**Design**

**On**

**Diff Set Based Automation**

**SUBMITTED BY**

**PRAVEENGOUDA S P 1MS12CS076**



**M. S. Ramaiah Institute of Technology**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

**Department of Computer Science & Engineering**

***Under the guidance of***

|  |  |  |
| --- | --- | --- |
| **Rohan Pandit**  **Technical Leader**  **Cisco Systems, Inc.**  **Bangalore** | **&** | **Chandrika Prasad**  **Assistant Professor**  **Department of Computer Science & Engineering** |

1. **Introduction**
   1. Description of the organization of the document

[Section 1.2](#numberofmodules): Number of Modules- Total number of modules

[Section 1.3](#modulesdescription): Modules Description- Description of the operations performed by each module

[Section 1.4](#algorithmdesign): Algorithm Design- Provides an algorithm for automated patch mechanism

[Section 2](#architecturedesign): Architecture Design- Design of the entire automated patch software system. It provides information on the decomposition of the system into modules.

[Section 3](#dataflowdiagram): Data flow diagram- Shows the flow of data through the software system.

[Section 4](#references): References- List of books, URLs, tools consulted and used to design the document.

* 1. Number of Module: 4
  2. Modules description
* *Identify the diff set which are changed on the on the developer’s system:*

The application takes code coverage files as a list of modified files and converts it to JSON format. It finds the test suites that use the modified file and adds them to the database.

* *Identify the files which are not changed in the database:*

The application identifies all those files which are not changed and ignores when it is looking for changed files and corresponding test suites.

* *All the code coverage files are converted to json for input to database:*

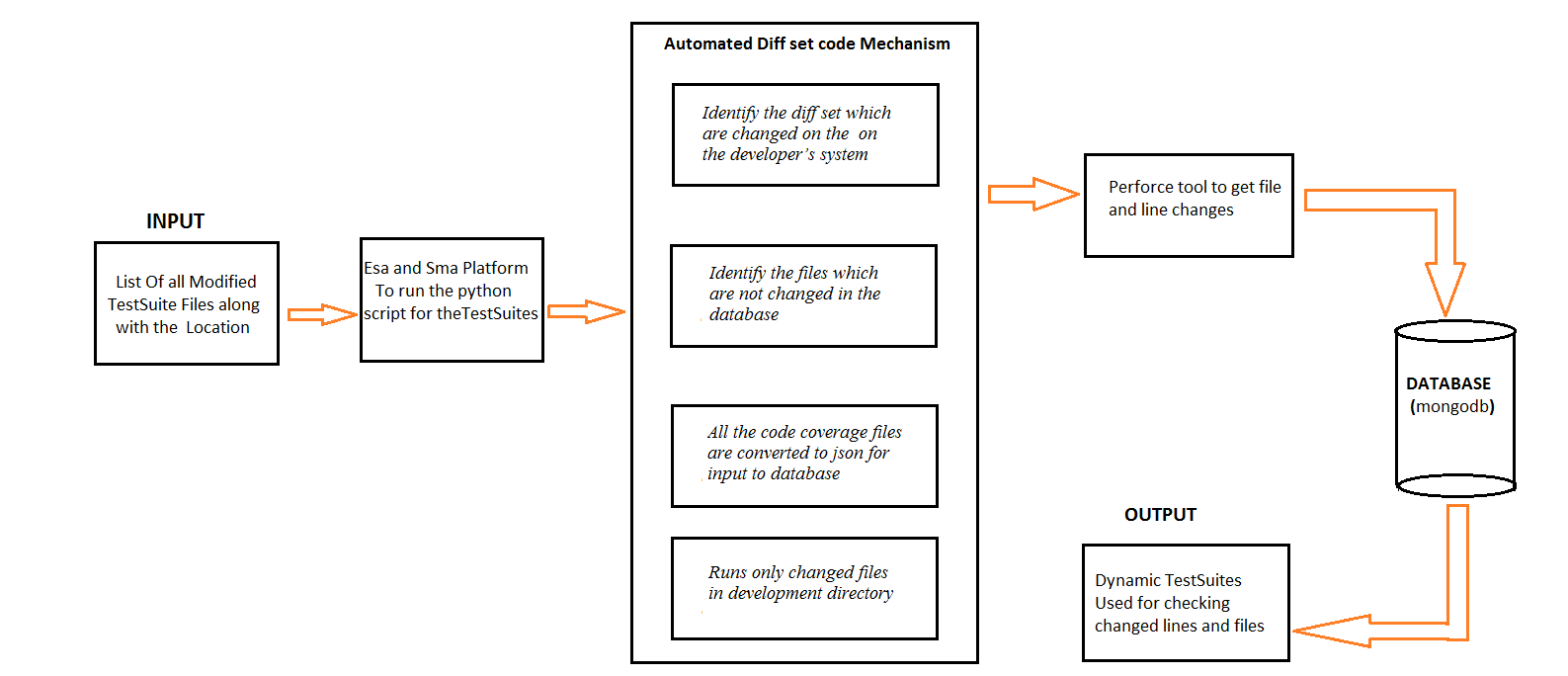
The application takes care of updating the database whenever a new file comes in the development directory.

