Predictive cum Adaptive Systems Development Methodology for HydroGIS Tool Development

RMM Pradeep[1#], NTS Wijesekara [2]

[1] General Sir John Kothelawala Defence University, Sri Lanka
[2] University of Moratuwa ,Sri Lanka
[#]Corresponding author; <pradeep@army.lk>

Abstract— *In system development methodologies, main consideration has being changed from processes to users since waterfall development. As the business logic became less complex due to familiarity; Extreme Programming methodologies like Agile and Scrum become popular among the programmers. Nevertheless, the engineering processes like hydrology modelling, which are still evolving, remain in the same complex. As well historically established hydrology calculations, which are base to evolving models, are remaining complex. When model development, hydrologists have to identify and sequencing those established calculations that best suited to the model scenarios. Then as the final step, they have to calibrate and validate the model, which takes considerable time and effort, before apply to decision making. Hence, once the models are crated, those cannot be changed very easily whilst the automation. As well as the programmers has to pay a considerable attention to the get the 100% accurate result. However, the most users of engineering applications are novice and required more user centric tools same as other users. Hence, there is a difficulty to develop such systems following either predictive (focus on process) or adaptive (focus on users) methodologies which presently available.*

The present work's main objective is to identify and calibrate the most suited combination of methodologies to development of a HydroGIS (Hydrological Geographic Information System) tool, which should accurately automate the complex hydrology process in GIS environment whilst satisfying the user requirements.

Whilst developing the tool, it carries out two parallel developments (1) Automation of engineering process and (2) Achieving the user-friendliness. Whilst automating the processes, a comprehensive devotion was paid to calibrate the accuracy of the calculations. When achieving the maximum user-friendliness, a repetitive developing prototype was used. Once the both parallels come to the accepted level, it amalgamates the prototype with engineering processes. Then integration and system testing were carried out before releasing the final product. The developed tool named GIS2MUSCLE.

The HydroGIS tool which developed, demonstrates 100% accuracy in hydrological and GIS calculations whilst 92% user-friendliness in tool operation. Presently this calibrated methodology, which follows a process centric development to gain user centric tool (PcD.UcT), is being verified with six software development projects.

Keywords— **Engineering System Development Methodology, HydroGIS tool, Predictive cum Adaptive**

I. INTRODUCTION

A. System Development Methodologies

System Development Methodology refers to the approach in implementing the system development life cycle (SDLC) phases in the practical software development. Planning, analysis, design and implementation are the four basic steps of SDLC. In the planning phase, it identifies and reasoning why the system should build and determining how it will go about building it. Through studying the present systems and identifying the problems & opportunities, team conceptualize the new system whilst the analysis phase. In the design phase, team finalise how the system operate by the means of functional and non-functional requirements. At the implementation stage, team builds the system. (Dennis et al., 2009)

These methodologies have being classified according to different perspectives. Following are few examples (1) Plan-driven/traditional or heavyweight and Agile/ lightweight (2) Predictive and Adaptive (3) Process centred and Data centred.(Dennis et al., 2009; Fowler, 2001; Picek, 2009). The present study considers the predictive (scope of the project can be expressed accurately) and adaptive (scope and requirements are difficult to clearly expressed early in the SDLC) perspectives.

B. Engineering Applications

Engineering is a matured profession with experiences. Comprehensive planning, designing and drawing are inherent characteristics of the engineering process. Due to the failure-costs are immensely unbearable, the engineers invest considerable time and resources in these initial stages.

Once the software engineering profession emerge in 1960's, it was a subset of engineering profession which women's job (Meyer, 2013). In that time, planning, design, implementing and maintaining of the hardware was the main computing work whilst software development was a painting work. Since then, the initial software development methodologies such as waterfall and parallel development models emerge with the increasing of the software utilization in the computing. The phases in these development methodologies are based on same characteristics of general engineering process such as comprehensive study, design and then development.

Nevertheless, with the vast distribution of the users among the different knowledge level of computing, the user interface of the software required to be more user-friendly. As well due to the uncertainty of the requirements, the development of the software using predictive methodologies becomes hectic work. Then as a solution, adaptive development methodologies such as extreme programming, agile development versions got the popularity.

The automation of the engineering applications also subjected to same user requirement variations. The situations become worst, as the non-technical decision makers are prone to use the engineering applications when decision-making. As the engineering calculations become more complex with the development of profession, it needs more time and resource in the analysing, design and development phases. Therefore, engineering applications required predictive as well as adaptive development methods simultaneous which is not applicable.

