THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

Metric Dashboard Enhancements for Self Optimizing Networks

Hari Krishna B.
M. Tech. Student, Software Enginering, Department of ISE, RVCE Bangalore, India
Dr. Jitendranath Mungara
Profesor & Dean PG Studies, Dept of ISE, RVCE Bangalore, India
Priya D.
Assistant Professor, Department of ISE, RVCE Bangalore, India

Abstract:
This paper Presents the study of how the metric dashboard may be used in the corporate world for viewing the goals of the teams as well as to full fill the customer requirements and deliver the products in time and provides one point access to all stakeholders in viewing important metrics required to manage the SON(Self Optimizing Network) Program releases and SON R&D(Self Optimizing Network Research & Development) Efficiency .Basically the Metric Based Dashboards Provide Valuable Information and insights for managers as well as the to the team members in the software development organizations. However launching and implementing a company wide metrics program is very hard and time consuming. This helps to built better and more sustainable metrics and infrastructures. This paper also describes the experience with the development of products at corporate level and reviews the current status of the Product, Release, Value Stream and Team levels.

Keywords: *Metric dashboard, value stream, SON, NFR*

1. Introduction

Metric dashboard for SON provides one point access to all stakeholders in viewing important metrics required to manage the Network Program releases and SON R&D Efficiency. Current metrics monitored in the apparatus are, User story and Cycle Time of feature, Feature Readiness for a specific release and feature throughput, Cycle time of the Features in Focal point. views of the above are available at Product, Release, Value Stream as well as the Team levels. Current Metric Dashboard Interface is to get data from Accept360 and Focal point tool.

Teams can use Cycle time metric trend to check the speed of their deliveries. Value stream leaders and Team Leaders can check the average speed of deliveries at Value stream level and Team level. Value stream leaders can help teams and Value streams to improve the speed of deliveries of the product where it needs. Product and Program manager can view Feature readiness for a program release. Teams can check user story or feature throughput trend for all releases or for a specific release. Value Stream Leaders can check the user story and feature throughput trend at Value Stream Level and also Features completion status for specific release. Program manager can check the status of the feature deliveries.

Agile development is a method of building software by empowering and trusting the people, acknowledges the change as norm, and promoting constant feedback, and it is based on a set of values and principles that are acknowledged as truly important. Different technical, organizational, and management practices supports implementing these values and principles.

With Metric Dashboard any problems become visible for the teams, program to take required action. Some of the Enhancements proposed are following – NFR (Non functional requirements) to be included and made visible in the statistics, User story burn down per releases.

Metric Dashboard to be enhanced with the interface to other tool for example pronto, for displaying fault management statistics and also to provide additional metrics required to monitor efficiency at required levels.

2. Related work and Background

Metric Dashboard aims to help in understanding and visualizing [10] the tasks which are assigned to the value streams. Several tools have been proposed to analyze, understand and visualize [11] the tasks of the teams and their deliveries of the product. The goal of the dashboard is similar in sprint, namely to provide a web based dashboard for analyzing and understanding the User story and Cycle Time of feature, Feature Readiness .

Metric Dashboard is a web application that provides multiple views of a large dataset. Dashboards are customized by developers for specific business intelligence with in organizations [8] and helps in viewing the completion of task between the remote teams [9]. In this system the user connects visualizations with onclick gesture and then specifies relationships between the visualizations in a popup dialog.

It helps in monitoring the work flow's and workflow representation and the real time status information of the workflow. They have been widely applied in related areas like user interface design and reverse engineering [5][6] and software reuse[3][4]. Business process management enables the management and analysis of operational business process [7]. There is also much interest around the concept of dashboards with several companies providing relevant solutions such as IBM [12] and Hyperion [13].

Entities such as input events, metrics, outbound events and actions can be monitored and scheduled. Based on this situations may involve generating alerts or displaying statistics and aggregated information on to the dashboard. This approach utilizes XML schemas for showing the visibility and the visibility between the teams.

3. Motivation

With the proposal and development of Metric Dashboard we plan to automate this task. The program manager, R&D Leaders and Teams are able to have one point access to these metrics via web link which help them to visualize their progress and take required actions to continuously improve their deliveries. Now in the accept 360 the new user stories are being updated manually. We are going to introduce a Scheduler job called Quartz job which can update the user stories for every 3 hours from the accept and put it up in to the tool as in the state of first come first serve and it can delete the done user stories automatically.

