Managing an ITIL SaaS Implementation IT program: A Case Study

Gottfried Rudorfer, PMP, ITIL Expert, CA Technologies

Abstract

The case study “Managing an ITIL SaaS Implementation IT program” provides insight into the management of multi-year MSP-program. Techniques will be shown how to manage dynamic complexity of such programs, how to setup and run in parallel running projects with pre-defined feedback loops to control scope, time and budget. This paper describes activities carried out during program initiation, program setup, delivery of program benefits and program closure. On project-level activities are described to concept, design, build and operate SaaS services. This paper concludes that building feedback loops on program and project level will help to obtain program benefits in a controlled way.

Introduction

Dynamic complexity is seen by researches in the project management field as one of the primary cause for project failure. This is especially true for large IT programs with durations of several years. This case study shows how system dynamics of large-scale programs can be managed by setting up feedback-loops to preserve program benefits. We investigate in detail processes, tools and techniques used in a complex multi-year ITIL SaaS implementation IT program. Interested program managers may adopt these techniques into their own engagements. During the program lifecycle constantly new projects will start within the program. Many projects will also run in parallel with other projects sharing the majority of the project team and using the same shared infrastructure resources. We will look at used techniques to repeatedly generate projects out of the program and how this can be done in an efficient and simplified way based on scope statement and project plan templates. Shared human and infrastructure resources are managed in weekly plans which impose visible resource constraints on parallel running projects. Project managers of the program monitor progress and budget of their projects with pre-defined feedback loops by using on one hand the percent complete figure inserted into the project plan and on the other hand the actual hours submitted on timesheets from each assigned resource of tasks. With this information an estimate to complete is calculated on a weekly basis to give an early warning about a deviation from the plan with regard to schedule, efforts or budget. The presentation concludes with a monthly bird’s eye look at the whole program using accumulated KPIs over all projects compared to initial plans and budget of the program.

Motivation

According to the PMI Standard for Program Management (PMI, 2008a, Chapter 1.2 What is a Program, Kindle location 1103), “A program is a group of related projects managed in a coordinated way to obtain benefits and control not available from managing them individually”. This paper is about the experiences of the author gained managing an ITIL SaaS implementation IT program. IT Program Management is the process of the management of several related IT projects to obtain the defined utility to the business of the customer. In our case the target of such a program is to build a shared IT software environment which is used by more than one end customer as tool support for several ITIL processes like incident or problem management. The customer is providing these services as Software as a Service (SaaS) and the end customers are interacting with their instance or tenant trough the web browser or via E-Mail. The SaaS environment provides tool support for one or more ITIL processes for one or more end customers. The hosting infrastructure environment is operated by the customer as a SaaS solution (Wikipedia, 2012a) for their end customers accessing this shared environment. After the shared infrastructure environment was built and setup for the first customer who is on-boarded as a tenant of the multi-tenanted system further SaaS customers are added to the same shared environment. To avoid disruption of already productive tenants this environment consists beside of the production (PROD) environment of an architecturally identical (DEV) and testing (QA) environment. Activities, changes and maintenance on any of the environments are managed through a weekly
Origin

ally published as a part of 2012 PMI Global Congress Proceedings – Vancouver, Canada

rolling plan. It is a special task schedule to avoid productive outages of already productive tenants during onboarding of new tenants. For a detailed description see Exhibit 11.

This paper is showing the process and techniques how 10 projects could be managed in “green state” with regard to the criteria schedule, budget, scope, resources (see in Exhibit 5).

The importance of feedback loops in dynamic systems

According to the ITIL strategy book (Taylor, S., Iqbal, M., Nieves, M., 2007, p23), IT organizations can be visualized as dynamic systems with functions and processes. Each process or function can provide to each other feedback towards the goal of meeting the customer objectives. Any delay in negative feedback towards the goal may lead to oscillations in this dynamic system due to intervening corrections. In the end any delay may have a destabilizing effect to the system. Feedback loops in form of reporting and improved measurement can reduce this destabilizing effect.

Grösser, S. N. (2011) sees the reason for many unsuccessful projects in unmanaged dynamic complexity.

The implementation of a sufficient number of feedback loops is especially important for programs as it involves multiple projects where each project can be seen as dynamic system interacting with other projects, with the program itself, with the IT organization and with the business of the customer.

On the other side the effort for program and project managers to define and regularly evaluate feedback loops has to be considered. It is probably one of the most challenging tasks for the program manager is to select the correct types and number of feedback loops so that on one hand the program is not destabilized and on other hand the administrative effort for maintaining feedback loops is in an acceptable range.

