Mass market development strategies of software industries: Case study based research

Varun Gupta a,*, Kamlesh Dutta b, Durg Singh Chauhan c

a Amity University, Noida, Uttar Pradesh, India  
b National Institute of Technology, Hamirpur, India  
c GLA University, Mathura, Uttar Pradesh, India

Received 19 February 2016; accepted 6 April 2016  
Available online 20 April 2016

KEYWORDS  
Mass market software development;  
Market share;  
Market segments

Summary  
The success in competitive mass market software development depends on the quality of software development and market segments targeted. Market segments are categorized by uncertainties contributed by "Newness" and "turbulences", making the software success stochastic in nature. Selecting good market segments, delivering high quality software versions in the lowest time than competitors, result in increasing demand in markets and ultimately revenues. Enhanced customer base is beneficial for current product as well as for future products of industry in the form of increased reputation and increased involvement of customers in future development. The case study was conducted with 13 representatives drawing experiences of 14 mass market projects. Results indicate that software solutions are delivered to few investors or in highly competitive markets, as per the survey's findings of the marketing departments. The software organizations are reluctant to deliver relatively complex solutions in new markets unless and until strongly convinced with the probable success. The method for selection of market segments belonging to new and existing markets for undertaking the software delivery is also proposed in this paper. The model will help software industry decide the market segments and high abstract level features that could increase probability of software success. Poor selection of markets or targeting markets of "improper" size affects the market share of the industry to a great extend.

© 2016 Published by Elsevier GmbH. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Introduction  

The Software Industry had occupied a central position in satisfying the Information Technology (IT) needs of the organization. Increased demands of the valuable software in various domain industries have resulted in an increase
of 4.8% in worldwide software revenue in 2013 as when compared to 2012 revenue (Gartner report available at http://www.gartner.com/newsroom/id/2696317). The software Industries are a good source of high salary employments and enhance the economy of any country (http://www.bsa.org/country/public%20policy/~/media/files/policy/security/general/sw_factsfigures.ashx). The software industry is a boom for any country. The ability of the software industry to deliver high quality software to its customers acts as the catalyst for large revenues and strong market share. Not only the ability to deliver high quality software, but also the high market share, attracts more investors. This trend finally converges into the country been known as a software market leader.

The demand of the software industry is not uniform in all the countries of the world and few countries may provide large markets while some may prove to be saturated ones. For example, the sales of packaged software in the BRIC (Brazil, Russia, India, and China) markets are likely to be $22 billion dollars, showing an increase of 44% which is far more than the projected growth of 15% in the United States; Central, Eastern, and Western Europe; and Japan collectively between 2009 and 2012 (http://www.bsa.org/country/public%20policy/~/media/files/policy/security/general/sw_factsfigures.ashx).

To capture high market share, a software industry must target those markets that have high demand for software solutions rather than trying fate in saturated market segments. Saturated market segments provide a high degree of risk because attracting the competitive software customers and new customers is a bit difficult. The difficulty can be overcome by release of innovative software features at fairly good competitive prices that make the transition to new software possible and affordable.

Once the software is released in market segments, it can be customized as per the needs of small, medium or large business firms. Once again the reputation and market share of the software industry provides the necessary magnetic attraction to the business firms as investors. The ability of software industry to target and satisfy the market(s) is prime parameter for increasing revenues and market shares. Market shares are always beneficial to any industry in terms of profit margin, declining purchase to sales ratio, lower marketing cost to sale, higher prices of products, unique competitive strategies, etc. (Harvard Business review article at https://hbr.org/1975/01/market-share-a-key-to-profitability).

Today, many software industries claim to be delivering software solutions to mass markets and it would not be strange if you find the solutions of the same firms delivered to small, medium and large business firms as well. The ultimate reason is to convince the investor, sell the software and gain profits. Failed software is an ugly gift to any software industry, which may turn out to be a loss in reputation (less investors/customers), loss in market share (losses), enhanced profit to competitors, complete shutdown of business and much more. Software industries must figure out the proper methods to target suitable markets to attract customers, enhance sales, build reputation, etc.

