Asia Pacific Journal of Information Systems
Vol. 24, No. 3, September 2014 http://dx.doi.org/10.14329/apjis.2014.24.3.371

Applications Development: a Value-Laden Process

Christobal Cayaba*, Zelinna Pablo**

A long-drawn out debate in the field of technology is whether it is value-neutral or value-laden. While some have argued that this debate has been resolved given the increasingly accepted view that technology is socially constructed, this is still not reflected in mainstream research which still assumes that technology is neutral. What is clear is that both views tend to be linked to studies that primarily focus on explicit forms of technology such as technological designs and their usage. These studies, though significant, may be limited in terms of an underemphasis on the process by which these technologies emerge, a process that requires decision-making activities made by different stakeholders and thus involves value judgments. In order to understand the extent of value-neutrality or value-ladenness of technologies, therefore, it is important to examine not only the final outcomes but also the process involved in technological development (including the artifacts created and used). In this study, we explored how values may be embedded in a specific IT application, and in cases of conflict of values, how they are prioritized. We did this in the context of applications development through an examination of the stages ranging from requirements analysis to coding to testing and deployment.

Keywords : Value-Neutrality Versus Value-Ladenness Debate on Technology, Discourse Analysis, Program Codes

---

* First and Corresponding Author, PhD Candidate, De La Salle University
** Sessional Lecturer, Department of Management and Marketing, The University of Melbourne

Ⅰ. Introduction

In this paper, we explored the extent of value-neutrality or value-ladenness of technology by examining the decision-making activities in the applications development context. There is a large body of work that suggests technology is static and objective, thus value-neutral [Carey, 1992; Sundstrom, 1998; Bilbao-Osorio *et al.*, 2014; Santori, 2014]. On the other hand, there is also a growing body of work that suggests that technology is socially constructed, which means that the development of technologies is influenced by the judgments by different stakeholders [Orlikowski, 2000; Klein and Kleinman, 2002]. In the process, choices and decisions made reflect biases of these stakeholders, and therefore deemed as value-laden [Postman, 1993; Friedman and Nissenbaum, 1996].

Attempts have been made to study the value-neutrality or value-ladenness of technology. However, Janz [2005] explains that a major gap in these efforts is due to the fact that most take a technology perspective, emphasizing new technological innovations and collaborative technologies that emphasize "explicit" (tangible ones such as user interface designs and actual product use) knowledge. This results in the lack of an overarching perspective that studies the major activities involving decisions made in the process. Further studies on value-neutrality or value-ladenness and how these might apply to technology are therefore significant. The research question we seek to address is therefore: ***what does the study of the evolution of decisions involved during the technology development process reveal about the extent of (value-neutrality or) value-ladenness of technology, specifically application programs?***

To answer this question, we examined decision-making activities within the context of the process of applications development in a US-based organization with an offshore unit. We employed a qualitative case study, primarily because our research question calls for studying the interactions and discursive activities among human beings. We gathered data through document reviews as well as observations involving key entities such as the developers, program codes, technical design, online discussions, emails, and chats. We then analyzed the data using discourse analysis techniques.

This study has at least three contributions to theory, practice, and method. First, addressing the research question helps us understand the value-neutrality vs value-ladenness debate with respect to technology. Recent literature still assumes that technology is neutral. There is the belief that this debate has been resolved given the increasingly accepted view that technology is socially constructed; however this is still not reflected in mainstream research, and therefore social constructionist approaches need to be further strengthened. This body of work also contributes to empirical insights as to how technologies are socially constructed, by examining the micro-processes that go into applications development. Second, we also contribute to practice by presenting the implications based on our findings allowing professionals from the applications development domain to become more aware of their vulnerabilities that may be shaped by internal (self) and external (environment) factors, which then become possible areas for improvements. Third, we make a methodological contribution through the use of discourse analysis on non-linguistic contexts, specifically in a technical domain involving program codes, is less explored and therefore, the views we can get from this approach can provide significant knowledge in future relevant works.

The rest of the paper is structured as follows. The next section presents literature that broadly frames the debate on technology and values. The following section discusses the methodology, including a presentation what a case study is, and discussions for selected data gathering and analysis techniques. We then present our findings and end with a section containing the conclusion, contributions, and future work.

Ⅱ. Debates on Technology and Values

2.1 Social Constructionism and Technology

The question as to whether technology is value-neutral or not has its roots from early interdisciplinary works involving science, technology, and society. One body of work assumes that technology is value-neutral. Feenberg [1991], for example, conceives technology as neutral and without valuative content on its own because technologies are seen as tools that only stand ready to serve the purposes of their users. Sundstrom [1998] further explores technology as a cognitive possibility abstracted from each specific social context and argued that it is value-neutral before an action has been made on a particular technological tool. Carey [1992] believes that technology such as "electronics is neither the arrival of apocalypse nor the dispensation of grace" and that it is used for communication and transportation over space and nothing more. Recent literature still assumes that technology is neutral [Alexiadis and Cole, 2004; Bilbao-Osorio *et al.*, 2014; Koops, 2014; NIST, 2014; Santori, 2014].

However, there is a growing body of work that suggests that technology in general is socially constructed. This suggests that "technology is an open process that can produce different outcomes depending on the social circumstances of development" [Klein and Kleinman, 2002]. The concept of social constructionism has been applied to various technologies such as MacKenzie's [1990] account of the negotiations over missile guidance systems and through Bijker's [1987] significant work on early bicycles. It has also been argued that in the development of application programs specifically, multiple groups embody specific interpretations of these artifacts, therefore there is the need for negotiation among different social groups with different definitions of a working product [Klein and Kleinman, 2002]. Studies on these artifacts reveal that these are "products" (in quotes because products here do not refer to objective and static artifacts) of social constructionism mostly via design and use. DeSanctis and Poole [1994] note that "structures are found in institutions such as reporting hierarchies, organizational knowledge, and standard operating procedures. Designers incorporate some of these structures into the technology… Once complete, the technology presents an array of social structures for possible use in interpersonal interaction, including rules (e.g., voting procedures) and resources (e.g., stored data, public display screens)…"

Design and innovation of technology, as cogently argued by Williams and Edge [1996], are not linear, rational, straightforward, or supplier-driven. Instead, they are non-linear, not fully rational, complex, and influenced by many parties such as suppliers, users, funding bodies, and even households. They examine how the technological design can have diversified and socially-laden decision points. Hence, they argue that technology has been permeated primarily by socio-economic factors. For instance, the design of application programs ceases not because the final product works in an objective sense but because the deciding social groups accepts that the set of features in the design works for them [Klein and Kleinman, 2002].

