International Journal of Information Technology and Knowledge Management
January-June 2011, Volume 4, No. 1, pp. 101-104

METRICS, TOOLS AND SCIENTIFIC APPLICATIONS: REVOLUTIONIZING ENGINEERING DESIGN

Anisha Gupta[1] & Deepshikha Jamwal[2]

As the software industry has matured, the implementation of a suitable metrics program, software tools and scientific software skills has become significant in almost all organizations. The paper discusses the experience of some organizations that have developed and encouraged the use of metrics. It gives an overview of how these industries have incorporated software metrics in almost all stages of software development. It then discusses the benefits of application of various software tools for various development stages followed by a brief description of some of scientific software applications including Matlab, Mathematica and Maple. This paper stresses on the importance of using all: "metrics, software tools and scientific software applications" for revolutionizing software development and engineering design capability to very high levels of reliability and accuracy. Lastly, certain recommendations have been made regarding the usage of all these three applications.

Keywords: Software Metrics, Software Tools and Scientific Software Applications

1. Introduction

According to IEEE "standard of software Quality Metrics Methodology" Software Metrics is a function with input as software data and output is a value which could decide on how the given attribute affect the software.

Software Development Tool is a program or application that software developers use to create, debug, maintain, or otherwise support other programs and applications.

Scientific Software applications, which have become a fundamental tool for sophisticated designers in industry, allow users to perform numeric and symbolic computations at very high levels of accuracy. They have given educators the ability to convey advanced mathematical and engineering concepts in new ways and spend more time on analysis of engineering systems and less time on remedial mathematics [1].

All these applications can be used together in the development of software to remove existing Software Crisis. Section 2, 3 and 4 of the paper discuss the experiences of industries in implementation of metrics, tools and scientific packages respectively. Section 5 lists certain improvement areas which if taken care of can lead to better application and result.

2. Metrics in Software Industry

Software measurement started in early 1970s in the IT Industry. Organizations such as NASA, Sapient, Infosys, HP, Motorola and many others make extensive use of software metrics. To present the current usage scenario of metrics in Industry, examples from Sapient and Infosys are presented in the following section to highlight the implementation of metrics in all stages of software development.

2.1. Sapient

As one of the world's leading global service firms, Sapient is renowned for its ability to drive development projects quickly and seamlessly because of its unique culture, methodology and collaborative tools. Sapient believes that metrics are critical to the success of any project. Some of the development stages and metrics used in them are:

Fusion: It refers to broadly discussing the features of application. In this phase, scope matrix is created which makes use of metrics like associating business value and risk complexity for each of the high level feature. Business value refers to the importance of feature for the product and risk complexity is the degree of loss that would be incurred if the feature is not implemented.

Estimation Phase: Each functionality(feature) is assigned some complexity level based on technology. For most of the projects there are mainly 5 levels of complexities: very low, low, medium, high and very high. Days of completion are associated with each complexity level. These are known as story metrics. Story metrics are further divided into task metrics which give hourly measurement of progress achieved. Thus task metrics enable the technical and management teams to track any delay on an hourly basis rather than waiting for the whole story to get completed.

Implementation: Talent indicators like efficiency, quality, attendance etc and time metrics are considered critical in understanding workforce effectiveness and performance. Using top performer data as a starting point,

[1]Computer Science Department, Bharati Vidyapeeth College of Engineering, India
[2]Department Computer Science & IT, University of Jammu, India
Email: [1]anishagupta2@gmail.com, [2]jamwal.shivani@gmail.com

utilities can baseline the average time required to execute critical orders, compare workers within a specific territory, make cross–territory assessments and ultimately even compare utilities [2].For tracking these metrics Sapient makes use of Result Space Tool.

Requirement Traceability metric gives complete information about each functionality including its implementation and test scripts. The closure for this metric is usually done by the business analysts associated with the project.

2.2. Infosys

For process definition and improvement, Infosys first adopted the ISO 9000 framework and got ISO certification in 1993. To further improve the software process, it adopted the CMM framework. It was first assessed at level 4 in 1997, and then at level 5 in 1999.

