FISHBONE ANALYSIS ON WASTES IN SOFTWARE DEVELOPMENT USING THE LEAN I.T. PRINCIPLES

Daniel D. Dasig, Jr.[1], Arlene Mae C. Valderama[2], Mengvi P. Gatpandan[2],
Mary Ann Bernadette Taduyo[2], Roel Traballo[2], Shirley M. Pangan[2],
and Paulino H. Gatpandan[3]

[1]Computer Engineering Department, Jose Rizal University, Philippines
[2]Information Technology Department, Jose Rizal University, Philippines
[3]College of Sciences, De La Salle University, Dasmarinas, Cavite, Philippines

ABSTRACT

The transformative global economy posed challenges to businesses in service management. In this computing age, the perceptual and operational edge of a certain business or organization manifested on the kind of technology it offers in the Service Management. Organizations have long recognized the importance of managing key resources such as people and information. Information has now moved to its rightful place as a key resource in the organization and therefore management of the same can be instituted by employing methodology. To keep their brand promise, technology has been used;The number of new entrants to every sectors of economy has grown significantly in recent years, and each firm strives to make their daily operation efficient in which demand for business software or application software getting higher and businesses or organizations opted to build or buy this software. Because of new entrants, it had offered opportunity to software developers to translate business processes into systems. This study investigates waste in the software development by application of Lean principles. Like any conventional projects, software becomes buggy and oftentimes it fails. Software failure is always attributed to the software engineering, not the incompetence of project managers, inadequacy of the people on the project, or lack of clear goal. The researchers' contentions are there wastes in the software development and serve as mechanism and evidence to why software fails. Software failure is not attributed to the software itself, it includes however the acceptance of the clients and end-users. Descriptive secondary data analysis, participant observation and Fishbone Analysis were the methodology used in the study. Wastes include unfinished or partially done work, extra features, relearning, handoffs, delays, task switching, and defects.

KEYWORDS

Deadly Wastes, Lean IT, Lean Manufacturing, Lean Principles, Software Development

1. INTRODUCTION

In this computing age, the perceptual and operational edge of a certain business or organization manifested on the kind of technology it offers in the Service Management. For businesses and other sectors of economy, information system has been a necessity in maintaining market position even there are new entrants on their strands of products and services equated in this transformative global economy. Organizations have long recognized the importance of managing key resources such as people and information. Information has now moved to its rightful place as a key resource [1] in the organization and therefore management of the same can be instituted by employing methodology. In allowing free-flow of information, information system must be aligned with the business processes, infrastructure and business service methodology also known as for information technology management by aligning IT services and the IT infrastructure supporting those services with business processes [2].To obtain

DOI : 10.14810/ecij.2014.3403

operational excellence, companies utilize application and business software to perform tasks, from keeping track of inventory and billing customers to maintaining accounts, speeding up business process and, in fact, even improving application effectiveness [3], and to keep the brand promise alike, services and products shall serve the ***value*** that the customer is willing to pay for. It is where the challenges of assigning value to IT services becomes most granular and meaningful, while managing costs become service-centric versus component-centric[2] including their IT departments, view their activities as services that are provided to external and internal customers[4].

The number of new entrants to every sectors of economy has grown significantly in recent years, and each firm strives to make their daily operation efficient in which demand for business software or application software getting higher and businesses or organizations opted to build or buy this software. Because of new entrants, it had offered opportunity to software developers to translate business processes into systems. Business process is naturally complex and due to hyper-competitive economy its complexity is becoming extreme. Software development is complex and difficult due to intangible nature of product, requirements, quality and compliance measurement for effectiveness, and among others.

This study investigates waste in the software development by application of Lean principles. Like any conventional projects, software becomes buggy and oftentimes it fails. Software failure is always attributed to the software engineering, not the incompetence of project managers, inadequacy of the people on the project, or lack of clear goals [7]. The researchers' contentions are there wastes in the software development and serve as mechanism and evidence to why software fails. Software failure is not just on the software itself, it includes however the acceptance of the clients and end-users.

2. SOFTWARE DEVELOPMENT

Software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products [5] and is defined as the process of writing and maintaining the source code, but in a broader sense of the term it includes all that is involved between the conception of the desired software through to the final manifestation of the software, ideally in a planned and structured process [6]. Software is developed for a variety of purposes but the primary notion is to meet specific needs of the end-users. Software development lies on the application and the development processes based on the methodology and activities being employed.