C. HydroGIS Tool Development Methodology

The present work considers a HydroGIS tool, hydrological engineering application for urban decision-making. The decision-making users in the local authorities need to suggest and stress the public when urban land modifications for the purpose of urban flood prevention. Then a series of hydrological calculations need to be performed to evaluate the affect of modifications on the flood generation and select a suitable preventive option. As it is impractical to employ a hydrologist in the process, the requirement is to automate the hydrological calculations, which can perform by local decision makers. Then the automation process should allow users to incorporate land modifications in spatial format, perform hydrological calculation, display output in spatial format and opt a preventive option. (Pradeep and Wijesekara, 2012)

Considering hydrologists are the most naturalistic environmental modellers. The hydrology models developed for natural phenomena are widely accepted and more established than other environmental models (Sui and Maggio, 1999). For the Sri Lankan context, most of the hydrology calculations based on the Ponrajah's guidelines published in 1980s, which have being successfully utilised since then. Such basic calculations, which verified, not only through data but also through time, have become the base for most of todays developing hydrology models.(Chemjong and Wijesekara, 2017; Dahanayake and Wijesekera, 2017; Keerthirathne and Wijesekara, 2017; Sakthivadivel et al., 1997; Thakuri and Wijesekara, 2017). Whilst hydrology model development, the most suitable set of established hydrological calculations is selected. Then the sequence of the calculations is being arraigned to get the required results. Once complete the calculation process sequencing, start the model calibration with available data. Then calibrated model subjected to validate with the real time data. Automation of the process is allowable only after this process, which urged considerable time and resource. Whilst the automation process, the results of each, intermediate to final hydrological calculations steps, has to be verified and confirmed to the standards and norms. Hence, automation becomes a process centric development, which need to carryout with close relation with hydrologists.

As well, hydrological models are based on the geographical distribution of features such as slope, soil, landcover, ground water levels, soil moisture etc. Therefore when perform the hydrological models it required to manipulate the geographic information too. For this purpose, GIS, a specific tool developed to manipulate the geographic information, become a supporting tool in hydrological model calculations (Maidment, 1992). Therefore, to receive the accurate result in automation of the hydrological models it need to develop coding not only for hydrological calculations but also for geographical layer manipulations. As the hydrological systems are based on time and GIS based on the space the automation process get more complex. Then developers need to develop codes with the consultation of the GIS professionals too.

Apart from the complex process, the potential users of the HydroGIS tool are varying from very few highly technical hydrologists to large number of non-technical decision makers like government officials in local government authorities. Importantly the decision makers required the accurate result with minimum interaction with the tools.

Specially, decision makers need to make attribute modifications and viewing the intermediate results, calls trial-and-error process, to arrive the concluding decision.

C. The Problem

Then the final problem is, how a HydroGIS tool that process and user-friendliness both are having same importance, can be developed accurately and satisfactorily.

E. Objective

Therefore, the main objective of the present work is to identify and calibrate the most suited combination of methodologies to development of a HydroGIS tool, which accurately automate the complex hydrology process in GIS environment whilst satisfying the non-technical user requirements.

II. LITERATURE SURVEY

A. System Development Methodologies

A large number of system development methodologies are available as shown in the table 1.0. All these methodologies based on the phases of system development cycle; planning, analysis, Design and implementation. As well, the waterfall method can be considered as the first most development methodology that extensively used in 1970s. As an alternative in 1980s, to satisfy the unsatisfied requirements in development the prototype evolved. Finally, when the "Requirements of users" evolved in 2000s the Agile methodologies were introduced (Avison and Fitzgerald, 2006). Therefore the methodologies shown in the Table 1.0 can be describe as the versions of waterfall and prototype developments.

When study the development methodologies, it can identify the attention has being changed from process automation to user engineering with the time. Nevertheless, it can be considered, waterfall methodology provides the foundation to all the methodologies. Even today, the popular methods like scrum show the features of waterfall and prototype methodologies. However, attention to the user requirement should have a limit. It has identified the user engineering may have a risk of excessive software development with gold-plating or bells-and-whistles or mission/feature/scope/requirement creeping. Then the repercussions may be negatively effect on system development project schedule, quality and cost (Shmueli and Ronen, 2017). Hence, a balance between process automation and user requirement satisfaction should to be maintained.

Therefore, the developers need to a make decisions to select the best methodology to their development process. When deciding the satisfaction of promises (expected positive impacts) and practises (essential steps of methodology) of the selected methodology to the developers' requirement is important. There are number or evaluation mechanism such as Cost-benefit analysis, Scoring evaluation, Feasibility study, Value Analysis and Multi- Objective Multi-Criteria methods and so on are available (Mohagheghi, 2008). But the present work evaluate the methodologies based on much simple approach, analysis the ease of automating the process as well as user requirements. Then the suitability of the each approach for the HydroGIS tool development is shown in the Table 1.0.