The use of technology in context-specific ways gives rise to the development of subjective views of technology. Fountain [2001] sets the distinction between [objective] technological artifacts and their subjective enactment or use. Orlikowski [2000] suggests that users may use technology in ways not anticipated by the developers. For example, a tax preparation application may be used to simply print out blank forms but still complete tax returns manually, or use it to learn how to design application program interfaces. She further notes that users may also choose to use technologies because they have become more knowledgeable about their use through class training or by watching how another colleague uses such. Variations in tasks can also alter and adjust differently how one uses technology. Orlikowski [2001] demonstrates this through a case involving the deployment of the Notes application program, which was developed by the Lotus Development Corporation. The product is an objective artifact with a fixed set of functionalities, among them text editing and document management. However, the application program became a medium for the creation of different structures, among them are collaboration, collective problem-solving, individual productivity, and support.

Because technology is socially constructed, it reflects values in choices and decisions by different stakeholders at different points of the life cycle of technology. These “choices” and “decisions” can be referred to as biases, which in turn, result in different outcomes or effects, mostly unintended such as development of optimistic/pessimistic views on technology, artifacts “behaving independently”, and features result in multiple unintended consequences. These are primary justifications why technology is therefore deemed value-laden.

2.2 Value-Ladenness of Technology

Postman [1993] argues that “Embedded in every tool is an ideological bias, a predisposition to construct the world as one thing rather than another, to value one thing over another, to amplify one sense or skill or attitude more loudly than another…New technologies alter the structure of our interests: the things we think about. They alter the character of our symbols: the things we think with. And they alter the nature of community: the arena in which thoughts develop” (p. 13). In the succeeding subsections, we dissect this belief and present the arguments how technology is seen as being value-laden.

2.2.1 Embedded Values through Biases

Friedman and Nissenbaum [1996] argue that technology is shaped by three types of biases. First, pre-existing biases emerge from the values and attitudes of the designers of the applications. This may either be at the individual or societal level, and may also be intentional or unintentional. An individual bias results directly from the values of those who have direct and significant influence to the design. For example, designers of technical systems and devices hold some variables fixed as background assumptions that extend to becoming explicit functional ends [Flanagan *et al*., 2008]. An organizational/societal bias comes from the broader cultural context in which the systems are being developed. For example, the general view of overall gender biases that consequently lead to the preconceived notion that computer games should be designed based on the preferences of male users [Friedman and Nissenbaum, 1996].

The second type of bias is linked to technical limitations and factors [Friedman and Nissenbaum, 1996]. Authors argue that although they might not be intentional, these technical constraints may still lead to value-laden designs. For instance, smaller screen size of mobile devices allow those that are displayed first at an advantage, and that when applications development techniques may not allow for further security, this results in privacy breach [Brey, 2009].

It is claimed that if the designers are aware of the ways in which values may be embedded in technology such as computer applications, they may be able to anticipate future uses of these applications and may be able to appropriately design for specific contexts. However, it is also inevitable that these applications have different unanticipated implications [Brey, 2009], which is linked to the third type of bias: emergent bias. Emergent bias comes about when the intended design is applied in a social context that triggers unintended outcomes. An example of this is the case of text-based Automated Teller Machines (ATMs) being installed in a community where there is low literacy rate with unpredictable dffects [Friedman and Nissenbaum, 1996]. A second example involves the Internet, an inherently democratic technology, being suppressed by human and other technological interventions such as by filtering access to select groups of individuals [Johnson, 1997], an outcome being the technology able to reflect both democratic and non-democratic potentialities [Camp, 1999].

Brey [2009] further proposes two other types of bias: user and information biases. User biases are those that discriminate against other users of these applications, and information biases are those that discriminate against other information that exist in the applications. Some examples include selective access rights to users and selective information content.

2.2.2 Unintended Outcomes from Biases

One group of studies suggests that embedded values have positive effects on users. Those who subscribe to this optimistic view believe technology brings a better life and hope for this world and the human race [Brey, 2009]. Moreover, Kahn and Wiener [1967] present technology as a source of happiness, a promise of utopia, which also brings progress and solutions to problems. Furthermore, Barlatier *et al*. [2006] believe that technological tools are seen as enablers for people in that they encourage people to give their opinions allowing them to feel more free and less observed, and even inhibit fears of expressing oneself in public, thereby increasing participation and commitment.

A second group of studies suggests that embedded values generally work to the disadvantage of the user, which may become the source for more problems in society as a means for dehumanization [Mumford, 1967]. Friedman and Nissenbaum [1997] argue that the limited capabilities offered by technology make the performance of tasks less desirable. Also, the imposition of the coded operational logic within these technologies makes the user experience less autonomous and limits creativity and choice, making the users become helplessly dependent on them.

A third group focuses on how embedded values can be mobilized even without human users. Some technological artifacts may even behave autonomously and are capable of exhibiting behaviors on their own such as robots and artificial agents [Brey, 2009]. These artifacts therefore create consequences, oftentimes social consequences that reflect embedded values [Latour, 1992].

A fourth group has sought to explore how embedded values can be explored independently of the actual use and consequences of these technologies [Brey, 2009]. In such cases, embedded values become distant because the views of the applications developers, designers and other professionals who work on these technologies are "removed" from the views of the users. Thus, there are many technology-related practices that are morally opaque because they operate in difficult to understand ways for laypersons. This consequently makes these embedded values also hidden from the view of the average users.

Another group of studies suggests that there are no inherent embedded values in technological artifacts but they do argue that there are multiple ways in which these may be used and each use has its own sets of consequences [Brey, 2009]. For example, a laptop does not embody specific values, but while it can be envisioned to assist individuals on certain tasks such as emails, its design materials can also be used to break objects, or thought out as a tool to even kill someone.

