SYSTEM DESIGN, DEVELOPMENT & DEPLOYMENT USING RAPID BY-CUSTOMER DEMAND WITH BUSINESS PRINCIPLES APPROACH

Felipe P. Vista IV [1], ***Kil To Chong*** [2]
[1]Department of Electronics Engineering, Jeonbuk National University, Jeonju City,
561-756 South Korea
[2]Advanced Research Center for Electronics and Information, Jeonbuk National University, Jeonju City,
561-756 South Korea
Email: [1] boduke@jbnu.ac.kr, [2] kitchong@jbnu.ac.kr

ABSTRACT

We present a methodology employed to address the technical as well as business process needs in rapidly designing, developing and deploying systems. The system design methodology is composed of three major parts labeled as the executive decision, development and finals in which the concept of systems features analysis, business process criteria and delivery schedule as essential parts of the methodology is also presented. The proposed methodology is compared with existing formally defined methodologies though their principles, processes and practices. A marine information system that was developed using the proposed methodology is also presented to show the applicability and effectiveness of the proposed method.

Keywords: Software System Development, Programming, Customer Demand, Business Principles.

1.0 INTRODUCTION

The successful design, development and deployment of a system do not just involve the technical aspect but also the inclusion of industry considerations most especially in cases of business related information systems. There are several situations encountered and corresponding solutions provided by current researches such as the one that surveyed different approaches in solving the problem of application portability between different software and hardware platforms [1]. A recent research presented to the user the Capability Maturity Model Integration for improving software development process [2] while another paper proposed a creative engineering design method to increase the innovativeness of complex technical systems [3]. There are also a number of papers on developing kiosk applications in rapid manner with less programming errors [4], accelerate the process of learning from mistakes of an individual programmer by defect logging and defect data analysis [5], and the development of system components by using a mathematical model [6].

We present in this paper the Rapid, By-Customer Demand with Business Principles Approach (RBCDwBPA) which was developed to address the need for a technical, procedural and business answer to the demanding requirements of a client. This method is an improvement of the Rapid, Non-Formal and By-Customer-Demand Approach [7] mainly due to its more detailed methodology, features analysis, list of criteria and delivery scheduling. RBCDwBPA aims to lessen if not remove the issue of application portability between different software and hardware platforms at the design and development level itself. RBCDwBPA presented the introduction of business principles for improving software development process and includes business principles in the development process to increase the effectiveness of the system. We address the errors and mistakes by trying to avoid the mistakes in the design stage itself while the components are designed and developed based on their features through system features analysis. We see the need to focus on the specific needs and requirements of the clients instead of pushing our own products to them [8] and at the same time produce software engineers who are not just technically proficient but also capable of delivering sound operations, business consultancy and financial services [9] to contribute for the eventual success of an information technology project they are involved in. While simulators [10] and a specific programming language for specific group of students [11] are taught to students to improve the learning curve we advocate inclusion of business principles that would be useful in the real world.

The system design and development with systems features analysis of the proposed methodology is presented first in Section 2. The customer demand and processes aspect composed of the criteria components is then discussed in Section 3 followed by the delivery scheduling aspect in Section 4. A comparison of the principles, processes and practices of the proposed methodology with other existing ones is then listed in Section 5. The application of the proposed methodology and its resulting output is given in Section 6 followed by the concluding remarks.