Table 1.0 System Development Methodologies

Srl	**Meth[1]**	**On[2]**	**Suitability for HydroGIS tool**
1	Waterfall	P	Best suited for process automation, difficulty incorporate user requirements later
2	Prototyping	U	Best for user friendly development, effects on workflows
3	Iterative and incremental	P	Shorter waterfall steps, but difficult in incorporation of user requirements
4	Spiral	P	The methodology is based on risk reduction
5	Rapid application development	U	Time boxing approach based on fulfilling the requirements
6	Extreme programming	U	Directly automate the user requirements
7	V-Model	P	Based on the testing, difficult in user requirement facilitation
8	Scrum	U	Rather small projects and scrum based
9	Cleanroom	U	Iterations with box structure
10	Dynamic systems development	U	Time boxing approach based on fulfilling the requirements
11	Rational Unified Process	U	A complicated system with iterative development
12	Lean software development	U	Automate the minimum requirements with users
13	Test-driven development	U	Development based on the testing
14	Behaviour-driven development	U	User behaviour centric
15	Feature-driven development	P	The larger project development process which is having less complex processes. Need a complete staff of developers
16	Model-driven engineering	P	Model oriented automations, user requirements are difficult to handle
17	Crystal Methods Methodology	U	Develop based on the developers capability over user requirement
18	Joint Application Development	D	Developer centric, not much describe as a development methodology, but as a tool
19	Adaptive software development	U	Based on User requirement satisfaction

Srl	Meth[1]	On[2]	Suitability for HydroGIS tool
20	Open source software development	D	Developer centric, not much describe as a development methodology, but as a tool
21	Microsoft Solutions Framework	D	Developer centric, not much describe as a development methodology, but as a tool
22	Agile Development	U	A adaptive approach based on the user requirement satisfaction
23	Scrum	U	A adaptive approach based on the user requirement satisfaction, development of agile methodology
24	Kanban (Just-in-time)	P	Provide the most required part of the software to the correct time with quality. Based on pre defined user requirement.

Meth[1]: Methodology On[2]: Focus on
U: User P: Process D: Developer
Source: Author after Despa, 2014; Jamsheer, 2016; Lei et al., 2015

B. HydroGIS Tool Requirements

Pradeep and Wijesekara (2012) study has found that a requirement to develop a HydroGIS tool to manage the urban flash flood. Hence urban flash flood is a repercussion of urban land allotment modifications, the tool need to identify the affect of the modification on the stormwater generation. Further tool should allow the users to arrive dynamic engineering solution to manage excess stormwater. They have identified three process modules to carryout the entire task, (1) Incorporation of land parcel modifications (2) Calculated storm generations pre and post scenarios (3) incorporation of detention pit. This tool urged to have a user centric map based interface with on-screen capability in data input and dynamically modify the attributes/land parcel modifications. As well due to the manipulations are done in rather smaller urban land extents, the accuracy of the results becomes sensitive. The Important factor is, the potential users of the tool are non-hydrological land managers.

B. User Centric Design

As the users feel User Interface is the System, development of user interface for tools are very important. User-Centred Design (UCD) is the process of designing a tool, from the perspective of how it will be understood and used by a human user which formulated around 1980s. To learn the software users have to adapt their attitudes and behaviours. But when UCD, the software designed to assist potential users' existing attitudes, and behaviours. To achieve this, it places users at the centre of the design process from the stages of planning and designing the system requirements to implementing and testing the product. This result an efficient, satisfying, and user-friendly tool. (Abras et al., 2004; Baek et al., 2008)

The user experience design (UXD) and usability are the other two terms with the UCD. UXD needs to understand the users through a research. The research includes user observations, interviews, and different techniques to capture the users' emotions, motivations, and underlying concepts and beliefs. Then this knowledge will be used to develop user interfaces which align and support user behaviour. Usability is a measure of the interactive user experience associated with a user interface. It is a evaluation of user-friendliness which is easy-to-learn, and easy-to-use capability. The usability measures evaluating the users capability of use of the tool without any assistance. For the evaluation there are number of methods such as heuristic, cognitive walkthrough, Formal usability inspections, Pluralistic walkthroughs and etc; are available. The main aim of the usability evaluation is find the problems in user-friendliness and fix those before release the final product.("Introduction to User - Centered Design," 2017; Nielsen and Molich, 1990; Nielsen, 2012, 1994)