Software has to conform to the generally accepted, tried and true quality tools which therefore the need for quality control with the use of a process-centric systematic approach in the engineering perspective called software engineering. Software engineering is a systematic approach to the analysis, design, assessment, implementation, test, maintenance and reengineering of software which is the application of engineering to software [7]. To countervail issues and constraints on the software development activities developers and software development team may employ Software Project Management which is defined as the science of planning and leading software projects [8]. Project is a temporary endeavor designed to produce a unique product, service or result [9]. Software Project Management is attributed as a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled [8]. In software development, there are task dependencies that the software development team has to consider. The following sub-sections provide walk-through of software development.

2.1.1 Software Development Methodology

Software development methodology defines the “how”.. it identifies how to perform activities for each period, how to represent activities and products, and how to generate the product [7]. Over the years, software development methodologies have evolved, software development methodology also attributed as SD Process, model or life cycle is a framework itself. It is used to structure, plan and control the processes of developing the software.

A software development team may use varied methodology because a methodology may not suitable for use by all projects or project activities. These approaches maybe more structured, engineering-based, incremental and others which suits to the specific kind of projects, requirements, organization and team needs which are the traditional and agile methodologies.

2.1.2 **Software Development Activities**

With a consideration to continuum in software development, the activities depend on the kind of methodology used. If the project team used SDLC, the 7 phases SDLC will be considered as project activities. Kendall & Kendall defined SDLC as “a phased approach to analysis and design that holds that systems are best developed through the use of a specific cycle of analyst and user activities” [10] which has been used by the developers. The figure 1 depicts the Stages of System Development Life Cycle. [10] SDLC is a framework that describes the activities performed at each stage of a software development project.

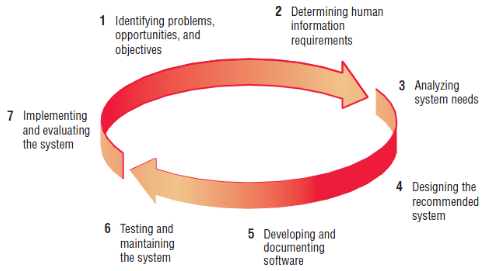

1 Identifying problems, opportunities, and objectives
2 Determining human information requirements
3 Analyzing system needs
4 Designing the recommended system
5 Developing and documenting software
6 Testing and maintaining the system
7 Implementing and evaluating the system

Figure 1. The seven phases of the systems development life cycle

Each phase has unique activities, input and outputs. Output of the preceding activities is necessary to start the current phase or succeeding phase, which therefore the task dependencies took place.

2.1.3 Software Development Project Management

Project management is the process and activity of planning, organizing, motivating, and controlling resources, procedures and protocols to achieve specific goals in scientific or daily problems. Project Management System helps to manage estimation and planning, scheduling, cost control and budget management, resource allocation and estimates, collaboration software, communication, decision-making, quality management and documentation or administration systems[11]

3. LEAN PRINCIPLES

Lean is a deviation from the perpendicular; or an inclination. Lean manufacturing, Lean Enterprise, or lean production, often simply, "lean", is a production philosophy that considers the expenditure of resources in any aspect other than the direct creation of value for the end customer to be wasteful, and thus a target for elimination [12]. Working from the perspective of the client who consumes a product or service, "value" is any action or process that a customer would be willing to pay for [12]. Lean production not only successfully challenged the accepted mass production practices in the automotive industry, significantly shifting the trade-off between productivity and quality, but it also led to a rethinking of a wide range of manufacturing and service operations beyond the high-volume repetitive manufacturing environment [13].

Lean is coined and accounted by many as the set of "tools" that assist in the identification and steady elimination of waste or muda. The muda is a Japanese word meaning "futility; uselessness; idleness; superfluity; waste; wastage; wastefulness" [14]. As waste is eliminated quality improves while production time and cost are reduced [12]. Lean aims to make the work simple enough to understand, do and manage. Lean implementation is therefore focused on getting the right things to the right place at the right time in the right quantity to achieve perfect work flow, while minimizing waste and being flexible and able to change [12]. Although the elimination of waste may seem like a simple and clear subject it is noticeable that waste is often very conservatively identified [15]. Non-value adding work is waste that must be done under the present work conditions. One key is to measure, or estimate, the size of these wastes, to demonstrate the effect of the changes achieved and therefore the movement toward the goal [15]. Because of waste elimination, the product provides now the value. Value is anything that the customer is willing to pay for, while waste is the opposite.

