Development of Software-Based University Research Tools for Metocean Engineering Applications: A Reflective Case Study

E.S.Lim[1], M.S.Liew[1], G.Dinis Jr.[1]
[1]Universiti Teknologi PETRONAS, Perak, Malaysia

Abstract:
The oil and gas industry has traditionally been one of the most demanding forms of engineering as the lucrative returns form the basis of global development. As such, oil operators have made it a norm to reinvest a significant portion of their profits into research and development (R&D). This stemmed as a result of depleting natural resources which has forced operators to go further and deeper to explore for hydrocarbons. To balance the economics of such ventures, R&D plays a critical role in optimization and defining standards in which to operate safely with economical consideration. As such, various software tools for various disciplines have been developed for this purpose, i.e. SESAM, SACS and etc. However, there has been a lack of R&D tools that have been tailor-made for metocean operations; most of those that are currently in existence are not open to public use/sales. This has created a demand by Malaysian oil operators to have such tools being readily available for in-house use. The paper herein will discuss the framework and development of an integrated and tailor-made metocean software, namely Blue Hive (BH).

Index term *–Statistical analysis, metocean, software development, .NET, Blue Hive*

I. INTRODUCTION

Metocean is an integral part ofoffshore engineering and has alwaysbeen significantly more complex in design compared to land-based civil engineering works due to the fact that it is based in conditions where fluid dynamics develop dominant forces on the structure. The stochastic nature of wave conditions are coupled with the effect of ocean currents on top of the more extreme wind conditions prevalent in open seas. As such, a strong understanding or fundamental of these environmentalmetocean loads at sea form a critical component in almost every stage of an offshore facility's life cycle. These stages include the design of facilities and forecasting of operations for offshore vessels. The understanding of these metocean loads will be critical from two ends of the engineering consideration, which is to prevent loss of life as a result of structural failure or capsizing (underdesign) and to prevent the overdesign of offshore structures (which results in excessive usage of steel in fabrication).

Strong understanding of metocean loads is essential in aiding the optimization of engineering design and definitions. This can only be achieved via investment into dedicated R&D divisions or institutions. For example,multinational operatorssuch as Shell has a R&D division known as Shell Global Solutions which have a running contract with Fugro GEOS to conduct metocean studies and redefinitions. There are however also operators in the region who do not have extensive metocean research units and as such would have to leverage on existing codes and standards that may not be optimized for the region. For example, most of South East Asian (SEA) operations are very much dependent on the American Petroleum Institute (API) standard which bases itself on Gulf of Mexico conditions (which are far more conservative than SEA). What compounds the situation is that there is little sharing of developed metocean knowledge amongst operators which makes individualized research efforts critical to establishing optimized metocean standards.

This case study is based upon the efforts of Universiti Teknologi PETRONAS (UTP), Malaysia in particular to develop localized metocean definitions for one of the biggest oil operators in SEA, PETRONAS. The effort is a fast-track initial research that spans approximately 1-year long in which the deliverables are to develop the following metocean tools, a) correlation factoring between measured and hindcast data, b) joint density analysis of metocean parameters, c) spectral analysis and modeling of metocean parameters and d) ARIMA modeling and forecasting of metocean parameters. In order to undertake the research, there was a need to perform extensive statistical data analysis and time series analysis. Throughout the initial phases of the research, there was a need to constantly interoperate between software, i.e. SPSS PASW, Excel and MATLAB.

While this may be acceptable to some, efficiency of research works can be greatly enhanced if there is an integrated platform that combined the benefits of each program such as MATLAB's powerful scripting options and the user-friendly interface of SPSS PASW which makes the input-output process seamless. Since the format of data being input into the software is standardized across PETRONAS' operating units, there was room for improved efficiency by cutting down steps required during the data import process. Moreover, it was envisioned that there would be a need to materialize the algorithms and methodologies employed in research into commercially usable software. As such, the gears of motion were set in place to develop Blue Hive (BH) which is intended to integrate metocean analysis specific tools under a seamless input-output process. BH is also intended to be the precursor to future and continually evolving research in metocean engineering such as transfer function modeling.