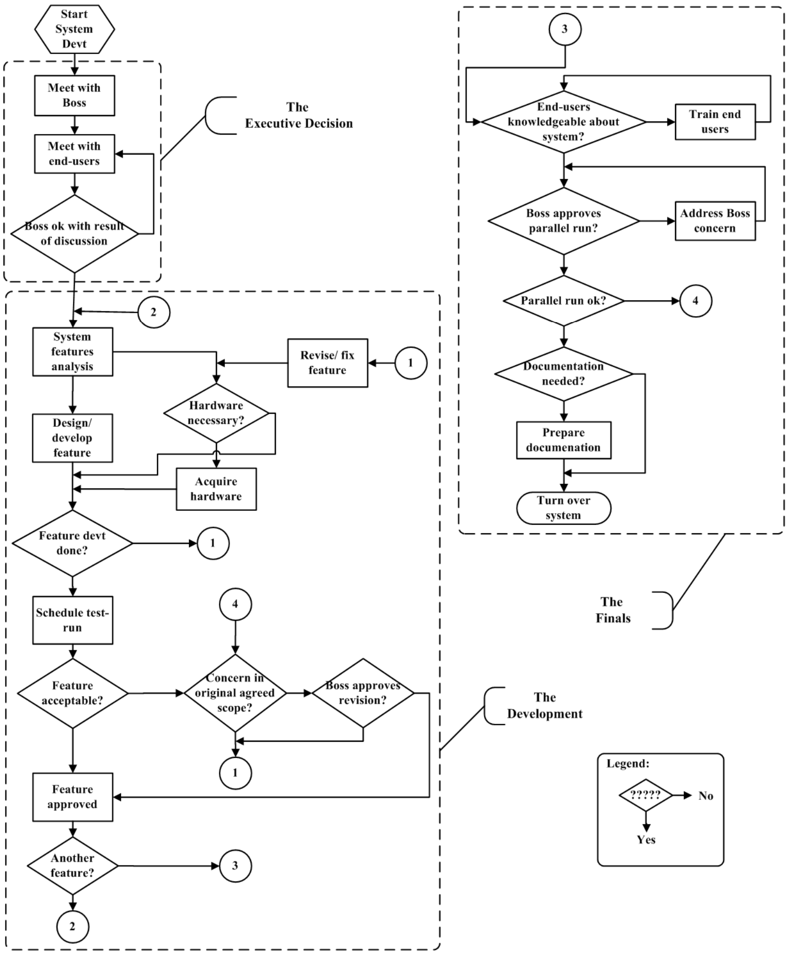


Figure 1. Design, development and deployment flowchart.

2.0 DESIGN AND DEVELOPMENT ASPECT

2.1 System Design and Development

The process flow of the proposed methodology is presented in Fig. 1, which is divided into three major stages namely: (1) the executive decision, (2) the development, and (3) the finals.

2.1.1 The Executive Decision

At the start of the system development using the method being proposed, the developer has to discuss with the owner or the executive sponsor about the salient features of the system that will serve as the guide as to how the future system would look like. The end users will then be the one to fill up the specific details for the system, upon which the owner or executive sponsor must still be consulted afterwards if the information received is in accordance with what is prescribed. This is an important aspect in using the methodology since it is possible that the business process being done by the end users is not in line with what the executive sponsor actually wants.

2.1.2 The Development

The developer can then proceed into dissecting the proposed system by doing specific features analysis which is explained later. Development of the specific feature occurs in tandem with the acquisition and configuration of any necessary hardware peripherals for this particular feature. A test-run for the specific feature that has just been completed will then be scheduled to check for the acceptability and effectiveness of this particular feature in which the developer usually deals with the end users when evaluating the features developed. The feature would then be deemed acceptable if it passes the criteria set beforehand by the end users and the executive sponsor. This feature evaluation part varies greatly depending on its usage, scope, and degree of usefulness with regards to the whole system. Any concern or problem that may be brought up during the feature evaluation process will be duly noted by the developer who will check if the concerns part of the scope of the initial system design. The developer must discuss with the owner or executive sponsor a particular concern if it is deemed already out of the scope of the initial system design before proceeding to address the said concern. This process is done repeatedly until all the features have been developed, evaluated and approved by the end users.

2.1.3 The Finals

The project would now move into the third or final phase of the proposed methodology which starts by checking how knowledgeable the end users are in using the developed system. This is to decide if the end users need additional training aside from their hands on experience while evaluating per feature and how intensive the users training should be. When the end users are knowledgeable enough, it is then time to discuss with the owner or the executive sponsor the conduct of a "parallel run".

The main difference between a parallel run and the per feature evaluation is the scope of the process involved and the audience which in this case includes the owner or executive sponsor aside from the end users. Successful parallel run after addressing any concerns that may arise will then lead to the provision of documentations if there is a need. The project is then termed officially finalized and completed with the turnover of the system. The developer is advised to keep the notes and source codes used in developing the system since it is possible that modifications might be required after the turnover that would require the services of the original developer.

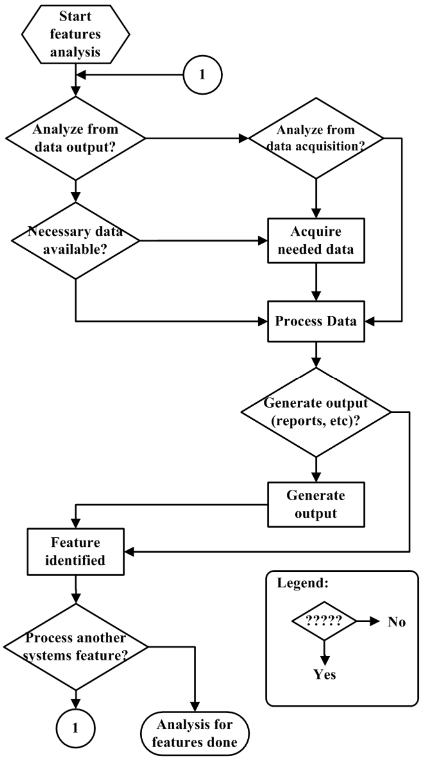

Start features analysis
1
Analyze from data output?
Analyze from data acquisition?
Necessary data available?
Acquire needed data
Process Data
Generate output (reports, etc)?
Generate output
Feature identified
Legend:
?????
No
Yes
Process another systems feature?
1
Analysis for features done

Figure 2. System features analysis flowchart

2.2 System Features Analysis

The sub process flow of the system features analysis as presented in Fig. 2 is used in dissecting the proposed system and decomposing it into components or "features" as defined. The systems features analysis process begins if a particular component should be studied based on the target output or from the data acquisition point of view. If the analysis would start from the generated output, is the data needed to generate the specified output already available? When the data needed is not yet available, data acquisition process for the particular output should be specified. The processing of data that must be done to generate the specified output is then analyzed to complete the identification of a specific feature.

