

Satyam Computer Services Limited

presents

**“Effective Management of ‘Fixed Bid’ Software
Development Projects”**

for

Project Management Leadership Conference 2007

Authors:

Balasubramanian Kumaraswamy – Balasubramanian_Kumaraswamy@satyam.com
Narayan Thirumalaithattai – Narayan_Thirumalai@satyam.com

Address:

**Satyam Computer Services Limited
Chennai**

Effective Management of 'Fixed Bid' Software Development Projects

By Balasubramanian K and Narayan TT

Introduction:

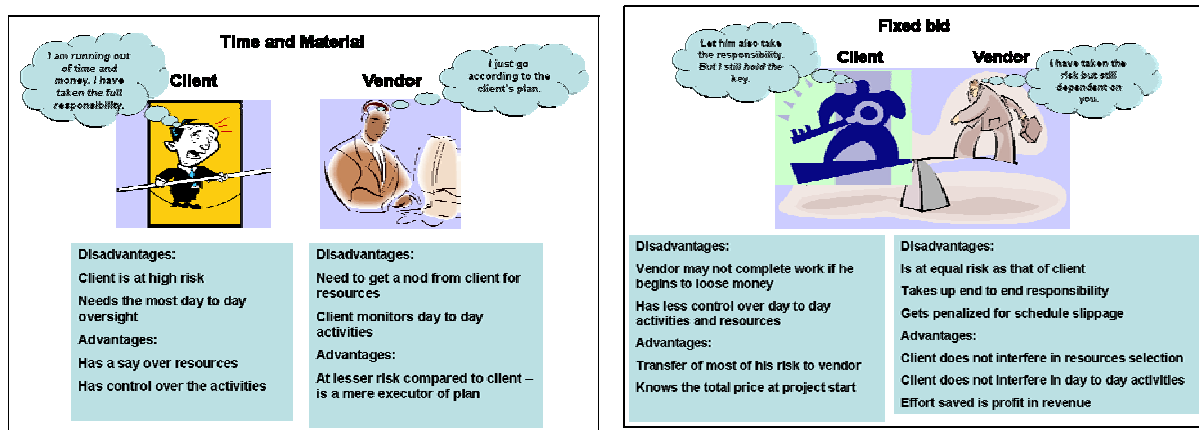
Project management is the discipline of organizing and managing resources in such a way that these resources deliver all the work required to complete a project within defined scope, time, and cost constraints. The project is a carefully selected set of activities chosen to use resources (time, money, people, materials, energy, space, provisions, communication, quality, risk, etc.) to meet the pre-defined objectives. Software project management also falls into the above definitions.

Traditionally, the constraints in project management have been listed as: scope, time, and cost.

These three constraints are often competing constraints. The discipline of project management is about providing the tools and techniques that enable the project team (not just the project manager) to organize their work to meet these constraints. These constraints have varying degree of impact on the project management based on the pricing model adopted for the project. One of the most important decision parameters in a project is the cost. Most of the other important parameters are also equated to cost. Therefore the pricing model assumes great importance in a project.

Predominantly there are two pricing models in software projects – Time and Material and fixed pricing or fixed bid model (Time and Material with cap is also similar to a fixed bid). The former was the widely used model upto late 1990's during the early stages of IT evolution when both the client and vendor were not sure on the requirements and the development effort. Later on in the late 1990's with the IT evolution entering into a matured phase, the fixed bid model also started gaining more prominent. "Fixed price contract is the most common type of contract in the world" – Rita Mulcahy, PMP, 2005. In current days, clients want to be in a safer side and that's why they prefer Fixed bid projects. **In fixed bid projects it is the vendor who will have to chase the client for approval or other requirements.** In T&M, it was the client who was behind the vendors, since he was paying for every hour for each person on the vendor side. So his responsibility was to see all the vendor people are loaded sufficiently and the project is completed as per schedule.

There are success stories in T&M projects. But very recently clients are beginning to feel that Fixed bid type is very beneficial. At this point they should remember **that for a project to be completed successfully in fixed bid type, immense client maturity is required in process and project management activities and the requirements should be stable. Without the client is being disciplined in his approach, no fixed bid project can succeed.**



As said in the picture above, the customer transfers most of his risk to the client in a fixed price model. But the irony here is that it is the client who still holds the key, which poses a great challenge for the vendor.

Fixed Price model is suitable for projects with well defined requirements and schedules. When the need is clearly specified in sufficient detail, the vendor estimates the project scope and complexity, and provides a project delivery schedule along with the fixed price for the complete product development. Under this business model, the vendor should ensure

Effective Management of 'Fixed Bid' Software Development Projects **By Balasubramanian K and Narayan TT**

quality delivery of the final product within a pre-scheduled timeframe, and the customer pays a mutually agreed fixed price for the project implementation.

