

Calculating Website's Usability Metrics Using Log File Information

Marcus Vinicius Cerrato and Marcelo Morandini

School of Arts, Science and Humanities, University of São Paulo, São Paulo, Brazil
{mcerrato,m.morandini}@usp.br

Abstract. Log files are interesting.

Keywords: usability, evaluation, log file, metrics.

1 Introduction

Usability describes how a system is easy to be used. It is the quality of a system that allows its users, in a given context of use, to get specific objectives in an efficient and effective manner, while promoting satisfaction.¹ Usability may be a matter of survival, because users will prefer not to use systems that present difficulties to be learned or that do not permit an easy achieving of specific goals.² This is particularly true in the context of websites.

Usability evaluation is increasingly becoming part of software development process. As long as technology evolves, users are becoming more exigent and demand more frequent system's updates. The product of this updates may be new interfaces which should be evaluated to verify whether the changes made to the system were efficient and efficacious.

Log files are text files where all interactions with a website are kept. Using web server log files to evaluate a website is a remote evaluation method. This is a practice and costless way to evaluate a website's usability, and helps the gathering of usability metrics.³

This method has the advantage of not interfering in the common use of the website. Users will not feel part of the evaluation process this way. Evaluators will not even conduct the evaluation process at the same time the system is being used.

On the other hand, this process has its own problems, so that, for example, sometimes it will not be possible to know if the user found what he was looking for or what action the user was intending to do. This happens because web server log files only have results of interactions with websites, but they don't have the purpose of such interactions neither their objectives. And as we are using a remote evaluation method, we have only log file information to use.

¹ ISO 9241 (1997)

² Nielsen (2000)

³ Ivory & Hearst (2001)

Although accessing and analyzing the log files should not provide meaningful answers, they can give us an approach of which areas of the website need a further evaluation. Other more expensive evaluation methods should be used in these cases to analyze these areas.

Doing this way we are promoting a quick evaluation with the information we have at hand, to evaluate with more details later only in the problematic points. It is important to mention that log file analysis give us only a tip of what area should be further evaluated and don't take the place of other evaluation methods, specially the qualitative ones as questionnaires and checklists. We suggest a methodology of how this can be done with a website.

We have chosen Apache as the web server model to our research, because it has almost 55% of the whole market.⁴ As Apache log files have their information in two standard formats and most people don't change it, we know exactly what information we will find and in what order.

The process of evaluating a website using the analysis of the web server log files should be automated, because log files have a lot of information to be analyzed. Our tool makes use of Regular Expressions (a special set of texts to describe a search pattern) to separate small significant pieces from the log file lines, so that we can get relevant and meaningful information to be compared.

The standard formats of Apache log files are:

```
LogFormat "%h %l %u %t \"%r\" %>>s %b" common
LogFormat "%h %l %u %t \"%r\" %>>s %b \"%{Referer}i\"
\"%{User-agent}i\" "
combined
```

The common type is included in the combined type. The information contained in such formats are: remote server, remote logname, remote user, moment of request receiving, request first line, request status, answer size in bytes, browser and operating system that make the request, and the request properly. Request status may have the following values: success, redirecting, client error, and server error.

Apache log files also have some other information that are not essential resources in order to describe what users have been doing while using the website.

We used a PHP parser in order to obtain the parts of the Apache log file parts by regular expression. The code line to do this is:

```
preg_match("/^(\\S+) (\\S+) (\\S+) \\[([[:]]+):(\\d+:\\d+:\\d+)
([^[\\]]+)] \\\"(\\S+) (.*) (\\S+)\\\" (\\S+) (\\S+) (\\..*?\\")
(\\..*?\\")$/", $line, $matches); // pattern to format the
line
```

Initially we must identify all the web pages of the website being analyzed. The job of finding the best task mapping for the website is the most important of all the process and affects the final result.

⁴ Netcraft (2013)

After that we map the tasks of the website that we would like to analyze. In other words, we select the sequences of web pages that characterize the independent actions that are possible to be done in the website and that we would like to analyze. Only the lines of the log file which describe the tasks possible to be done with the web site are important in this analysis. Filtering the entries of the log file helps to achieve a more accurate result.

As log files are such big text files, we must take from the web server only the ones which correspond to the dates we would like to analyze. Besides that, we may filter only some IPs or take the whole file. This is done by a command from the operational system. We present examples of the commands in Windows and in Linux.

```
cat default_access.log.3 | grep 127.0.0.217 >> teste.log
cat default_access.log.3 | egrep
'127.0.0.217|127.0.0.218' >> teste.log
```

[Example of command line to filter by IP in Linux]

```
type default_access.log.3 | findstr 127.0.0.217 >
teste.log
type default_access.log.3 | findstr "127.0.0.217
127.0.0.218" > teste.log
```

[Example of command line to filter by IP in Windows]

Our tool will look inside the selected and filtered log files, and compare each entry with the tasks previously registered. The time difference will be calculated also. After collecting the rates related with the completed tasks and the times for doing that, metrics will be calculated.

Our main interest is collecting metrics to quantify the usability of the web site. The metrics we calculate are based on some rates that were chosen because they are closely related to usability. The rates collected are:

- Hit Rate: it counts every URL that were contemplated by the search conditions, for example dates interval, beginning and ending times;
- Task Complete Rate: it counts how many times a certain task was completed, in other words, all its URLs were hited, from first to last, without interruption and in the specified order;
- Task Incomplete Rate: it make us know how many times a task was initiated but not completed for such reason (not in the order, not from first to last);
- No Task Rate: it indicates the moments where the URL found is not part of any task being analysed; for great values we suggest the inclusion of other tasks or the splitting of the existing ones, so that the conditions can be fitted;
- Agent Change Rate: it may indicate moments in which the user couldn't continue using the browser or the operating system because of some problem between this resource and the web site;

- Long Time Rate: it may indicate a problem in using the web site, because a long time has passed without internet activity, but it is not conclusive because the user could go out for a cup of water or was in a telephone conversation.

The metrics are calculated by the division of the rates for the first one. This way we obtain values which help in a usability website evaluation that is remote and automated. For each task this metrics get how the use of the functionalities of the website is.

References

1. ISO/DIS 9241. Dialogue Principles in Guiding the Evaluation of User Interfaces - Guidance on Usability (1997)
2. Ivory, M.Y., Hearst, M.A.: The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys* 33(4), 470–516 (2001)
3. Netcraft News, <http://news.netcraft.com/archives/2013/> (accessed, February 2013)
4. Nielsen, J.: *Designing Web Usability*. New Riders, California (2000)