A NEW DJANGO WEB FRAMEWORK SOFTWARE METRICS MEASUREMENT USING RADON AND PYLINT

Guided By:Dr.J.RAJA RAM

Assistant Professor, Dept.of CSE, **MALLA REDDY ENGINEERING COLLEGE FOR WOMEN**, TELANGANA, India.

MD.AYESHA FATHIMA BEGUM[1], P.SINDHU[2], M.CHANDANA[3], M.HEBSHEEBA[4]

B.Tech Pursuing, Dept.of CSE, **MALLA REDDY ENGINEERING COLLEGE FOR WOMEN**, TELANGANA, India.

Abstract:

Nowadays, sites are complex applications that carry out transactions, render real-time information and offer interaction. Creation of web applications that allow many designers, advanced tools, and many more options. Web frameworks deliver a brilliant way between creating an application from the ground and using a content management system. This article centers around an open source web development framework, to be specific to Django. The methodology that used in the study is measuring Django Web Frameworks code quality metrics using Radon and Pylint. Django option in the main directory has 2,200 lines of code, Cyclomatic Complexity score is 16.375 considered as average complexity, and 6.69/10 by the pylint score.

INDEX TERMS: Complex Applications,Content Management System,Radon And Pylint

Introduction:

Nowadays, in a fast-paced, very demanding and competitive era people tend to use a web application framework to build their web-enabled applications. Software frameworks used to establish web application, and website development, web services, and web resources are called Web application frameworks. A favorite type of web app framework is the Model-View-Controller (MVC) design separates the code for each application component into modules [1], [2]. Python gains its popularity since 2016 and become number 1. Together with Python, C, Java, and C++ remain in the top four in popularity; this paper focuses on Python web frameworks [3]. The Python

web frameworks are full of options, such as Django framework is a compact Python framework intended for quick development in quickly paced environments that function well with relational databases [4]. Flask framework is a Python micro-framework with a moderate approach and powerful by itself. Flask perfect for stand-alone applications and rapid prototyping. Pyramid known as Pylons is a framework that gives reproducibility with NoSQL integration. Pyramid best for APIs, prototyping, and extensive web apps, like content management systems development [5], [6]. Tornado is an event-based, nonblocking Python web server and web application framework serve high-traffic volume [7]. The Bottle is a light and simple microframework [8]. Django is chosen because of its popularity and widely used. In this article, we will find out software metrics for Django using Radon that provides cyclomatic complexity raw metrics that consists SLOC, comment lines, number blank lines, Maintainability Index and Halstead metrics, also use pylint [9], [10]. Pylint is a source code analyzer that finds for errors in programming, assists to use a coding standard strictly.

‘

LITERATURE SURVEY:

1)Designing an MVC model for rapid web application development. Procedia Engineering AUTHORS: D. P. Pop and A. Altar. (2014).

In this paper, we present a model for rapid web application development. This model is based on the Model-View-Controller architecture (MVC) and has several other useful components like security, form generation and validation, database access and routing. This model was implemented using the PHP programming language, but it can be implemented in other development languages and environments using the same concepts. Improvements in both development and maintenance time have been the main objectives of this research, with the added benefit of correct and maintainable code.

2)Code smells for Model-View-Controller architectures

AUTHORS: M. Aniche, G. Bavota, C. Treude, M. A. Gerosa, and A. van Deursen

Previous studies have shown the negative effects that low-quality code can have on maintainability proxies, such as code change- and defect-proneness. One of the symptoms of low-quality code are code smells, defined as sub-optimal

implementation choices. While this definition is quite general and seems to suggest a wide spectrum of smells that can affect software systems, the research literature mostly focuses on the set of smells defined in the catalog by Fowler and Beck, reporting design issues that can potentially affect any kind of system, regardless of their architecture (e.g., Complex Class). However, systems adopting a specific architecture (e.g., the Model-View-Controller pattern) can be affected by other types of poor practices that only manifest themselves in the chosen architecture. We present a catalog of six smells tailored to MVC applications and defined by surveying/interviewing 53 MVC developers. We validate our catalog from different perspectives. First, we assess the relationship between the defined smells and the code change- and defect-proneness. Second, we investigate when these smells are introduced and how long they survive. Third, we survey 21 developers to verify their perception of the defined smells. Fourth, since our catalog has been mainly defined together with developers adopting a specific Java framework in their MVC applications (e.g., Spring), we interview four expert developers working with different technologies for the implementation of their MVC applications to check the generalizability of our catalog. The achieved results show that the defined Web MVC smells (i) more often than not, have more chances of being subject to changes and defects, (ii) are mostly introduced when the affected file (i.e., the file containing the smell) is committed for the first time in the repository and survive for long time in the system, (iii) are perceived by developers as severe problems, and (iv) generalize to other languages/frameworks.