Challenges in handling a fixed bid project:

As said before, in a fixed bid project, the vendor should be in the driver's seat. The project manager from the vendor side should satisfy the following areas:

- Adhering to the deadline fixed by the client.
- Profitability for the organization
- Quality of deliverables
- Minimum attrition level in the project

To satisfy the above areas, the following are the challenges:

- ✓ Managing the idle time of resources
- ✓ Excellent project monitoring and tracking system
- ✓ Adherence to Quality Management systems and processes
- ✓ Work load balancing
- ✓ Reduction or prevention of defects
- ✓ Standardization of process to improve quality of deliverables
- ✓ Rigid change management system
- ✓ Clearly mentioned acceptance criteria
- ✓ Freezing of the scope
- ✓ Introducing Rewards and Recognition program for improvements

So this paper discusses on the project management practices for the vendor to guide him in this challenge in managing a fixed bid software development project.

It focuses on the ways by which challenges in the fixed bid software project can be overcome with effective project management methodologies.

It elaborates on:

- The applicability of Lean project management principles in Fixed bid Software project management with practical illustrations
 - This section introduces the lean management principles which are also known as Toyota concepts of management. It also elaborates on each of the lean management principles with practical illustrations on applicable concepts
- Suggested best practices for fixed bid software project management based on real time experiences
 - This section illustrates on the best practices of fixed bid software project management based on practical experiences of project managers.

Software Project management should become mature:

As discussed above, there are challenges which need to be overcome to achieve success in a fixed bid project. For this to happen, the software project management processes should become more mature and rigid, something very similar to the Manufacturing and Automotive industry projects scenario. Today Manufacturing companies are monitoring their defect rates in parts per million (ppm) and even in that they are targeting line defects at 100ppm. Their processes have stabilized and they are able to focus on continuous and breakthrough improvements. In the similar lines software project management also should get matured by making it process driven and adopting stringent processes. By this, the projects can become more disciplined which is a vital factor for the success of fixed bid projects. For this to happen it needs an out of box thinking on the software project management. Let's think a software project management in line with a manufacturing project management scenario and try to map in those concepts.

One another similarity is that manufacturing projects are all fixed bid, except for erection and commissioning projects which may be T&M.

When we say manufacturing or automotive management scenarios the first thing which strikes our mind is the Lean management principles as introduced and implemented successfully by Toyota.

Effective Management of 'Fixed Bid' Software Development Projects By Balasubramanian K and Narayan TT

Toyota concepts to manage software projects:

As software systems grow more complex and more people become involved, managing software development becomes increasingly difficult. For years, enterprises have been told that to cope with the increased complexity, they need to add more measurements, more controls, more checks and balances, more rigors. To ensure that people do as they are told, many enterprises expend significant resources on process compliance audits. What we need is a way of thinking about software development that emphasizes rapid organizational learning and directs all our efforts towards generating customer value. That is precisely what Lean Software Development seeks to do.

The notion of the "Lean Enterprise" originated at Toyota after the Second World War. Over the years it evolved into an industry-independent management philosophy focusing on delivering more customer value while at the same time improving growth and profitability.

There are five core Lean principles:

- Value: understand what adds value for the customer
- Value Stream: understand how the organization generates customer value
- Flow: maximize speed and minimize cost by achieving continuous flow
- Pull: deliver value on a just-in-time basis based on actual customer demand
- Perfection: continuously improve the performance of your value streams

These principles form the basis for a large number of enabling practices - powerful tools and techniques that help enterprises maximize customer value and avoid waste. Companies implementing Lean have experienced up to 50% cost reductions and 95% lead time reductions. In this article we will discuss how Lean principles can be applied to managing software development.

There are 14 Lean management principles from Toyota, which are the best practices in Lean management. Following are some of the lean principles which are framed for suitability for software development projects that can be adopted for success in Fixed bid projects. The concepts of lean management were discussed with various project managers who are managing fixed bid projects. Based on their experience the some of the best practices that can be adopted to effectively manage a FB project have also been given for the lean principles.

Lean principle No.1:

Generate value for customers

Evaluate every function in the organization in the terms of its ability to generate value for the customer and other stakeholders.

Many product managers do not really understand how the customer's business makes a profit. The justification for a product feature should be that it adds value by helping the customer perform better in some quantifiable way, such as improving profits or sales, or saving time or money.

