ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

REVIEW PAPER

Available Online at www.ijarcs.info

A Metric Based Approach for Managing Architecture-Related Impediments

Harshavardhan Metla
Department of computer science,
VIT University,
Vellore, India.

Dharmendhra makineni
Department of computer science,
VIT University,
Vellore, India.

Manikanta Ellenki
Department of computer science,
VIT University,
Vellore, India.

Pranay Ratakonda,
Department of computer science,
VIT University,
Vellore, India.

Abstract: A Quantitative measure of the degree to which a system, component, process possesses given attribute. Metric is first required to allow to measure the resource and energy consumption of various machines in production system to manage the architecture. It also contains classification of energy users and a comparison between machine tools and other tools. The major focus will be kept on optimization of design-, modelling, simulation and evaluation of product before any architecture will built or any modifications to the existing architecture. Anything that slows down the software development is called Impediment. It is a issue which may include risks typically high probability risks. Every software is not much efficient, there are some impediments present in it which can be resolved.

Keywords: Quantitative; metrics; architecture; optimization; development.

I. INTRODUCTION

The number of metrics and tools for the assessment of object oriented project is increasing. In the past years efforts were made defining new metrics has not been followed by a comparable effort in establishing methods and procedures for their systematic application. To make the investment on project assessment effective, specific methods and tools for product and process control have to be applied and customized on the basis of specific needs. The main scope is not only to deal with the technical issues for managing an architecture but also discuss how the integrated frame works are applied to support the future generation frame works. The tool and method have been defined in years of work in identifying tool features and general guidelines to define a modus operandi with metrics, with a special care to detect analysis and design problems as soon as possible, and for effort estimation and prediction. The adaptability of the program is evaluated quantitatively using a requirements volatility measure and the probability of correct recognition.

The best example of the hypermedia system is the world-wide web with high number of organizations and developing thousands of commercial and educational web sites. Web was the delivery platform for web hypermedia and web applications. The cost for the web development is difficult to estimate because the web development processes are vary from the existing approaches. The main objective of the web projects are to design and release into the market very quickly. Till today very few papers had proposed web size metrics about web cost estimation. The two common metrics that have been taken into account were the total number of web pages and what are the features provided by the application.

To improve the quality of the software which is delivered, management of the software process is one of the key factor. The Sequential steps of the any software process which delivers the output software is called Software Process Model. The software process model is divided into mainly two categories. They are descriptive models and active models . Descriptive models are meant for describing the processes and their behavior and relationship among them. Active models are meant for constructing the executable systems that supports by enabling the process. To describe the software process model some suitable metrics are required.

II. LITERATURE SURVEY

We need a methodology consisting of notations, methods/techniques, and guidelines, that also allows for establishing traceability to the “whys” of the techniques the requirements. Our partial survey of the existing literature on adaptation has led us to categorize techniques used to deal with adaptation into the following: architecture-based techniques, component-based techniques, code-based techniques, genetic algorithm techniques, dynamic adaptation techniques, and adaptation methodologies. A comprehensive adaptation technique that spans various adaptation requirements is given in. In this technique [1], an adaptable system has embedded in it two managers – one for adaptation and the other for evolution. The adaptation manager takes high-level decisions which are implemented by the evolution manager; the entire process is iterative. In a framework for real-time software system adaptation, called RSAS (Real-time Software Adaptation System) is proposed. This framework permits a real-time system to adapt to timing constraints and to hardware failures. In an adaptive software architecture (called multi-graph architecture) for digital signal processing applications has been developed. This architecture [2], which is a signal flow graph, permits dynamic changing of graph nodes, to alter the execution sequence on the fly. In the Odyssey, architecture is proposed. Here, the operating system controls the fidelities of various applications running on a mobile phone based on the rate of power consumption, the aim being to conserve power as much as possible or as much as required by the user of the mobile. In the Simplex Architecture has been described. This architecture permits the online evolution of real-time systems by using a middleware that talks to the various components of the architecture using the publish/subscribe mechanism. This mechanism lets new components be created on the fly and replace existing components. In the VEHICLES developing an environment of NASA has been described.

Vehicles provides a flexible developing environment which has been used for developing mission-critical systems.

III. RELATED WORK

This section explores distinct papers, similar to our own. These works are focused on the distribution of responsibilities between context sensitive service platforms, such as broadband or number of clients connected. Put another way, they don't take the quality attributes into account. The ACRM (Architectural Runtime Configuration Management) approach creates a model capturing the adaptable system configurations and corresponding behavior and organizes them into a historical graph of configurations. [3] To complement this model along with the adaptable processes metadata, ARCM creates a historic perspective of process adaptations. ARCM also provides active controls to undo an operation or activate a saved configuration. This work is focused on the administration and visualization of said running adaptions.

