A new vision for information technology project management through selecting SDLC model

Jasour A. Obeidat,[a] Hebah H. O. Nasereddin[b]

a MEU — Middle East University, Amman, Jordan
jasour_2004@yahoo.com
b MEU — Middle East University, Amman, Jordan

Abstract. This document Any software development processes have to follow a specific system development life cycle SDLC. There are many SDLC proposed and assigned as a guide lines for developing software. Many conditions force software development team to follow specific SDLC model depending on development strategy and the nature of organization. Many decisions generated through development process which required a highly skilled project leader. This study concentrated on developing a project management tool as a service, to provide solutions to organizations especially in selecting the suitable SDLC because of its necessity in meeting software development project objectives and closing it successfully. In this paper, the author is proposing a new model for managing information technology projects starting from selecting SDLC till closing project phases. For this, it was important to study many techniques used for comparing between SDLC models and provide the suitable model for the requested software project. The proposed system will increase concentration on project requirements and the changes occurred during development phases.

Keywords: optics, System Development Life Cycle, Waterfall, Incremental, Spiral, Project Characteristics, SWOT, Requirements characteristics., templates.

1 INTRODUCTION

The nature of business in organizations become more complex which leads for increasing the degree of complexity in information systems used at these organizations, and increasing organization dependency on computer software. The qualifications of computer systems also increased (i.e. powerful processing, and storing) which represented as a second factor for computerizing organization's systems.

In addition, of using computer systems for managing and controlling organization's business, people are using those systems for fun and entertainment. By the mentioned factors computer software represent a core issue, and it is important to be suited in a way that fit the business of organization.

System Development Life Cycle SDLC a set of sequential activities that the development teams have to follow in developing software. The general framework for any model of SDLC passes software development through common phases such as: initiation, planning, design, implementing, and testing. In the other hand, each SDLC model has its own strength points, and weaknesses points represented by the functionalities that provides or not.

Deeply, we can say that we will face a question "Which is the suitable SDLC model for my project?", there are many answers on this question such as "*it depend on the nature of business in organization*", or "*it is a trade offs among what we want to have and what is possible to do*".

Project management is the art of managing project resources and components, and the science of integrating them depending on clear plan and schedule (i.e. SDLC). Project may be

build from scratch or updating and developing current ones. In both sides it is temporary have a starting point and ending point. Project management deals with diversity resources such as people, materials, time, and cost for running the project (Abdullah Saeed Bani Ali, 2005).

2 COMMON SDLC MODELS

This section will summarize the procedures in each SDLC model, and applying SWOT analysis for each one of them.

2.1 The waterfall model

Assigned to be the classical software engineering model because it is the oldest model, and it is widely used in small organizations that have frozen requirements changes in its working nature (International organization for standardization / International Electronical Commission, 2004). This model force development team to make emphasize planning and well understanding system requirements before starting the project (Comer,E, 1997). This model consisted to completely non-intersectional stages (i.e. no return back procedures for completing stages). Waterfall model starts from defining and collecting software and system requirements, then designing the software architecture, later on translating these designs into codes, and finally testing final project product.

After each completed stage, it involves documenting the development which reflect an extra cost, and delaying the stating of next stage, and slow down the development progress of total development time, that is expensive for customers and top management (Vishwas Massey, K.J Sato, 2012).

