A Model for Value Based Requirement Engineering

Sk. Md. Nahid Hasan [†]**, Mohammad Shabbir Hasan** [†]**, Abdullah Al Mahmood** [†]**, Md. Jahangir Alam**[††] **,**

,

[†]Panacea Research Lab, Dhaka, Bangladesh
[††] Dhaka International University, Dhaka, Bangladesh.

Summary

Software companies are forced to become increasingly competitive and responsive to consumers as well as market development due to rapid changes in the current world and global competition. It is important for software companies to maximize product value through proper requirement engineering. The purpose of requirement engineering activities is to add business value that is accounted for in terms of return-on-investment of a software product. This paper introduces some of the fundamental aspects of value from the perspectives of economic theory, discusses a number of the challenges faced by a requirement engineer and finally describes a model to define value from business, product and project perspectives.

Key words:

Value Based Requirement Engineering, Product Value, Business Value, Project Value..

1. Introduction

Incremental software development is becoming an increasingly commonplace practice among software companies as they have discovered the potential of this approach to reduce the amount of effort and time that needs to be invested in a product development prior to its release [7, 10, 11]. Now a day, increasing global competition, dynamic market needs and new technologies are some of the challenges faced by the software companies and hence there is an incredible pressure on these companies to achieve and sustain competitive advantage. During product development, it is important for the software companies to focus on the value of different customers and markets for keeping themselves competitive in an era of increasing uncertainty and market globalization. The critical success factor for software companies is their ability of developing a product that meets customer requirements while offering high value that provides increased reassurance of market success, provided that the product is released at the appropriate time [1] and offers a superior level of quality relative to competitors [2].

Value is created when a company makes a profit and adding value is an economic activity that has to be taken into account from a software business perspective. Since the ultimate aim for a software company is to maximize value creation for a given investment, it is essential to understand the relationships between technical decisions and the business strategy that drives the value [3]. Boehm ([2], chap 1) stated software engineering (SE) is largely practiced in a value neutral setting with every requirement, use case, test case, defect and object being considered equally important, even though not all requirements are equal. Traditionally, there is a separation of concerns – software developers are confined to turning requirements into verified code [4]. Yet Bullock (2000) found that eighty percent of business value comes from only twenty percent of software components. Furthermore, there is often a mismatch between the decision criteria used by software developers at the organizational level, and the value creation criteria used by software development organizations [3]. All these lead to the statement that the alignment of product, project and business decisions is a major problem in the software industry.

The value-based approach in requirements engineering (RE) promotes the alignment of product, project and business decisions [1, 5], and the involvement of multiple stakeholders' perspectives in the creation of product, project and business value, while aiming to maximize the value of a release of software through the selection and prioritization of requirements [6]. Despite the fact that most release planning literature covers prioritization and dependencies between requirements [7-11], there has been little research into the criteria used in this decision-making process around requirement selection [6].

The paper organization is approximately as follows. In section 2 we discuss the background knowledge on value based approach in software development inspired by the terminology and concepts borrowed from economic theory. Section 3 describes a model for value based requirement engineering. This is followed by a discussion in section 4. We finish by drawing some broad and necessarily speculative and personal conclusions about alignment of technical decision with business strategies for ensuring effective value creation and management.

2. Background

2.1 The Concept of Value

John Stuart Mill, who had a strong influence on economic theory, defines the concept of value, at a very high abstract level, in terms of *use* and *exchange value* [12]. A *use value* is what the customer is willing to pay for the product, and an *exchange value* is the market value of the product. As Mill's definition of value dominated economic theory in 19th century, the term "value-adding" became very popular in the early 20th century. By the end of 1980s the focus of product development was placed on the relationships between the customer service and customer needs. This approach was based on the notion that value was related to long-term relationships between the customer and the company (customer oriented approach). Value was created in cooperation with the customer where the customer was an active participant in value creation activities [13, 14].

In late 1990s, the concept of the value-based approach in SE was introduced in the context of decision-making about product lines [15], managing investments in reusable software [16] and software economics [3]. Since then the value-based approach has attracted both software practitioners and academics and leading them to integrate value considerations in existing and emerging software principles and practice [2].