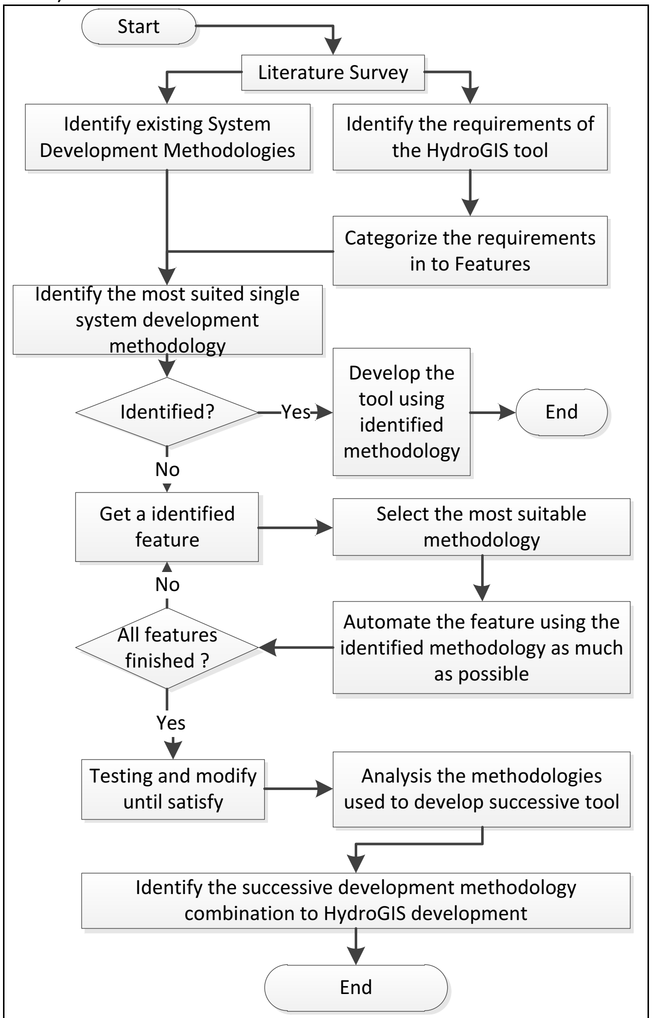

Figure 1.0 Overall Development Methodology

II. METHODOLOGY

A. Overall Methodology

The present work methodology flowchart is shown in the Figure 1.0. The work starts with the comprehensive literature survey to identify existing system development methodologies. Then, it identifies the requirements of the tool such as user needs, data needs, process / calculation needs and technology needs. After that, it categorises those requirements in to different features. In the next step, it analyses the system development methodology over the tools requirements to identify which methodology to be used. If it could be able to identify the methodology, then follow the same. Else, identify the suitable methodologies to automate the categorised feature by feature. Once all of the different features automated, integrate them together and carryout integration, system and acceptance testing till reach the satisfactory level, before release the tool. Once the tool release, progressively scan through the methodologies practises to develop the tool and identify successive path to development of an accurate and user friendly HydroGIS tool.

B. User Requirement Automation with Prototype

When user requirement automation, the tool underwent three kinds of user evaluations; software adequacy, formative and summative evaluations as shown in the Figure 2.0. Through the requirement analysis, it identified the basic functionalities of the tool then a prototype was developed. The developing prototypes were evaluated with the users through questionnaire. The objectives of the evaluations are shown in the Table 2.0. This software adequacy questionnaire evaluates the useability of the initial functions of the tool. Then with the user commented modifications second prototype was developed and again it was tested two times, which until satisfy the users. The acceptable prototype then integrated with the calculations modules. The final system was subjected to perform a summative evaluation. In the each evaluations, the tool was modified based on the view of usability. These process and the evaluations were align with the National Research Council (2007) guidelines of human-system integration in system development process.

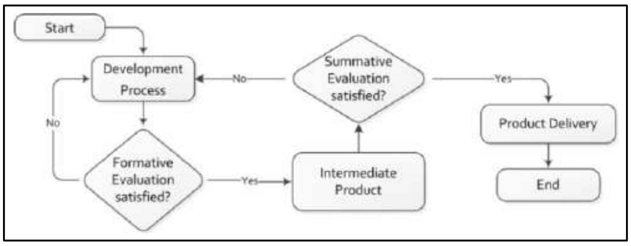

Figure 2.0 Prototype Development Process

Table 2: Basic Requirements gathered from the questionnaires

Basic Requirement	Question target to acquire the Users' satisfaction on
Assess the achievement of Objectives	Installation of the tool, Start the tool , Layer selection, Modify the selected layers and attributes, Update modifications , Modification of onscreen map and attributes, Do the modification in all four layers, Printed outputs generation, Secure the operation
Assess the usability of the developed GUI with GIS concepts / usage	Ease of navigation, Zooming and panning, Scale facility, Permit the user to navigate while keeping track of current reference frame, Provide tools for capturing, editing, and printing maps, Map in a larger percentage of the screen area
Assess the General Principles of GUI development	User centred design , Visual clarity, Consistency, Explicitness, Appropriate functionality, Flexibility and control, Error prevention and correction, Compatibility / Portability, User guidance and support, Informative feedback