4. Design and Architecture

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the one in moving from the problem domain to the solution domain. The phase of the system is the most critical factor affecting the quality of the software. Here we build the system Architecture that will be helpful to understand the behavior of the system. Here we divide problem into modules. Software sometimes can be a complex entity. Its development usually follows what is known as Software Development Life Cycle (SDLC). Design is one of the initial stage in Software Development Life-cycle (SDLC) has four stages: Requirements, Design, Coding and Testing. The objective of the design stage is to produce the overall design of the software. In Design Technical Architect of the project will study the proposed applications functional and non-functional (qualitative) requirements and design overall solution architecture of the application which can handle those needs.

Figure 1. Describes the overall System Architecture of the Metric Dashboard tool consists of three Different tires Namely they are Presentation Layer, Service Layer , Data access layer. In each of these layer may consists of different modules namely they are JSON with Jersy Rest API and Bootstrap, JSTree, High charts, JQuery.

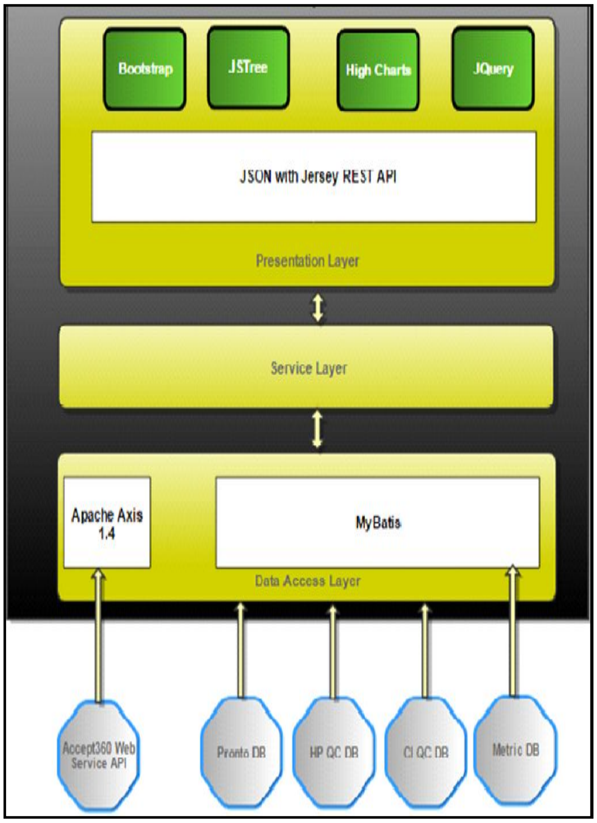

Figure 1: Overall Architecture and Design of the Metric Dashboard Tool

The data access layer may consists of My Batis, Apache Axis 1.4 the ACCEPT 360 Web Service may be connected to apache axis and to My Batis provides the connection to Pronto DB, MetricDB, CIQCDB, HPQCDB. These are the modules of these three tires.

5. Interface Design

Figure 2 shows the overall interface Design of the metric dashboard. This dashboard consists of different value streams so that better visualization of each team can take place without any ambiguity.