A final group of studies focuses on embedded values related to specific issues. One example is studies that emphasize privacy and trust. For instance, Friedman *et al*. [2006] reported that the installation of camera outside the office and the display hung inside the office was meant to increase the emotional well-being, physical health, and creativity in workers, however, it was identified that those individuals being filmed by the camera felt that the system violated their privacy.

2.3 An Overemphasis on Outcomes and Underemphasis on Processes

There are a number of studies that explore the value-ladenness of IT artifacts. For example, Singh *et al*.'s [2003] study on American-based and Chinese-based websites focused on how these websites are shaped by cultural values. These, along with other studies [Pablo, 2007; Seneviratne and Monroy-Hernandez, 2010] are important but can be built on, primarily because they have so far focused on the cultural ladenness of IT products or outcomes. We argue here that we should examine and understand not only the value-ladenness of final outcomes but also the process how these values came to be embedded in the first place.

In Saarni *et al*. [2008] study, it was found that selection of methods and assessment topics is essentially a value-laden decision. Specifically, they believe that health technology assessment (HTA) is not value-free, and therefore values must be made explicit in order to increase the international transferability and relevance of the HTA policies. However, their study dwelt on the interwoven nature of ethics and HTA such as medical and safety issues and not on the nature of an in-depth look on the actual value-ladenness of technology. Nonetheless, Saarni *et al*. [2008] believe that the study on the consequences or the final outcome of technological design or use alone is not enough but should also consider the whole HTA process, that is, from start to finish.

Schuurman [2007] also theorizes whether there are any diffrences in studying and teaching computer science as a consequence of our beliefs. In his inquiry, given two programmers, one a Christian and the other non-Christian, who will use the same programming language, compiler, operating system, and software engineering techniques, he asks whether end users are able to discern the religious convictions of the programmers. Such studies point to the necessity of looking beyond the final outcomes by examining the process by which the two programmers employ their decision-making and prioritization tasks. It is in such inquiry that we are able to assess what and how values may have been embedded or not through the decisions they make and the artifacts that are used and created in the process. The process of technological development is therefore of significance.

Franssen *et al*. [2013] define this technological process generally in several steps as follows. First is the definition of functional requirements, where felt required because customers only focus on one to two features and are unable to fully articulate all functionality they really desire. These functional requirements are then translated into the design specifications, which detail the specific steps in order to create a blueprint. Blueprints are used for prototyping and testing processes that require several interactions between various stakeholders. They further note that it is an important modern development that a complete life cycle of a technological artifact is becoming a design engineer's concern. Others consider blueprints as end results, while others believe prototypes are end results as well. However, the development of such technological artifacts goes through a continuous process of development/evolution, to the point of not being able to identify what the end product is.

Notably, another study by Amiel and Reeves [2008] argues that the current role of educational technologies fails in improving the educational practices and outcomes. They suggest that this is because current research focuses on the actual use of technologies as current views mostly present technology as tools. They add that more democratic research practices involving recognition of technology as a system, possibly directed by values, beyond its tools must be pursued. They further argue that technology, as a process, implies value-ladenness possibilities and has implications how educational technologists can further improve research. Once it is fully recognized as a process, the aims/ends of these technologies can then come to the foreground.

Ⅲ. Research Question

As presented earlier, a growing body of work assumes that technology is value-laden. This has led researchers to explore how these technologies are designed; however, such studies have been conducted by primarily looking into final outcome or end results such as user interfaces of websites. We argue, therefore, that there has been an overemphasis on end results of technology resulting in an under-emphasis on the decisions involved during the actual development phases of these technologies. We can further understand how values may have been embedded through scrutiny of the process of requirements analysis to coding to testing and deployment stages. In the process, we can look at initial designs, discussions of team membersas well as analysis on the basic constructs of technology such as program codes.

We therefore framed our research question as:

What does the study of the evolution of decisions involved during the technology development process reveal about the extent of (value-neutrality or) value-ladenness of technology, specifically application programs?

Ⅳ. Methodology

To address this research question, we used a case study and discourse analytic techniques.

4.1 Case Study

The research approach that we employed is a case study. The case study is a research strategy that aims to conduct a systematic investigation of complex dynamics involved within a bounded setting [Eisenhardt, 1989]. It may involve either single or multiple cases, as well one or numerous levels of analysis [Yin, 2003]. Most case studies combine several data gathering techniques such as observations, interviews, and document reviews [Eisenhardt, 1989].

The case study as an approach is sometimes criticized for a number of reasons. Lee [1989] presented an overview of methodological problems involving cases in Management of Information Systems (MIS) studies. One challenge raised by researchers is how to make controlled observations and deductions in such real-life settings. In the case of information systems, studying a phenomenon in its real-world setting impedes laboratory controls used in laboratory experiments. There are also more variables than data points, which make statistical controls and experiments less applicable. IS cases are also seen to be difficult to capture using mathematical approaches.

Lee [1989] addresses these points in a number of ways. He argues that by scanning the empirical materials within the case study, wecan look for the presence of natural controls or factors that may lead to testable predictions. He also clarified that it is mathematics that is a subset of formal logic, not vice versa. This means that logical deductions do not necessarily require mathematics. Verbal propositions may not exhibit the convenience of mathematical propositions, but they can exhibit rules of formal logic [Lee, 1989]. Yin [2003] also proposes the value of the process of induction, while Krefting [1990] and Rolfe [2006] also pointed out the importance of rigor for excellent analyses of case studies.

Other problems involve replicability of events and generalizability of findings. MIS researchers are unlikely to get the same observations across situations and this will hinder any subsequent attempts to make independent investigations to verify the findings made. While repeatable events and observations may be difficult to achieve, Lee [1989] argued that an alternative solution is to begin with the same sets of initial conditions when looking for another case. Though observations in MIS cases may be non-replicable, the findings that they may confirm or refute may be replicable, and consequently, generalizable. He added that generalizability is a quality describing a tested and confirmed theory in different studies such as laboratory and statistical experiments or case research [Lee, 1989]. Flyvbjerg [2011] also discusses how case studies can in fact be generalizable.

