

**Create a maven archetype for creating GWT based XML
configuration management and implement JMX, syslog and WMI**

W.Chinthaka Prasanna Senanayaka

15/1A, Perera Mw, Meethotamulla,
Wellampitiya, Colombo, Sri Lanka.
Tel. No. +94718443972

E-mail:

senanayakachinthaka@gmail.com /
wcp.senanayaka@ymail.com

Personal Details

Name: Wahalatantrige Chinthaka Prasanna Senanayaka (W.C.P. Senanayaka)

University: University of Kelaniya, Sri Lanka.

Course: B.Sc. in Management and Information Technology (3rd year, Special)

Email address: senanayakachinthaka@gmail.com

Physical address: 15/1 A, Perera Mw, Meethotamulla, Wellampitiya, Colombo, Sri Lanka.

Phone numbers: +94718443972

Web page: <http://chinthakarobotikka.blogspot.com/>

IM contacts: Skype: chinthakas87

IRC nick: chinthakas

Availability:

1. I can spend on GSoC OpenNMS project, 5–7 hours per day practically.
2. As GSoC time plan I can start and end the project, and I will continue the contact with OpenNMS projects.
3. Other factors affects my availability are university examinations (I have examinations on 2011/04/23, 2011/04/30, 2011/05/02. And before the next semester exams come, GSoC will be finished).

Starting date delays: no delays.

Background Information

Education background: Did Mathematics, Physics for Advance Level examination. Then I entered to University and now I am in 3rd year special batch.

Software development background:

I have worked at H'Senid Software International Pvt Ltd (mobile software development, as an undergraduate trainee software engineer).

Computer languages: C, JAVA, VB (VB.NET and VB script), PHP (HTML, Javascript), ASP.NET (average), and Scala (average)

Other Java compliant tools: Spring, Hibernate, Vaadin

Software: Linux – Ubuntu, MySQL, Flash, SQL SERVER

Software design using UML (just started learning BPMN)

Other: Maven, special interest in open source

Why I am interested in OpenNMS:

1. It is open source.
2. I have an average knowledge on networking and network management. I am very much interested in network technologies. And it feels challenging to learn networking and network management.
3. OpenNMS is using Java and Java compliant technologies (Hibernate, Spring & SpringMVC, etc) to build the software that I know much. And then I can easily get adapted to their projects.
4. Friendly mentoring and develop with fun.

Why I am interested in FOSS:

1. Most of the FOSS software is freely available for all (no license fees).
2. FOSS Oses have fewer viruses.
3. Platform independency (like Java).
4. Community base and open to participate in others to development and non-development work to get the experience and confidence in the field.
5. They are good for small and medium scale enterprises.

What is my experience with FOSS?

1. For everyday use, I use Ubuntu. Other than Ms Office.
2. For self-learning and for University work, I use FOSS, because they work efficient.
3. At university training period I did a lot of projects using FOSS technologies.

How do I define FOSS?

Open source is, software source code is open to everyone and can be edited and used by anyone. That will give the software to flexibility to use the software in a wide range of purposes, and for learners can easily get into the development of software because of open source code. Most of the FOSS available with no license fee, and FOSS community is more focused on provide software as a service not only as a product.

My previous experience:

1. I went through a training program as an undergraduate trainee software engineer.
2. I have done some software projects for my graduate course.
3. I have participated in open source community work, such as distributing FOSS in University exhibition, and Open Office translating marathon event (Open Office help documentation translation English to Sinhala).

One fun fact about myself:

Love to spend time with friends (and girls), have a chat with them and stay free.

Rating scale:

Technology	Rating
Java	8
Distributed Version Control System	6
Eclipse	7
GWT	5
Hibernate	7
REST	6
Apache Jersey	5
Spring	8
OpenNMS	5
JMX	6
SNMP	4
PostgreSQL	5
javascript	7
Maven	7
Linux	8
Mac OS	4
Windows	8
Junit	8
IPv6	7
JNA	3
Docbook	3
TCP/IP Network stack	8

Proposal Requirements

Overview:

“Create a maven archetype for creating GWT based XML configuration management and implement three of them” project is to create Maven project to access XML configuration files to system users through their web user interface. There are thirty odd XML based files to keep the configuration information about management of the system. This project must handle multiple accessing to same XML configuration file, and scope of the project will include implementing only three (JMX, syslog and WMI) out of thirty odd XML configuration files. These XML configuration files include syslog event mapping into OpenNMS system event, JMX, and WMI etc. These XML configuration files will hold information about what nodes in OpenNMS should perform the collection on, what security should it use, what info will be collected, etc. GWT will be used to develop web user interface. Throughout the project there will be proper unit tests, integration tests, documentations, manuals, etc. And in designing the project I will use design patterns appropriately. And there will be new user friendly, user controls on web user interface of the OpenNMS in order to configure those system statuses by the user.

Deliverables:

Final output of the project:

1. Design documents (Milestone1).
2. New archetype for handle user configurations, and access controlling for those (Milestone2).
3. New web user interface to configure XML based configurations (Milestone3).
4. Unit testing, integration testing and documentations for testings (sometimes performance tests as well) (Milestone4).

5. JavaDocs for each of those interfaces of the XML configurations and other development related help documentations (Milestone5).
6. Some additions to help documents of web user interface about XML configurations (Milestone6).
7. Other GSoC related reports, coding, and evaluations.

Schedule:

April 26 – May 9 – Get adapted to OpenNMS organization, go through documentation, and get the deep level understanding. Get through high self learning and growth about OpenNMS and their technologies.

May 10 – May 14 - Initial design documents preparation.

May 15 – May 22 - Detail design documents preparation, error corrections, self learning and growth (Milestone1 – design documents).

May 23 – May 25 – Setting up initial archetype on OpenNMS JIT and integration GWT, and etc into the archetype and beginning of running archetype without having coding

May 26 – June 26 – Coding begins (Milestone2 – core functionality implementation, and unit tests for core functionality).

June 27 – June 30 – Design review and code review. And at the same time coding goes parallel.

July 1 – July 10 – Coding (Milestone5 – JavaDocs preparation, and other documentations).

July 11 – July 15 – Midterm evaluation reports preparation, submission.

July 16 - Possible project schedule up gradation.

July 17 – July 31 – Coding, testing, document preparation (Milestone3 – web user interface, and integration with core functionality).

August 1 – August 3 - Design review and code review. And at the same time coding goes parallel.

August 4 – August 15 – Coding (Milestone4 – unit tests, integration tests, and documents preparation for those tests, and Milestone6 – user interface help documents preparation).

August 16 – August 21 – Finalize the project source code, documentations. And stop the project.

August 22 – August 26 – Project final evaluation.

August 30 – September 2 – submit code samples to Google.

(Consideration: One coding part is considered as finished when design, coding, testing and docs are finished).

Value to the community:

1. First, through the project a value addition to users for those who use OpenNMS.
2. Can have wide spread community through having projects like these.
3. From this project, impact on the users of using OpenNMS is ease of configuring the OpenNMS system, and users can define what kind of information on what kind of features, OpenNMS system must collect and keep.
4. High usability through self descriptive and easily understandable user interface and help documents.
5. Can be extended this feature in OpenNMS easily, with the project documentations prepared with this project.
6. Can provide efficiently and effectively running features to the users of OpenNMS with the test driven development.