International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 3, June 2019, pp. 2006~2017
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i3.pp2006-2017

How software size influence productivity and project duration

Mridul Bhardwaj[1], Ajay Rana[2], Neeraj Kumar Sharma[3]
[1,2]Amity School of Engineering and Technology, Amity University, Noida Up, India
[3]HIMT, Greater NOIDA, UP, India

Article Info

Article history:

Received Jan 14, 2018
Revised Oct 25, 2018
Accepted Nov 20, 2018

Keywords:

Effort
Productivity
Project duration
Software size

ABSTRACT

To make a perfect project plan, the software size of the order from the customer is the most important factor. The biggest challenge for the project manager is to estimate the project end date in the beginning of the project i.e. in project planning phase with realistic accuracy. Apart from other major inputs to estimate the project end date, expected team capability (productivity) and estimated software size are the major inputs that may influence the project end date. Software size is one of the most significant independent metric available in the planning phase and project manager has to estimate the other metrics based on the initial estimated software size. There is no direct relationship available between software size and project duration or software size and team productivity, however, there are industry data published by Quantitative Software Management and ISBSG that shows how these metrics influence each other. In this paper, using the data published by ISBSG and Quantitative Software Management, we try to statistically establish how productivity and project duration are influenced by software size. We have done linear regression analysis by generating the secondary data based on the data published by ISBSG and Quantitative Software Management. Linear regression equation validated with the actual project data and experimental results suggest that that productivity is significantly dependent on software size, however, project duration does not significantly depend on software size but may also be dependent on other metrics like team size, apart from software size.

Corresponding Author:

Mridul Bhardwaj,
Amity School of Engineering and Technology,
Amity University, Noida Up, India.
Email: mridul2707@gmail.com

1. INTRODUCTION

All Software and Information Technology Organizations are Project Based organizations. The Organization is an integration of Project Teams working in tandem. Each Manager whether vertically or horizontally aligned has to work and deliver projects with varied scope and size. Estimation on the project end date at the beginning of the project, not only ensures a smooth execution but also helps the organization on optimum utilization of available resources and an enhanced dollar value for future projects. Software size is one of the most significant metric available in planning phase and project manager has to estimate the other metrics based on the initial estimated software size. The importance of software size is explained in “Estimate Software Functional Size before Requirement phase of Development Life Cycle” [1]. “Imagine you are driving on an important trip to a distant place where you have not been before. You will not feel comfortable to start such a journey without knowing at least the general direction of the destination, e.g. distance, the available routes, road conditions etc. Armed with this information and a good map one can feel more comfortable about taking the trip. These are essential; however your comfort may be seriously compromised during the journey. Road works or diversions can also hit your journey plan, but at least you have a good chance of having an early warning, if you check it before starting the journey. Car breakdowns or wheel punctures can be less predictable, but they can happen. Managing a software project is much harder than planning a trip. The biggest difference is that no matter how hard you try, the specifications are not static and there are challenges in predicting team productivity and project duration."

Each software development project is unique and possess different challenges to control and monitor project execution. The project manager faces major challenge to identify the important metrics to control and monitor the project execution. As per Putnam, Lawrence H., and Ware [2], these metrics are software size, effort, project duration and productivity. These metrics tells project manager about what to deliver (size), how it was delivered in past (productivity) and how long will it take to deliver with current team capability (project duration). It is always a challenge before the project manager to estimate these metrics because there is no direct relationship among these metrics, however, there are industry data published by Quantitative Software Management and ISBSG that shows how these metrics influence each other. In this paper, using the data published by ISBSG and Quantitative Software Management, we try to statistically establish how productivity and project duration is influenced by software size.

We performed the statistical analysis to achieve the following

a. Liner regression analysis for software size (independent variable) and productivity (dependent variable)
b. Liner regression analysis for software size (independent variable) and project duration (dependent variable)
c. Verification of liner regression equations with actual data points

2. PROBLEM STATEMENT

Most of the Software development projects are conceptualized to implement the requirements provided by the business users. Users could be an end user, a group of users or a company. Usually in the age of B2B Commerce the software development projects are essentially taken up by software industry companies and there are many local and external stakeholders to the project. The stakeholders are varied between technical and non-technical people. Generally, the marketing team and customer interfacing teams of the client order software. The scope of a software development project may be by the virtue of a new development of software or enhancement of existing software depending upon the user base and their requirements. The business user wants these requirements (most of these are functional in nature) to be implemented as soon as possible, however, they may or may not be aware about the technical challenges that would require to overcome before it can actually be delivered. The Project Manager, who is accountable for successful completion of project, needs to estimate the project duration and the total efforts. In the planning phase, project manager faces the challenge to estimate the project duration to complete the project. Only metrics available is the functional size that can be calculated based on the requirements provided by the user, however, these requirements are also expected to be changed. To estimate the project duration, Project manager must understand his team capability and composition. Team composition and capability can be measured in terms of productivity i.e. number of function points delivered in 1 month, therefore, the project manager needs to understand how he can estimate productivity and project duration using the software size. Unfortunately, most of the time the project manager does not have any method or metrics that help him in estimating the project duration and productivity with the help of software size. A project manager ponders whether there is any measurable relation which exists between software size and duration or software size and productivity.

