EasyChair Preprint

№ 10736

Strategies to Overcome Slow Growth in Software Firms: Project Diversity Perspective

Sayeda Rahnuma Akthar, Farzana Sadia and Mahady Hasan

August 18, 2023

Strategies to Overcome Slow Growth in Software Firms: Project Diversity Perspective

Sayeda Rahnuma Akthar[1][0009-0009-6049--8182] and Farzana Sadia [0009-0002-0118-9039] Mahady Hasan[3][0009-0002-9037--0181]

[1]*Department of Computer Science, Independent University Bangladesh, Dhaka, Bangladesh*
[2] *Faculty of Electronic Engineering & Technology, Universiti Malaysia Perlis (UniMAP), Perlis, Malaysia*
[3]*Department of Computer Science, Independent University Bangladesh, Dhaka, Bangladesh*

Abstract. The Software engineering focuses mostly on knowledge and technology domain. However, the success of a soft- ware firm depends on many other aspects beyond the realm of software engineering knowledge. It is a concerning observation of ours that software firms in Bangladesh are not showing yet their expected growth. This research identifies high level project diversity is one of the main reasons to cost software firms to face difficut ties to reach their expected growth. The firms tend to choose projects from multiple market segments which disables the firms to take benefit from the economics of scale and scope. Moreover, this business trend creates obstacle in increase customers' willingness to pay. In this paper, we have focused on the tendency of slow growth of the software firms due to high diversification of project selection and proposed a possible decision making framework to contribute in the area of software engineering knowledge.

Keywords: Economies of scale and scope, Project diversification, Decision making algorithm, Features ex traction, Software company growth factors

1 Introduction

In-house developers, outside contractors, and inventors have all contributed to the development of software in Bangladesh. Despite expansion, the IT sector is still far smaller than that of India. 90% of businesses have fewer than 30 employees. Software and IT services account for 63% of local market share, but growth here is not as strong as it is in India. Bangladesh relies on international consulting firms since it lacks large IT juggernauts like Infosys or TCS. Growth in revenue and employment is hampered by this. Businesses try to enhance their delivery times, quality, and costs, which results in a variety of initiatives but prevents systematic reuse. Lack of funding prevents product improvement. Innovation could lengthen product life cycles to address these problems, enabling targeted market segments and reuse advantages.

In order to provide software companies with a project selection framework, this

article connects the traits of both new and current projects. By lowering the high project diversity, this will assist businesses in choosing new initiatives inside their industry.

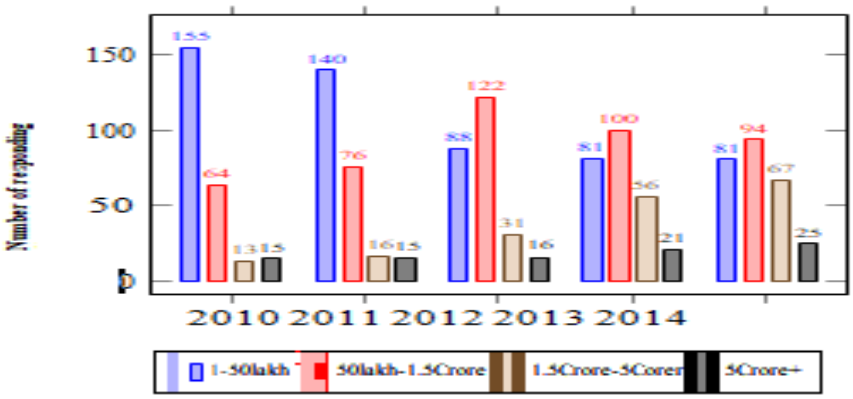

Figure 1: Revenue distribution

2 Problem Statement

The predicted success of Bangladeshi software enterprises is shown by the revenue distribution (Figure 1) and firm size growth trends (Figure 2), which are based on a survey covering the years 2009 to 2014 . The year is indicated by the horizontal line, while the vertical lines (in log scale) represent the respondent firms. Each line in Figure 2 denotes the size of a company. Consistency in business sizes has been observed, which suggests that employee count growth is static. Despite BASIS's assertion that 100 new registrations occur per year, the survey only found 309 active businesses.

