Co-Creating Value in Systems Development: A Shift towards Service-Dominant Logic

Jeffry S. Babb, Jr.
jbabb@wtamu.edu
Computer Information & Decision Management,
West Texas A&M University
Canyon, TX 79119

Mark Keith
mjkeith@cba.ua.edu
Management Information Systems,
University of Alabama
Tuscaloosa, AL 35487

Abstract

As agile systems development methods can be viewed from a disruptive technology perspective, what have we learned from the perturbation? Our perspective does not focus on how agility changed existing methods, but rather on what changes in the environment precipitated agile methods and what can be learned about the future of systems development from these changes. In this paper, we re-conceptualize systems development methods from both a service-dominant logic perspective and from the perspective of the co-creation of value between the systems developer and the customer during the systems development life cycle (SDLC). In software development, value co-creation happens in the form of meeting customer needs as well as the creation of new operant resources. We provide a new conceptualization of systems development method selection based on these ideas and illustrate some implications from both the S-DL and Co-creation perspectives. This conceptualization should afford new areas for future research which assumes that agile vs. plan-driven methodology choice is a false dichotomy.

Keywords: Systems Development Methods, Service-Dominant Logic, Co-Creation, Agile.

1. IMPORTANT INFORMATION

The question of systems methodology selection is still of primary concern and the landscape of systems development methodologies seems as confusing as ever. Increasingly, both in practice and in research, we see the "disciplining" of agile methods and the "lightening" or "agile-ization" of traditional plan-based methods such that hybridization is becoming normative. Whereas systems development method selection is typically made with an eye towards better performance, the sum of contributing factors informing method selection are still not entirely understood or agreed upon (Chow and Cao 2008). While the advent and adoption of agile methods has been well-documented (Boehm and Turner 2004), we would argue that the societal and environmental precipitations to agile and lightweight methods are not fully understood.

This paper draws from the marketing theories surrounding service-dominant logic (S-DL) which can be used to explain the disruption of "old"

systems development methods and the evolution and maturation of new "lightweight" methods which favor agility. Changes are afoot both in plan-driven and agile methods, where each are increasingly influencing the other such that orthodoxy in either methodological tradition is the exception rather than the rule. We propose that both ends of the methodological spectrum are responding to the new and emergent demands in the marketplace for co-creation between the software developer and the customer (Prahalad & Ramaswamy, 2004a, 2004b). In this sense, we include, extend, and refine the notion of *task environment* to account for a service-dominant logic and the co-creation of value between developer and customer. From this perspective, we propose that agile methods have been conceptualized under a false dichotomy – agile vs. plan-driven methods – which may have left other more important questions regarding systems development method selection unanswered.

Perhaps the question we should ask is: "what has changed in the environment?" or perhaps "what changed in the software development market?" If methodology choice has traditionally been under the project manager's purview to match the characteristics of the software development method to the constraints of the task environment (Saunders & Scammel, 1986), then something in the environment—or perhaps market—must have changed to necessitate agile methods. We posit that the actual structure of most systems development methodologies has become more fluid as hybrid methods are appearing. This trends towards fluidity isn't accidental; there is now a shift towards new principles, which have evolved in parallel in the marketing discipline (Vargo & Lusch, 2004), governing the way markets work and how value is created between a service provider and a customer. Have these changes in the structuration of markets between customer and service-provider—changes in the sense of Giddens (1984), and perhaps even Kuhn (1996)—contributed to systems development method confusion? What can systems development practitioners and researchers learn from this shift? How can we re-envision the advent of agile and what does it mean to future method selection?

Too often we use traditional cost-based or goods-based approaches to measure project success, such as the degree to which software was developed on time or on budget. However, we see a new perspective where the customer and provider co-create value and where the service provider and the customer mutually shape the meaning of value as they interact. In this sense, the software service provider and the customer are mutually constructing and co-creating value through their interactions. This fresh perspective on method selection, method effectiveness, and on the philosophies informing systems development methods stems from the work on co-creation (Prahalad & Ramaswamy, 2000, 2004a, 2004b, 2004c, 2006) and S-DL (Vargo and Lusch, 2004, 2008) which transpired, seemingly in parallel, with the advent and rise of agile methods. Their new ideas suggest that the structuring of value in systems development is no longer a tit-for-tat stepwise process between releases. Rather, value is now co-created continually between the systems development provider and the customer as they progress through the stages of the systems development life cycle (SDLC). In this sense, the co-creation and structuring of customer (and developer) value needn't wait until the project is done, structuring happens from the moment of inception.