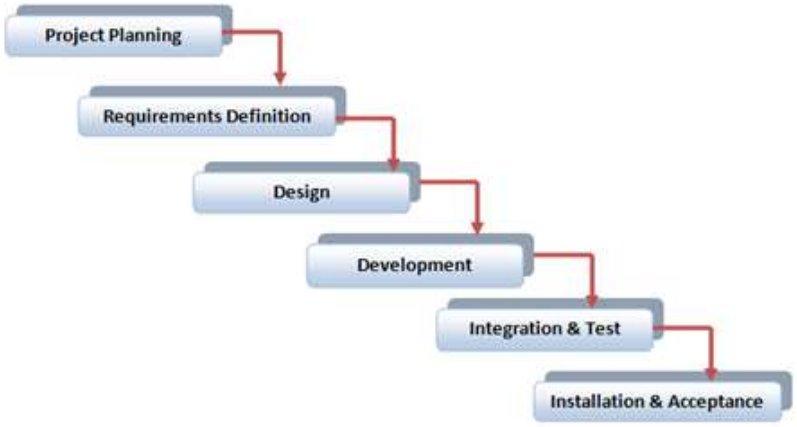

Fig. 1. Waterfall SDLC

2.1.1 Waterfall model advantages and disadvantages

(PK.Ragunth, S.Velmourougan. P.Davachelvan, S.kayalvizhi, R.Ravimohan, 2012) Summarized the advantages and disadvantages of waterfall model as the following:

Advantages:

1. Simple to understand, easy to implement and clear to view.
2. Reinforces management at each stage by checking deliverables of each completed stage.
3. Good model for small organization that have limited number of requirements.

Disadvantages:

1. By using this model we will face high risk especially in complex organizations.
2. Rigid model and inflexible especially in cases of changing requirements.
3. Doesn't support long ongoing projects because it forces development teams to work with long term plans.
4. Does not support object oriented projects.

2.1.2 Applying SWOT analysis on waterfall model

(Ashish B. Sasankar, Dr.Viny Chavan, 2011) they found after applying SWOT analysis on waterfall model the following results:

Strengths:

1. Ability to be managed by not IT persons (i.e. End users).
2. The easiest model for structuring non experienced team to work with.
3. Good choice for small organizations that working with clear and stable requirements.

Weaknesses:

1. Requirements definition should be previously collected and arranged.
2. Does not support problem solving techniques since there are no overlapping phases for verifications issues.
3. Decrease the chance for customers to preview the system in the progress.
4. Bad choice for big organizations contains complex operations with variable requirements.

Opportunities:

1. Technology used is understood.
2. Stable definition of product.
3. Well formatted requirements.

Threats:

The evolution in the nature of information technology environments, and the rapid changes in the strategies that increased concentration on reuse proportion - By implementing packages and reusing the frameworks - The waterfall model will not be economic or practical to meet these changes. For these issues waterfall model represented as restricted model.

2.2 Incremental Model

Information systems nowadays featured as customizable, updatable, and changeable software. We need a system development model that takes into consideration the ability to change without rebuilding systems from scratch which it is highly costs process, or restructuring new

changes which take a long time. Incremental SDLC represented as an evolutionary software model.

The basic idea behind incremental model is to implement a partial structure of the system, and adding new releases to the basic structure - in a way that any new subsequent release added will insert new functionality to the proposed system- until all requirements and objectives of the system are met. In the incremental model, requirements collected, sorted, and prioritized in order to be grouped later on. The primary stage in this model is "*How to partition system into increments?*" - Noting that the desired increment is the small as possible increment- in order to add a useful functionality to the system (Biresh Kumar, Ritesh Kumar, Anita Kumar, 2012).

