Towards Quality-Aware Development of Big Data Applications with DICE

Pooyan Jamshidi
Imperial College London, UK

Project Coordinator:
Giuliano Casale
Imperial College London, UK

DICE Horizon 2020 Project
Grant Agreement no. 644869
http://www.dice-h2020.eu

Funded by the Horizon 2020 Framework Programme of the European Union
DICE Project

- Horizon 2020 Research & Innovation Action (RIA)
  - Quality-Aware Development for Big Data applications
  - Feb 2015 - Jan 2018, 4M Euros budget
  - 9 partners (Academia & SMEs), 7 EU countries
Motivation

- Software market rapidly shifting to Big Data
  - 32% compound annual growth rate in EU through 2016
  - 35% Big data projects are successful [CapGemini 2015]
- ICT-9 call focused on SW quality assurance (QA)
  - ISTAG: call to define environments “for understanding the consequences of different implementation alternatives (e.g. quality, robustness, performance, maintenance, evolvability, ...)”
- QA evolving too slowly compared to the technology trends (Big data, Cloud, DevOps ...)
  - DICE aims at closing the gap
Quality Dimensions

- Reliability
  - Availability
  - Fault-tolerance

- Efficiency
  - Performance
  - Costs

- Safety & Privacy
  - Verification (e.g., deadlines)
  - Data protection
High-Level Objectives

- Tackling **skill shortage** and steep learning curves
  - Data-aware methods, models, and OSS tools
- Shorter **time to market** for Big Data applications
  - Cost reduction, without sacrificing product quality
- Decrease development and testing **costs**
  - Select optimal architectures that can meet SLAs
- Reduce number and severity of **quality incidents**
  - Iterative refinement of application design
Some Challenges in Big Data...

- Lack of quality-aware development for Big Data
  - How to described in MDE Big Data technologies
    - Spark, Hadoop/MapReduce, Storm, Cassandra, ...
    - Cloud storage, auto-scaling, private/public/hybrid, ...
  - Today no QA toolchain can help reasoning on data-intensive applications
    - What if I double memory?
    - What if I parallelize more the application?
... in a DevOps fashion

- Software development methods are evolving
- DevOps closes the gap between Dev and Ops
  - From agile development to **agile delivery**
  - Lean release cycles with automated tests and tools
  - Deep modelling of systems is the key to automation
## Demonstrators

<table>
<thead>
<tr>
<th>Case study</th>
<th>Domain</th>
<th>Features &amp; Challenges</th>
</tr>
</thead>
</table>
| Distributed data-intensive media system (ATC) | • News & Media  
• Social media | • Large-scale software  
• Data velocities  
• Data volumes  
• Data granularity  
• Multiple data sources and channels  
• Privacy |
| Big Data for e-Government (Netfective) | • E-Gov application | • Data volumes  
• Legacy data  
• Data consolidation  
• Data stores  
• Privacy  
• Forecasting and data analysis |
| Geo-fencing (Prodevelop) | • Maritime sector | • Vessels movements  
• Safety requirements  
• Streaming & CEP  
• Geographical information |
DevOps in DICE: Measurement

- Dev
- Ops
- Deployment & CI
- jenkins (performance unit tests)
- Users
- monitoring and incident report
- incident report
- DIA Node 1
  - MySQL
  - NoSQL
- DIA Node 2
  - S3
DevOps in DICE: Early-stage MDE

- early-stage quality assessment
- release
- Dev
- Ops Deployment & CI
- jenkins (performance unit tests)
- chef
- incident report
- monitoring and incident report
- DIA Node 1
  - MySQL
- DIA Node 2
  - NoSQL
  - S3
Quality-Aware MDE

- UML MARTE profile, UML DAM profile, Palladio, ...
Quality-Aware MDE

Platform-Indep. Model

Architecture Model

Platform-Specific Model

Domain Models

QA Models

Simulation Tools

Cost Optimization Tools

Platform-Dependent Model

Platform Description

Data Intensive Application

MARTE

UNIFIED MODELING LANGUAGE

Platform-Specific Model

Code stub generation

Simulation of Data Intensive Application

DICE RIA - Overview
DevOps in DICE: Enhancement

Continuous quality engineering
("shared system view" via MDE)

Ops
Deployment & CI

Release
jenkins

(Performance unit tests)

incident report

Incident report & model correlation

Users

DIA Node 1
MySQL

DIA Node 2
NoSQL

S3

Continuous monitoring and enhancement
## Year 1 Milestones

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Deliverables</th>
</tr>
</thead>
</table>
| Baseline and Requirements - July 2015 [COMPLETED] | • State of the art analysis  
• Requirement specification  
• Dissemination, communication, collaboration and standardisation report  
• Data management plan |
| Architecture Definition - January 2016 | • Design and quality abstractions  
• DICE simulation tools  
• DICE verification tools  
• Monitoring and data warehousing tools  
• DICE delivery tools  
• Architecture definition and integration plan  
• Exploitation plan |
Thanks
www.dice-h2020.eu