Source : (Pradeep and Wijesekara, 2015)

C. Process development with waterfall development

In the calculation process automation, it identified the required hydrology models and calculation sequences. As well, a specific attention paid to automate the dynamic calculation modules. For that, it incorporates Rational Method to calculate composite runoff coefficient. As well, the concept of inflow hydrograph attenuation is being used to determine the detention storage size (Pradeep and Wijesekara, 2012). These two complex processes run based on the inputs from not only users or results of intermediate calculations but also GIS manipulation outcomes. Then the entire process run through manually and all the final and intermediate results were recorded in excel sheets. This process can be considered as the study the manual system and identify the bottle-necks in exiting system which is a step of waterfall development. Then automation started and each step evaluated against the results in the Excel sheets for accuracy. This become more difficult work, as once it found the errors in the manual system, then the manual system has to perform from the very beginning and as well, the automation had to reengineered. Then a considerable time spent to evaluate the manual process then start the automation. A module level evaluation of the software output against the manual results carried out throughout the automation. The development of foresaid excel sheet was a different research of University of Moratuwa.(Wijesinghe and Wijesekera, 2010)

D. Integrating the User Centric Tool with Process Centric Algorithm

Once the user friendliness reach to a acceptable level, 70%, and need to integrate the process for further evaluations, it decided to stop prototype modification. But the process automation carryout until reach 100% against the recoded set of results. Once it satisfied the process accuracy condition, then integrated the user interfaces with the process codes which have developed using same languages on same platforms. Whilst the integration, the user modification requests received at the 70% satisfaction level, were incorporated and went to the final summative evaluation.

IV. RESULTS & DISCUSSION

A. The development model - Process Centric Development to User Centric Tool (PcD.UcT)

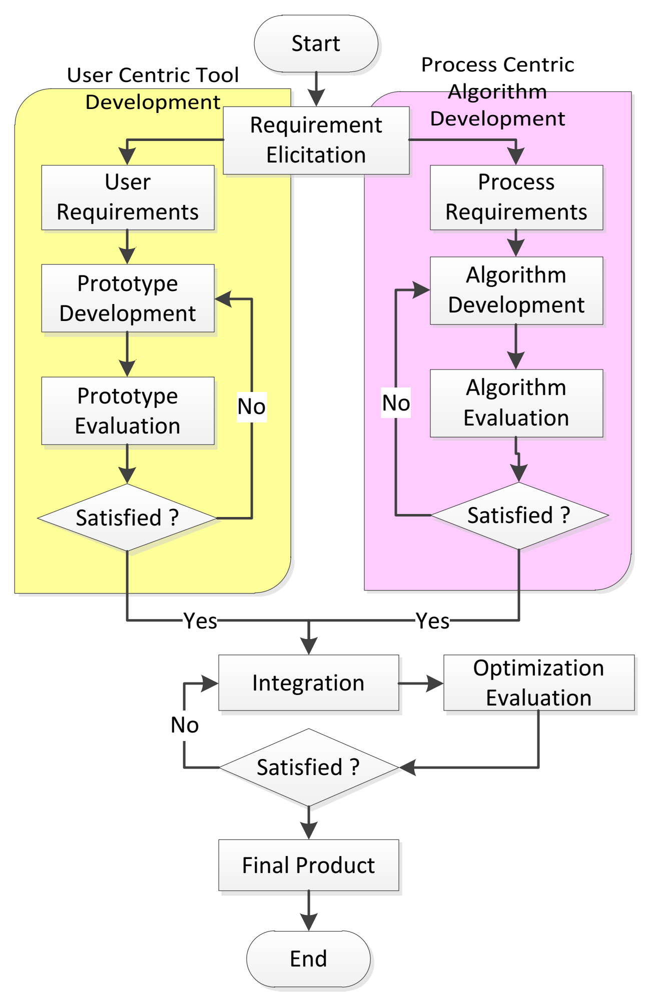

Figure 3.0 PcD.UcT Development Model

Once the satisfied tool resulted, it evaluates the software development methodology, which it has actually practised. Then it could be able to identify a hybrid development of waterfall and prototype development. The user centric tool development shows the characteristics of prototype development whilst the process centric algorithm development process shows the characteristics of waterfall development. The practised and proved software development methodology was named Process Centric Development to User Centric Tool (PcD.UcT) model and shown in the Figure 3.0.