When the system feature analysis has to start from the data acquisition, the developer has to design how the necessary data will be acquired, the data processing to be done, and output generation if there is a need. Another possible way of analyzing the specifics of a particular feature is when it is not needed to acquire new data or to start from the generated output but by using the existing data that is known to be readily accessible and available. It is almost similar as the second procedure but the "acquire data" step is skipped. The whole process is done repeatedly until all the necessary features for the proposed system has been identified.

3.0 CUSTOMER DEMAND AND PROCESSES ASPECT

System development for small and medium sized enterprises especially in developing countries for improving their business processes and increasing profits are usually done by a small group of persons. These people designated to implement the project are chosen not just by their skills in system development but also due to recommendations from personal contacts of the business owners themselves. Implementing information technology projects on small, medium and large government institutions and local government units (LGUs) has its own unique twists especially with the autonomous state of the local government units such as barangays (villages), municipalities (town), cities and provinces from the national government in managing their affairs of state. The biggest concerns for an e-Government project are great differences between government and private firms, initial plans are too ambitious and financially unsound, and lack of experienced project managers [12]. It should be noted that government institutions and LGUs are mentioned here since they can be considered as utilizing business principles in their operations most especially the inherent power of the executives to start, revise, continue or stop the system development project at any time.

Going to the methodology itself, one has to answer the “two when’s”: *(1) When can they do a dry-run or parallel implementation to check if the system complies with the requirements specified?*, and *(2) When can they fully-implement the new system?* These questions can be answered by addressing the following criteria: *(a) Few or no information technology experts from the client’s organization*, *(b) Salient features required of by the executive sponsor or owner*, *(c) Utilization of general layout and design of current reports and certificates*, *(d) Details of process from the end users but with approval of executive sponsor or owner*, *(e) Timetable as short as one month with visible progress*, *(f) Immediate costs of hardware, software, peripherals including consumables*, *(g) Maintenance costs*, and *(h) Training and Education*. This criteria set were the result of meetings and discussion with several people from the executive sponsor, end-users, technical and developers group who had experience in information system projects both from the public and private sector. The succeeding parts discuss in detail the criteria that will be used in trying to come up with an effective scheduling for the conduct of a dry-run and the full-implementation.

3.1 Dealing With The Criteria

3.1.1 Few or no information technology experts from the client’s organization

The client is a small or medium sized organization with little or no Information Technology experts on its staff who are capable of delivering the required system, the developer tapped to do the proposed system has to deal with either low expectations or very high expectations of the executive sponsor and end users. Data gathering and studying the processes when there are low expectations is tedious and patience and understanding in dealing with persons directly involved most especially if they feel threatened with the way they normally do their work and change is imminent. Low expectations would seem at first hand to be preferable but there actually exists the pressure to deliver an excellent working system that would make the end users feel not left-out in the computerization of a particular business process which they are involved in. Having to deal with very high expectations is altogether another issue in which the developer’s immediate responsibility to temper down these expectations to manageable and deliverable levels before going into the actual development process itself.

3.1.2 Salient features required of by the executive sponsor or owner

The executive sponsor has specific design specifications and content on reports generated that matters. They are very specific on the fonts to be used, the size of the font, the page layout (portrait or landscape), whether to fit everything into a single page or allow multiple pages. The concept of “executive reports” which are out of reach of the other members of the organization is a common request that the developer must deliver through a user-level based system. The specifics on how to arrive at the data needed for the proper generation of the reports and certificates are left out for the developer to find out from existing reports, certificates and discussions. In dealing with this requirement, the developer is actually bound to follow all the specifics being laid down by the executive sponsor since they have the power to push-through, put on hold or even totally scrap the project when they feel that what they want cannot be delivered. This is one reason that some lose a project already in the initial discussion phase because of the developer’s reluctance to follow the request of the executive sponsor or owner. This part of dealing with the client does not just require technical skills but also good interpersonal skill in dealing with the executive sponsor who are used to getting it their way.

3.1.3 Utilization of general layout and design of current reports and certificates

The client requires that existing format and layout of the receipts, reports, and certificates currently in use will be adapted into the proposed system. They cover a broad range of customization from the physical layout, font details, single or multiple pages of the reports, memorandums or other document types that the system should be able to generate. The developer will do well to take note of all the customizations that is required so as to faithfully comply and replicate the ones that are in use. It must be accepted as a fact by the developer that what he/ she sees and understands about the business process is most probably not exactly in tune with what the client sees.

