A Study to Enhance Human-Resource Performance Efficiency for Minimizing Cost in Software Development Projects

Amrinder Kaur, Kamaljeet Singh

Abstract— Human resources in software development projects require a high level of individual intensity devoted to project tasks, which then is integrated collaboratively to complete the project. Human resources technical skills and implementation experience are key factors for project success and therefore must be allocated and managed judiciously. Despite of the efforts of organizations, problems related to cost escalation have been encountered regularly .Human resource efficiency and productivity shall be key for future success and sustainability of the businesses. Through this paper, the causes have been outlined in a study conducted on software development projects. The reasons outlined affect efficiency and productivity of employees in software development projects across various phases in software development viz. requirement, designing, coding/testing, implementation phase. And it is been devised that job satisfaction enhances when the job is more exploratory in nature.

Index Terms— human resource efficiency, software development phases, requirement phase, designing phase, coding phase, testing/verification phase.

1 Introduction

Human resource efficiency

Efficiency is one of the very few sustainable advantages that creates a significant barrier to entry and at the same time ensures profitability for companies [1]. The efficiency is defined as the ratio between input and output, and the effectiveness is defined as the achieving level of the expected production output by a production system. In fact, efficiency and effectiveness represent different levels of performance, and there is no guarantee that both of them can be achieved simultaneously. However, an efficient organization must handle both of them well, and use the most efficient way to pursue maximum effectiveness.

Therefore, these benefits to efficiency can be categorized as tangible and intangible [2] The tangible benefits include the following:

- Reduced cost
- Improved productivity (i.e., amount of output produced per unit of input)
- Increased market share
- Savings in labor
- Increased consumer surplus (i.e., the accumulated difference between consumer demand and market price)
- Improved customer service quality
- Improved organizational efficiency
- Quicker response to customers
- Deeper knowledge and understanding of customers

On the other hand, the intangible benefits include:

- Improved decision-making ability
- Superior product quality
- Knowledge/information management and sharing
- Improved coordination/relationships with partners

In all, Contribution to the Company shall be

- **Effective management of the project**: Increase in the efficiency of human resource will lead to effective management of the project. Thus, it will lead to appropriate utilization of resources, and attainment of the desired deliverables in settled time frame and within the expected cost and quality.

- **Lesser redundancy and repetition of work**: effective utilization of resources will cut down the extra work and thus the energies can be synergized for work that is more productive.

- **Better customer image**: Effective project management will create better customer image and hence larger gains in future.

Software Development process

A **software development process**, also known as a **software development lifecycle**, is a structure imposed on the development of a software product [3]

The development process according to water fall model consists of a series of steps starting from requirement collection, designing phase, coding phases, testing and verification phase followed with maintenance. The waterfall model shows a process, where developers are to follow these phases in order:

1. Requirements specification (Requirements analysis)
2. Software Design
3. Integration
4. Testing (or Validation)
5. Deployment (or Installation)
6. Maintenance

In a strict Waterfall model, after each phase is finished, it proceeds to the next one. Reviews may occur before moving to the next phase which allows for the possibility of changes (which may involve a formal change control process). Reviews may also be employed to ensure that the phase is indeed complete; the phase completion criteria are often referred to as a "gate" that the project must pass through to move to the next phase. Waterfall discourages revisiting and revising any prior phase once it's complete.[1]

Requirement Phase

The important task in creating a software product is extracting the requirements or requirements analysis. Customers typically have an abstract idea of what they want as an end result, but not what software should do. Incomplete, ambiguous, or even contradictory requirements are recognized by skilled and experienced software engineers at this point. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect.

Once the general requirements are gathered from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document.

Certain functionality may be out of scope of the project as a function of cost or as a result of unclear requirements at the start of development. If the development is done externally, this document can be considered a legal document so that if there are ever disputes, any ambiguity of what was promised to the client can be clarified.

Implementation, testing and documenting

Implementation is the part of the process where software engineers actually program the code for the project.

Software testing is an integral and important part of the software development process. This part of the process ensures that defects are recognized as early as possible.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development. This may also include the writing of an API, be it external or internal. It is very important to document everything in the project.