The Principles of Lean [18] with five-step thought process for guiding the implementation of lean techniques is easy to remember, but not always easy to achieve.

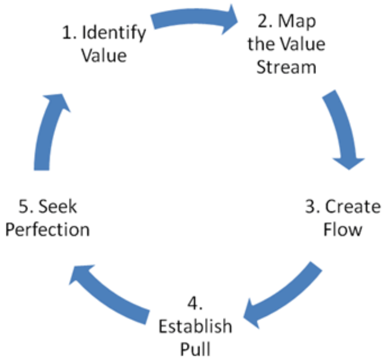


Figure 2. The Principles of Lean Implementation.

This Figure 2 depicts the lean implementation including; (a)Specify value from the standpoint of the end customer by product family, (b) Identify all the steps in the value stream for each product family, eliminating whenever possible those steps that do not create value, (c) Make the value-creating steps occur in tight sequence so the product will flow smoothly toward the customer, (d) As flow is introduced, let customers pull value from the next upstream activity, (e) As value is specified, value streams are identified, wasted steps are removed, and flow and pull are introduced, begin the process again and continue it until a state of perfection is reached in which perfect value is created with no waste.

The following are wastes in the production line which if eliminated, the product produces value. The wastes according to [15] include; (a) ***Transport*** (moving products that are not actually required to perform the processing), (b) ***Inventory*** (all components, work in process, and finished product not being processed) (c) ***Motion*** (people or equipment moving or walking more than is required to perform the processing), (d) ***Waiting*** (waiting for the next production step, interruptions of production during shift change), (e) ***Overproduction*** (production ahead of demand), (f) ***Over Processing*** (resulting from poor tool or product design creating activity), (g) ***Defects*** (the effort involved in inspecting for and fixing defects).

These wastes that are non-value-adding and deadly in the production lines can be controlled using the Lean Manufacturing. There are five principles of Lean Thinking… and when implemented the philosophy that will become "just the way things are done" [16] which include; (a) Identify Customers and Specify Value, (b) Identify and Map the Value Stream, (c) Create Flow by Eliminating Waste, (d) Respond to Customer Pull, and (e) Pursue Perfection. Lean is about more than just cutting costs in the factory [17].

3.1 Lean Software Development

In software development, Computer-Aided Lean Management can be instituted as a methodology of developing software-controlled, lean systems integration [19]. By this means, the developers can use rigor of computational software in an integrated fashion to reduce risk and drive out inefficiencies [19]. In Agile development, Lean software development (LSD) is being used as a translation of manufacturing and lean IT principles and practices to the software development domain. [19].

4. METHODOLOGY

Descriptive secondary data analysis has been used in the study. Databases pre-industry system developers have been analyzed composed of the 4 year list of 1,525 developers with 305 software development teams, also their contention to why software fails, or why software fails during business acceptance test were utilized. The researchers' have been active participants during the software development and handles several software developers. Fishbone Analysis also integrated as tool in wastes identification.

5. RESULTS AND DISCUSSION

The Fishbone Diagram –Level 1 and Level 2 depicts causes of software wastes that the developers attributed the same. Fishbone Analysis (also called as herringbone or Ishikawa Diagram) is used to identify root cause of the problems by streamlining and uncovering the bottlenecks in the production line or business processes. This tool devise and organization or a manager to consider the quality of product or services by using the 6Ms in the Manufacturing industry. The 6Ms however was used in software development in some of the business solution and software development organization using the Agile approaches. These thematic representations has been associated to the developers or clients configuration items in terms of ***Management***, ***Man*** (team staff), and client staffing, ***Methodology*** or business processes (SAD processes, or clients' business processes), ***Measurement*** (testing issues and cases, metrics and scales), ***Machine*** and ***Material*** (substance of system).

These 6Ms are however interrelated which always affects the behavior of system quality or even the behavior of the process which itself contribute to the failure of the software. As noted earlier, software failure is always associated to the product itself, not to the extent why the software failed.

