International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 2nd Jan 2021. Link :http://www.ijiemr.org/downloads.php?vol=Volume-09&issue=ISSUE-12

DOI: 10.48047/IJIEMR/V09/I12/162

Title: **DJANGO WEB FRAMEWORK SOFTWARE METRICS MEASUREMENT USING RADON AND PYLINT**

Volume 09, Issue 12, Pages: 949-953

Paper Authors

N.L.V.S. HARITHA, BAINDLA AMALA, V. MANASA, T.KANAKAIAH

International Journal for Innovative Engineering and Management Research ★

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DJANGO WEB FRAMEWORK SOFTWARE METRICS MEASUREMENT USING RADON AND PYLINT

N.L.V.S. HARITHA[1], BAINDLA AMALA[2], V. MANASA[3], T.KANAKAIAH[4]

[1,2,3,] B TECH Students, Department of CSE, Princeton Institute of Engineering & Technology For Women, Hyderabad, Telangana, India.

[4] Assistant Professor, Department of CSE, Princeton Institute of Engineering & Technology For Women, Hyderabad, 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 on 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.

Keywords: Cyclomatic Complexity, Maintainability Index, Software Metrics, Python Web Framework

I. 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.

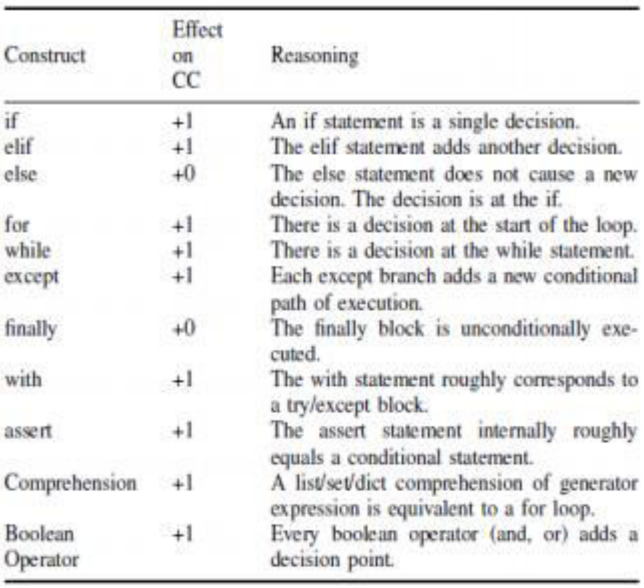
II. Related data

Django Web Framework: In the fall of 2003, Django was created at the Lawrence Journal-World newspaper. Adrian Holovaty and Simon Willison started using Python to create web applications. In July 2005, Django was released under a BSD license for the public. Django is used by The National Aeronautics and Space Administration (NASA), Google, Public Broadcasting Service (PBS), Bit-Bucket, and other newspapers. Guitarist 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 [11]. Inside Django is MVC-style, a high-level, opensource 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. Furthermore, Django is also scalable, mature, and fast with a numerous community of developer and a robust set of built-in components. Django can access or create JSON or XML data instances and deal with out-of-the-box with relational database management systems such as Oracle, MySQL, SQLite, and PostgreSQL. At deployment, Django is fully backed up by the cloud platform AWS Elastic Beanstalk and Heroku. Three items make Django excel are its maturity, structure, and community. In contrast to Flask and Pyramid, Django stresses on getting started right out of the box using particular modules. In other words, Djangos bundled applications and modular components can reduce developer time when he/ she was trying to make a web application ready and functioning. Data science is a buzzword nowadays, and multitalented developers are needed more than ever before. A simple preparation in building apps with a framework as powerful, minimalistic, and easy-to-learn as Django will be an essential skill when launching their career as the web developer or an entrepreneur. B. Software Complexity Software complexity is defined as the tool to measure the difficulty level of any software. Software complexity is a branch of software metrics which is concentrated on direct measurement of software qualities, as disparate to additional software trials such as testified system failure and project innovative status. There are many software complexity trials, fluctuating from the simplest, e.g., lines of code, to the complex, e.g., the number of usage associations or variable definition [12]. It is the calculation of efforts required for the development, maintenance, and execution of software code. As the complexity of the code increases the time and effort also required increases. 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. His/ her goal is to control that legacy project, and the only way for any project to get back under control is to have metrics and to measure how the code base fares and then to keep weekly (even daily), a track of those metrics and monitor how they are varying. 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. Any metrics are better than none. At the point when first beginning work on the legacy project developer require a baseline to recognize what sort of code he/ she has acquired. The baseline will enable the developer to know whether he/ she moves towards viable excellent code which encourages rapid advancement or whether he/ she fails towards slower improvement with messier code. If the project is in excellent condition, the baseline will be high enough, and at one point any changes or bug fixes will have reducing returns. In the other hand, if the project is in a dangerous condition, the baseline will be lower, and he/ she can make sure that every change he/ she make is giving essential benefits on finishing the project.

