

TEN GOLDEN RULES FOR COST SAVING IN SOFTWARE

PREMA S. THOMAS¹ & SATHEESH THOMAS²

¹Assistant Professor, MCA, Rajagiri College, M.G University, Kerala, India

²Managing Director, Pumex Infotech Pvt. Ltd, Infopark, Kochi, Kerala, India

ABSTRACT

Technology trends are changing very fast compared to earlier days and it is a challenge for companies in Software development to implement more complex systems at a cheaper cost maintaining better quality. This article analyses some of the common causes for higher cost of implementation and suggest some rules which can help in development lifecycles for cost saving.

KEYWORDS: Cost Saving, Software Development, Schedule Slippages, T & M Model Opportunity Cost, Fixed Bid Project, Resource Planning, Risk Management Plan

INTRODUCTION

The competition in the industry on faster, better and cost effectiveness is increasing day by day. So companies need to find out innovative approaches to reduce the cost of implementation to sustain in the market. One easy way adopted by many companies is to reduce the direct expense by cutting the meals or reducing salaries or few layoffs. These are drastic methods adopted to cut down on the cost but there are better ways to do it by adopting proper project management practices.

In software development the major cost of implementation is the effort or the time spent by human resources on the project. So key aspect is to reduce the schedule slippages and improve efficiency which result in cost saving. Schedule slippages are common in software development even after companies spend lot of time in risk mitigation and implementing best project management practices. Since slippage is directly proportional to cost and it is a major concern for customers and software development companies based on who bears the risk in the development model. In T & M (Time and Material) model the customer has the costing risk and in a fixed bid model the contractor or the development company has the risk of budget over run and thus impacting the bottom line.

This paper presents some key points which can avoid or minimize the impact of slippages through proper risk management practices which can in turn save cost. We are not planning to discuss the standard risk planning or risk management practices as there are enough material available on this subject which was created over the years. Instead we will discuss some practical approaches which can mitigate the risks which can result in significant cost saving.

The first thing is to identify which phase in the project is taking most of the time. Ideally the implementation phase should take maximum time but in most practical cases the testing and bug fixing cycle has the maximum slippage. This paper analyzes the issues and proposes some tips for improvement especially with the software development projects in mind. But it is applicable to other industries as well with minor customization.

IMPACT OF SLIPPAGE

The project slippage can affect the stake holders in multiple ways. Some of them are

- Opportunity cost for the customer
- Additional cost for the developing company if it is a fixed bid project
- Additional cost for the customer if it is a T & M engagement
- Project completion pressure can end up in lower quality
- Opportunity cost for the developing companies to move the resource to other projects
- Frustration in resources as they want a change
- Stake holders may become defensive to protect their stand which is counter productive
- Loss of Brand / Trust

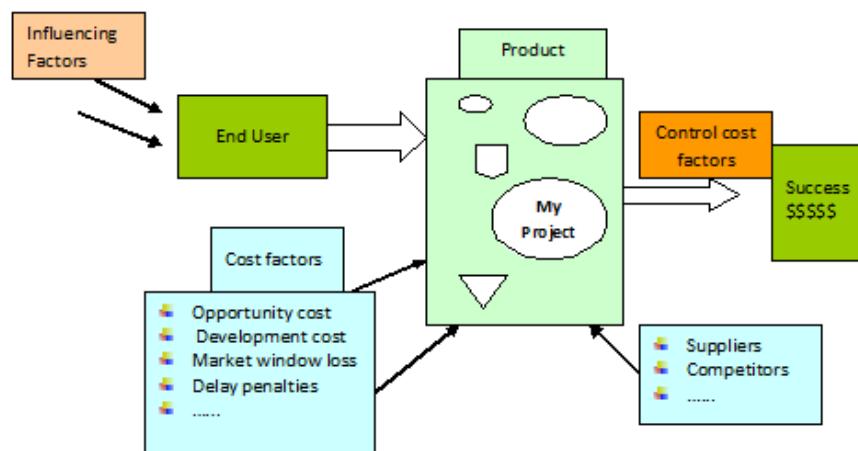


Figure 1

TEN GOLDEN RULES

Very few projects fail because of technical reasons. Most of the project managers focus on mitigating the technical risks which usually don't occur in software development projects. We will discuss the most common non-technical rules which can give a lot of cost advantage to the companies.

Resource Planning – Have Right People in the Right Project

Human resources are the most expensive component in knowledge based industry like software development. So it is important to make sure that the right resources are selected for a project. Most of the companies/project managers just assign number of people without evaluating their capability or suitability for a particular project.

Understand and Agree on the Key Critical Factors

The key success factors for each project is different. We should have a clear understanding on the key success factors and agree on that with the customer. The project will be a success if the customer and the development company are able to identify key success factors. Some of the customers have everything in high priority and it will never happen.

Most probably you will end up in a state where you implemented a feature which you don't need and miss an important one if you target for everything. You will have a problem if you cannot prioritize. For example the key success factor for an airline navigation system or medical diagnostic system is completely different from the priorities for a social networking site or a general web portal. The priorities for a telecom application are different from an e-commerce application and so on.

Select the Best Tools and Technologies

Make sure that you select the best tools and technologies. Most of the projects require very minimum technology innovations. So the innovation should be to utilize best tools and technologies available for faster execution cycle. The infrastructure should be setup to make sure seamless working among the team members (Eg: Version Management, Database sharing, Change control, capability of hardware). We should also make sure that the best debugging tools are used as part of the development environment.