To observe a fail-safe and error-free operation, early detection of waste is needed to remove the same quickly. Waste elimination is initiated in conformance with software quality control (result-oriented) and quality assurance (process-oriented). In this context, it promotes process-centric software development which oversees compliance to requirement specification on each process or stage of the development. The pre-industry developers revealed the following which constitutes the Cause and Effect- Level 1.

5.1 Causes of Waste- Level 1

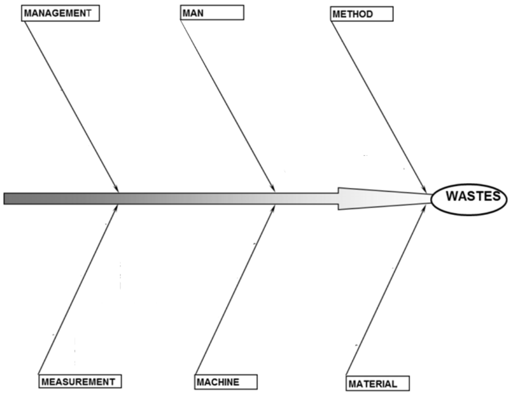


Figure 3. The Cause and Effect Diagram-Level 1.

From the Figure 3 representation, waste is considered as effect of 6Ms, and based on the results of content analyses; transcripts from the focus group discussion, field notes and observations, there have been several causes of wastes in software development. The respondents were asked to elicit and elaborate their contention to the following;

5.1 .1 Management

The clients or end-users of the system developed are attributed to as the Management. Management includes the strategic down to the operational level employees who were contend to the system users during the unit testing, modular testing and system testing. The management includes the General or Operations Manager, Department Supervisors, Heads and MIS Director for small scale clients, and for larger entities they are the Chief Executive Officers (CEO), Chief Operations Officers (COO), Chief Information Officers, Presidents, and Managers. During the requirements gathering, these company representatives were asked in the User Specific Requirements, Functional Requirements, Non-Functional Requirements, Data Analysis and others. During the documentation, design and development, the developers blocked a schedule to present the system to meet User Characteristics and other functional value of the system.

5.1.2 Man

During the software development, those involved in the development are considered also as the sources of wastes. The participants of software development include, Project Manager, Software Engineer, System Analyst, Business System Analyst, Programmer/Software Developer, Document Specialist and the Technical Adviser while the System Analysis and Design subject adviser served as the Project Consultant. More importantly, the end-users are included amongst the Project Stakeholder.

5.1.3Methodology

The developers' clients have their customary way of doing things, and this is called the **Business Process,** it is a structured and related set of activities in producing a product or service. From the current business processes, the developers will streamline and recommends solution to simplify ways of doing of their clients' business processes through information system which will eventually automate some of the processes. Once streamlined, inefficient processes will be removed and replaced with tried and true approach and will be integrated into the function of the system. Also, being a pre-industry developer, their constraints are on the System Analysis and Design course processes as well.

5.1.4 Measurement

To comply with the quality assurance, quality control and even software quality assurance, the software being developed undergone several type of tests to meet set of specified requirements on the users' perspective, Project Manager, Technical Adviser, and Project Consultant. These tests include unit test, module tests, integration tests, systems tests, and Business Acceptance Testing. There have been established rubrics for each perspective, however considering the Lean Principles, the End-User or customers' perspective supersedes the other. Because of several stakeholder involved, several changes are employed which includes the emergency changes to system, major and minor changes, standard and normal changes.

5.1.5 Machine

The clients' infrastructure is the most notable consideration on the developers' view of the system. The developers' always look into the integration plan of the company should the system be deployed over their infrastructure. Developing web-based system/ application that has to be deployed over the internet (online) once deployed, the client preferences is just by intranet implementation is of waste. The respondents revealed that the course requirement for System Analysis and Design must be online, but some of the clients' insisted to deploy the application in their internal network only. The course advisers' does not allow use of software development frameworks like Laravel, Code Igniter, etc. however some of the industry developers have actually been using the same.

5.1.6 Material

In this study, the Materials are attributed to the information taken from the developers' client. During the requirements gathering, they will conduct Focus Group Discussion, Semi-Structured Interview and other means of getting necessary documents which will be used as input to their documentation and system design and development. With this however, some of their clients provides vague information, or others are incomplete. Also, associated to materials are the program codes, modules, and other functions.

