

QuiQ: A New Systems Development Methodology Suitable for E-Business Projects

IVY M. TARUN *¹

¹ Isabela State University, Cabagan, Isabela 3328 Philippines

Abstract

Many businesses are already taking place online. This poses new challenges to systems development (SD) teams as to the adoption of methodologies since there are varied existing methodologies at the moment that are not specifically identified to be suitable for this purpose. Hence, this study focused on designing a new systems development methodology suitable for E-business projects. Specifically, it identified: the SD best practices of IT practitioners; the strengths and weaknesses of the existing SD methodology; and it helped design a SD methodology that was applicable to e-business projects tailored from existing ones and from best practices of IT practitioners. A survey of over sixty IT practitioners indicated that they are using two or more methodologies during SD and that majority of them are using the Waterfall. Their selection of methodology depended on the developers technical know-how, relevance of the methodology to the project, and to the well-defined steps of the methodology. Their top priorities were budget, time, and cost. Consequently, QuiQ (Quick and Quality-oriented) SD methodology was developed based on the respondents practices and strengths of the existing methodologies that were adopted while weaknesses were addressed.

Keywords: E-Business, Systems Development, Systems Development Methodology, Systems Development Best Practices

Introduction

E-business or electronic business, according to Nelson, T. (2005) is an online business that involves buying, selling, providing services, and associating with partners (Nelson, 2005). This has been the trend in the past five years or more. Many businesses are taking place in the Internet due to its flexibility, accessibility, convenience, and efficiency in managing internal functions. Sofia Ali said that Information and Communications Technology (ICT) when used in business, may improve productivity, increase customer participation, reduce costs, and allow customization. With the increasing demand from business organizations, Information Technology (IT) Project Management (PM) teams have to modify their development methodologies to meet the needs of their clients. This poses new challenges to SD methodology adoption since there are varied existing methodologies at the moment.

SD Methodology, as explained in *yourdic-*

tionary.com, describes the objectives of the different phases and the expected result for each phase (Computer Desktop Encyclopedia, 2010). However, not all of the existing methodologies are appropriate for all kinds of projects. There may be available methodologies that are well accepted but may not effectively support the varying characteristics of the business processes such as the e-business. Shi, N. and Murthy, K. (2001) stated in their book that a standardized development methodology is not best suited for all, or even most, e-business projects (Si et al., 2002) as cited by Knight, L., Steinback, T. and Kellen, V. in their research. However, according to Solicito, M. (2001), the difference between the traditional system development approach and e-business development approach is not as big as others imagine. She said that project managers must adhere closely to a SD development methodology when building e-business systems. Furthermore, she explained that e-business system may fail if there is no correct planning and there are no methodologies employed (Sollicito, 2010).

Perks, M. (2003) stated that prior selection

*Corresponding author. Email: ivy_tarun@yahoo.com; Tel.: +63 78 323-2126

of appropriate development lifecycle process is important because this will serve as guide in undertaking the needed activities (Perks, 2003). In the study of Russo, N., Wynekoop, J., and Walz, D. (1995), there are several organizations who have SD methodology but strict compliance to it was not observed and the extent of following the methodology differs in every project (Russo et al.). This discovery points out that there should be a methodology distinct to a certain project or system. Such finding was affirmed by Parker, X. (2008) who declared that, even Web-based initiatives required a sound development methodology based on an overall strategy, that fits into the entire organizations business and IT plans. In addition, the needs of business should be taken into account in the choice of methodologies (Sauer and Lau) as these may bring structured tasks and functions, controlled costs, project and human resources, minimized uncertainties, exercised power and security, and accumulated knowledge and experiences.

It is on the above-mentioned contexts that the researcher has undertaken this study to design a SD methodology that suitable to e-business projects. Specifically, the study sought to determine the SD best practices of IT practitioners; determine the strengths and weaknesses of the existing SD methodology; and design a SD methodology applicable to e-business projects tailored from the existing ones and from best practices of IT practitioners.

Methodology

Two approaches were used to deal with the objectives of the study. The first approach was through survey in which a questionnaire was used to address the first objective that is to gain facts about the SD best practices of IT practitioners. The respondents of the study were IT practitioners who were comprised of 8 IT faculty members, 20 IT personnel employed in private organizations, and 38 members of the different SD teams. The questionnaire used was created using Google Forms and its print version was distributed personally to selected respondents while its online version was distributed through Facebook to accommodate more responses.