3. STUDY OBJECTIVE

During project execution, the project manager has to monitor and control the following metrics to analyze the impact on estimated project duration and the team productivity. They are:

a. Software size (It is expected to increase as the project progresses)
b. Team Size (stable team will improve the overall team productivity)

Software Size is the most critical input for planning and rollout of a project. The software size is measurable and it is considered as an input to form the development team i.e. team size which in turn decides the project duration and the delivery date. A mathematical model will not only help the project manager to establish a correlation between software size and the project duration but also will help project managers to deliver the project on time. Accuracy of the calculation will trim down project losses and will enable organizations to deliver software within agreed timelines. Is there a mathematical correlation between software size and productivity and the project duration? If yes, how it can be calculated and measured accurately? This is problem which the authors address in this particular study. The objective of the study is to perform data analysis to develop in-depth understanding of software size influence on productivity and project duration.

a. Is there any co-relation between productivity and software size?
b. Can we assume higher team productivity for big projects?
c. Is there any co-relation between project duration and software size?

Key Software development metrics like efforts, project duration, project cost etc. are dependent on software size and productivity. Our study is an attempt to find the answer of above mention questions. We aim to analyze data published by industry of various projects and arrive at mathematical solution to establish the influence of software size on the productivity of the project as well as how it impact the project duration. This study will help the software professionals and IT Project Managers worldwide in proactivity planning and mitigate the risk if any. Technical Challenges

Major challenge to achieve the research objective was the availability of actual primary dataset for statistical analysis. We have used secondary data that was generated using the statistics (mean and standard deviation) published by ISBSG. Data was generated using the Mini tab tool and data was assumed as normally distributed.

4. DEFINITION

4.1. Software size

Software Size is the measure of functionality that will be delivered by software and that can be validated by end user of the software. It is a numerical measure of functional and non-functional requirements. Software size is an independent metric and all other metrics can be viewed as function of the software size. Software Size only depends on what to deliver rather on how to deliver. If software size of two projects are same then it does not imply that both the projects are same or they deliver the same requirement but it only indicates that both software are of same size. Software Size is the important metric to estimate other metrics e.g. total effort and project duration. It always remains a challenge to estimate the software size in the beginning of the project because only high level requirements are available during the pre-requirement phase of the project, however, in size-based project estimation, details of software requirement are not important but relative size of the project is sufficient for initial estimate of the metrics. For example we may not be able to estimate the software size of proposed project as detail software requirements are not available, but we can compare the complexity of the proposed project with the already delivered projects. This will help the project manager to establish initial baseline of key metrics based on the past performance of the similar projects. There are many software sizing methods viz. function points, use case points, story points (for agile projects), object based count [3]-[6] etc. but IFPUG function point is the most widely used and acceptable software sizing method.

4.2. Productivity or rate of delivery

Effort required to deliver unit software size is defined as Productivity or Rate of delivery. Productivity is require to estimate the overall project effort to deliver the project. Productivity depends on the various factor like team experience or capability to work in similar technology or domain, the team will be more productive if they have earlier worked on similar technology or domain. It also depends on the business process understanding as it will help to translate the business requirements to technical requirements. If the same team had earlier worked together then it will help in improving the productivity as it will reduce the time required to resolve the collaboration issues. It is challenging to define team productivity and to give productivity a number but it an essential metric that not only need to be estimated in the beginning but also need to be consistently monitored during the project life cycle. The project manager can use the historical data of similar projects to estimation the productivity of the team. As most of the productivity numbers are published in range so project manager should carefully chose the productivity number within that range. Figure 1 [7] explain how productivity can be used to drive the overall project efforts.

If there is uncertainty to establish the team productivity then Iterative development model is more preferable to waterfall model. In waterfall model actual productivity of the team will only be known after the construction phase so project manage will not find any opportunity to re-define the productivity number and hence the other key metrics (effort and time), however, iterative development model provide project manager the opportunity to re-define these key metrics. Mridul Bhardwaj and Ajay Rana [8]-[11] explains how effort, schedule and project duration should be re-define using iterative model as shown in Figure 2.