One of the key strengths of the case study discussed by Flyvbjerg [2011] is the in-depth understanding of contexts and processes that such case studies generate. By mobilizing a qualitative case study, we explore in detail within the context of a major application program by looking at the processes and dynamics that are involved, which is discussed in the next section.

4.2 Case Subject: Supplies Purchase Ordering (PO) System

We carried out this case study in a firm that is considered to be the largest family-owned private dental distributor in the United States. It has over 30,000 customers and is home to over 1,300 employees spread across the fifty states including its five distribution centers in Florida, Indiana, Nevada, Texas, and its flagship location in Pittston, Pennsylvania. In February 2010, the company ventured into outsourcing, choosing the Philippines as its first international location and making a partnership with a Philippine-based company with the goal of obtaining offshore support for its telemarketing and ICT needs. One of its projects was the development of the Supplies Purchase Ordering (PO) System.

The Supplies Purchase Ordering (PO) System was expected to be the mechanism by which clients would place orders for its inventories. The project involved the migration from an older system to use the latest technology. The development of the PO System followed the agile methodology, with several stakeholders involved. The development process took a total of six months to complete. Major requirements came from the developer in the US who previously implemented the system in an older technology. The project was marred by difficulties, among them technical requirements from the senior applications developers not being met by assigned developer, and difficulties during the testing process due to lack of clients' close involvement. This caused considerable delays, shifts in deadlines, and affected production delivery. As there are also multiple users, there were difficulties in ensuring that all possible scenarios are met.

Due to these difficulties, the development of the PO System had several initial solutions, ranging from different multiple user interface designs and different ways of parsing the files. What is interesting about this case is that eventually, the solution was found to be a single link that accepts files that follow the cXML (commerce eXtensible Markup Language) format. This link also points to a middle-tier application that processes these files.

The revised PO System (the link) was eventually set up to accepts orders from clients through submission of files. These files are parsed by looking at specific tags to get the order details and will then be shipped to clients based on the "ship to address" tags defined in the files. Some of the clients include private companies and departments from different universities in the US.

4.3 Case Analysis

We examined the process by which the company arrived at this new PO system. We used participant observation and discourse analytical techniques during the six months duration of the PO System capturing the entire development process from the requirements analysis to system deployment. Primary sources of data included instant chat conversations, email conversations, and online intranet discussionsbetween developers, users, and the management team. We also reviewed program codes.

To analyze the data gathered, we used discourse analysis, which can be defined in a narrow sense or a broad sense. In its narrow sense, discourse analysis may simply be defined as the detailed analysis of a language [Crowe, 2000; de Wet, 2001], which in practice usually takes the form of a text such as transcribed natural speech, professional documentation, and Internet communication materials. Discourse analysis is considered by some to be a research method that involves the examination of communication materials to gain new insights [Hewitt, 2009]. In its broader sense, discourse analysis includes a deeper understanding of several contextual factors such as the relationship between discourse and power [Crowe, 2000], discourse and politics [de Wet, 2001], discourse and cultural inclinations [Alldred and Burman, 2004], and the like. To add to this complexity, there are other research studies that are beginning to explore its value to other contexts such as what is now a growing research analysis technique called visual discourse, where images and other symbolic forms are interpreted according to the context of what is being studied.

Whatever definition we take, there is a common understanding that discourse analysis emphasizes the construction of a text; various approaches exist as to how much focus to give on context. It is important to note that in process of providing these contextual interpretations, researchers may be affected by implicit elements such as the researcher's own intentions, beliefs, and feelings, there are then those that assume that people, too, are subject to discourse [Jacobs, 2006; Powers, 2007].

As discourse analysis may be understood to be the result of a human activity linked to individual or social ideologies, a number of researchers believe that there is no definitive method for discourse analysis [de Wet, 2001; Alldred and Burman, 2004; Graham, 2005]. Hajer [2006], for instance, prescribed an iterative approach, composed of ten steps, in doing discourse analysis such as interviews with key players, interpretations and revisiting the key actors. On the contrary, Jansen [2008] argues against the use of series of steps as discourse analysis must be viewed as a non-definitive process. Consequently, this offers a wide set of approaches or perspectives, depending on the focus or type of knowledge researchers claim to make.

The analysis was patterned from the suggestion of Boaduo [2011]. First, we assigned labels/codes to the raw data to link these bits of data to ideas in order to see patterns. We then categorized the labels/codes by identifying how they can be grouped together to reflect the general idea known as the categories/themes. By prioritizing and coming up of hierarchy to the categories/themes, we then determined the relationships that exist between them.

In discourse analysis, contextualization of the phenomena or discourses is important [de Wet,

<Table 1> Comparison of Natural Languages and Programming Languages

	Natural Languages (e.g. English)	*Programming Languages (e.g. C#)*
Basic Elements	They are called **words** such as articles (e.g. *the*) and verbs (e.g. *login*).	They are called **tokens** such as data types (e.g. *int*) and variable names (e.g. *counter*).
Group of Elements	They are called **statements** (sentences). Construction of statements is bound by formal rules.	They are called **statements** (instructions or lines of code). Construction of statements is bound by formal rules.
Formal Rules in Statements	They follow a set of grammatical rules or simply **grammar**. These rules differ based on the different types of statements.	They follow a set of syntactical rules or simply **syntax**. These rules differ based on the different types of statements.
Types of Statements	Statements have several types such as declarative, interrogative, etc.	Statements have several types such as declaration, initialization, etc.
What Sequence and Blocks of Statements Indicate	The sequence of statements is aimed to build line of thought (e.g. description of requirements) that composes **paragraphs**. For instance, topic statements need to be cited first before the detailed discussion in order to make a more coherent point.	The sequence of statements is aimed to build line of thought (e.g. mathematical operations) that composes **code blocks** such as if and method blocks. For instance, declarations need to be written down first before initialization may be performed.
Special Notes and Instructions	They may be placed as a **postscript**, for instance, but may also be embedded within other statements.	They may be embedded within other statements (e.g. use of "//" or "/**/"), and are generally called **comments**.

2001; Borrell, 2008]. Therefore, as we are also doing participant observation, we gain an insider's point of view or perspective about the context of development of applications; therefore, much of the context used in the analysis is assisted and informed by our experiences and interpretations derived from the observations process.