3.1.4 Details of process from the end users but with approval of executive sponsor or owner

The internal operations of the proposed system can be garnered from the end users by studying the reports and procedures currently in use. The end users and other people involved needs to be interviewed and directly observed as to how they perform the process itself. It is highly recommended for the developer to be "embedded" in the organization to get a full grasp and idea of the intricacies and details involved, even the subtle ones, from the start to the end of the whole business process. The main reason of being embedded is the possibility that parts of the business process can be clearly defined only through firsthand experience that the current reports/ outputs being used cannot explain. Any and all questions and concerns that the developer encounters with regards to the proposed system must be properly answered in detail by the concerned persons involved and it is possible for the developer to ask the executive sponsor if the answers received are not satisfactory.

3.1.5 Timetable as short as one month with visible progress

Since the timetable given by the executive sponsor is sometimes as short as one month, it is imperative that the developer find out at the earliest possible time the priority and core operations of the proposed system. To be able to present and deploy a working module or part of the system that caters to the need of the end users (being able to encode information and generate the report) and the executive sponsor (generate production output per user for a particular period) at the same time is a considerable achievement in terms of the project timetable. The developer must be constantly aware that the order of development and delivery based on their importance can be revised or completely re-arranged depending upon the requirements of the executive sponsor or when the need for a particular feature arises such as a more stringent and feature-specific user-level capability.

3.1.6 Immediate costs of hardware, software, peripherals and consumables

The purchase of any hardware, storage devices, input devices, peripherals, networking devices or sensors entails the developer must present to the executive sponsor the details and specifications of each including the projected purchase cost as well as the availability, and it's corresponding advantages and disadvantages. A thorny issue when it comes to the design, development and deployment of an information system is the costs incurred when utilizing software or tools that needs to be licensed for commercial use especially when there is a need for database system or biometrics. It takes some time and considerable effort from the developer to patiently explain why the need for software licensing if they are not aware about it. The consumables must be analyzed and discussed with the end users and the executive sponsor who would have to decide eventually about its cost effectiveness.

3.1.7 Maintenance costs

Another aspect of the proposed information system that is considered low priority from the point of view of a developer but of vital importance to the executive sponsor is the issue on maintenance costs such that for electricity usage, internet connectivity, pc/ hardware and communication expenses. The cost of electricity usage could refer to the power consumption of a server (database or web) which would need to be up and running at the least during office hours or even on a 24-7 basis. This could translate to huge electricity bills especially in places where the power rate is high. Running servers, some hardware, and peripherals, requires that they be kept and stored in temperature controlled rooms for optimal performance. Internet connectivity usage might arise when the system being proposed is web based such as a website which can be hosted on either their own web server or on a web-hosting service which either way requires internet connectivity.

3.1.8 Training and Education

From the testing of the initial modules to the delivery of the final version, continuous training and education about the information system is given to the end users while brief updates are given to the executive sponsor mostly through reports generated and screen captures. They will be the one to demonstrate the system to the executive sponsor for final evaluation of approval.. The end users are also educated on the basics of maintaining the system such as the process of regular backup of data, proper booting up and down of the operating systems as well as the very important aspect of following the proper procedure in running a particular service such as a database management system so as to avoid corrupting some parts or all of the data. The executive sponsor has their own orientation on the salient features of the information system which is not accessible to the regular users of the system. It is good practice that the developer ask beforehand, if the salient features that the executive sponsor requires be made known to the other users of the information system even if they cannot access it so as to avoid antagonizing the executive sponsor who have reasons why the salient features should be kept a secret or not.

4.0 DELIVERY SCHEDULING

4.1 When can they do a dry-run to check if the system complies with the requirements specified?

A dry run of the proposed system with the existing one is an integral step to be able to check the effectiveness and performance of the new system as against the existing one. Careful analysis would show that the criteria affecting the conduct of a parallel run are: (a) Few or no information technology experts from the client's organization; (b) Salient features required of by the executive sponsor or owner; (c) Utilization of general layout and design of current reports and certificates; (d) Details of process from the end users but with approval of executive sponsor or owner; (e) Timetable as short as one month with visible progress; (f) Immediate costs of hardware, software, peripherals including consumables; and (h) Training and Education.

4.2 When can they fully implement the system?

The final big question of when the complete system can be finally implemented and fully deployed depends in part on the result of the conduct of the dry run and the following criteria: (b) Salient features required of by the executive sponsor or owner; (g) Maintenance costs; and (h) Training and education.

Criteria (b) and (h) are utilized for both the conduct of a dry-run and full implementation of the system since it is always possible that the executive sponsor or boss can ask for any additional salient feature for the system which needs to be accommodated due to their inherent power to push through, modify or even cancel the project.

5.0 PRINCIPLES, PROCESSES, PRACTICES

Our development methodology is now compared in Table 1 to several system development methodologies such as Waterfall [13], Prototyping [14], Incremental development [15], Spiral development [16], Rapid Application Development [17], Agile [18, 19], Lean [20, 21] and Dynamic Systems Development Method (DSDM) [22, 23] in terms of their operating principles.

Table 1. Comparison of the operating principles between several well-defined methodologies and our method.