Examples of most commonly used metrics for various activities at Infosys are as follows:

Reuse has been a major focus area for improving productivity of many software organizations and enhancing the quality of software products delivered. Metrics used to access the impact of reusability in several applications are:

- Asset Usage ratio;
- Total effort saving due to reuse;
- Time to market improvement due to reuse;
- Defect reduction due to reuse;
- Effort saved due to reuse.

Infosys implements quantitative quality management through defect prediction. Defect patterns observed in past projects are used for predicting defect levels. Metrics commonly used for defect analysis here is:

- Delivered Defect Density: number of defects per unit size in the delivered software;
- Defect Removal Efficiency: DRE is a measure to detect defects before delivery. It is calculated as a percentage of the defects identified and corrected internally with respect to the total defects in the complete project life cycle. Thus, DRE is the percentage of bugs eliminated by reviews, inspections, tests etc.

Test metrics are the key “facts” that project managers can use to understand the current position and prioritize various activities. Key metric used for testing at Infosys is:

Test Progress Tracking metric: Track the cumulative test cases or test points- planned, attempted and successful, over the test execution period. Plot these three parameters on the Y axis and the X axis can be the timeline. The purpose of this metric is to closely monitor the testing progress. Any slippages from the plan can be clearly seen [3, 4, 5].

2.3. NASA

It is one example where getting the software right the first time is critical since they may only get one chance. To improve reliability, they consider three life cycle phases: requirements, coding, and testing. Software metrics used are:

Requirements Metrics: For reliability, NASA’s metric tool (called ARM - Automated Requirements Measurement) parses requirements document file line by line searching for certain words and phrases. It verifies the structure of requirements document and indicates unsuitable levels of details in software design.

Design and Code Reliability Metrics: NASA’s Software Assurance Technology Center (SATC) developed a tool that analyzes source code for architecture features and structure and to help locate error-prone modules using a combination of size and complexity.

Testing Reliability Metrics: A simulation model based on Musa model is used for error discovery and projecting number of remaining errors in the source code. .Effective verification aims to ensure that every requirement is being tested[6].

3. SOFTWARE TOOLS IN INDUSTRY

Using software tools can improve the software quality and reduce overall development time. Software CASE tools are used in almost all stages of the software development. They are used to speed up the capture of top-level architecture and design and detect inconsistencies in logic and structure of the software even before the code is written. Tools that are used for coding and unit testing are compilers, linkers, loaders, debugger, simulators etc. Further, the usage of Code browsers, static error analysis and rule enforcement tools has been discussed. The commercial products identified have been selected from those available with Navy.

Code Browser Tools: It is used to navigate through the code with just a few mouse clicks and is much more powerful than a typical editor included with a compiler. It displays tree hierarchy of function calls. It also displays the definition of a variable and all the places its used in a cross reference table.

Static Error Analysis tools: This type of tool catches all the errors including those which get missed by the compiler. An example is PC-Lint™. A Lint tool will find the use of uninitialized variables, null pointers, array bound errors, overflow in expressions, precedence problems, misplaced semicolons and many other problems. It also gives warning messages about unsafe practices such as shifting signed integers, not using default statement with a switch statement etc.

Rule Enforcement Tools: Many organizations such as Motor Industry Software Reliability Association (MISRA) have a set of rules for programmers to follow in order to minimize defects. This rule enforcement process can be very tedious and cumbersome without using automated tools. After the code has been compiled a rule formatting tool can be used to automatically reformat the source to conform to the rules. For example: software tool such as CodeWizard™ has over 500 rules to compare with the code. Visual SlickEdit™ has the feature for brace placement and putting blanks where needed [7, 8].

All the above mentioned automated tools can bring a significant improvement in quality, reliability and productivity of the software produced if properly implemented. Apart from using the help manuals provided with these tools, proper training programs for their usage must be conducted so that all their features can b fully explored and utilized.