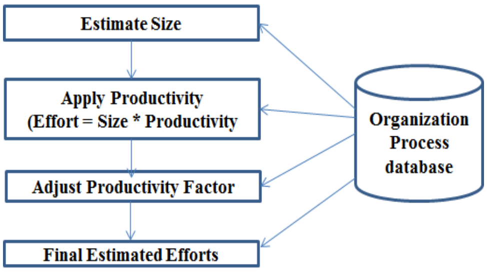

Figure 1. Driving efforts using size and productivity

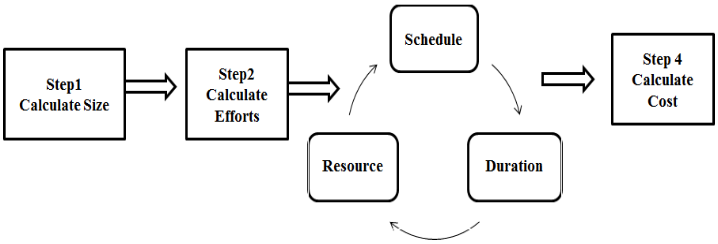

Figure 2. Iterative model to re-define effort and time during project life cycle

4.3. Project duration

Project calendar duration represents time required to complete the project. Project duration is the estimation of project end date for given project start date and estimate the date when we can delivery the project. It must be understood that project duration and total project effort are not interchangeable metric. There is no direct relationship which exists between time and effort. It must be understood that by adding more team members we can only reduce the overall project duration to a certain level but adding more resources may result in longer project duration and increase cost. e.g if total estimated project effort is 12 person months and there is a project team of 4 people. In this case estimated project duration will be 3 months provided there is no planned idle time. We can't deliver this project in one month with 12 member team because adding more member will not only add new communication channels but also increase the integration effort. This relationship must be understood and it is better to have an optimal team size. Figure 3 [7] describe the relationship between Team size and Project duration, increase in team size will help in reducing the overall project duration only till point of reflection but increase in team size beyond the point of reflection will increase the project duration. Project managers must understand the point of reflection as it will help him to commit timelines with project stakeholders. It is not easy to identify the point of reflection but work break down structure or Delphi techniques can help project manager in identifying the optimal team size.

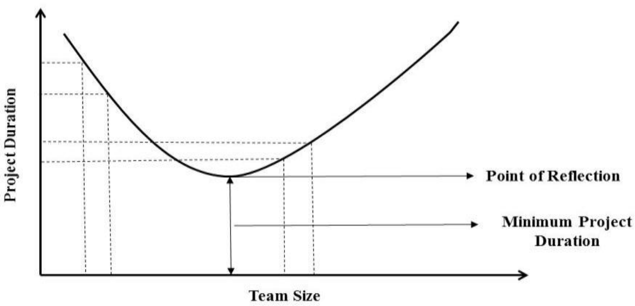

Figure 3. How increase in team size will impact project duration?

5. RELATED WORK

As per Putnam, Lawrence H., and Ware [2], software size, effort, project duration and productivity are the key metrics for monitoring and control of software development project. There is no direct relationship among these metrics, however, there are industry data available that shows how these metrics influence each other.

5.1. Data published by quantitative software management

Quantitative Software Management (QSM) [12] has published the data based on the study of 2,231 closed projects. Projects are classified as new development, Major/Minor enhancement, Conversion or Maintenance project based on the ratio of new functionality to be added versus total functionality (added, modified and deleted). Table 1 shows the median project metrics for various project type.

a. New Development (more than 75% new functionality)
b. Major Enhancement (new functionality between 25%-75%)
c. Minor Enhancement (new functionality between 5%-25%)
d. Conversion (less than 5% functionality) s
e. Maintenance (No new functionality)

Table 1. Median Project Metrics for Various Project Type

Project Type	Median Project Metrics				
	New Development	Major Enhancement	Minor Enhancement	Conversion	Maintenance
% of Projects	16%	61%	14%	7%	2%
Median Size (FP)	291.00	119.00	153.00	109.00	68.00
Median Effort Months	29.70	19.30	28.10	23.40	18.60
Median % Functional Efforts	12%	11%	12%	10%	19%
Median Productivity FP/PM	9.16	5.79	5.19	5.06	2.7
Median Duration	7.57	7.23	6.42	6.43	4.73
Median Defects	37	16	38.5	35	16

Data shows, Rate of delivery (productivity) is higher for new development projects compared to maintenance and enhancement projects. Productivity (rate of delivery) for new development projects is 3 times to the productivity (rate of delivery) of maintenance projects and 50% more than productivity of enhancement projects, however, there are more defects injected in development projects compare to maintenance projects. Data shows Figure 4, Figure 5 and Figure 6 that median number of defects for new development projects are more than double for number of defects for maintenance projects.

Figure 7 shows how productivity (rate of delivery) changes against software size. It shows that productivity improves significantly as project size improve. Data in Table 1 shows that median project size for new development project is 291 function points and median project size for maintenance project is 68 function points. New development projects are bigger in size and rate of delivery is significantly higher than maintenance projects.