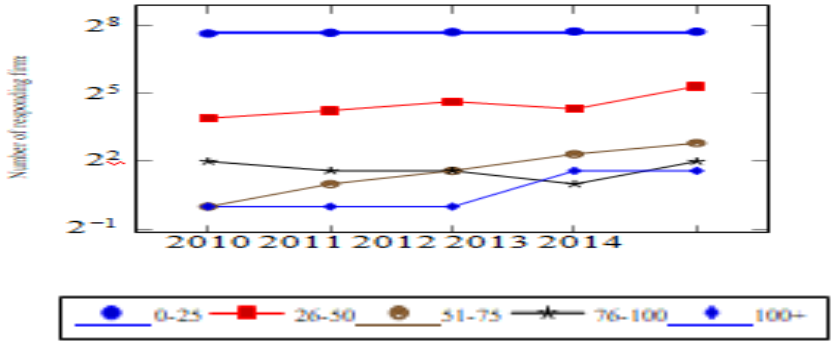

Figure 2: Firm size and Growth Pattern distribution

The Bangladesh IT sector has seen growth in income, personnel size, and firm count, according to a BASIS survey.3 million professionals work in 4500 businesses, generating $800 million, according to the poll. ITES businesses make up 44%. However, due to a shortage of large IT companies with more than 500 workers to manage significant local and foreign projects, growth is slower than planned. India has a history of being present in this area, in contrast.

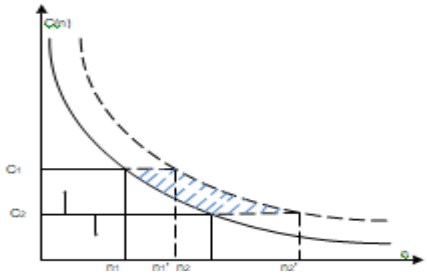

Figure 3: Downward drift of willingness to pay curve due to competition

In this paper, we are investigating high level project diversity which is one of the major reasons of slow growth of the software firms. Our study shows, in context of Bangladesh most of the firms are doing business in multiple market segment shown in Figure 3 and Figure 4.

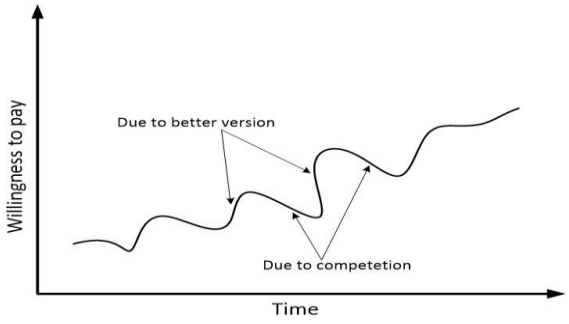

Figure 4: Effect of better version and competition on Willingness to pay

As a result, firms are not able to take advantages of scale and scope benefit. This paper intends to propose a guideline for solving the above mentioned problem. Through economies of scale and scope, the suggested approach might have a favorable effect on the nation's economy, helping software companies as well as clients. High project variety presents several challenges, including:

2.1 Poor economic scale

Consumer dispersion caused by competition in similar market segments limits the advantages of economic scale.

2.2 Lack of Research and Development

Due to scarce resources and R&D spending, diverse programs impede innovation. Therefore, software firms are not being able to increase the willingness to pay as shown in Figure 3.

2.3 Restraining Creativity

Due to project diversification employees are switching from one project to an- other project. Developers are failing to be expert in any particular domain. Creativity of the employees is not developing adequately.

2.4 Poor management

From the management point of view, managing multiple products or projects is a stressful process. Frequently switching in between projects leads to poor management [0].

2.5 Poor marketing strategy

Consumer dispersion caused by competition in similar market segments limits the advantages of economic scale. Hence, products fall into decline stage before going to its maturity level.

2.6 Miscommunication

With the increasing number of diversified projects, firms need to hire more and more professionals which create miscommunication within the team, henceforth rework are generated.