2.2 Defining Value

Value constructions in economic theory are based on customer satisfaction, loyalty and re purchasing behavior [13]. By borrowing the economic theory, three aspects of value can be addressed, namely *product value*, *customer's perceived value* and *relationship value*. *Product value* is related to the product price and influenced by the quality attributes of the software product. The value of a product increases in direct proportion to its advantage over competitive products or decreases in proportion to its disadvantage [14]. A *customer's perceived value* is the benefit derived from the product and is a measure of how much a customer is willing to pay for it, i.e. *perceived value = perceived benefits/perceived price*, where perceived benefits and price are both measured relative to competing products [18, 19]. A customer's perceived value is influenced by some factors like customer's needs, expectations, past experiences, and culture. *Relationship value* is created through the social relationships between the software company and the customer. It exists through the product and customer's perceived value.
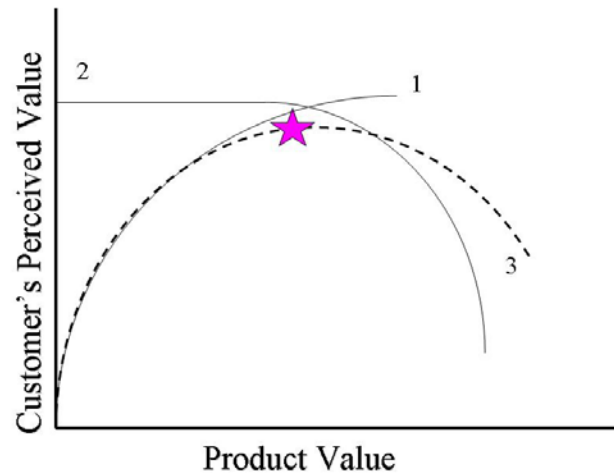

Figure. 1. Customer's Perceived Value and Product Value Relationship (adapted from [18])

It is important to understand the relationship between customers' perceived value, and the time and money spent on product development [18] (see Figure 1). A customer views a purchase as a good deal when *customer's perceived value > perceived price* of the product. If the *price > product cost* then the software company makes a profit on their sale. If the customer's perceived value is assumed to be equal to the technical performance of the product (which is the traditional approach in software development), then the perceived value continues to increase, up to a certain point, as more time and money are spent on product development [18] (curve 1 in Figure 1). At the same time, the product price will increase, because of the more time and efforts put into product development. As a result the customer will not be able to afford the product; he/she will start looking for similar products within the market or will decide to wait until the product price will go down. In turn, this will cause a decline in customer's perceived value (curve 2 in Figure 1). Hence, it is important to understand at which point additional effort (on product development) is not worth to marginal improvements as it will effect product level decisions [18]. Allocation of this point (star on curve 3 in Figure 1) will be strongly influenced by customers' perceived value and other products within the existing market. In the context of incremental product development, this point needs to be re-calculated during every requirements selection process which is necessary when a new release is about to be made and it should be remembered that the customer's perceived value is subject to change due to their varying expectation, needs and past experiences.

The critical success factor for software companies depends on ensuring a certain guarantee of market success [3, 20] which is achieved through their ability to develop a product with high value to meet customer requirements. Since a software company proceeds with an aim to

maximize value creation for a given investment, it is essential to understand the relationships between product, project and business level decisions and the business strategy that drives the value [1, 5, 21].

2.3 Value Based Approach in Requirement Process

Today software has a major effect on the cost, value and schedule of projects [22]. However, an organization's success in terms of profitability or market capitalization does not necessarily correlate with their level of investment in IT [23]. This may happen in some cases as money spent does not always translate into the realization of benefits. Most studies on the critical success factors of successful and failed projects, find that the primary critical success factors lie in the value domain. Most projects fail due to lack of user input, incomplete requirements, lack of resources, unrealistic expectations, unclear objectives and/or unclear timeframes [22]. The reason behind this is, projects are tracked by monitoring project cost and schedule [24] which unfortunately, does not consider stakeholder or business value. A project can be successful in terms of cost (i.e. if it is finalized within its budget), but may fail to provide any business value. This can be due to ineffective tracking of a project when the project plan changes rapidly, flaws in user acceptability, the system not being cost effective operationally and success of the project also require timely market entry.