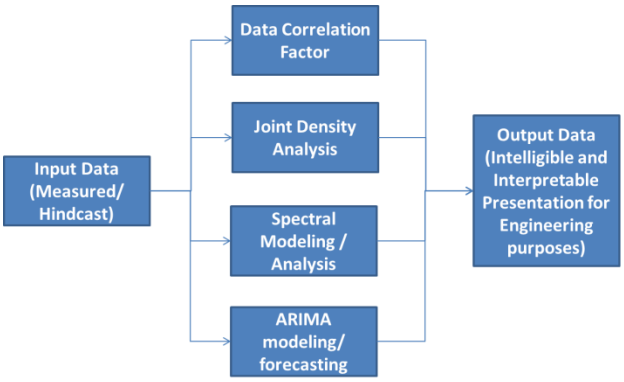

Fig. 1.Proposed process flow for BH

II. FRAMEWORK, METHODOLOGY AND MODEL

The buzzwords framework, methodology, model and even process, are not alien to any development effort. Despite being routinely used, these terms do not always carry the same meaning for everyone that uses them.Based on common definition, the terms are defined as follows:

2.1 Framework:an architectural skeleton of what should be done, or stages that should be walked through without defining any activities or tasks within them.

2.2 Methodology: the definition of what should be done in each stage of a development effort, without defining how to do it.

2.3 Process: a step by step guide on how to perform a task.

2.4 Model: the representation or simulation of a software process.

Building software, i.e. software engineering, is a highly debated topic in the industry. Although software engineering was created based on the foundation of common engineering, development of software systemsis influenced by unique factors that require special attentionapart from traditional engineering conceptsbased on sequential procedures which are not always able to address these factors effectively (Bern, et al., 2007).The customer, the business environment, and the organizational structures are examples of such factors, as they influence software projects in a non-sequential or structured manner (Bern, et al, 2007).Thus, the emergence of new frameworks and models, i.e. Agile models,for the past twenty years can be seen as an attempt to try and address these issues(Mnkandla, 2009 and Bernetal., 2007); large scale projects do however still work better with older methodologies.

Metocean engineering is highly statistical in nature and requires extensive data analysis tools. The purpose of BH is to incorporate whatever methodologies for data analysis that are developed by the team of researchers into a simplified User Experience (UX), that brings about results for immediate usage in an industrial environment situation, such as the ones mentioned in *Section I.* Software engineering can be a volatile process, where the needs of the customers rapidly change, and either new components need to be added or removed to accommodate new features or completely remove an existing one(Qureshi & Hussain, 2008). Adding to this complexity is the fact that research in itself is not a static process in nature, thus existing components are also liable to frequent modification and adaption, as experienced during the developmental cycle of this project. This trend is prevalent in almost all fields of research as progress in research will constantly see improvements and redefinitions to the engineering technique in which the software is to replicate. Despite all of these, there is still a need to have a framework defined that will allow developers to know what

the end goal is, or at least what it should look like, while maintaining the flexibility of development demanded by the nature of their work. To achieve this, the team opted to follow theAgile framework, using eXtreme Programming (XP) programming methodology, with a component based modeling approach.

The Agile framework, developed by the Agile Alliance back in 2001 in response to the growing demands of the software market,was set to value individuals and interactions, working software, customer collaboration and responding to change; as opposed to processes and tools, comprehensive documentation, contract negotiation and following a plan, respectively (Agile Alliance, 2001). The framework permits construction of a system architecturethat is fixed in terms of its structure while being highly flexible in terms of its internal components. This isvery suitable for the developmentenvironmentwhere in-situ change is pertinent, as opposed to traditional frameworkswhich demanded a fixed software structure, components definition,procedures and end goals.Traditional frameworks are complemented by extensive documentation on what was done throughout each stage, making accommodation to changes virtually impossible (Munassar &Govardhan, 2010).