General information about the ITIL SaaS IT implementation program

We want to give you first a brief general description about the program:

Setup of the Program

The program was set up with a predefined budget and amount of days. The size of the program is determined by the size of projects in the pipeline. Additionally there is a split of the effort among different project roles like Project Manager, Sr. Consultant, Offshore Delivery and Architect. According to the planned effort per role the total program contract value can be calculated.

The project pipeline is defined in a table and lists on the one side all planned projects and on the other side the monthly expected effort. This baseline planning sheet is used to compare it with past project months and the updated project pipeline (see Exhibit 5)

Setup of Projects within the Program

All work performed needs a formal request and approval to set up a new project within the program. Each project requires a detailed WBS and a reviewed estimate by at least one Architect. The project manager builds a Microsoft Project plan with a detailed set of tasks, assigns resources, adds dependencies between tasks, adds non-availability in the project team calendar and verifies that the estimated effort is matching the total work of the plan. After all is defined it is important that the project manager schedules a final meeting to walk through all work packages, assumptions and all documents in a final review meeting. Any changes in any of the defined documents need to be clarified between the supplier and the customer.
Execution

Early detection of any cost or effort overruns

Project progress is updated weekly by the project manager by a walk-through of all tasks to be performed with each assigned project member. Each member is asked about the percent work complete figure of his assigned tasks which is then updated in the project schedule.

In the resource view of the project plan actual and remaining work for each task is shown. Additionally actual work performed by each consultant is collected from weekly timesheets. With this information the expected total effort and total costs of the project are calculated by using the spreadsheet shown in Exhibit 10. This process is repeated by each project manager for each active project at least on a weekly basis. This indicator delivers an early warning for a possible effort and budget overrun. For IT programs efforts and budgets are of extreme importance because the customer is expecting cost savings through the implementation of the program. Often the program benefits are lost if this indicator is ignored and cost overruns occur.

This approach also allows a better detection of scope changes or additional work packages that aren’t part of the existing project plan. Whenever this happens a new change request will be added to the change request register which then will be analyzed, described and discussed with the customer if and when it makes sense to be implemented.

Closure

It is important to deliver early and regularly benefits to the program. In our program it is important to reach at least every month a milestone which gets visible to the business. Good examples are the on-boarding and go-live of new tenants (end-customers) into the shared environment and the enrichment of supported ITIL processes as well as the extension of base functionality i.e. to add single sign-on and federation for end-customer to avoid user/password logins in the shared service desk environment.

The implementation of our SaaS environment is based on the concept of the ITIL Service Life Cycle: “Service Design, Service Transition and Service Operation are progressive phases of the Lifecycle that represent change and transformation. Service Strategy represents policies and objectives. Continual Service Improvement represents learning and improvement. Service Strategy (SS) is the axis around which the lifecycle rotates. Service Design (SD), Service Transition (ST), and Service Operation (SO) implement strategy. Continual Service Improvement (CSI) helps place and prioritize improvement programmes and projects based on strategic objectives.” (Taylor, S., Iqbal, M., Nieves, M. , 2007, p24).

Exhibit 1 illustrates this concept: Figure from: (Taylor, S., Iqbal, M., Nieves, M. , 2007, p24).

Feedback loops

This section presents examples which implement feedback loops. These examples show the techniques and tools that have been used in our program and how you as interested program manager might adopt these into your engagement.

Please note that for nondisclosure reasons the numbers shown in the tables are random.

For the success of the program it is important to introduce a set of feedback loops which are regularly evaluated to see if benefits of the program are achieved and if all projects are in scope, time and budget.

We distinguish on feedback loops on program and project level. If possible we also like to have more than one view to the same indicator to see if the calculations behind are correct.

Program Level Feedback Loops

On program level it is important to see the status of the whole program and its projects in a view at a glance. This paper will present two views: A bottom-up view from project level and a top down view from program level.

- Bird’s eye few to the program with bottom-up reported project information

This view is showing the current resource consumption and financial status for each project. Because of space limitations in this paper Exhibit 2 shows a region of that view - the status of project “Tenant1”. In column F the estimated hours per role are shown. Column G allows for additional planned efforts coming from change requests (CRs). Total hours in column E is the sum of the corresponding cells in F and G. Available hours in column D is the difference between column E and B. Actual hours in column B is the sum of hours automatically calculated from a tab containing timesheet raw data of all projects of the program. Monthly project revenue in columns I to M is automatically calculated based on the timesheet raw data.