The purpose of the requirements process is to add business value [16]. In the era of global competition, the emerging discussions in SE indicate that a value-based approach makes all the difference in creating successful product and value for software business as it puts the requirement engineer in the position of managing requirements to make the most strategic opportunities. Although companies put a great amount of effort into increasing customers' perceived value in the product development process, determining *how* and *when* value is added is still a challenge even in marketing and management science (Gordijn and Akkermans, 2005). This is because value creation strategies are highly contextual [25] and must be analyzed as part of a multidimensional array of variables [26]. Unfortunately, there is no prescribed approach to achieving this perception.

Strategy and product management should be changed with market demand [25]. Although this is only one of many factors that could influence the relative importance for criteria in the selection of requirements, it is not always possible to identify a complete list of such factors.

Techniques to reconcile conflicts include requirement prioritization techniques, business case analysis techniques, stakeholder identification and requirements negotiation techniques [24]. It is also important to consider that the value a market attaches to different requirements changes over time [27], but it is not understood how they will change [6, 28]. The critical success factor for software vendors depends on their quick response to changing requirements while maintaining a focus on their value proposition, which may, for example, yield a quicker return on investment.

Value-based requirements engineering (VBRE) aims to maximize the value of a release of software through the selection of requirements [5, 6, 28]. VBRE is a very young area of academic study, however, the problem of creating product value through requirements selection and prioritization is real to the development of software products in today's competitive environment. Companies have been forced to change their practices due to current market forces, but there is little theory providing an approach for development of IT intensive solutions that are valuable to all stakeholders [29].

2.4 Incremental Development and Release Planning

Incremental software development is a top-down approach to development in which a minimal software product is developed and released in the first increment, and a function(s) or a requirement(s) is added in each successive increment until the product is complete [4]. Each increment or product release may contain all previously elicited requirements in addition to some new requirements or functions that allow the cumulative growth of the product. This approach to software development requires the analysis of requirements, assigning them to increments [30] and releasing each increment with an aim of meeting the expectations and values of stakeholders who are involved in product development.

Release planning is the process of selecting an optimal subset of requirements for realization in a certain release in incremental software development [7]. The aim of release planning is to determine the optimal set of requirements, when they should be released and at what cost this should be achieved [27]. Researchers agree that release planning is a crucial determinant of the success of the software product as it determines which requirements will go the next release [8]. Release planning can only be conducted after a product's requirements have been elicited, analyzed and specified [6]. If release planning is done badly it increases risk. For example, leaving critical features or difficult tasks to last or ignoring dependencies and interdependencies can result in time and budget overruns and a loss of market share [27].

Karlsson suggested that release planning be approached through the prioritization of requirements [10]. Carlshamre et al. [8] and Dahlstedt and Persson [9]

furthered prioritization by recognizing and accommodating the dependencies that exist between requirements. Wiegers proposed a method of prioritization that recognized and combined benefits and penalties of proposed functionality between multiple stakeholder perspectives, relative costs and relative risks [31].

2.5 Aligning Release Planning and Value Perspectives

Many stakeholders are involved in software product development, each with their own perspective on value in the software development process [24]. For example, a project sponsor defines value in terms of the cost of the software and the benefits it provides; a software company measures value in profit; while a user looks at how well the software meets their needs. These perspectives are often incompatible, and must be reconciled [24]. For example:

• Users of the software want many features, while the project sponsor wants to limit cost by minimizing the development effort;

• Developers want stable requirements, but users want to be able to change the requirements; and

• The system maintainers want their job to be made easier, but the developers and project sponsors want control over the solution provided.

One of the most important activities in incremental development is to decide upon the most appropriate release plan [11]. Maurice et al. [27] recognized that while iterative development facilitates early customer feedback, allowing faster delivery and a more interactive process; it also creates difficulties with reconciling conflicting stakeholder perspectives. VBRE exploits the concept of economic value during the requirements engineering (RE) process [29]. Boehm [22, 24] states that in order to achieve this; VBRE must include practices and principles for: identifying stakeholders and eliciting their value propositions and reconciling these propositions into a set of mutually agreed objectives for the system.