Meanwhile, the second approach used was literature review and analysis to address the second objective that is to gain an overview of the strengths and weaknesses of the existing SD methodologies. Results of these two approaches were used as bases in developing a new methodology that is applicable to e-business projects.

Results and Discussion

Out of the 66 respondents, 36 IT practitioners responded in the survey invitation through Facebook while 30 were retrieved after having distributed the questionnaire personally to the selected respondents.

A. SD Best Practices

In Table 1, percentages of those who were either using a methodology or not are presented. It is interesting to note that all of the respondents were using methodology/ies when developing systems. Roughly 45.5% were using two or more methodologies during SD. This result is in congruence with the study of Ruso, N. et al. (1995) that 84.2% of organizations are using one or more SD methodologies (Russo et al.).

Moreover, the respondents who were using methodology/ies were asked specifically about what methodology they used. It could be gleaned from Table 2 that majority of the IT practitioners, at 63.6%, were using Waterfall Development methodology. This is perhaps because of its easy-to-implement methods (Sousa, 2009). Another notable result is that none of them were using the object-oriented methodology despite its promise of improved reliability and flexibility (Burlerson Consulting, 2012).

The respondents were also asked about what methodology they think is suitable for e-business projects. Although most of the IT practitioners were using Waterfall, 36.4% of them, as presented in Table 3, believed that Spiral Development is suitable for e-business projects. This observation is perhaps due to the shortcomings of Waterfall that are being addressed by Spiral. According to Galaxy008 (2012), the disadvantage of the waterfall model is the advantage of spiral model.

Table 1. Use of Methodology

Use of Methodology	Frequency	%
Uses only one methodology	12	18.2
Uses one methodology but didnt follow entirely	18	27.3
2 or more methodologies combined	30	45.5
Prepared their own and tailored from the existing methodologies	6	9.09
No methodology used	0	0

Table 2. Methodology Used

Methodology	Frequency	%
Waterfall Development	42	63.6
Prototyping	24	36.4
Incremental Development	12	18.2
Spiral Development	12	18.2
Rapid Application Development	12	18.2
Object oriented	0	0
Top-down programming	6	9.09
Unified Process	6	9.09
Agile Methodology	42	63.6

Table 3. Methodology Suitable for E-Business Projects

Methodology	Frequency	%
Waterfall Development	18	27.3
Prototyping	18	27.3
Incremental Development	18	27.3
Spiral Development	24	36.4
Rapid Application Development	6	9.09
Object oriented	12	18.2
Top-down programming	0	0
Unified Process	6	9.09
Agile Methodology	0	0

Factors in the selection of a methodology are shown in Table 4. The methodology respondents used depended primarily on the system requirements. They also considered the developers technical know-how, relevance of the methodology to the project, and the well-defined steps of the methodology. Finally, they did not mind the popularity of the methodology being used.

When developing systems, the respondents looked into budget, time, and cost as their top three priorities. Documentation, user satisfaction, and execution appeared to be the least of the priorities of the IT practitioners as shown in Table 5.

During SD, most of the IT practitioners at 81.8%, gathered all requirements and had it fixed before any design and coding were done. This practice is consistent with the methodology that they are using, the Waterfall model. However, this also contradicts their practice in performing continuous end-user involvement and feedback since Waterfall is a method that progresses downwards from requirements analysis and there is no way to make changes. Once more, this method is again in conflict with their priorities because user satisfaction is not among these. Since execution was found out to be one of their least priorities, 72.7% of the IT practitioners, as shown in Table 6, prefer to put more emphasis on process rather than on execution. Another notable result of this study is that only a few of the IT practitioners believed that SD should be led by business people rather than an IT team.

B. Strengths and Weaknesses of SD Methodologies

The synopsis of the strengths and weaknesses of the selected system development methodologies are based on published literature. The selected system development methodologies are Waterfall, Rapid Application Development, Iterative and incremental, Spiral, and Agile methodology. Before proceeding to the synopsis, a detailed description of the selected methodologies is provided using various references:

Waterfall Development Methodology.The waterfall model is a sequential development approach, in which development is seen as flowing steadily downwards (like a waterfall) through the

phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

Rapid Application Development Methodology.Rapid application development methodology is a software development methodology, which involves iterative development and the construction of prototypes.

