Reliability Evaluation of Web Applications from Click-stream Data

Prof. R.Manjula
Associate Professor
School of Computing Sciences and Engineering
VIT University, Vellore – 632014.
Tamil Nadu, India.

Eswar Anand Sriram
M.S(Software Engineering)
School of Information Technology,
VIT University, Vellore – 632014.
Tamil Nadu, India.

ABSTRACT
In the age of information and communication technology (ICT), Web and internet have brought significant changes in information technology. The dramatic change in website development and their relative usage has led to the need of Web based metrics .These metrics will accurately assess the efforts in the web based applications .So the basic idea is to identify the web metrics for evaluating reliability and maintainability of Hypermedia applications where we characterize usage and problems for web applications, evaluate their reliability and also the potential aspects for reliability assessment and improvement. Based on the characteristics of web applications and the overall web environment, we classify web problems and focus on the subset of source content problems. Using information about web accesses, we drive various measurements that can characterize web site workload at different levels of granularity .These workload measurements, together with failure information extracted from recorded errors are used to evaluate the operational reliability for source events at a given website and the potential for reliability improvement. As a result, to support this strategy or methodology we extract web usage and failure information from existing web logs. This failure information is used to measure the reliability of web applications. Hence these results obtained from the web based metrics can help us analytically identically identify the effort assessment and failure points in web based systems and make evaluation of reliability of these systems simple.

The rest of the paper is organized as follows: Section 2 Analyzes the general characteristics of the Web and its reliability problems. Section 3 examines the contents of Web logs and their use in evaluating Web site workload and Web software reliability. Conclusions and perspectives are presented in Section 4.

Index Terms
World Wide Web (WWW) and Internet, Web applications and Web server logs, quality and reliability, reliability modeling, workload measurement.

1. INTRODUCTION
With the prevalence of the World Wide Web (WWW, or simply the Web) and people's reliance on it in society today, ensuring its satisfactory reliability is becoming increasingly important. Various techniques exist today to characterize workload for general software and computer systems and to measure and assure their reliability [9], [12], [21]. However, the Web environment presents many new challenges [6], [14] and requires adapted or newly developed techniques based on the characterization of the Web, its usage, and related problems. For Web applications, various log files are routinely kept at Web servers. In this paper, we extract Web usage and failure information from these log files to evaluate Web software reliability and the potential for reliability improvement.

2. RELIABILITY AND THE WEB
We next examine the general characteristics of the Web and common problems in Web applications to set the stage for us to evaluate Web software reliability.

2.1 Defining Reliability for Web Applications and Their Components
The reliability for Web applications can be defined as the probability of failure-free Web operation completions. We define Web failures as the inability to correctly obtain or deliver information, such as documents or computational results, requested by Web users. This definition conforms to the standard definition of failures as being the behavioral deviations from user expectations [5]. Based on this definition, we can consider the following failure sources:

- **Host, network, or browser failures,** that prevent the delivery of requested information to Web users. These failures are similar to failures in regular computer systems, network, or software, which can be analyzed and assured by existing techniques [9], [12], [21].
- **Source content failures,** which prevent the acquisition of the requested information by Web users because of problems such as missing or un-accessible files, trouble with starting JavaScript, etc. These failures are closely related to the specific Web-based services that a site provides, and possess various characteristics unique to the Web environment [6], [14].
- **User errors,** such as improper usage, mistyped URL, etc., may also cause problems, which can be addressed through user education, better usability design, etc. These failures are beyond the control of Web service or content providers.

The end-to-end reliability defined earlier, which measures the probability of failure-free completions of Web operations, includes all the problems listed above in its reliability evaluation. However, as also noticed above, many of these problems can be either addressed by existing approaches or are simply beyond the control and responsibility of the local Web content providers. In addition, ensuring reliability defined this way would require concerted quality assurance effort over the whole Internet by the global community. On the other hand, Web site software problems, or Web source content problems noted above, are a significant part of the overall problems for Web operations. In addition, they can generally be addressed locally at the Web site by the content providers. Consequently, we focus on the Web source content failures and the related Web software reliability in this study. Also worth noting is the differences between Web software reliability we restrict ourselves to and Web site availability. Normal maintenance activities and network problems may make a Web site temporarily unavailable. However, such problems are generally perceived as less serious by Web users than Web software problems because the users are more likely to succeed in accessing required information after temporary unavailability, while software problems would persist unless the underlying causes are identified and fixed. This fact also partially justifies our focus on Web software reliability.