Methodology	Principles
Waterfall	divide project into phases (planning, schedules, budgets, implementation), with strict control
Prototyping	involves end user, development of prototypes, understanding fundamentals of business problem to solve the right problem
Incremental development	successive small versions of waterfall , definition of system specifications then sub systems, original system specifications, design via waterfall development through prototyping
Spiral development	emphasis on continuous evaluation of risks, involve same process for every aspect of the system
Rapid Application Development	fast delivery quality system at low cost, minimize risks , emphasis on meeting business requirement instead of technical excellence, requirements are reduced to meet the deadline instead of extending it, active user involvement
Agile	fast delivery of usable system, changes accepted fact, frequent delivery, daily cooperation business people and developers, motivated individuals with support, face to face communication, working software, constant pace, technical excellence, simplicity, self organizing teams, adapt behavior
Lean	elimination of waste, amplify learning, decide late, deliver fast, team empowerment, build integrity, and see the whole
DSDM (Atern)	business need, on time, collaborate, never compromise quality, build incrementally from firm foundations, develop iteratively, communicate continuously and clearly, control
RBCDwBPA	***technical excellence and customer satisfaction (processes and business need)must go hand in hand, always check with the decision maker first, identify core processes, involve decision maker and end-user***

The process and practices of Waterfall, Agile-XP, Agile-Scrum, Crystal Family, Feature Driven Development, Rational Unified Process, Dynamic System Development, Adaptive software development, Open-source software, RAD and Lean are also described with that of our own method in Table 2.

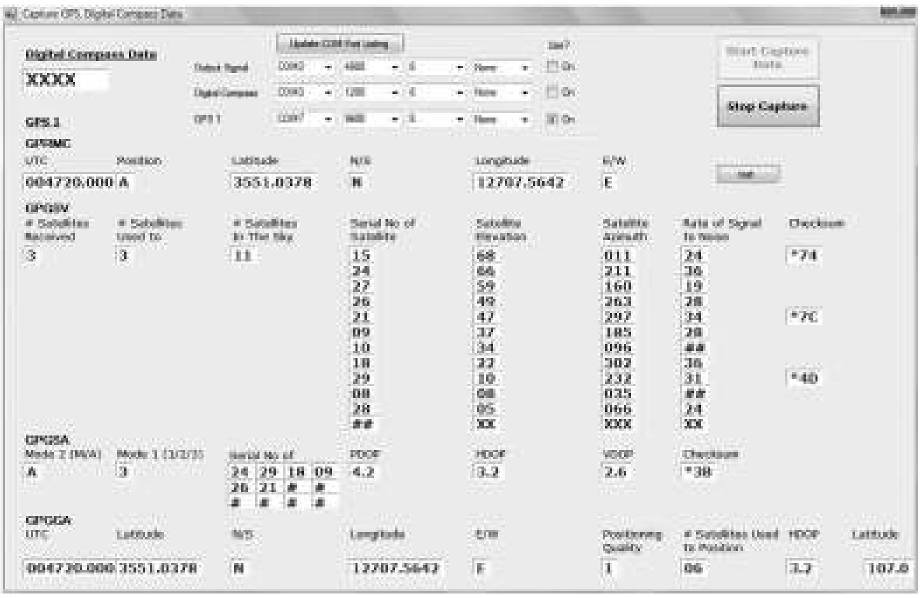
Table 2. Comparison of the process and practices between several well-defined methodologies and our method.

Methodology	**Process**	**Practices**
Waterfall	System feasibility, Requirements analysis, System design, Coding & unit testing, Integration & system testing, Deployment & maintenance	Program requirements set in stone before design begins, Emphasis on documents, Structured approach , Easily identifiable milestones
Agile-XP	Exploration, Planning, Iterations to Release, Productionizing, Maintenance and Death	Planning game, Small/ short releases, Metaphor, Simple design, Testing, Refactoring, Pair Programming, Collective ownership, Continuous integration, 40 hr work week, On site customer, Coding standards, Open workspace, Just rules
Agile-Scrum	Pre-Game, Development, Post-Game	Product backlog, Effort estimation, Sprint
Crystal Family	Policy Standards, Work Products, Local Matters, Tools, Standards, Activities	Staging, Revision & review, Monitoring, Parallelism & flux, Holistic diversity strategy. Methodology-tuning technique, User viewings, Reflection workshops
Feature Driven Development	Focused on design & building phases rather than cover entire software development process Develop over-all model, Build a features list, Plan by feature, Design by feature, Build by feature	"Best practices" – Domain object modeling, Developing by feature, Individual class (code) ownership, Feature teams, Inspection, Regular builds, Configuration management, Progress reporting
Rational Unified Process	Inception, Elaboration, Construction, Transition	Develop iteratively, Manage requirements, Component-based architectures, Visual model, Verify quality, Control changes
Dynamic Systems Development	Pre-project, feasibility, foundations, exploration, engineering, deployment, post-project	Active user, Teams empowered, Frequent delivery, Fitness for business purpose, Iterative & incremental development, Revertible changes, Baselined requirements, Integrated testing , Collaborative & cooperative
Adaptive Software Development	Speculate, Collaborate, Learn phase cycles	Iterative development, Feature-based (component-based) planning, Customer focus group reviews
Open-source software (OSS)	Problem discovery, Finding volunteers, Solution identification, Code development & testing, Code change review, Code summit & documentation, Release management	OSS systems built by a number of volunteers, People choose work, No explicit system-level design, No project plan, schedule or list of deliverables
RAD	Requirements Planning, User design, Construction, Implementation	Prototyping, Iterative development, Time boxing, Team members, Management approach, RAD Tools