3) A Low Effort Analytics Platform for Visualizing Evolving Flask-Based Python Web Services

AUTHORS: P. Vogel, T. Klooster, V. Andrikopoulos, and M. Lungu

Tens of thousands of web applications are written in Flask, a Python-based web framework. Despite a rich ecosystem of extensions, there is none that supports the developer in gaining insight into the evolving performance of their service. In this paper, we introduce Flask Dashboard, a library that addresses this problem. We present the ease with which the library can be integrated in an already existing web application, discuss some of the visualization perspectives that the library provides and point to some future challenges for similar libraries.

4) Code Quality: Examining the Efficacy of Automated Tools

AUTHORS:Sara Hooshangi The George Washington University shoosh@gwu.edu Subhasish Dasgupta The George Washington University dasgupta@gwu.edu

A number of software tools have been designed to determine code quality. These automated tools can examine a computer program and provide a score based on a designed set of software metrics. In this exploratory study, we examined the quality of code written by students in an introductory Python programming course on one programming assignment. Each student submission was evaluated using an automated tool, Pylint, a code analyzer widely used by the Python community. The instructor also graded these submissions using predefined rubrics that evaluated code logic, syntax and style. We compared the two code quality scores. We found that Pylint does a good job of identifying errors, and formatting issues. But, the Pylint scores were lower than those provided manually by instructors.

5)OpenHub: A Scalable Architecture for the Analysis of Software Quality Attributes

AUTHORS: G. Farah, J. S. Tejada, and D. Correal,

This paper portrays a utilization of Maurice Halstead's product hypothesis' effect on modern programming languages like Python. The Halstead Metric and the product apparatus created for registering them are examined. Investigation of the metric information demonstrates that the level of the exchanging dialect was not consistent crosswise over algorithms and that product blunder information was not a straight capacity of volume. We also benchmark Halstead Metric against COCOMO II with respect to the accuracy of effort estimation.

EXISTING SYSTEM:

Flask framework is a Python micro-framework with a moderate approach and powerful by itself. Flask perfect for stand-alone applications and rapid prototyping. Pyramid known as Pylons is a framework that gives reproducibility with NoSQL integration. Pyramid best for APIs, prototyping, and extensive web apps, like content management systems development. Tornado is an event-based, non-blocking Python web server and web application framework serve high-traffic volume. The Bottle is a light and simple micro framework.

DISADVANTAGES OF EXISTING SYSTEM:

- There will be times as a software developer when a programmer works on projects that do not have unit tests, a daily build, a mess of legacy spaghetti code
- Lines of code (LOC) are the most simple indicators and warnings. A massive, complicated project will have hundreds of thousands of LOC and that normal if the project is well-managed.

PROPOSED SYSTEM:

Django Reinhardt is used as the framework name Django is a framework for web development using python and created for the quick development of database driven sites. Inside Django is MVC-style, a high-level, open source bunch of libraries programmed in Python. Django is the most popular server-side web frameworks. Don't repeat yourself is Django's motto. As much as Python, it focuses inefficiency, giving developer to do almost all tasks with as little coding effort as possible.

ADVANTAGES OF PROPOSED SYSTEM:

- LLOC The variety of logical strains of code. Each logical line of code contains precisely one statement.
- SLOC The quantity of source lines of code. It does not relate to the LLOC.
- LOC The aggregate number of lines of code. It doesn't relate to the number of lines in the file.
- Multi The number of lines that represent multiline strings.

Comments The number of comment lines. Multiline strings are not considered comment, to the Python, they are just strings.

System Architecture:

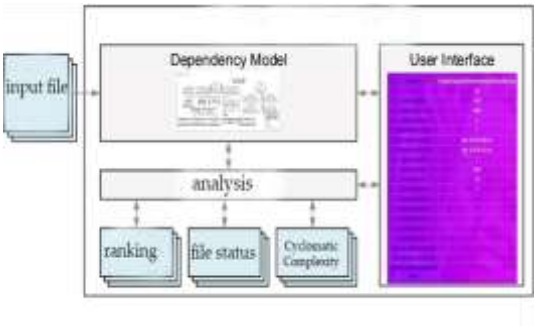
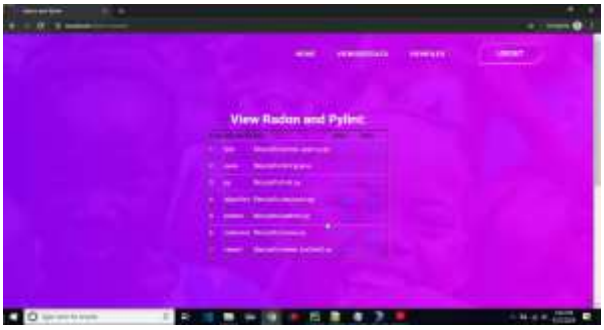
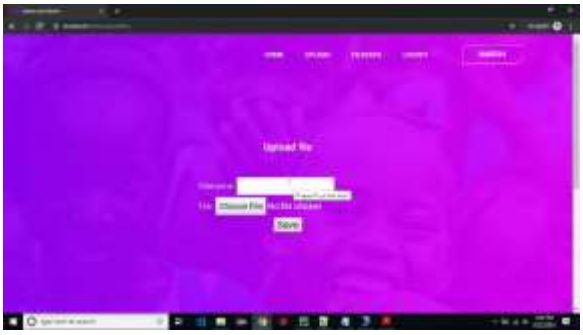


