

# MEANS OF AUTOMATING THE PROCESS OF EVALUATING THE QUALITY OF CITIZEN ORIENTED SOFTWARE SOLUTIONS USING SOFTWARE METRICS

**Emanuel Eduard HERȚELIU**

PhD Candidate, Doctoral School of Economical Informatics  
The Bucharest University of Economic Studies, Romania



**E-mail:** emanuel.herteliu@gmail.com

## **Abstract:**

*This paper presents means of evaluating the quality of citizen oriented software solutions using software metrics. Highlights about citizen oriented software solutions are presented. The need of automation is presented when dealing with a big number of informatics applications. The importance of choosing the right metrics that lead to meaningful results when dealing with citizen oriented informatics applications is presented. A sample of web based software solutions is used and the modality of getting and parsing the source code using automated tools is highlighted.*

**Key words:** quality, citizen oriented, software metrics, stability, evaluation

## **1. Introduction**

Software solutions provided by local government authorities are used on a daily basis in order to help citizens solve their administrative tasks. The quality of this type of informatics applications is very important because the computer skills of the users are unknown. Thus efforts must be made whilst developing citizens oriented solutions in order to assure a high level of quality complying to international software quality standards and providing a high utilizability level in the same time.

The variety of automated services provided to citizens is different from a government authority to another. A large number of web based solutions has been analyzed from the prospective of the quality of the services they provide for the citizens and the results show that there are improvements to be made in terms of what services they provide and how good the tools to implement the services are.

Because of the large number of solutions the need of automating the quality evaluation process had risen, this paper presents an approach towards automating this process.

## **2. Software metrics for citizen oriented web based solutions**

As the analyzed software solutions are citizen oriented choosing the right metrics is very important for obtaining meaningful quality evaluating results.

Two prospectives of evaluating the quality are approached:

- The quality of the service provided;
- The quality of the implemented software solutions which provide these services;

A sample of localities from Romania have been chosen and the websites provided by the local government authorities were analyzed in order to emphasize their quality level. The analysis showed that not all of the localities provided online services to citizens and this brings the need of extra governmental funding to correct this matter. The websites of the localities that provide online services were used in an automated quality evaluation process. This process speeds up the evaluation by crawling each individual website, parsing and analyzing the HTML code using software metrics. The metrics have been chosen to show whether a specific service exists or not on the analyzed web based solution. The services provided to citizens are:

- Tax paying;
- Administrative forms available for downloads;
- Informative services;
- Contact information on individual departments of the local government authorities;
- New events calendar schedule;
- Media content provided to citizens.

The metrics are presented in table 1.