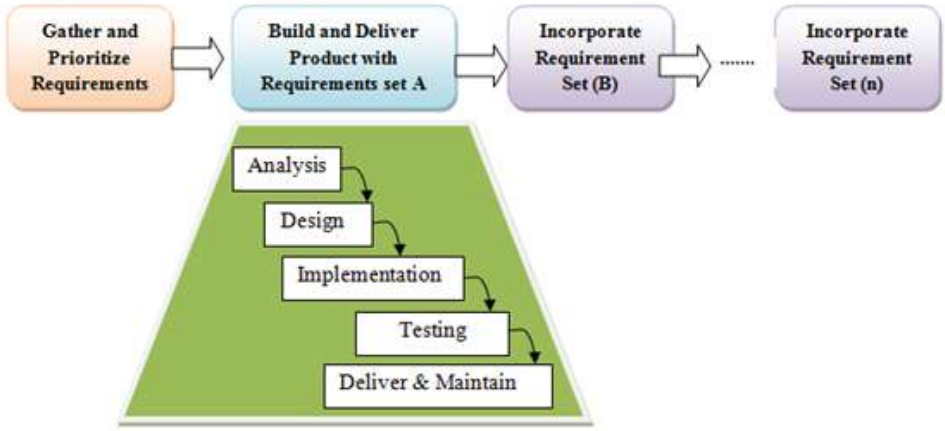

Fig. 2. Incremental SDLC

2.2.1 Incremental SDLC advantages and disadvantages

(Biresh Kumar, Ritesh Kumar, Anita Kumar, 2012) they discussed incremental SDLC and find the following advantages and disadvantages:

Advantages:

1. Flexible manipulating with requirement changes.
2. Support debugging and problem solving after adding any new increment to system.
3. Support parallel development; by assigning partitioned to multiple development teams.
4. Increases the chance of viewing completed system parts to customers.
5. Decreasing failure occurrence chances, because of multiple milestones in every increment.

Disadvantages:

1. Incremental SDLC depend on gathering requirements and prioritizing them, but some of requirements are not available to meet organizational architecture through planning phase.
2. Incremental SDLC modules are completely separated and no overlapping among them. This leads losing interfaces among modules, which will reflect a future problem through integration phase.
3. Developing costs getting higher, because of delivering project as parts.

2.2.2 Applying SOWT analysis on incremental SDLC

(Ashish B. Sasankar, Dr.Viny Chavan, 2011) they found after applying SWOT analysis on incremental model the following results:

Strengths:

1. Comparing with waterfall model it is more flexible model.
2. Any completed release represents successful running application.
3. Concentrate on main parts in the organization.
4. Working with sub parts easier than working with the whole system.
5. Support the first sight for customers at any completed release.

Weaknesses:

1. Required domain experts to analyze the architecture of organization.
2. Verification and testing will be at each new release which will add new extra costs.
3. Ignoring the importance of module interfaces management will decrease the chances of successful integration.

Opportunities:

1. Capable to be used in developing updatable systems.
2. Capable to be used with new technologies software.
3. Capable to enter market with trial versions of software.

Threats:

Incremental SDLC ignores partitions dependency among system parts and concentrated on grouping requirements, which increases the possibility of risk occurrence in the system.

2.3 Spiral SDLC model

Spiral model consist into four phases: Planning, Risk analysis, Engineering, and Evaluation. This model approximately similar to incremental model with more focus on analyzing risks in spiral. Requirements gathered in the planning phase. Risk analysis of each spiral processed, calculated, and alternative solutions proposed. Physical implementation for each sub application conducted in the engineering phase. Customers review each sub application in the evaluation phase to assist its outputs with goals and objectives presented in the planning phase. The cost of each phase could be computed by measuring the radius of each phase (i.e. spiral) (Biresh Kumar, Ritesh Kumar, Anita Kumar, 2012).

Fig. 3. Spiral SDLC

2.3.1 Spiral model advantages and disadvantages

(Biresh Kumar, Ritesh Kumar, Anita Kumar, 2012) they discussed incremental SDLC and find the following advantages and disadvantages:

Advantages:

1. Decreasing system failures and risks; because of large amount of risk analysis and alternatives generation.
2. Support flexible dealing with new changes and updates.
3. Early production of completed parts of the proposed system.
4. Successful choice for high complex organizations nature.

Disadvantages:

1. Expensive model to be applied.
2. Requires highly skilled project leaders and domain experts.
3. Not applicable for small organizations and projects.
4. Project success depends on the way of analyzing risks.

2.3.2 Applying SWOT analysis in spiral SDLC model

(Ashish B. Sasankar, Dr.Viny Chavan, 2011) they found after applying SWOT analysis on spiral model the following results:

Strengths:

1. Customers have a big chance to see the project at evaluation phase and provide frequent feedback about sub systems outputs.
2. Starting from designing and implementing high risk functions.
3. Building prototypes offers risk detection in early phases.
4. Comparing with incremental model integration process in project management will not be a big deal.