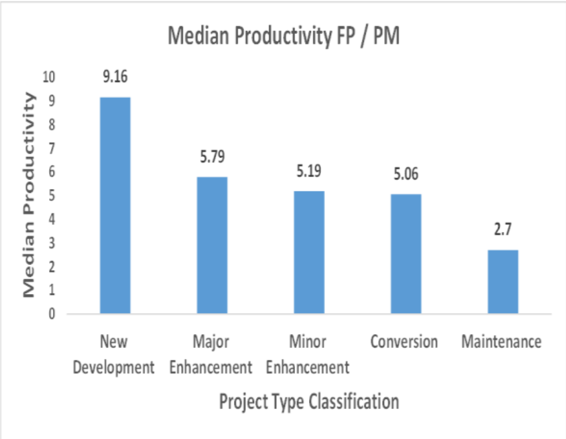

Figure 4. Median productivity for various project type

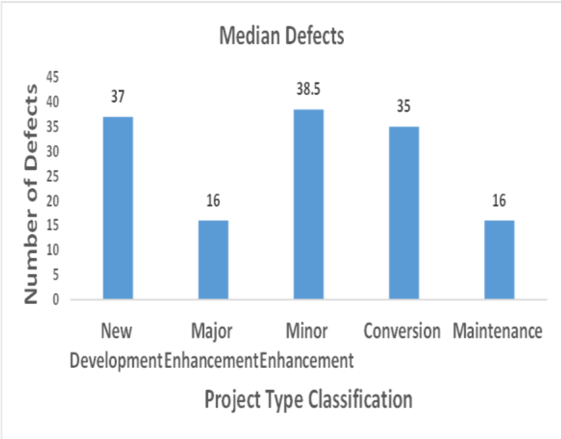

Figure 5. Median defects for various project type

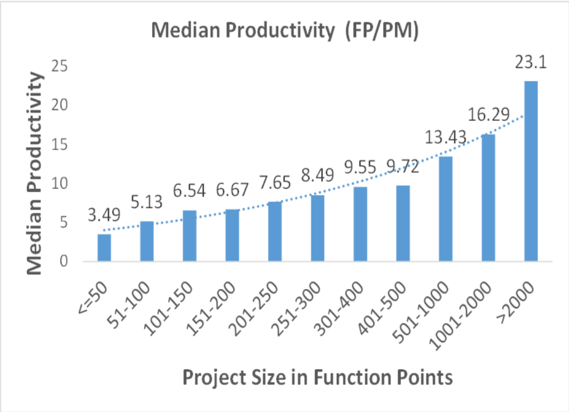

Figure 6. Median productivity against project size

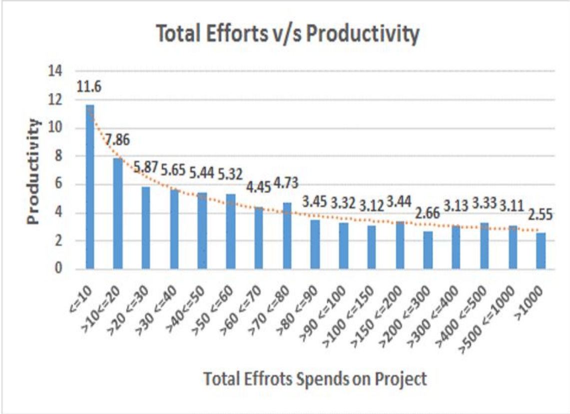

Figure 7. Total efforts and productivity

Figure 8 shows that rate of delivery decreases sharply as we spend more efforts. It shows that if we increase the team size it may impact negatively on overall team productivity. When we distribute the overall efforts to the various project phases, it is important to know how effort distribution to various phases would impact the key metrics like schedule, productivity and defects etc. Study of Quantitative Software Management (QSM) [12] shows that overall productivity improves to 28% if more than 20% efforts spends during analysis and design. Table 2 shows how productivity, schedule and defects improves if we spend more than 20% efforts during analysis and design phases. Overall duration of the project can be reduced by 14% and also overall project effort can be reduced by 10% if more than 20% efforts spend during the analysis and design phase.

Table 2. Impact of Effort Spend in Analysis and Design on Key Metrics

	Impact of Effort Spend in Analysis and Design		
	Effort Spend >20%	Effort Spend <20%	Impact
Productivity FP/PM	7.93	6.2	28%
Median Duration (Months)	6.2	7.23	-14%
Median Efforts (PM)	20.29	22.59	-10%
Median Defects	19.5	20	-3%