Deployment and maintenance

Deployment starts after the code is appropriately tested, is approved for release and sold or otherwise distributed into a production environment.

Software Training and Support is important and a lot of developers fail to realize that. It would not matter how much time and planning a development team puts into creating software if nobody in an organization ends up using it. People are often resistant to change and avoid venturing into an unfamiliar area, so as a part of the deployment phase, it is very important to have training classes for new clients of your software.

Maintaining and enhancing software to cope with newly discovered problems or new requirements can take far more time than the initial development of the software. It may be necessary to add code that does not fit the original design to correct an unforeseen problem or it may be that a customer is requesting more functionality and code can be added to accommodate their requests. If the labor cost of the maintenance phase exceeds 25% of the prior-phases' labor cost, then it is likely that the overall quality of at least one prior phase is poor.[*citation needed*] In that case, management should consider the option of rebuilding the system (or portions) before maintenance cost is out of control.

Bug Tracking System tools are often deployed at this stage of the process to allow development teams to interface with customer/field teams testing the software to identify any real or perceived issues. These software tools, both open source and commercially licensed, provide a customizable process to acquire, review, acknowledge, and respond to reported issues. (Software maintenance).

Literature Review

Literature analysis shows that organizations usually adopt the process of business analysis in IT to enhance efficiency. Business analysis as a discipline has a heavy overlap with requirement analysis, but focuses on identifying the changes to an organization that are required for it to achieve strategic goals. These changes include changes to strategies, structures, policies, processes, and information systems.[3].Typically, business processes work to enhance project efficiency by reducing re-work and reducing the process length to contain the cost.

Efficiency improvements for the projects will have the main resource in form of human resource and there performance efficiency is crucial for the success of projects

In 1978, Harry pointed out that efficiency; effectiveness and productivity are three major parts of performance. In 1988, Fortuin placed the organizational goal in two categories: efficiency and effectiveness [1].

Thus, effective critical knowledge and skills may indeed create some frog-leaping of some economies as was predicted by [4] that:
"Fifty years from now—if not much sooner—the leadership in the world economy will
be moved to the countries and to the industries that have most systematically and most successfully raised knowledge-worker productivity."

Objective of the study

Enhancing efficiency to minimize cost in software development projects.

Methodology of the study

Research Design

The research design was exploratory till identification of software development parameters. Later it became descriptive when it came to evaluation of parameters in relation to the cost escalation.

Research Sample

The work had been the case of a software company viz. Vayam technologies Ltd. and four projects in various stages of software development. The respondents were the employees and teams working at various locations in delhi /NCR. Around 50 employees were the respondents for the study.

Data Collection tool

Data was collected through structured questionnaires. The questionnaires were designed by the help of Questionnaire Design and Survey Sampling [8] and project managers working at various projects in the software development phases in the company. The filled-in questionnaires were collected from the respondents in two weeks' time.

Data Presentation

The data collected from the questionnaire was checked for its reliability using standard deviation and standard error techniques. All the questions had their population mean within 95% of the confidence interval.

And hence the parameters for cost reduction with the help of efficiency improvement had following categories.

Project Clarity

This is regarding the goals/objectives to be achieved for the individual project.

Work done with structured plans

Commencement of work in the individual phases in the software development is done through careful planning.

Report preparation

Work at every phase is complied with reports for future reference and for further tracking by the project managers and team members.

Discussion with other team members

This is necessary for bottlenecking, innovation and clarity. At the same time it can work to maintain the job satisfaction of the team members.

Monitoring and training

As the name suggests monitoring of work flow at appropriate stages is significant for tracking of project within the timelines .Training helps the employees to update their skills and knowledge.

Job satisfaction

Job satisfaction describes how content an individual is with his or her job. The happier people are within their job, the more satisfied they are said to be. Job satisfaction is not the same as motivation, although it is clearly linked. Job design aims to enhance job satisfaction and performance; methods include job rotation, job enlargement and job enrichment. Other influences on satisfaction include the management style and culture, employee involvement, empowerment and autonomous work position. [6]

Communication problem with customers and team mates

Ninety percent of the common problems within teams are poor communication skills or a lack of communication skills, and a lack of understanding or appreciation of each team member.