3. A Model for Value Based Approach in RE

A value-based approach supports the alignment of decisions at business, project and product level with the aim of maximizing business value while maintaining a profit for a given investment [5]. By following this argument, it is expected that a company needs to create, measure and manage value from business, product and

project perspectives. In other words the following value perspectives are important to software developers as illustrated in Figure 2:

• **Value for business perspective:** Business value to Software Company which stems from product sale.

• **Value for product perspective:** Product value to Software Company which stems from Customer and Market requirements.

• **Value for project perspective:** Project value to Software Company stems from project budget/timing/delivery etc.

Figure 2 illustrates the relationship between the value perspectives, Software Company and Customer. The objective of this model is to show where the measurements as well as management of value are required. It is important to note that as the software company aims to maximize their business value through their product sale and related to its project, in the same way, customers' aim to maximize the value for their own business through the product purchase [5]. The model also shows the relationship value between the company and the customer which is formed through the product buy/sale transactions between these two entities. Figure 2 also includes some additional factors that influence the value creation/measurement/management for both the software company and the customer, i.e. Competitor and Market.
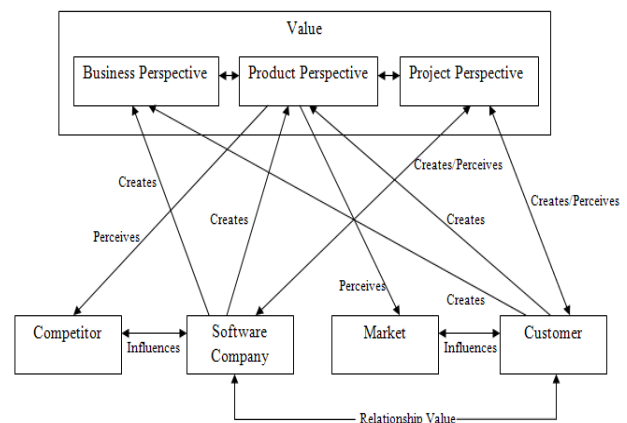

Figure. 2. A Model for value based approach in RE

It is important to note that there are some other factors that have an effect on value creation such as economic movement and social culture which are not illustrated in this model as it gets more complicated. The intention of the model in Figure 2 is to mainly address the value perspectives from a software company's point of view and provide a guideline to practitioners to give them an idea

about where the value needs to be created, measured and managed while making sure that the product, project and business level decisions are aligned and different value perspectives are involved in the decision making process.

4. Discussion and Conclusion

A value-based approach is about linking strategy, measurement and operational decisions. Unfortunately there is no "one size fits all" model for software developers that shows *when* and *how* to create, manage and measure for value [5].

Value creation in software development is not a one-off event rather it is an iterative approach which is supported by aligning product, project and business level decisions throughout the development process [1, 5, 33]. This requires that software developers firstly consider customers' requirements, business requirements and technical opportunities when making decisions. Secondly, they need to have a sound understanding of both technical and business implications of decisions that have been made throughout the development process. Thirdly, it is essential to understand the business dynamics that drive software development in terms of cost, time, and product quality as well as how software processes and products interconnect. Though value creation strategies are highly contextual, unfortunately, companies basically adopt only one strategy that best suits to their circumstances and that is successful within the context of their business environment.

Again, an effective management of the product development process contributes to sustainable competitive advantage for software companies. Managing for value requires sound understanding of company structure, business objectives, market and product strategy as well as the social culture of the company to manage for value [5].

Hence,

• It is necessary to provide timely feedback between business and technical level decision makers and to support communication between them.

• It is crucial that software developers put more effort into expressing the technical decisions as a business case while adding value to the product at hand. In the same way, management should have a good understanding of the internal structure of the company, the product and operation level decisions.

Measurement of value is always crucial for a software company. There are metrics used to measure technical performance. In many cases, regrettably, technical performance metrics mismatch (or disconnected) the business strategy that drives the value in software development [5]. Hence, alignment of key performance metrics with strategic objectives is crucial.

In this paper we tried to provide some preliminary discussion on value aspects of RE inspired by the importance of understanding the terminology and the concepts borrowed from economic theory. We believe that the alignment of technical decisions with business strategy continues to be a challenge as requirements engineers, product managers and IT managers operate at different levels. Product quality, its performance and product/project cost control (short or long term) will remain important, but the attention must be refocused on flexibility, creativity and timing.