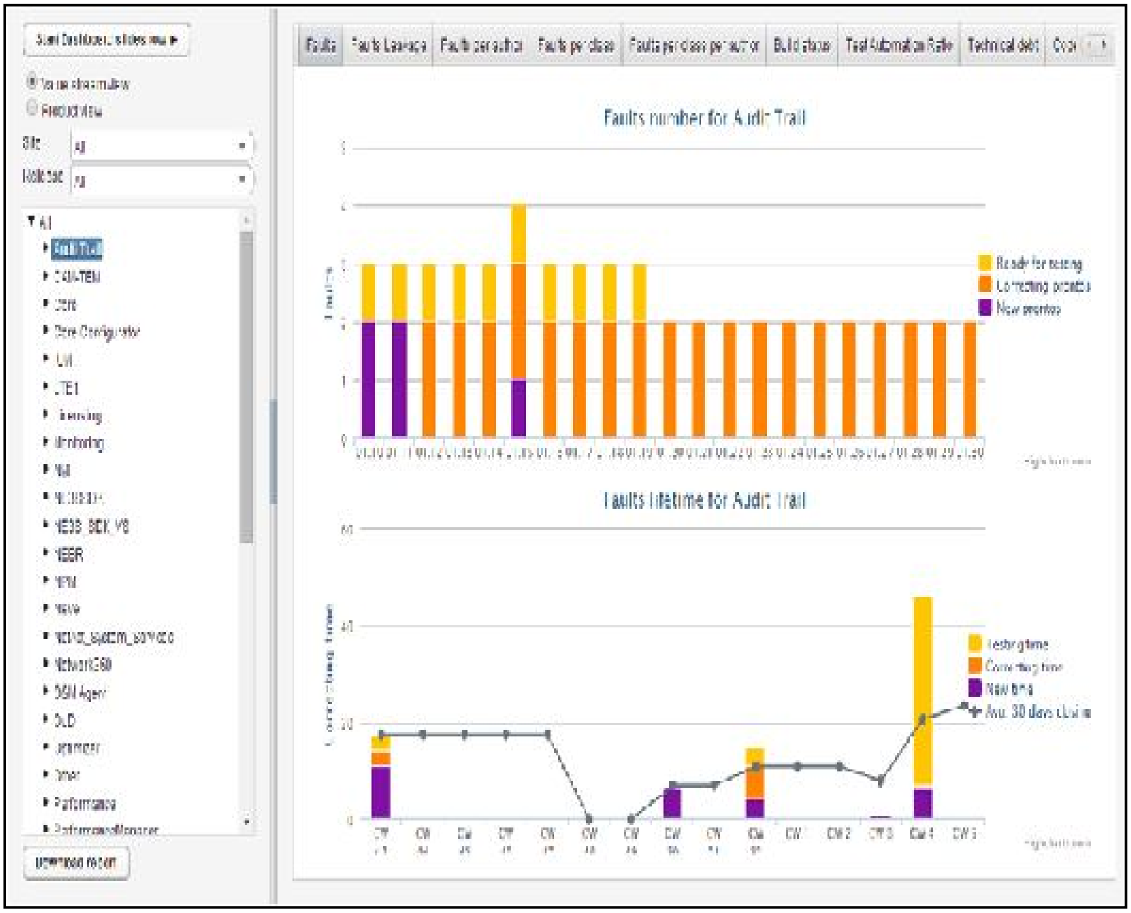

Figure 2: Web-based dashboard view of the Metric tool

6. Implementation

Metric Dashboard is implemented as a set of high charts that give a visual, real-time representation of important aggregated task related to several value streams has been well studied [1][2]. The metric dashboard project leveraged the abundance of teams information stored in the centre's MongoDB online-accessible database. This document database is the back end data repository of teams as well as the product mangers and project leaders. It main purpose is to drive a suite of web charts that teams and team members ,managers utilize and update their daily tasks in scrum.

The Metric dashboard implementation process can be broken down into four steps:

User story => Query => Feature Throughput => Feature Cycle Time => Feature Readiness for a release => Cycle time of Features in Focal point => value stream.

- User Story: The userstory is a refined requirement of SON Manager product . A feature has many user stories.
- Feature Throughput: Number of Features done per month is known as Feature Throughput.
- Feature Readiness: Number of user stories done in the feature is called Feature Readiness.
- Feature Cycle Time: Time taken for a Feature form creation state to done. Feature state – Created to reaching FS4.
- Focal point: Focal Point is where all the features of SON Manager are stored.
- Value Stream: Entity or an development area which develops features for the SON Manager product , thereby providing value. Here Self Optimization, Self Configuration, Self Healing are the value streams.

