

SUSTAINABLE DEVELOPMENT, BIG DATA AND SUPERVISORY CONTROL: SERVICE QUALITY IN BANKING SECTOR

David Luigi Fuschi¹, Manuela Tvaronavičienė²

¹*Brunel University, Uxbridge UB8 3PH, London, United Kingdom*

²*Vilnius Gediminas Technical University, Saulėtekio Av. 11, LT-10223 Vilnius, Lithuania*

E-mails: ¹David.Fuschi@brunel.ac.uk; ²Manuela.Tvaronavičienė@vgtu.lt

Received 15 October 2013; 20 December 2013 accepted

Abstract. Sustainable development process is affected by contemporary phenomena. Big Data processing inefficiency is detrimental for banks' activity excellence. The software used for running and handling the inter-bank network framework provides services with extremely strict uptime (above 99.98 percent) and quality requirements, thus tools to trace and manage changes as well as metrics to measure process quality are essential. Having conducted a two year long campaign of data collection and activity monitoring it has been possible to analyze a huge amount of process data from which many aggregated indicators were derived, selected and evaluated for providing a managerial dash-board to monitor software development. The paper provides insights about the issues related to Big Data processing inefficiencies. Context of sustainable development is being taken into account.

Keywords: Big Data, banking sector, software reliability assessment; sustainable development.

Reference to this paper should be made as follows: Fuschi, D.L.; Tvaronavičienė M. 2014. Sustainable development, Big Data and supervisory control: service quality in banking sector, *Journal of Security and Sustainability Issues* 3(3): 5–14. DOI: [http://dx.doi.org/10.9770/jssi.2014.3.3\(1\)](http://dx.doi.org/10.9770/jssi.2014.3.3(1))

JEL Classifications: C81, C9, F02, G10

1. Introduction

There is unanimous agreement among scientists, practitioners, politicians and civil society that all possible efforts of all societal actors have to be directed toward sustainable development of separately taken countries, geographic and political regions and unions, blocs, more or less advanced societies (Lankauskienė, Tvaronavičienė 2012; Dudzevičiūtė 2012; Vosylius *et al.* 2013; Bileišis 2012; Mačiulis, Tvaronavičienė 2013). On the other hand, having agreed on principal direction on rather high level of abstraction, the same protagonists of practice and science immerse into intense ongoing discussion, when level of abstraction has to be lowered and benchmarking of desired state has to be started (Stańczyk 2011; Makštutis *et al.* 2012; Balkienė 2013; Laužikas, Krasauskas 2013). The problem is

that contemporary processes of development due to overwhelmingly rapid development of technologies, and, especially information technologies, start raising new urgent issues, which threaten businesses, industries, economies and societies (Białoskórski 2012; Matei, Savulescu 2012; Raišienė, Jonušauskas 2013). Aim of the presented paper is to discuss and articulate questions, which have to be draw attention of wider audience and serve as ground for further elaboration of scientific discussion in field of sustainable development in digital age of Big Data (Hsinchun *et al.* 2012; Fuschi, Tvaronavičienė 2014). In order to discuss how Big Data context affects perception, and, further, estimation of sustainable development state and progress, let us briefly overview sustainable development perception evolution.

Sustainable development as already classic concept

embraces economic, social and environmental dimensions of development. All dimensions are being measured by sets of indicators. What particular indicators should comprise those sets, how to integrate and benchmark selected indicators remains a contentious question (Dudzevičiūtė 2012; Makštutis *et al.* 2012; Laužikas, Krasauskas 2013). Not going into technicalities, let us draw attention to the gradually changing contexts, in which sustainable development processes are being analyzed. Term “security”, starts

to be used in economic literature (Białoskórski 2012; Makštutis *et al.* 2012; Lankauskienė, Tvaronavičienė 2012). At first we talk about social security, later energy security becomes a topicality. Anyway, sustainable development and secure development appears to be related and, in some cases, overlapped (Lankauskienė, Tvaronavičienė 2012; Vosylius *et al.* 2013). Interrelation between sustainability and security can be graphically depicted (Figure 1; Figure 2).