Findings

Based on the parameters identified above, employee's responses were quantified. And the table 1 below represents the summary of the data on the basis of responses of fifty respondents.

Factors	Requirement Phase	Designing Phase	Coding Phase	Testing/Verification phase
Project Clarity	50%	80%	90%	90%

Work is done with structured Plans	80%	80%	80%	70%
Report Preparation	100%	100%	40%	70%
Discussion with other team members	80%	90%	90%	70%
Monitoring and Training	60%	60%	40%	70%
Job satisfaction	90%	70%	80%	80%
Communication problems with customer or team mates	80%	60%	10%	20%

In the table, the columns are the percentages of respondents who believe that they do hold the corresponding factor in the row.

Analysis of the collected data

The above result is head wise analyzed as follows:

Requirement Phase

The conclusions of requirement phase are as follows:

- Team members were clear about the project goals and Objectives.
- According to the results, if respondents face problems to carry out certain work then only 30 percent of respondents try to solve problems on their own. Remaining either keep pleading for details or talk to seniors for direction. Hence, there has to be more self-reliance in the team to carry out their work. This needs suitable experience, regular mentoring and training in the candidates. Moreover, the candidates selected for requirement phase should have experience in dealing with customers, should have leadership skills and should be responsible to carry out their task. Definitely, seniors need to be consulted because the time factor and cost minimization are important. However, referring the seniors for every mundane task leads to more dependency, chaos and increases time to accomplish a task. This is because the seniors are more costly resources and they must put their energy and time to relatively more important work.
- The team / respondents need to be fairly trained, have leadership and problem solving traits, and customer-negotiation skills to cut down the cost.
- According to the results, around 60 percent of respondents need more training and monitoring to handle their jobs. However, around 20 percent of respondents sometimes feel that they need training and monitoring. Hence, requirement phase needs improvisation in training and mentoring to candidates so that they can better equip themselves with the task.
- According to the results, around 80 percent of respondents get communication problems with the customer while gathering the data. However, only 20 percent never feel such problems. Hence, the respondents / team need to be clearer about their project goals. The team needs to be assertive, to have good problem-solving skills. In addition, the team needs to have a fair amount of expertise in gathering and extracting information from customers to accomplish project goals.

Designing Phase

The conclusions of designing phase are as follows:

- According to the results, around 50 percent of respondents believe that they talk to seniors for direction if they meet some team members (from requirement phase) who are reluctant to divulge necessary details. Around 30 percent of respondents try to get details from team members of requirement phase assertively. Remaining 10 percent keep pleading for details. These results clearly show that interaction among the teams is less, which results in discrepancies, chaos, time lapse and inefficiency. This can be dealt by having fixed procedure form to transfer information from requirement to coding phase, which will make the work flow smoother.
- Designing and its techniques need to be defined very clearly to increase efficiency.
- The team needs to be given more training and mentoring to accomplish job.

Coding Phase

The conclusions of coding phase are as follows:

- There is a need to be little more impetus to prepare coding process report, which will help to reduce redundancy and increase efficiency of the team/respondents.

- Respondents need training in automation and optimization tools so that they can enhance their efficiency to many folds.

- Respondents require leadership and problem-solving skills to solve their problems on their own instead of relying on other team members and seniors for direction. Relying on others decreases efficiency and results in time lapse, which leads to under utilization of resources.

- According to the results, around 40 percent of respondents need more training and monitoring to handle their jobs in a better way. Moreover, around 50 percent of respondents sometimes feel that they need training and monitoring. However, remaining 10 percent do not think the same. Hence, this phase also needs improvisation in training and mentoring to candidates so that they can better equip themselves with the task.

Testing/Verification Phase

The conclusions of testing/verification phase are as follows:

- Respondents need leadership and problem-solving skills to solve their problems on their own instead of relying on other team members and seniors for direction. Relying on others decreases efficiency and results in time lapse, which leads to under utilization of resources.