V. Findings

In the course of our analysis, we identified that since one of the primary sources of data come from program codes, it is important to see how it can be dissected for analysis. Therefore, while requirements for the development of programs are expressed in natural languages, these ideas are expressed in program codes through the use of programming languages during the actual coding. A close examination of program codes suggests that as a language, specifically programming language, they have similarities with those of the natural languages such as English, which make further analysis and interpretations involving programming languages easier. These are summarized in <Table 1>.

5.1 The Social Shaping of Decision-Making Process Results in Certain Values being Prioritized

<Table 2> reveals the evidences how the process of applications development is socially shaped. We can also see how certain values are prioritized as specific stakeholders take on and win over the stages of the decision-making process.

<Table 2> Social Shaping of the Decision-Making Process

Stages	*Stakeholders*	*Communication Mechanism*	*Outcomes*	*Emerging Prioritized Values*
Initial contact with the first user to explain business-initiated change	Business analyst (former systems developer) Single user(technical user)	Face to face meeting	List of user requirements showing "limits" of changes (to manage user expectations)	Basic automation of order submission function increasing efficiency, productivity, and convenience
Initial contact with development team to cascade user requirements	Business analyst (former systems developer) Systems developer IT director	Email with a single page user requirements	Three proofs of concepts that employ several options with user interfaces	Basic automation of order submission function increasing efficiency, productivity, convenience, and sales goal (as noted by the director)
Modification of proofs of concept	Business analyst (former systems developer) Systems developer Four senior systems developers Later: one of the senior systems developers took charge	Face to face meeting, email, and teleconferencing	Rejection of the earlier three proofs of concept; instead of displaying user interfaces, the final output turned out to be just a link that processes the orders	User familiarity: stick to practice of using a link instead of new solutions that involve presentation of new user interfaces; Usability: quick response time as proposed by the senior systems developers
Launching of version 2 to the first user	Business analyst (former systems developer) Systems developer Senior systems developer IT director	Face to face meeting, email, and teleconferencing	Version 2(a) showing modified input requirement (the use of cXML format)	Functionality: parsing of files; types of files; Preference in best coding practices defined by the senior systems developer
Testing of version 2	Business analyst (former systems developer) Systems developer Single user(technical user) In-house quality tester	Email and teleconferencing	Version 2(b) showing accommodations for the user's request for output response messages	Flexibility and Customization to grant the client's request in order to proceed with testing and continued use of the system.
Launching of version 3 to other multiple users	Business analyst (former systems developer) Systems developer Senior systems developer IT director	Face to face meeting, email, and teleconferencing	Version 3 showing accommodations multiple users logging in, as well as being able to record all information about the users' requests (date, time, specific orders) in a text file.	Technical Support and Ease in Debugging: since there are multiple users that access the system, it is important to know the details of the transactions being sent to the system.
Testing of version 3	Business analyst (former systems developer) Systems developer Multiple users (combined technical and non-technical users) In-house quality tester	Email and teleconferencing	Loss of non-routine transactions for specific users and therefore needs to redeploy. Some features include parsing of special orders (those that use special product numbers, and therefore different way of processing). In cases of errors, a notification email will be sent to the systems developers, aside from being able to log the details in a text file.	Customization for Users' Satisfaction and Achievement of Sales Goals. The code is revised to process special orders according to the request of the users. Not allowing them to do this reduces the chance that they use the system. Also, possible errors are fixed even before the customers report them through the email notification.
Deployment	Business analyst (former systems developer) Systems developer Multiple users (combined technical and non-technical users) IT director	Email	No change	Customer support is provided through email and teleconference; Developers tweak the code in rare cases but most of the time, users ask how to make successful transactions for specific files and therefore adapt to the technical implementation

In general, we can say that there are at least three sets of factors that affected the decision-making processand therefore shaped the applications development process.

The first set of factors is referred to as *Technical Limitations and Strengths*, which include mainly technical considerations such as non-standard workaround due to unavailability of features, as well as the use of recommended, built-in implementations of features. We saw these prevail during early stages. During the conceptualization of the application program, the focus was on the primary stakeholder, that is, the users who are intended to use the system, and the strategy was to get the requirements from them. However, problems arose when these users could not specify the requirements with a level of precision required to addresses current problems. Therefore, in most cases, a default priority that emerged within the team to simply replace such detailed requirements was the simple, basic priority of *automation of business functions* in order to streamline a manual process. This was coupled with expectations of efficiency, productivity, and convenience. This pattern of thinking links back to the common notion that technological artifacts are primarily solutions to problems of automation and efficiency. We observed this pattern of thinking during the meetings where most questions asked were limited to specific (technical) implementations in order for the development team to plan the technical design – identifying the limitations and current strengths based on purely technical requirements, to the exclusion of other factors like user friendliness or company business priorities. After perfunctory consultation with users, design tasks were then defined by the applications development team members themselves, often solely by familiarity of existing technical capabilities of the current resource team. The output of this stage was a list of user requirements, with limited emphasis (on automation, efficiency, and familiarity, to the exclusion of other values), a set of specifications that therefore marginalized opportunities for best practices, resulting in limitations to the functionalities and affecting customer experience.

As development progressed, we noted that priority values shifted as new sets of stakeholders got involved. When the time came for cascading user requirements to the development team, the IT director entered into the arena and emphasized the company's need for such a project in order to increase sales goals. Although the IT director did not normally concern himself with sales, he was in this case jointly assigned to be the coordinator, along with the sales director for the sales team, specifically to understand their requirements and come up with solutions. His prioritization of projects was therefore influenced not only by technical directions but also by his functional role. This opened up the development process to a second category of factors, referred to here as *Management-Imposed* priorities, or goals linked to the directives of the management team (such as resulting priorities based on sales goals, and customer satisfaction). As a result, these business-oriented values were now brought into the picture and were seen to be imposed on the developers, thus reshaping the technical design and coding process. One of the outcomes of this process was that during the middle (second and third stages) of development, the development team had to seek other solutions to support PO transactions that would reconcile both user requirements and business goals. The outcome of this stage involved three proofs of concept with several user interfaces, a customer orientation meant to reflect the company goal of "pushing" the project as part of its sales goals.