XP, a methodology originally designed for usage by small teams in projects with non-clearly defined and mutating requirements, was chosen to be used, because of its main core elements: communication, feedback, simplicity, and courage, and the fact that is highly suitable for small development teams, as is the case in this project, where there is more emphasis on customer communication compared to anything else. Now, XP defines that projects should be carried out in periodic cycles, ofthree (3) weeks, and at each cycle a different component or feature is addressed (Wolak, 2003). In research, such periodicity is not easy to achieve, nonetheless, the separation of each feature-cycle is desirable as at each cycle, each feature implemented has some degree of independency from the others and the irregularity of time frames between each feature–cycle does not affect the objectives of the project. Contrarily, it helps suit the dynamic and irregular schedule of research work which is very much reliant on the dynamic requirements of clients in which they are recipients of the deliverables. A research was conducted to quantify the results of XP usage in several fields of development across Europe, USA, Asia, and Australia, in companies both young and mature, and found an almost one hundred percent satisfaction rating from a total of 45 respondents(Rumpe and Schroder, 2001). This pattern was seen even for large teams, i.e. teams with more than 10 people. It is also worth mentioning is the fact that 73% of the projects interviewed were new, and made use of high level languages, as in thisprojects' case, where the main language in use is C#. When asked to rate the factors of time delivery, costs to last minute changes and quality, the respondents gave ratings between 3.77 and 4.44 on a -5 to 5 scale for finished and running projects, indicating positive perceptions of XP's effectiveness.

The nature of this project is based on constant exchange and feedback between customers, i.e. the researchers, and the developers. On-site customer presence is one of the main principles of XP. In the same study by Rumpe and Schroder, it was found that the absence of on-site customer was reported as the second highest risk factor to XP projects. Studies on performance of XPby researchers were able to adopt this principle and reported it to work "extremely well"(Ganeshan and Ganesham, 2003). Primavera Software, a project management portfolio vendor now owned by Oracle Corporation, managed to successfully increase their customers satisfaction base, establish a highly motivated development environment and produce working and reliable software by adopting Scrum and XP agile practices back in 2003, which helped save the company from its low productivity and customer satisfaction ratings at the time(Object Mentor, Inc. and Advanced Development Methods, 2004).

Finally, the component based approach is chief for the entire purpose of BH. As a standalone application package being developed to address various analytical needs in the field of metocean data analysis, BH needs to have a solid foundation that allows it to support different modules that are independent of each other for operation. In section III, the architecture of the system will illustrate this better. Here, we will limit the discussion to the need of using this approach, as well as its advantages and drawbacks. As previously mentioned, BH was designed to support a variety of metocean data analysis computational needs, without much binding in between different modules, so that each module could be setup and used without requiring the others to be present. This would permit both concurrent development of different modules and distribution or release of different modules at different stages for either testing or usage. The concept is widely adopted, not only by statistical software package vendors such as MathWorks, but by other industry area software providers, such as Adobe Systems.This is one of the main development goals in the project, along with having each module capable of producing ready to use interpreted results from the analysis processes they perform. Looking at generic analysis software packages such as IBM's Statistical Package for the Social Sciences (SPSS) and Microsoft Excel, they have the distinct characteristic of providing a basic platform for analysts to define

procedures based on their needs. The issue with such approach is that there is learning curve required for each package, and in some cases, the users have to familiarize themselves with defining computational procedures on a machine due to these packages offering powerful and elaborate scripting options. This is seen in software such as MATLAB that requires comprehensive control and knowledge of the scripts where BH would only be a matter of point-and-click the required options. This advantage nests itself when there is a need to shorten the project duration by the client; it however comes at the cost of inability to create customized scripts. This was a compromise that was accepted by BH as the multitude of metocean research projects that it serves had a lifecycle of no more than 1 year.As such, rapid mastery and development of software is required. Another issue that arises is that, more than often,a single package simply does not provide all the functionality required, as it was in the application of the aforementioned metocean projects. In order to perform the required analysis processes required for the studies being carried out, there was a constant need to be interchanging between one package and the other.In an environment where results are demanded at a fast rate, there is a need to shorten the time it takes for one to obtain intelligible results that can be interpreted and utilized almost immediately.