2.7 Less chance of getting to profitability

The less focused a firm is on its single specialized product the lesser the chance of being able to provide products with enhanced user experience, causing difficulties to get to profit generation.

These obstacles can be solved by concentrating on a single product, stressing software reuse, particular techniques, and human factors.

3 Research Issues

Major software companies are forced to improve cost, quality, and development speed by fierce rivalry . Beyond technical knowledge, software marketing encompasses. Managing a variety of tasks frequently reduces the desire of clients to pay and hinders growth . More efforts result in higher costs for research and development . Sustainable growth can be fueled by focused project diversity and reusable assets . We suggest predicting profitability using neural networks, MCDM, and clustering . This improves visibility, minimizes the risk of failure, and helps choose projects . Accurate evaluation of the factors and decision criteria is necessary for leveraging economies of scale and scope. Prioritize focused initiatives and reusable assets for increased efficiency to solve excessive project diversity. Principles for incremental software asset development encourage growth, increase reuse, and lower delivery costs.

3.1 Market Research

Analyzing market requirements, size, and competition is essential before developing software. Understanding client willingness to pay is crucial, influenced by imitation, innovation, and substitution. Innovation can counter downward trends by offering improved versions. Firms should compare product features and dissimilarities during research to guide decision-making. Software marketing has become critical due to industry agility. It's recognized in.SWEBOK & PMBOK .

3.2 Profitability prediction of a product

For project prediction, we provide three decision metrics and four essential product features. Language assessments, like Table 1, assess the importance of a factor. Organizations can introduce weights based on the significance of the project's features, accommodating various factors' relevance with specific weights that the organization has set.

Table 1: Linguistic Scales For The Importance Weight

The parameter is less important, less relevant, less available	Extremely	0.0
	Highly	0.1
	Very	0.2
	Strongly	0.3
	Moderately	0.4
	Equally	0.5
The parameter is more important, more relevant, more available	Moderately	0.6
	Strongly	0.7
	Very	0.8
	Highly	0.9
	Extremely	1.0

Featurei, ∑NFeaturei WFeaturei f actor= 1.0 and NFeaturei represent total factors of $Feature_i$. In this paper we introduce total 11 features and we provide the features and key measuring factors.

Back-end (database) 'F_1': The measuring factors under back-end F_1 might include the following where $N_{F_1} = 7$ and Im- pact is shown by *Impact* (F_1)

- F_{1_1}: Price of database;
- F_{1_2}: Security of t he database;
- F_{1_3}: Environment to deploy;
- F_{1_4}: No. of active connection per second;
- F_{1_5}: Competitiveness of database;
- F_{1_6}: Support service
- F_{1_7}: Viability and sustainability concerns;

$$\text{Impact}(F1) = \sum_{i=1.}^{NF1} F1i \quad WF1i \quad (1)$$

Middleware 'F_2'; This structure, content management and Business logic of the product. Below is the possible measuring factor

where $N_{F_2} = 5$.

- F_{2_1}: Business strategy of the firm;
- F_{2_2}: Security of the system;
- F_{2_3}: Stability and Reliability.
- F_{2_4}: Performance of the system.
- F_{2_5}: Viability and sustainability concerns;

$$\text{impact}(F2) = \sum_{i=1}^{NF2} F2i * WF2i \quad (2)$$

Application (Front-end) layer 'F_3': Factors are provided below where N_{F_3} = 9: to interact with the system. Possible measuring fact-

- F_{3_1}: Data loading time;
- F_{3_2}: Responsiveness of the system;
- F_{3_3}: User friendliness.
- F_{3_4}: Compatible with back-end database.
- F_{3_5}: Ease of use;
- F_{3_6}: User friendliness;
- F_{3_7}: Competitiveness to front-end design.
- F_{3_8}: Design concerns;
- F_{3_9}: Application concerns;

$$\text{Impact}(F3) = \sum_{i=1}^{NF3} F3i * WF3i \quad (3)$$

User Requirements 'F_4': User requirement deals with user awareness of the requirement of the soft- ware and the features. Moreover, users' understand- ability is part of this segment. Below are the measuresthat contribute in this feature where $N_{F_4} = 6$.