Lean	- focus on project management aspect and specifies no technical details	Seeing waste, value, stream mapping, feedback, iterations, synchronization, set-based-development, options thinking, last responsible method, making decisions, pull systems, queuing theory, cost of delay, self-determination, motivation, leadership, expertise, perceived integrity, refactoring, testing, measurements, contracts
RBCDwBPA	***The executive decision, the development, the finals***	***The two when's answerable by a set of criteria***

6.0 APPLICATION AND RESULTS

The proposed methodology was used in the design and development of a Marine Vehicle Monitoring System (MVMS) initially started as a utility for polling, identifying and segregating data from serial devices such as the digital compass (DC) and global positioning system (GPS) as shown in Fig.3. The MVMS was requested by the executive sponsor with a rough description of the system features such as ship display, map display, GPS and digital compass. The details of the desired features were then discussed with the technical persons and other involved personnel.whose results were then presented to the designated project leader for approval. Systems features analysis was then performed based on the initial given list of features from the previous step. The derived list of features after using SFA increased to six (6) major components from the original three (3) namely : (1) Pre-processing of raw electronic navigational chart (ENC) data, (2) Base-map processing, (3) Map details, (4) Ship representation and plotting, (5) GPS data processing and (6) Digital compass data processing. The components were designed and developed per feature in the order they were given. The resulting system was then presented to the project leader for testing purposes and evaluation in which preliminary system tests resulted to some major changes as per the project leader's instructions. These changes were the removal of waypoints and path-planning and then the inclusion of device connection configuration, variable shaped ship representation and base map rotation, This demonstrated the flexibility of the RBCDwBPa with regards to client's continuous evolving requirements and delivery schedule by focusing on the core operations and processes. The utility shown in Fig. 3 was incorporated into the new system. The data acquisition setup utilized a serial-to-USB adapter in order to address the need for polling other locational or proximity devices such as GPS and digital compass. The use of the serial-to-USB adapter was decided upon by going through the criteria set. The following figures show the initial (Fig. 4.a, from middle of 2009) and final (Fig. 4b, early 2010) main window display for the MVMS.



Digital Compass Data

XXXX

					Use?
Output Signal	COM2	4800	8	None	□ On
Digital Compass	COM3	1200	8	None	□ On
GPS 1	COM7	4800	8	None	☒ On

GPS 1

GPRMC

UTC	Position	Latitude	N/S	Longitude	E/W
004720.000	A	3551.0378	N	12707.5642	E

GPGSV

# Satellites Received	# Satellites Used to	# Satellites In The Sky	Serial No of Satellite	Satellite Elevation	Satellite Azimuth	Rate of Signal to Noise	Checksum
3	3	11	15	68	011	24	*74
			24	66	211	36	
			27	59	160	19	
			26	49	263	28	
			21	47	297	34	*7C
			09	37	185	28	
			10	34	096	##	
			18	22	302	36	
			29	10	232	31	*40
			08	08	035	##	
			28	05	066	24	
			##	XX	XXX	XX	

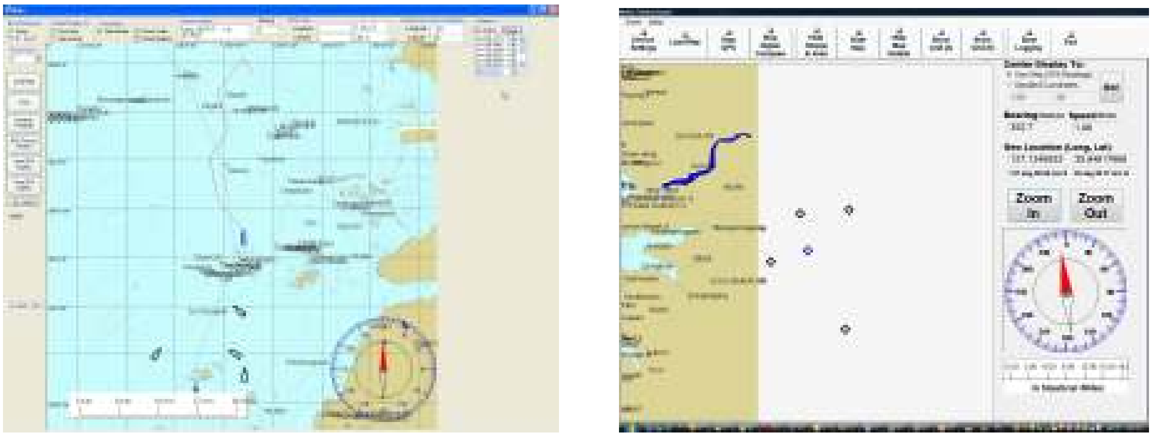
GPGSA

Mode 2 (M/A)	Mode 1 (1/2/3)	Serial No of				PDOP	HDOP	VDOP	Checksum
A	3	24	29	18	09	4.2	3.2	2.6	*38
		26	21	#	#				
		#	#	#	#				