2.2 Measuring Web Software Reliability and Workload

In general, the failure information alone is not adequate to characterize and measure the reliability of a software system, unless there is a constant workload [9], [12]. Due to the vastly uneven Web traffic observed in previous studies [1], [15], we need to measure both the Web failures and related workload for reliability analyses. Specific characteristics that make Web workload measurement different from that for traditional software systems include:

- **Massiveness and diversity:** Web applications provide cross-platform universal access to Web resources for everyone with an Internet access. The massive user population, the diverse hardware or software configurations, and the varied usage patterns need to be reflected in the selected workload measures.
- **Document and information focus,** as compared to the computational focus for most traditional workload. Although some computational capability has evolved in newer Web applications, information search and retrieval still remain the dominant usage for most Web users. A fundamental difference exists between these two workload types.

These characteristics require us to measure actual Web workload to ensure its satisfactory reliability instead of indiscriminately using generic measures suitable for traditional computation-intensive workload. Due to the nature of uneven Web workload, only usage-dependent workload measures among the traditional ones, such as CPU execution time, runs, and transactions, need to be considered for reliability evaluation [9],[12]. However, the user focus and substantial amount of idle time during browsing sessions make any variation of execution time unsuitable for Web workload measurement. Similarly, the dominance of non-computational tasks also makes computational task oriented transactions unsuitable for Web workload measurement.

2.3 Basics of Reliability Analysis and Modeling

Both the failure information and the related workload measurements provide us with data input to various software reliability models [9], [12], [18]. The output of these models can help us evaluate the Web software reliability and the potential for reliability improvement.

Two basic types of software reliability models are: input domain reliability models (IDRMs) and time domain software reliability growth models (SRGMs) [9], [12]. IDRMs can provide a snapshot of the Web site's current reliability. For example, if a total number of f failures are observed for *n* workload units, the estimated reliability R according to the Nelson model [13], one of the most widely used IDRMs, can be obtained as:

$$R= (n-f)/n=1-(f/n) =1-r$$

Where r is the failure rate, which is also often used to characterize reliability. When usage time ti is available for each workload unit i, the summary reliability measure, mean-time-between failures (MTBF), can be calculated as:

$$MTBF= (1/f) \sum_{I} t_i$$

When the usage time ti is not available, we can use the number of workload units as the rough time measure. In this case,

$$MTBF=n/f$$

If discovered defects are fixed over the observation period, the defect fixing effect on reliability (or reliability growth due to defect removal) can be analyzed by using various software reliability growth models (SRGMs) [9], [12]. For example, in the widely used Goel-Okumoto model [4] the failure arrival process is assumed to be a non-homogeneous Poisson process. The expected cumulative failures, m (t), over time t is given by the formula:

$$m(t)=N(1-e^{-bt})$$

Where the model constants N (total number of defects in the system) and b (model curvature) need to be estimated from the observation data. SRGMs can also be used to assess the potential for reliability improvement.

2.4 Analyzing Web Logs for Reliability Evaluation

Monitoring Web usage and keeping various logs are necessary to keep a Web site operational. Two types of log files are commonly used by Web servers: Individual Web accesses, or hits, are recorded in access logs, with sample entries given in Table 1; related problems are recorded in error logs, with sample entries given in Table 2. Analyzing information stored in such logs can help us evaluate Web site workload and Web software reliability, as discussed below.

Table 1. Sample Entries in an Access Log

```
129.119.4.17 - - [16/Aug/1999:00:00:11 -0500] "GET /img/XredSeal.gif HTTP/1.1" 301 328
"http://www.seas.smu.edu/" "Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)"
129.119.4.17 - - [16/Aug/1999:00:00:11 -0500] "GET /img/ecom.gif HTTP/1.1" 304 -
"http://www.seas.smu.edu/" "Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)".
```

Table 2. Sample Entries in an Error Log

```
[Mon Aug 16 13:17:24 1999] [error] [client 207.136.6.6] File does not exist:
/users/seasadm/webmastr/htdocs/library/images/gifs/homepage/yellowgradlayers .gif
[Mon Aug 16 13:17:37 1999] [info] [client 199.100.49.104] Fixed spelling:  /img/XredSeal.gif
to /img/xredSeal.gif from http://www.seas.smu.edu/
```

3. ERROR LOG ANALYSIS

Error logs typically include details about the problems encountered. The format is simple: a time-stamp followed by the error or warning message, such as in Table 2.Common problems or error types are listed in Table 3.Notice that most of these errors conform closely to the source content failures we defined in Section 2. Therefore, they can be used in our Web software reliability evaluation. Questions about error occurrences and distribution can be answered directly by analyzing error logs. However, as discussed in Section 2, evaluation of Web software reliability also needs the measurement data for Web usage or workload. The Web usage information and the related workload measurements can be extracted from Web server access logs, as described below.

