# Optimization of Effort Variance using Interpolation in Software Development Projects

**Basavaraj M.J**
Perot Systems, EPIP Phase II,
Whitefield Industrial Area, Bangalore-560 066

**Dr. K.C Shet**
Professor, Computer Department, National Institute of Technology Karnataka, Surathkal

**ABSTRACT**
Many software development Projects mainly use Function point software estimation methodology or Lines of Code Estimation methodologies for Efforts Estimation. Effort variance is an important metrics parameter which needs to be more focused. Optimization of effort variance is a very significant factor which influences internal organization driven and customer driven goals.
We have taken one Project data of 4301 Actual Person days and 4008 Person days for analysis purpose. We have used Langrange's Interpolation methodology to predict actual efforts at any point of stage of software development life cycle(SDLC) for the planned efforts for that stage. By adopting this methodology we have minimized the effort variance from 7% to 0.65%. This methodology and validation need to done for many projects to apply in generic way.
***Keywords:***
*Software Estimation, Interpolation*

## 1. Introduction

Software estimation is one of the biggest challenges in IT industries[1]. Optimization of effort variance helps the organization to improve the productivity in various Metrics Parameters like defect density, review effectiveness, testing effectiveness, defect detection efficiency, resolution effectiveness & etc[2].

### 1.2 Scope of this work

This Paper focuses on Optimizing of Effort variance in case of development projects by adopting Interpolation methodology.

### 1.3 How this is useful to Software community?

Software community can be benefited by adopting this proposed approach in their development projects to achieve minimal effort variance. During the course of time by following this proposed approach helps the organization to come out with baseline values for many metrics parameters which were mentioned above.
In the intention of maintaining security and confidentiality of data, authors are constrained not to disclose the company or client name or project name or exact named data in his research.

Manuscript received February 5, 2008
Manuscript revised February 20, 2008

## 2. Methodology of the Work

Fig 1 represents the efforts data for development Project "A". Planned efforts and Actual efforts for each phase has been listed.

| Project A | | |
|---|---|---|
| Tasks | Planned Efforts | Actual Efforts |
| | % of total | % of total |
| Analysis Phase | 7.29% | 7.21% |
| Design Phase | 18.15% | 16.91% |
| Construction Phase | 8.78% | 9.30% |
| Testing | 37.60% | 39.53% |
| Project Planning | 3.93% | 3.66% |
| Project Tracking | 4.93% | 4.65% |
| Software Quality Assurance | 1.98% | 1.74% |
| Configuration Management | 1.98% | 1.86% |
| Project Documentation | 8.42% | 7.21% |
| Reviews | 3.95% | 4.65% |
| Training | 2.00% | 2.30% |
| Inter-group coordination | 1.00% | 0.98% |

**Fig 1: Project "A" – Efforts Details**

Fig 2 represents the metrics data details for Project "A" by following the conventional Function Point and Complexity scenarios assigned weight based estimation approach.

| | Metrics | Planned | Actual | Variance |
|---|---|---|---|---|
| Project A | Effort(PD) | 4008 | 4301 | 7% |
| | Productivity | 0.41 FP/PD | 0.38 FP/PD | |
| | Estimated Size | 1637 Programs | | |

**Fig 2: Project "A "– Metrics details**

### 2.1 Project "A" – X & Y

Fig 3 represents the efforts data details for Project "A". We are considering "X" for planned efforts and "Y" for Actual efforts.

| Project A - Efforts in % | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Planned Efforts | 7.29 | 18.15 | 8.78 | 37.6 | 3.93 | 4.93 | 1.98 | 1.98 | 8.42 | 3.95 | 2.00 | 1.00 |
| Actual Efforts | 7.21 | 16.91 | 9.3 | 39.53 | 3.66 | 4.65 | 1.74 | 1.86 | 7.21 | 4.65 | 2.30 | 0.98 |

**Fig 3: Project "A" – X & Y**

### 2.2 Lagrange's Interpolation

The Lagrange interpolating[3][4][5] polynomial is the polynomial $P(x)$ of degree $\leq (n-1)$ that passes through

the $n$ points $(x_1, y_1 = f(x_1))$ , $(x_2, y_2 = f(x_2))$ , ..., $(x_n, y_n = f(x_n))$, and is given by

$$P(x) = \sum_{j=1}^{n} P_j(x),$$

$$P_j(x) = y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

This can be written as …….

$$\frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)} y_1 + \frac{(x-x_1)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)} y_2 + \cdots + \frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})} y_n.$$

Example for n = 3,

$$\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} y_3$$

We can predict Actual efforts at any point of value of X(Planned Efforts). In this X represents Planned Efforts. If we want to predict actual efforts when planned efforts is 10%, by applying Langrange's Interpolation above formulae we got 10.83107%.