The paper proceeds as follows. First, we present the current thinking in the background literatures on the agile vs. plan-driven methods conundrum. We also discuss how this false dichotomy influences attitudes on method selection. Next, we illustrate new thinking based on S-DL and the co-creation of value and how these concepts change our own views on systems development method selection. In the next section, we postulate a revised conceptualization for systems development methods informed by utilizing an S-DL to focus on the co-creation of value with the customer. We then proceed to illustrate the implications this new conceptualization would have in a few cases. Lastly, we next offer discussion and directions for future research.

2. CURRENT THINKING

The traditional imprint on thinking about systems development methods holds that the outputs of this endeavor are "goods," and increasingly, a service. However, what if a systems development method also, and primarily, facilitated the ongoing co-creation of value in a customer-provider relationship? Under this light, prevailing ideas on the purpose of a systems development methodology, understood on axes related to tolerance of risk

and change, and understood as a choice between plan-driven and agile methods, are a false dichotomy. Rather, it is possible to realize co-created value from a variety of systems development methods. This is so as method selection undertaken from an S-DL and co-creation perspective allows for a customer-provider co-creative team to achieve method fit and service/good personalization. The following sections outline the theories informing this assertion.

Co-creation of Value

Co-creation has arisen with the upset of traditional systems development methods, in favor of agile methods, as a response to environmental changes. Originally offered as a strategy in the marketing field, co-creation holds that a service-/solution-provider and a customer mutually construct and reconstruct value by exploring design, learning, and meaning in a shared partnership (Pralahad & Ramaswamy, 2000, 2004a; Sanders & Stappers, 2008; Payne et al. 2008). When one examines the tenets of agile software development methods, the idea at once becomes familiar. However, the association between trends towards co-creation and changes in software and systems development methods are only slowly coming to light (Kar, 2006; Madsen & Matook, 2010). What co-creation compellingly shares with the development and evolution of agile methods is a change in perspective regarding the role the customer plays in value creation.

Co-creation can be understood in the following scenario. In the days where consumer software was commonly purchased in a brick-and-mortar mode, a software developer, such as Microsoft, would use several techniques to gauge and gather customer input on desirable software features prior to the release of a new iteration of an old product, or as a facet of market analysis for a new product. In any case, the customer was the recipient of a good (the packaged software), which was completed after running a fairly traditional requirements and analysis phase of the SDLC. In this model, the customer wasn't a partner, rather the customer was a semi-passive recipient of goods after the developer had sequestered away when the requirements, analysis and design phases were complete. When the product was released, the customer either received something that was "one-size-fits-all," or was, at the very least, customizable.

Co-creation takes a different tact and has been facilitated by new avenues for customer/provider interaction over the past two decades. Most of these new avenues for interaction involved utilization of the Internet subsequent to its mass commercialization in the 1990s. The Internet, and its applications such as the World Wide Web, has allowed the customer and service-provider to co-create value in the form of unique, tailored, bespoke, and personalized services and experiences (Prahalad & Ramaswamy, 2004c). Rather more compelling, from a systems development perspective, are the payoffs of learning and the loyalty, relationships, and renown that the service-provider enjoys in the co-creation relationship. That the customer provides early signals of value and a means of learning has been well-established in the literature on agile methods (Babb, 2009; Boehm and Turner, 2004). Co-creation can explain changes in the environment related to the means by which information is disseminated: networked, ever-present, contextual, and memetic. For many of the compelling and society-changing technologies enabled by the Internet and World Wide Web – eCommerce, social networks, peer to peer - value is usually co-created with customers as the customer is able to personalize their interactions with the service-provider and the goods/services they consume. This personalization, this tailoring, is a large part of the co-creation proposition.

We see this co-creation phenomenon in agile methods and we see it in hybrid methods: the degree to which the demand for personalization – not necessarily specialization, or customization – influences customer relations and influences method selection. If agile is about managing change, then perhaps the growing acculturation to co-creative pathways of customer/provider relations is what is driving this change. We speak of Apple, Netflix, eBay, Amazon, and Facebook as being respective technology leaders in so far as they each afford the customer simple and personalized choices for interaction. Thus, if value is increasingly co-created within the provider/customer relationship, rather than being created entirely by the provider, then systems development methods which accommodate this bilateral flow will allow the service-provider to adapt and foster the myriad relationships made possible by adopting systems development methods which are co-creative in nature.