**Table 1** – Web page metric for citizen oriented computer applications [1]

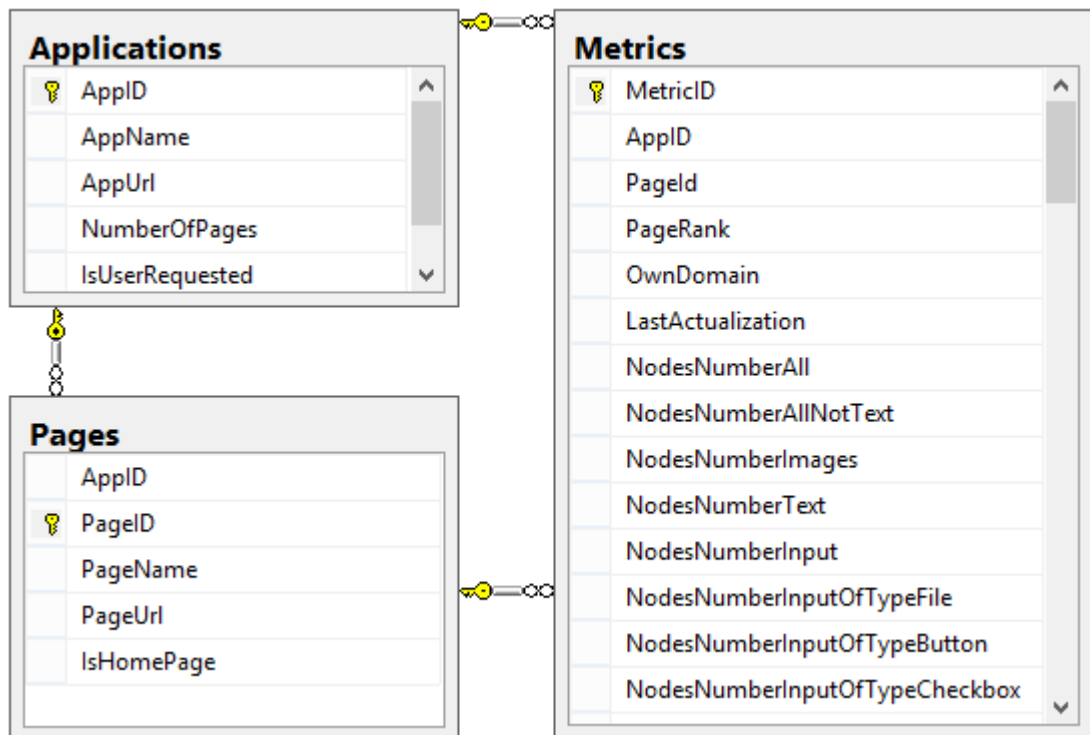
<b>Metric</b>	<b>Applications</b>	<b>Significance regarding to the service provided</b>
Total number of pages per application	Summing up all pages crawled for an app	The complexity of the problem solved by the application
The total number of nodes per page/app	Summing up all nodes parsed per page/app	The structure of page/application
Form nodes present on page/app	Searching for nodes of type <i>form</i>	Application is interactive
The weight of <i>input</i> nodes in the total number of app/page nodes	Dividing the total number of <i>input</i> nodes to the total number of nodes found on page/app	Application's extent of interactivity
Input nodes of type <i>file</i> present on page/app	Summing up the input nodes of type <i>file</i>	Application allows file uploads
Total number of files available for download per page/app	Summing up all files available for input per page/app	Application allows file downloads
Secure protocol present	Checking the protocol of app's URL	Application allows secure transactions

The metrics presented in table 1 are calculated based on real data crawled using the main URL of the websites corresponding to the localities included in the sample. Then each individual page is crawled and based on the number of the URLs crawled on the same domains the number of the pages for each website is calculated. Each page is searched for the specific HTML nodes presented in the first column of table one.

### 3. The automated quality analyzing process

The applications sample is built based on the homepage URL of the websites provided by the local government authorities of each locality. The information is stored in a database and one of the tables stores all of the analyzed URLs. Each individual URL is used during an iterative process. The homepage is crawled then out of the links found on the page the ones corresponding to the same domain are crawled as pages of the same website without including the links to pdf, document, image files and media content which are used to calculate the amount of the downloadable content provided to citizens.

The structure of saved information about the applications analyzed is presented in figure one.



**Figure 1** – Persistent information about the analyzed applications

The list of URLs is passed in an iterative process. The number of steps is equal to the number of applications analyzed. At the beginning about the app the only known information is the URL. A new application entity gets created as a pre phase of the process and an identification token is generated for it. This information is saved in the database inside Applications table. App name will be filled in after parsing the title of the homepage. When all pages of the website are completely crawled the number of pages is filled in. The pre phase of the process is finished as long as the application entity is created and successfully saved into the database. The table of applications has the purpose of storing generic information about the app as url, number of pages and name.

The first step of the automated quality analyzing process is crawling the homepage. Crawling involves a standardized procedure for all pages so at this step the app url previously saved in the database is used to create a page entity whose info gets stored in Pages table. In pages table the only difference between the homepage and other pages on

the same app is a boolean value which tells weather a specific page is the entry point of crawling process or not. Page name is filled in as soon as the title of the page is parsed.