Anything that does not directly contribute value to the customer is waste. Perhaps the single biggest source of waste in software development is unused functionality. **Only 20% of features in enterprise software applications are used on a frequent basis, 45% are never used, and 19% are rarely used, according to a research by the Standish Group.** Other common sources of waste in software development include: include defects (rework and testing), unnecessary paperwork, waiting, task switching, and waste of human talent, unused metrics, and improper use of software tools.

By analyzing your customers' business process you are in a much better position to understand where and how you can add real value. If possible, do your study in a way that involves your customers - visit with them, form joint analysis teams, write joint reports, etc.

Best Practices suggested:

- Identify Acceptance Criteria for the deliverables.
Identify a clear acceptance criterion for all functionalities and get it mutually agreed with the client.

Effective Management of 'Fixed Bid' Software Development Projects By Balasubramanian K and Narayan TT

- Introduce "Mock-up" screen sign off
It is recommended that sign offs for the screens can be obtained through building a "mock up" screen in MS Excel. This way of prototyping is better to be followed than the usual Software Requirement specification document, since it will give a clear understanding of what is required by the client. It should be taken care that this prototype needs to be versioned and the change history to be maintained.
- It is recommended to get to know the number of Reports that may be needed to be developed as they form a vital part of scope. It will have a significant impact on schedule and effort if the numbers of reports are on the rise. Therefore it is suggested to freeze the estimated number of reports and further addition to the list to be done on an extra cost.

Challenges addressed:

- ✓ Clearly mentioned acceptance criteria
- ✓ Freezing of the scope

Lean principle No.2:

Redesign work processes to achieve high value added continuous flow

Organize a Value Stream with a Value stream manager

A value stream is the set of activities required to develop and deliver a product or service. It usually involves multiple processes, teams, and departments.

If we look at each of the processes in a value stream and ask who is responsible for day-to-day performance as well as process improvement, the answer is usually that no single person has overall responsibility. Instead the responsibility is distributed across several functional managers, who may have conflicting interests. This is why process improvement efforts often get "out of synch" with day-to-day execution. By appointing a Value Stream Manager as responsible for both day-to-day results and for coordinating process improvement efforts for each value stream, we get clear accountability for execution excellence.

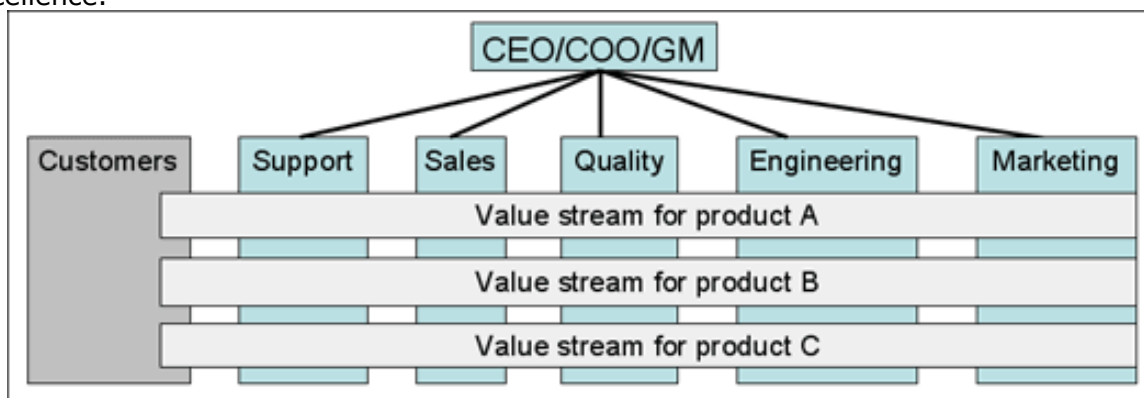


Figure 1: An organization can be viewed as a portfolio of value streams

In commercial product development, the product manager role is the most natural job to expand in this manner. In an IT organization, it might be a project manager. By organizing in this way, the project manager can easily come to know any problem which is affecting his value stream and he can take steps to resolve it at the earliest.

Maximize speed and minimize cost by achieving Continuous Flow

Once we have discovered what our value streams actually look like, we can begin to look at how efficiently they generate customer value. The concept of flow requires us to rethink how we utilize elapsed time. How much of the time elapsed is actually spent on activities that add value to the customer?

Flow implies a significant change of perspective. We are no longer so interested in how busy we are keeping our people, or how "efficiently" they perform individual tasks. Instead, our primary focus is on making efficient use of elapsed time. Flow is what occurs when we utilize elapsed time while adding value to the customer.