These proofs of concept were upheld as the solution until another stage of the development process then ushered in a new set of stakeholders. Senior developers were brought in to do a code review, and in this process it was found that there was already an existing mechanism between the company and the users, in the form of an existing link in the old system. Senior developers proceeded to focus on this link as the solution, upholding the reasons that it provided more usability and faster response time.In the end, the previously-developed user interfaces were scrapped in favor of this single link. At this stage, business goals could be seen to yield, at least in part, to more pragmatic priorities like speed and usability.

It is also observed that although the technical development team was present in all meetings and that technical enhancements were first focused by stakeholders, it was only in the later stage, specifically during the fourth and fifth stages, that one important technical consideration was drawn into the decision-making arena. This involved the specific input file type has been specified, and this consideration showed how a third set of factors, *Personal Practice*, can also shape the development process. Personal practice refers to the individuals' undertakings (such as by programmers' preferences, and career goals).

During the testing (fifth) stage, an in-house tester was, at first, hesitant to send test files to the link because the link did not have the keyword "dev" or "test", which she felt might lead to test files being processed as actual production files. Although the developers already argued that it may be used for testing, the quality tester insisted that such inconsistent practice would lead to future problems such as not being able to properly identify which ones are for development, for testing, and for production. Tentative compromises were made to address this; for example, it was agreed that when sending test order files, senders would follow up immediately with an mail. However, despite regular notices, not all users complied with this agreement. A new mechanism that therefore emerged from this disagreement (and from the personal practice of a tester) was a technical solution called a logging mechanism that would allow one to track and monitor all transactions sent by the users.

Apart from these clusters of factors, there were other considerations that shaped the final outcome, among them historical considerations and special needs. Since the application is already a legacy system, there were special types of orders that need to be processed in special ways too. It was noted by some users that the application may be less usable in cases where these special orders would not be handled. For instance, it would be inconvenient to use the system for routinary orders and use manual process for special ones. The logging mechanism was also not enough to alert any of the development team especially when something went wrong, like an incorrect parsing of the test order file. All of these resulted in other enhancements being made in the system.

5.2 Conflicting Values

The interplay of these layers of values was not always straightforward or sequential. At times, it was clearly observable how sets of values could contradict one another. Four examples are given here.

Conflicts of *Personal Practice* could readily be noted upon examination of the comments on program codes. The use of in-line comments differed per coder. Some developers considered these comments as "clutter" that affected readability. Comments could be seen as a team members' personal practice being imposed on the assigned developer. They were also seen to affect the progress of the project because the developer would have to note the practices of the code reviewers and subsequently adapt a different programming style according to their practices. The process of considering not only the functional requirements but also the practices of the other team members therefore affected judgment during the development process too.

Another example of conflicts in personal practice involved how the new application should be mobilized in a testing situation. On one occasion, the in-house quality tester used the new tool not just for testing whether the application was able to process the test order files, but also to test whether the connections to the other servers were working in order to resolve the problem in another applications development project that she is involved in. This is an example of conflicting values in that a tool that was designed for one purpose is mobilized for other purposes, hence potentially leading to unexpected outcomes.

A third set of circumstances where conflict is evident is during discussions with the users and the development team. In this context, it became apparent that there was a need to negotiate tradeofs among factors, or example tradeoffs between Functionality, Convenience, and Security. The team had to consider that the more features were developed, the more convenience the application is thought to bring to stakeholders. However, increasing functionality also made the output more vulnerable to glitches, thereby affecting Security. In this case study, security was prioritized because this was a common concern among the stakeholders, and also because the system processed payments.

A fourth point where conflict had to be resolved was at the level of specific if-else conditions in program codes. These show that even after functional requirements have already been defined, and the technical framework has been put in place, developers still have choices in more detailed processing matters. One decision would involve deciding which of several users to process first. This judgment call implies that the first in the list may experience faster transactional experience. On the other hand, those users with other business logical checks can experience longer response times. This shows that even though adjustments by the developers are rare, there are still specific points within the development process where a developer's value judgments can shape an artifact's outcomes

Ⅵ. Conclusion, Contributions, and Future Work

In carrying out this study, we have made a number of contributions. We have shed some light on the debate on whether technology is value-neutral or value-laden. Our study has shown how the process of technological development, specifically applications development, is permeated with human choices, and by extension, judgments, biases, and values. We have also demonstrated empirically how, when multiple stakeholders are involved, conflicting values add complexity to development scenarios. Our study suggests that concluding decisions are mostly made by the developers who take account the completion of functional and technical requirements, and who in doing so embark on a process of prioritization of given goals using his personal biases and values. Our study therefore supports the view that technology is value-laden. An enriched understanding of this debate is important because it has implications to the process of learning, interactions, and knowledge sharing among individuals in many fields.

Second, we have made a methodological contribution by extending the application of discourse analysis. In this study, we have found close similarities between natural and programming languages, making a significant impact as to how future discourse analysis studies can be made.

Finally, we have made a contribution to practice. By exploring the micro-processes that are implicated in the applications development process, we hope to sensitize managers and practitioners to the potential vulnerabilities caused by internal (self) and external (environment) factors that may shape their decision-making processes.

We can identify various areas for future work. Because much of this study also involves moral values and social consequences, we further note that the study on ethics has gradually shifted away from focusing solely on human factors to exploring the role of computer programs and other technological artifacts in ethics as well [Brey, 2009]. It has been suggested, for example, that since computer programs are argued to have embedded moral values, they may also be accountable to legal or moral responsibilities [Johnson, 2006]. In tackling the value-ladenness or value-neutrality of these artifacts, it is therefore necessary to involve ethical theories in further case studies in the future.

〈References〉

[1] Alexiadis, P. and Cole, M., the concept of technology neutrality… ECTA Review (European Competitive Telecommunications Association) Retrieved on September 10, 2014 from ECTA Review Online: http://www.gibsondunn.com/fstore/docuzments/pubs/Alexiadis-ECTA_Review_2004.pdf.

[2] Alldred, P. and Burman, E., Analysing Children's Accounts using Discourse Analysis. Researching Children's Experience, 2004.