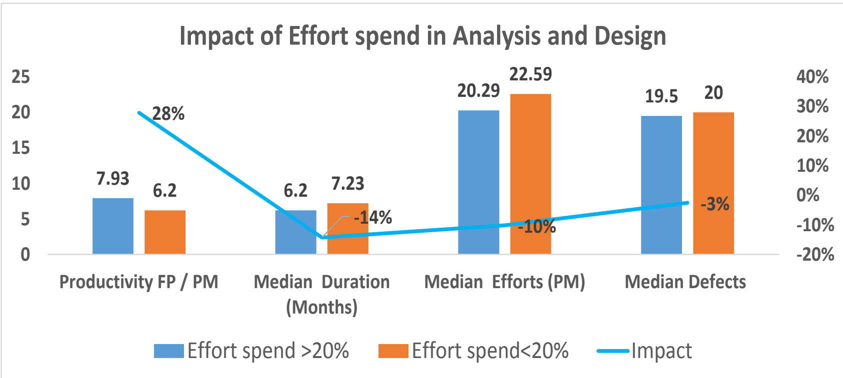

Figure 8. Impact of effort spend in analysis and design

5.2. Data published by international software benchmarking standard group (ISBSG)

International Software Benchmarking Standard Group (ISBSG) in its Benchmarking release published the relationship for software size, efforts, project duration, productivity and maximum team size. These relationship is based on 4,106 software projects from around the world, and made available on Release 10 of Estimating, Benchmarking & Research Suite CD [13]. This relationship have been derived after the statistical analysis of projects in the ISBSG repository. ISBSG study [14] showed that software size and maximum team size are the key metrics for estimating project duration and efforts.

5.2.1. Productivity as function of software size and maximum team size

Dependent Variable=Productivity (Project Delivery Rate in Hours per Function Point)
Independent Variable 1=Size (Software Size in Function Points)
Independent Variable 2=MaxTeamSize (Full Time Equivalent of Maximum Team Size)

$$\text{Productivity} = 37.48 * \text{Size}^{(-0.496)} * \text{MaxTeamSize}^{(0.759)}$$

Above equation is derived based on 203 new development projects in ISBSG project repository and having R2(adj) value as 0.32 which means 32% variability is explained by the above equation. Median MRE value is 0.37 which implies that there could be 37% variance in actual value and value obtain using the derived equation.

5.2.2. Total project efforts as function of software size and maximum team size

Dependent Variable=Efforts (Project Efforts in Person Hours)
Independent Variable 1=Size (Software Size in Function Points)
Independent Variable 2=Maximum Team Size (Full Time Equivalent of Maximum Team Size)

$$\text{Efforts} = 37.48 * \text{Size}^{(.504)} * \text{Maximum Team Size}^{(0.759)}$$

Above equation is derived based on 203 new development projects in ISBSG project repository and having R2(adj) value as 0.61 which means 61% variability is explained by the above equation. Median MRE value is 0.37 which implies that there could be 37% variance in actual value and value obtain using the derived equation.

5.2.3. Total project efforts as function of software size

Dependent Variable=Efforts (Project Efforts in Person Hours)
Independent Variable 1=Size (Software Size in Function Points)

$$\text{Efforts} = 23.25 * \text{Size}^{(.814)}$$

Above equation is derived, based on 534 new development projects in ISBSG project repository and having R2(adj) value as 0.45 which means 45% variability is explained by the above equation. Median MRE value is 0.55 which implies that there could be 55% variance in actual value and value obtain using the derived equation.

5.2.4. Project duration as function of software size

Dependent Variable=Project Duration (Total active elapsed time in months)
Independent Variable 1=Size (Software Size in Function Points)

$$\text{Project duration} = 0.543 * \text{Size}^{(.408)}$$

Above equation is derived, based on 494 new development projects in ISBSG project repository and having R2(adj) value as 0.30 which means 30% variability is explained by the above equation. Median MRE value is 0.39 which implies that there could be 39% variance in actual value and value obtain using the derived equation.

5.2.5. Project duration as function of effort

Dependent Variable=Project Duration (Total active elapsed time in months)
Independent Variable 1=Effort (Total Planned effort in person hours)

$$\text{Project duration}=0.370 * \text{Effort}^{(.328)}$$

Above equation is derived, based on 1681 new development projects in ISBSG project repository and having R2(adj) value as 0.35 which means 35% variability is explained by the above equation. Median MRE value is 0.36 which implies that there could be 36% variance in actual value and value obtain using the derived equation.

6. APPROACH

ISBSG data does not represent any meaningful regression equation between productivity and software size which is one of the most significant metric known in the beginning of the project, however, it does represent the regression equation of productivity with software size and maximum team size (explain in 5.2.1) but most of the time maximum team size is not known in the beginning of the project. Following sections explain the step by step approach taken to establish the regression equation.

6.1. Steps to obtain the linear regression equation

Using the model published by ISBSG and discussed in section 6.2, we simulate the productivity and project duration for different software size in the range 50 to 2000 function points, assuming software size it is normally distributed, we followed the following steps to establish the linear regression equation for productivity and project duration with software size. Table in annexure shows the complete dataset.

a. For all projects, efforts were derived using the regression equation explain in section 6.2.3. Figure 9 shows the graph for the size and corresponding efforts.