Effective Management of 'Fixed Bid' Software Development Projects By Balasubramanian K and Narayan TT

Continuous flow means that individual units of customer value (a product function, for example) move through the value stream with no time spent on non-value-added activity. In a typical organization, only 1/3 of the 24-hour day is spent on working, the rest is downtime. Many companies still organize their development processes as a sequence of distinct stages, such as concept development, requirements definition, design, etc. Each of these stages results in a big batch of partially completed work. Partially complete work constitutes in-process inventory, which is a form of waste. Inventory clogs up the development pipeline, ties up resources, and increases the time it takes to respond to customer needs.

Smaller batches of work lead to less inventory, which means that less time is lost due to waiting. It also makes easier for customers who are responsible for sign off. This is why iterative development is faster than the traditional waterfall approach.

Aside from waiting stemming from big batches, inhibitors to flow in software development include:

- Poor or missing specifications - lack of information brings work to a standstill
- Inflexible architectures - these can make it awkward to progress incrementally
- Defects - they make us waste time on non-value-added rework and retesting
- Unbalanced resources - these cause inventory pile-ups and waiting
- Poor configuration management - not being able to access artifacts causes delays
- Task switching - individual's waste time trying to juggle multiple activities

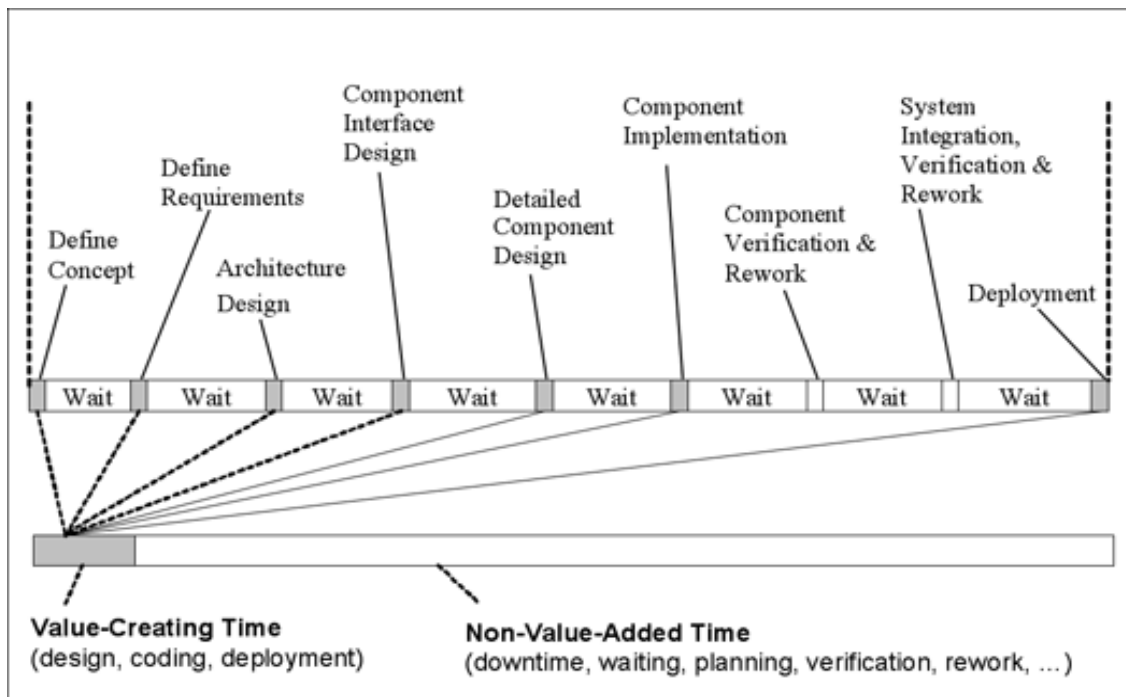


Figure 2: Lead time for an individual software development

In software development, much of the development effort is NOT value-added from the customer's perspective. Even in organizations using "Agile" methods with smaller iterations, there are significant opportunities for improving flow. By improving flow, we not only improve speed, we also achieve substantial cost savings.

