What is claimed is:

1. A computer-implemented method, comprising:

   generating, with at least one processor and based on a test list and a source file list, a test mapping comprising a mapping between a plurality of tests and a plurality of source files, wherein the plurality of tests and the plurality of source files are associated with a software repository, each source file in the test mapping associated with a code change, wherein at least one source file in the test mapping is associated with a code change to the at least one source file, and wherein at least one other source file in the test mapping is associated with a code change to a source file that belongs to at least one common component of the at least one other source file;

   determining, with the at least one processor and at least one machine learning model, a defective score for each test of the plurality of tests based on historical test data of the test;

   receiving, with the at least one processor, a component criticality score and a defect definition for each source file of the plurality of source files, the defect definition comprising at least one of a predefined score or rating based on one or more issues within the source file and stored in an issue tracking database;

   generating, with the at least one processor, a key value for each test of the plurality of tests, the key value based on values of the defective score, the component criticality score, and the defect definition;

   determining, with the at least one processor and the at least one machine learning model, a subset of tests of the plurality of tests based on key values corresponding to the plurality of tests; and

   executing, with the at least one processor, the subset of tests with the software repository.

2. The computer-implemented method of claim 1, further comprising:

   generating, with the at least one processor, test results based on execution of the subset of tests; and

   inputting, to the at least one machine learning model, the test results for further training of the at least one machine learning model.

3. The computer-implemented method of claim 1, wherein generating the test mapping comprises:

   detecting at least one source file based on a change to at least one source file, wherein the change comprises differences between a first version of the at least one source file and a second version of the at least one source file;

   selecting at least one test corresponding to the at least one source file based on a predefined mapping of source files to tests;

   compiling the source file list and the test list, wherein the source file list comprises a source filename of the at least one source file and the test list comprises a test filename of the at least one test; and

   generating the test mapping from the source file list and the test list by mapping the source filename to its corresponding test filename based on the predefined mapping of source files to tests.

4. The computer-implemented method of claim 3, wherein the at least one source file is a new source file and the change comprises the new source file added to the software repository.

5. The computer-implemented method of claim 1, wherein determining the subset of tests comprises:

generating a recommended subset of tests based on the key values and a predefined test category;

generating a mapping file based on the recommended subset of tests and the source files;

storing the mapping file in a database; and

formatting the recommended subset of tests into an executable format.

6. The computer-implemented method of claim 1, wherein the at least one machine learning model uses supervised learning and prediction to determine the subset of tests.

7. The computer-implemented method of claim 5, wherein the mapping file stored in the database is used as input to the at least one machine learning model for further training.

8. A system, comprising:

a test selection system;

at least one machine learning model;

a software repository; and

at least one processor, wherein the at least one processor is programmed or configured to:

generate, based on a test list and a source file list, a test mapping comprising a mapping between a plurality of tests and a plurality of source files, wherein the plurality of tests and the plurality of source files are associated with the software repository, each source file in the test mapping associated with a code change, wherein at least one source file in the test mapping is associated with a code change to the at least one source file, and wherein at least one other source file in the test mapping is associated with a code change to a source file that belongs to at least one common component with of the at least one other source file;

determine a defective score for each test of the plurality of tests based on historical test data of the test;

receive a component criticality score and a defect definition for each source file of the plurality of source files, the defect definition comprising at least one score, the at least one score based on data in an issue tracking database associated with the source file;

generate a key value for each test of the plurality of tests, the key value based on values of the defective score, the component criticality score, and the defect definition;

determine a subset of tests of the plurality of tests based on key values corresponding to the plurality of tests; and

execute the subset of tests with the software repository.

9. The system of claim 8, wherein the at least one processor is further programmed or configured to:

generate test results based on execution of the subset of tests; and

input, to the at least one machine learning model, the test results for further training of the at least one machine learning model.

10. The system of claim 8, wherein, when generating the test mapping, the at least one processor is programmed or configured to:

detect at least one source file of the plurality of source files based on a change to the at least one source file, wherein the change comprises differences between a first version of the at least one source file and a second version of the at least one source file;

select, from the plurality of tests, at least one test corresponding to the at least one source file based on a predefined mapping of source files to tests;

compile the source file list and the test list, wherein the source file list comprises a source filename of the at least one source file and the test list comprises a test filename of the at least one test; and

generate the test mapping from the source file list and the test list by mapping the source filename to its corresponding test filename based on the predefined mapping of source files to tests.

11. The system of claim 10, wherein the at least one source file is a new source file and the change comprises the new source file added to the software repository.

12. The system of claim 8, wherein, when determining the subset of tests, the at least one processor is programmed or configured to:

generate a recommended subset of tests based on the key values and a predefined test category;

generate a mapping file based on the recommended subset of tests and corresponding source files;

store the mapping file in a database; and

format the recommended subset of tests into an executable format.

13. The system of claim 8, wherein the at least one machine learning model uses supervised learning and prediction to determine the subset of tests.

14. The system of claim 12, wherein the mapping file stored in the database is used as input to the at least one machine learning model for further training.

15. A computer program product, the computer program product comprising at least one non-transitory computer readable medium including one or more instructions that, when executed by at least one processor, cause the at least one processor to:

generate, based on a test list and a source file list, a test mapping comprising a mapping between a plurality of tests and a plurality of source files, wherein the plurality of tests and the plurality of source files are associated with a software repository, each source file in the test mapping associated with a code change, wherein at least one source file in the test mapping is associated with a code change to the at least one source file, and wherein at least one other source file in the test mapping is associated with a code change to a source file that belongs to at least one common component of the at least one other source file;

determine a defective score for each test of the plurality of tests based on historical test data of the test;

receive a component criticality score and a defect definition for each source file of the plurality of source files, the defect definition comprising at least one score;

generate a key value for each test of the plurality of tests, the key value based on values of the defective score, the component criticality score, and the defect definition;

determine a subset of tests of the plurality of tests based on key values corresponding to the plurality of tests; and

execute the subset of tests with the software repository.

16. The computer program product of claim 15, wherein the one or more instructions further cause the at least one processor to:

    generate test results based on execution of the subset of tests; and

    input, to at least one machine learning model, the test results for further training of the at least one machine learning model.

17. The computer program product of claim 15, wherein, when generating the test mapping, the one or more instructions further cause the at least one processor to:

    detect at least one source file based on a change to the at least one source file, wherein the change comprises differences between a first version of the at least one source file and a second version of the at least one source file;

    select at least one test corresponding to the at least one source file based on a predefined mapping of source files to tests;

    compile the source file list and the test list, wherein the source file list comprises a source filename of the at least one source file and the test list comprises a test filename of the at least one test; and

    generate the test mapping from the source file list and the test list by mapping the source filename to its corresponding test filename based on the predefined mapping of source files to tests.

18. The computer program product of claim 17, wherein the at least one source file is a new source file and the change comprises the new source file added to the software repository.

19. The computer program product of claim 15,

    wherein, when determining the subset of tests, the one or more instructions further cause the at least one processor to:

    generate a recommended subset of tests based on the key values and a predefined test category;

    generate a mapping file based on the recommended subset of tests and corresponding source files;

    store the mapping file in a database; and

    format the recommended subset of tests into an executable format.

20. The computer program product of claim 19, wherein the mapping file stored in the database is used as input to at least one machine learning model for further training.