## 3. Improvements

We have applied the Langrange's Interpolation methodology while executing project "B" starting from start phase of SDLC to end phase of SDLC. Fig 4 represents the efforts data for development Project "B". Planned efforts and Actual efforts for each phase has been listed.

| Project B | | |
|---|---|---|
| Tasks | Planned Efforts | Actual Efforts |
| | % of total | % of total |
| Analysis Phase | 2.80% | 2.58% |
| Design Phase | 9.00% | 6.59% |
| Construction Phase | 38.50% | 43.03% |
| Testing(ST,UAT & Warranty) | 26.60% | 26.78% |
| Project Planning | 3.30% | 3.18% |
| Project Tracking | 3.10% | 3.51% |
| Software Quality Assurance | 1.00% | 1.43% |
| Configuration Management | 3.00% | 2.52% |
| Project Documentation | 3.00% | 2.04% |
| Reviews | 7.70% | 7.90% |
| Training | 1.00% | 0.44% |
| Inter-group coordination | 1.00% | 0.00% |

**Fig 4: Project "B"– Efforts Details**

Fig 5 represents the efforts data details for Project "B" . We are considering "X" for planned efforts and "Y" for Actual efforts. We have achieved the effort variance as 0.65% which is very much optimized compared to 7% effort variance earlier got for Project "A".

| | | | | |
|---|---|---|---|---|
| Project B | Effort(PD) | 917 | 911 | 0.65% |
| | Productivity | 0.52 FP/PD | 0.52 FP/PD | |
| | Estimated Size | 479 FP | | |

**Fig 5: Project "B" – Metrics details**

## 4. Further Scope of Research and limitation

This methodology need be exercised on many development projects by focusing on various metrics parameters which influences effort variance to implement in generic way.
Currently authors are not done validation by exercising this methodology on various projects which is a limitation.

## 5. Conclusion

There is a significant reduction in the effort variance to the extent of 6 percentages by adopting this proposed approach for development projects. This will help in saving of time. Besides, efforts can be diverted to other to other project activities or to new requests. During the course of time by following this proposed approach helps the organization to come out with baseline values for each phase wise level. Finally, this exercise rejuvenates in revisiting the earlier agreed SLAs(Service Level Agreements) and to negotiate with the customer for getting new business.

## References

[1] Capers Jones, "Estimating Software Costs", Tata McGraw Hill, Edition 2005
[2] Basavaraj M.J and Dr. K.C. Shet, "Predicting Efforts Model for Level-1 and Level-2 Support functions of Application Service Maintenance Projects using historical data" – 2nd Annual International Software Estimation Colloquium (SEC 2007), August 3, 2007- Bangalore, India.
[3] Douglas C. Montgomery, Eligabeth A. Peck, G. Geoffery Vinning," Introductoion to Liner Regression Analysis"- Third edition,2006
[4] S.C. Gupta, " Fundamentals of Statistics", Sixth Revised and Enlarged Edition,2006
[5] http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html

**Basavaraj M.J**, Project Manager in Perot Systems, Bangalore, has an industry experience of more than 14 Years in project management, software estimation and software quality. He has done his B.E in Computer Science and Engineering from Bapuji Institute of Engineering & Technology, Karnataka and M.Tech in Computer Science and Engineering from Karnataka Regional Engineering college, Karnataka. Currently undergoing his Part-time PhD in the area of Software Estimation from National Institute of Technology, Karnataka, he is associated with MISTE and is a certified Software Test Manager and certified Software Quality Analyst. He has published 5 Papers.

**Dr. K.C Shet**, Professor in Computer Engineering Department, National Institute of Technology, Karnataka has more than 35 years of experience in Teaching, Research and Development. He has done his B.E in Electronics & Communication from University of Mysore, M.SC Engineering from University of Sambalpur and his PhD from IIT Bombay.He is as Member of Computer Society of India, Mumbai and Indian Society of Technical Education, New Delhi. His Research work specialized areas are software testing, Security Solution for Web Services, Cyber Laws, Anti spam solutions, Wireless Networks, Mobile Computing, Ad hoc Networks. He has published more than 185 papers.