B. User Centric Tool

The HydroGIS tool which developed through the described process shows a 92% user acceptance as shown in the Table 3.0. User evaluation 1,2 and 3 were performed during the user centric tool development and final evaluation was done at the optimization evaluation after the integration of prototype to process algorithms.

Table 3: Evaluations Result

Srl	Main Considerations and attributes	User Friendliness Evaluations			
		1st	2nd	3rd	4th
1	**Continuity in Operation**				
	Process Liberty of other GIS functions	70%	70%	78%	90%
2	**Error handling & Accuracy Confirmation**				
	Error prevention and correction	70%	72%	78%	98%
3	**GIS software version Compatibility**				
	Compatibility / Portability	70%	78%	78%	99%
4	**Information Security**				
	Spatial Data Security	0%	0%	35%	89%
5	**Non-GIS User Operation Capability**				
	Update modifications	80%	80%	80%	97%
	Flexibility and control	70%	70%	79%	88%
	Appropriate functionality	68%	70%	77%	90%
	Modify the selected layers and attributes	68%	77%	77%	97%
	Explicitness	67%	75%	76%	97%
6	**Easy operation Capability**				
	User centered design	65%	65%	73%	88%
	Consistency	55%	69%	71%	89%
	Informative feedback	5%	68%	70%	99%
	User guidance and support	3%	40%	49%	90%
	Printed outputs	3%	30%	43%	95%

Srl	Main Considerations and attributes	User Friendliness Evaluations			
		1st	2nd	3rd	4nd
	Provide tools for capturing, editing, and printing maps	3%	30%	43%	91%
7	On-Screen Operational Capability				
	Ease of navigation	70%	75%	78%	86%
	Scale facility	69%	75%	78%	90%
	Zooming and panning	68%	75%	78%	89%
	Modification of onscreen map and attributes	67%	72%	75%	93%
	Visual clarity	65%	68%	73%	98%
	Map display size	5%	65%	67%	89%
8	Tested and Verified Results				
	Modification in all required layers	65%	70%	75%	85%
	Average	50%	63%	70%	92%

Source : (Pradeep and Wijesekara, 2015)

C. Process Centric Development

The result has shown the followed methodology could able to provide 100% of the process. It has tested 80 test cases whilst the algorithm development and optimization evaluation. Result of five samples out of 80 carried out at the optimization evaluation are shown in the Table 4.0.

Table 4: Evaluation accuracy result

Test case No	Runoff coefficient (manual method)		Runoff coefficient (Tool)		Accuracy	
	Before Mod:	After mod:	Before mod:	After mod:	Before mod:	After mod:
1	0.548	0.695	0.548	0.695	100%	100%
2	0.984	0.816	0.984	0.816	100%	100%
3	0.327	0.364	0.327	0.364	100%	100%
4	0.228	0.389	0.228	0.389	100%	100%
5	0.200	0.376	0.200	0.376	100%	100%

D. Effect on Software Project Management

According to the practised development model (Figure 3.0) it has identified three different phases (1) Requirement clarification (2) Development and (3) Evaluation and modification. The recoded average weeks (1 week is equal to 14 hours effort of a programmer) taken for each phases are showing in the Table 5.0.

The Requirement Clarification of Prototype was taken much shorter time than the Process Development, as the requirement clarification is in the Process development was a hydrological model development activity.

In the development, it took average equal time for both development processes. However due to the requirement of following standards such as user-friendliness development and usability guidelines, prototype taken much longer time.

Whilst the evaluation, Prototype was taken longer time not only as it need to meet the users and get the feedback, but also need to satisfy users requirements which arises at each evaluation.

Table 5: Average Weeks Taken to Complete Phases

Phase	Prototype Development	Process Development
Requirement	2	5
Development	3	2
Evaluation	4	1

Considering the all three phases are having equal weights towards the completion of the development described in this sub section, Figure 4.0 shows the comparison of phases of two different developments over the time.