[3] Amiel, T. and Reeves, T.C., "Design-Based Research and Educational Technology: Rethinking Technology and the Research Agenda," *Educational Technology and Society*, Vol. 11, No. 4, 2008, pp. 29-40.

[4] Anyanwu, L., "Object Information Repository in a Christian Theory of Technology," Association of Christians in the Mathema-

tical Sciences, Retrieved on March 16, 2014 from ACMS Online: http://www.acmsonline.org/journal/2006/Anyanwu.pdf.

[5] Barlatier, P., Vidou, G., Jacquemart, S., and Latour, T., "Don't Value the Valueless: Toward a Model of Evaluation of Knowledge within E-Communities of Practice," Proceedings of the 2nd International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, 2006, pp. 43-57.

[6] Bijker, W., "The social construction of bakelit: Toward a theory of invention, The social construction of technological systems: New directions in the sociology and history of technology," Cambridge, MA: MIT Press, 1987.

[7] Bilbao-Osorio, B., Dutta, S., and Lanvin, B., "The Global Information Technology Report 2014. Rewards and Risks of Big Data," World Economic Forum, INSEAD: The Business School for the World, JOHNSON Cornell University, 2014.

[8] Boaduo, N., "Systematic Analysis and Interpretation of Collected Data for a Research Study: A Practical Methodological Framework for Writing Research Report," *Educational Research and Review*, Vol. 6, No. 2, 2011, pp. 140-146.

[9] Borrell, J., "A Thematic Analysis Identifying Concepts of Problem Gambling Agency: With Preliminary Exploration of Discourses in Selected Industry and Research Documents," *Journal of Gambling Issues*, Vol. 22, 2008, pp. 195-218.

[10] Brey, P., "Values in Technology and Disclosive Computer Ethics," The Cambridge Handbook of Information and Computer Ethics, Ed. L. Floridi, Cambridge: Cambridge University Press, 2009.

[11] Camp, J., "Democratic Implications of Internet Protocols," *The Information Society*, Vol. 15, 1999, pp. 249-256.

[12] Carey, J., "Communication As Culture: Essays on Media and Society," New York, Routledge, 1992, p. 139.

[13] Crowe, M., "Constructing normality: a discourse analysis of the DSM-IV," *Journal of Psychiatric and Mental Health Nursing*, Vol. 7, 2000, pp. 69-77.

[14] de Wet, C., "A media discourse analysis of racism in South African schools," *International Education Journal*, Vol. 2, No. 5, 2001, pp. 98-112.

[15] DeSanctis, G. and Poole, M., "Capturing the complexity in advanced technology use: Adaptive structuration theory," *Organizational Science*, Vol. 5, No. 2, 1994, pp. 121-147.

[16] Eisenhardt, K., "Building Theories from Case Study Research," *Academy of Management Review*, Vol. 14, No. 4, 1989, pp. 532-550.

[17] Feenberg, A., Critical Theory of Technology, Oxford University Press, Vol. 5, 1991.

[18] Flanagan, M., Howe, D., and Nissenbaum, H., "Values at Play: Design Tradeoffs in Socially-Oriented Game Design," Proceedings of the CHI 2005 conference on Human factors in computing systems, New York: ACM Press, 2005.

[19] Flanagan, M., Howe, D., and Nissenbaum, H., "Embodying Values in Technology: Theory and Practice," *Information Technology and Moral Philosophy* (eds. J. van den Hoven and J. Weckert), Cambridge: Cambridge University Press, 2008.

[20] Flyvbjerg, B., "Case Study," in Norman K. Denzin and Yvonna S. Lincoln, eds., The Sage Handbook of Qualitative Research, 4th Edition (Thousand Oaks, CA: Sage), 2011, pp. 301-316.

[21] Fountain, J., Building the Virtual State: Information Technology and Institutional Change, Washington, D.C.: The Brooktings Institution, 2001.

[22] Franssen, M., Lokhorst, G., and van de Poel, I., Philosophy of Technology, The Stanford Encyclopedia of Philosophy, 2013.

[23] Friedman, B. and Nissenbaum, H., "Bias in Computer Systems," *ACM Transactions on Information Systems*, Vol. 14, No. 3, 1996, pp. 330-347.

[24] Friedman, B. and Nissenbaum, H., "Software Agents and User Autonomy, International Conference on Autonomous Agents," Proceedings of the first international conference on Autonomous Agents, Marina del Rey, California, United States. New York: ACM, 1997.

[25] Graham, L., "Discourse Analysis and the Critical Use of Foucault," 2005 Annual Conference-Australian Association for Research in Education, 2005.

[26] Hajer, M., "Doing Discourse Analysis: Coalitions, Practices, Meaning, Words Matter in Policy and Planning: Discourse Theory and Method in the Social Sciences," Netherlands Geographical Studies, 2006, pp. 65-74,

[27] Hewitt, S., "Discourse Analysis and Public Policy Research," Centre for Rural Economy Newcastle University Discussion Paper Series, No. 24, 2009.

[28] Iacono, J., Brown, A., and Holtham, C., "Research Methods-a Case Example of Participant Observation," *The Electronic Journal of Business Research Methods*, Vol. 7, No. 1, 2005, pp. 39-46.

[29] Introna, L., "Disclosive Ethics and Information Technology: Disclosing Facial Recognition Systems," *Ethics and Information Technology*, Vol. 7, No. 2, 2005, pp. 75-86.

[30] Introna, L. and Nissenbaum, H., "Shaping the Web: Why the Politics of Search Engines Matters," *The Information Society*, Vol. 16, No. 3, 2000, pp. 1-17.

[31] Jacobs, K., "Discourse Analysis and its Utility for Urban Policy Research," *Urban Policy and Research, Routledge*, Vol. 24, No. 1, 2006, pp. 39-52.

[32] Jansen, I., "Discourse analysis and Foucault's 'Archaeology of knowledge'," *International Journal of Caring Sciences*, Vol. 1, No. 3, 2008, pp. 107-111.

[33] Janz, B., Understanding Knowledge Creation, Transfer, and Application: Investigating Cooperative, Autonomous Systems Development Teams. Proceedings of the 38th Annual Hawaii International Conference, 2005.