In a method based on quality attribute scenarios to find and analyze to- potential points of self-adaptation in software architecture during the design stage is proposed. Extend an ADL called ABC / ADL, to store the architecture information. Information is used directly by a reflective -based middleware architecture, called PKUAS, for making self-adjustments in implementation. Some limitations in this proposal are using EJB components. While the use of these components is not really a limitation at the functionality level, it certainly is an interoperability and scalability one. In a word, this proposal is tied to implementation of technology. This proposal is not clear about the possibilities of including new components. The system design is separate from the implementation, so the implementation may be different from the solutions proposed in the architecture. In practice [4], this should not happen, but it is a risk taken to manage this separation. The year 1960 opens a decade in which the practice of project management in the modern sense gained a foothold. In the years before, projects were championed by different professional and occupational groups, using the knowledge and experience which they acquired in their chosen lines of work.

Architects applied them under-standing of space, aesthetics, materials and construction techniques in managing projects. General contractors used their knowledge of construction trades to organize field works. Seasoned procurement officers controlled major military acquisitions by relying on their hard-won knowledge of procurement [5]. With the possible exception of Gantt charts, there were no generally recognized management methods and tools associated with project delivery. Project management as a body of knowledge did not exist.

In the earliest development which concerns us, Henry Gantt, a mechanical engineer, and management consultant developed the Gantt chart in the 1910s. Gantt charts were employed on major infrastructure projects including dams and high- ways and remain to this day an important tool in project management. Gantt's charts [6] showed the actual working time for each day and the cumulative working time for a week. Each row of the chart corresponded to an individual machine or operator. Curiously, unlike today, tasks performed were not displayed on the charts.

In the decades prior to 1960, some critical developments occurred which laid the foundations for the modern integrative practice of project management. The General Systems Theory (GST) was first developed by Karl Ludwig von Bertalanffy in the 1930s. He gave expression to the sense of the widespread interconnectedness of objects and phenomena. [7]In October 1954 von Bertalanffy, Kenneth Boulding, Ralph Gerard and Anatol Rapoport founded the Society for the Advancement of General Systems Theory. Having been inspired by studies of living organisms, von Bertalanffy proposed the GST as a paradigm for controlling model construction in all of the sciences in qualitative non- formalized terms. The new system concept was to represent a set of interrelated components, a complex entity in space–time, which tends to restore itself after disturbances.

IV. METRICS OF OBJECT ORIENTED SYSTEMS

For a narrowly-focused presentation of the existing OO metrics [8], we use our general metrics classification as

A. ***PROCESS METRICS***
1. Maturity Metrics
- organization metrics
- resources, personnel and training metrics
- technology management metrics
- documented standards metrics
- process controlling metrics
- data management and analysis
2. Management Metrics
- milestone metrics
- risks metrics
- workflow metrics
- controlling metrics
- management data base metrics
- quality management metrics
- configuration management.
3. Life Cycle Metrics
- problem definition metrics
- requirement analysis and specification metrics
- design metrics
- implementation metrics
- maintenance metrics

B. ***PRODUCT METRICS***
1. Size Metrics
- elements counting
- development size metrics
- size of components metrics

2. Architecture Metrics
- components metrics
- architecture characteristics
- architecture standards metrics
3. Structure Metrics
- component characteristics
- structure characteristics
- psychological rules metrics
4. Quality Metrics
- functionality metrics
- reliability metrics
- usability metrics
- efficiency metrics
- maintainability metrics
- portability metrics
5. Complexity Metrics
- computational complexity metrics
- psychological complexity metrics

C. ***RESOURCES METRICS***
1. Personnel Metrics
- programmer experience metrics
- communication level metrics
- productivity metrics
- team structure metrics
2. Software Metrics
- performance metrics
- paradigm metrics
- replacement metrics
3. Hardware Metrics
- performance metrics
- reliability metrics
- availability metrics

Based on the recent work on OO metrics, we can establish the following metrics to evaluate the OO(OO – Object Oriented) products and the processes including some empirical evaluations.

V. WEB SIZE METRICS

Size is represented with these three attributes. They are length, functionality and complexity. The attribute length measures the physical size of the application. The attribute functionality measures the functions which are provided by the application to the user. The attribute complexity is measured by the structure of the application [9]. Stratum and Compactness are both the measures which are calculated from the structure of the hyper document. The value of the compactness varies between 0 and 1 and it tries to measure the how better the links are connected. The value of the stratum varies between 0 and 1 which measures the degree of the reading order is imposed on the user. The value 0 refers to non- imposed reading order whereas 1 refers to linear hypermedia.