5.2 Causes of Waste- Level 2

Customer satisfaction is always equated to loyalty, more revenue and more projects. To close more projects with impressive team records, each of the 6Ms must be visited to eliminate foregoing causes of wastes below. This section presents details on the root cause analysis of software development wastes.

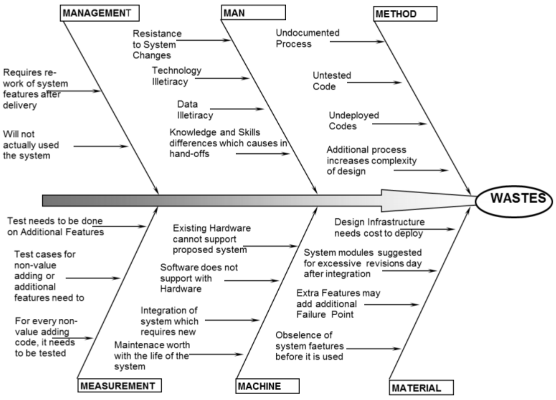

MANAGEMENT
MAN
METHOD
Resistance to System Changes
Undocumented Process
Requires re-work of system features after delivery
Technology Illetiracy
Untested Code
Data Illetiracy
Undeployed Codes
Will not actually used the system
Knowledge and Skills differences which causes in hand-offs
Additional process increases complexity of design
WASTES
Test needs to be done on Additional Features
Existing Hardware cannot support proposed system
Design Infrastructure needs cost to deploy
Test cases for non-value adding or additional features need to
Software does not support with Hardware
System modules suggested for excessive revisions day after integration
Extra Features may add additional Failure Point
Integration of system which requires new
For every non-value adding code, it needs to be tested
Maintenace worth with the life of the system
Obselence of system faetures before it is used
MEASUREMENT
MACHINE
MATERIAL

Figure 4. The Cause and Effect Diagram-Level 2.

5.2.1 Management

The system has been designed and developed for free by the pre-industry developers. Before the system is delivered and deployed, an integration plan is implemented, and User Training has been provided to the clients’ employee. With this therefore, reworks after the deployment and maintenance period is not a subject of their responsibility, and they have had considered it waste, should there be system improvement after the period it is regarded as another milestone of the project. With this argument, it negates to the definition of waste which is on the customers’ perspective, not on the developers. Moreover, if the clients’ will not actually use the developed system, it is converted to a waste.

5.2.2 Man

Due to diversity of the project stakeholders, a common language served as their means of communication. The developers communicate in the technical language they are into while the clients have their common understanding on the lay-mans and or business language on their workplace culture. In this context, Project Human Resources, Communication and Integration Management must be instituted by the Project Managers. During the development, there are neophyte to information system, employees who are resistant to changes during implementation, technology illiterate employee which is a person who don’t know how and when to use the technology, data illiteracy and knowledge and skills of the development team. You may have skilled team on Java, JSP, JavaScript and others, however your client just need a system which requires skills in Visual Basic, your team is a waste also. In the concept of Project Management, cases like this can be resolved by having multiple projects or accounts handled by a team so that idle time is eliminated.

5.2.3 Methodology

Respondents revealed that some of their clients were overwhelmed on the system they have had presented during their weekly visit for system presentation and have made immediate recommendations and suggestions which are actually not part of the documented Requirement Gathering. This makes emergency changes on the system interface, function and the entire processes. Also, for some developers, there were identified codes which do not add value and function to the system. Should there be changes to processes and functions, documentation should be put in place and document control must be maintained for versions of changes and updates. Participants noted, that in every changes or process modification, system complexity arises. Wastes such as undeployed codes, undocumented processes, and untested codes are attributed to fall under this category.

5.2.4 Measurement

Metrics and software quality tools is a necessity, however end-users perspective is always the priority. Layout, functions, design of system which is pleasing on the Managers perspective if not appealing and not accepted by the client is a waste. The Business Acceptance Test supersedes other test. Use case, Test case needs to be done on additional features; features which actually not required by the user, non-value adding features, and tests for every non-value adding features are considered wastes. Therefore, developers must consider the significant-important, and the necessary-sufficient conditions.