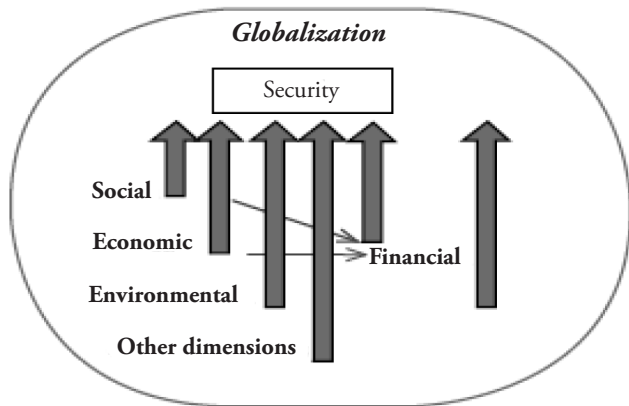


Figure 1. Security dimensions

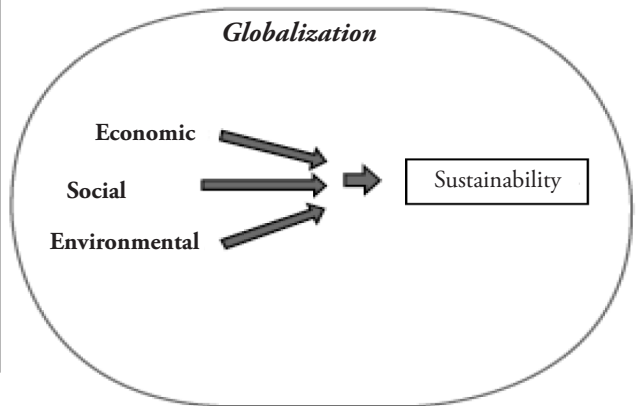


Figure 2. Sustainability dimensions

Fig.1. Security dimensions

Fig.2. Sustainability dimensions

Source: Lankauskienė, Tvaronavičienė (2012)

To generalize, sustainable development, or, rather, economic and technologic development naturally triggers threats, which could be considered as side effect of development (Raišienė, Jonušauskas 2013). To put in other words, threats is development, or “development stage” either “development context” sensitive. Hence, we can generalize that rapid development of IT, IT penetration into virtually all spheres of life and business generated respective “side” effects.

Emerging of Big Data phenomenon conditions necessity of its Critical Infrastructures (CIs) monitor and control. In order to tackle this new threat to sustainable development, let us consider a case study of software random failure in the process of fulfilling tasks in banking sector. Before going into technicalities of the case study, which witness practice, we need to reveal what role Big Data processing plays in sustainable development process and what clarify why Big Data processing lacks control. We will base our elaboration on practical banking sector example.

Here we need to remind readers, who not necessarily are economists, that sustainable development actu-

ally is not possible without proper economic growth. There are a lot of theories, schools of economic thought, which try to explain why one country develops faster than other (Borseková *et al.* 2012; Wahl, Prause 2013). Despite myriad of nuances (Laužikas, Mokšėckienė 2013), which those theories discuss and disagree, there are several fundamental ideas, which do not cause argumentation. Those fundamental ideas are: economic growth requires investments, which are used for acquisition of productive capacities and building infrastructures (Antanavičienė 2014). Productive capacities are being called fixed assets. Very important part of fixed assets is machinery, which is a tool for production (“capital” in economics. Note, in that context term “capital” does not mean “money” (Antanavičienė 2014); “capital” here is fixed capital, comprised of movable part – machinery and immovable – building). In our considered case, specifically, banking case, hardware performs function of “machinery” with all following guarantees provided by “machinery” supplier. Alas, not least role here is being played by software. Software, especially sufficiently expensive, falls under “fixed assets” heading as well

(in case software is free, economically is non-existent). Now a question stands: if “fixed assets” – computers are being hedged by its producers, why there is no virtually responsibility for software functioning. What software producers care about are property rights, what is undisputable. Alas, what about responsibility. Should it not be an area, where we could search for threats preventing and diminishing? Shouldn't that enhanced economic performance of all sector by preventing vast losses and shouldn't that contributed to economic growth, and consequently sustainable development? Big Data processing issues affect various spheres of our lives, as it was mentioned above; alas banking sector remains that one, where errors can be estimated and implications ultimately shared by multiple society actors (Dubauskas 2011; Baikovs, Zariņš 2013; Kaźmierczyk 2012.).

2. A case study in the banking and financial sector: witnessing Big Data processing shortcomings

Let us now turn to a case study, which will disclose Big Data processing inefficiency in banking sector. For commercial confidentiality reasons the operative context and location of the experiment cannot be fully disclosed. We just indicate, that the activity has been conducted in one of the G8 countries within the institutional service provider of the networking and application facilities to the country banking system.

The institution structure and operation context. The institution under exam is structured in three main divisions: banking applications, financial markets, and network infrastructure. Each division is responsible for internal software development and maintenance, interaction with subcontractors, service operation, and provision of help-desk services to customers. Provided services are extremely critical being used to manage the country's financial and banking system. For this reason the provider is subject to strong contractual constraints and commitments (for example a minimum up-time of 99.98%). Based on the criticality of users' request and the status of its internal operations, the scheduling of new releases is decided and planned carefully, and it is essential to evaluate existing relationship among incidents (i.e. customer reported problems), software anomalies (i.e. possible latent, yet not confirmed, errors), and software development so as to make informed decision for the suitability for release of a specific software release.