The Traditional View of the Market

The Co-creation of value and agile methods share the view that the customer is not merely a passive target for transactions, but rather that the customer is an integral part of the processes of design, planning, and strategy. As with co-creation, agile methods allow a customer representative the opportunity to craft their own product, and their own experience, by engaging the agile partnership.

The relationship between the firm and the customer in traditional markets is represented in Figure 1 (Prahalad & Ramaswamy, 2004a). In this conceptualization of a market, the firm utilizes the market in order to extract value from the customer. In turn, the customer extracts value from the receipt of goods or services. In this conceptualization, there is strong directionality from the firm, making the customers' role very lopsided and unequal. In this traditional market scenario, the firm believes it is the sole creator of value in providing optimized and undifferentiated services. Furthermore, in this mode of operation the firm is the sole source of expertise and acts as exclusive arbiter of value, optimization, and cost reduction. As a result, IT project success is viewed in terms of being on-time and on-budget. As Prahalad and Ramaswamy (2004a) put it:

As long as firms believe that the market can be separated from the value creation process, firms in search of sources of value will have no choice but to squeeze as much costs from their "value chain" activities as possible. Meanwhile, globalization, deregulation, outsourcing, and the convergence of industries and technologies are making it much harder for managers to differentiate their offerings. Products and services are facing commoditization as never before. Companies can certainly not escape being super-efficient. However, if consumers do not see any differentiation they will buy smart and cheap. The result is the "Walmartization" of everything, from clothes to DVD players.

We can see this phenomenon in traditional systems development methods as well. While well-meaning, process optimization approaches, such as the Capability Maturity Model Integrated, or the Rational Unified Process, work at systems of efficiency and cost-savings in order to produce the same type of product reliably. While these traditional plan-driven approaches provide a reasonable hedge against risk, the requirements process tends to force customer needs into templates, such as those provided by the UML and by the CASE tools that support them. This is a mode of customer accommodation Pralahad and Ramaswamy (2008) liken unto *customization* – you can get a variant of the product, but one that is not truly personalized.

The Co-creation Approach

The Co-creation approach completely re-envisions customers as being a partner in the creation of value. Many of those with the highest reputation in the marketplace seem to have availed themselves of a personalizable relationship with their customers at the earliest possible moment via the Internet. Prahalad and Ramaswamy (2004c) and others (Kazman and Chen, 2009; Payne, et al., 2008) have each maintained that that co-creation goes beyond co-designing products and services; it establishes the mode under which the market will operate – a mode of equality. Such empowerment is evident in the open source software community, in crowd-sourcing, in Amazon's *Mechanical Turk* service, and in myriad other channel's for customer empowerment.

Thus, the attraction of a co-creative marketplace is in the value created, and in the values informing the interactions between firm and customer in a co-creative relationship. Figure 4 presents Prahalad and Ramaswamy's (2004c) building blocks to facilitate co-creative interactions between the firm and the customer. These principles read as though they are annotations from the Agile Manifesto (Fowler and Highsmith, 2001).

We can see the parallels between agility and co-creation, in terms of the principles each espouse, by focusing on Figure 4. Each of these "building blocks" are evident in both the manifesto and principles for agility, but also in the descriptions of many of agile methods (Boehm and Turner, 2004). Table 1 presents a comparison between the building blocks of developing a co-creative customer relationship and a generalization of similar principles from agile methods (particularly eXreme Programming). The similarities are striking in that realizations regarding changing markets, presented in 2000 by Prahalad and Ramaswamy

from a marketing perspective, were also described, in parallel, from a software development methods perspective (Fowler and Highsmith, 2001).

3. SERVICE-DOMINANT LOGIC

As the research and literature on agile methodologies and co-creation has progressed, other related concepts have evolved in the marketing literature. In particular, the concept of S-DL (Vargo and Lusch, 2004, 2008) refers to the shift in philosophy from a goods-dominant to service-dominant logic. This means that both goods and services (e.g. goods enhancement or the services offered by health, government, and education industries) should be viewed in terms of the services they provide.