Iterative and Incremental Development Methodology.This is any combination of both iterative design or iterative method and incremental build model for development. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system.

Spiral Development Methodology.The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. It is a meta-model, a model that can be used by other models.

Agile Methodology.Agile methodology is a group of software development methods based on iterative and incremental development where requirements and solutions evolve through collaboration between self-organizing and cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change.

Table 8 shows the synopsis of the strengths and weaknesses of the selected SD methodologies taken from the studies of Pursell, J., Valentine, C. , Awad, M., and Patel, D. (Awad, 2005).

C. QuiQ: Proposed E-Business Development Methodology

This section describes the researchers new SD methodology that is suitable for e-business projects. This is established based on the practices of IT practitioners and the preceding synopsis. The weaknesses of the abovementioned methodologies were carefully addressed in this new methodology. In addition, the strengths of Rapid Application Development methodology were incorporated as it is applicable to web-based

Table 4. Factors to Consider in Selecting a Methodology

Factors	Frequency	%
Well-defined steps of the methodology	36	54.5
Developers technical know-how	36	54.5
Size of the development team	12	18.2
System requirements	54	81.8
Customer culture	12	18.2
Popularity of the methodology	0	0
Organizational culture	6	9.09
Relevance of the methodology to the project	36	54.5

Table 5. Systems Development Priorities

Priority	Rank
Budget	1
Cost	3
Deliverables	6
Documentation	7
Execution	9
Methodology	4
Resources	5
Time	2
User satisfaction	7

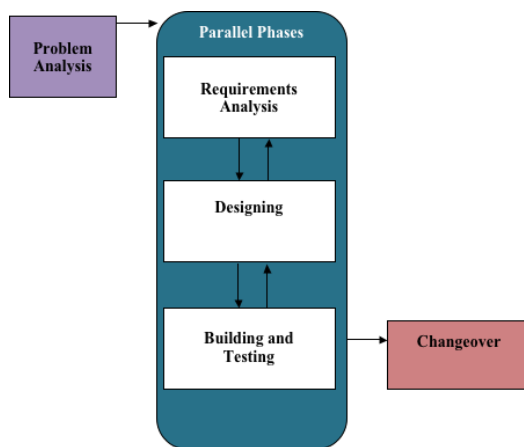


Figure 1. QuiQ Model

applications including e-business projects.

The new system development methodology that is suitable for e-business projects is named as QuiQ (pronounced as quick) which means quick and quality-oriented methodology. This said methodology aims for a speedy development of applications without disregarding its quality. A methodology must provide for the steps to take, and how and why these steps are performed. Consequently, the features of QuiQ are shown on Table 9 and its corresponding model is presented on Figure 1.

Table 9 shows the strengths of the existing SD methodologies that were adopted by the new model called QuiQ.

Table 10 show the weaknesses of the existing SD methodologies that were addressed by the new model called QuiQ.

Conclusions

With respect to the results obtained, the researcher concludes that most IT practitioners are using two or more methodologies during SD and that majority of them are using the Waterfall. However, the selection of methodologies depends on the developers technical know-how, relevance of the methodology to the project, and the well-defined steps of the methodology. For this reason, the new SD methodology named QuiQ can be a good option using these selection criteria of the IT practitioners. QuiQ was developed with phases carefully established based from practices.; It adopted the strengths of the existing development methodologies while the weaknesses were addressed.

The IT practitioners top priorities are budget, time, and cost. Consequently, QuiQ was de-

Table 6. Systems Development Practices

Practices	Frequency	%
a. Put more emphasis on process rather than on execution.	64	72.7
b. Gather all requirements and make it fixed before any design and coding are done.	72	81.8
c. Do integration earlier and perform daily build and smoke-test.	48	54.5
d. Review other people's work.	48	54.5
e. Plan test cases before coding starts, and develop test cases while the application is being designed and coded.	40	45.5
f. Keep planning to a minimum.	48	54.5
g. Perform continuous end user involvement and feedback.	64	72.7
h. Do Testing throughout the project.	32	36.4
i. Document every activity done by the team.	40	45.5
j. Design and code for the needs of today instead of those of tomorrow.	32	36.4
k. Allow on-site customer change requests informally, often by verbally informing the development team.	40	45.5
l. System development should be business led rather than purely led by IT team.	16	18.2

signed with parallel phases to reduce time and cost thereby addressing problems on budget constraints.