Software engineering practice in place and their revision. Back in 1994 a first software process improvement initiative aimed to monitor and measure products and processes was carried out. This was accomplished within the context of the ESSI project 21244 MIDAS. Activities were mainly focused on Change Management (CM) procedures, policies, and introduction of related tools (Cugola *et al.* 1997). Several metrics were regularly collected in relation to software development process and efforts incurred in the development. Available metrics and tools were assessed through a measurement program based on the Goal-Question-Metrics (GQM) paradigm. A further selection of the most relevant and the introduction of some derived indexes resulted in a significant improvement of the software development practices passing from Capability Maturity Model (CMM) level one to level four.

Let us describe metrics, which were used in the case study. The software production and maintenance process relates to a distributed infrastructure providing a full range of services for the management and operation of the national inter-bank network. The overall system was developed, maintained, and operated by the network division. The dimension of the software under exam amounts to more than five millions Lines of Code (LOC) in C programming language. The company attention was concentrated on anomalies management and the consequent impact on software maintenance, test and development process across releases.

The board of anomalies. The process to produce high-level indicators and reports evaluating data extracted from collected metrics was initially performed manually. Therefore the first activity performed was the automated extraction of data and report preparation. Anomaly related reports would be the starting point of discussion for Board of Anomalies weekly meeting. During the Board of Anomalies meetings newly opened and already opened anomalies would be individually discussed. Development team leaders would describe works progresses and provide indication on the availability of a fix (or the expected timeframe to such an event). According to the discussion, test activities would be planned; in particular test-on-demand activities could be requested to investigate anomalies origin. Suggested solution, work around, and information gained would then be spread through the company. Both the result of the discus-

sion and the report produced would also be given to the board of directors along with an historical report depicting the anomalies trend.

Anomalies report automation. Automation greatly reduced the effort required to produce anomalies report thus enabling its distribution via Intranet. Each project leader / manager could directly access trends and data on-line through a simple dashboard from which to choose what to see and how (overall counts, details, trends, etc.).

According to Malaiy and Denton (1998) the number

of residual errors decreases with the testing coverage. Therefore when considering a series of subsequent releases of the same software it is reasonable to expect that an initial peak of anomalies should be found in correspondence with a new re-lease (or a fix) due to introduced changes and enhancements, followed by a constant decrease in anomalies number as testing proceeds. In other words, it is reasonable to expect an overall monotonically decreasing trend towards a lower bound due to the fact that it will be impossible to totally avoid errors. Such behavior is shown in Figure 3.

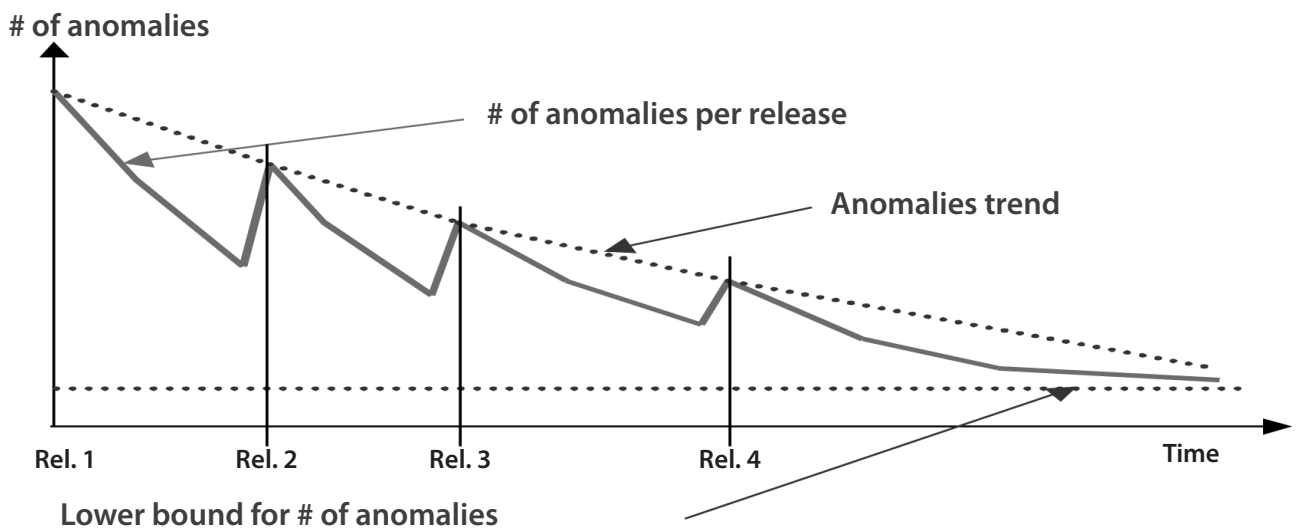


Fig.3. Theoretical behavior of anomalies trend

Source: Malaiy and Denton (1998); the authors

On the other hand if the actual trend, as observed from field measures, is substantially different from the theoretical one, an accurate investigation should be carried out.