From this perspective, there are two types of resources in a market or organization: *operant* and *operand*. Operand resources are those which must be acted upon in order to be beneficial. For example, operand resources in IT projects would refer to existing hardware and software which the organization owns and which can be useful in a new IT project. However, these operand resources are only useful if the project team has the appropriate operant resources—those which act on behalf of other resources to create value. In other words, the implicit knowledge held by systems development team members is necessary in order to produce the benefits of existing software components or hardware. Essentially, programmers use their operant resources (i.e. their knowledge and skills) to provide services to the IT project just as any provider does for its customers. We argue that additionally, in the co-creation model, a programmer also acts as an operand resource which is "acted upon" by the customer, who is an operant resource. The customer interacts with the programmer who is like an empty canvas waiting to be turned into personalized value. However, in the co-creation mode, the expertise of the programmer assists the customer by enabling a greater understanding of the palette of options available for the canvas.

There are ten foundational premises of S-DL (Vargo & Lusch 2008). We highlight the applicability of just three of those ten here (for brevity). First, service is the fundamental basis of exchange. In other words, the value produced by IT systems is viewed in terms of the service it provides to the customer—not the cost of the IT project or its on-time performance. Second (but fourth in Vargo and Lusch's [2008] list), operant resources are the fundamental source of competitive advantage. This is especially apparent in systems development. Any IT project team can get access to the hardware and software necessary to produce new IT systems. However, it is the ability of an IT project team to manipulate those operand resources which provides value to the customer. A project team's ability to actualize those resources to meet customer preference and needs is the primary source of competitive advantage over other project teams. Third (but sixth in Vargo and Lusch's [2008] list), in a service-oriented view of markets, the customer is a co-creator of value. In this case, where the customer also plays the role of the operant resource affecting an operand IT project team, the customer achieves competitive advantage in their own market by way of the degree to which they can use and manipulate an IT project team in order to produce a high-value IT system.

After understanding the foundational premises of S-DL, it is easy to view the shift toward agile and hybrid methodologies as a being part of a larger shift toward the service-oriented paradigm taking place everywhere, including the systems development market. While it would be an over-simplification to state that the sole reason for agile methodologies is to facilitate greater customer co-creation, agile methods are naturally well-suited to involve customers to a greater degree and to induce their input into the creative process more completely throughout the project life cycle.

Similarly, many organizations are not well-suited for agile methodologies, yet are searching for ways to adopt agile principles in their plan-driven techniques and are forming "hybrid" approaches. While there are a great many other factors (often related to project risk) which influence the methodology selection decision, we argue that this shift toward S-DL and co-creation is playing a large role whether directly or indirectly.

In summary, the S-DL view (Vargo & Lusch 2008) emerged in parallel to the co-creation concepts (Prahalad & Ramaswamy 2004a) in systems development, yet the two complement and inform each other. In the next section, we outline how these two conceptualizations affect extant theoretical models of methodology selection.

4. RE-CONCEPTUALIZED THEORETICAL MODEL OF METHODOLOGY SELECTION

Traditionally, methodology selection involves realizing a fit between the characteristics and assumptions of the systems development methodology and the degree of risk and uncertainty in the task environment (Barki et al. 2001). Once these risks were assuaged by the discipline of the systems development method, customer requirements would be met and customer value realized. However, Figure 6 inserts concepts of S-DL and Co-creation into the equation to suggest that customer involvement in design should also influence the fit that a method presents in a given problem domain and within a given set of task environment risks, uncertainties, and constraints. In other words, rather than switching methodologies, PMs may simply need to facilitate better co-creation into their process.

As we conceptualize how this new co-creative relationship will transpire between systems developers and customers, it is important to bear in mind that each party plays an equal-yet-distinct role in the partnership. This can be illustrated by Payne et al. (2008) in distinguishing between the role each party plays, and is also supported by other research into collaborative design between customers and software developers (Lee, 2007; Babb 2009). As it would be in the case of action research, participatory design, and other similar arrangements where dissimilar partners collaborate, the co-creative process is not necessarily one where both partners share the same expertise or concerns. According to Payne et al. (2008), the separation of concerns between the co-creative partners can be seen as a process ascribed to the customer, a process ascribed to the supplier, and the process of the encounters between them (Figure 7).