3.1 Analysis of Access Log Contents

A "hit" is registered in an access log if a file corresponding to an HTML page, a document, or other Web content is explicitly requested, or if some embedded content, such as graphics or a Java class within an HTML page, is implicitly requested or activated.

Table 3. Error Types

type	description
A	permission denied
B	no such file or directory
C	stale NFS file handle
D	client denied by server configuration
E	file does not exist
F	invalid method in request
G	invalid URL in request connection
H	mod_mime_magic
I	request failed
J	script not found or unable to start
K	connection reset by peer

Information recorded in access logs typically includes: the requesting computer, date and time of the request, name and size of the requested file, HTTP status code, referral page, and client name. Specific information useful to our workload analysis recorded in access logs includes:

- The IP number of the machine making the request.
- Date and time that the transfer took place.
- Total number of bytes transferred.

They are recorded as the first, fourth, and seventh field, respectively, of each hit entry in the access log, as illustrated in Table 4. If the value for any field is not available, a "_" is put in its place.

3.2 Extracting Workload Measures from Access Logs

As mentioned in Section 2, various software reliability models relate observed failures to usage time for reliability evaluation. From the perspective of Web service providers, the usage time for Web applications is the actual time spent by every user at the local Web site. However, the exact time is difficult to obtain and may involve prohibitive cost or overhead associated with monitoring and recording dynamic behavior by individual Web users [15]. One additional complication is the situation where a user opens a Web page and continues with other tasks unrelated to the page just accessed. In this situation, the large gap between successive hits is not a reflection of the actual Web usage time by this user. To approximate the usage time, we can use various workload measures considered below.

The most obvious workload measure is to count the number of hits, because each hit represents a specific activity associated with Web usage, and each entry in an access log corresponds to a single hit, thus it can be extracted easily. In fact, this has already been done for statistical Web testing and reliability assurance, which also demonstrated that hit count is a viable candidate for the evaluation of Web site workload and Web software reliability.

Table 4. Summary of Total Recorded Errors by Type

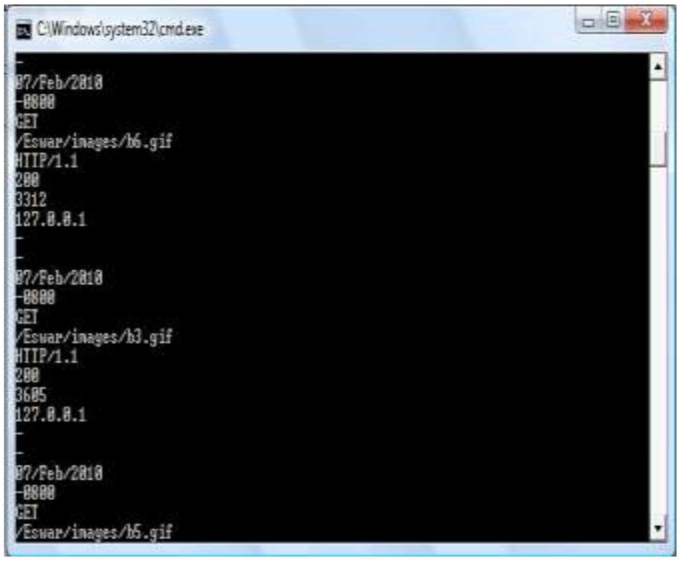
Error type	A	B	C	D	E	F	G	H	I	J	K	Total
Number of errors	2079	14	4	2	28631	0	1	1	1	27	0	30760

Table 4 illustrates the number of errors for a particular error type. Based on the total number of errors and the workload measures reliability can be calculated by using input domain reliability models.