Indicators selection criteria. While many metrics can be collected, not all of them will be of use. Furthermore, different metrics are of interest to different kind of

people. For example, project managers and technicians will be interested in KLOC, number of faults, MTBF, etc.; line managers will be interested in Productivity, Quality, Cost, On-time-delivery, etc.; finally, top managers will be interested in aggregated metrics useful to support decision taking. Metrics can be derived by adopting the model presented in Figure 4 hereafter.

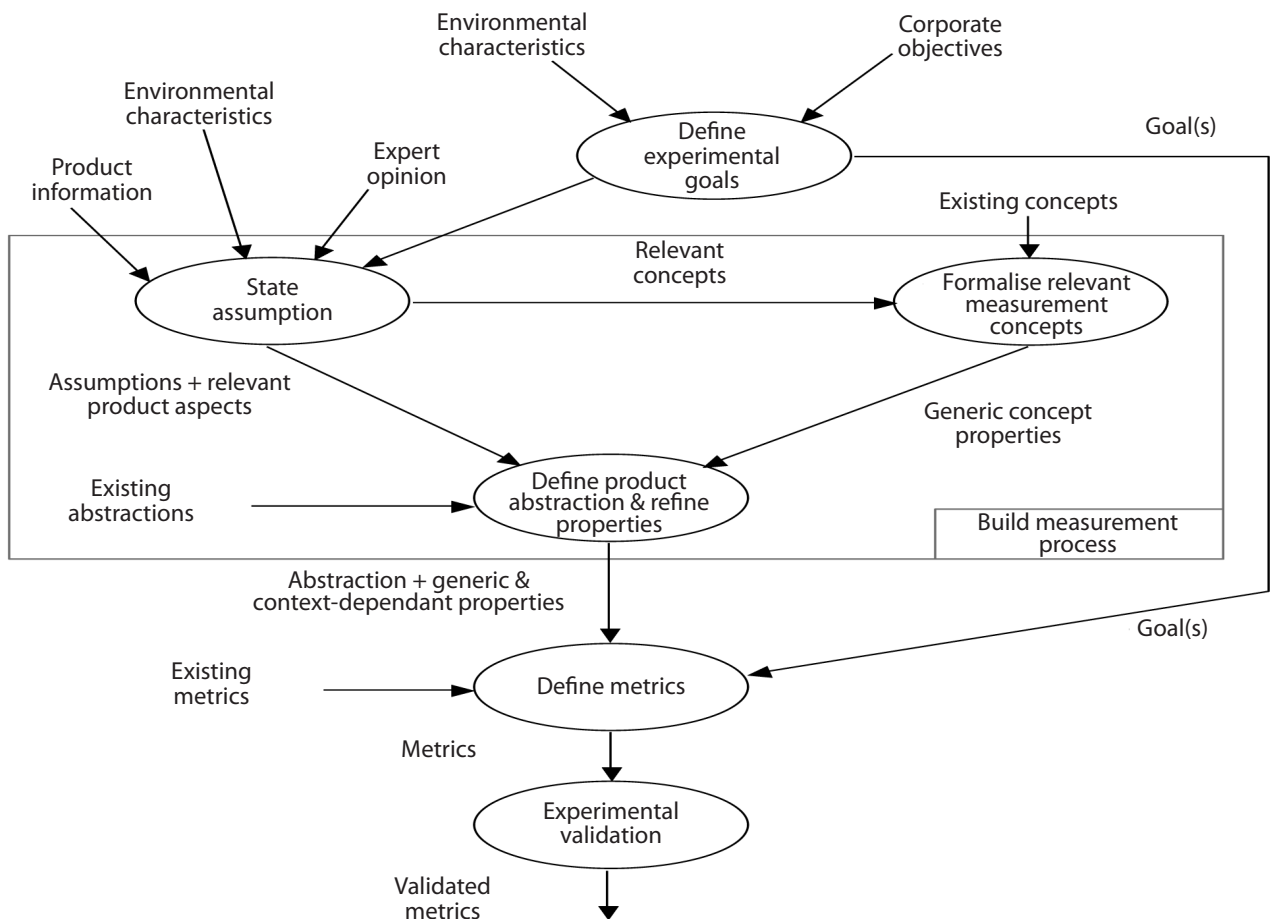


Fig.4. Model for metrics deriving

Source: Briand *et al.* (1994)

The application of provided above model proved very effective, although it was necessary to add a feed-back loop in the process so as to refine both the assumptions and the set of metrics to be used following the comments and requests placed by managers engaged in the process. This allowed achieving the best possible decision making support environment.