This process may continue with other market segments. Enhanced customer base will enhance revenues of the industry and may also provide a good market for future product of the industry due to enhanced reputation.

This paper aims to explore the mass market development strategies of well known large companies at a higher level of abstractions, i.e. without going through the details of the development process and keeping the work restricted to key decision makings in delivering to mass markets.

**Interviewed companies**

The researchers interviewed three multinational software industries that had experience in delivering numerous software solutions. Each solution mostly has a target audience of millions and is often aimed at mass markets. However, the industry had provided the customized solutions to few business firms and also developed solutions especially for few clients. The industries will be denoted by A, B and C, because as per the condition of the interview, the researchers agreed that neither the company name nor the employee identity will be disclosed. Each company is represented by a few representatives at the level of project managers and software engineers. The representatives have the experience with the development of numerous projects of varying complexities. The details are given in Table 1.

The representatives of the software industries were asked about the manner their industries decide to deliver the software solution in mass markets. The aim is to know the different strategies employed by these industries in targeting the market segments. The reason for employing the software engineer is to get more details about the process (in case project manager forgets to bring out something meaningful) and to analyse the responses made by project managers in order to strike off wrong interpretation due to conflicting terminology and falsifying information if any to hide any failures/weaknesses by project managers.

**Results**

The industry representatives told us the method employed from the phase of conceiving the idea up to the phase of delivery of the release. All the companies employed AGILE Scrum methodology for releasing the version; however, the activities related to targeting the suitable mass markets vary considerably only in words and not in deeds. The actual method was same across all the companies. The prominent strategies employed by these companies are as follows:

- **Fixed investors:** The companies normally have fixed investors, those who had been buying software for long and those who brought them recently. The companies are

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Company name</th>
<th>Number of representatives</th>
<th>Number of projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>A</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>B</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3.</td>
<td>C</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
the first choice of these investors whenever new development has to take place, especially for getting price quotations. It does it reasonably well for these industries to hit the deal by offering somewhat more innovative solutions at a lucrative price if they feel the presence of competitors. To gain more profits, these companies charge huge amounts for implementation of extra requirements after the release of those agreed. Development under such a scenario is not bespoke because companies try not only to deliver the product to selected investor, but also try to add a little bit more functionality to deliver simultaneously to other needy business firms.

- **Competitor offerings**: The software industry marketing departments keep an eye on the demands and competitors’ activities in different market segments. The companies try to focus on innovative features at lowest rates. Efforts are to figure out weaknesses in existing competitor products by strong marketing of the new solutions developed.

- **Completely new software solution**: The companies had resistance in admitting that high degree of uncertainty in new markets forces them out in such type of developments until and unless the companies have some expertise in development of similar products in markets. For example, if one company has delivered many versions of operating systems in the markets and the system is a any better operating system (OS) and is to be developed for mobile with model number say "X".

In new markets, how much the customer would like to pay for the software and how many will buy it are not known. If the industry has experience with the development of the OS, then it may be possible to undertake the development with in less cost and time due to availability of reusable OS code and experience. The lack of sound stakeholder identification techniques, complete reliance on marketing department, unavailability of short time delivery methods, high degree of risks in new markets and market turbulences/fluctuations hinder the idea of launch of new software product until and unless higher executives are highly confident of the success.

However, fewer representatives told the researchers that the marketing department is continuously involved in exploring the possibility of new markets for existing products and new products and development is not restricted for delivery of projects similar to those delivered previously. Software solution could be completely new in domain and technology.

**Proposed model**

The proposed model provides a set of activities that would help the software industries to undertake software development in new and existing markets without the fear of "'newness'" and existence of market fluctuations in existing markets. In new markets, uncertainty pertains about customer preferences, ability and extend of purchasing capability of customers, etc. Existing markets provide a

