Empirical Case Study of Measuring Productivity of Programming Language Ruby and Ruby on Rails

Tetsuo NDOA
Information-processing Center
Shimane University
Matsue, Japan
nodat@soc.shimane-u.ac.jp

Chi JIA
Faculty of Law and Literature
Shimane University
Matsue, Japan
jiachi@soc.shimane-u.ac.jp

***Abstract*— This short paper is intended as a trial balloon of the evaluation of open source software, by measuring the productivity of programming language Ruby and web application framework Ruby on Rails, compared with other open source software.**

Keywords-component; Open Source; Ruby; Ruby on Rails; Productivity; Programming Languag

I. INTRODUCTION

Ruby is the Object-oriented Script Language released by Mr. Yukihiro Matsumoto, called "Matz" in open source communities, and was opened to the public in 1993. Matz lives in Matsue City in Japan and has been developing Ruby with many open source developers all over the world through the Internet. The number of core committers of Ruby is about seventy in 2011, and the two-third of them are Japanese. So Ruby is one of very few open source projects that Japanese engineers are mainly engaged in developing.

At first, though Ruby commanded attention through geeks, it had not been spread in business uses. But, in 2005, David Heinemeier Hansson – a programmer in Denmark, released Ruby on Rails, web application framework constructed by Ruby. Hence, Ruby came to attract attention and to be used also in enterprise areas. According to TIOBE Programming Community Index, which announces the ranking based on the retrieval by keyword of the search engine, the share of Java is 18.5% in the investigation of in 2010, and PHP is confronted to 7.8%, and Ruby is at level of 1.9% (ranking 10th place) [1]. But the number of Ruby's engineers has been increasing remarkably. It is forecast that the engineer who will use Ruby by 2013 reaches four million people according to the investigation of United States research company Gartner [2].

Then IPA (Information-technologies Processing Agency) [3], the Japanese government agency, started to support the Ruby project. It has been driving forward the standardization of Ruby. Because Ruby is open source, there are many implementations of Ruby. Besides the Ruby 1.8 affiliate (implemented by C language) and this Ruby 1.9 affiliate (implemented by virtual machine YARV), IronRuby (implemented to operate Ruby on .NET Framework), MacRuby(implemented to operate Ruby on Mac OS X), and Rubinius (bytecode interpreter on a virtual machine), etc. Thus there are variety of implementations. But the standard specifications of Ruby language had not existed. So, IPA started standard specifications making, first domestically based on standard specifications in 2008, and constituted it as JIS (Japanese Industrial Standards) in 2011. Now, Ruby is proposed to ISO (International Standard).

This process will improve the interconnectivity of the portability and external systems by making it in accordance with this standard. Moreover, it will develop the foothold of the specification when the server environments to execute the program written with Ruby. The reason why IPA, the Japanese government agency, is bringing forward this process is to increase the market of Ruby, Japanese-Oriented Programming Language, in enterprise areas.

However, the reason why Ruby and Ruby on Rails have been used recently is the productivity of them. The productivity of Ruby and Ruby on Rails has been said tendentiously as such "The productivity of Ruby is ten times higher than that of Java". But we must proof the productivity of them empirically and scientifically, if Ruby and Ruby on Rails are good for enterprise areas. So, we measured the productivity of them compared with other script languages. In this context, the term of "productivity" means software productivity by man-hours, including experience years of using language. In this paper we mention the method and the result of the productivity's comparison, and we also hope this method will be an "active pointer" of measuring software's productivity.

II. PRODUCTIVITY OF RUBY

The script languages like Ruby have to be compiled the source codes to object codes at each execution, so that the processing speed of them tends to slow extremely. The simpler the characteristic of language is, the slower the processing speed of it becomes. However, due to faster grow of the information processing abilities symbolized in the Moore's Law, the improvement of the processing speed has become to be owed to computer hardware, mainly the power of CPU. And as for the service of Web, prompt (agile) and flexible development and release is required.

The amount of the description of codes by Ruby is less than that of other programming languages, and the grammar expresses man's imagination similarly near human language, so that the its productivity of development becomes higher as a result. Therefore, the productivity of Ruby is evaluated in

 ISBN: 978-1-61208-165-6

the development of the Web application from which quick release and a frequent change are required. If the domination of such productivity is actually proven, introducing Ruby into the production site of development will be able to not only raise the productivity of development, but also decrease the stress of engineers. So we measured the productivity of Ruby compared with other programming languages.