5.2.5 Machine

In this changing time, technology had evolved as useful arts with specific applications. For the developers, technology is the machine use to do their related tasks in the project, the integrated development environment, and the infrastructure of the workplace. On the clients' perspective, technologies are the tools and devices used in doing their daily tasks, and activities to providing services or product to their clients and or customers. The respondents' contentions to wastes are as; (a) the current infrastructure of the clients is not capable of the proposed system; recommended acquisition for new hardware has been made, however during the implementation there is not new hardware, (b) the new system as software counterpart exceeds hardware specification to be synchronized, (c) integration of the system may require new acquisitions, (d) maintenance of the system may worth the life of the new system. Obsolete technology is a waste as well; surely because if it is more costly to maintain existing technology than acquiring new ones, it is impractical to a business or organization.

5.2.6 Material

Information is considered as the key resource in any information system as previously quoted. The granularity is always put on top and processing mechanism. In this study, developers have noted wastes attributed as Material. Information is considered as variable in a system. Good information system provides generation of new knowledge, reusability, and scalability and among others. Developers noted that excessive revisions on the documentation, processes, and extra features not specify by the clients requirements are amongst wastes in software development.

6. CONCLUSIONS AND RECOMMENDATIONS

Based on the study, the following conclusions and suggested recommendation have been derived. It has been noted that Lean principles in software development summarizes seven principles which includes; (a) **eliminate waste**, (b)**amplify learning**, (c) **decide as late as possible**, (d) **deliver as fast as possible**, (e) **empower the team**, (f) **build integrity in**, and (g) **see the whole [20]**. The following are the walk-troughs of lean principles according to Mary Poppendieck, Tom Poppendieck in their book Lean Software Development: An Agile Toolkit (Kindle Edition). As previously noted on the introductory, implementation of Lean Software Development can eliminate wastes at an early stage.

6.1 Lean Software Development

6.1.1 Eliminate waste

Waste is anything that does not add value to a product; value is perceived by the customer. In lean thinking, the concept of waste is high hurdle. If a component is sitting on a shelf gathering dust, that is waste. If a development cycle has collected requirements in a book gathering dust that is waste. If the developers code more features that are immediately needed, that is waste. The ideal is to find out what a customer wants, and then make or develop it and deliver exactly what they want, virtually immediately [20]. Additional features, codes are therefore waste.
Lean philosophy regards everything not value- adding to the customer as waste [21], and in order to eliminate waste it should be recognized first. A value stream mapping technique can be used to identify wastes. Waste to SD may include; (a) unnecessary code and functionality, (b) delay in the software development process, (c) unclear and vague requirements, (d) insufficient testing (leading to avoidable process repetition), (e) bureaucracy, and (f) slow internal communication [21].

6.1.2 Amplify learning

Development is an exercise in discovery, while production is an exercise in reducing variation, and for this reason, a lean approach to development results in practices that are quite different than lean production practices [20]. Software development is a continuous learning process with the additional challenge of development teams and end product sizes… the accumulation of defects should be prevented by running tests as soon as the code is written [22]. The best approach to improving a software development environment is to amplify learning since SD is best conceived of as a similar learning process with added challenge that development teams are large and the results are far more complex [20].

6.1.3 Decide as late as possible

Delaying decisions is valuable because better decisions can be made when they are based on fact, not speculation. Development practices that provide for late decision making are effective in domains that involve uncertainty, because they provide option-based approach [20]. In handling projects and software development, the team should consider that idea that software development is always associated with some uncertainty, better results should be achieved with an options-based approach, delaying decisions as much as possible until they can be made based on facts and not on uncertain assumptions and predictions, also the iterative approach promotes this principle – the ability to adapt to changes and correct mistakes, which might be very costly if discovered after the release of the system [23]. It's an important tenet for self-managing teams, encouraging them to avoid detailed discussions of items not currently under development or nearby on the planning horizon [24].

6.1.4 Deliver as fast as possible

Practically, with speed comes efficiency; when decisions do need to be made, the likelihood of delaying is diminished because the speed of the project will force rapid decisions when the decision must be made [25]. The software should be designed in such a way so that it can deliver as much increments as possible. Any increment is working software and opens up the feedback loop for any potential problems [26]. The speed cost more" to join "quality cost more" development the discovery cycle is critical for learning: design, implement, feedback, improve [20]. Mary and Tom noted that The shorter the cycles are, the more can be learned.