The experiment has been conducted processing metrics collected for the following system components:

- MTP - the front-end access point to the transport network (usually resident on a UNIX® machine);
- EAS - the actual application front-end access point to the network (an application usually resident on customer’s mainframe environment and interfacing with one or more MTP).

To ensure the management could base its decisions on a correct evaluation of provided indicators, only data related to software releases used in actual operation have been considered (i.e. releases that were not used

in production environment have not been considered). Indexes have been classified in major categories according to the measured object or derivation metric. Examined classes can be summarized as follows. Analyzed indexes categories are provided in Table 1.

Table 1. Analyzed indexes categories

Product quality	Process quality	Size & Productivity
MTBF	MTTF	KLOC
Total failures	Efficiency degree	KLOCC
Failure rate	Test quality index	Product change rate
Integration failure rate	Maintenance test time for KLOC	
Quality index		
Availability		

Source: normative literature, authors

Most Key Performance Indicators (KPI) adopted regarding product development life cycle quality assurance have been based on theory (i.e. as per normative literature Gall *et al.* 1997; Schneiewind 1997). However, there are problems that are not described (or only partially addressed) in literature, therefore some indicators were adapted (or even derived anew) based on observation and practical experience. Indicators selected are reported hereafter.

Mean Time To Failures (MTTF)

It shows the mean elapsed time between two consecutive faults (Schneiewind 1997). It is computed as the ratio between total life time (*TL*) and total number of failures during life (*LA*). An increasing trend denotes an improvement in the reliability of the software.

$$MTTF = TL / LA \quad (1)$$

Mean Time To Repair (MTTR)

It gives a synthetic view of the quality of software repair process. A low value implies either that anomalies are of minor relevance or that the repair process is efficient. Here TNA is the total number of anomalies found for the examined release at the time being. A high value denotes that anomalies solution is difficult. This is usually the case when a product is mature and therefore remaining uncovered bugs are difficult to find or solve.

$$MTTR = [\sum_{i=1, TNA} (t_{AN\ closure} - t_{AN\ opening})] / TNA \quad (2)$$

Mean Time Between Failures (MTBF)

It gives a synthetic view of the quality of software. In fact if *MTTF* is high and *MTTR* is low, the overall process is running smoothly. It can be used for predicting fault occurrence only if derived from a long series of historical data. Anyhow as these are mean values, derived indication should be compared with anomalies distribution, failure rate, effort and elapsed time per process phase.

$$MTBF = MTTF + MTTR \quad (3)$$

Total Failures

It shows the normalized ratio between failures during life and the total amount of changed lines in code. A decreasing trend denotes an improvement in

software, only if there is an overall decrease in total number of failures during life (*LA*) as well as an increase in *MTTF*. In the specific case *KLCC* stands for *KLOC* changed in code that we have computed as the sum of newly inserted lines with changed and deleted ones.

$$TF / KLOC = LA / KLCC \quad (4)$$

Failure Rate (FR)

Provides an intuitive indication of how the system is running as it is computed as the ratio between total number of failures during life (*LA*) and total life time (*TL*). If it is decreasing the process is running correctly and the product quality and reliability is increasing.

$$FR = LA / TL \quad (5)$$

Quality index (Quality)

This is a very simple indication of product quality if plotted for subsequent releases. It is simple to compute and can give a clear indication of how the product development process is running. If everything is correct it will present a monotonically decreasing trend.

$$Quality = Defect / KLOC \quad (6)$$

Availability (AV %)

This index provides an average indication of the product availability between failures. It is computed taking into account both *MTTF* and *MTTR*. If the development process was behaving correctly this indicator should decrease in time (it grows linearly with increasing *MTTF* and decreasing *MTTR*, while it increases in a logarithmic form with increasing *MTTF* and increasing *MTTR*) (Figure 5).