Another important three complexity measures are :

1) *Interface Shallowness measure*
2) Downward compactness measure
3) Downward Navigability measure

Interface Shallowness measure: It measures the weight of the load on the users. The main basic idea is such that nodes are connected by links in such a way that they may or may not preserve interface linearity.

Downward Compactness measure: It measures the complexity of structure of reaching the nodes from the root.

Downward navigability Measure: It measures the navigability of hypermedia. In this an hypermedia application has a shallow interface layer from roots to the nodes and is very compact from nodes to the root.

Path complexity, tree impurity, modularity, individual nodes complexity are complexity measures proposed by Hatzimanikatis et al. Path complexity is a minimal linear hyper document which is measured by number of varied paths or cycles found in the hyper document [10]. Tree impurity is the measure of a graph which is deviating from being a tree. Modularity measures the dependency of the nodes. Individual node complexity is measured as single mode that imposes on overall hypermedia structure. Inbound and outbound connections are the another type of complexity measures .Outbound connection measures the number of external links. Inbound connection measures the number of links from the other links.

There are some web size measures specially for hypermedia applications. They are:

Length measures:

->Page Count
->Media Count
->Program Count
->Total Page Allocation
->Total media Allocation.

Complexity measures

->Connectivity
->Connectivity Density
->Cyclomatic complexity

VI. METRICS FOR SOFTWARE PROCESS MODELS

There are some set of metrics for Software process models in order to check the software maintainability. The metrics which are defined for software process model are model scope metrics. These model scope metrics measures the structure properties of the software process model. A software process model which has high degree of structure complexity is very difficult to maintain [11], in that situation model scope metrics are good scope for maintainence of software.

1) *Analyzability:*

Analyzability is about the easiness of finding deficiencies or errors in the model which are to be modified.

2) *Understandability:*

Understandability is all about how much faster we can understand the model .

3) *Modifiability:*

Modifiability is all about how easily we can modify the errors and deficiencies for some specific enhancements or for some new requirements

VII. METRICS ASSESSMENTS IN ELECTRICITY SECTOR

The most important challenge that arises while doing a metric assessment in electricity sector is to choose a benchmark factor in comparisons. It means we have to measure different performances metrics and compare that metrics with the metrics that had already present before the restructuring the sector. The best way to identify the changes in the performance over the time using time series data and to improve the attributes or disturbing performances before and after reforms are implemented to reform themselves [12]. This way of analysis is based on explicit or implicit assumptions about various outside variables affect the performance and how can they have been controlled adequately. The other way of approach is comparision of performance among various countries or states which implemented reforms with the countries that are following traditional system and regulatory arrangements. This way of approach is used frequently in US for comparision of various performances of different regulatory and deregulatory policy initiatives in different states which are implemented particular policies in different ways.

Another model to analyze the regulatory reforms and deregulatory initiative is a structural simulation model. This model explains what are the important performance metrics had done till date. After that the analyzed performance is compared with actual performance. All these approaches can show an useful aspects of policy reforms on various performance Indica. Metric assessments must show that the observed performance should be compared with performance under set of arrangements [15]. In this there are significant benefits which also holds some risk significant costs whether these assessments are implemented successfully or unsuccessfully. The expectations after the assessments should be refined by using the information that is available during the assessments.

Billions of cash are spent on software projects improvements because of their significance to departments, offices, and administrators as announced by Gorla and Lin in 2010. software development projects are mind boggling to alleviate risks and excessively numerous of them end in disappointment.

VIII. RISK MANAGEMENT FOR METRICS

Risk can be defined as imperfect knowledge where each action leads to a set of possible outcomes each with an unknown probability. There are many risk analysis techniques currently in use to evaluate and estimate software risks but it is very important to choose appropriate model to reduce software risks .In the Concurrent Simultaneous Engineering Resource View (CONSERV) was created for a keen learning in view of project management techniques and can be likewise utilized as risk management system.

By improved quality of software projects of taking an organization while estimating the quality–affecting risks in IT software projects. The outcomes demonstrate that there were 40 common risks in software projects of IT organizations in Palestine. The amount of technical and non-technical troubles was vast. [16]An optimal control problem without constraint solution will leads to a large scale block linear system. A two stage method which is well suited to an implementation with flexible communication. The need for optimal control of continuous complex dynamic process subject to some disturbance can require the solution for large scale system. Asynchronous iterations with flexible communications are tightly bound to the concept of submappings and supermappings which is associated to the generation of monotone sequence of vectors. This implementation is done in two types of architecture:

1. A super computer with distributed memory CRAY T3E by using message passing libraries such as MPI or SHMEM
2. A shared memory symmetric multiprocessor(SMP) by using a POSIX thread library and a network of SPM by using message-passing and POSIX thread libraries.