There was an inconsistency among respondents priorities and practices; hence, a more thorough study is recommended where a better sample of IT practitioners should be considered. Moreover, the respondents perspectives in this study may not be a general representation of the IT practitioners but this may serve as basis for future works. Lastly, it is further recommended that the new SD methodology called QuiQ be validated and evaluated for enhancement purposes.

References

- Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. Web page.
- Burleson Consulting (2012). Advantages and disadvantages of object-oriented approach. http://www.dba-oracle.com/t_object_oriented_approach.htm.
- Computer Desktop Encyclopedia (2010). System development methodology - technical definition.
- Kim, J., Kim, J., Park, S., and Sugumaran, V. (2004). A multi-view approach for requirements analysis using goal and scenario. *Industrial Management and Data Systems*, 104(9).
- Nelson, T. (2005). Definition of e-business (electronic business).
- Perks, M. (2003). Best practices for software development projects. Technical report, International Business Machines Corporation.
- Russo, N., Wynekoop, J., and Walz, D. The use and adaptation of system development methodologies. <http://www.andrews.edu/vyh-meistr/papers/sdm.html>.
- Sauer, C. and Lau, C. Trying to adopt systems development methodologies – a case-based exploration of business users' interests. *Information Systems Journal*.
- Si, N. N., Shi, V., and Murthy, K. (2002). Architectural issues of web-enabled electronic business. Technical report.
- Sollicito, M. (2010). A methodology for e-business. <http://www.quepublishing.com/articles/article.aspx?p=2>

Table 7. Strengths And Weaknesses of the Selected System Development Methodologies

SD Methodology	Strengths	Weaknesses
Waterfall	Provides fully documented processes Easy to understand because of clear and defined stages Good choice for projects that have well-defined requirements	Documents processes fully before the development begins Not suitable for end-user applications or user-focused systems No formal way to make changes to the project as requirements almost always change
(Document-Driven)	Suited to classes of system development that focuses on documents such as compilers Has quick requirements gathering phase	Not a good choice for projects where requirements are not well-defined Process can be too fast, hence proper testing may not be done
Rapid Application	Uses feedback from the user to improve the application	Heavily dependent on user feedback
(Prototype-oriented)	Time-to-market is reduced by skipping many of the steps in traditional system development models Suitable for web-based applications	Resource requirements may be more
Iterative and Incremental	Delivers tangible results towards the final system Permits parallel development	Early increments may not be flexible enough to add increments or requirements Open to abuse as they can always improve upon it in future iterations
(Functionality-oriented)	High priority requirements and high risk part are usually done and delivered first	More resources may be required
Spiral	It begins even when all the system requirements are not known or understood Uses feedback from users to make sure the project is on track	Lacks controls that signal moves from one phase to the next There is Constant re-iterations thereby missing budgets and schedule
(Risk-driven)	Good choice for online customer support system Appropriate for situations where risk avoidance is a high priority	Encourages users to create unrealistic wish lists
Agile	Focuses on people Drives system development from a value focused-perspective	More about concepts and culture than the software practice Not controlled by conformance to plan
(Values-oriented)	Responsive to challenges that arise from unstable business environment Team communicates together on all aspects of the project	Not suitable for projects with larger teams