- Respondents require more training and mentoring as results show that around 70 percent of respondents feel they need it.

- Respondents must be given training in automation tools so that they can enhance their efficiency to many folds.

- According to results, only 60 percent of respondents think that testing/verification issues are regularly scrutinized and casual analysis is done for testing defects. Therefore, there is a need to improve this.

Comparison of Phases

Comparison of requirement, designing, coding and testing/verification phases is as follows:

So according to the table:

- **Project Clarity**

  Percentage is highest in coding phase as, work is done through set procedures /processes and the team becomes clearer with project requisites, objectives and goals till it reaches the coding phase. On the other hand, project clarity is lowest in requirement phase as the requirement phase is beginning of the project. Hence, this phase needs special focus on this arena. Ambiguity of project leads to increased cost in the later phases. This is because software bugs are carried over to next phases, which results in redundancy, chaos and wastage of time and resources.

> **Note:** As per the industry data, fixing a bug in the requirement phase costs 1 Unit and fixing it in the user site at production level costs 100 Units. Hence, Project clarity at requirement phase is imperative and very significant

- **Work is Done with Structured Plans**

  This factor is virtually same in all phases but marginally different for testing/verification phase. The reason for the difference is that it is the last phase and therefore, team has more confidence to handle the work without structured plans. The difference could also be due to sheer lethargy nature of the team. However, this factor can be checked for variance in performance if encountered at this stage.

- **Report Preparation**

  The percentage is lowest in coding phase. The reason for the difference is that it is the phase of settled process/procedures and hence, team has more confidence to handle the work without structured plans. The difference could also be due to sheer lethargy. However, this factor can be checked for variance in performance if encountered at this stage.

- **Discussion with Other Team Members**

  This factor is almost same in all phases.

- **Monitoring and Training**:

The need for monitoring and training is highest for coding phase. This can be attributed to the fact that coding phase has the highest technical portion, which gets updated with time. Hence, the mentoring and training definitely will enhance efficiency in this phase.

- **Job Satisfaction**

This factor is virtually same in all phases but highest in requirement phase. This is due to the reason that this phase requires interaction with customer and hence, no set process/procedures are followed. Newer challenges make work interesting, which results in the highest job satisfaction. This factor is an interesting observation for reference.

- **Communication Problems with Customer or Team Members**

This factor is highest in requirement phase, as this phase requires interaction with customer. Therefore, the team confronts discrepancies and problems in this phase. On the other hand, this factor is lowest in coding phase because this phase has set procedures. This factor is for reference and can be used to increase efficiency at different phases.

Conclusions:

- At requirement phase the project clarity is minimum, which depicts the arena for more work to enhance efficiency.

- Employees in requirement phase have more problems but more job satisfaction which shows that human interactions enhance the happiness quotient for better efficiencies.

- More automation and optimization procedures reduce the communication troubles within the teams.

Hence, software organizations can use this study in software development projects to enhance human resource efficiencies at various stages.

Limitation of study

The study was conducted on the parameters defined and identified after discussions with the software engineers, academicians, and subject experts.

Scope for further study

Parameters defined above can be individually researched to further outline the efficiency and productivity factors.

References

Citations

1. Measuring Human Resource Efficiency and Effectiveness
Human Resource Professionals Are Best Positioned to Significantly Impact an Organizational Effectiveness and Efficiency as They Interact with Everyone in the Company

2. Enhancing Human Resource Productivity Using Information and Communication Technologies: Opportunities and Challenges for Tanzania1
By Faustin Kamuzora, Mzumbe University, Tanzania, Pg 7-8

3. http://en.wikipedia.org/wiki/Software_development_process

4. Drucker, P. (1999). *Management challenges for the 21st century*. New York: Harper Business. Pg-158.
5. http://en.wikipedia.org/wiki/Business_analysis
6. http://en.wikipedia.org/wiki/Job_satisfaction

Reference

7. Statistics for Management by Richard I.levin and David S.Rubin.Page no. 360-375.

8. Questionnaire Design and Survey Sampling by Professor Hossein Arsham.Link:University of Baltimore.