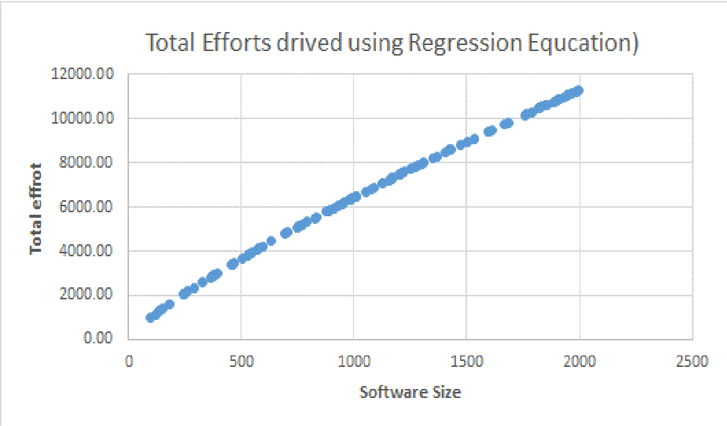

Figure 9. Total effort v/s size

b. Productivity for each project was calculate using the effort and size calculated in step 1 (by dividing software size and effort). Figure 10 shows the graph for software size and productivity.

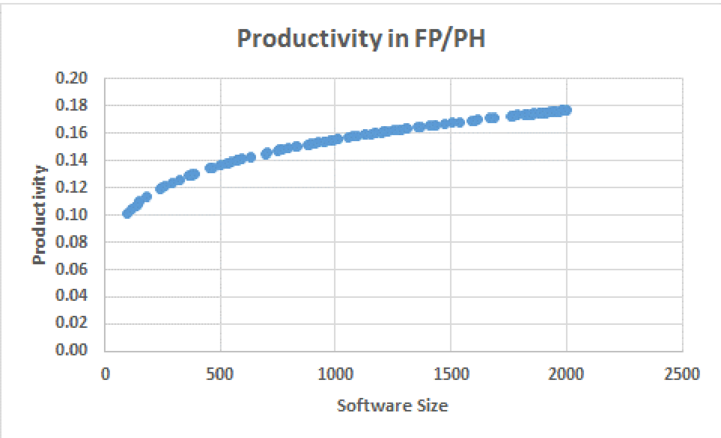

Figure 10. Productivity v/s size

c. Project duration in person month was estimated using the regression equation describe in 6.2.5. Figure 11 shows the graph for project duration and software size.

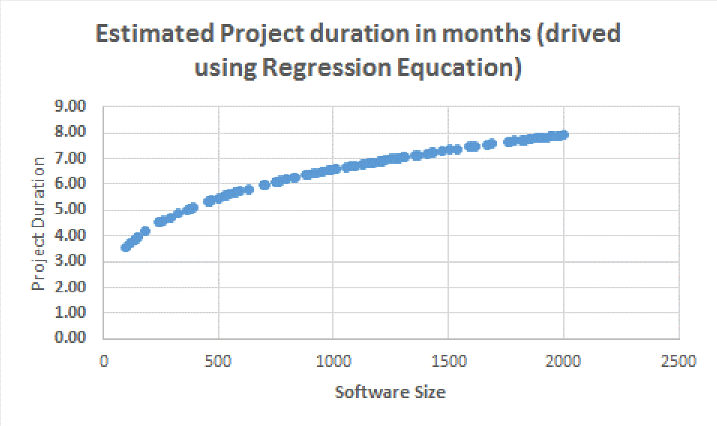

Figure 11. Project duration v/s size

d. Using the statistical tools, obtain the linear regression equations for productivity and project duration depending on software size. In both the equations software size is taken as independent variables, while productivity and project duration is taken as dependent variables
e. Verify the linear equations obtain in step 5 with actual data for 10 closed projects.

7. EXPERIMENT RESULTS

Using the statistical software Minitab version 16, following linear regression equation for productivity and size is obtained. R2(Adj) for this equation is 93% that means linear equation explain 92% of variability in dataset. Figure 12. to Figure 15. shows the output of regression analysis of size versus productivity and size versus project time line. Following linear regression equation for project duration and size is also obtained. R2(Adj) for this equation is 92% that means linear equation explain 92% of variability in dataset.

Productivity = 4.378+0.001923***Size**

Regression Analysis: Productivity versus Size

```
The regression equation is
Productivity = 0.1180 + 0.000032 Size


S = 0.00548439   R-Sq = 92.1%   R-Sq(adj) = 92.1%


Analysis of Variance

Source      DF         SS         MS        F      P
Regression   1  0.0345448  0.0345448  1148.49  0.000
Error       98  0.0029477  0.0000301
Total       99  0.0374925
```