GPGGA

UTC	Latitude	N/S	Longitude	E/W	Positioning Quality	# Satellites Used to Position	HDOP	Latitude
004720.000	3551.0378	N	12707.5642	E	1	06	3.2	107.0

Figure 3. GPS, DC Data polling and display window



(a) (b)

Figure 4. Main display window as of: (a) late 2009 and (b) mid 2010

Given in Fig. 5 is the system flowchart which was formulated by using Rapid By-Customer Demand with Business Principles Approach. They are the device connection, base map orientation display, GPS data polling, digital compass data polling, ship representation, map details display, map grid (4 or 8) display, logging, display view switching.

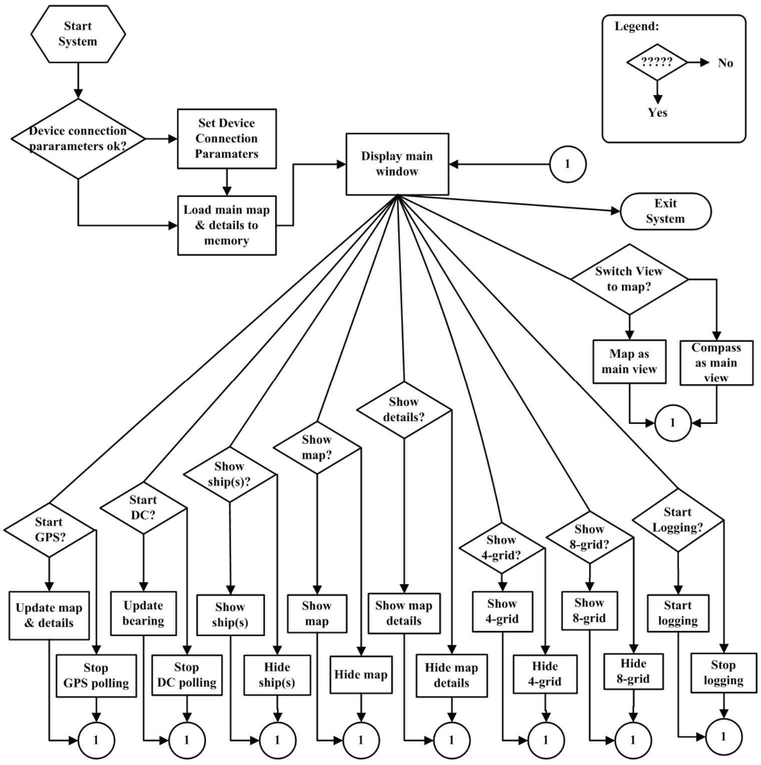


Figure 5. Complete system flow of the MVMS.

7.0 CONCLUSION

This paper presents a method of developing software systems through RBCDwBPA by combining technical engineering with business principles to be able to effectively design, develop and deliver a working system rapidly for a particular software development project as per the requirements of the client. As one paper presented a teaching methodology to bridge the gap between technology found in traditional text books and current industry standards [24] we introduce our methodology that infuses the business principles into the software development process itself.

The system developed will be used as a template or reference for further developments of marine systems such as a navigation system with collision avoidance and path planning or for keeping track of vehicles in a specified area. The capability to port additional devices for input or output of information is made easier due to its modular design based on system features analysis. With regards to the effect of using RBCDwBPA in designing, developing and deploying systems, it is envisioned to assist the developer in dealing with the software system aspects of the project as well as its equally critical business and financial aspects. If information system (IS) managers are more focused on organization, productivity and development quality while system developers preferred verification and validation [25] then our method attempts to inculcate these traits into the developer in preparation for him/ her to move up from being a systems developer to an IS manager. It goes by saying that we would like to help produce software engineers with business sense that matters.

ACKNOWLEDGMENTS

This work was partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0002434).

REFERENCES

[1] D. Silakov and A. Khoroshilov, "Ensuring Portability of Software," *Programming and Computer Software,* vol. 37, 2011, pp. 41-47.

[2] M. Y. al-Tarawneh, M. S. Abdullah, and A. B. M. Ali, "A Proposed Methodology for Establishing Software Process Development Improvement for Small Software Development Firms," *Procedia Computer Science,* vol. 3, 2011, pp. 893-897.

[3] E. Carayannis and J. Coleman, "Creative System Design Methodologies: The Case of Complex Technical Systems," *Technovation,* vol. 25, Aug 2005, pp. 831-840.

[4] Z. Zivanov, P. Rakic, and M. Hajdukovic, "Using Code Generation Approach in Developing Kiosk Applications," *Computer Science and Information Systems,* vol. 5, Jun 2008, pp. 41-59.

