## CLAIMS

What is claimed is:

1. A method, implemented in a computer system comprising a processor, memory accessible by the processor, and computer program instructions stored in the memory and executable by the processor, the method comprising:

generating a first performance and security information relating to a software system during a development of the software system, wherein generating the first performance and security information relating to the software system during the development of the software system comprises performing a static and dynamic testing of code of the software system during the development of the software system to collect a history of one or more generated alerts;

generating a second performance and security information relating to the software system during a deployed operation of the software system;

matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system to determine one or more performance and security alerts to escalate; and

reporting the one or more escalated performance and security alerts.

2. The method of claim 1, wherein generating the first performance and security information relating to the software system during the development of the software system comprises at least one of:

performing a first dynamic testing of one or more subsystems of code of the software system during the development of the software system to generate one or more application logs for the one or more generated alerts; and

performing a second dynamic testing of one or more units of code of the software system during the development of the software system to generate one or more error-handling logs for the one or more generated alerts.

3. The method of claim 2, further comprising associating each generated alert of the one or more generated alerts with an immediate log-sequence using one or more generated call graphs representing one or more calling relationships between routines in the software system.

4. The method of claim 3, wherein generating the second performance and security information relating to the software system during the deployed operation of the software system comprises monitoring the deployed operation of the software and collecting one or more monitor alerts produced by monitoring the deployed operation of the software.

5. The method of claim 4, wherein generating the second performance and security information relating to the software system during the deployed operation of the software system comprises collecting at least one of an information regarding one or more configuration changes and one or more produced alerts regarding the one or more configuration changes and the matching comprises matching the collected information regarding at least one of the one or more configuration changes and the one or more produced alerts regarding the one or more configuration changes with the second performance and security information generated during the deployed operation of the software system to determine a performance and the one or more performance and security alerts to escalate.

6. The method of claim 4, wherein matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system comprises:

determining a root cause of an alert produced by monitoring the deployed operation of the software using the history of alerts generated during the static and dynamic testing; and

responsive to the root cause of an alert produced by monitoring the deployed operation of the software matches a recent alert in the history of alerts generated during the static and dynamic testing, escalating the alert produced by monitoring the deployed operation of the software or of both the alert produced by monitoring the deployed operation of the software and the alert in the history of alerts generated during the static and dynamic testing.

7. The method of claim 6, wherein matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system to determine the one or more performance and security alerts to escalate further comprises:

matching the root cause of an alert produced by monitoring the deployed operation of the software with the recent alert in the history of alerts based on at least one of one or more similar log-sequences and one or more log-groups and is performed using one or more machine learning techniques;

matching one or more unit-test error handling logs of alerts generated during the static and dynamic testing with one or more monitor alerts produced by monitoring the deployed operation of the software by examining the one or more application logs using a recent time window, searching for one or more matches with known log-signatures of problems, and upon finding at least one match, escalating a severity of the alert produced by monitoring the deployed operation of the software; and

associating information relating to the one or more alerts produced by monitoring the deployed operation of the software with source code causing the one or more alerts generated during the static and dynamic testing.

8. A system comprising a processor, memory accessible by the processor, and computer program instructions stored in the memory and executable by the processor to perform:

generating a first performance and security information relating to a software system during a development of the software system, wherein generating the first performance and security information relating to the software system during the development of the software system comprises performing a static and dynamic testing of code of the software system during the development of the software system to collect a history of one or more generated alerts;

generating a second performance and security information relating to the software system during a deployed operation of the software system;

matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system to determine one or more performance and security alerts to escalate; and

reporting the one or more escalated performance and security alerts.

9. The system of claim 8, wherein generating the first performance and security information relating to the software system during the development of the software system comprises at least one of:

performing a first dynamic testing of one or more subsystems of code of the software system during the development of the software system to generate one or more application logs for the one or more generated alerts; and

performing a second dynamic testing of one or more units of code of the software system during the development of the software system to generate one or more error-handling logs for the one or more generated alerts.

10. The system of claim 9, further comprising associating each generated alert of the one or more generated alerts with an immediate log-sequence using one or more generated call graphs representing one or more calling relationships between routines in the software system.

11. The system of claim 10, wherein generating the second performance and security information relating to the software system during the deployed operation of the software system comprises monitoring the deployed operation of the software and collecting one or more monitor alerts produced by monitoring the deployed operation of the software.

12. The system of claim 11, wherein generating the second performance and security information relating to the software system during the deployed operation of the software system comprises collecting at least one of an information regarding one or more configuration changes and one or more produced alerts regarding the one or more configuration changes and the matching comprises matching the collected information regarding at least one of the one or more configuration changes and the one or more produced alerts regarding the one or more configuration changes with the second performance and security information generated during the deployed operation of the software system to determine a performance and the one or more performance and security alerts to escalate.

