27th November, 2009, from http://www.softdevarticles.com/modules/weblinks/singlelink.php?lid=229

Cockburn A. (2004). *Crystal Clear, A Human-Powered Methodology for Small Teams*. Boston-Addison Wesley Professional,

*Dr. Dobb's (April 16, 2007). The consequences of **fixed-price** IT projects.* Retrieved 1st December, 2009 from *http://www.ddj.com/architect/199001126*

Flyvbjerg, B. (2006). From Nobel Prize to Project Management: getting risks right. *Project Management Institute*, Vol. 37, No. 3, 5-15,

Gharajedaghi, J. (2006). *Systems Thinking: Managing Chaos and Complexity*. California: Butterworth-Heinemann

Glass, R. L. (2004). Anarchy and the Effects of Schedule Pressure. *IEEE Software, vol. 21, no. 5, pp. 111-112*

ISO/DIS 31000 (2009 b). *Risk management-Principles and guidelines on implementation*. International Organization for Standardization.

ISO/IEC Guide 73:2009 (2009 a). *Risk management-Vocabulary*. International Organization for Standardization.

Kerth, N. L. (2001). *Project Retrospectives: A Handbook for Team Reviews*. New York: Dorset House.

Morris P.W.G. (1994, p. 217). *The Management of Projects*. London : Thomas Telford Ltd.

Murthy, P.N. (1994). Systems Practice in Consulting. *Systemic Practice and Action Research*, Volume 7, No. 4, pp. 419-438

Poppendieck, M. (2003). Lean Software Development. *C++ Magazine Methodology Issue (Publication Fall 2003).*

Project Management Institute, (2008). *A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE (*4th Ed.). Pennsylvania: Project Management Institute, Inc.
Royce, W. W. (1970). Managing the Development of Large Software System. *Proceedings of IEEE WESCON, pp. 1-9.*

Rai V K (2009). *A Probable Structure Underlying PMO Functions*. In the proceedings of the 5th Annual International Project Management Leadership conference (PML, 2009), New Delhi, India.

Rai V. K. and Mahanty B. (2001). *Dynamics of Schedule Pressure in Software Projects.* The Proceedings of the 20th International Conference of the System Dynamics Society, Palermo, Italy

Rai V. K. and Mehta Sanjit (2012). *Systems Approach to As-Is State Formation of an Engagement: A case Study Illustration.* In proceedings of 6th IEEE International Systems Conference (IEEE Syscon 2012).

Rai V. K. and Swaminathan N. (2010). *Constructing Program Management Framework- A system of Systems Approach.* In proceedings of 4th IEEE International Systems Conference (IEEE Syscon 2010).

Rai V.K., Vijayasaradhi B., Subramanian K and Umamaheswari S. (2010). *ODC Portfolio Management Ensuring Sustainability in the Face of Growth Surge- A Case Study*. In proceedings of 4th Annual IEEE International Systems Conference (IEEE Syscon 2010).

Senge, P. M. (1990). *The Fifth Discipline: The art and practice of the learning organization*. New York: Doubleday

Sterman, John D. (2001). System dynamics modeling: Tools for learning in a complex world. *California management review* **43** (1): 8–25.

Turner, J. R. and Simister S. J. (2001). Project Contract Management and a Theory of Organization. *International Journal of Project Management*, Volume 19, Issue 8, November 2001, pp. 457-464

Weinberg, G. M. (1992). *Quality Software Management (Vol 1): Systems Thinking.* New York: Dorset House.

Van Oorschot, Kim E., Kishore Sengupta, and Luk van Wassenhove. "Dynamics of agile software development." In 27th International Conference of the System Dynamics Society, Albuquerque, New Mexico. 2009.