Quality assessment in DevOps:

Automated Analysis of a Tax Fraud Detection System

Diego Perez-Palacin, Youssef Ridene, José Merseguer

University of Zaragoza, Netfective Technology



eGov Tax Fraud Detection System Under Development by Netfective Technology





eGov Tax Fraud Detection System Under Development by Netfective Technology



Tax fraud represents a huge problem for governments.



https://ec.europa.eu/taxation_customs/fight-against-tax-fraud-tax-evasion/missing-part_en

 EU has estimated tax evasion to be of the order of 1 trillion euros



Big Blu is developed following Agile and DevOps principles

- Follow an iterative process with incremental iterations pursuing
 - > Quick design
 - > Quick delivery of enhancements
 - Quick feedback

- Bring closer Development and Operations activities to improve
 the effectiveness of each incremental iteration
 - > Achieve faster iterations
 - > Achieve higher proportion of iterations with satisfactory results



- GUI: web based application. Unique interface
- Web Services: implement RESTful interoperability and deployed on Tomcat
- Back-end: Data processing elements



Software Architecture composed of 3 main layers:

• GUI: web based application. Unique interface

big biu 😑			8600							
Ridene Youssef	Home Welcome to Big Blu : an eGov Tax Fraud Detector Big Blu is designed and developed in the content of the DCE project (http://www.dce.N0020.ex). Big Blu is an MVP to prove the capabilities of Big Data frameworks in the content of governmental applications. The evasion and flauds represents a hope problem for governments, causing them a big loss of money each year. Governments are increasily upits (Big Data in multiple actors to their) their applications. There evasion and flauds represents a hope problem for governments, causing them a big loss of money each year. Governments are increasily upits (Big Data in multiple actors to their) their applications. There evasion and flauds represents and businesses. In Big Data has the potential to make the application businesses. In Big Data has the potentian business and ended in the developed in the developed in the developed in the big the target excitation. HET's plans to baid a some entropy to demonstrate are into baid an e-government.				eroperability and deployed					
Administrator & (i) (i) Home										
Fraud Detection Fraud Indicators	of "fraudulent conduct" from the automatic operations on existing tax data files such as business creat • identifying taxpayers who are registered in different regions in order to collect fraudulently social Big	g Blu 📃					eGov Ta	ax Fraud Detector 💠		
Statistics Administration	 The fictilious relocation of the taxpayer who improperly claim domiciled abroad in order to not pr Companies normally collect WRT from their customers but "forget" to pay back to the Treasury a companies do not hestate to include the WRT debt labilities in their balance sheet, which proves 		Frauds Detection							
	 We may even consider detecting ID tax final since it has been the most attractive type of ident money. The demonstrator will duclitate the task of littering and gathering data for facat agents in order to incre exploring and analyzing high volumes of data from various heterogeneous sources should be scalable. This demonstrator will another any syntacy/confedentially issues since processed markon data which ar specifically for being realistic (battures could apply to a real system), generic (features and data mod stage of the DICE project the metamodel describes a tappayer, and the data generator produces real 	Administrator	Launch New Detection Expl Expl	Launch New Detection						
			Successfully submitted a new detection with the	e id detection20170124	171050420					
		Fraud Detection	Detection ID	Launch Dat	te Database	Indicators	Status	Actions		
	n€ s	Statistics	detection20170124171007574	24/01/2017 17:	10:07 database20170117164508855	5 FI1[Income decrease = 23]	RUNNING	Kill Job		
	© 2016 Big	Administration	detection20170124171050420	24/01/2017 17:	10:50 database20170117154626276	6 FI1[Income decrease = 23]	SUBMITTED	Kill Job		



- GUI: web based application. Unique interface
- Web Services: implement RESTful interoperability and deployed on Tomcat
- Back-end: Data processing elements











DICE approach



Researches towards building a quality-driven framework for development, deployment, monitoring and continuous improvement of Data-Intensive Cloud Applications.

