❏   242

# Key Software Metrics and its Impact on each other for Software Development Projects

**Mridul Bhardwaj, Ajay Rana**
Amity School of Engineering and Technology, Amity University, Noida Up, India

| Article Info | ABSTRACT |
|---|---|
| | Every software development project is unique and different from repeatable manufacturing process. Each software project share different challenges related to technology, people and timelines. If every project is unique, how project manager can estimate project in a consistent way by applying his past experience. One of the major challenges faced by the project manager is to identify the key software metrics to control and monitor the project execution. Each software development project may be unique but share some common metric that can be used to control and monitor the project execution. 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 (time and effort). In this paper, we explain the relationship among these key metrics and how they statistically impact each other. These relationships have been derived based on the data published in book "Practical Software Estimation" by International Software Benchmarking Group. This paper also explains how these metrics can be used in predicting the total number of defects. Study suggests that out of the four key software metrics software size significantly impact the other three metrics (project effort, duration and productivity). Productivity does not significantly depend on the software size but it represents the nonlinear relationship with software size and maximum team size, hence, it is recommended not to have a very big team size as it might impact the overall productivity. Total project duration only depends on the software size and it does not depend on the maximum team size. It implies that we cannot reduce project duration by increasing the team size. This fact is contrary to the perception that we can reduce the project duration by increasing the project team size. We can conclude that software size is the important metrics and a significant effort must be put during project initiation phases to estimate the project size. As software size will help in estimating the project duration and project efforts so error in estimating the software size will have significant impact on the accuracy of project duration and effort. All these key metrics must be re-calibrated during the project development life cycle. |
| | |

*Corresponding Author:*

Mridul Bhardwaj,
Amity School of Engineering and Technology
Amity University,NOIDA UP, INDIA
Email: mridul2707@gmail.com

## 1. INTRODUCTION

Every software development project is unique and different from repeatable manufacturing process. Each software project share different challenges related to technology, people and timelines. If every project is unique, how project manager can estimate project in a consistent way by applying his past experience. One of the major challenges faced by the project manager is to identify the key software metrics to control and

monitor the project execution. Each software development project may be unique but share some common metric that can be used to control and monitor the project execution. As per Putnam, Lawrence H., and Ware [1], these metrics are software size, effort, project duration and productivity. These metrics tellsproject manager about what to deliver (size), how it wasdelivered in past (productivity) and how long will it take todeliver with current team capability (time and effort).

## 1.1. Software Size –Measurement of Functionality

Software size is the measurement of software functionality that is being delivered or expected to be delivered by software. Software size is a numerical measure of software requirement that are defined qualitatively (in most case in word document) by user. Software size is most natural metric of software as it is independent from all other metrics. Software size depended only on what to deliver rather on how to deliver. As software size is a numerical measure of software functional requirements so project manager must remember that no two software project will be same in functionality but they may have same software size. In size-based project estimation, details of software requirement are not important. What important is the relative size of project in comparison to the already completed projects. 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 predict the project performance based on past performance of the similar project. There are many software sizing method viz. function points, use case points, story points (for agile projects), object based count etc. IFPUG function point is the most widely used software sizing method. For function points, you can find industrial data on software project metric published by International Software Benchmarking Standard Group (ISBSG) [2], Quantitative Software Management (QSM) [3] but still there is a challenge to measure the software size in the early stage of project life cycle as detail requirements are not available.Mridul Bhardwaj and Ajay Rana [4]  in his paper "Estimate Software Functional Size before Requirement phase of Development Life Cycle" suggested the method to estimate software functional size when details requirements are not available.

## 1.2. Effort and Time

Project effort is the cumulative time spends by the entire project team on the project. Effort generally measure in person hours, person months or person days but person hours is most suitable and unambiguous unit as other units require conversion from hour to day or month. This conversion is not standard because in some country (specially developed countries) there is 8 hours working in a day while in some country (developing country) it is more than 8 hours.

Time represents the calendar duration of the project. It tells the expected project start and end dates. Though it appears that Time and efforts are interchangeable metric but actually it is not. There is no linear relationship between time and effort. Project manager must understand that we can reduce the overall project duration by adding more team members but beyond certain point increase in team size will not result in reducing the project duration but it will increase project duration. To understand it in more detail, if total estimated project effort is 12 person months and there is project team of 4 people. In this case estimated project duration will be 3 months provided there is no planned idle time. Can we deliver this project in one month with 12 member team? Obvious logical answer is "No" because adding more member will add new communication channels and also increase the integration effort. Project manager must understand this relationship and should choose optimal team size and should prefer constant team size throughout the project life cycle. Figure 1 explain the relationship between team size and project duration till point of reflection increase in team size will help in reducing the overall project duration but increase in team size beyond the point of reflection will increase the project duration. Project manager must understand this 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. Since cost of people is the major cost in any software development that is why in most of the Agile software development projects the team size remain constant. Project manager must understand these relationships and based on the project objective (as defined in project charter#) he should make correct balance of these metrics (cost, effort and duration).