Figure 12. Regression analysis for productivity and size

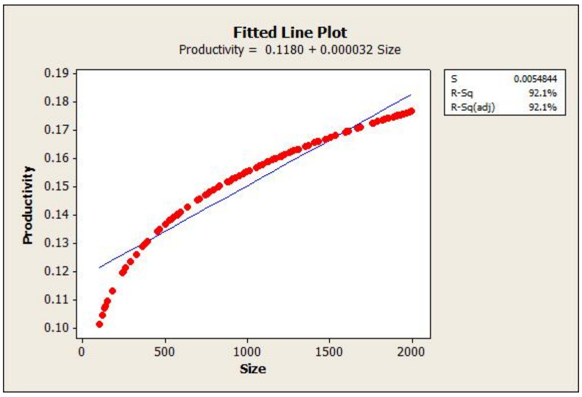

Figure 13. Fitted line plot for regression analysis for productivity and size

Project duration = 4.378+0.001923*Size

```
Regression Analysis: Duration versus Size

The regression equation is
Duration = 4.378 + 0.001923 Size


S = 0.286852   R-Sq = 93.8%   R-Sq(adj) = 93.7%


Analysis of Variance

Source      DF       SS       MS        F      P
Regression   1  121.638  121.638  1478.26  0.000
Error       98    8.064    0.082
Total       99  129.701
```

Figure 14. Regression analysis for project duration and size

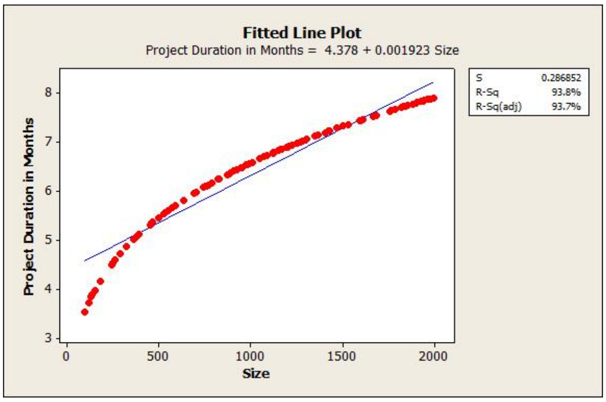

Figure 15. Fitted line plot of regression analysis for project duration and size

8. VERIFICATION OF EXPERIMENT RESULTS

Linear regression equation for productivity and project duration was verified for actual project metrics of 10 closed projects. Actual project size of these projects varies from 175 function points to 1468 function points. For all these projects, actual size, total effort and project duration was collected. We calculated the actual productivity using total effort and size.

Productivity and project duration was also calculated based on the regression equation obtained in section 7. Variation in the actual metric and calculated metric is calculated. Variation in productivity (Actual v/s Calculated) was between -2.6% to 20.4%, except one data point. All the projects is having a positive variation which means the actual productivity was more than the productivity calculated using the regression equation. We can conclude that productivity is significantly dependent on software size. Productivity and software size linear regression relationship provides productivity estimates with 20% variation thus regression equation can be a good estimate of productivity in the initial phases of the project.

Table 3. Verification Results

			Actual Project Data of Closed Projects		As Per Regression Equation		% Variation	
S.N	Actual Size	Total Efforts	Productivity (FP/PH)	Project Duration in Months	Productivity (FP/PH)	Project Duration In Months	In Productivity	Project Duration in Months
1	175	1400	0.125	4.465	0.1236	4.714525	1.1%	-5.3%
2	228	1583	0.144	4.974	0.125296	4.816444	14.9%	3.3%
3	258	1985	0.130	5.232	0.126256	4.874134	3.0%	7.3%
4	325	2600	0.125	5.749	0.1284	5.002975	-2.6%	14.9%
5	339	2568	0.132	5.848	0.128848	5.029897	2.4%	16.3%
6	467	3199	0.146	6.665	0.132944	5.276041	9.8%	26.3%
7	567	3611	0.157	7.214	0.136144	5.468341	15.3%	31.9%
8	775	4506	0.172	8.195	0.1428	5.868325	20.4%	39.6%
9	1245	6803	0.183	9.944	0.15784	6.772135	15.9%	46.8%
10	1468	7567	0.194	10.635	0.164976	7.200964	17.6%	47.7%

Variation of project duration (Actual v/s Calculated) is increasing with project size and this increment is not linear, thus we can conclude that linear equation does not explain the relationship for project duration and software size. We can conclude that project duration does not significantly depend on software size only but may also depend on another metric apart from software size. This other metric could be the team size, this leaves scope to further analysis the project duration with software size and team size as depended variables.

9. RESULT AND DISSCUSION

Linear regression analysis of data published by ISBSG [13] provides linear regression equation for productivity and project duration with software size. Verification of linear regression with actual data of 10 projects conclude that productivity is significantly dependent on software size. Productivity and software size linear regression relationship provides productivity estimates with 20% variation thus regression equation can be the good estimate of productivity in the initial phases of the project.