Effective Management of 'Fixed Bid' Software Development Projects By Balasubramanian K and Narayan TT

Best Practices suggested:

- Scheduling:
 1. In any project there will be intermediate deliverables and their sign off before proceeding to next phase. For example, in our project we had intermediate deliverable like Technical design document and its sign off before moving to the development phase. An independent activity can be scheduled overlapping the Technical design phase, which can be carried out and finished before the development phase starts. The overlap time can be accumulated at the end of the schedule as a follow up activity for bug fixing or sign off.
 2. Similarly, the most difficult part to manage would be the idle time of the project team members. The above scheduling technique can address this problem and another way to manage would be to consider idle time as a part of plan itself. By addressing idle time, it would also keep the resources engaged which can also reduce the frustration of being idle and can lessen resource attrition.
- Development:
 1. Implement a practice of "Re-usability" of codes that have already been developed when a similar functionality is being developed. This will save time and will avoid wastage of effort.
 2. Implement a joint change advisory board to decide upon the change requests and approving only the needed changes during development. The change requests and impact should be communicated to all stake holders immediately. All change requests need to be charged extra based on the mutually agreed per hour rates.
- Estimation:
 1. Incorrect estimation will always lead to disaster. We should be able to estimate atleast 95% closer to the actual that can be obtained i. e our estimation accuracy should be 95% or more. In the recent times estimates based on Function points have been more accurate than the other traditional methods. So it is recommended to use Function point estimates. In other cases where FP is not comfortable there should be a mutually agreed estimation methodology.
 2. The project should also follow a practice of re-estimating when there is an anticipated delay or change. For fixed bid projects they should be highlighted as risks to the client and re-estimate should be given to them.
 3. It is recommended that the contract should contain a clause for "changes will be charged extra as per the actuals"

Challenges Addressed:

- ✓ Managing the idle time of resources
- ✓ Excellent project monitoring and tracking system
- ✓ Rigid change management system
- ✓ Resource attrition

Lean principle No.3:

Use "Pull" systems to avoid overproduction

Synchronize development with the actual demand rate

Even with continuous flow, if we cannot deliver functionality on time, the result is that the customer is left waiting, as is our financial reward. If we deliver functionality that is not (yet) needed, we are also wasting time and money. To avoid both types of waste, we need actual customer demand to explicitly drive the content, volume, and tempo of our work in real time.

Instead of "pushing" work through the value stream based on forecasted demand and up-front-planning, Pull means that we only build something when it is actually needed by a customer. In other words, we want customer(s) to "pull" new functionality releases from the value stream. Traditional plan-driven work is replaced by demand-based work, reducing inventory and improving our ability to react very quickly to customer needs.

In many cases customers cannot take deliver of new product releases as quickly as we are capable of releasing them.

Effective Management of 'Fixed Bid' Software Development Projects

By Balasubramanian K and Narayan TT

To avoid big batches of work, this requires us to establish "artificial pull", where internal releases are produced based on feedback from a representative subset of our customers. Intermediate small batch releases with customer feedback substantially reduce the risk of developing irrelevant functionality or design solutions that turn out not to work well practice.

If we want customer feedback to drive our development efforts, we must address a number of practical issues, including the selection, composition and priority of customers providing feedback. In a situation of large numbers of customers, we must carefully choose a representative group of customers, and carefully weigh the relative priority of their feedback. Another practical issue is infrastructure - if we are going to deliver and deploy new versions frequently, we must deploy the necessary infrastructure to make this work smoothly. Frequency of feedback is also a key issue - the more often we can get customer feedback the better, but this means the participating customers must be convinced of the necessity to stay involved throughout the effort.

Frequent customer feedback requires trust. One way to establish such trust is to visibly invest in an effort to understand the customers' business problems and opportunities. We must also explain how the feedback will be used to add real value to customers. To retain and enhance the trust that has been built, we must show again and again that we act promptly on the feedback we receive. In some situations it also makes sense to contractually formalize the understanding that has been reached. This is especially true for custom software development.

Best Practices suggested:

- **A practical example:**

When this system was referred to a software project manager, there were two questions raised by him:

- 1) How to implement this practically?
- 2) How can you ensure Quality in a pull system?

Both these questions are very valid, considering the fact that this is an infant in the software projects, while it is matured in manufacturing and automotive scenarios.

The following answer had convinced him.

For implementing the pull system, we should understand that the pull originates from the customer for the whole development, and for the phases, it originates from the team members of the succeeding phase.

For example, the deliveries are to be planned based on the client's requirements. There may be a project plan but it may not contain micro details. For example, in a SAP project contains the following hierarchy in descending order - Modules, Packages, Tracks, Objects and Methods. Thus only if all the methods are delivered the object can be delivered and only if all the objects are delivered, tracks can be delivered and so on. The project plan might be at track level. The objects to be delivered are prioritized based on their importance and relevance for the functioning of other interdependent objects. In the pull system, the deliveries should be demand based, which ensures that we react quickly to customer needs and reduces unnecessary or waste work.

