

Regression Testing

Date: March 29, 2021
Group Number 6

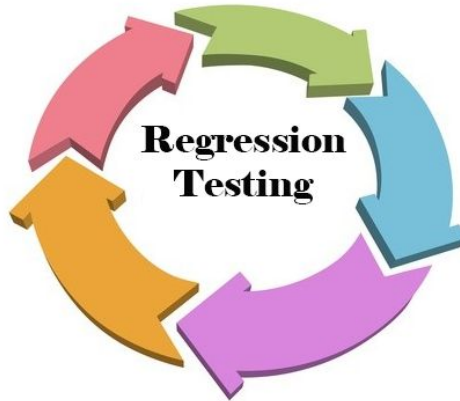
Presenters:
Jack McLeod
Krishna Brahmbhatt
Lyndon Renaud

Outline

- Introduction
- Approach / Method
- Results
- Discussion
- References

Introduction

Regression Testing is used to check if a change in functionality of a software has caused the software to regress. This means the goal is to ensure that the new version of the software works the same as the old except for where new behaviour is intended.



Importance of Regression Testing

It is extremely important that after a modification to a program that regression testing is done. This is because small modifications to a program may affect unmodified parts of the program.

New Changes May Cause:

- Correct outputs for new cases.
- Incorrect outputs for previously correctly outputting cases.
- Build up of failed test cases over time.

Objectives

- Build a small web application (Bank application).
- Implement regression testing framework.
- Create cases that give a variety of correct and incorrect outputs for testing.
- Make automated tests to run for further iterations of the web application.
- Document all passing and failing cases for results.

Method (steps)

- Create base web application functionality using Django and host it on Heroku.
- Apply testing to the application through Django and Selenium.
- Add functionality changes of more than 1 of the three types from the literature (to be explained).
- Apply the Regression testing framework in our testing approach for all new iterations of the application.

Tools

Django - Web application creation, Run Tests

Selenium - Testing tool

GitLab - Store tests and Web app, create issues.

GitLab CI/CD - Configures and runs automated testing and deployment

Docker - Run GitLab Runner container for automated testing

Heroku - Host web application

Test plan

We ran our tests when a change was made based on the following modification types found in the literature[3]:

Type 1: Integrating a newly established web service into the application.

Type 2: Adding, removing or fixing an operation or a timing constraint in an existing component.

Type 3: Modifying the specification (operations or timing constraints) of the web application.

Test Cases

- Deposit Integer numbers.(Type 1)
- Withdraw Integer numbers. (Type 2)
- Deposit/Withdraw Double numbers rounded to two decimal places. (Type 3)
- Create a Register page for user creation. (Type 1)
- Create a Login page for the new created user to login to. (Type 2)

Eliminate Redundant/Unused Tests

Algorithm from [3]

Consider T an old set of tests and T' the new set of tests:

- All test sequences found in T' and not found in T are executed, that is, added to the retest set T' .
- All test sequences found in T and not found in T' are deleted from the test history.
- All test sequences found in both T and T' are kept in the test history but not re-executed.
- Add test sequences in T' to the test history.

Test Management Approach

GitLab - CI/CD pipeline in gitlab

<https://gitlab.cs.uwindsor.ca/renau11s/4110-project>

Gitlab Issues - Report missing features and bugs.

Discord - Communication among team members

Automated Regression Testing

- A Docker container containing the test environment is created each time GitLab regression tests are ran. GitLab uses this container for test execution.
- GitLab regression tests are triggered on each merge or tagged commit to the master branch.
- Django test coverage is displayed in GitLab after each Django test job.
- GitLab deploys the web app to the Heroku staging branch if all tests pass
- If an HTTP request to the Heroku staging site return HTTP code 200, then GitLab deploys the web app to the Heroku production branch.
- If an error occurs during automated testing, GitLab displays an error, sends an error email to the developers in the repo, and does not deploy the new build to Heroku.

Results(Thoughts on the testing tools)

Django: Django provides a set of tools to aid in testing the Django app. By using Django's test API, we were able to test form functionality, page contents, and authentication system, while having control over the database contents at each test. One of the main limitations of Django's test API is that it is unable to test a web page interactively.

Selenium

Positive:

Selenium Ide has an easy to use interface for testing web applications. The tests can easily be converted into java code which helped for automating the testing.

Negative:

It cannot perform database testing. It cannot provide a detailed test report.

Results - Deposit/Withdraw

Bank Account

Account Balance: \$100.00

DepositAmount:	<input type="text"/>	<input type="button" value="Submit"/>
WithdrawAmount:	<input type="text"/>	<input type="button" value="Submit"/>

Deposit History

\$100.00

\$10.00

\$390.00

\$100.00

\$10.00

\$390.00

\$100.00

\$10.00

\$390.00

\$100.00

\$10.00

\$390.00

\$100.00

\$10.00

\$390.00

Withdrawal History

\$400.00

\$400.00

\$200.00

\$500.00

\$400.00

Results - Login/Register Pages

Register

Username:

Required: 150 characters or fewer. Letters, digits and @/./+/-/_ only. Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Login

Username: Password:

Results (Our tests)

Type 1: Add login functionality

Regression test results after implementing login functionality:

Django tests:

Name	Stmts	Miss	Cover	Missing

Bank/__init__.py	0	0	100%	
Bank/admin.py	4	0	100%	
Bank/apps.py	3	0	100%	
Bank/forms.py	21	0	100%	
Bank/models.py	12	0	100%	
Bank/tests.py	40	0	100%	
Bank/urls.py	4	0	100%	
Bank/views.py	26	0	100%	

TOTAL	110	0	100%	

Selenium tests:

```
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 5.758 sec
Results :
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO] -----
[INFO] BUILD SUCCESS
```


Results (Our tests)

Type 2: Provide an alert when given invalid input to the deposit or withdraw form.