13. The system of claim 11, wherein matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system comprises:

determining a root cause of an alert produced by monitoring the deployed operation of the software using the history of alerts generated during the static and dynamic testing; and

responsive to the root cause of an alert produced by monitoring the deployed operation of the software matches a recent alert in the history of alerts generated during the static and dynamic

testing, escalating the alert produced by monitoring the deployed operation of the software or of both the alert produced by monitoring the deployed operation of the software and the alert in the history of alerts generated during the static and dynamic testing.

14. The system of claim 13, wherein matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system to determine the one or more performance and security alerts to escalate further comprises:

matching the root cause of an alert produced by monitoring the deployed operation of the software with the recent alert in the history of alerts based on at least one of one or more similar log-sequences and one or more log-groups and is performed using one or more machine learning techniques;

matching one or more unit-test error handling logs of alerts generated during the static and dynamic testing with one or more monitor alerts produced by monitoring the deployed operation of the software by examining the one or more application logs using a recent time window, searching for one or more matches with known log-signatures of problems, and upon finding at least one match, escalating a severity of the alert produced by monitoring the deployed operation of the software; and

associating information relating to the one or more alerts produced by monitoring the deployed operation of the software with source code causing the one or more alerts generated during the static and dynamic testing.

15. A computer program product comprising a non-transitory computer readable storage having program instructions embodied therewith, the program instructions executable by a computer, to cause the computer to perform a method comprising:

generating a first performance and security information relating to a software system during a development of the software system, wherein generating the first performance and security information relating to the software system during the development of the software system comprises performing a static and dynamic testing of code of the software system during the development of the software system to collect a history of one or more generated alerts;

generating a second performance and security information relating to the software system during a deployed operation of the software system;

matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system to determine one or more performance and security alerts to escalate; and

reporting the one or more escalated performance and security alerts.

16. The computer program product of claim 15, wherein generating the first performance and security information relating to the software system during the development of the software system comprises at least one of:

performing a first dynamic testing of one or more subsystems of code of the software system during the development of the software system to generate one or more application logs for the one or more generated alerts; and

performing a second dynamic testing of one or more units of code of the software system during the development of the software system to generate one or more error-handling logs for the one or more generated alerts.

17. The computer program product of claim 16, further comprising associating each generated alert of the one or more generated alerts with an immediate log-sequence using one or more

generated call graphs representing one or more calling relationships between routines in the software system.

18. The computer program product of claim 17, wherein generating the second performance and security information relating to the software system during the deployed operation of the software system comprises collecting at least one of an information regarding one or more configuration changes and one or more produced alerts regarding the one or more configuration changes and the matching comprises matching the collected information regarding at least one of the one or more configuration changes and the one or more produced alerts regarding the one or more configuration changes with the second performance and security information generated during the deployed operation of the software system to determine a performance and the one or more performance and security alerts to escalate.

19. The computer program product of claim 18, wherein matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system comprises:

determining a root cause of an alert produced by monitoring the deployed operation of the software using the history of alerts generated during the static and dynamic testing; and

responsive to the root cause of an alert produced by monitoring the deployed operation of the software matches a recent alert in the history of alerts generated during the static and dynamic testing, escalating the alert produced by monitoring the deployed operation of the software or of both the alert produced by monitoring the deployed operation of the software and the alert in the history of alerts generated during the static and dynamic testing;

wherein matching the root cause of an alert produced by monitoring the deployed operation of the software with a recent alert in the history of alerts is based on similar log-sequences or log-groups and is performed using machine learning techniques.

20. The computer program product of claim 19, wherein matching the first performance and security information generated during the development of the software system with the second performance and security information generated during the deployed operation of the software system to determine the one or more performance and security alerts to escalate further comprises:

matching the root cause of an alert produced by monitoring the deployed operation of the software with the recent alert in the history of alerts based on at least one of one or more similar log-sequences and one or more log-groups and is performed using one or more machine learning techniques;

matching one or more unit-test error handling logs of alerts generated during the static and dynamic testing with one or more monitor alerts produced by monitoring the deployed operation of the software by examining the one or more application logs using a recent time window, searching for one or more matches with known log-signatures of problems, and upon finding at least one match, escalating a severity of the alert produced by monitoring the deployed operation of the software; and

associating information relating to the one or more alerts produced by monitoring the deployed operation of the software with source code causing the one or more alerts generated during the static and dynamic testing.