The encounter processes depicted in Figure 7 represent the important synergy made possible between the customer's processes and the supplier's processes during co-creation. Of particular interest is the inclusion of learning in the Payne et al. (2008) model. This model can be extended to incorporate the concerns, assumptions, and mechanics of systems development method selection in order to understand how this method facilitates the co-creation of value. Therefore, we must assume that these two parties, as they mutually construct meaning and value in their co-creation, each bring a unique perspective to the partnership. We can also conceive of these encounters between system developer and customer as an ongoing dialog, one in which value is created through exchanges of expertise and knowledge.

Re-conceptualizing Method Selection

We re-conceptualize the method selection process to include the co-creative concepts of S-DL and Co-creation. In Figure 8, we propose that systems development method selection involves choosing a technique which both fits the risk and uncertainty inherent in the task environment as well as fosters the necessary value co-creation with the customer—which is ultimately what determines project success, rather than time and cost.

Our model retains the importance of balancing the innate characteristics of a methodology with uncertainty and risk tolerance in the task environment. However, a co-creative partnership suggests that both the customer and systems developers stand to benefit in sharing the concerns of method selection, use, and evolution. In our conceptual model, the co-creation of value between customer and systems developer results in pathways of learning which allow the systems developer to realize an optimum methodological fit while the customer realizes increasing personalization. The value-creation encounters between customer and systems developer, facilitated by the chosen software development method, create new and unique operant resources for each party. For the systems developer, the operant resource is the new knowledge gained from their creative interactions with the customer. For the customer, the operant resource is the new knowledge concerning IT system capabilities and ideas for new systems potential. The co-created value to the developer is an optimized and tailored systems development method; and, the co-created value to the customer is the personalized IT system.

When we consider the Prahalad and Ramaswamy (2004a) building-blocks for establishing a co-creative relationship – dialog, access, risk-benefit, and transparency, it becomes easier to see why the demand for agile methods has arisen. Agile and hybrid methods are best-suited to facilitating a co-creative relationship between customer and systems developer. This is not to say that in all cases the customer-

provider relationship should be, or even can be, co-creative. However, in cases where co-creation is desired and/or warranted, agile methods appear to be the best fit thus far.

5. DISCUSSION AND IMPLICATIONS

We have proposed in this research that the evolution of agile and hybrid systems development methodologies are examples of a larger shift toward S-DL. Therefore, one of the primary implications is that project managers who are considering a switch to agile methodologies may need to pause and consider whether the cause of their current project failures are the result of traditional risk factors (e.g. lack of top management support or lack of operand resources such as knowledge/capabilities, etc.) or the result of poor customer co-creation when co-creation was in fact warranted. It is possible to involve customers to a greater degree using some form of hybrid methodology such as Boehm and Turner's *Incremental Commitment Model* (2004) or a service-oriented systems development technique (Keith et al., 2009) rather than forcing a more drastic change to a completely agile method. In addition, the S-DL perspective highlights the importance of operant resources as the primary source of competitive advantage over other systems developers. By choosing to outsource portions of a project, the PM is losing opportunities to develop new operant resources through customer co-creation.

Lastly, this research highlights the importance of valuing an IT project based on its ability to meet customer needs rather than create a product within a given time and cost constraint. Examples of large IT projects which are completed only to find that it doesn't meet the customer's needs are easy to find.

6. CONCLUSION

While this paper proposes that S-DL and Co-creation are reflections of changes in the task environment and market which demand new thinking in systems development methods, it is hard not to realize that agile methods are most in step with S-DL and Co-creation. If the same disruptive technology which prompted a revisit of our conceptualization of markets also prompted a revisit of systems development methods, then agile methods represents a swing in a pendulum in response to paradigmatic change. Eventually, as we are reminded by Kuhn (1996), stability in this new paradigm should arrive eventually. However, during the time of flux, as we have experienced for over a decade, practitioners and scholars of systems development methods would do well to understand changes in their own discipline – the advent of agile methods – through the experiences and wisdom of another discipline: marketing. Our re-conceptualization of systems development method selection accounts for the new and emerging relationship between customer and provider outlined in S-DL and Co-creation. We feel that this conceptualization opens up opportunities to study method selection and to study agile methods adoption and use in a new and useful light.