Selenium Tests fail since the alert isn't implemented.

```
Results :
Tests in error:
    DnegativeValueErrorCheck(TestSuiteTest): no such alert(..)
Tests run: 4, Failures: 0, Errors: 1, Skipped: 0
```

After implementing the alert, the Selenium Tests pass and the app is deployed.

```
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 5.757 sec
Results :
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO] -----
[INFO] BUILD SUCCESS
```

Discussion

- Due to the fact that the application that we created was not very large in size we can not assume that our test method would work well on a very large application.
- In regression testing we aim to achieve maximum test coverage with the creation of limited test cases. This becomes difficult when we have stringent timelines.
- Any problems we had with the tools we used. (Database testing, Error handling)
- Tester needs to be proficient on the automation test techniques, selection of tool and creation of automation test scripts, lack of automation expertise in the team can lead to a bad automated regression testing.

References

1. G. Rothermel, R.H. Untch, C. Chu, and M.J. Harrold, "Prioritizing test cases for regression testing," *IEEE Transactions on software engineering*, pp.929-948, 2001. Available:IEEE Xplore, [online document], <https://ieeexplore.ieee.org/abstract/document/962562>. [Accessed Feb. 2, 2021].
2. H. Agrawal, J. Horgan, E. Krauser, and S. London, "Incremental Regression Testing," *Proc. Conf. Software Maintenance-1993*, pp. 348- 357, Sept. 1993. Available:IEEE Xplore, [online document], <https://ieeexplore.ieee.org/abstract/document/366927>. [Accessed Feb. 2, 2021].
3. A. Tarhini, H. Fouchal, and N. Mansour, "Regression testing web services-based applications," *In Proc. International Conference on Computer Systems and Applications*, pp 163–170, 2006. Available: Lebanese American University, [online document], <https://laur.lau.edu.lb:8443/xmlui/handle/10725/7858>. [Accessed Feb. 2, 2021].
4. S. Yoo and M. Harman, "Regression testing minimization, selection and prioritization: a survey," *Wiley Online Library*, 11-Mar-2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stvr.430?casa_token=mLMYGAgQYeMAAAAA%3AQeRaohldRR7R2xl4LxtT9plcO410_yBXITOKN1imVcb7oWfLLozfq6CB699_hl1LwMj8ls7Wkqv31r4. [Accessed: 02-Feb-2021].
5. W. E. Wong, J. R. Horgan, S. London, and H. Agrawal, "A Study of Effective Regression Testing in Practice," <https://www.researchgate.net>. [Online]. Available: https://www.researchgate.net/profile/Hira_Agrawal/publication/3719734_Study_of_effective_regression_testing_in_practice/link/s/0deec517066a5a3cb6000000/Study-of-effective-regression-testing-in-practice.pdf. [Accessed: 02-Feb-2021].

References

6. T. L. Graves, M. J. Harrold, J.-M. Kim, A. Porter, and G. Rothermel, "An empirical study of regression test selection techniques," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 01-Apr-2001. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/367008.367020?casa_token=opDoIW_Ih40AAAAA%3Ad7LjUNS_Dq687fmB7tBlbDy8yzqf1RQCqnlQxVgonaRHjZwRMFBQqGPrw5XRt7dEvFMfeUCdy82A. [Accessed: 02-Feb-2021].
7. T. M. Ahmed, C.-P. Bezemer, T.-H. Chen, A. E. Hassan, and W. Shang, "Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web applications: an experience report," *Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web applications | Proceedings of the 13th International Conference on Mining Software Repositories*, 01-May-2016. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/2901739.2901774?casa_token=EtHkeXKglCYAAAAA%3Ai7zhdXDp1QNTuh2jcUA30YFuwTbC7NBA5A742a5wLxZrbR3xk4rBK32u9A63hx0tK7woKJJ8SOMG. [Accessed: 02-Feb-2021].
8. R. Gupta, M.J. Harrold, and M.L. Soffa, "An Approach to Regression Testing Using Slicing," *Proc. Conf. Software Maintenance* 1992, pp. 299-308, Nov. 1992. Available: Citeseer, [online document], <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.25.6155&rep=rep1&type=pdf> . [Accessed Feb. 2, 2021].
9. D. Roest, A. Mesbah, and A.v. Deursen, "Regression testing ajax applications: Coping with dynamism," in *Software Testing, Verification and Validation (ICST)*, 2010 Third International Conference, pp. 127 –136, April 2010. Available: IEEE Xplore, [online document], <https://ieeexplore.ieee.org/abstract/document/5477090>. [Accessed Feb. 2, 2021].
10. S. Yoo and M. Harman, "Regression testing minimisation, selection and prioritisation: A survey," *Softw. Testing, Verification, Rel.*, vol. 22, no. 2, pp. 67–120, Mar. 2012. Available: Wiley Online Library, [online document], <https://onlinelibrary.wiley.com/doi/abs/10.1002/stvr.430>. [Accessed Feb. 2, 2021].

Youtube Link

<https://www.youtube.com/watch?v=oDKWFDx62Nw>