III. Mccabe's Cyclomatic Complexity

McCabe's Cyclomatic Complexity metric is applied to discover an upper bound on the model for rating the number of defects. Complexity is equated with the range of selections a software makes so that overly elaborate sections can be recoded. McCabe's metric provides the upper bound on module complexity. The graph and formula below define Cyclomatic Complexity [18], [19]. The Abstract Syntax Tree (AST) tree of a Python program to compute Cyclomatic Complexity is analyzed by Radon. As shown in Table 2 commands that have the effects on Cyclomatic Complexity. Cyclomatic complexity calculates the number of decision logic in a software module, based on the structure of the software control flow graph. It is measured using the control flow graph of the program: nodes of the graph match to separabwhole groups of statements of a program, and an engaged edge joins two nodes if the second statement is probably executed immediately after the first statement. Cyclomatic complexity may likewise be pragmatic to discrete components, functions, classes or methods within a package or a program.

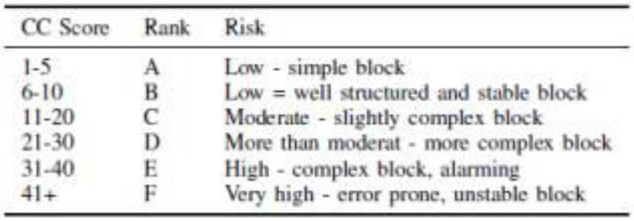
Table 1: **Cyclomatic Complexity**



Construct	Effect on CC	Reasoning
if	+1	An if statement is a single decision.
elif	+1	The elif statement adds another decision.
else	+0	The else statement does not cause a new decision. The decision is at the if.
for	+1	There is a decision at the start of the loop.
while	+1	There is a decision at the while statement.
except	+1	Each except branch adds a new conditional path of execution.
finally	+0	The finally block is unconditionally executed.
with	+1	The with statement roughly corresponds to a try/except block.
assert	+1	The assert statement internally roughly equals a conditional statement.
Comprehension	+1	A list/set/dict comprehension of generator expression is equivalent to a for loop.
Boolean Operator	+1	Every boolean operator (and, or) adds a decision point.

McCabe number is equivalent to the quantity of linearly independent paths throughout the code. This number can be utilized as a guide when testing conditional logic in blocks. Every block will be positioned from A as best complexity score to F as worst complexity score. Ranks correspond to complexity scores as shown in Table 1.

Table 2: **Cyclomatic Complexity Rank**



CC Score	Rank	Risk
1-5	A	Low - simple block
6-10	B	Low = well structured and stable block
11-20	C	Moderate - slightly complex block
21-30	D	More than moderat - more complex block
31-40	E	High - complex block, alarming
41+	F	Very high - error prone, unstable block

Pylint: Pylint is a Python instrument which verifies a module for coding standards. As indicated by the TurboGears project coding guidelines, PEP8 is the standard and pylint is a decent mechanical test to help the developer in accomplishing the goal. The scope of verifies run from Python errors, unused imports, missing docstrings, an unintended redefinition of built-ins, to awful naming and others. Following the analysis message, Pylint can show a group of reports, everyone concentrating on a specific part of the project, for example, number of messages by categories and modules dependencies. These highlights can be enabled via the – reports=y parameter, or –rn.

IV. 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.

V. 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..