III. SOFTWARE ARCHITECTURE

BH was designed as a .NET application, to take full advantage of the software framework developed by Microsoft for Windows. The .NET is cited by some as the best Windows development framework (Magenic, 2012)for the variety of advantages that it offers to businesses and developers.Its most prominent features of interest to the developmentare the (1) extensive number of classes readily available for usage, (2) the short time span required from development under any of its languages, with interoperability among all of them, (3) the fact that it allows applications to be designed for divergent purposes at the same time, without requiring much effort to make them run on different environments, i.e. desktop, web-based and mobile apps, and (4) the existence of an open source community supported by Microsoft, that is dedicated to building .NET projects (Magenic, 2012).What follows is the description of how these features were used in relevance to the project.First, the .NEToffers many classes for basic and low level program control requirements such as data management, network connectivity, and UX design.As mentioned previously, BH needed to be built on a solid platform that would allow it to define the lower level features required by all components in such a way that independent modules could be built on top of this platform without much effort. With the .NET, the team did not have many difficulties in this process, given the wide array of predefined classes and data structures already available in the framework, e.g. *System.IO* module and the *DataTable* class.

Second, one of the most time consuming, yet required, development activities are those carried out in the stages of coding and debugging, and currently, there is no other framework that delivers a better system for both, than the .NET combined with the Visual Studio (Magenic). From this project, this was easily verified, as most debugging environments lack the high level user communication efficiency of Visual Studio. With the Visual Studio Debugger, present in VS2005 onwards, useful functions such as code stepping in, stepping out and over, and parallel code debugging are intuitively available, apart from the standard functions such as break-points and value modification.

Third, as team's intents of usage environment for BH is still in the process of redefinition as the research progresses, there was a need for a framework that would allow the team to build application logic that could easily be ported onto any environment of choice. Obviously, this is possible with other languages such as C and C++through binding. However, with the .NET, binding would be virtually unnecessary, as the framework natively supports a desktop, web application, and mobile environment. Finally, given that teams' needs were to develop a solid foundation that could support independent modules, the possibility of using existing open source libraries to shorten development time and costs on non-critical system modules could not be ignored. Microsoft currently supports an open source community for .NET projects, i.e. Codeplex, where useful resources were found and adopted to shorten the time spent on some system modules so focus could be given to more critical ones.

Each component shown in Fig. 2was built as a standalone library. We have the option of building our needed libraries using any of the .NET languages (F#, C#, VB), and use them all for any system component. The level of abstraction for each computing module is given by the Application Processing Logic (APL) componentwhich interfaces with the REPORT, UX and COREcomponents of the system. As for the remaining components, they were built on top of one another to provide those four (4) components with the services they need. Onthe top of diagram, we have the UX component, which provides a layer of abstraction for the application's environment. Currently, BH is being developed as a desktop-application; however porting it into a web-based system in ASP.NET would be feasible, requiring only changes to be made to the upper layer, i.e. the design of new interfaces for the target environment.It is believed that this architecture design addresses both the

current and future needs of BH, as it sets a robust framework for the addition, removal and modification of any specific application analysis module without the need to modify the lower level layers of the system. Additionally, the architecture model allows for ease of integration with third party libraries, which provides us with considerable savings in terms of time and costs.

