Managing Complexity in the Face of Uncertainty

Ch08: How to Close a Project

After successful studying this chapter You should be able to:

- Use tools, templates, and processes to close a TPM project,
- Write & maintain client acceptance procedures
- Close a project
- Get client acceptance
- Install project deliverables
- Document the project
- Conduct the post-implementation audit
- Write the final report
- Celebrate success
After successful studying this chapter You should be able to:

- Acceptance Test Procedure (ATP)
- Implementation strategies
- Project documentation
- Post-implementation audit
- Final project report

**Closing a Project**

1. Getting client acceptance of deliverables.
2. Ensuring that all deliverables are installed.
3. Ensuring that all documentation is in place.
5. Conducting the post-implementation audit.
Ch08: How to Close a Project?

Why Do I Need Client Acceptance Procedures?

- Acceptance criteria must be clearly defined
- Criteria defined during project planning
- Avoid 11th hour disputes

1. The criteria is defined in collaboration with the client.
2. The criteria is administered by the project team during final acceptance testing.

Installing Project Deliverables

- Phased Approach
- Cut-Over Approach
- Parallel Approach
- By Business Unit Approach
Installing Project Deliverables

- **Phased Approach**
  - The phased approach decomposes the deliverable into meaningful chunks and implements the chunks in the appropriate sequence. This approach would be appropriate in cases where resource limitations prevent any other approach from being used.

- **Cut-Over Approach**
  - It replaces the old deliverable with the ONE in one action.
  - The testing of the new system must have been successfully completed in a test environment that is exactly the same as the production environment, before the replacement an be done.

- **Parallel Approach**
  - In this approach, the new deliverables are installed while the old deliverables are still operational.
  - Both the old and the new deliverables are simultaneously in production mode.
  - It allows the new system to be compared with the old system on real live data.
Installing Project Deliverables

- **By Business Unit Approach**
  - In this approach, the new deliverables are installed in one business unit at a time, usually in the chronological order that the system is used.
  - Like the phased approach, this approach is appropriate when resource constraints prohibit a full implementation at one time.
  - Similar to the by-business-unit approach would be a geographic approach where the system is installed at one geographical location at a time. This facilitates geographic differences, too.

Documenting the Project

- Reference for future changes in deliverables
- Historical record for estimating duration and cost on future projects, activities and tasks
- Training resource for new project managers
- Input for further training and development of the project team
- Input for performance evaluation by the functional managers of the project team members
Ch08: How to Close a Project?

Documenting the Project – The Project Notebook

- POS
- Proposal and backup data
- Original and revised project schedules
- Minutes of all project team meetings
- Copies of all status reports
- Design documents
- Copies of all change notices

- Outstanding issues reports
- Final report
- Sample deliverables (if appropriate)
- Client acceptance documents
- Post-implementation audit report

NOTE: Start the project notebook on the first day
Conducting the Post-Implementation Audit

- Was the project goal achieved?
  - Does it do what project team said it would?
  - Does it do what client said it would?
- Was the project work done on time, within budget, and according to specification?
- Was the client satisfied with the project results?
- Was business value realized?
  - Check success criteria
- What lessons were learned about your project management methodology?
- How well did the team follow the methodology?

Reasons for not doing a post-implementation audit

- Managers don’t want to know
- Managers don’t want to pay the cost
- It’s not a high priority
- There’s too much other scheduled work to do
Ch08: How to Close a Project?

Final Project Report

- Executive Summary
- Overall success and performance of project
- Organization and administration of project
- Techniques used to accomplish results
- Strengths and weaknesses of the approach
- Recommendations
  - Appendices
    - POS
    - WBS
    - Resource Schedule
    - Change Requests
    - Final Deliverables
    - Other

Celebrating Success

(Steepening incentives or financial reward)
End of Chapters (8)

Managing Complexity in the Face of Uncertainty

Ch09: Complexity and Uncertainty in the Project Management Landscape
Complexity and Uncertainty

After successful studying this chapter You should be able to:

- Understand the complexity/uncertainty domain of projects
  - Requirements, Flexibility, Adaptability
  - Risk vs. the Complexity/Uncertainty Domain
  - Team Cohesiveness vs. the Complexity/Uncertainty Domain
  - Communications vs. the Complexity/Uncertainty Domain
  - Client Involvement vs. the Complexity/Uncertainty Domain
  - Specification vs. the Complexity/Uncertainty Domain
  - Change vs. the Complexity/Uncertainty Domain
  - Business Value vs. the Complexity/Uncertainty Domain

Complexity/Uncertainty Domain of SDPM

- Complexity and uncertainty are positively correlated with one another. As software development projects become more complex, they become more uncertain. This follows from at least four factors:
  - Requirements – As project complexity increases, the likelihood of fixing requirements decreases. In a complex software product the extent of the number of requirements, functionality, and features can be surprising. Some will conflict, some will be redundant, and some will be missing.
  - Flexibility – as project complexity increases, so does the need for process flexibility. Increased complexity brings with the need to be creative and adaptive. This is difficult to achieve in the company of rigid processes.
  - Adaptability – this is directly related to the extent to which the team members are empowered to act. When a project is less certain in terms of requirements, functionality, and features, the more the need to be adaptable with respect to process and procedure.
  - Change – As complexity increases, the frequency and need to receive and process change requests increase as well. A plan driven software development project is not designed to effectively respond to change.
Islamic University, Gaza - Palestine