#Project charter defines the project objective approved by project sponsor. In some cases (product development projects where time to market is the key), project end date is sacrosanct while project scope and cost can be changed. In some cases (tax based software application), project scope is sacrosanct while project end date can be changed.
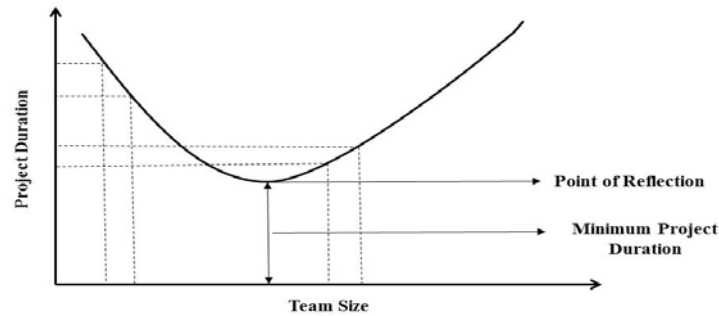
Figure 1. How increase in Team size will impact project duration?

### 1.3. Productivity or Rate of Delivery

Productivity can be defined as effort required to deliver unit software size. It helps the project manager to predict the overall effort require to deliver the project. Productivity depends on the team expertise or experience to work in similar technology, Team will be more productive if they have worked on similar technology earlier. Productivity will also depend on the business process understanding, if team understand the business processes, it will help them to translate the business requirements to technical requirements. If 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. Project manager must understand the critical factor that can influence the team productivity. Following are the critical points that project manager must consider while estimating the team productivity.

- Team experience to work on similar technology
- Understanding of business processes of software application under development.
- Experience of team working together
- Understanding of client environment  as it will help to factor in the client dependency, if any

It is challenging to define team productivity and bigger challenge to give productivity a number but it an essential metrics that not only need to be predicted but also consistently monitored during the project life cycle. In Agile project productivity is termed as "Velocity" and defined as the number of story points delivered per iteration. Velocity depends on team size and number of story points delivered, as most of the agile projects have constant team size throughout the project life cycle and length of the each iteration is also constant so higher the velocity, higher the team productivity. In Agile project velocity or team productivity improves as project progress. Velocity will be high in later iteration of the projects.

Project manager should 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 2 explain how productivity can be used to drive the overall project efforts.
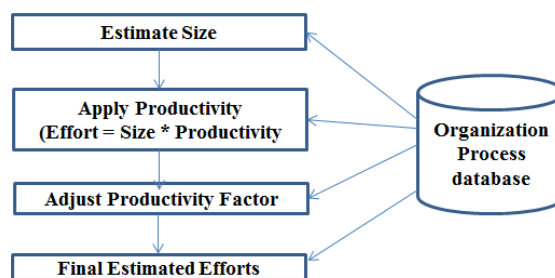


Figure 2. Driving Efforts Using Size and Productivity

Agile or Iterative development model are more preferable to waterfall model if there is higher uncertainty to define the team productivity. In waterfall model actual productivity of the team will only be known after the construction phase so project manage does find any opportunity to redefine the productivity number and hence the other key metrics (effort and time), however, iteration and agile development model

provide project manager the opportunity to re-define these key metrics.Mridul Bhardwaj and Ajay Rana [4] explains the steps to calculate these key metrics.
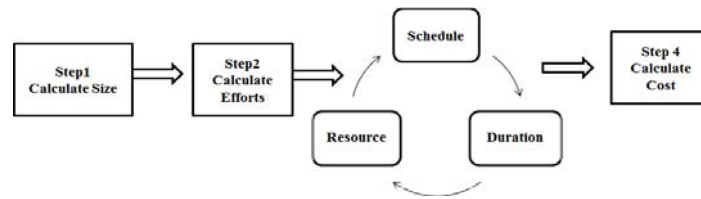


Figure 3.

## 2. RESEARCH METHOD

As we discussed in section 1 size, effort, duration and productivity are the key software metrics that can help project manager in project monitoring and execution but it will helpful only when project manager knows the relationship among these metrics. The relationship among these metrics can be derived using the historical data of similar projects of the organization but it is possible only when organization process maturity level (e.g. SEI CMMi level 4 or above) is high. If organization process level is not high or organization data repository is not sufficient to drive any statistical relationship (organization data repository should contain sufficient data points e.g. sample size > 10 to drive any meaningful statistical relation) then relationship published by various groups ISBSG [2] and QSM [3] can be used. Relationship derived using organization historical data of similar project is always better than the industry published data. Following sub section will explain method to drive relationship using historical data and also some of the industry published relationship. If historical data is not available then then project manager can use the relationship published by various software bench marking organization or by relationship published by researcher, however, these relationship may not be exactly applicable to the project but it can definitely give idea or direction how these metrics are related to each other. We are giving here the reference of our research work that has been has been published

### 2.1. Equation for Total Number of Defects Estimated from Software Size, Efforts and Productivity

In his paper Mridul Bhardwaj and Ajay Rana [5], "Impact of size and productivity on testing and rework efforts for web-based projects", established the following relationship between number of defects, size, productivity and efforts. This relationship was established by using statistical techniques using the industry benchmarking data published by ISBSG [6].