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Algorithm to identify market segments for developed software.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALGORITHM</strong>: SELECT (): This algorithm identifies the suitable market segments that could be targeted by the software industry to boost up the sales. The algorithm will be called repeatedly by the industry till the industry is in business of software development. Let M be the worldwide market and Ni be the individual market segment. Let D be the set of application domains, in which software industry has specialization to develop. Clearly, M will be fragments into Ni’s for each Di belonging to D.</td>
<td></td>
</tr>
<tr>
<td>1. For each Di in D, classify M into three categories:</td>
<td></td>
</tr>
<tr>
<td>- Unexplored: M is undivided meaning it’s a new market for the software industry. Either of two situations pertains — Competitors are offering solutions in various segments of M (call it state = &quot;Virtual_unexplored&quot;) or the software industry is first to think about the solution (call it state = &quot;True_unexplored&quot;).</td>
<td></td>
</tr>
<tr>
<td>- Explored: M is divided into Ni’s meaning that software industry had already few versions available for Ni number of segments. Here state = &quot;Explored&quot;.</td>
<td></td>
</tr>
<tr>
<td>2. If(State = &quot;True_unexplored&quot;)</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>True_unexplored();</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>if(State = &quot;Virtual_unexplored&quot;)</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>Virtual_unexplored();</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>if(State = &quot;Explored&quot;)</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>Explored();</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>3. Exit</td>
<td></td>
</tr>
</tbody>
</table>
software industry historical data about customers and competitors, reasonably positioned to look at the market from a competitive viewpoint.

Existing markets may have a high degree of fluctuations, i.e. customer expectations, or may be saturated one. In case of the highly fluctuating market, there is a need to deliver only that which provides competitive advantage and has a high degree of acceptability among customers in minimum possible time. The time of delivery depends on the type of market, i.e. it is better to deliver in minimum possible time the new software version in order to attain a high level of project success in unfamiliar, emerging, or fast-changing market (newness and market turbulences) rather than stable markets (Chen et al., 2005).

Saturated markets do not have continuously changing requirements and targeting those would be quite difficult. Targeting them would probably mean delivering the highly innovative functionality that even customers would not have thought of but will be able to please them. The sales would depend on the industry’s ability to market the developed product (marketing involves huge costs) and selling price must be the one that attracts customers and overcomes development plus marketing costs. Thus the variable profit becomes highly stochastic in nature.

In the market, the customers may have different types of demands like Homogenous, Clustered and Diffused demands (Barnes et al., 1997). Pertaining to the domain of Software Industries, the customers are highly diffused. The initial demands of customers in new markets may be homogenous, which gets transformed into clustered and finally diffused due to ever growing familiarity with the working version of the software. Another reason for the transformation is availability of Agile Based time boxed software engineering approaches that make it possible to do probabilistic delivery of small version and improved deliveries in the future.

As the software industry integrated new market segments, the increase in customer base will increase, leading to increase in the effort of software engineering activities especially requirement engineering that happens to be a highly participatory one.

Pseudocodes given in Tables 2–5 provide a set of activities that make it possible to identify the market segments for developed/underdeveloped software in mass markets by extracting innovative and impressive features as required by the customers. The algorithm considers the uncertainties in markets and only assumes that the software industry is capable enough to deliver the software as per computed market sensitive parameters like customer needs, cost, time, etc. The software industry can get useful information about the market segments and their turbulences but adopting sound software engineering practices is their responsibility. The working algorithm for True_unexplored(), Virtual_unexplored() and Explored() is given in Tables 3–5.

Table 3 Algorithm to Implement True_unexplored() routine.