**Complexity/Uncertainty Domain of Software Development Project Management (SDPM)**

<table>
<thead>
<tr>
<th>GOAL</th>
<th>SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>1</td>
</tr>
<tr>
<td>Not clear</td>
<td>2</td>
</tr>
<tr>
<td>Clear</td>
<td>3</td>
</tr>
<tr>
<td>Not clear</td>
<td>4</td>
</tr>
</tbody>
</table>

**Risk versus the complexity/uncertainty domain**

- Risk increases as you move from quadrant 1 to 2 to 3.
- In quadrant 1 you know the goal and the solution and can build a definitive plan for getting there. The focus can shift to process failure.
- Because a list of candidates risks have been compiled over past similar projects, the likelihood, impact, and the appropriate mitigations is known and documented.
- In quadrant 2 two forces come into play – the Software Development Project Management (SDPM) strategy becomes more flexible and lighter; and at same time, the product risk increases.
- In quadrant 3, risk is the highest because you are in a research and development environment. Process risk is almost nonexistent because the ultimate in flexibility has been reached but the product risk is extremely high.
Team cohesiveness versus the complexity/uncertainty domain

- In quadrant 1 the successful team doesn’t really have to be a team at all. A team of specialists will do. The plan is sacred and the plan guides the team through their tasks.
- The situation quickly changes in quadrant 2 or 3 project. There is gradual shift from a team of specialists to team of generalists.
- Teams become self-sufficient and self-directing as the project moves from a quadrant 2 to a quadrant 3.
- Quadrant 1 team are not co-located while quadrant 2 and 3 teams are co-located.

- Research has shown that co-location adds significantly to the successful completion of the project. The figure above reflects this from a loosely formed team to one that is tightly coupled.

(Cohesiveness: togetherness of the feeling of being a team or a group)

Communications versus the complexity/uncertainty domain

- Lack of timely and clear people-to-people communications has been shown to be the single most frequent reason for project failure.
- As you move in the direction of increased complexity and heightened uncertainty, communication requirements increase and change.
- When complexity and uncertainty are low, the predominant form of communications is written.
- As uncertainty and complexity increases, written communication give way to verbal communication.

- With increased uncertainty and complexity, the burden of plan-driven approaches is lightened, and the communications requirements of value-driven approaches take over.
- Value-driven communications approaches are the derivatives of meaningful customer involvement.
Customer involvement versus the complexity/uncertainty domain

- Customer involvement is usually limited in quadrant 1 projects because they are team-driven.
- The customers take an important role in quadrant 2 and 3 projects.
- Meaningful customer involvement can be daunting task because of the following three reasons:
  - The customer’s comfort zone - traditionally customers have been trained to be passive but, that is changing. Now you have a lot of technically sound customers that know what it takes to build a software.
  - Ownership by the customer – this is critical but care need to be taken so that the customer will not delay the project.
- Customer sign-off – This is often nervousness -filled task that you ever ask of your customer because customers think that they are signing their lives away when they approve a document or a deliverable. Your task is to dispel that perception. This is always a challenge.

Specification versus the complexity/uncertainty domain

- What does this means? Simply put, it advises you that the choice of SDPM strategy should be based on an understanding of the confidence you have that the specifications have been completely and clearly defined and documented.
- Also, that scope change requests will not arise from any shortcomings in the specifications documents.
- As that specification certainty diminishes, your best choices lie in the iterative strategies that populate quadrant 2 projects.

Finally, if you are not sure that you have clearly and completely documented the specifications, then your SDPM strategy takes on the flavor of the research and development strategies that populate quadrant 3.
• Uncertainty at the requirements level has more impact on choice of software development approach than does uncertainty at the functionality level, which has more impact than that at the features level.

Islamic University, Gaza - Palestine

Change versus the complexity/uncertainty domain

• The less you know about requirements, functionality, and features, the more you have to expect change.
• Quadrant 1 – you know everything (almost) that needs to be known so change is minimal.
• Quadrant 2 – Any change in this quadrant comes about through the normal learning process that takes place in any software development project.
• Quadrant 3 – The projects in this quadrant require change in order to have any chance at finding a successful solution. Change is the only vehicle that will lead to solution.

The figure above reflects the frequency of change as projects move across the landscape.
Complexity and Uncertainty

Non-Value-Added Work

Non-value-added work involves the consumption of resources (usually people or time) on activities that do not add business value to the final product or process.

Business value versus the complexity/uncertainty domain

- It seems all software projects are created to return business value to the enterprise. This is all true. However, traditional project approaches focus on meeting the plan-driven parameters: time, cost, scope. When originally proposed the business climate was such that the proposed solution was the best that could be had. Unfortunately business world is not static.
- Quadrant 1 development projects are not equipped with the right stuff to deliver business value.
- Simply put, whatever SDPM strategy you adopt for the project, it must be one that allows redirection as business conditions change. More uncertainty means more redirection.

As you move from quadrant 1 to 2 to 3, risk increases and that means that higher-values projects need to be commissioned because the expected business value of a project is the product of (1 – risk) and value. Risk here is the probability of failure and the probability of success is therefore (1 – risk).
End of Chapters (9)