

Claims

1. A method for application environment migration, the method comprising:
 - assessing, by a processor (122) of an application server (102), a source application code of a source application environment;
 - ascertaining, by the processor (122), a quantum change for migrating the source application code to a target application code;
 - forecasting, by the processor (122), an assessment statistic (302), wherein the assessment statistic (302) provides at least one functional readiness (304) and a timeline (306) to complete the migration of the source application code;
 - scanning, by the processor (122), the source application code for identifying a business logic;
 - generating, by the processor (122), a re-factored code for the source application code, wherein the re-factored code is generated by breaking the source application code into macro-services (426a, ..., 426n) and repackaging the macro-services (426a, ..., 426n) in accordance with the target application code while retaining the business logic;
 - updating, by the processor (122), components of the source application environment as per the forecasted assessment statistic (302) and the re-factored code;
 - migrating, by the processor (122), the source application environment to the target application environment, wherein the migration re-platforms the updated components and the re-factored code of the source application environment to the target application environment.
2. The method as claimed in claim 1, wherein the quantum change for migrating the source application code to the target application code is calculated on the basis of the size and complexity of the source application environment.
3. The method as claimed in claim 1, wherein the re-factored code is generated utilizing a continuous integration and deployment framework (150) and an automated test framework (150).
4. The method as claimed in claim 1, wherein the assessment statistic (302) provides at least one inhibitor to complete the migration of the source application code.

5. The method as claimed in claim 1, wherein scanning the source application code includes scanning links between components of the source application environment.
6. The method as claimed in claim 1, wherein repackaging the macro-services (426a, ..., 426n) in accordance with the target application code is completed while retaining links between components of the source application environment.
7. The method as claimed in claim 1, wherein the macro-services (426a, ..., 426n) are identified by an AI Engine (152).
8. The method as claimed in claim 1, further comprising: migrating, by the processor (122), libraries and design patterns from the source application environment in accordance with the target application environment.
9. The method as claimed in claim 1, further comprising: implementing, by the processor (122), security rules from the source application environment in accordance with the target application environment.
10. The method as claimed in claim 1, further comprising: packaging and building, by the processor (122), each service into a separate container by generating an application program interface proxy utilizing a gateway and a continuous integration and continuous deployment framework (150).
11. The method as claimed in claim 1, further comprising: accessing, by the processor (122), a plurality of storage units (144) comprising a business logic, links, rule engines, libraries of available environments, standard tools, and coding languages.
12. The method as claimed in claim 1, wherein the source application environment migrates to the target application environment comprising the steps of: a source to target environment re-development, a source to target application re-factoring, a source to target application re-hosting, and a source to target application re-platforming.
13. A system for application environment migration, the system comprising:
a processor (122); and

a memory (126) coupled to the processor (122), wherein the processor (122) executes a plurality of modules (128) stored in the memory (126), and wherein the plurality of modules (128) comprising:

an assessment module (130), for assessing a source application code of a source application environment and ascertaining a quantum change for migrating the source application code to a target application code, wherein the assessment module (130) forecasts an assessment statistic (302) that provides at least one functional readiness (304) and a timeline (306) to complete the migration of the source application code;

a re-factor module (132), for scanning the source application code to understand a business logic and generating a re-factored code for the source application code, wherein the re-factored code is generated by breaking the source application code into macro-services (426a, ..., 426n) and repackaging the macro-services (426a, ..., 426n) in accordance with the target application code while retaining the business logic;

a re-platform module (134), for updating components of the source application environment as per the forecasted assessment statistic (302) and the re-factored code and migrating the source application environment to the target application environment, wherein the migration re-platforms the updated components and the re-factored code of the source application environment to the target application environment.

14. The system as claimed in claim 13, wherein the quantum change for migrating the source application code to a target application code is calculated on the basis of the size and complexity of the source application environment.
15. The system as claimed in claim 13, wherein the re-factored code is generated utilizing a continuous integration and deployment framework (150) and an automated test framework (150).
16. The system as claimed in claim 13, wherein the assessment statistic (302) provides at least one inhibitor to complete the migration of the source application code.
17. The system as claimed in claim 13, wherein scanning the source application code includes scanning links between components of the source application environment.

18. The system as claimed in claim 12, wherein repackaging the macro-services (426a, ..., 426n) in accordance with the target application code is completed while retaining links between components of the source application environment.
19. The system as claimed in claim 13, wherein the macro-services (426a, ..., 426n) are identified by an AI Engine (152).
20. The system as claimed in claim 13, wherein the re-factor module (132) further comprising: migrating, by the processor (122), libraries and design patterns from the source application environment in accordance with the target application environment.
21. The system as claimed in claim 13, wherein the re-platform module (134) further comprising: implementing, by the processor (122), security rules from the source application environment in accordance with the target application environment.
22. The system as claimed in claim 13, wherein the re-factor module (132) further comprising: packaging and building, by the processor (122), each service into a separate container by generating an application program interface proxy utilizing a gateway and a continuous integration and continuous deployment framework (150).
23. The system as claimed in claim 13, wherein the plurality of modules (128) further executes accessing, by the processor (122), a plurality of storage units (144) comprising a business logic, links, rule engines, libraries of available environments, standard tools, and coding languages.
24. The system as claimed in claim 13, wherein the source application environment migrates to the target application environment comprising the steps of: a source to target environment re-development, a source to target application re-factoring, a source to target application re-hosting, and a source to target application re-platforming.