<table>
<thead>
<tr>
<th>ALGORITHM: TRUE_UNEXPLORED(): This algorithm utilizes the historical information of similar domain projects released so far by various companies. Historical information of prime importance involves previous sales record, stakeholder’s details, etc. The segments according to their sales are grouped in either one of the three arrays i.e. Less, Medium and high. Variable &quot;N&quot; contains the selected segments.</th>
</tr>
</thead>
</table>
| 1. Identify the small, medium and large businesses those who had delivered software solutions of similar domain in market M. For example, if new software is Spoken tutorial software then the companies dealing with engineering software in the market M could be of prime importance. Let call the segment "k" created by the company "i" for software project "j" be given by triplet (i, j, k).
2. For every "j", group the segments in accordance with the sales of the project "j" in market segment (i, k). The sales could be taken according the measurement scale of Less (0–20%), Medium (21–40%) and high (41–100%) and accordingly the entry of (i, j, k) could be made. At the end, the three sets will house the segments numbers along project "j" and company "k" details in accordance with the sales.
3. Place the segments belonging to both Medium and Large sets in segment array "N". Further, identify the segments those belonging to both Medium and Less arrays. Place these segments in N_less array.
4. Identify the potential stakeholders of the software in accordance with the application domain especially legal experts and end users of above identified segments.
5. Identify the stakeholders belonging to segment area of N_less array. Analyse the current situation of the business affairs of the identified users (N_less array) without the launch of proposed software solution. Highlight these identified important areas affected by the launch of the software. Denote the individual areas by Ai where Ai belongs to set A.
6. Let Ci be the cost and Ti be the time in carrying out the identified area activity Ai without the software. The values of these parameters will be different for different segments. They may be added together if values are similar to few segments. Let the frequency of the activity (Ai) be Fi. Frequency denotes the number of times in a year, the activity is repeated. For all areas Ai, I = 1 to M, C = Fi * ∑Ci for i = 1 to M T = Fi * ∑Ti for i = 1 to M Estimate the cost and development time of proposed software employing the previous project experience. Let the cost be Cs and time be Ts. Add only those segments in N_less to N if and only if Cs << C and Ts << T. Ignore remaining segments.
7. Further analyse the segments in "N" and merge or decompose them in accordance with their behaviour towards new software as per marketing surveys. |
Table 4 Algorithm to Implement Virtual_unexplored() routine.

**ALGORITHM**: Virtual_unexplored(): This algorithm identifies markets and the features that will provide competitive edge in release of highly advanced software version.

1. List the features provided by the competitor’s software’s. Let the features provided by company ‘‘i’’ be denoted by \( F_{i,j} \).
2. Group the features \( F_{i,j} \) into similar requirements and call the similar features as \( F_i \). Let the set \( F_d \) denotes the different features of competitors software’s.
3. Select the segments associated with less sales of the software’s. Denote the set by \( N_s \). Elicit the potential features as expected by customers associated with these segments, denoted by \( F_t \). Also, gather the cost and time requirements of the customers.
4. Let the set \( F \) be the features to be implemented and \( F = F_d \).
5. Predict the possible features as expected from the proposed solution using historical data (if available) and experience of software engineers. Let the set be denoted by \( F_p \).
   \[ F = (F_p - F_d) \cup F_d \]
6. Select the features from the set ‘‘F’’ that could be implemented within cost and time constraints.

Table 5 Algorithm to Implement Explored() routine.

**ALGORITHM**: Explored(): This algorithm further explores the market for any additional needs of customers that may inhibit the entry of competitor.

1. Perform sales analysis and select the segments associated with less and medium sales.
2. Gather the reasons for low sales and map the reasons to either feature set \( F \) or constraint sets \( C \). Constraint set contains various development constraints like cost, time, etc.
3. Mapping to feature set means the need for implementation of delayed or new requirements while mapping to constraint set means unusual values of parameters like cost and time. Perform more exercise with customer to know what they require more from the product and consider the constraints as well.

Conclusion and future work

The software industry delivers software in those market segments of mass markets that have high a degree of being successful in terms of enhanced sales. The value of probability of success is communicated mostly by the marketing department. Marketing department analyses the competitor actions, needs of existing investors and to a little extend the new segments that require some software solution. The industries require very strong reasons from marketing department to undertaking the software development for completely new market segments due to presence of uncertainties of ‘‘newness’’, fear of losses, potential ‘‘local’’ competitor, etc. The process model is to be employed in the selection of market segments (segments being the part of new and/or existing markets) during the implementation of software solution. The proposal could be of quite great importance to software industry due to its ability because ability to select good markets enhances the customer base, which is always beneficial for current and future software developments. In the future, we expect the automation of the proposed model and subsequent applicability in the software industries involved in mass market developments.

Authors’ contributions

All the authors of this paper have equally contributed to this paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors are thankful to the software engineers who participated in the empirical survey. The data shared by the interviewed software industry representatives laid the basis for the empirical research work.

References