[5] L. Prechelt, "Accelerating Learning from Experience: Avoiding Defects Faster," *IEEE Software,* vol. 18, Nov-Dec 2001, pp. 56-61.

[6] M. Broy, "A Logical Basis for Component-Oriented Software and Systems Engineering," *Computer Journal,* vol. 53, Dec 2010, pp. 1758-1782.

[7] F. Vista, IV, K. Chong, and C. Lee, "An Experience Report on Systems Development and Design Using a Rapid, Non-Formal and by-Customer-Demand Approach", in *Proc. Software Engineering and Practice*, Las Vegas, 2009, pp. 192 - 194.

[8] J. Galbraith, R., "Organizing to Deliver Solutions," *Organizational Dynamics,* vol. 31, 2002, pp. 194-207.

[9] A. Davies, "Moving Base into High-Halue Integrated Solutions: A Value Stream Approach," *Industrial and Corporate Change,* vol. 13, Oct 2004, pp. 727-756.

[10] S. Aydin and H. Kaptan, "Learning by Developing a Satellite Navigation Simulator," *Computer Applications in Engineering Education,* vol. 19, 2011, pp. 501-507.

[11] J. A. Orosa García, "Programming Languages for Marine Engineers," *Computer Applications in Engineering Education,* vol. 19, 2011, pp. 591-597.

[12] H. Buhl and M. Löffler, "The Role of Business and Information Systems Engineering in E-Government," *Business & Information Systems Engineering,* vol. 3, 2011, pp. 341-344.

[13] W. Royce, "Managing the Development of Large Software Systems," *IEEE WESCON Proceedings,* August 1970, pp. 01-09.

[14] R. B. K. Dewar, E. Fisher, Jr., E. Schonerg, R. Froehich, S. Bryant, C. Goss, and M. Burke, "The Nyu Ada Translator and Interpreter," *ACM SIGPLAN Notices - Proceedings of the ACM-SIGPLAN Symposium on the Ada Programming Language,* vol. 15, 1980, pp. 194 - 201.

[15] C. Larman and V. R. Basili, "Iterative and Incremental Development: A Brief History," *Computer,* vol. 36, Jun 2003, pp. 47 - 56.

[16] B. W. Boehm, "A Spiral Model of Software-Development and Enhancement," *Computer,* vol. 21, May 1988, pp. 61-72.

[17] P. Beynon-Davies, C. Carne, H. Mackay, and D. Tudhope, "Rapid Application Development (Rad): An Empirical Review," *European Journal of Information Systems,* vol. 8, Sep 1999, pp. 211-223.

[18] F. K. Y. Chan and J. Y. L. Thong, "Acceptance of Agile Methodologies: A Critical Review and Conceptual Framework," *Decision Support Systems,* vol. 46, Mar 2009, pp. 803-814.

[19] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. (2002, February 8). *Principles Behind the Agile Manifesto*. Available: http://agilemanifesto.org/principles.html

[20] M. Poppendieck and T. Poppendieck, "Introduction to Lean Software Development - Practical Approaches for Applying Lean Principles to Software Development," *Extreme Programming and Agile Processes in Software Engineering, Proceedings,* vol. 3556, 2005, pp. 280-280.

[21] M. Poppendieck and T. D. Poppendieck, *Lean Software Development : An Agile Toolkit*. Boston: Addison-Wesley, 2003.

[22] J. Stapleton and DSDM Consortium., *Dynamic Systems Development Method : Version 3*. Ashford, UK: Tesseract Pub. on behalf of the DSDM Consortium, 1999.

[23] D. Consortium. (2008, February 8). *Atern: The Handbook*. Available: http://www.dsdm.org/atern-handbook/flash.html#/18/

[24] G. Wang, "Bridging the Gap between Textbook and Real Applications: A Teaching Methodology in Digital Electronics Education," *Computer Applications in Engineering Education,* vol. 19, 2011, pp. 268-279.

[25] M. Huisman and J. Iivari, "Deployment of Systems Development Methodologies: Perceptual Congruence between Is Managers and Systems Developers," *Information & Management,* vol. 43, Jan 2006, pp. 29-49.

BIOGRAPHY

Felipe P. Vista IV received his Master of Engineering degree in Computer Engineering in 2005 from Western Institute of Technology, Iloilo City, Philippines. He is currently a Ph.D. student of Network System Control Laboratory at Jeonbuk National University, South Korea. His research interests include systems, networks and fuzzy logic. (E-mail boduke@jbnu.ac.kr)

Kil To Chong ***(corresponding author)*** received his Ph.D. in mechanical engineering from Texas A&M University, College Station in 1995. He is currently a Professor at the School of Electronics and Information Engineering, Jeonbuk National University, Korea and Head of the Mechatronics Research Center granted from the Korea Science Foundation. His research interests are motor fault detection, network system control, time-delay systems and neural networks. (E-mail kitchong@jbnu.ac.kr)