The status of further projects of the program is listed on the same sheet in the same format using the same formulas and the same timesheet raw data.

### Exhibit 2

Timesheet raw data is shown in Exhibit 3 whereas only the Name, Date, Project and Hours need to be entered. All other information is calculated with formulas and used for the project status view. This data is the base for all reports in this view.
The status of the program shown in Exhibit 4 is aggregated from all projects. Actual, available and total numbers are shown in hours as well as financial values.

Exhibit 3

Exhibit 4

- Bird’s eye view on program level: Actual reported effort versus initially at program setup planned effort per month. For future months the project pipeline is updated and compared with the initial plan.

This view gives quite good status information of the program by using monthly accumulated work in days. The screen shot in Exhibit 5 shows the program status as of March 31, 2012.

The top of the spreadsheet is showing planned projects and the distribution of the effort during the duration of the program. The numbers in this view remain all time the same from report to report.

The bottom of the spreadsheet shows the current status. There is an overlaid transparent box between 08/11 and 03/12. The numbers within this range are the actual performed work per month and per project. Efforts with a date from 04/12 are showing the actual project pipeline. Please note that the pipeline in the screenshot is the same as during draft planning. In reality one can expect that the pipeline, the list of projects and the order of execution of the projects will change over time and will be different to the initial plan.

For the steering committee three key indicators are important:

- Draft / planned vs. actual total days per each month
- Draft / planned vs. actual accumulated total days
- Draft / planned vs. actual average heads

One goal of the program could be that the accumulated days per month between the initial plan and the actuals should be more or less the same. If there are less actual accumulated days consumed the steering committee might decide to start additional projects and vice versa.

© 2012, Gottfried Rudorfer
Originally published as a part of 2012 PMI Global Congress Proceedings – Vancouver, Canada
Exhibit 5

- Traffic Light status overview for each project

The traffic light report as shown in Exhibit 5 is commonly used to report the program status at the steering committee meetings. It shows all projects of the program categorized into ongoing, finished and new projects. To avoid discussions about the coloring of each status it is recommended to agree on common reporting categories. In our program we are using the reporting categories Schedule, Budget, Scope and Resources as shown in Exhibit 6.
Project Level Feedback Loops

- Estimated final project effort in man-days and estimated final project costs

To successfully manage projects it is important to first setup the project including the WBS, reviewed effort estimation, task sequencing, resource assignments and leveling.

During project execution it is important to set up a feedback loop to enable an estimate to complete.

In our project we define a project plan which corresponds to the estimated work effort in the statement of work. All our plans use several resource roles like Project Manager, Architect and Senior Consultant. We extract the estimated effort from the project plan into the first column “Estimated MD”. Should there be any change request then we add these efforts to this number as an approved change request is in scope of the project.

At the weekly project status meetings project members assigned to a task are asked about their estimate on the percent work complete i.e. 0%, 50%, 100% complete. The project manager updates these values for all tasks in the project plan. In resource view the “Actual Work” and “Remaining Work” can be seen and copied into the estimate to complete Excel-Sheet (see Exhibit 8).

Exhibit 7

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Project - Green</th>
<th>Project - Amber</th>
<th>Project - Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>Next Milestone and overall plans on track.</td>
<td>Issues achieving next Milestone and/or overall timeline.</td>
<td>Next milestone and/or overall timeline will not be met.</td>
</tr>
<tr>
<td>Budget</td>
<td>Currently within budget and forecast to remain within budget.</td>
<td>Currently not in budget and/or overall budget will be exceeded by &lt;5%.</td>
<td>Currently not in budget and/or overall budget will be exceeded by &gt;5%.</td>
</tr>
<tr>
<td>Scope</td>
<td>Scope and objectives are agreed with all stakeholders and can be met.</td>
<td>Different understanding within stakeholder and/or objectives are in danger.</td>
<td>Scope/Objectives cannot be met currently.</td>
</tr>
<tr>
<td>Resources</td>
<td>Staffing is appropriate</td>
<td>Issues with staffing.</td>
<td>Cannot be executed with current staffing.</td>
</tr>
</tbody>
</table>

Exhibit 8

![Image of Exhibit 8](image)

Also on a weekly basis all project members submit their timesheets with actual hours worked for each task. This effort value is entered into the column “Actual work performed”. By subtracting this work effort from the estimated effort we’re getting the available man-days which are available on that contract to be used (see Exhibit 9).
Finally an estimate to complete can be calculated using the previously collected data. Available man-days or an overrun will be shown per resource role. Adding hourly rates for each role also allows for the calculation of cost savings or cost overruns when the project will finish. In the example shown in Exhibit 10 it is possible to have an overrun with regard to the man-days whereas the financials show available buffer. The explanation for this is that in this example there was a shift of Architect work to the cheaper Offshore Delivery team.