Similarly, we can apply pull system internally, where the team member responsible for the succeeding phase acts as a customer who demands the deliverables based on his plan, which he would have arrived based on the demand raised by the next level. By this the demand by the customer will be linked to the demand raised by the owners of the phase and ultimately the final customer's demand will be satisfied.

The answer for the next question was a suggestion taken from the manufacturing scenario. In a world class manufacturing company, you could find that there are 3 levels of checking of the product identified in the work procedure for that station. They are incoming, process and outgoing checks. A similar system can also be implemented in software projects.

For each phase there can be incoming, process and outgoing checks of the product that is being developed. For example imagine a development phase having a preceding phase of Technical design development and a succeeding phase of Unit Testing.

Effective Management of 'Fixed Bid' Software Development Projects

By Balasubramanian K and Narayan TT

In this case, the incoming checks can consist of:

- Reviewing the checklist with which the Technical design document was reviewed.
- Confirming that all the defects and observations during the TD document review have been incorporated in the document, by verifying with the peer review report.
- Verifying for the latest version of the TD document.

The process checks during development can be:

- Checks for not missing any requirements as mentioned in TD document.
- Checks for following the standard coding guidelines.
- Proper commenting of codes

The outgoing checks can be:

- Code review with the code review checklist.
- Updating the peer review reports for the findings.
- Updating the defect log for the findings.
- Correct the findings and close the review.

• **Back to Back SLAs with customer:**

This is very much required for a Fixed bid project. By having a back to back SLA with customer, we make him responsible to pull the deliverables from the vendor. The time that can be taken for review and sign off should also be given in SLA. By this there will be a binding on the customer to review and sign off a deliverable within a specific period and pull the next deliverable. The SLA should also specify that deliverables will be made based on client's requirement and within the mutually agreed time between the client and the vendor.

- There should also be a timeline or plan specified for sign off of every intermediate and end deliverables. It should also be ensured at the time of planning and scheduling itself that the client is available for issue resolution and sign off

Challenges Addressed:

- ✓ Managing the idle time of resources
- ✓ Excellent project monitoring and tracking system
- ✓ Adherence to Quality Management systems and processes

Lean principle No.4:

Use visual control so no problems are hidden

This concept is based on the idea that "A picture is worth 1000 words". A picture communicates things faster and easier than a page of report. To manage a project successfully, the status of the project should be monitored critically. More than this the Project manager should be able to predict and anticipate problems and be highly pro active to mitigate risks. This is very much true in the cases of Fixed bid projects, where you need to be extremely pro active.

Use simple visual indicators to determine whether the project is proceeding in its normal course or deviating from it.

"Kanban" comes very handy here. Kanban is a Japanese word for ticket or sign.

It is a very good visual control system, which helps in identifying the deliveries to be made.

Best Practices suggested:

- Have a big white board divided into 4 sections – Deliveries, Issues and Inputs, Issues.
- Have packs of green and red cards which can be stuck (removable like a gum strap).
- List down the work products to be delivered for a month or for a week.
- Place a green card against the work products which do not have any issue for delivery and are going as per schedule. Place a red card for deliverables which have issues and are not moving as per the schedule. Place the issues which are to be addressed for timely delivery. Follow the same steps for inputs that are to be received for delivery.
- Review the board daily with the team and module leads. This practice ensures that all the issues are risen up and are solved on a day to day basis. This will avoid surprises at a later stage. A similar set up can also be made for monitoring effort in the project.

Effective Management of 'Fixed Bid' Software Development Projects By Balasubramanian K and Narayan TT

- The project manager can also have the run chart for basic metrics like schedule and effort variance, defect density, rework effort, etc... which will enable to identify improvement actions. Always remember "Out of sight is Out of Mind"

Challenges Addressed:

- ✓ Excellent project monitoring and tracking system

Lean principle No.5:

Level out the load (Heijunka)

The best-known Lean technique is the elimination of "Muda" or waste, but you can only do that if you also eliminate two other M's:

- **Muri or Overburdening people or equipment.**

Overburdening people results in safety and quality problems. It can lead to burnout, which makes you lose good people and lots of knowledge and experience.

Following are the major contributors of waste in software development:

They are:

- Partially done work
- Irrelevant processes
- Extra features
- Task switching
- Waiting
- Duplication of documents
- Defects

Eliminating these wastages can lessen the load in a software project.

- **Mura or Unevenness.**

Dividing a software release into relatively small batches is an example of Mura. If each batch can be implemented in (at most) a few days, another batch can be started by which we can feel that we're making progress.