At the second step of the process the crawler gets the html content of the page as a crawled page entity. This entity is not saved in the database but is used as an entry point for the third step of the process which involves parsing the content of the page and calculating counting metrics on the go while analyzing specific html nodes.

The third step is where parsing the content of the page and feeding metrics calculation modules with information takes place. The content is analyzed at a node level and specific nodes are counted. The nodes to be counted are chosen based on the importance to citizen oriented applications as stated in table one. The work of the third step is done by the parser module of the tool that was built particularly for this process. The parser consists of two specialized submodules: a links parser and a html nodes parser.

The first submodule is the links parser that transforms all the crawled urls from string structures to uniform resource identifier structures using Microsoft Visual Studio specific libraries. Further, specific url parts are analyzed in order to get link metrics based on the URL.

The metrics analyzed using the links parser are:

- a boolean metric that specifies if the used protocol is secure; this metric is meaningful for citizen oriented applications because it tells about how secure user data is and it tells about how risky is to use data of high importance on the specific application; when dealing with administrative software solutions all data filled in the forms, all the documents and media files need to be safe so using secure protocol is recommended [2];
- counting metrics that specify if the analyzed url points to a certain type of file; on citizen oriented informatics applications users should be able to download administrative forms in order to prefill them, informative and media files; the file extensions analyzed are categorized; documents: .pdf, .doc, .docx, .rtf, .xls, .xlsx; pictures and media files: .gif, .jpg, .png, .flv; presentations: .ppt, .pptx; the number of files of each type is counted as parsing all links found on the page;
- specific domain names that tell if the link found on the page points to a certain website; sites checked: Facebook, Twitter, YouTube, PayPal; the social network links found on a citizen oriented page tells about how easy is the information reachable; in the automated quality check the number of links social links is a metric that tells how easy people can find useful administrative information; the existence of paypal links is a way to tell weather the application supports paypal payments so users can pay for certain services online; the links are analyzed as being internal or external; the internal links are the ones that point to a page of the app and the external links point to external sources.

The second submodule of the parser module is the HTML nodes parser. The crawled content is divided into nodes using HtmlAgilityPack library and xpath expressions. This is done in order to count the number of specific nodes on the page which tells about the nature of the application.

The metrics analyzed with the nodes parser are usually counting metrics calculated as searching for specific node types and for certain nodes an attribute level counting is done. All the metrics are presented in table 2.

#### 4. Calculating the metrics

The presented steps of the automated quality evaluation process have been applied on a large number of apps. Table two presents raw results for a number of five apps out of them together with a total an average and a page level result.

**Table 2 – Web page metrics calculated for a set of citizen oriented informatics applications**

Metric	Ap 1	Ap 2	Ap 3	Ap 4	Ap 5	Total	Average	Per Page
PagesNumber	106	21	37	81	4	249	50	
NodesNumberAll	54292	6696	17429	13353	1423	93193	18639	374.269
NodesNumberAllNotText	26513	2511	7739	5832	812	43407	8681	174.325
NodesNumberImages	570	100	9	45	7	731	146	2.936
NodesNumberText	27779	4185	9690	7521	611	49786	9957	199.944
NodesNumberInput	302	3	158	3	0	466	93	1.871
NodesNumberInputOfTypeFile	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeText	104	1	42	3	0	150	30	0.602
NodesNumberInputOfTypeButton	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypePassword	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeSubmit	98	0	38	0	0	136	27	0.546
NodesNumberInputOfTypeRadio	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeCheckbox	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeNumber	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeEmail	0	1	0	0	0	1	0	0.004
NodesNumberInputOfTypeTime	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeDateTime	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeSearch	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeDate	0	0	0	0	0	0	0	0.000
NodesNumberInputOfTypeUrl	0	0	0	0	0	0	0	0.000
NodesNumberCanvas	0	0	0	0	0	0	0	0.000
NodesNumberIframe	0	1	0	0	0	1	0	0.004
NodesNumberForm	98	1	38	1	0	138	28	0.554
NodesNumberScript	532	109	259	287	0	1187	237	4.767
NodesNumberStyle	2	0	36	0	4	42	8	0.169
NodesNumberLink	285	80	218	369	5	957	191	3.843
NodesNumberDiv	8493	431	626	708	10	10268	2054	41.237
NodesNumberSpan	2052	14	712	47	153	2978	596	11.960
NodesNumberLabel	125	4	8	5	0	142	28	0.570
NodesNumberUI	1048	60	409	246	0	1763	353	7.080
NodesNumberLi	3909	481	1701	1353	0	7444	1489	29.896
NodesNumberComment	888	179	474	574	43	2158	432	8.667
NodesNumberProcessingInstruction	0	0	0	0	0	0	0	0.000
LinksNumberAll	4994	301	1444	1303	27	8069	1614	32.406
LinksNumberDoc	10	0	6	0	0	16	3	0.064