$$AV \% = (MTTF) / (MTTF + MTTR) \% \quad (7)$$

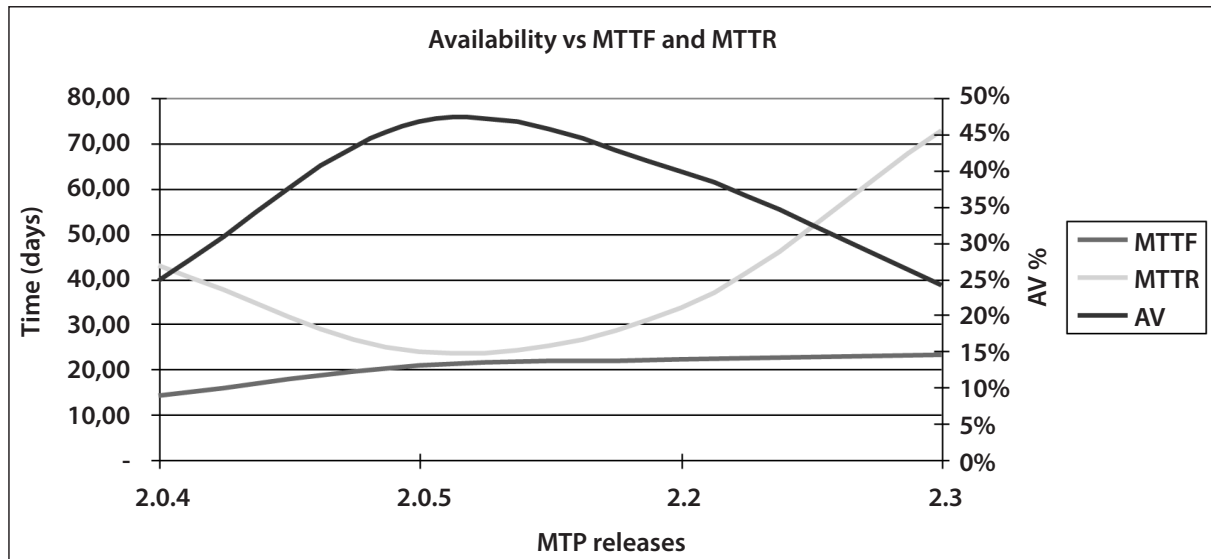


Fig.5. Average indication of the product availability between failures

Source: Fuschi, Tvaronavičienė (2014)

Efficiency Degree (ED)

It gives an indication of the efficiency of the product as it compares Product Change Rate (PCR) to total product life time (TL). The smaller this ratio the more efficient have been product revision / maintenance process.

$$ED = PCR / TL \quad (8)$$

Integration Failure Rate (IFR)

It gives an indication on the quality of the product under integration (Schneiewind 1997). It is computed as the ratio of number of failures during test (TA) to test time (T_{Test}). Retaining constant test time, a growth in its value denotes a decreased quality of product. While a decreasing value denotes an increased quality of the product.

$$IFR = TA / T_{Test} \quad (9)$$

Test Quality Index(TQI)

Provides an indication of the test process quality as it is derived from the ratio of number of failures during life (LA) with number of failures during test (TA). Its increase means that the quality of the overall test process is increasing.

$$TQI = LA / TA \quad (10)$$

Maintenance Test Time for KLOC (MTT)

It shows how test time (T_{TEST}) for KLOC changed

in code (KLCC) varies (Gall *et al.*1997). An increase in its value means a heavier test effort or a more difficult test phase. A decrease suggests a reduced test activity or a very large increase in changes. In any case such value has to be compared with integration failure rate (IFR) and failure rate (FR) in order to gain proper information on the quality of the test process.

$$MTT / KLOC = T_{TEST} / KLCC \quad (11)$$

Product Change Rate (PCR)

It gives an indication on the rate of change in the product (Gall *et al.*1997). To compute this index is required to perform the ratio between KLOC changed in code (KLCC) and total product KLOC (TPKL). KLCC has been considered as the sum of new, changed and deleted lines.

$$PCR = KLCC / TPKL \quad (12)$$

KLOC versus Function Points (FP)

Many indexes have been derived from KLOC data even if most of them could have also been derived from FP data. KLOC was preferred as FPs are more difficult to be evaluated. According to classics, FP can be roughly derived starting from KLOC using the following rule:

$$FP = LOC_{application} / [(LOC/FP) * BAF] \quad (13)$$

where *LOC/FP* is an average value dependent on the programming language used for development (in this case C programming language in which case the value is 120), while *BAF* is a backfiring adjustment factor related to complexity, here assumed to be 1.30 for complex applications.