[34] Johnson, D., "Is the Global Information Infrastructure a Democratic Technology?" *Computers and Society*, Vol. 27, 1997, pp. 20-26.

[35] Johnson, D., "Computer Systems: Moral Entities but Not Moral Agents," *Ethics and Information Technology*, Vol. 8, No. 4, 2006, pp. 195-204.

[36] Kahn, H. and Wiener, A., The Year 2000, A Framework for Speculation on the Next Thirty-Three Years. Macmillan, New York, 1967, p. 51.

[37] Klein, H. and Kleinman, D., "The Social Con-

struction of Technology: Structural Considerations, Science," *Technology and Human Values, SAGE Publications*, Vol. 27, No. 1, 2002, pp. 28-52.

[38] Koops, B.-J., "Should ICT Regulation be Technology-Neutral. Starting Points for ICT Regulation: Deconstructing Prevalent Policy One-Liners," The Hague: T.M.C. Asser Press, *IT and Law Series*, Vol. 9, 2014, pp. 77-108.

[39] Krefting, L., "Rigor in Qualitative Research: The Assessment of Trustworthiness," *American Journal of Occupational Therapy*, Vol. 45, No. 3, 1990, pp. 214-222.

[40] Latour, B., Where are the Missing Masses? The Sociology of a Few Mundane Artifacts,' in Shaping Technology/Building Society: Studies in Sociotechnical Change (eds. W. Bijker and J. Law), Cambridge, MA: MIT, 1992.

[41] Lee, A., "A Scientific Methodology For MIS Case Studies," *MIS Quarterly*, Vol. 13, No. 1, 1989, p. 33.

[42] MacKenzie, D., "Inventing accuracy: A historical sociology of nuclear missile guidance," Cambridge, MA: MIT Press, 1990.

[43] Miller, S., Where is Cyberspace? Civilizing Cyberspace: Policy, Power, and the Information Superhighway, New York: ACM Press, 1996.

[44] Mumford, L., The Myth of the Machine, Technics and Human Development. New York: Harcourt, Brace and World, Inc, 1967.

[45] Nandhakumar, J. and Jones, M., Development Gain? Participant Observation in Interpretive Management Information Systems Research, Qualitative Research: SAGE, 2002.

[46] NIST, Framework for Improving Critical Infrastructure Cybersecurity. National Institute of Standards and Technology. Retrieved on September 10, 2014 from NIST Online: http://www.nist.gov/cyberframework/upload/cybersecurity-framework-021214-final.pdf, 2014.

[47] Orlikowski, W., "Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations," *Organization Science*, Vol. 11, No. 4, 2000, pp. 404-428.

[48] Orlikowski, W. and Baroudi, J., "Studying Information Technology in Organizations: Research Approaches and Assumptions," *Information Systems Research*, Vol. 2, No. 1, 1991, pp. 1-28.

[49] Rolfe, G., "Validity, Trustworthiness and Rigour: Quality and the Idea of Qualitative Research," *Journal of Advanced Nursing*, Vol. 53, No. 3, 2006, pp. 304-310.

[50] Saarni, S., Hofmann, B., Lampe, K., Luhmann, D., Makela, M., Velasco-Garrido, M., and Autti-Ramo, L., "Ethical analysis to improve decision-making on health technologies," *Bulletin of the World Health Organization*, Vol. 86, No. 8, 2008, pp. 617-623.

[51] Schuurman, D., "Forming a Christian View of Computer Technology," *Journal of the Association of Christians in the Mathematical Sciences (ACMS)*, December, 2007.

[52] Santori, M., The Bitcoin Foundation. NY Dept of Financial Services Hearing on Virtual Currencies, 2014.

[53] Seneviratne, O. and Monroy-Hernandez, A., Remix culture on the web: A survey of content reuse on different User-Generated content websites, Web Science, 2010.

[54] Singh, N., Zhao, H., and Hu, X., "Cultural

adaptation on the Web: A study of American companies' domestic and Chinese websites," *Journal of Global Information Management*, Vol. 11, No. 3, 2003, pp. 63-80.

[55] Sundstrom, P., "Interpreting the Notion That Technology is Value-Neutral," *Medicine, Health Care and Philosophy*, Vol. 1, 1998, pp. 41-45.

[56] Pablo, Z., "The Discursive Construction of Metaphors on Web-Based Portals: the Case of the Development Gateway," *Journal of Research in Science, Computing, and Engineering*, Vol. 4, No. 2, 2007, pp. 73-84.

[57] Postman, N., Technopoly, New York: Vintage Books, 1993.

[58] Powers, P., "The Philosophical Foundations of Foucaultian Discourse Analysis," *Critical Approaches to Discourse Analysis across Disciplines*, 2007, pp. 18-34.

[59] Tapscott, D. and Williams, A., Macrowikinomics: Rebooting Business and the World. London: Atlantic Books, 2010.

[60] Tavani, H., "Informational Privacy, Data Mining, and the Internet," *Ethics and Information Technology*, Vol. 1, No. 2, 1999, pp. 137-145.

[61] Williams, R. and Edge, D., "The social shaping of technology," *Research policy*, Vol. 25, 1996, pp. 865-899.

[62] Yin, R., Case Study Research Design and Methods, Third Edition. Applied Social Research Method Series Vol.5, 2003, Sage Publishing.

◆ About the Authors ◆

Christobal Cayaba
Chris has over 8 years of technical expertise in software engineering for different clientele (American, European, and Asian). Chris earned his bachelor's degree in computer science at the De La Salle University, and obtained master's degree at the Ateneo de Manila University, where he was sent as a delegate to the Nara Institute of Science and Technology, Japan. Chris is now on his final year for his doctorate degree. His research interests include "Natural Language Processing," "Value-Ladenness of Technology," and "Knowledge Creation and Conversion."

Zelinna Pablo
Dr. Zelinna Pablo is a sessional lecturer at the Department of Marketing and Management at The University of Melbourne. She completed her PhD at The University of Melbourne and a postgraduate diploma at the University of Amsterdam. Her research interests include social, organizational, and ethical dimensions of IT. She was formerly Associate Professor of Information Technology at the De La Salle University Manila.

Submitted : May 30, 2014
1st revision : September 12, 2014
Accepted : September 15, 2014