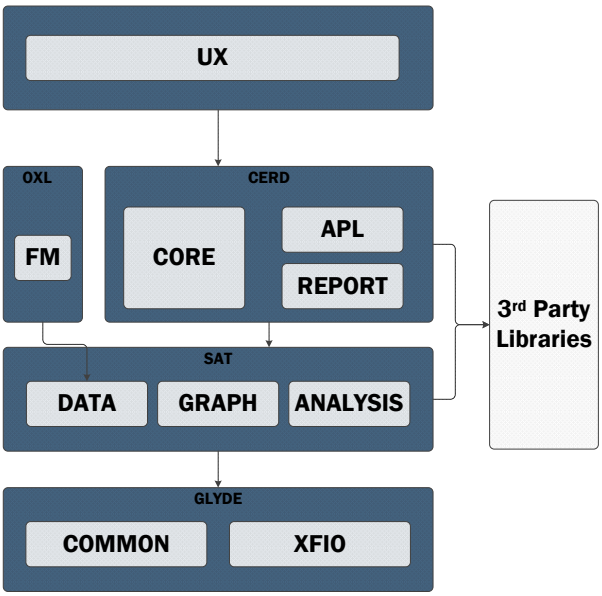

Fig. 2. Software architecture model of BH

IV. CONCLUSION

The BH softwarein essence is targeted to be a fast-track research tool that serves the purpose of fulfilling the deliverables of university-industry collaborations. As such, the utilization of Agile frameworks and XP is essential to achieving such goals in short term. This is needed as the nature of such research projects are dynamic in nature as methodologies of research and algorithms employed may evolve as the deliverables reach maturity. Moreover, BH is intended to be the precursor to long-term development of commercially useable software for oil operators in SEA by providing alternative figures to existing metocean practices on top of BH also potentially evolving into more advanced research tools in similar areas.

REFERENCES

[1] Mnkandla, E. (2009). About Software Engineering Frameworks and Methologies. IEEE AFRICON 2009. September 23-25, 2009. Retrieved 5th September, 2012

[2] Bern, A., Nikula, U., Pasi, S. J. A., Smolander, K. (2007) Contextual Factors Affecting the Software Development Process. Proceedings of the 2nd AIS SIGSAND European Symposium on Systems Analysis and Design, Gdansk, Poland, June 5, 2007

[3] Munassar, N. M. A., and Govardhan, A. (September, 2010). A Comparison Between Five Models Of Software Engineering. IJCSI International Journal of Computer Science Issues. Vol 7(5). pp94-101

[4] Qureshi, M. R. J., & Hussain, S.A. (2008). A reusable software component-based development model. Advances in Engineering Software. Vol 39(2). pp88-94

[5] Agile Alliance. (2001). Manifesto for Agile Software Development. Retrieved 5th September, 2012. From http://www.agilealliance.org/the-alliance/the-agile-manifesto/

[6] Ganeshan, K., Ganesham, V. (2003). Extreme Programming - A Success Story. Poster Paper in Proceedings of the 16th Annual NACCQ, Palmerston North New Zeland, July 2003. Retrieved 5th September, 2012. From www.naccq.ac.nz/conferences/2003/papers/483.pdf

[7] Magenic. (2012). Why .NET. Retrieved 5th September, 2012. From http://magenic.com/Portals/0/Magenic-White-Paper-Why-NET.pdf

[8] Object Mentor, Inc. and Advanced Development Methods. (2004)Best Practices in Scrum Project Management and XP Agile Software Development. Retrieved 5th September, 2012. From http://www.objectmentor.com/resources/articles/Primavera.pdf

[9] Rumpe, B., Schroder, A. (2001). Quantitative Survey on Extreme Programming Projects. Munich University of Technology. Retrieved 6th september, 2012. Fromhttp://www.se-rwth.de/~rumpe/publications/ps/XP02.RumpeSchroeder.Study.pdf

[10] Wolak, R. G.(2001). DISS 725 - System Development: Research Paper 1 SDLC on a Diet. Nova Southeastern University. Retrieved 5th September, 2012. From http://scisstudyguides.addr.com/papers/rwDISS725researchpaper1.pdf