The Web workload measures at different levels of granularity and from different perspectives that we can extract from Web server access logs include:

- **Number Of Hits,** The overall hit count defined above can be misleading if the workload represented by individual hits shows high variability.
- **Number of Bytes Transferred**, The number of bytes transferred, or byte count, as the workload measure of finer granularity, which can be easily obtained by counting the number of bytes transferred for each hit recorded in access logs.
- **Number Of Users,** User count is another alternative workload measure meaningful to the organizations that maintain the Web sites and support various services at the user level. When calculating the number of users for each day, we treat each unique IP address as one user. So, no matter how many hits were made from the same computer, they are considered to be made by the same user. This measure gives us a rough picture of the overall workload handled by the Web site. One of the drawbacks of user count is its coarse granularity, which can be refined by counting the number of user sessions. In this case, along with the IP address, access time can be used to calculate user sessions: If there is a significant gap between successive hits from the same IP address, we count the later one as a new session. In practice, the gap size can be adjusted to better reflect appropriate session identification for the specific types of Web applications.
- **Number of User Sessions**, The number of user sessions per day may be a better measure of overall workload than the number of users, because big access gaps are typically associated with changes of users or non-Web related activities by the same user. Each user who accesses the same Web site from the same computer over successive intervals will be counted by user sessions, as long as such a gap exists in between. Even for a single user, a significant access gap is more likely to be associated with different usage patterns than within a single time burst. Therefore, by using user sessions, we can count the users' active contribution to the overall Web site workload more accurately.
  .

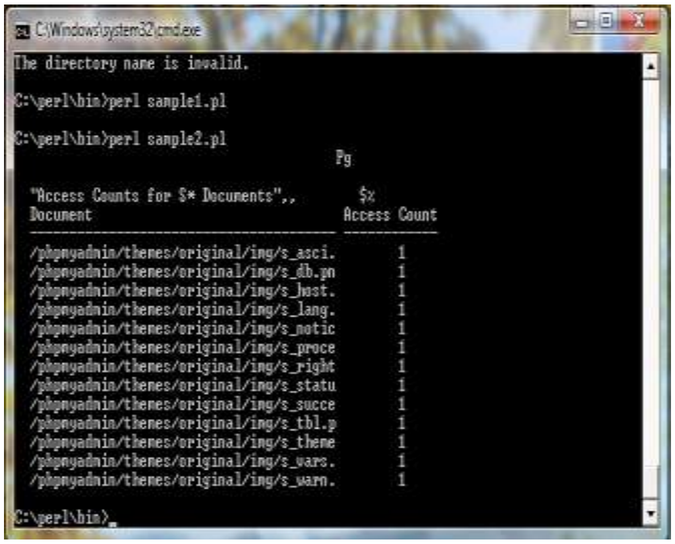
We implemented utility programs in Perl to count the number of errors, number of hits, and frequently used navigation patterns. These utility programs analyze the workload data defined above. The output for the utility programs is demonstrated below.



```
-
07/Feb/2010
-0800
GET
/Eswar/images/b6.gif
HTTP/1.1
200
3312
127.0.0.1
-
-
07/Feb/2010
-0800
GET
/Eswar/images/b3.gif
HTTP/1.1
200
3605
127.0.0.1
-
-
07/Feb/2010
-0800
GET
/Eswar/images/b5.gif
```