Software systems are becoming up plainly more open, appropriated, unavoidable, and connected. In such systems, the connections between inexactly coupled application components progress toward becoming non-deterministic. Coordination can be seen as a method for making such approximately coupled systems more adaptable. In this paper they demonstrate how coordination-systems, which are comparable to sensory systems, can be defined autonomously from the practical systems they manage. Such coordination-systems are a network of coordinators and contracts.

One essential way to deal with building runtime adaptive systems is to develop them of approximately coupled components. These components are progressively planned and reconfigured to meet natural requests or evolving objectives. This basic approach is normal to autonomic computing, agent systems and element models. This paper demonstrate coordination capacities can be actualized as a different subsystem that oversees heterogeneous components.

Electronic markets produce significant new exchanging openings and grow the open door for the dynamic evaluating of merchandise, and prompt enhanced market efficiency in numerous settings. Electronic markets find application not just for individual to-individual exchanges, additionally progressively for business-to-purchaser closeouts, for example [13], pitching surplus stock and business-to-business sourcing occasions. One major impediment to the adoption of mechanized exchanging operators is that clients frequently have insufficient trust of software agents that would chip away at their benefit. One critical, if obvious, angle that appears to be essential in the reception of computerized specialists is that these operators maintain a strategic distance from, and be believed to abstain from, committing errors.

Traceability is the capacity to describe and take after the life of a software artifact and a methods for displaying the relations between software artifacts in an express way. Traceability has been effectively connected in numerous software engineering groups and has as of late embraced to archive the move among necessities, design and execution. Traceability support is required whenever large and complex software systems are to be maintained[14]. On the other hand, traceability is very expensive, both to reconstruct and to maintain. The trade-off between necessity and cost is not resolved yet. Nonetheless, this is possible only partially and in very specific contexts. What and how to automate depends very much on the domain, the current practice within the developing company , and the knowledge and experience of developers.

IX. CONCLUSION

To conclude our survey, we return to the problem that how to get a suitable set of metrics to reduce the impediments in the software architecture and find out errors. But it is really not possible to find out all the errors in the program. Thus, the major question arises, which type of metrics we would adopt to reduce impediments. For this purpose, we have taken and analyzed number of metrics. Finally, the results of the analysis have been presented. The major conclusions are that, our current testing technique knowledge is very limited and is based on impressions and perceptions.

X. REFERENCES

[1] Using role-based coordination to achieve software adaptability ,Alan Colman_Jun Han, Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, Victoria, Australia 2005

[2] Introduction to Electricity Sector Liberalization, PAUL L. JOSKOW, Department of Economics, Centre for Energy and Environmental Policy Research, Massachussetts Institute of Technology, Cambridge, USA 2005

[3] Information technology and systems justification , A. Gunasekaran , E. W. T. Ngai and R.E. McGaughey 2007

[4] A systems engineering perspective on the human-centered design of health information systems,George M. Samarasa,*, Richard L. Horstb 2006

[5] A systems engineering perspective on the human-centered design of health information systems, George M. Samaras, Richard L. Horst

[6] Beesley, M. and Littlechild, S. (1989). The regulation of privatized monopolies in the United Kingdom. Rand Journal of Economics

[7] L. Andrade, J.L. Fiadeiro, J. Gouveia, G. Koutsoukos, A. Lopes, M. Wermelinger, Patterns for coordination, in: Coordination'00, in: LNCS, vol. 1906, 2000.

[8] Notkin, D., Griswold, W.G. "Extension and Software Development", *Proceedings of the 10th International Conference on Software Engineering*, April 1988

[9] Beck, K., 1999.Embracing change with extreme programming.*IEEE Computer,* 32(10), 70-77

[10] Metrics-based Evaluation of Object-Oriented Software Development Methods, *Reiner R. Dumke, Erik Foltin* 1996

[11] Software Architecture Adaptability: An NFR Approach ,Nary Subramanian Lawrence Chung

[12] Quantitative and Intelligent Risk Models in Risk Management for Constructing Software Development Projects: A Review Abdelrafe Elzamly1* and Burairah Hussin2

[13] nvestigating Early Web Size Measures for Web Cost Estimation: Emilia Mendes, Nile Mosley, Steve Counsell

[14] A family of experiments to validate metrics for software process models: G. Canfora a, F. Garcı´a b,*, M. Piattini b, F. Ruiz b, C.A. Visaggio a 2004

[15] A Model Driven Approach to the Analysis of Quality Scenarios within Self-Adaptable SOA Systems **:**Boris Perez1 ,Dario Correal2

[16] J. Miler, "A Method of Software Project Risk Identification and Analysis," Gdansk University of Technology, **(2005).**