We also conclude that project duration does not significantly depend on software size but may also depend on other metric apart from software size. This other metric could be the team size, this leaves a scope to further analyze the project duration with software size and team size as depended variables. It can be inferred that software size is the key metrics to manage software development project effectively. QSM study suggests that rate of delivery (productivity) is higher for new development projects compared to maintenance and enhancement projects. Productivity for new development projects are 3 times to the rate of delivery for maintenance projects and 50% more than the productivity for enhancement projects, however, there are more defects injected in development projects compared to maintenance projects. QSM data also suggest that productivity improves significantly as the project size increases, however, it decreases with the increase in efforts, and hence, the increasing team size may have negative impact on productivity, cost and increase the risk of delivering the high cost but low quality projects.

Our study also concluded that distribution of efforts over various project phases also impacts the productivity and schedule. If we spend more than 20% effort during analysis and design phase then we can improve productivity and project duration, by 28%, 14% respectively.

Hence, if a Project Manager applies estimation and techniques prescribed in this study to determine the productivity of a project, the Project Manager should take the software size as the most important input factor. Also, factors like the team size are subjective factors, it may or may not have a direct correlation on the productivity of any software project. Team size depends upon the number of team resources and the productivity of the team resources is a subjective matter which cannot be implied empirically or mathematically.

10. FUTURE SCOPE

The Current study concludes that software size has a significant impact on team productivity, however, the study is unable to establish any significant impact of software size on project duration on measurable terms. Project duration has a positive correlation with software size, however, we need to study further to analyze

a. How project duration influence by software size and team size.
b. As software size is a key metrics so we need to establish how we can effectively estimate the software size in the early stage of project life cycle when only high level requirements are available.

The results of the future scope of study will benefit Software Organizations to plan and execute projects on a mathematical grid accuracy which in turn could become a standardized operating procedure as further studies may establish mathematical relationship between software size and project duration.

REFERENCES

[1] Mridul Bhardwaj and Ajay Rana, “Estimate Software Functional Size before Requirement phase of Development Life Cycle,” *International Journal of Innovations & Advancement in Computer Science*, vol. 3 Issue 4 June-2014, pp 79-83, 2014.

[2] Putnam, Lawrence H., and Ware Myers, “Five Core Metrics-The Intelligence behind Successful Software Management,” New York: Dorset House Publishing Company, Inc., 2002.

[3] S. M. Satapathy, B. P. Acharya and S. K. Rath, "Early Stage Software Effort Estimation Using Random Forest Technique Based on Use Case Points," in *IET Software*, vol. 10, no. 1, pp. 10-17, 2 2016. doi: 10.1049/iet-sen.2014.0122

[4] M. Rizwan and J. Qureshi, "Agile Software Development Methodology for Medium and Large Projects," in *IET Software*, vol. 6, no. 4, pp. 358-363, August 2012. doi: 10.1049/iet-sen.2011.0110

[5] J. Popovic, D. Bojic and N. Korolija, "Analysis of Task Effort Estimation Accuracy Based on Use Case Point Size," in *IET Software*, vol. 9, no. 6, pp. 166-173, 12 2015. doi: 10.1049/iet-sen.2014.0254

[6] E. Khatibi and V. Khatibi Bardsiri, "Model to Estimate the Software Development Effort Based on In-Depth Analysis of Project Attributes," in *IET Software*, vol. 9, no. 4, pp. 109-118, 8 2015. doi: 10.1049/iet-sen.2014.0169

[7] Mridul Bhardwaj and Ajay Rana, “Key Software Metrics and its Impact on each other for Software Development Projects,” *International Journal of Electrical and Computer Engineering (IJECE)* Vol. 6, No. 1, February 2016, pp. 242~248, 2015.

[8] Mridul Bhardwaj and Ajay Rana, “Impact of Size and Productivity on Testing and Rework Efforts for Web-based Development Projects,” *ACM SIGSOFT Software Engineering Notes*, vol. 40 Number 2, March-2015, pp 1-4, 2015.

[9] Mridul Bhardwaj and Ajay Rana, “Estimation of Testing and Rework Efforts for non-Web-based Software Development Projects,” *IEEE international conference* on “Futuristic Trends in Computational Analysis and Knowledge Management” at *Greater NOIDA*, 25-27, February 2015.

[10] Bhardwaj, Mridul; Rana, Ajay. "Estimation of Testing and Rework Efforts for Software Development Projects," *Asian Journal of Computer Science and Information Technology*, [S.l.], v. 5, n. 5, p. 33-37, May 2015. ISSN 2249-5126.
[11] Bhardwaj, Mridul; Rana, Ajay. "Impact of Size and Productivity on Testing and Rework Efforts for Non-Web-based Development Projects," *International Journal of Advanced Technology in Engineering and Science*, vol 3, spl issue 2, March-15
[12] "QSM Software Almanac, Application Development Series," Research Edition, 2014.
[13] ISBSG "The Benchmark data for Software estimation," Release 10, 2011.
[14] Book, "Practical Software Project Estimation," by *International Software Benchmarking Standards Group*, Page no 246-248