References

[1] Aurum, A., Wohlin, C., Porter, A.: Aligning Software Engineering Decisions. International Journal on Software Engineering and Knowledge Engineering (IJSEKE) 16(6), 795–818 (2006)

[2] Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grunbacher, P. (eds.): Value-Based Software Engineering. Springer, Heidelberg (2005), ISBN 3-540-25993-7

[3] Boehm, B.W., Sullivan, K.J.: Software Economics: A Roadmap. In: Proceedings of The Future of Software Engineering Conference, pp. 319–343 (2000)

[4] S. Barney, A. Aurum and C. Wohlin, "A Product Management Challenge: Creating Software Product Value through Requirements Selection, accepted for publication in Journal of Systems Architecture, 2008

[5] A. Aurum, C. Wohlin, A Value-Based Approach in Requirements Engineering: Explaining Some of the Fundamental Concepts, International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'07), 11-12 Trondheim Norway. Lecture Notes in Computer Science 4542, 2007, pp109-115

[6] C Wohlin, and A Aurum, "Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study", in Value Based Software Engineering, S Biffl, A Aurum, B Boehm, H Erdogmus, P Grünbacher, (Eds.), pp. 183-206 Springer, Germany, 2005a

[7] P Carlshamre, "Release Planning in Market-driven Software Product Development: Provoking an Understanding", Requirements Engineering, 7:139-151, 2002

[8] P Carlshamre, K Sandahl, M Lindvall, B Regnell, and J Natt och Dag, "An Industrial Survey of Requirements Interdependencies in Software Product Release Planning", Int Symp on Empirical Software Eng, CA, pp. 84-92, 2001

[9] Å Dahlstedt., A Persson, "Requirements Interdependencies – Moulding the State of Research into a Research Agenda", 9th Int Workshop on Requirements Engineering - Foundation of Software Quality Klagenfurt/Velden, Austria, pp. 71-80, 2003

[10] J Karlsson, C Wohlin and B Regnell, "An Evaluation of Methods for Prioritizing Software Requirements", Information and Software Tech, 39(14-15):939-947, 1998

[11] G Ruhe, and D Greer, " Quantitative Studies in Software Release Planning under Risk and Resource Constraints", Int Symp on Empirical Software Eng, CA, pp. 262-271, 2003

[12] Mill, J.S.: Principles of Political Economy with Some of Their Applications to Social Philosophy (First published in 1848). Winch, D. (ed.) Harmondsworth, Penguin (1970)

[13] Heinonen, K.: Reconceptualizing Customer Perceived Value: The Value of Time and Place. Managing Service Quality 14(2/3), 205–215 (2004)

[14] Storbacka, K., Lehtinen, J.R.: Customer Relationship Management: Creating Competitive Advantage through Win-Win Relationship Strategies. McGraw-Hill, New York (2001)

[15] Faulk, S.R., Harmon, R.R., Raffo, D.M.: Value-Base Software Engineering: A Value- Driven Approach to Product-Line Engineering. 1st International Conference on Software Product-Line Engineering, Colorado (2000)

[16] Favaro, J.: Value-Based Management and Agile Methods. In: Proceedings of 4th International Conference on XP and Agile Methods (2003)

[17] Alwis, D., Hlupic, V., Fitzgerald, G.: Intellectual Capital Factors that Impact of Value Creation. In: 25th Int. Conf. Information Technology Interfaces, Cavtat, Croatia, pp. 411–416 (2003)

[18] Browning, T.R.: On Customer Value and Improvement in Product Development Processes. Systems Engineering 6(1), 49-61 (2003)

[19] Weinstein, A., Johnson, W.C.: Designing and Delivering Superior Customer Value: Concepts, Cases, and Applications. St. Lucie Press, Boca Raton, Florida, USA (1999)

[20] D.G. Messerschmitt, C. Szyperski, Marketplace Issues in Software Planning and Design, IEEE Software, 2004, 62-70

[21] CIMA, Maximizing Shareholder Value -- Achieving Clarity in Decision Making, Technical Report. The Chartered Institute of Management Accountants, UK, 2004