7. REFERENCES

Babb, J. (2009): Towards a Reflective-Agile Learning Model and Method In The Case Of Small-Shop Software Development: Evidence from an Action Research Study, PhD Dissertation, Virginia Commonwealth University, Richmond, VA.

Bardhan, I.R., Demirkan, H., Kannan, P.K., Kauffman, R.J., and Sougstad, R. (2010). An Interdisciplinary Perspective on IT Services Management and Service Science, *Journal of Management Information Systems*, 26, (4), 13-64.

Barki, H., Rivard, S., and Talbot, J. (2001). An integrative contingency model of software project risk management, *Journal of Management Information Systems*, 17, (4), 37-69.

Boehm, B. and Tuner, R. (2004). *Balancing Agility and Discipline: A Guide for the Perplexed*, Addison-Wesley, Boston. pp. 304.

Chow, T. and Cao, D. B. (2008). A survey study of critical success factors in agile software projects, *Journal of Systems and Software*, 81, (6), 961-971.

Dong, B., Evans, K.R., and Zou, S. The effects of customer participation in co-created service recovery, *Journal of the Academy of Marketing Science*, 36, 123-137.

Fowler, M. and Highsmith, J. (2001). The Agile Manifesto, *IEEE Software Development*, 9, (8), 28-35.

Giddens, A. (1984). *The Constitution of Society*, University of California Press, Berkeley, CA.

Kar, N.J. (2006). Adopting Agile Methodologies of Software Development: Agile Methodologies are Challenging Established Paradigms of Software Development, *Infosys SETLabs Briefings*, 4, (1), 3-10

Kazman, R. and Chen, H. (2009). The Metropolis Model: A New Logic for Development of Crowd-sourced Systems, *Communications of the ACM*, 52, (7), 76-84.

Keith, et al. (2009). Service-Oriented Software Development. *AMCIS 2009 Proceedings*. Paper 100.

Kuhn, T.S. (1996). *The structure of scientific revolutions (2nd Ed.)*, University of Chicago Press, Chicago.

Lee, A. S. (2007). Action is Artifact. In N. Kock, *Information Systems Action Research* (pp. 42-60). New York: Springer.

Madsen, S. & Matook, S. (2010). Conceptualizing Interpersonal Relationships in Agile IS Development, *Proceedings of the International Conference on Information Systems (ICIS2010)*, December 12-15, Saint Louis, MO, USA.

Payne, A.F., Storbacka, J., & Frow, P. (2008). Managing the co-creation of value, *Journal of the Academy of Marketing Science*, 36, 83-96.

Prahalad, C.K. and Ramaswamy, V. (2000). Co-opting Customer Competence, *Harvard Business Review*, 78, (1), 79-88.

Prahalad, C.K. and Ramaswamy, V. (2004a). Co-Creation Experiences: The Next Practice in Value Creation. Journal of Interactive Marketing, 18, 3, pp.14.

Prahalad, C.K. & Ramaswamy, V. (2004b). Co-creating unique value with customers, *Strategy & Leadership*, 32, (3), 4-9.

Prahalad, C.K. and Ramaswamy, V. (2004c). *The Future of Competition: Co-creating Unique Value with Customers*, Harvard Business School Publishing, Boston.

Ramaswamy, V. (2006). Co-Creating Experiences of Value with Customers, *Infosys SETLabs Briefings*, 4, (1), 25-36.

Sanders, E.B. & Stappers, P.J. (2008). Co-creation and the new landscapes of design, *CoDesign*, 4, (1), 5-18.

Saunders, C.S. & Scammel, R.W. (1986). Organizational power and the information services department: a reexamination. Communications of the ACM, 29, (2), 142-147.

Spohrer, J., Vargo, S.L., Caswell, N., and Maglio, P. (2008). The Service System is the Basic Abstraction of Service Science, *Proceedings of the 41st Hawaii International Conference on System Sciences*, January 7-10, Waikoloa, Big Island, Hawaii, USA.

Vargo, S.L. and Lusch R.F. (2004). Evolving to a New Dominant Logic for Marketing, *Journal of Marketing*, 68, 1-17.

Vargo, S.L. and Lusch R.F. (2008). Service-Dominant Logic: Continuing the Evolution, *Journal of the Academy of Marketing Science*, 36, 1-10.