It is also easier to estimate, more flexible for planning, make stakeholders really think about what we need, give a micro level tracking, which is easier to understand. By leveling out the work load people in the project also feel that they are not overloaded which can reduce the resource attrition level in the project.

Best Practices suggested:

- To avoid partially done work, adopt a system of "multiple responsibilities". This is something like making one responsible for Technical design and another for code development. This will create an internal pull and will ensure that no work is partially complete unless forced by external circumstances. This will also help in leveling out the load especially in the case of complex programs.
- It should also be ensured that people are not loaded due to some additional changes which were not part of initial development. In those cases the additional changes should be charged separately. This is true in cases where team members in a tendency to be good books of client, mutually agree to do certain additional works which might increase the load affecting the schedule. This should be monitored and avoided.
- The project should also take initiatives in reducing the effort and reducing the defects. For this six sigma methodology can be adopted which can give a significant improvement in this direction.

Challenges Addressed:

- ✓ Work load balancing
- ✓ Resource attrition

Effective Management of 'Fixed Bid' Software Development Projects

By Balasubramanian K and Narayan TT

Lean principle No.6:

Build a culture of stopping to fix problems, to get quality right the first time

This is an interesting concept called First time right – Right every time.

Building software today means that you are in it for the long haul. And you know in the fixed bid projects, that there are more demands—and fewer resources—to meet those demands. Getting the software right—the first time—is the only way to succeed under these circumstances. Today's requirements process is incremental with quick cycle times. It uses prototypes and scenarios, and the requirements process ensures that you get the right result by writing a fit criterion—a test case for the requirement.

When software has too many defects, developers spend more time fixing the software defect than they spend writing it in the first place. Most organizations have found that an important key to achieving shortest possible schedules is focusing their development processes so that they do their work right the first time. "If you don't have time to do the job right, where will you find the time to do it over?"

Here the Project managers should focus on defect prevention. It not only means conducting defect prevention activities, but a little more than that.

"Zero defect" should be made a culture in the project and Rework is to be eliminated.

Best Practices suggested:

- Track defects in a regular frequency set by the project manager based on the project's requirement. It is suggested that it can be weekly twice for high critical applications and weekly once for low and medium applications in a project. Use a simple excel sheet to capture the details of the defects like who, when, where and how much.
- For each defect conduct a small review for identifying the cause of the defect
- Use Plan-Do-Check-Act (PDCA) approach for identifying and implementing the action plans for defect prevention
- Monitor the status of the action plan and re occurrence of the defect. A CAPA sheet can be used for this purpose – CAPA – Corrective action and Preventive action – the defects can be monitored for re occurrence after implementation of the actions using this sheet.
- Recognize the team members for a Zero defect day and get their commitment for sustaining the Zero defect status.
- Aim for 100% First time right and keep raising the stick starting from 50, upto 100%

Challenges Addressed:

- ✓ Reduction or prevention of defects

Lean principle No.7:

Standardized tasks are the foundation for continuous improvement and employee empowerment

Processes add value to both management and the development team, although the reasons differ.

*From a management perspective, processes are needed to:

- Explain to team members how to perform work tasks
- Transfer knowledge from more experienced to less experienced team members
- Assure predictability of work activities so that approximately the same deliverables will be produced with the same resources each time the process is followed
- Establish a basic set of work tasks that can be continuously improved
- Provide a means for involving team members in improving quality, productivity, and customer satisfaction
- Free the management from their activities associated with "expediting work products" to spend more time on activities such as planning, and customer interaction

* From CSQA Body of Knowledge

Effective Management of 'Fixed Bid' Software Development Projects

By Balasubramanian K and Narayan TT

*From a team member perspective, work processes are important to:

- Increase the probability that the deliverables produced will be the desired deliverables
- Enable team members to better plan their workday because of the predictability resulting from work processes

This mainly focuses having a well defined procedure for each activity. This is different from the normal Quality processes which will be present for the organization as a whole. Use stable, repeatable processes everywhere to maintain predictability, regular timing and regular output of the processes. Capture the accumulated learnings from best practices and standardize them in the processes.

Best Practices suggested:

- Prepare work processes for the project activities. These are work processes at a lower level than the Quality processes for the organization. For e.g., there can be a work process for work products development and review, estimation of effort, metrics collection, etc...
- There can also be project specific review checklists, which can guide reviewers for an effective and efficient review, instead of having a general checklist which takes more time in understanding and then reviewing. By having a basic checklist it can be improved upon to improve the effectiveness of review.
- It is also a good practice to set a time period after which the processes will be reviewed for alterations to suit the existing standards and practices. They can also be reviewed to reduce the effort for the activity.