- F_{4_1}: Ambiguity of requirements;
- F_{4_2}: Preciseness of requirements;
- F_{4_3}: Modifiable requirements;
- F_{4_4}: Independent design requirements;
- F_{4_5}: Understandable requirements;
- F_{4_6}: Organized requirements.

$$\text{Impact}(F4) = \sum_{i=1}^{NF4} F4i \quad * \quad WF4i \quad (4)$$

Hardware 'F_5':Types of software to develop the expected product is the concern in

this feature. Below we list the possible measuring factors where $N_{F5} = 6$.

- F_{51}: Price of the hardware;
- F_{52}: Performance of the hardware;
- F_{53}: Availability and Stability;
- F_{54}: Adaptability;
- F_{55}: Support service.
- F_{56}: Viability and sustainability concerns;

$$\text{Impact}(F5) = \sum_{i=1}^{NF5} F5i * WF5i \quad (5)$$

Software 'F_6': Software or tools which are re- quired to develop for the expected product are issues that have impact on the selection of the project. Be- low the measuring factors under this feature is shown where $N_{F6} = 7$.

- F_{61}: Software price;
- F_{62}: Performance;
- F_{63}: User friendliness;
- F_{64}: Availability and Stability;
- F_{65}: Support service.
- F_{66}: Framework;
- F_{67}: Load speed and Response time

$$\text{Impact}(F6) = \sum_{i=1}^{NF6} F6i * WF6i \quad (6)$$

Software Architecture 'F_7': This feature refers to the high level structure of the system, usability, pe rformance, reuse and technological constraints. Possible measuring factors might include where $N_{F7} = 7$

- F_{71}: Business process;
- F_{72}: Time to implement the product;
- F_{73}: Research and Development cost.
- F_{74}: Stability and Reliability.
- F_{75}: Accessibility;
- F_{76}: User specification of environment.
- F_{77}: Viability and sustainability concerns;

Table 2: Decision Making Parameters (DMP)

DMP	Details	Effect
Strategies	It defines scope of the firms, product planning and strategies considering either key measurement factor to market the product and where to position	An over value of 0 to 1 by considering either key measurement factor or features or area depending on the business strategy of the firm.
Market Competition	To get a product/project the number of firms working in the same market segment.	Depending on the market segment or competition firm will assign weight

		from 0 to 1.
The Organizations competitiveness	Ability of a firm to build product and services to meet the user demand	Firm will assign a value 0 to 1 depending on the competitiveness of either each of key measurement factor or features or area.

$$\text{Impact}(F7) = \sum_{i=1}^{NF7} F7i\ WF7i \quad (7)$$

However,

$$Total\ Impact(F) = \frac{1}{N_F} \sum_{j=1}^{N_F} \sum_{i=1}^{N_{F_j}} (F_{ji} \quad W_{F_{ji}}) \quad (8)$$

Using the DMP business strategy as an illustration, a software company might take on a project that is different from its current portfolio but will have a large impact on the market in the future. Table 2 lists the DMP specifics and their results. Any software company starting a new project can use our suggested fix. For prediction analysis, they can use information from prior projects' features and factors (such as very significant, relevant, moderate, etc.) as well as DMP values from Table 2. DMP values can be added to the formula8 to improve it as seen below.

$$\text{Total Impact FJ} \quad 1\ (\ ) = \text{NDMP} * \text{NF}=1 \sum_{j=1} \sum_{i=1} \text{Fji WFji DMPk} \quad (9)$$

For a particular project (P), by providing values of the factors related to the features and the decision making parameters provided in Table 1 and Table 2 respectfully, the impact value $Total\ Impact_P$ can be generated using the equation 9. The company must devise a threshold value (δ) to compare the impact and come up with a Final Decision to "Accept" the project or "Reject" the project (P) by define the (FD_P) shown below.