APPENDIX

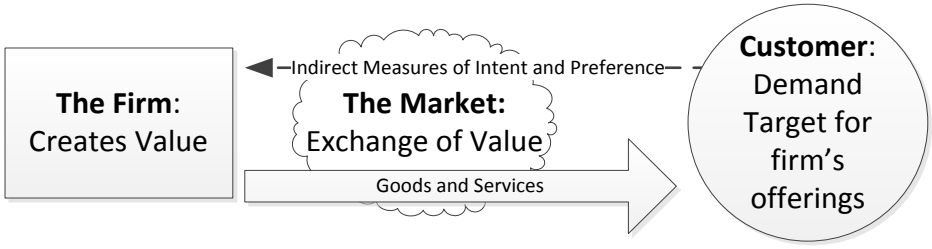

Figure 1 Traditional Concept of a Market (Prahalad and Ramaswamy, 2004)

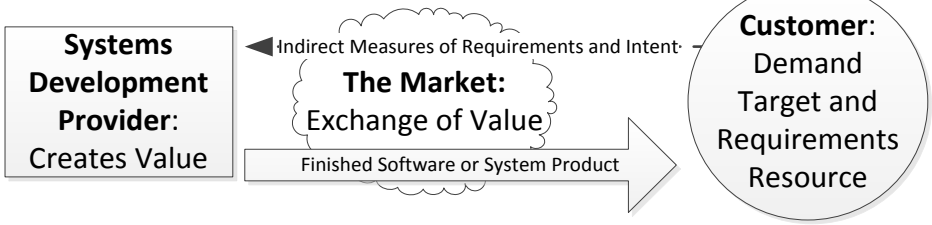

Figure 2 The Function of Traditional Software Development Methods

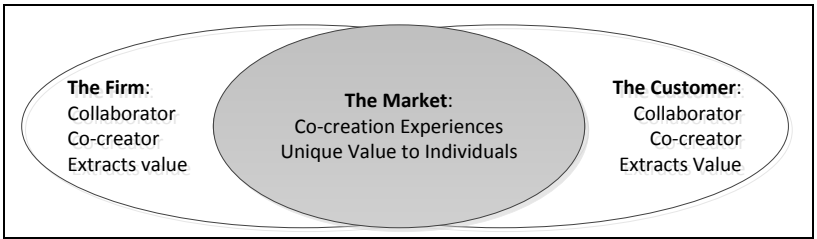

Figure 3 The Function of Traditional Software Development Methods (adapted from Pralahad and Ramaswamy, 2004c)

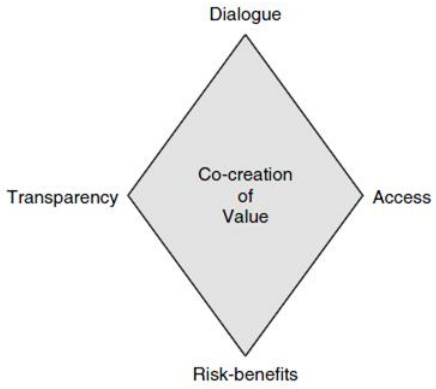

Figure 4 Building Blocks of Interactions for Co-creation of Value (Prahalad and Ramaswamy, 2004c)

	Co-creation	**Agile methods**
Dialogue	➢ Markets are a set of conversations between equal partners ➢ Joint Problem-Solvers ➢ Rules of Engagement for equality	➢ Design occurs between a client/developer partnership ➢ Regular interactions and releases of working software solves problems ➢ Regular meetings for equality
Access	➢ Firms provide reliable access in order to avoid information asymmetry	➢ Customer is a team member and is afforded regular access to discourage information asymmetry
Risk-benefits	➢ Firms provide reliable access in order to avoid information in order to foster a learning and empowering environment – Google: do no harm	➢ Both developer and customer are informed by participating in the process and each knows consequence and costs of change
Transparency	➢ Firm maintains an openness to customers in order to facilitate a dialog which creates value	➢ The customer is aware of release iterations and project velocity.