Pursues developments with Iterative Quality enhancements

- Delivers a toolchain for:
 - Design
 - Quality analysis
 - Deployment
 - Testing



- Monitoring (collect data, visualization, anomaly detection, trace checking)
- Enhancement

DICE approach



Researches towards building a quality-driven framework for development, deployment, monitoring and continuous improvement of Data-Intensive Cloud Applications.

Pursues developments with Iterative Quality enhancements

UNIFIED

MODELING

LANGUAGE

- Delivers a toolchain for:
 - Design
 - Quality analysis-
 - Deployment
 - Testing
 - Monitoring (collect data, visualization, anomaly detection, trace checking)

DICE Simulation tool

PROFILING

Enhancement









































Usefulness in Agile cycles following DevOps

Scenario 1: Development of new functionalities

PROBLEM

- In agile cycles, the required quality of the new functionalities may not be clear for developers
 - > The quality requirements refer to the overall system quality

CONSEQUENCES

 Obtained quality of the new functionality is not good enough and the cycle has to be repeated





Usefulness in Agile cycles following DevOps

Scenario 1: Development of new functionalities

APPROACH TO SOLUTION

 Obtain values for ``appropriate quality" of the new functionality that can be already asserted during the unit tests



 Developers deliver a functionality that passes these unit tests a go to next phases of the cycle with some confidence about the quality





Universidad Zaragoza

Diego Perez-Palacin

Usefulness in Agile cycles following DevOps

Scenario 2: Maintenance of functionalities

PROBLEM

- Quality of a functionality has to be improved...
 - > Due to changes in the utilization of the application
 - > Due to new quality restrictions and improvable designs
- ...and can be improved in different phases of DevOps toolchain

CONSEQUENCES

- Maintenance may not achieve the expected quality
- Modifications result more expensive than necessary



Usefulness in Agile cycles following DevOps

Scenario 2: Maintenance of functionalities

APPROACH TO SOLUTION

- Update the models with recent monitored data
- Identify quality issues
- Evaluate the alternatives to solve the issues
- Decide for the maintenance that seems the "smartest" action





























1542

- Quality malfunction reported \rightarrow maintenance
- Using the SimTool we obtain
- Developers see two possible solutions
 - Acquire more computing nodes to parallelise requests
 - Reengineer Launch Fraud Detection activity to make it faster
- Using the SimTool we obtain







Diego Perez-Palacin



Universidad Zaragoza

1542

Diego Perez-Palacin

- Adding a new functionality
 - API that is invoked frequently
 - Provides volatile information to all clients



- Adding a new functionality
 - API that is invoked frequently
 - Provides volatile information to all clients
 - It executes in the Web Services layer





- Adding a new functionality
 - API that is invoked frequently
 - Provides volatile information to all clients
 - It executes in the Web Services layer
 - Each of the current 200 clients will make 1 request per minute
 - Developers believe that they can make the demand of the API to be less than 20 milliseconds for execution, but there are no new performance requirements



- Adding a new functionality
 - API that is invoked frequently
 - Provides volatile information to all clients
 - It executes in the Web Services layer
 - Each of the current 200 clients will make 1 request per minute
 - Developers believe that they can make the demand of the API to be less than 20 milliseconds for execution, but there are no new performance requirements
 - Developers do not know if their development will be good enough until integration or operation





Universidad Zaragoza

Diego Perez-Palacin



Universidad Zaragoza

Diego Perez-Palacin

Conclusions

- Report an experience on the utilization of a quality evaluation tool during DevOps-oriented software development
- Reduce the number of development cycles until reaching a satisfactory modification of the system
- Reported two common scenarios in development cycles
 - Maintenance activity
 - Development of new functionality

FUTURE:

- SimTool will incorporate characteristics of Big Data technologies (Big Blu uses Apache Spark)
- Complete the full integration of the different quality analysis tools
 used within the DICE methodology

Quality assessment in DevOps:

Automated Analysis of a Tax Fraud Detection System

THANK YOU FOR YOUR ATTENTION!



Diego Perez-Palacin