Challenges Addressed:

- ✓ Standardization of process to improve quality of deliverables

Lean principle No.8:

Continuous Improvement and People development

Companies implementing Lean provide teams with skills and incentives to relentlessly identify and remove waste. A typical improvement process used by a cross-functional change team to improve a value stream's performance might look as follows:

1. (Re-) Define Customer Value
2. Map the value stream
3. Identify sources of waste
4. Design "future state" of value stream (what we want it to look like)
5. Implement the future state
6. Present results

To encourage a higher rate of improvement, teams can be engaged in friendly competition, but they should be required to share their innovations on a regular basis.

For continued performance improvement, an organization has to improve its ability to harvest, create, and share knowledge. This can be done on several levels. Within teams, try cross-training team members and occasionally switching roles. Rotating people between product teams can also be helpful. Advanced practitioners such as Toyota and Nokia organize groups around areas of domain expertise. Each domain group encodes its expertise in the form of continuously improving iterations of reusable components.

For additional learning across teams, consider establishing Communities of Practice in strategic areas such as Customer Relationship Management, Software Design, Usability, Finance, Strategy, Lean Thinking, and Team Leadership. Such groups should combine ideas from other companies with lessons learned from past and current projects. This will also make the relationship of the team member more strong with the project, which will help in retaining the resource for longer times, where needed.

Effective Management of 'Fixed Bid' Software Development Projects By Balasubramanian K and Narayan TT

Best Practices suggested:

- Identify the major contributors of wastage in the project.
- Form teams among team members, with around 3 per team.
- Provide the teams the list of problems in the project leading to wastage.
- Other than wastage there can also be other issues which affect the Quality, Productivity, Cost and Delivery of the Software product.
- The teams can choose their area and perform the continuous improvement projects known as "Kaizens".
- It is necessary to have a check point or metrics as number of Kaizens performed every month. It is also suggested to have a target for this.
- The Kaizens can be reviewed and the best of them can be published in a news letter of the unit or department. They can also be rewarded suitably.

Challenges Addressed:

- ✓ Introducing Rewards and Recognition program for improvements
- ✓ Resource attrition

Other Best Practices suggested:

The following recommendations are being made for general project management for the vendor to protect himself during Fixed bid project execution.

- Requirements gathering phase should be in T&M mode and the other phases in the development can be in Fixed bid mode. This should be especially insisted where the client is not sure of the requirements and the requirements are not complete. If the client is very clear on the requirements then this can also be taken on the fixed bid mode.
- Initially one module can be taken in fixed bid mode. Based on the success factor and the problems faced, it can be decided whether to continue with fixed bid mode or change to T&M mode.

Appendix:

Acronym / Term	Definitions
T&M	Time and Material
FP	Function Points
TD	Technical Design
SLA	Service Level Agreement
PDCA	Plan Do Check Act
CAPA	Corrective Action Preventive Action

Effective Management of 'Fixed Bid' Software Development Projects By Balasubramanian K and Narayan TT

References:

- The 14 "Lean Management Principles" of the Toyota Way by Jeffrey Liker
- Article on Introducing Lean Software Development by Frode L. Odegard
- Lean Software Development - An agile toolkit By Mary Poppendieck, Tom Poppendieck
- Lean Software Strategies: Proven Techniques for Managers and Developers By Peter Middleton, James Sutton

About the authors:

Balasubramanian Kumaraswamy is a Graduate Mechanical Engineer from Regional Engineering College (REC), Suratkal, India. He is a Certified Project Management Professional (PMP) from PMI with eight years of experience in SAP, in the areas of Implementation, Development, Production support & Maintenance projects. He has successfully managed the Satyam's onsite-offshore development model project for five years with global clients. He has also got expertise in SEI-CMM processes and six sigma methodologies. He has seven plus years of domain expertise of Logistics, Sales & Marketing in Capital Equipment & Engineering industries

Narayan Thirumalaithattai is a Graduate Mechanical engineer from University of Madras and has over 5 years of experience in Quality Assurance. He is a "Certified Software Quality Analyst" (CSQA) from QAI. He had worked in the implementation of ISO 9001: TS 16949 and QS 9000 Quality Management systems. He is a qualified Internal Auditor. He is working with Satyam in implementation of CMMI processes on Software development and Maintenance projects. He has completed two six sigma projects under his leadership. He is also a certified "Green Belt" in six sigma methodology. He had also worked for the prestigious DEMING AWARD for Total Quality Management in his previous manufacturing organization.