Fig:System Architecture

Result:



Here we are verify Cyclomatic Complexity through code file



Here we have to upload code file and save the file

Here finnaly we have to identify the Cyclomatic Complexity score by sequentially

Conclusion:

In conclusion, this paper has measure Django Web Framework code quality metrics. Django option in the main directory has 2,200 lines of code, Cyclomatic Complexity score is 16.375 considered as moderate complexity, and 6.69/10 by the pylint score. With the same method, other web frameworks can be measured too.

References:

[1] D. P. Pop and A. Altar. (2014). Designing an MVC model for rapid web application development. Procedia Engineering, 69, 1172-1179.

[2] M. Aniche, G. Bavota, C. Treude, M. A. Gerosa, and A. van Deursen, "Code smells for Model-View-Controller architectures," Empir. Softw. Eng., vol. 23, no. 4, pp. 2121–2157, 2018.

[3] S. Cass. (2017). The 2017 top programming languages. https://spectrum.ieee.org/computing/software/the-2017-topprogramming-languages. Accessed 18 October 2017.

[4] A. Pelme. (2014). Evaluation of a web application architecture. (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn: nbn:se:liu:diva-107011

[5] R. Brown. (2015). Django vs. flask vs.

pyramid: Choosing a python web framework. Recuperado el, 31.

[6] P. Vogel, T. Klooster, V. Andrikopoulos, and M. Lungu, "A LowEffort Analytics Platform for Visualizing Evolving Flask-Based Python Web Services," in 2017 IEEE Working Conference on Software Visualization (VISSOFT), 2017, pp. 109–113.

[7] Anonym. Tornado Python framework. http://www.tornadoweb.org/en/stable/

[8] Hellkamp, M. (2017). Bottle: Python Web Framework. Available: https://bottlepy.org/docs/dev/. Accessed 28 October 2017.

[9] S. Dasgupta and S. Hooshangi, "Code Quality: Examining the Efficacy of Automated Tools," AMCIS 2017 Proc., Aug. 2017.

[10] G. Farah, J. S. Tejada, and D. Correal, "OpenHub: A Scalable Architecture for the Analysis of Software Quality Attributes," in Proceedings of the 11th Working Conference on Mining Software Repositories, 2014, pp. 420–423.

[11] Y. Saleh. (2017). How does the django framework in python overcome the frameworks in PHP? https://www.egrovesys.com/blog/how-does-the-django-framework-inpython-overcome-the-frameworks-in-php/ Accessed 24 November 2017.

[12] M. P. Malhotra and M. K. Shah. Python Based Software for Calculating Cyclomatic Complexity. (2015). International Journal of Innovative Science, Engineering and Technology, 2(3):546–549, March 2015.

[13] M. del Pilar Salas-Zárate, G. Alor-Hernández, R. Valencia-García, L. Rodríguez-Mazahua, A. Rodríguez-González, and J. L. L. Cuadrado (2015). Analyzing best practices on Web development frameworks: The lift approach. Science of Computer Programming, 102, 1-19.

[14] N. Gift. (2017). Writing clean, testable, high-quality code in python. https://www.ibm.com/developerworks/aix/library/au-cleancode/ Accessed 19 December 2017.

[15] C. Thirumalai, R. R. Shridharshan, and L. R. Reynold, "An assessment of halstead and COCOMO model for effort estimation," in 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), 2017, pp. 1–4.

[16] Anonymous. (2017). Radon. https://pypi.python.org/pypi/radon Accessed 30 November 2017.

[17] A. Calleja, J. Tapiador and Caballero, J. (2016, September). A look into 30 years of malware development from a software

metrics perspective. In International Symposium on Research in Attacks, Intrusions, and Defenses, pp. 325-345. Springer, Cham.

[18] T. J. McCabe. (1976). A complexity measure. IEEE Transactions on Software Engineering, SE-2(4):308–320, Dec 1976

[19] R. Smith, T. Tang, J. Warren and S. Rixner. (2017, June). An Automated System for Interactively Learning Software Testing. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, pp. 98-103.

[20] Anonymous. Pylint. https://www.pylint.org Accessed 12 December 2017. T. J. McCabe. A complexity measure. IEEE Transactions on Software Engineering, SE-2(4):308–320, Dec 1976.

AUTHOR

Dr.J.Raja ram Professor Department of CSE MallaReddy Engineering College for Women, Hyderabad, drjrajaram81 @gmail.com