**Number of defects = - 2.84 + 0.000114 * efforts + 0.0290 * size - 0.122 * productivity  (1)**

As concluded byMridul Bhardwaj and Ajay Rana [5] "Co-efficient of size is much higher than the co-efficient of effort that means size has much significant impact on the total number of defects in comparison to effort. Change in software size will have bigger impact on the number of defects identified, therefore, we can say that while planning the software project we should use appropriate software estimation tools and technique to reduce the margin of error in size and we should re-estimate the size after every phase to re-calibrate overall efforts. Low constant number in equation signifies that there is very less environment noise on the total number of defects and it is significantly depends on software size so error in size estimation may led to error in estimation of testing efforts. Results also suggest that in web-based projects the number of defects identified is directly proportional to the productivity i.e. higher productivity will led to more defects found and lower productivity will lead to fewer defects found, therefor, less testing and rework effort will be required if we spend more time on development (i.e. time spend till construction phase). The total number of defects will get reduced and it will directly contribute in reducing the rework efforts."

For the non-web-based projects similar relationship was presented Mridul Bhardwaj and Ajay Rana [7] in IEEE international conference on "International Conference on Futuristic Trends in Computational Analysis and Knowledge Management " held at Greater NOIDA, following relationship between number of defects, size, productivity and efforts was presented. This relationship was also established by using industry benchmarking data published by ISBSG [6].

**Number of defects = - 11.3 - 0.000272 * efforts +  0.0529 * size + 0.538 * productivity (2)**

As concluded by Mridul Bhardwaj and Ajay Rana [7], "higher productivity will lead to the less number of defects so non-web based project should be planned with much experience team (higher productivity) so that we should have less defect and rework effort. It also suggests that size has much significant impact on total number of defect in comparison to efforts. In multi linear regression equation size co-efficient is much higher than the efforts co-efficient that means size has much significant impact on total number of defect in comparison to efforts. Change in unit size will have bigger impact in comparison to unit change in effort. So study conclude that while planning the software project we should use appropriate tools to reduce the margin of error in size estimation and we should re-estimate the size, after every phase of development life cycle, to re-calibrate overall efforts and to minimize the impact on the project plan."

## 2.2. Relationship among Productivity, Effort, Software Size and Project Duration as Per ISBSG

ISBSG in its book "Practical Software Project Estimation" [6] has published equation for software size, efforts, project duration, productivity and maximum team size. These equations have been derived after the statistical analysis of projects in the ISBSG repository. ISBSG study showed that software size and maximum team size are the key metrics for estimating project duration and efforts. Software size and maximum team size are the independent metrics and other metrics productivity, efforts and project duration depends on these two. We have classified the equation in following 2 groups.

**Group 1** - Equations to show how productivity, efforts and project duration depends on software size and maximum team size (section 1.2.1 to section 1.2.3)

**Group 2** - Equations to show how productivity, efforts and project duration depends on software size (section 1.2.1 to section 1.2.3)

### 2.2.1. Equation for Productivity, Estimated from Software Size and Maximum Team Size

As per ISBSG [7] for new development projects, productivity can be estimated using software size and maximum team size as per the following equation.

$$\textbf{Productivity} = 37.48 * \textbf{Size}^{(-0.496)} * \textbf{TeamSize}^{(0.759)}$$

Where: Size  = Software size in function points
        TeamSize = Maximum team size

### 2.2.2. Equation for Efforts, Estimated from Software Size and Maximum Team Size

As per ISBSG [7], for new development projects, total project efforts can be estimated using software size and maximum team size as per the following equation.

$$\textbf{Efforts} = 37.48 * \textbf{Size}^{(.504)} * \textbf{TeamSize}^{(0.759)}$$

Where: Size = Software size in function points
        TeamSize = Maximum team size

### 2.2.3. Equation for Project Duration, Estimated from Software Size and Maximum Team Size

As per ISBSG [7], for new development projects, no suitable equationcan be derived as maximum team size does not make significant impact on project duration.

### 2.2.4. Equation for Productivity, Estimated from Software Size

As per ISBSG [7], for new development projects, no suitable equation can be derived as software size does not make significant impact on the productivity.