6.1.5 Empower the team

Top-notch execution lies in getting the details right, and no one understands the details better than the people who actually do the work [20]. The most important thing on a project is good leadership; the least important thing is who leads [27]. Leadership is about seeing a problem then getting people together to solve it. Any team member who knows enough to foresee a problem also knows enough to lead the solution [27]. It is the responsibility of the Project Manager to do so; emotional attachment of people to their projects must be considered by the manager. The lean practices use pull techniques to schedule work and contain local signaling mechanism is an increasingly refined versions of working software at regular intervals [20].

6.1.6 Build integrity in

A system is perceived to have integrity when a user thinks. Conceptual integrity means that the systems central concepts work together as a smooth, cohesive whole and it is a critical factor in creating perceived integrity [20]. Cohesion is a qualitative measure meaning that the source code text to be measured is examined using a rubric to determine a cohesion classification [28]. In SAFECODE, software integrity is attributed to as a processes for sourcing, creating and delivering software contain controls to enhance confidence that the software functions as the supplier intended [29]. Integrity comes from wise leadership, relevant expertise, effective communication, and healthy discipline; processes, procedures, and measurements are not adequate substitutes [20]. Conceptual integrity means that the system's separate components work well together as a whole with balance between flexibility, maintainability, efficiency, and responsiveness [30].

6.1.7 See the whole.

Integrity in complex system requires a deep expertise in many diverse areas [20]. Software systems nowadays are not simply the sum of their parts, but also the product of their interactions. Defects in software tend to accumulate during the development process – by decomposing the big tasks into smaller tasks, and by standardizing different stages of development, the root causes of defects should be found and eliminated [31]. Lean thinking has to be understood well by all members of a project, before implementing in a concrete, real-life situation [31]. Finally, product optimization as a whole can be integrated into the development. Putting all of this together with the better optimized workflow, the benefits or organizing in this way can be extremely significant – not only in terms of the team's performance, but also in terms of the quality of the product, which ultimately can make your organization more competitive [32].

6.2 Concluding Remarks

In managing business solutions, software projects, the team always looks into the system as end-product. Championing software that will sustain clients or business operation is but more interesting. Now in eliminating deadly wastes in software development, system developers or projects management may employ suitable approach based on the criteria, system functions, software project management and conformance to software quality. Project Managers or Software Project Manager may adopt agile practices to manage software projects even in smaller or larger context. Putting into practice the Lean Software Development helps identify and eventual removal of waste, focus on value and flow and the people as resource.

To eliminate wastes means maximizing throughput and additional revenue for the project team. Lean IT however have had tried and true success stories, but more research is encourage and suggested to determine its implications to software developers, project managers, and software development and business solutions company. Empirical study is recommended on Lean Principles to further its value and flourish underway. Should lean be put into practice in software development, the team composition and organization will see a promise of savings, speed, quality and business alignment.

Acknowledgments

The researchers would like to thank the pre-industry system developers of Information Technology Department of Jose Rizal University specially those batches of S.Y. 2010-2014.

REFERENCES

[1] Kendall, Kenneth E., Kendall, Julie E. (2011) System Analysis and Design, New Jersey (8th ed.). Pearson Publishing.

[2] CIO Update: “Assessing Your BSM Technology Investments.” Drogseth Retrieved August 23, 2014 from http://www.cioupdate.com/trends/article.php/3852821/Assessing-Your-BSM-Technology-Investments.htm

[3] Application Development. Retrieved August 22, 2014 from http://www.bestpricecomputers.co.uk/glossary/application-development.htm

[4] Bishop, Tom, BSM Review: “How to Spell ‘BSM.’” Bishop. Retrieved August 23, 2014 from http://www.bsmreview.com/spell_bsm.shtml

[5] DRM Associates (2002). "New Product Development Glossary". Retrieved 2014-08-20. from http://www.npd-solutions.com/glossary.html

[6] "Application Development (AppDev) Defined and Explained". Bestpricecomputers.co.uk. 2007-08-13. Retrieved 2014-08-20. from http://www.bestpricecomputers.co.uk/glossary/application-development.htm

[7] Laplante, Phillip (2007). What Every Engineer Should Know about Software Engineering. Boca Raton: CRC. ISBN 978-0-8493-7228-5. Retrieved 2014-08-20

[8] Stellman, Andrew; Greene, Jennifer (2005). Applied Software Project Management. O'Reilly Media. ISBN 978-0-596-00948-9.