Figure 1. Extracting the Access Log File

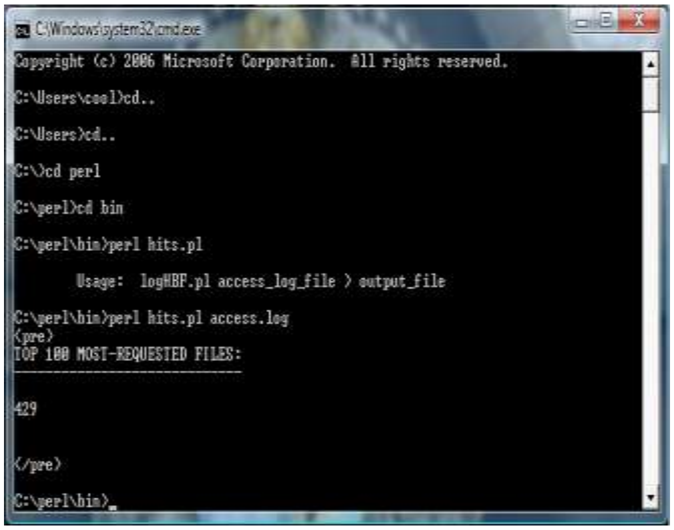
The above figure demonstrates extracting the access log file and dividing it into readable format to analyze the log file.



```
The directory name is invalid.

C:\perl\bin>perl sample1.pl

C:\perl\bin>perl sample2.pl
                                       Pg

  "Access Counts for $* Documents",,       $x
  Document                                 Access Count
  ---------------------------------------- ------------
  /phpmyadmin/themes/original/img/s_asci.        1
  /phpmyadmin/themes/original/img/s_db.pn        1
  /phpmyadmin/themes/original/img/s_host.        1
  /phpmyadmin/themes/original/img/s_lang.        1
  /phpmyadmin/themes/original/img/s_notic        1
  /phpmyadmin/themes/original/img/s_proce        1
  /phpmyadmin/themes/original/img/s_right        1
  /phpmyadmin/themes/original/img/s_statu        1
  /phpmyadmin/themes/original/img/s_succe        1
  /phpmyadmin/themes/original/img/s_tbl.p        1
  /phpmyadmin/themes/original/img/s_theme        1
  /phpmyadmin/themes/original/img/s_vars.        1
  /phpmyadmin/themes/original/img/s_warn.        1

C:\perl\bin>_
```

Figure 2. Calculation of No. of Hits per page

The above figure demonstrates the individual hit count per each page and this result can be used as a workload measure.



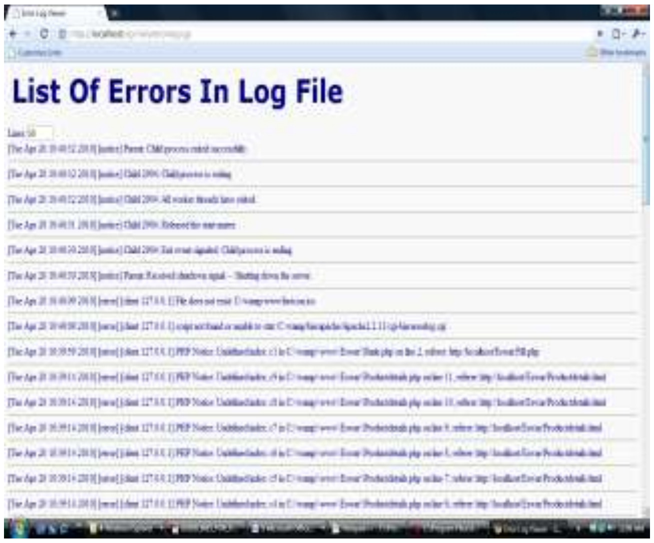
```
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\cool>cd..

C:\Users>cd..

C:\>cd perl

C:\perl>cd bin

C:\perl\bin>perl hits.pl

        Usage:  logHBF.pl access_log_file > output_file

C:\perl\bin>perl hits.pl access.log
<pre>
TOP 100 MOST-REQUESTED FILES:
------------------------------

429


</pre>

C:\perl\bin>_
```

Figure 3. Calculation of Total No: of Hits

Fig 3 demonstrates the total number of hits for a website which can be considered as a workload measure for the calculation of reliability.

3.3 Error Log Analysis and Reliability Assessment

Questions about error occurrences and distribution, as well as overall reliability of the Web site, can be answered by analyzing error logs and related information. However, although there are many Web log analysis tools available today, they only provide very limited capability for error analysis. For example, the log analyzer's Analog and FastStats only analyze access logs for common HTML errors. Most log analyzers do not analyze error logs because of the lack of a consistent format. Therefore, existing tools are only used in our approach to analyze the general usage information, while new utility programs are constructed to analyze error data for testing effectiveness and reliability analyses. As discussed before, access logs only contain less detailed information about a subset of the errors reported in error logs. In addition, if we see a large number of similar errors in an access log, there might be other problems (not recorded in the access log but may be recorded in the error log) not related to the application or the Web server but with the ISP (Internet service provider). Therefore, analyzing error logs is necessary to help us locate and debug problems and to assure the quality of Web applications.We have constructed utility programs written in Perl to obtain the required error or defect data. These programs also extract various usage data from access logs. Such usage and error data can be used to provide an objective assessment of the reliability of Web applications, by fitting these data to various reliability models. The output for the utility programs is demonstrated below.