Weaknesses:

1. Complex model and not suitable with direct or small projects.
2. Focus on risk analysis and assessment which take a long time (i.e. consuming development time).

3. Developers work status is idle while working with non-development phases.
4. Weak in defining milestones.

Opportunities:

1. Success in complex projects.
2. Good choice for long ongoing projects.
3. Expect significant changes.

Threats:

Risk in spiral model may occur in case of misunderstanding user needs and bad analysis of organizations architecture.

3 SDLC MODELS COMPARISON

(Sanjana Taya, Shaveta Gupta, 2011) focus on the comparitive analysis of SDLC models, and summariezed results. Table 1 below show the comparion between (Waterfall, RAD, Spiral, Agile, and Incremental).

Table 1. SDLC comparison

Characterstic	Waterfall	RAD	SPIRAL	AGILE	Incremental
Requirment Specification	Beginning	Time box released	Beginning	Frequently Changed	Beginning
Understanding Requierments	Well Understood	Easily Understood	Well Understood	Well Understood	Well Understood
Cost	Low	Low	Expensive	Very High	Low
Gurantee of Success	Low	Good	High	Very High	High
Resource Control	Yes	Yes	Yes	No	Yes
Simplicity	Simple	Very Simple	Intermediate	Complex	Intermediate
Risk Involvment	High	Very Low	Low	Reduced	Easily Managed
User Involvment	Only at the beginning	Only at the beginning	High	High	Intermediate
Overlapping Phases	No such phases	No	Yes	Yes	No
Flexibility	Rigid	High	Flexibile	Highly Flexible	Less Flexibile
Time	Long	Short	Long	Least Possible	Very Long
Reusability	Limited	Some Extent	Yes	'Reuse' use case	Yes

4 SELECTING SDLC

The most important task faces many organization is selecting the suitable SDLC for their projects. There are many SDLC models proposed and used in the area of developing software projects. Each one of them has advantages and disadvantages. Hence, choosing SDLC model is hard decision, and should be based on practical knowledge about the actual characteristics

of organization architecture, users, and nature of work. The biggest challenge in this process that we are working with qualitative based processes, not quantitative based processes. In this paper, the author tried to convert qualities – which represent system characteristics and requirements – into quantities.

4.1 Scope of proposed selection technique

The proposed technique will analyze the following factors:

1. Type of requirements.
2. Quality of requirements.
3. Project size.
4. Develo**p**ment team experience.
5. Organizational culture and work nature.
6. Hardware Dependency.
7. Graphical User Interface (GUI) nature.
8. Product delivery.
9. Product security and data safety.

Algorithm (I)
Requirements Characteristics Analyzer (RCA)
1. Development team Filling RCA Form.
2. Comparing Traditional SDLC Vs. Iterative SDLC Models in case of requirements.
3. Application Generates Result | The highest score wins.
// One point for selection.

4.2 Applying algorithm on Waterfall SDLC models

This section shows the results of applying the proposed algorithm. Table 2 below describe the results after applying the proposed algorithm.

Table 2. The results after applying algorithm

Characteristics	**1 point if TRUE**	**1 Point if True**
Type of requirements	1	0
Quality of requirements	0	1
Project size	1	0
Development team experience	1	0
Organizational culture and work nature	0	1
Graphical user interface nature	1	0
Product delivery	1	0
Product security and data safety.	0	1
***Total Score* (out of 8)**	**5**	**3**

