# **TRI-3ECR ROI calculation**

## **3E Automation Overview**

3E is a big enterprise legacy application which is exist for more than 10 years and will work for years. System is being continuously developing, upgrading, and integrating with 3rd party components. #E has 1,300,00 lines of executable code which is supported by 17 teams. Changes made by one team could affect functional area of the other teams. Best way to avoid such problems in to cover code with unit test and implement Automated Test Solution which will detect such breaking changes on early stages.

Automation is already implemented on the project as on this stage of application lifecycle automation brings profit. Test types that automation is covering:  
a) Code is covered with unit test and implemented in CI pipeline. Unit tests are running each time when build is created. Unit test coverage should be not less than 30 %.  
b) Test cases for 3rd Party Integrated components are API tests. P1, P2 test cases are automated.  
c) Basic end-to-end scenarios are covered with API/UI tests. P1, P2 test cases are automated.  
d) 3E API functional scenarios are covered with API tests. P1, P2 test cases are automated.  
e) 3E UI functional scenarios are covered with UI tests, if possible. P1, P2 test cases are automated.

Data Driven Approach for test creation is implemented. It Allows to Create new Release environment and Run Test Cases on Instances restored from Database backpack created from Automation environments.

## **ROI Calculation**

**TRI-3ECR Project** (3E application and Integrated Components) is supported by EPAM for 8 years and planned to be supported at least for 3 more years. Automated Test cases executed on 3E Project based on Team’s needs. Project has 18604 Test Cases of priority 1 and 2, that should be executed each release.

ROI will be calculated based on Regression Suit for 3E Releases.

In accordance with Project Requitements Prerelease Regression Testing includes run of Test Case of Priority 1 and 2. The amount of Functional Test Cases (UI, API, E-2-E, Integration) which is executed during Prerelease is 18604.

**Parameters for Calculation:**  
For current moment 12121 Test Cases are automated.   
Product is released 3 times per year. For next 3 years 9 releases are planned.  
Average time for Manual Test Case Run is 30 minutes.  
Average time for Test Case Automation is 2h 30 min. Time is longer when framework is implemented. Adding scenarios using implemented functionality could take less than 15 minutes.  
Let’s consider that time for Framework implementation was 1000 h and each release Automation engineers spent 300 hours to support test cases.  
Cost of manual testing in man hours for 9 release: 12121 test cases\*0.5 h \*9 releases = **54544** man-hours.  
Time spent for creating automated scenarios and support for 9 releases: 12121 test cases \* 2,5 h \* +1000 h (Framework implementation) + 300h \* 9 releases (Support) = **34003** man-hours

**ROI = (54544-(34003))/ (34003)) = 60,40%**

## **Summary**

In order to maximize value that automation if bringing to the Project, the next rules to Automation Run/Implementation should be executed:  
a) Automation should be executed as a part of CI/CD after each deployment on Manual Testing Environment based on the Team’s needs (API tests/UI Smoke)  
b) Automation should be executed Daily on Separate Automated Environment (Integration Scenarios, API tests)  
c) Automation should be Weekly on special Automation Environment (Long Running Integration Scenarios, UI Scenarios).  
d) Full Automation Regression suit should be executed during Prerelease Testing.  
e) Test Cases should be implemented using Data Driven Approach, so that is could be Run on different environments.