4. Scientific Software Applications in Industry

Emergence of scientific software applications has revolutionized engineering design. They allow users to perform computations and calculations with much accuracy and sophistication. Many traditional analytical techniques are being replaced by them. Some of the most commonly used scientific applications have been briefly discussed below:

MATLAB: Developed by MathWorks, primarily used for numerical computing including functions like matrix multiplications, plotting of functions and data, implementation of algorithms, creation of user interfaces and interfacing with programs written in other languages including C,C++ and Fortran. It also comprises of the MuPad which is a computer algebra system to manipulate formulas symbolically.

MATHEMATICA: Developed by Wolfram Research Group, an integrated system which delivers unprecedented workflow, coherence, reliability, and innovation [9]. Rather than different toolkits for different jobs, it has been built to deliver one vision: the ultimate technical application and environment. It has built in multi-core, platform-optimized numerical computation, making it suitable for the most computationally intensive problems. It contains ready-to-use curated data in areas as diverse as chemistry, geopolitics, finance, and geometry. It also comprises of tools for parallel programming.

MAPLE: Developed by Symbolic Computation Group, it is a general-purpose commercial computer algebra system. Users can enter mathematics in traditional mathematical notation. There is extensive support for numeric computations to arbitrary precision, as well as symbolic computation and visualization. Its smart document environment automatically captures all of your technical knowledge in an electronic form that seamlessly integrates calculations, explanatory text and math, graphics, images, sound, and diagrams.

The emergence of all these scientific software is a new capability, which can become a fundamental tool for sophisticated designers in industry. However, for its success, it should be fully embraced in engineering curricula.

5. Recommendations

After discussing the usage of metrics, software tools and scientific software applications in various industries certain potential improvement areas have been suggested as below:

- Since measurements tend to change human behavior, it is important to select those measurements that will cause behavioral changes in positive and beneficial directions. Measurement activities should be developed as a habit rather than burden on developers and the organization.
- Implementation of automated tools for various software development stages can greatly improve productivity and quality. So proper training must be provided by the organization for the optimum use of these tools.
- To reduce the gap between viewpoint of industry and academia, knowledge regarding scientific software tools should be made a part of engineering curriculum to enrich the deigning capabilities of students.

6. Conclusion

Successful application of all the three: “Metrics, software tools and scientific software applications” simultaneously, can revolutionize the software development industry. Though many industries are making extensive use of software metrics, the use of software tools and scientific software packages is still limited. Despite the widespread application of metrics, many firms lack a comprise picture of workforce effectiveness. The main reason behind this is volume and cost based metrics which can’t easily correlate to workforce productivity and engineer’s performance. The automated software tools can greatly reduce the overall development and debugging time. So, the software industry people should realize that it is well worth the money and time it takes to purchase, install, learn and use these tools. The emergence of scientific software applications bring a new trend in engineering design by allowing users to perform computations and calculations at very high levels of sophistication. With the help of these three utilities a base framework can be created which can be used in other projects as well.

REFERENCES

[1] Pramod Abhichandani, Richard Primerano and Moshe Kam, "Symbolic Scientific Software Skills for Engineering Students", 2010 IEEE.

[2] http://edge.sapient.com/assets/ImageDownloader/157/Realizing_Hidden_Value_FINAL.pdf

[3] Pankaj Jalote, "Managing Software Projects:The Infosys Model", April 2002.

[4] Abhay Joshi, "A Model for Effective Asset Reuse in Software Projects", Infosys White Paper.

[5] Pankaj Jalote, et.al "Quantitative Quality Management through Defect Prediction and Statistical Process Control", www.cse.iitk.ac.in/jalote/papers/2WCSQPaper.pdf

[6] J. Ordonez and M.Haddad, "The State of Metrics in Software Industry", Fifth International Conference on Information Technology, 2008 IEEE.

[7] J.L. Anderson Jr., "Using Software Tools and Metrics to Produce Better Quality Test Software", 2004 IEEE AUTOTESTCON.

[8] Sadhna Kapur and Chak Sriprasad, "Software Development Tools for Embedded Systems", Intel Corp, 1995 IEEE.

[9] http://www.wolfram.com/products/mathematica/index.html