$$FD_P = \begin{cases} \text{Accepted} & \text{if } (Total\ Impact_P) \geq \delta \\ \text{Reject} & \text{otherwise} \end{cases} \quad (10)$$

4 Conclusion

To prevent problems with slow growth, it is essential to achieve economies of scale, scope, and innovation. High project diversity forces businesses to put in a lot of work without reusable parts, which hurts their bottom line and impedes their

ability to compete. Low project diversity can encourage the development of software-based wealth. Companies can assign values to projects, as was previously discussed, focusing on important measuring elements to lessen diversity. Results establish their impact after looking at market research and utilizing these factors. Based on findings, further improvement and modification using decision-making algorithms can be made. By implementing the suggested method, key project-related characteristics and aspects are stored, enabling comparison with prior projects.

References

1. Y. Kim and E. A. Stohr, “Software reuse: survey and re- search directions,” Journal of Management Informa- tion Systems, vol. 14, no. 4, pp. 113–147, 1998.
2. I. Mundi, “Bangladesh vs. india,” 2021, ac- cessed: 2021-02-06. [Online]. Available: https://www.indexmundi.com/factbook/compare/bangladesh.india BASIS, “It and ites industry overview,” BASIS, Tech. Rep., 2017.

   M. M. Astrahan, B. Housman, J. F. Jacobs, R. P. Mayer, and W. H. Thomas, “Logical design of the digital com- puter for the sage system,” IBM Journal of Research and Development, vol. 1, no. 1, pp. 76–83, 1957.
3. Small and M. E. Foundation, “Software industry in bangladesh,” SME, Tech. Rep., 2017.
4. R. Zaman, “Why could not bangladesh’s software industry grow?” 2015, accessed: 2021-02-06. [Online]. Avail- able: https://www.linkedin.com/pulse/why-could- bangladeshs-software-industry-grow-rokon-zaman
5. P. Mohagheghi and R. Conradi, “Quality, productivity and economic benefits of software reuse: a review of industrial studies,” Empirical Software Engineering, vol. 12, no. 5, pp. 471–516, 2007.
6. Y. Kim and E. A. Stohr, “Software reuse: survey and re- search directions,” Journal of Management Informa- tion Systems, vol. 14, no. 4, pp. 113–147, 1998.
7. T. J. Biggerstaff, “An assessment and analysis of software reuse,” in Advances in Computers. Elsevier, 1992, vol. 34, pp. 1–57.
8. Y. Fuqing, M. Hong, and L. Keqin, “Software reuse and software component technology [j],” Acta Electronica Sinica, vol. 2, 1999.
9. L. N. Raha, A. W. Hossain, T. Faiyaz, M. Hasan, N. Na- har, and M. Rokonuzzaman, “A guide for building the knowledgebase for software entrepreneurs, firms, and professional students,” in 2018 IEEE 16th Interna- tional Conference on Software Engineering Research, Management and Applications (SERA). IEEE, 2018, pp. 165–171.
10. S. Tahsin, A. Munim, M. Hasan, N. Nahar, and M. Rokonuzzaman, “Market analysis as a possible ac- tivity of software project management,” in 2018 IEEE 16th International Conference on Software Engineer- ing Research, Management and Applications (SERA). IEEE, 2018, pp. 180–183.
11. T. Z. Khan, S. H. Tusher, M. Hasan, and M. Rokonuzza- man, “Designing and developing a game with market- ing concepts,” in Advances in Computer, Communica- tion and Computational Sciences. Springer, 2021, pp. 245–252.
12. K. P. Sinaga and M.-S. Yang, “Unsupervised k-means clus- tering algorithm,” IEEE Access, vol. 8, pp. 80 716– 80 727, 2020.
13. M. Pavan and R. Todeschini, “Optimization: Multi-criteria decision making methods,” 2008.
14. C. Zhuang, S. Yan, A. Nayebi, M. Schrimpf, M. C. Frank,J. J. DiCarlo, and D. L. Yamins, “Unsupervised neural network models of the ventral visual stream,” Pro- ceedings of the National Academy of Sciences, vol. 118, no. 3, 2021.