Metric	Ap 1	Ap 2	Ap 3	Ap 4	Ap 5	Total	Average	Per Page
LinksNumberDocx	0	0	0	0	0	0	0	0.000
LinksNumberGif	0	0	0	0	0	0	0	0.000
LinksNumberJpg	0	0	1	0	1	2	0	0.008
LinksNumberPng	0	0	0	0	0	0	0	0.000
LinksNumberPdf	585	8	32	0	21	646	129	2.594
LinksNumberRtf	0	0	0	0	0	0	0	0.000
LinksNumberFlv	0	0	0	0	0	0	0	0.000
LinksNumberPpt	0	0	0	0	0	0	0	0.000
LinksNumberPptx	0	0	0	0	0	0	0	0.000
LinksNumberXls	0	0	0	0	0	0	0	0.000
LinksNumberXlsx	0	0	0	0	0	0	0	0.000
LinksNumberFiles	595	8	39	0	22	664	133	2.667
LinksNumberDocs	10	0	6	0	0	16	3	0.064
LinksNumberPhotos	0	0	1	0	1	2	0	0.008
LinksNumberPresentations	0	0	0	0	0	0	0	0.000
LinksNumberExcel	0	0	0	0	0	0	0	0.000
LinksNumberInternal	54	15	37	29	4	139	28	0.558
LinksNumberExternal	4940	286	1407	1274	23	7930	1586	31.847
LinksNumberYouTube	0	0	0	0	0	0	0	0.000
LinksNumberFacebook	0	0	0	0	0	0	0	0.000
LinksNumberTwitter	0	0	0	0	0	0	0	0.000
LinksNumberPaypal	0	0	0	0	0	0	0	0.000
IsSecure	0	0	0	0	0	0	0	0.000

The table presents the results of crawling and parsing concreted in counting metrics for a set of five citizen oriented web based software solutions. The results have been calculated for all individual pages of the five apps and each "app" column presents the already aggregated results.

The first column holds the metric name. They are basically divided in two categories depending on the submodule used to parse them:

- links metrics: the links pointing to downloadable files categorized as: media files, documents and presentations; links pointing to social networking websites, links to paypal; the total number of links: external and internal links;
- html nodes metrics: the total number of nodes; the total number of text nodes; the total number of input nodes which tells about the level of interactivity; the input nodes are analyzed based on their "type" attribute as well as their number.

Following the results presented in table two on average each analyzed app has a number of fifty pages with a minimum of four pages for app five and a maximum of 106 pages for app one. This average value is quite high because the results for the entire sample shows that the administrative web based software solutions have a fewer number of pages.

On average each application has a number of 18639 html nodes with a corresponding nodes per page ratio of 374.26 nodes. A meaningful result is that out of the 374.26 nodes found per page 174.32 nodes are text nodes which corresponds to 46.57%.