List Of Errors In Log File

[illegible]

Figure 4. List of Errors in Log File

Properly selected usage and failure data can be fitted to various input domain reliability models to provide a snapshot of the Web applications' current reliability. For example, if a total number of f errors are recorded (referred to as failures in software reliability engineering, denoting behavioral deviations) for n hits, the estimated reliability R according to the Nelson model ,one of the most widely used input domain reliability models, can be obtained as:

$$R= (n-f)/n=1-(f/n) =1-r$$

4. CONCLUSION AND PERSPECTIVES

By analyzing the unique problems and challenges for the Web environment, we have developed an approach for Web software reliability evaluation based on information extracted from existing Web server logs. By using existing tools to extract usage information, we have kept the additional effort for implementing our approach to a reasonable level. We developed utility programs in Perl to analyze Web logs; we have provided necessary capabilities not supported by existing tools. Our key findings are summarized below:

- *Measure derivation and data extraction*: Specific Web software problems related to missing files and four workload measures, bytes transferred, hit count, number of users, and number of sessions, were derived in this paper for Web software reliability evaluation. Detailed failure data can be extracted from error logs. When such logs are not available, rough failure data can be extracted from access logs. Hit count, byte count, and user count can be easily extracted from access logs, due to their direct correspondence to access log entries and the embedded data fields "bytes transferred" and "IP address." Session count computation may involve history information for individual users or unique IP addresses, but properly identified user sessions with appropriate time-out values can reflect Web usage better than simply counting the users.

- Assessing the operational reliability for Web software: When used with failure data to estimate failure rate or reliability, all four workload measures proposed in this paper produced more consistent and stable reliability estimates than using daily errors alone. They offer reliability assessments from different perspectives, and each may be suitable for specific situations. For example, byte count might be more suitable for traffic measurement and related reliability interpretations; hit count might be more meaningful to Web users as they browse individual pages; while number of users or sessions might be more suitable for high level Web site reliability characterization.

There are also some open issues we plan to address in future studies, including:

- *The impact of Web site changes and related fault injections*: Our reliability analyses performed in this paper assumed the stability of the Web sites under study, and our evaluation of reliability growth potential additionally assumed that none or few new faults were injected. Therefore, a direct generalization of this study is to study the impact of Web changes and injection of new faults on Web software reliability.
- *Risk identification for reliability improvement*: The error distribution is highly uneven, as shown in this paper and demonstrated in further studies we performed to examine the error distributions across error types, originators, error sources, page types, etc. These uneven distributions point out the importance of applying risk identification techniques to identify problematic areas in the future for focused Web software reliability improvement.
- *Better ways to count bytes transferred*: Byte counting in this paper ignored about 15 percent of access log entries with missing information for their “byte transferred” field, which typically correspond to error entries and cached Web contents. Treating them as 0s is convenient but runs contrary to the general practice in software reliability engineering, where all usage time or activities should be counted regardless of whether the specific usage resulted in successful completions or failures. This fact points to the need for further investigation and possible alternative data treatment when we use bytes transferred for reliability analyses.

In addition, we also plan to identify better existing tools, develop new tools and utility programs, and integrate them to provide better implementation support for our strategy. All these efforts should lead us to a more practical and effective approach to achieve and maintain high reliability for Web applications.

5. REFERENCES

[1] M.F. Arlitt and C.L. Williamson, “Internet Web Servers: Workload Characterization and Performance Implications,” IEEE/ACM Trans. Networking, vol. 5, no. 5, pp. 631-645, Oct. 1997.

[2] B. Behlandorf, Running a Perfect Web Site with Apache, second ed. MacMillan Computer Publishing, 1996.

[3] M.E. Crovella and A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes,” IEEE/ACM Trans.Networking, vol. 5, no. 6, pp. 835-846, Dec. 1997.

[4] A.L. Goel and K. Okumoto, “A Time Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures,” IEEE Trans. Reliability, vol. 28, no. 3, pp. 206-211, 1979.

[5] IEEE Standard Glossary of Software Engineering Terminology, Number STD 610.12-1990, IEEE, 1990.

[6] C. Kallepalli and J. Tian, “Measuring and Modeling Usage and Reliability for Statistical Web Testing,” IEEE Trans. Software Eng., vol. 27, no. 11, pp. 1023-1036, Nov. 2001.

[7] A.G. Koru and J. Tian, “Defect Handling in Medium and Large Open Source Software Projects,” IEEE Software, vol. 21, no. 4, pp. 54-61, July 2004.