Table 1. A Sample Table

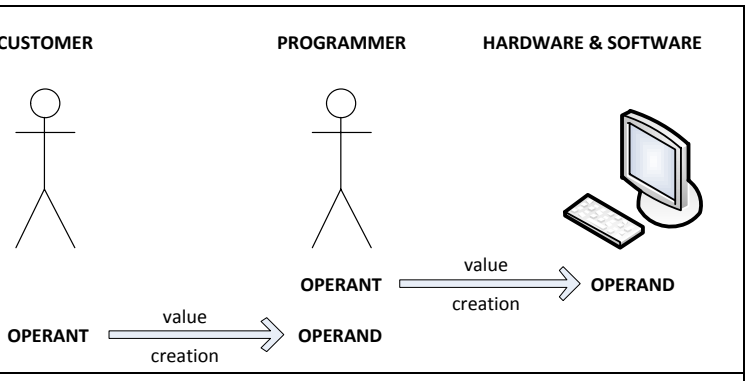

Figure 5 Types of Value Creation in Systems Development

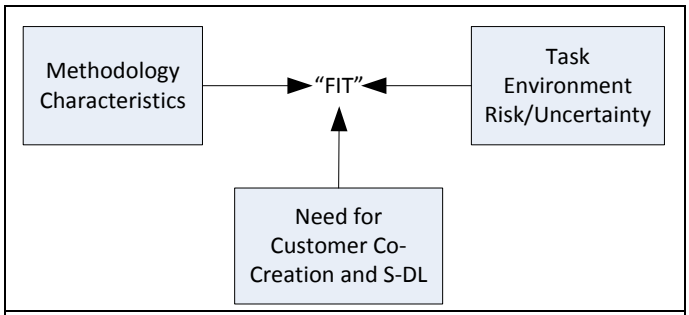

Figure 6 Nascent Theoretical Model of Methodology Selection Based on Co-Creation and S-DL

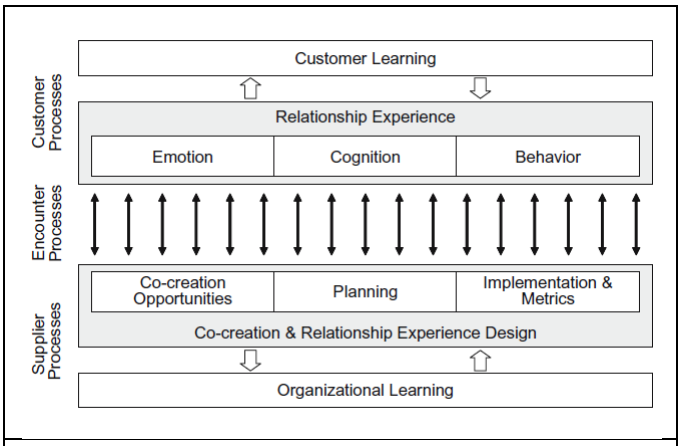

Figure 7 Conceptual Model for Value Co-creation (Payne et al., 2008)

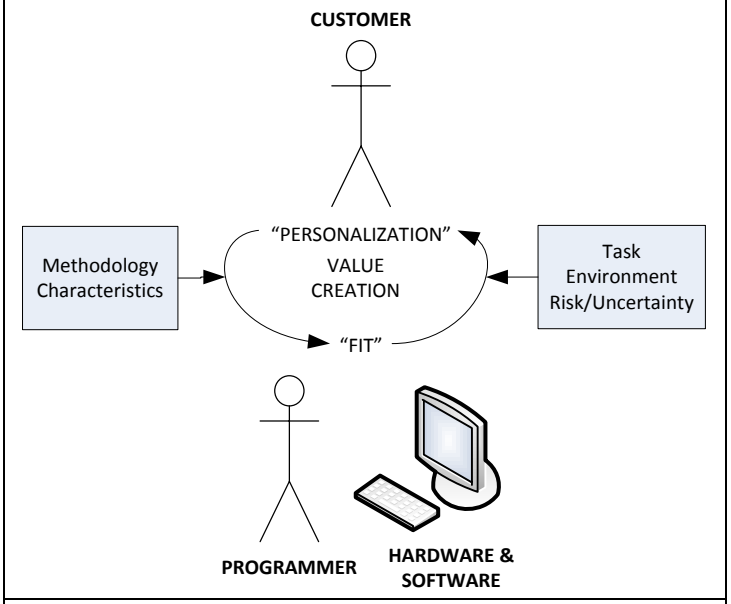

Figure 8 Re-conceptualized Theoretical Model of Methodology Selection