This tells that almost half of the nodes on a page are text nodes. This value shows a high diversity of nodes for these citizen oriented applications as long as usually the text nodes rate is higher. Although text content is important, a big ratio of non-text nodes, in this case almost 50%, tells that user experience is enhanced by providing extra features other than just plain text.

It is considered that for a citizen oriented solution providing user with the possibility to input data and presenting him results based on his data brings flexibility and dynamism to the human-computer interaction process [3][4]. Out of all the apps on app number five no input nodes were found which makes it categorized as a static app. Within the citizen oriented metrics calculated the number of downloadable files provided to citizen counts when analyzing the interactivity level of the page but for app number five the downloadable files number is zero. At the other end app number one has an absolute value of 302 input nodes but although app number three has an absolute value of 158 input nodes at a page level the ratio is better for app three (4.27 input nodes per page) then app one (2.84 input nodes per page). On average on the five apps have 1.87 input nodes per page so the assumption can be made that results are displayed based on citizen requests. This is because none of the apps have input nodes of type password which means that users cannot create accounts on the websites so the almost 2 input nodes per page are for different kind of interaction than logging in.

Analyzing the metrics calculated using the link parser module the last row of table two highlights that none of the apps has secure protocol pages. The same value was obtained for PayPal links so most probably none of the apps supports online payments.

On average 28 internal links per app were found and 1586 links pointing to external sources per app. This leads to 0.5 internal links per page. The internal links number tell about how easy the user can navigate from a section of the website to another and the higher the number the quicker the access to other pages. The value being under one means that on average users can navigate to maximum one different page on the same app which decreases application's usability level [5]. The external links number have a better ratio at a page level with 31.8 links per page pointing to an external resource.

Informatics applications provided by administrative authorities should provide downloadable useful content to citizens. On average each one of the analyzed apps provides to citizens 133 files for download. This average value is very high because of app number one which provides a total of 595 files for download but app number four provides no files and the rest of them under fifty. Out of the total of 664 files provided by all five apps 646 are .pdf files 4 photos and 16 .doc files.

None of the applications provides the feature of user file upload as long as the count for input nodes of type file is 0. Although the downloadable content exists at a good rate the upload feature doesn't. This highlights the fact that no services that require citizen scanned documents or forms are implemented.

## **5. Conclusions**

This paper presents a way to evaluate the quality of the services provided by a big number of citizen oriented web based software solutions using automated tools. The analyze is done based on the content the local governance authorities expose to the public via their websites. The chosen metrics are meaningful for evaluating how good the services provided



to citizens are. Further work is to be done in order to be able to get information about how users actually interact with the applications and correlate it with the results obtained using the metrics. This implies using analytics services at owners initiative and shared information about what the services measure.

### Bibliography

1. Emanuel Eduard HERȚELIU, **Quality Metrics for Evaluating Interactivity Level of Citizen Oriented Web Based Software Solutions**, The 14th International Conference on Informatics in Economy, Research & Business Technologies, Bucharest Academy of Economic Studies, 30 April, 2015, Bucharest, Romania
2. Sai Peck LEE, Malik HNEIF, **Using Guidelines to Improve Quality in Software Nonfunctional Attributes**, **IEEE Software**, Vol. 28, Issue 6, November/December 2011, pp. 72-77
3. Ion IVAN, Bogdan VINTILĂ, Dragoș PALAGHIȚĂ, **Types Of Citizen Orientated Informatics Applications**, Open Education Journal, Russia, No.6, 2009
4. Stephen KAN, **Metrics and Models in Software Quality Engineering**, Publisher: Addison Wesley, Boston, 2002, pp. 560
5. Robert BAGGEN, José Pedro CORREIA, Katrin SCHILL, Joost VISSER, **Standardized code quality benchmarking for improving software maintainability**, Software Quality Journal, Vol. 20, Issue 2, June 2012, pp. 287-307