[8] Z. Li and J. Tian, “Analyzing Web Logs to Identify Common Errors and Improve Web Reliability,” Proc. IADIS Int’l Conf. E-Society, pp. 235-242, June 2003.

[9] Handbook of Software Reliability Engineering. M.R. Lyu, ed. McGraw-Hill, 1995.

[10] L. Ma and J. Tian, “Analyzing Errors and Referral Pairs to Characterize Common Problems and Improve Web Reliability,” Proc. Third Int’l Conf. Web Eng., pp. 314-323, July 2003.

[11] A.L. Montgomery and C. Faloutsos, “Identifying Web Browsing Trends and Patterns,” IEEE Computer, vol. 34, no. 7, pp. 94-95, July 2001.

[12] J.D. Musa, A. Iannino, and K. Okumoto, Software Reliability: Measurement, Prediction, Application. McGraw-Hill, 1987.

[13] E. Nelson, “Estimating Software Reliability from Test Data,” Microelectronics and Reliability, vol. 17, no. 1, pp. 67-73, 1978.

[14] J. Offutt, “Quality Attributes of Web Applications,” Software, vol. 19, no. 2, pp. 25-32, Mar. 2002.

[15] J.E. Pitkow, “Summary of WWW Characterizations,” World Wide Web, vol. 2, nos. 1-2, pp. 3-13, 1999.

[16] N.F. Schneidewind, “Software Reliability Model with Optimal Selection of Failure Data,” Trans. Software Eng., vol. 19, no. 11, pp. 1095-1104, Nov. 1993.

[17] N. Singpurwalla, “Software Reliability Modeling by Concatenating Failure Rates,” Proc. Ninth Int’l Symp. Software Reliability Eng., pp. 106-110, Nov. 1998.

[18] R. Thayer, M. Lipow, and E. Nelson, Software Reliability. North- Holland, 1978.

[19] J. Tian, “Integrating Time Domain and Input Domain Analyses of Software Reliability Using Tree-Based Models,” IEEE Trans. Software Eng., vol. 21, no. 12, pp. 945-958, Dec. 1995.

[20] J. Tian, “Better Reliability Assessment and Prediction through Data Clustering,” IEEE Trans. Software Eng., vol. 28, no. 10, pp. 997- 1007, Oct. 2002.

[21] K.S. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, second ed. John Wiley & Sons, 2001.

[22] Tsai, W.T., Zhang, D., Chen, Y., Huang, H., Paul, R., and Liao, N., "A Software Reliability Model for Web Services," Proceedings of IASTED International Conference on Software Engineering and Applications, pp. 144-149, MIT Cambridge, USA, 2004.

[23] Venkatraman, S.: Mobile Computing Models – Are they Meeting the Mobile Computing Challenges? Association of Computing Machinery Journal. New Zealand. 1 (2005) 112

[24] Dudley, G., Joshi, N., Ogle, D.M., Subramanian, B. and Topol, B.B.:Autonomic Self-Healing Systems in a Cross-Product IT Environment. In Proceedings of the International Conference on Autonomic Computing (2004)312-313

[25] Vidales, P., Baliosian, J., Serrat, J., Mapp, G., Stajano, F. and Hopper, A.: Autonomic System for Mobility Support in 4G Networks. IEEE Journal on Selected Areas in Communications. 23 (12) (2005) 2288-2304

[26] Kumar, V., Cooper, B.F., Cai, Z., Eisenhauer, G., Schwan, Resource-aware distributed stream management using dynamic overlays. In: Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS 2005), Columbus, OH, USA, IEEE Computer Society (2005) 783–792.

Authors

Prof. R.Manjula received her B.E in Computer Science & Engineering from University of Vishwesvaraya and Engineering, Bangalore, Karnataka State, India in 1992 and M.E in Software Engineering from Anna University, Tamil Nadu, India in 2001. She is now working as Associate Professor and also as Ph.d Candidate affiliated with School of Computing Sciences and Engineering at Vellore Institute of Technology, Vellore, India. Her area of specialization includes Software Process modeling, Software Metrics,

Eswar Anand Sriram received his MS in Software Engineering from Vellore Institute Of Technology, Vellore , India in 2010.He is now working as Assistant Systems Engineer at TATA Consultancy Services , Chennai , India . His area of interest includes Software Metrics , Operating Systems and Design Patterns.