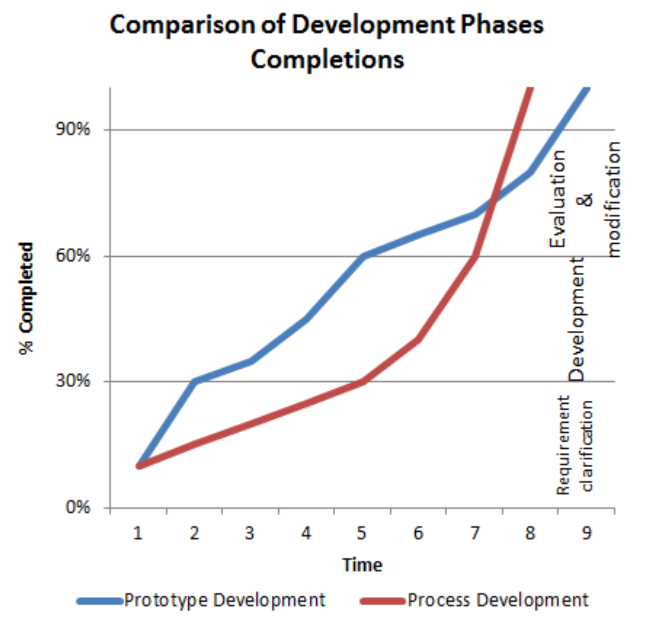

Figure 4.0 Development process completion comparison

E. Discussion

The present work attempted to develop the codes for process whilst developing the model. Then the work had trouble in time-taken reengineering with re-coding when a model has to correct. Hence, it has realised that requirement of tough patient, until the model development finish to start the coding of the processes / calculations.

Then without wasting time for model calibration, the work started the user interface development and evaluations for usability. Nevertheless, before finalise the prototype, sub process module out of 3 process modules were available to automate.

Then the present work started automates the process modules parallel to the prototype development that realised the methodology is an innovative use of available methodologies.

Even the model developers confirm the completion of the modules' accuracy, the work did not develop the user interfaces with the process automation, but carried out the automation of processes in software modules which can fix to prototype at anytime.

The work realised that, in each user evaluation a new user requirements are creating which are out of the scope of development. Due to this scope creeping requirements, more than 70% user friendliness was considered as an acceptable level to conclude the prototype development.

Once integrating the user interface to the developed process the final summative evaluation or the optimization evaluation was carried out. Then, as the tool is providing the actual results and satisfied the observations, which users made at the 2nd formative evaluation, user acceptance raised to 92%.

The processes and prototype were integrated once the acceptable levels of each reached. Then the expected errors were with only coding problems in integration. As it uses standards, publicly expressed variable descriptions and module level algorithm development, it was became easy task.

The time taken information in the Figure 4 shows late completion of the prototype development after the process development. Due to the two different natures of developments, it is not reasonable to match each other. However, Figure 4 shows the development phases of two developments start one after other. Hence this parallel development methodology utilise the developer for coding continuously which results early Software Development Project completion.

IV. CONCLUSION

The present work practised a hybrid development methodology, UcT.PcD, a combination of waterfall, prototype - repetitive development methodologies. This was a two parallel phased development methodology, (1) automate the engineering process through waterfall development and (2) develop the user interfaces using prototype - repetitive development methodologies. Once the accuracy reach to maximum and user-friendliness reach to a accepted levels, the two set of codes were integrated. The final evaluation carried out to confirm the success.

Then the present work demonstrate the capability of combine the approaches in system development methodologies that are following entirely different sequence to achieve the required outcome if the developers understating the desired outcomes.

REFERENCES

Abras, C., Maloney-Krichmar, D., Preece, J., 2004. User-centered design and evaluation of a next generation fixed-split ergonomic keyboard. Bainbridge, W. Encycl. Human-Computer Interact. Thousand Oaks Sage Publ. 37, 445–456. doi:10.3233/WOR-2010-1109

Avison, D., Fitzgerald, G., 2006. Methodologies for Developing Information Systems : A Historical Perspective. Past Futur. Inf. Syst. 1976–2006 Beyond,Springer US 27–38. doi:10.1007/978-0-387-34732-5_3

Baek, E.-O., Cagiltay, K., Boling, E., Frick, T., 2008. User-centered design and development. Handb. Res. Educ. Commun. Technol. 1, 660–668. doi:10.1145/1273961.1273973

Chemjong, H., Wijesekara, N.T.S., 2017. Evaluation of Irrigation Water Issue Practice for Better Water Management at Rajangana Reservoir, Sri Lanka, in: UMCSAWM Water Conference. UNESCO Madanjeet Singh Centre for South Asia Water Management, Moratuwa, pp. 41–51.

Dahanayake, A.C., Wijesekera, N.T.S., 2017. Watersheds with an Emphasis on Multi User Concepts – Demonstrating an Application at a Watershed in Dampe, Sri Lanka, in: UMCSAWM Water Conference. UNESCO Madanjeet Singh Centre for South Asia Water Management (UMCSAWM), Moratuwa, pp. 21–28.