[9] "What is Project Management? | Project Management Institute". Pmi.org. Retrieved 2014-07-26. from http://www.pmi.org/About-Us/About-Us-What-is-Project-Management.aspx

[10] Kendall, Kenneth E., Kendall, Julie E. (2011) System Analysis and Design, New Jersey (8th ed.). Pearson Publishing

[11] Janssen, Cory. "Project Management Software". Techopedia. Retrieved 26 July 2014 from http://www.techopedia.com/definition/13132/project-management-software

[12] Lean manufacturing Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Lean_manufacturing

[13] Holweg, Matthias (2007). "Abstract of The genealogy of lean production". Journal of Operation Management 25 (2): . doi:10.1016/j.jom.2006.04.001. Retrieved August 23, 2014 from http://www.sciencedirect.com/science/article/pii/S0272696306000313

[14] Muda (Japanese term) Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Muda_(Japanese_term)

[15] Types of waste: Lean manufacturing Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Lean_manufacturing

[16] The Five Principles of Lean Thinking, Lean University. Retrieved August 23, 2014 from http://www.cardiff.ac.uk/lean/principles

[17] Maskell & Baggaley (December 19, 2003). "Practical Lean Accounting". Productivity Press, New York, NY.

[18] Principles of Lean, Lean Enterprise Institute. Retrieved August 23, 2014 from http://www.lean.org/WhatsLean/Principles.cfm

[19] Yasuhiro Monden (1998), Toyota Production System, An Integrated Approach to Just-In-Time, Third edition, Norcross, GA: Engineering & Management Press, ISBN 0-412-83930-X.

[20] Mary Poppendieck, Tom Poppendieck (2003), "Lean Software Development: An Agile Toolkit", Addison-Wesley Professional, ISBN 0-321-15078-3

[21] Eliminate waste, Lean software development. Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Lean_software_development

[22] Amplify learning, Lean software development. Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Lean_software_development

[23] Decide as late as possible, Lean software development. Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Lean_software_development

[24] Preuss, DH, 2006, Discussion: "Decide as Late as Possible". Retrieved August 23, 2014 from http://www.infoq.com/news/Discussion-Decide-Late-as-Poss

[25] Deliver as fast as possible. Safari Books Online Retrieved August 23, 2014 from http://www.safaribooksonline.com/library/view/agile-it-security/9781849685702/ch07s04.html

[26] Deliver as fast as possible. ANDROMEDA-AGILE. Retrieved August 23, 2014 from http://www.andromeda-agile.com/deliver-as-fast-as-possible.html

[27] Team Empowerment; Agile Process. Retrieved August 23, 2014 from http://www.agile-process.org/team.html

[28] Cohesion (computer science). Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Cohesion_(computer_science)

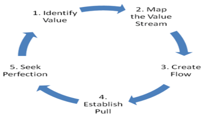
[29] Security, Integrity and Authenticity: The Tripod of Software Assurance. SAFECODE. Retrieved August 23, 2014 from http://www.safecode.org/security-integrity-and-authenticity-the-tripod-of-software-assurance/

[30] Build integrity in , Lean software development. Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Lean_software_development

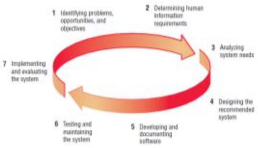
[31] See the whole, Lean software development. Retrieved August 23, 2014 from http://en.wikipedia.org/wiki/Lean_software_development

[32] Lean Principle #7 – Optimise The Whole: All about Agile. Retrieved August 23, 2014 from http://www.allaboutagile.com/lean-principle-7-optimise-the-whole/

[33] Principles of Lean, Lean Enterprise Institute. Retrieved August 23, 2014 from http://www.lean.org/WhatsLean/Principles.cfm



[34] Kendall, Kenneth E., Kendall, Julie E. (2011) *System Analysis and Design, New Jersey* (8th ed.). Pearson Publishing.


1 Identifying problems, opportunities, and objectives
2 Determining human information requirements
3 Analyzing system needs
4 Designing the recommended system
5 Developing and documenting software
6 Testing and maintaining the system
7 Implementing and evaluating the system