As far as anomalies are concerned the most interesting information can be gained by looking at anomalies distribution per release, severity distribution, detection environment (internal test, external test or production). Examining anomalies distribution implies assessing how many new anomalies have been found for each release, how many have been inherited (anomalies already found but not yet solved when the new code was released) and how many have been solved. Release specific data can be seen as indicators of product behavior, while average values as process related indexes. Severity distribution is particularly important as it gives a more in depth view on the kind of problems experienced. In the examined case it was found that on average were encountered more medium severity anomalies than high ones, moreover it was found that no blocking anomalies were detected over a series of consecutive releases. Anomalies detection environment data can be used to evaluate development processes when compared with test effort and time. This also gives a direct indication on how difficult it is to replicate field condition while testing. Interesting data on the overall development process evolution could be gained by observing time distribution per phase as it allows monitoring maintenance impact in terms of development and test time over product life time.

Other relevant data on the development process could be gained by examining both effort distribution per phase and overall effort distribution. In particular, effort data should always be used when examining process phases duration and anomalies trends. This is due to the fact that maintenance process is made of recursive development steps followed by test and validation ones. If the elapsed time of a specific phases appears to have been squeezed, it will be necessary to verify if in the same phase efforts has been significantly increased in order to avoid output has been compromised. At the same time a final assessment of the overall effectiveness of adopted countermeasures can only be gained by assessing the quality of the phase outputs as measurable with the previously described indicators.

At first the attention focus was on index derivation and therefore tools like Excel[®] and SPSS[®] were used to process data extracted from metrics and effort database (implemented in Oracle[®]). Data extraction was achieved using Oracle ProC[®] and SQL tools. Most pre-processing computation was carried out in Oracle ProC[®] as it was allowing to select and process data on line. As a first step derived data were exported to plain field delimited ASCII files (with semicolons “;” as separator) and then imported in Excel[®] and SPSS[®] where further analysis and graphic generation could be easily accomplished. The second step was a further processing one in which retrieved data is indexed and graphically plotted using GDLIB[®] graphic utilities so as to enable plotted data usage in the designed web-based scoreboard system. Anomalies report, metrics and indexes suite could be tailored on demand thanks to the aforementioned process, and made available to a web server via a specifically available CGI interface. Users could then access data online via a simple browser.

Given the nature of available data and indexes it was possible to compute the average, the standard deviation, correlation and/or regression index. Such statistical operation could be request for all values or a specific subset. For example it would be possible to evaluate the correlation between MTTR and effort or between PCR, MTTR and Quality, thus allowing experienced users to get a quantitative evaluation of hypothesis arisen from metrics inspection and qualitative evaluation.

3. Conclusions

The provided case study proved that adopted quality control policies and data collection practices need to be constantly refined. It has also showed that statistical analysis has to be carried out on collected data in order to gain even more info from the past. The adoption of this monitoring of the software development and maintenance process (that was part of a quality assurance and testing automation effort) allowed passing from a ratio of 20% of anomalies identified in internal testing and 80% identified in external testing and/or production to a 80% identified in internal testing and 20% in external testing and/or production with a substantial increase in the management and customers satisfaction.

In conclusion based on the results of this experience it is possible to formulate the following recommen-

dition:

- any software development company should base its software process on a sound and reliable CM environment;
- different level of detail / aggregation should be available as different are the needs of professional involved in day by day operation;
- GQM paradigm should be used in defining indicators and what to track;
- metrics should be collected but they must be identified according to company's need;
- indicators should be easily derivable from collected metrics;
- no unnecessary overhead should be introduced in the monitored process.

In order to implement suggested improvements Big Data processing complexity has to be taken into account and Big Data processing processes controlled. For efficient control appropriate legal framework has to be developed, since absence of *ad hoc* institutional context (e.g. legal responsibility of software producer for its functioning efficiency) will retard economic growth, and as in our case, will impede sustainable development processes.

References

Antanavičienė, J. 2014. Foreign direct Investment: driving factors and outcomes for secure and sustainable development, *Journal of Security and Sustainability Issues* 3(3). DOI: [http://dx.doi.org/10.9770/jssi.2014.3.3\(6\)](http://dx.doi.org/10.9770/jssi.2014.3.3(6))

Baikovs, A.; Zariņš, I. 2013. Philosophical, legal and general issues of legal liability, *Entrepreneurship and Sustainability Issues* 1(1): 23–36. DOI: [http://dx.doi.org/10.9770/jesi.2013.1.1\(3\)](http://dx.doi.org/10.9770/jesi.2013.1.1(3))

Balkienė, K. 2013. Sustainable innovativeness: issues and public policy, *Journal of Security and Sustainability Issues* 3(2): 53–76. DOI: [http://dx.doi.org/10.9770/jssi.2013.3.2\(5\)](http://dx.doi.org/10.9770/jssi.2013.3.2(5))

Białoskórski, R. 2012. Cyberthreats in the security environment of the 21st century: attempt of the conceptual analysis, *Journal of Security and Sustainability Issues* 1(4): 249–260. DOI: [http://dx.doi.org/10.9770/jssi.2012.1.4\(2\)](http://dx.doi.org/10.9770/jssi.2012.1.4(2))