Table 8. The QuiQ System Development Methodology

Phases	Procedure	Approach	Output
Problem Analysis	<p>Planning phase starts with problem analysis. This is to gather sufficient information regarding the nature of the problem.</p>	<p>Problem analysis can be done using the functionalist approach. The QuiQ methodology adapts the use of cause and effect analysis to better identify the organizational or business problems.</p>	Cause and Effect Analysis
Requirements Analysis	<p>Once the root cause is identified, this directs you to the next step which is requirements analysis in order to identify the true business needs. The functional requirements are then identified and analyzed based from stakeholder interviews and business process documentation.</p>	<p>The QuiQ methodology adapts the use of multi-view approach (Kim et al., 2004) of Kim, J. et.al. with some modifications. The approach initiates with business description, followed by stakeholder interviews or focus group discussions, goal-setting and mapping of requirements. All requirements must be approved by all concerned stakeholders.</p>	Requirements Analysis Report which include: Business Profile Requirements Map (a table presenting the Stakeholders requirements, the corresponding goal and proposed solution)
Designing	<p>To satisfy the requirements earlier identified, the next phase to be undertaken is designing. This phase is in parallel with the requirements analysis. It defines the web-interface design, database design and architectural design.</p>	User-centered Approach	Web-interface Design Database Design Architectural Design showing the Client-Server Model
Building and Testing	<p>This phase is again accomplished in parallel with the design. It is done module (component) by module. Each module is built and tested at least every day. Once the modules are finished, it will then be integrated and tested as a single application. Then, implemented in the web and tested again.</p>	<p>Daily Build and Smoke Test approach by Microsoft is adapted in this phase where every file is compiled, linked, and combined into an executable program every day, and the program is then put through a smoke test, a relatively simple check to see whether the product smokes when it runs [24].</p>	Fully-Functional E-business Application
Changeover	<p>This is the final phase where the new application is introduced; user training is conducted ready for changeover to the new application.</p>	Reliable and Acceptable E-business Application	Well-trained users

Table 9. Strengths of Existing Methodologies Adopted by QuiQ

SD Methodology	Strengths	QuiQ
Waterfall (Document-Driven)	<p>clear and defined stages</p> <p>well-defined requirements</p>	<p>QuiQ offers well-defined phases with specified outputs on each phase so that systems analysts and designers are easily directed on what document to prepare.</p>
Rapid Application (Prototype-oriented)	<p>quick requirements gathering phase</p> <p>uses feedback from the user to improve the application</p> <p>has skipped many of the steps in traditional system development models</p>	<p>QuiQ has requirements analysis phase that focuses on the stakeholders requirements through focus group discussions.</p> <p>QuiQ was designed with parallel phases that include steps that are deemed important for e-business projects that are web-based in nature.</p>
Iterative and Incremental (Functionality-oriented)	<p>Permits parallel development</p> <p>High priority requirements and high risk part are usually done and delivered first</p>	<p>QuiQ offers parallel phases for continuous monitoring of requirements satisfaction.</p>
Spiral (Risk-driven)	<p>Uses feedback from users to make sure the project is on track</p>	<p>QuiQ offers parallel phases for continuous monitoring of requirements satisfaction.</p>
Agile (Values-oriented)	<p>Team communicates together on all aspects of the project</p>	<p>QuiQ offers parallel phases which allows for continuous communication among members of the development team</p>

Table 10. Weaknesses of Existing Methodologies Addressed by QuiQ

SD Methodology	Strengths	QuiQ
Waterfall (Document-Driven)	Documents processes fully before the development begins No formal way to make changes to the project as requirements almost always change	QuiQ was designed with parallel phases to allow updates on requirements Development begins just as problems and initial requirements are identified
Rapid Application (Prototype-oriented)	Process can be too fast, hence proper testing may not be done	QuiQ suggests for Daily Build and Smoke Test in order not to overlook the applications quality.
Iterative and Incremental (Functionality-oriented)	Early increments may not be flexible enough to add increments or requirements Open to abuse as they can always improve upon it in future iterations	QuiQ suggests for the preparation of Requirements Map through focus group discussions which allows all stakeholders to participate in the identification of the needed requirements. It also suggests for the approval of the concerned stakeholders before any requirements are added.
Spiral (Risk-driven)	Lacks controls that signal moves from one phase to the next There is constant re-iterations thereby missing budgets and schedule	QuiQ directs the development team to start with problem analysis phase before proceeding to the parallel phases skipping other phases to cut on budget and cost.
Agile (Values-oriented)	More about concepts and culture than the software practice Not controlled by conformance to plan	QuiQ was designed based on the surveyed practices of system development teams as well as significant literature and studies. All requirements are mapped and should therefore be implemented as planned.

Sousa, S. (2009). The advantages and disadvantages of waterfall software development. <http://www.my-project-management-expert.com/the-advantages-and-disadvantages-of-waterfall-software-development.html>.