[22] B. Boehm, Value-based Software Engineering: Overview and Agenda. In Value Based Software Engineering, Biffl S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, 2005a

[23] J. Thorp, The Information Paradox, McGraw Hill, 1998

[24] B. Boehm, Value-based SE: Seven Key Elements and Ethical Considerations. In Value Based Software Engineering, Biffl S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, 2005b

[25] P. Kotler, The Major Tasks of Marketing Management, Marketing Management, 2 (3) (1993) 52-56

[26] V. Poladian, S. Butler, M. Shaw, D. Garlan, Time is Not Money: the case for multi-dimentional accounting in value based software engineering', Proceedings of the 5th International Workshop on Economics Driven Software Engineering Research (EDSER-5), affiliated with the International Conference in A Software Engineering, 2003

[27] S. Maurice, G. Ruhe, O.Saliu, A. Ngo-The, R. Brassard, Decision Support for Value-based Software Release Planning. In Value Based Software Engineering, Biffl S,

Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, (2005)

[28] C. Wohlin, A. Aurum, What is Important when Deciding to Include a Software Requirement in a Project or a Release? 4th International Symposium on Empirical Software Engineering, Noosa Heads, Australia, 17-18 Nov, 2005b

[29] J. Gordjin, J.M. Akkerman, Value-Based Requirements Engineering: Exploring Innovative e-commerce Ideas, Requirements Engineering, 8(2) (2003) 114-134

[30] Greer, G. Ruhe, Software Release Planning: An Evolutionary and Iterative Approach. Information and Software Technology, 46 (2004) 243-253

[31] K.E. Wiegers, First Things First: Prioritising Requirements, Software Development, 7(9) (1999) 48-53

[32] M. Berry, A. Aurum, Measurement and Decision Making. In Value-based Software Engineering by Biffl S, Aurum A, Boehm B, Erdogmus H, Grünbacher P (Eds). Springer, Berlin, Heidelberg, New York, 2005

[33] Aurum, A., Wohlin, C. (eds.): Engineering and Managing Software Requirements. Springer-Verlag, Heidelberg (2005), ISBN 3-540-25043-3

Sk. Md. Nahid Hasan received his B.Sc. (Engg.) in Computer Science and Engineering from Khulna University of Engineering and Technology (KUET), Bangladesh in 2010. His research interest includes different areas of Software Engineering like Software Metric, Requirement Gathering, Software Security and Software Maintenance. Digital Image Processing is also one of his research concerns. Currently he is working as a researcher of Panacea Research Lab, Dhaka, Bangladesh. He is also a lecturer of Department of Computer Science and Engineering in Institute of Science, Trade and Technology, Dhaka, Bangladesh.

Mohammad Shabbir Hasan received his B.Sc. (Engg.) in Computer Science and Engineering from Khulna University of Engineering and Technology (KUET), Bangladesh in 2008. His research interests include different areas of Software Engineering like Requirement Engineering, Software Metric, Software Security and Software Maintenance. He has coauthored numerous research papers published in International Journals and Conference Proceedings. Currently he is working as a researcher of Panacea Research Lab, Dhaka, Bangladesh. He is also a lecturer of Department of Computer Science and Engineering in Institute of Science, Trade and Technology, Dhaka, Bangladesh.

Abdullah Al Mahmood received his B.Sc. (Engg.) in Computer Science and Engineering from Khulna University of Engineering and Technology (KUET), Bangladesh in 2009. His research interest includes Robotics, Artificial Intelligence, Internet Security and various areas of Software Engineering like Requirement Gathering, Requirement Prioritization and Software Security. He has coauthored a good number of research papers published in International Journals and Conference Proceedings. Currently he is working as a researcher of Panacea Research Lab, Dhaka, Bangladesh. He is also a lecturer at Institute of Science, Trade and Technology, Dhaka, Bangladesh and a Project Coordinator of Technocrats BD.

Md. Jahangir Alam received his B.Sc. (Engg.) and M.Sc. in Computer Science and Engineering from Dhaka International University (DIU), Bangladesh. His research interests include Software Engineering and Data Structure. Currently he is working as a Senior Lecturer of Department of Computer Science and Engineering, Dhaka International University, Dhaka, Bangladesh.