Make QA Team Involved in the Beginning

Most of the cases the QA team will be involved after the implementation is completed. It is important to make sure of their involvement from the beginning to assure they understand the requirements and business impact of the project thoroughly. They should also be ready with the test cases and test environment by the time the feature implementation is completed. They can also get familiarized with the system in development, versions etc to start the real testing as soon as implementation is completed. This will assure that there is no time lag between development and testing phases.

Proper Risk Management Plans

Project managers should have a very good risk management strategy and mitigation plan. The plan should have processes for planning the risk management, identification, analysis, monitoring and control. Enough buffers or contingency is to be made to cover the occurrence of risks. Occurrence of a risk can drive the project schedule way out of the plan and may cause lot of rework. A common risk which usually occurs is a design error and improper architecture. So it is important to involve very senior architects to design the system to avoid this risk.

Identifying the risk should be done at the beginning itself before the project is initiated and the risk will increase as the project matures. The risk has to analyzed in terms of the degree of impact to the schedule, scope, cost etc of the project. This will be used for assigning priority to the risk. A risk register can be maintained to enter the risks and documented as a risk statement.

Table 1: Sample Risk Register

Project Id	Risk	Time (T)	Cost (C)	Quality (Q)	Impact (T+C+Q)
1	Risk 1	5	6	4	15
2	Risk 2	3	5	6	14
3	Risk 3	2	3	1	6

Do Fast Tracking in the Planning Stage

The project managers should spend lot of time in the planning stage to do fast tracking as much as possible. This can shrink the schedule a lot which can result in more project buffer to absorb any risk which occurs while we progress.

Select the Right Software Development Model(SDLC)

Each SDLC model has its own advantages and disadvantages. The project effort and its schedule can go out of control just by selecting a wrong development lifecycle. The stake holders should identify the right lifecycle after evaluating the factors such as:

- The maturity of the requirements,
- Market window,
- Product roadmap,
- Maturity level of the development team etc.

It is not an easy decision as there are multiple factors involved in this. So it is advisable to take an expert opinion in this decision.

Projects are divided into phases to assure the smooth execution and tracking. The projects are typically divided into phases at Pumex Infotech Pvt. Ltd to ensure that the project is executed well within the schedule and with definite deliverable at the end of each phase. The table given below is an example of the various phases of the development with the key activities and output or deliverable at each phase.

Table 2

SI No	Phases	Key Activities	Deliverables / Output
1	Pre -Sales	Solution definition, technology, Requirements analysis, Budgeting	Finalized requirements doc, Signed contract, NDA, High level schedule
2	Planning	Resource allocation, Project plan, Test strategy, Tools & Techniques, Risk analysis	PMP, Risk plan, SDLC finalization.
3	Execution	Implementation, Unit Testing, Documentation	Code, Installation files
4	Quality Assurance	Do quality assurance as per Quality procedure	Bug report, Updated test report, Updated code and install files
5	Beta Testing	Beta testing, Customer acceptance testing	Bug reports from beta phase
6	Closure	Administrative closure, Final deployment, Source code handover	Final Source code, Install files, Customer feedback

Quality of Development

The developers should be trained for basic quality in the implementation which can be achieved without any additional effort. Usual tendency for the developers is to develop something and give it to the testing team /customer to catch. Eg; Making a proper alignment/fonts of UI, Commenting the code as we develop, Following the coding standards while development etc will not take any additional time. We see that many programmers run an extra cycle to do these things which will consume project time.

Many of the projects have the maximum slippage in the testing phase. Multiple cycles of bug fixes, re-opens etc will hold up the resources and will end up in high cost overrun. In fix bid projects companies lose all their bottom line in this phase after the beta release and customers are not in hurry as they already have something for their marketing.

Definition of Acceptance Criteria

All projects should have a clear acceptance criteria defined and agreed by all the stake holders in the beginning of

the project. Most of the projects will have a tough time getting approval for the development cycle as there is only a thin line between development completion and start of maintenance phase. So we should use the acceptance criteria to define the development completion and transfer to maintenance phase.

Manage Changes Every Day

The project scenarios will keep changing and the major role of a project manager is to track the situation in every step and take alternative methods to minimize the impact. This can be with respect to

- The resources,
- Their output,
- Certain team work synergies,
- Risk mitigation,
- Technology changes,
- Requirement management etc

In most of the cases project will not progress as planned due to changes in the Scenarios listed above. So the project manager should contribute more in tracking and change as per the situation keeping the schedule commitments in mind. An initial plan will help to get started on with the things. But the plan should be constantly updated as we progress and the project manager will be successful if he/she can add more project buffers as we progress by innovative tracking/change management mechanisms.

CONCLUSIONS

The cost reduction or cost overrun challenges are key for software industry in the coming days. Surprisingly the cost increases for applications are usually not because of the complexity of applications or competence of people to get it executed. Usually it happens because of some non-technical factors described above. So little attention to these factors can have heavy cost impact on the projects.

REFERENCES

1. Bowman Bob Hughes, Mike Cotterdl. Fourth Edition Software Project Management.
2. Paul C. Dinsmore, Jeannette Cabanisbrewin, The AMA Handbook of Project Management, Second Edition.
3. Walker Royce, 2010 Software Project Management, a Unified Framework.
4. K. Chandrashekhar Shet, 2005, Software Engineering and Quality Assurance.
5. Roger S. Pressman, International Edition. Software Engineering, a Practitioner's Approach.