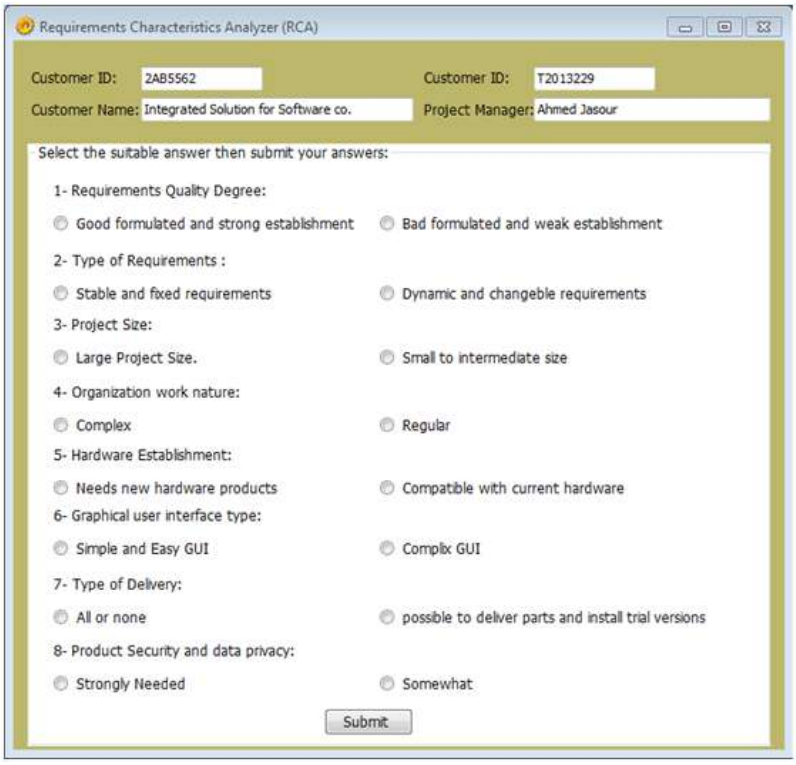

Fig. 4. Requirements characteristics of software

Table 2 shown the percentage of applying waterfall model is (62.5%, 37.5%) which will be as a standard for applying waterfall model as a selected SDLC. By comparing the results of applying the proposed project in the algorithm, then the similarity between project results and standards computed for all SDLC models. For instance, if the proposed project results was (62.5%, 37.5%) then the similarity will equal 100%, then the waterfall model is the suitable model in some circumstances.

5 CONCLUSIONS

Software engineering offered many types of SDLC. Proposing SDLC models was for easing the way to control and manage development and decreasing failures. By proposing many types of SDLC, project managers face great competition about selecting among many alternatives. Choosing wrong SDLC model leads for future problems. The proposed technique provides a starting point in the scope of selecting suitable SDLC for projects, the accuracy of results was acceptable, but with adding new features and characteristics to the algorithm, it will reflect better results.

References

Abdullah Saeed Bani Ali. "A study of project management system acceptance", 0-7695-2268-8/105 IEEE software, Proceeding of the 38th Annual Hawai International Conference on System Sciences (HICSS)-Track 8-Volume 08 IEEE 2005.

International organization for standardization / International Electronical Commission 2004. Software Engineering. "Guide to the software engineering body of knowledge (SWEBOK) and well understanding".

Comer,E. "Alternative software life cycle models", presented at international conference on software engineering, 1997.

Vishwas Massey, Prof.K.J Sato. "Comparing Various SDLC Models and The new Prposal Model on the Basis of Available Methodology", IJARCSSE, ISSN:2277 128X, Volume 2, ISSUE 4, April 2012.

PK.Ragunth, S.Velmourougan. P.Davachelvan, S.kayalvizhi, R.Ravimohan, "Evolving A New Model (SDLC-2012) For Software Development Life Cycle SDLC", IJCSNS International Journal for Computer Sciences and Network Security, Vol.10, No. 1, January, 2010.

Ashish B. Sasankar, Dr.Viny Chavan. "SWOT Analysis of Software Development Process Models", IJCSI International Journal of Computer Sciences Issues, Vol.8, No.3, September 2011.

Biresh Kumar, Ritesh Kumar, Anita Kumar. "A comparative Analysis of Software Engineering Models and its impact on project management", National Conference on Emerging Challenges for sustainable Business, Dec.2012

Sanjana Taya, Shaveta Gupta. "Comparative Analysis of Software Development Life Cycle Models", International Journal of Computer Science and Technology, IJCST, Dec.2011