### 2.2.5. Equation for Efforts, Estimated from Software Size

As per ISBSG [7], for new development projects, total project efforts can be estimated using software size

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

Where: Size = Software size in function points
        TeamSize = Maximum team size

### 2.2.6. Equation for Project Duration, Estimated from Software Size

As per ISBSG [7], for new development projects, total project duration can be estimated using software size

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

Where: Size = Software size in function points

## 3.    RESULTS AND DISSCUSION

For web based project, if we combine the equation (1) of section 1.1 and equations describe in section 1.2, we get the following defect equation in terms of software size and maximum team size.

**Number of Defects = - 2.84 + TeamSize** (0.759) **(0.0042 \*Size** (.504) **- 4.5725 \* Size** (-0.496) ) + **0.0290 \* Size**

Equations described in section 1.2.1 to 1..26 suggest that out of the four key software metrics software size significantly impact the other three metrics(project effort, duration and productivity). Productivity does not significantly depend on the software size but it represents the nonlinear relationship with software size and maximum team size, hence, It is recommended not to have a very big team size as it might impact the overall productivity. Total project duration only depends on the software size and it does not depend on the maximum team size. It implies that we cannot reduce project duration by increasing the team size. This fact is contrary to the perception that we can reduce the project duration by increasing the project team size. We can conclude that software size is the important metrics and a significant effort must be put during project initiation phases to estimate the project size. As software size will help in estimating the project duration and project efforts so error in estimating the software size will have significant impact on the accuracy of project duration and effort. All these key metrics must be re-calibrated during the project development life cycle

In his book "Five Core Metrics – The Intelligence behind Successful Software Management Project", Putnam, Lawrence H., and Ware [1] describe the five core metrics that can be helpful to manage the software development project execution. We have tried to establish the relationship among these metrics using the benchmarking data published by International Software Benchmarking Service Group, out of the five metrics, we have excluded the "Reliability" metric as it can only be measured once software development is completed and it will not help in project execution. Our study shows that project duration depends on software size and not on team size, our study also establish the fact that productivity does not depend on software size but on team size. Our findings are in line with the finding published by "QSM Software Almanac" [3] in his research edition 2014. Our study shows that Software Size influences all the core metrics and hence industry best practices (e.g. function point) must be used to measure the Software Size. Our finding also in line with our earlier research work published [5] in ACM SIGSOFT March 2015, where we established that to predict the number of defects, Software Size is the most influences metric among the five core metrics.

## 4.    CONCLUSION

We can conclude that software size is the important metricsand a significant effort must be put during project initiationphases to estimate the project size. As software size willhelp in estimating the project duration and project efforts soerror in estimating the software size will have significantimpact on the accuracy of project duration and effort. Allthese key metrics must be re-calibrated during the project development life cycle.

## REFERENCES

[1]    Putnam, Lawrence H., and Ware Myers, "Five Core Metrics—The Intelligence behind Successful Software Management", New York: Dorset House Publishing Company, Inc., 2002.
[2]    ISBSG "The Benchmark data for Software estimation", Release 10 (2011)
[3]    QSM Software Almanac, *Application Development Series*. 2014 Research Edition.
[4]    Mridul Bhardwaj and Ajay Rana, 2014, 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
[5]    Mridul Bhardwaj and Ajay Rana, 2015,  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
[6]    Book "*Practical Software Project Estimation*", by International Software Benchmarking Standards Group, Page no 246-248
[7]    Mridul Bhardwaj and Ajay Rana, 2015, 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.

## BIOGRAPHIES OF AUTHORS



Mridul Bhardwaj is having more than 17 years' experience of IT project and product development in various technologies. Currently, He is senior project manager in one of the leading IT company in INDIA and also doing research in Software quality metrics for software development project. He has published 5 papers in reputed international journal.



Prof (Dr.) Ajay Rana is having a rich experience of Industry and Academia of around 15 years. He is Founder Director /Group Director / Director Professor / Mentor of more than 27 different Institutions and Innovative Programs at Amity Group. He obtained Ph.D. in Computer Science and Engineering, M.Tech (Master of Technology) in Computer Science and Engineering and MBA (Master of Business Administration) He has published more than 177 Research Papers in reputed Journals and Proceedings of International and National Conferences. He has co-authored 05 Books and co-edited 36 Conference Proceedings. He has delivered Invited lectures in more than 36 Technical and Management Workshop / Conferences programs in India and abroad. He is a member of Board of Governance (BOG), Advisory Council (AC), Academic Executive (AE) Member, Board of Studies (BOS) and Special Member of many Indian and Foreign Universities as well as Industries. He is Editor in Chief, Technical Committee Member, Advisory Board Member for 18 Plus Technical Journals and Conferences at National and International Levels.