**Exhibit 9**

<table>
<thead>
<tr>
<th>Role</th>
<th>Available MD</th>
<th>Remaining work planned</th>
<th>Avail (+) MD / Overrun ( )</th>
<th>Hr. Rate</th>
<th>Avail (+) Cost / Overrun ( )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>9.0</td>
<td>7.4</td>
<td>1.6</td>
<td>100.0</td>
<td>1280.00</td>
</tr>
<tr>
<td>Architect</td>
<td>6.3</td>
<td>2.8</td>
<td>3.7</td>
<td>96.0</td>
<td>2875.20</td>
</tr>
<tr>
<td>Senior Consultant</td>
<td>9.8</td>
<td>8.5</td>
<td>1.3</td>
<td>90.0</td>
<td>900.00</td>
</tr>
<tr>
<td>Consultant</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>80.0</td>
<td>0.00</td>
</tr>
<tr>
<td>Associate Consultant</td>
<td>0.8</td>
<td>0.0</td>
<td>0.8</td>
<td>75.0</td>
<td>487.50</td>
</tr>
<tr>
<td>Offshore Delivery</td>
<td>-9.0</td>
<td>18.5</td>
<td>-1.5</td>
<td>506.0</td>
<td>-4680.00</td>
</tr>
<tr>
<td><strong>Sum</strong></td>
<td><strong>16.9</strong></td>
<td><strong>18.5</strong></td>
<td><strong>-1.5</strong></td>
<td><strong>506.0</strong></td>
<td><strong>-4680.00</strong></td>
</tr>
</tbody>
</table>

**Exhibit 10**

It is important to re-evaluate these numbers and communicate the status per role to all team members on a weekly basis. As project manager you should give each team member feedback if the efforts she or he is generating is in line with the plan or not.

For the project manager any significant deviation in the estimate complete should be used to start further investigations how this deviation can be explained.

- Human, hardware and other shared resources rolling plan

Implementing a shared hardware and software environment requires a more detailed planning of human, hardware and software resources.

Exhibit 11 shows a weekly rolling plan for project “Tenant1” and project “Tenant2”. Both projects run during this week in parallel. The rolling plan shows that for “Tenant1” there is only design work planned which does not need any shared resource. For “Tenant 2” there is a move to QA environment planned where the DEV and QA environments are used.

The complete workbook contains of a set of tabs showing the rolling plan for whole program duration.
Templates for the generation of new projects within the program

Scope statement templates
For the program it is important to define a simple formal process for the definition of new projects. A new project SOW should refer to the terms and conditions of the program. I.e. the SOW could exist only of the scope statement and it could be i.e. contracted via E-Mail to minimize administration effort.

For requesting new projects the customer should use a template to create formal requests defining the goals and requirements.

Project plan templates
In our program project plans are an important planning and feedback tool. It is important to pre-define phases like the creation of design documents, user acceptance testing and go-live.

Documentation structure for the program
For the program a document management system is used to store and manage all program related documents. It is a system which is accessible over the internet for all project members: Supplier, customer and vendors. For each folder special permissions were set up to restrict access between members of the program.

The directory structure as shown in the table in Exhibit 12 is used for the program on program level and the directory structure shown in Exhibit 13 is used on project level.
Conclusion and Future Work

By using feedback loops dynamic complexity of our ITIL SaaS Implementation IT program could be successfully controlled in an efficient way. We plan to create general work templates which will include all needed formulas and macros and make these available to interested project and program managers.
References and Appendices


Lacy, S., MacFarlane, I., Taylor, S. (2007): Service Transition; TSO (The Stationery Office); Published for the Office of Government Commerce (OGC)


Case, G., Spalding, G., Taylor, S. (2007): Continual Service Improvement; TSO (The Stationery Office); Published for the Office of Government Commerce (OGC)


Wikipedia (2012a), Software as a service, retrieved on 19 August, 2012 from http://en.wikipedia.org/wiki/Software_as_a_Service