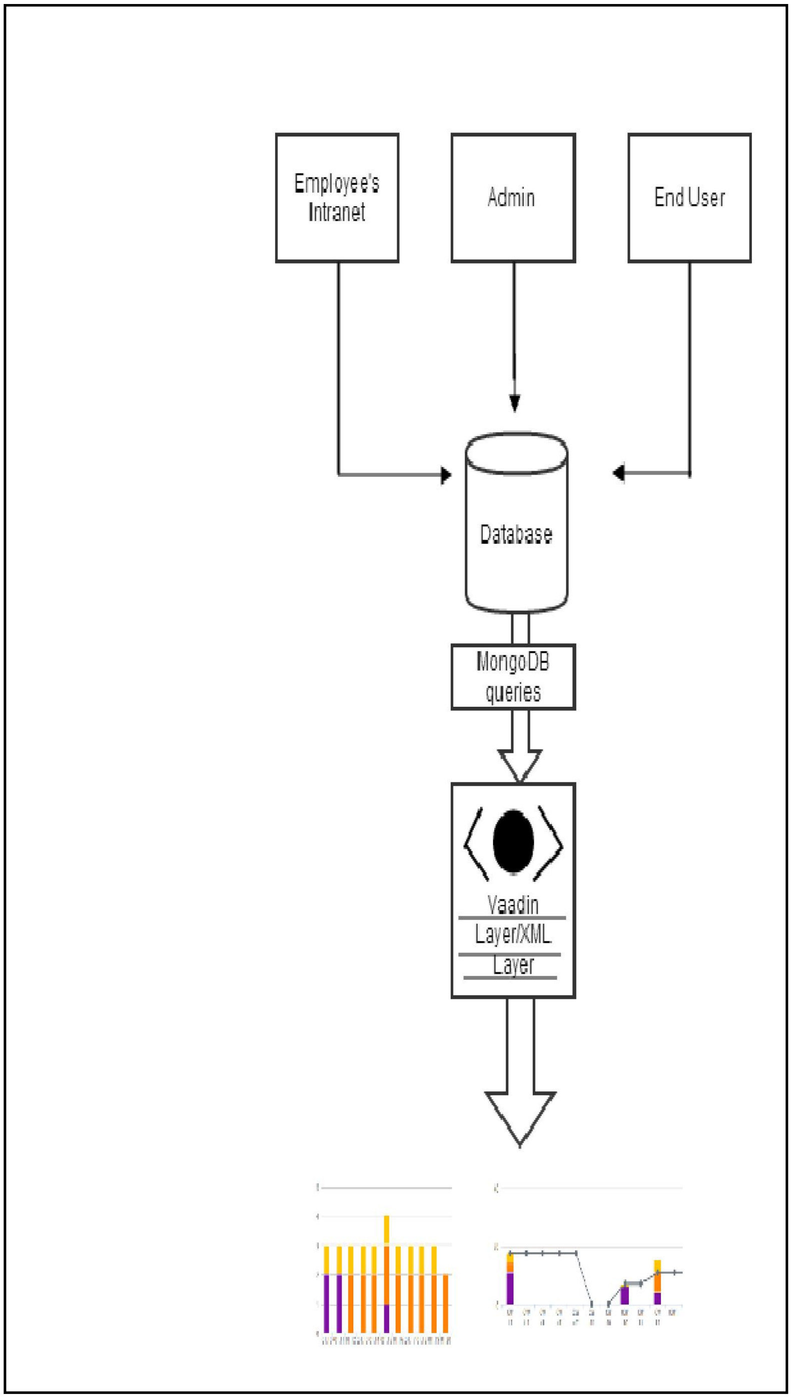

Figure 3: Architecture status of teams, including scripted updates of Dashboard with High Charts

7. Objective

The goal is to improve efficiency in SON R&D to deliver features in next releases on time and with required quality. With the automation of the metrics more visibility is brought in the progress of R&D deliveries on daily bases.

8. Problem statement

Currently required metrics need to be generated manually from the different tools and visibility on the R&D progress is less. To generate these metrics it requires effort and time. There was need to automate here.

9. Conclusion

In this paper, we have described the design and the deployment of metric dashboard that enables the managers and the team leaders as well as the value stream leaders can have the visualization of User Story, Feature Throughput, Feature Readiness, and Feature Cycle Time. The Experience at metric dashboard can be readily implemented based on data collected for required purposes of the value streams. At metric dashboard the representations changed over time as particular views were found to be more or less useful. Our users have successfully built several multi visualization tools that integrate with both static and live data. Our goal was to assemble visualization, construction, knowledge organization and multi visualization layout into a single application that is both well integrated with general applications and open to use in a variety of contexts. Based on our observations and the feedback we plan to automate this task.

10. References

1. Performance Dashboards, W.W. Eckerson, John Wiley and Sons, 2006.
2. Information Dashboard Design, S. Few, O'Reilly 2006.
3. W. B. Frakes and K. Kang, "Software Reuse Research:Status and Future," IEEE TSE 31, No. 7, 2005.
4. J. Greenfield, K. Short, S. Cook, and S. Kent, "Software Factories Assembling Applications with Patterns, Models, Frameworks and Tools," 18th Annual ACM OOPSLA, 2003.
5. S. Rugaber and K. Stirewalt, "Model-Driven ReverseEngineering," IEEE Software 21, No. 4, 2004.
6. Y. Yu, Y. Wang, J. Mylopoulos, S. Liaskos, A. Lapouchnian, J.Cesar Sampaio do Prado Leite: ReverseEngineering Goal Models from Legacy Code. ICRE,2005.
7. W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business Process Management: A Survey," BPM, Eindhoven, The Netherlands, 2003.
8. "SAP - Xcelsius 2008: Dashboards and Visualization for Better Decision Making." http://www.sap.com/solutions/sapbusinessobjects/sme/xcelsius/.
9. J.T. Biehl, M. Czerwinski, G. Smith, and G.G. Robertson, "FASTDash: a Visual Dashboard for Fostering Awareness in Software Teams," Proceedings of the SIGCHI conference on Human factors in computing systems, San Jose, California, USA: ACM, 2007, pp. 1313-1322.
10. D. Messerschmitt and C. Szyperski, Software ecosystem: Understanding and indispensable technology and industry. MIT Press, 2003.
11. S. Neu, M. Lanza, L. Hattori, and M. D'Ambros, "Telli ng stories about GNOME with Complicity," in Working Conf. Software Visualisation (VISSOFT). IEEE, 2011, pp. 1–8.
12. AlphaBlox. http://www.alphablox.com/
13. Hyperion. http://www.hyperion.com/

.