* *Runs only changed files in development directory:*

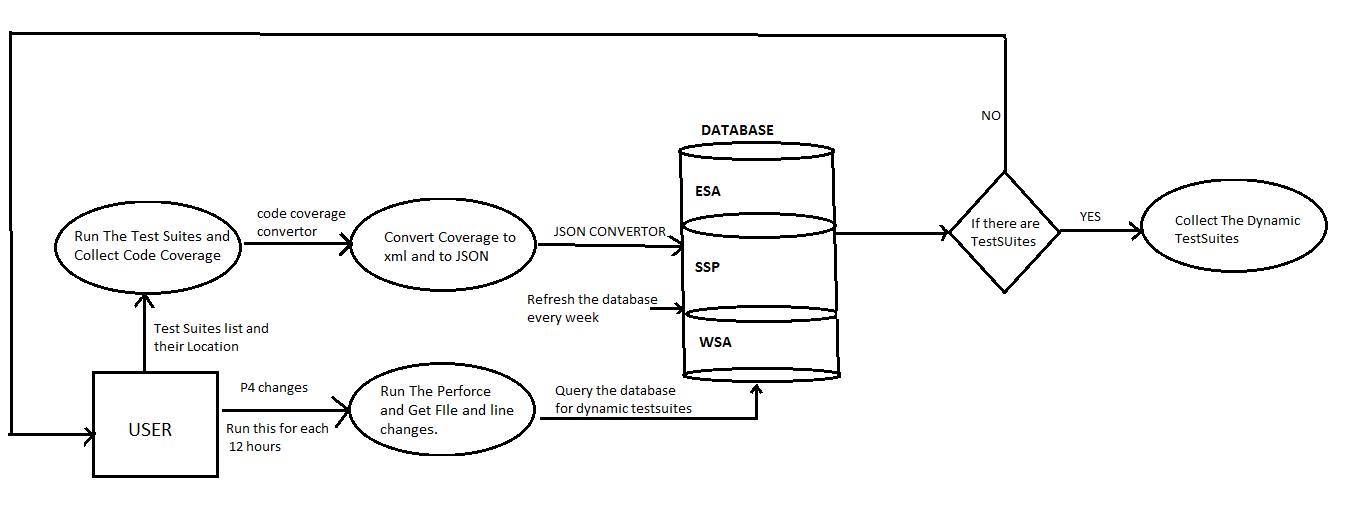
The changed files will run on the server once the complete code coverage and update in the database in done in the next cycle.

* 1. Algorithm design

1. Input 🡨 List of all modified test suite files and their location
2. Use of ESA and SMA platform to run the input test suites files and get the corresponding code coverage files.
3. For each code coverage file of the corresponding test suite **ϵ** code coverage files List obtained above
4. Do
   * 1. Convert the coverage file to xml
     2. Convert the xml to JSON using JSON code script
     3. Populate the database by giving the above JSON file as input
     4. Run the code review tool (perforce ) to get the latest changes
     5. Query the database using the output got from above step
     6. Collect the dynamic test suites
     7. Run those dynamic test suites only
     8. Update the database every week by following above steps.
5. Copy the dynamic test suites to the respective location
6. Refresh the database and populate the database with new updates obtained from the above dynamic test suites
7. **Architecture Design**

****

1. **Data flow diagram**

****

1. **References**
2. Pankaj Jalote: A Concise Introduction to Software Engineering , Springer, 2008
3. Roger S. Pressman: Software Engineering A Practitioner's Approach, 7th Edition, McGraw Hill, 2010
4. David Gustafson: Software Engineering, Schaum's Outline Series, McGraw Hill, 2002
5. <https://en.wikipedia.org/wiki/Data_flow_diagram>
6. <https://www.edrawsoft.com/Data-Flow-Diagram-Symbols.php>