Dennis, A., Wixom, B.H., Tegarden, D., Wiley, J., Tavares, K., Wolfman-robichaud, S., Dodds, K., 2009. Systems Analysis and Design with UML Version 2.0, 3rd Editio. ed. John Wiley & Sons, Inc.

Despa, M.L., 2014. Comparative study on software development methodologies. Database Syst. J. V, 37–56.

Fowler, M., 2001. The New Methodology. Wuhan Univ. J. Nat. Sci. 6, 12–24.

Introduction to User - Centered Design [WWW Document], 2017. . Usability First. URL http://www.usabilityfirst.com/about-usability/introduction-to-user-centered-design/ (accessed 3.3.17).

Jamsheer, K., 2016. 12 Best Software Development Methodologies with Pros & Cons. Softw. Dev. India.

Keerthirathne, W.H., Wijesekara, N.T.S., 2017. Determination of a Design Rainfall Pattern by Comparing with its Effect on Streamflow on Greater Colombo Watershed in Sri Lanka, in: UMCSAWM Water Conference. UNESCO Madanjeet Singh Centre for South Asia Water Management, Moratuwa, pp. 35–40.

Lei, H., Ganjeizadeh, F., Jayachandran, P.K., Ozcan, P., 2015. A statistical analysis of the effects of Scrum and Kanban on software development projects. Robot. Comput. Integr. Manuf. 43, 59–67. doi:10.1016/j.rcim.2015.12.001

Maidment, D.R., 1992. Grid-based Computation of Runoff : A Preliminary Assessment. Hydrologic Engieering Center, Davis, California.

Meyer, B., 2013. The origin of "Software engineering" [WWW Document]. URL https://bertrandmeyer.com/2013/04/04/the-origin-of-software-engineering/ (accessed 3.1.17).

Mohagheghi, P., 2008. Evaluating Software Development Methodologies Based on their Practices and Promises, in: New Trends in Software Methodologies, Tools and Techniques. pp. 14–35. doi:10.3233/978-1-58603-916-5-14

National Research Council, 2007. Human-System Integration in the System Development Process: A New Look. The National Acedemies Press, Washington D.C.

Nielsen, J., 2012. Usability 101 : Introduction to Usability [WWW Document]. Nielsen Norman Gr. URL https://www.nngroup.com/articles/usability-101-introduction-to-usability/ (accessed 4.17.17).

Nielsen, J., 1994. Usability inspection methods, in: Conference Companion on Human Factors in Computing Systems - CHI '94. pp. 413–414. doi:10.1145/259963.260531

Nielsen, J., Molich, R., 1990. Heuristic Evaluation of user interfaces, in: CHI '90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 249–256. doi:10.1145/97243.97281

Picek, R., 2009. Suitability of Modern Software Development Methodologies for Model Driven Development. J. Inf. Organ. Sci. 33, 285–295.

Pradeep, R.M.M., Wijesekara, N.T.S., 2015. Modification of User Friendliness in to a HydroGIS Tool, in: 8th Intrernational Research Conference 2015. Sir John Kothalawala Defence University, Kandawala, Ratmalana.

Pradeep, R.M.M., Wijesekara, N.T.S., 2012. GIS Tool to Mitigate Urban Flash Floods Through Capturing Land Allotment Modifications and Incorporating Conceptual Detention Pit, in: ITRU 2012. Universiry of Moratuwa, Moratuwa.

Sakthivadivel, R., Fernando, N., Brewer, J.D., 1997. Rehabilitation Planning for Small Tanks in Cascades: A Methodology Based on Rapid Assessment. Colombo.

Shmueli, O., Ronen, B., 2017. Excessive software development: Practices and penalties. Int. J. Proj. Manag. 35, 13–27. doi:10.1016/j.ijproman.2016.10.002

Sui, D.Z., Maggio, R.C., 1999. Integrating GIS with hydrological modeling: Practices, problems, and prospects. Comput. Environ. Urban Syst. 23, 33–51. doi:10.1016/S0198-9715(98)00052-0

Thakuri, P.S., Wijesekara, N.T.S., 2017. Climate Change Impacts and Adaptation Measures for Pahala Divul Wewa, Anuradhapura, Sri Lanka, in: UMCSAWM Water Conference. UNESCO Madanjeet Singh Centre for South Asia Water Management, Moratuwa, pp. 59–62.

Wijesinghe, W.M.D., Wijesekera, N.T.S., 2010. Reserch Monograph - Conceptual Detention Storage for Stormwater Management. Moratuwa.