We compare the productivity among Ruby, Java, and Perl cooperated with an IT company [4]. We developed the Web applications that have the same functions (Message board systems that have functions of comments contribution, multiple contribution prevention, indispensable check, and the automatic deletion) by Ruby, Java and Perl, daringly without using web application frameworks. TABLE I is the result of the comparison of each programming language's productivity.

TABLE I. COMPARISON OF PRODUCTIVITY (2010)

Languages	Java	Ruby	Perl
Lines	177	46	42
Man-Hour (Coding and Test)	Coding : 8 hours Test: 1 hours	Coding and Test: 2 hours	Coding and Test: 0.75 hours
Require Modules	19	2	4 (uses)
Operating Condition	Servlet	Http Server	Http Server
Operating Checking Server	Tomcat	Apache Anhttp	Apache Anhttp
Experience Years of Using Language	7 years	0 years	5 years
Experience Years of Development	7 years	7 years	7 years

As a result, Ruby exceeded both amounts of the codes and the manufacturing time greatly compared with Java in productivity. Ruby was proven to be as several times productive as Java at the manufacturing time (4.5 times if it simply compare). But it was proved that Ruby is not more productive than Perl. However, in spite of the first manufacturing in Ruby engineers could write the same amount of codes that Perl engineers, who need five years' experiences, write. Though, the speed manufacturing time of Ruby engineers is slower than that of Perl engineers, if they are trained coding, the productivity will be expected much higher.

III. PRODUCTIVITY OF RUBY ON RAILS

As has been previously described, Ruby became to attract attention since the release of Ruby on Rails. So we continuously measured the productivity of Ruby on Rails compared to Java's developing framework cooperated with an IT company [5]. We tried combustion and additional function requirement of 70% of the Working management system by Ruby on Rails. The system had previously (in 2007) developed by Java and JBoss Seam, web application framework.

To compare the amount of source codes, we divided the system into three elements, Model which is the kernel of processing of the software design, View which rules display and output, and Controller: which receives input and controls View and Model according to the content And we compared each number of steps for these three elements. TABLE II is the result of the comparison.

TABLE II. TABLE TYPE STYLES (2010)

Elements	Ruby on Rails	Java + Jboss Seam	Java + Jboss Seam /0.7
Controller	5.1K	18.4K	26.3K
Model	1.2K	12.6K	38.1K
View	4.2K	4K	5.7K
Totla	10.5K	35K	50K

If we simply compare the numbers of steps, the productivity of Ruby on Rails is three times of that of Java and its web application framework. Moreover, if we consider that the development by Ruby on Rails was 70% of that by Java, the productivity is five times of Java. And, by the function point method (FP/man-hour comparison), which is an unit of measurement to express the amount of business functionality an information system provides to a user. The cost (in dollars or hours) of a single unit is calculated from past projects, the productivity of Ruby on Rails is 1.4 times of that of Java.

IV. ISUUES AND FORESIGHT

Though the productivity of Ruby and Ruby on Rails was measured by these case studies, the number of cases is obviously few. So we must continue to study much more cases. And, it will be difficult to compare productivity under the same developing condition. However, as the method of comparison of productivity was put on a firm footing, continuance of this study will enable us to compare the productivity of software empirically and scientifically.

Moreover, the performance of software must be measured by considering the effective speed in processing. For this reason, IPA is driving forward the standardization of Ruby. At the same time, in this study we compared the productivity in developing. Then we must measure and evaluate the performance of software totally. This process will be conducible to the evaluation of open source software which does not receive baptism of market pricing.

[1] TIOBE Programming Community Index
http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

[2] Ruby is Fastest-Growing Web Development Language: Gartner
http://www.pdfzone.com/cp/bio/Darryl-K.-Taft/

[3] IPA (Information-technologies Processing Agency)
http://www.ipa.go.jp/index-e.html

[4] Central Information Coorporation in Hiroshima
http://www.cis-net.co.jp/outline.html

[5] TOSCO Coorporation in Okayama
http://www.tosco.co.jp/

 ISBN: 978-1-61208-165-6