Bileišis, M. 2012. Empowering institutions: a case for connecting business and the academe through *phronesis*, *Journal of Security and Sustainability Issues* 1(3): 177–185. DOI: [http://dx.doi.org/10.9770/jssi/2012.1.3\(3\)](http://dx.doi.org/10.9770/jssi/2012.1.3(3))

Borseková, K.; Petříková, K.; Vaňová, A. 2012. The methodology of use and building competitive advantage on the regional level, *Journal of Security and Sustainability Issues* 2(1): 41–50. DOI: [http://dx.doi.org/10.9770/jssi/2012.2.1\(4\)](http://dx.doi.org/10.9770/jssi/2012.2.1(4))

Briand, L.; Morasca, S.; Basili, V. 1994. Goal-Driven Definition of Product Metrics Based on Properties, CS-TR-3346, UMI-ACS-TR-94-106.

Cugola, G.; Gresse, C.; Fusaro, P.; Fuggetta, A.; Lavazza, L.; Manca, S.; Pagone, M.R.; Ruhe, G.; Soro, R. 1997. An application of Goal-Oriented Measurement to the Evaluation of the Impact of Configuration Management, Metrics '97 Proceedings.

Dubauskas, G. 2011. International banks in the Baltic States: aspects of growth sustainability, *Journal of Security and Sustainability Issues* 1(2): 91–101. DOI: [http://dx.doi.org/10.9770/jssi.2011.1.2\(2\)](http://dx.doi.org/10.9770/jssi.2011.1.2(2))

Dudzevičiūtė, G. 2012. Conceptual approaches towards sustainability, *Journal of Security and Sustainability Issues* 1(4): 261–272. DOI: [http://dx.doi.org/10.9770/jssi.2012.1.4\(3\)](http://dx.doi.org/10.9770/jssi.2012.1.4(3))

Fuschi, D.L.; Tvaronavičienė, M. 2014. Monitoring service quality: methods and solutions to implement a managerial dashboard to improve software development. A case study in the banking and financial sector Paper to be presented at The First International Workshop on Real-time.

Big Data Analytics for Critical Infrastructure Protection (BIG4CIP 2014), 13-16 May 2014, Newcastle upon Tyne (UK). Available on the Internet: <http://www.big4cip.org/>.

Gall, H.; Jazayeri, M.; Klösch, R.; Trausmuth, G. 1997. Software evolution observation based on product re-release history, ICSM'97 Proceedings.

Hsinchun, C; Chiang, R. H. L.; Storey, V. C. 2012. Business intelligence and analytics: from Big data to Big impact, *MIS Quarterly* 36(4): 1165–1188.

Kaźmierczyk, J., 2012. Information technology systems and their impact on the employment level in the Polish banking sector, *Journal of Security and Sustainability Issues* 1(3): 187–195. DOI: [http://dx.doi.org/10.9770/jssi/2012.1.3\(4\)](http://dx.doi.org/10.9770/jssi/2012.1.3(4))

Lankauskienė, T.; Tvaronavičienė, M. 2012. Security and sustainable development approaches and dimensions in the globalization context, *Journal of Security and Sustainability Issues* 1(4): 287–297. DOI: [http://dx.doi.org/10.9770/jssi.2012.1.4\(5\)](http://dx.doi.org/10.9770/jssi.2012.1.4(5))

Laužikas, M.; Krasauskas, A. 2013. Impacts of sustainable structural growth on the economic performance of listed companies, *Entrepreneurship and Sustainability Issues* 1(2): 81–91. DOI: [http://dx.doi.org/10.9770/jesi.2013.1.2\(2\)](http://dx.doi.org/10.9770/jesi.2013.1.2(2))

Laužikas, M.; Mokšėckienė. 2013. The role of creativity in sustainable business, *Entrepreneurship and Sustainability Issues* 1(1): 10–22. DOI: [http://dx.doi.org/10.9770/jesi.2013.1\(2\)](http://dx.doi.org/10.9770/jesi.2013.1(2))

Mačiulis, A.; Tvaronavičienė, M. 2013. Secure and sustainable development: Lithuania's new role in taking the Presidency of the EU, *Journal of Security and Sustainability Issues* 3(2):5–13. DOI: [http://dx.doi.org/10.9770/jssi.2013.3.2\(1\)](http://dx.doi.org/10.9770/jssi.2013.3.2(1))

Makštutis, A.; Balkytė, A.; Tumulavičius, V. 2012. Security, Sustainability and Competitiveness: Benchmarking attempts, *Journal of Security and Sustainability Issues* 2(1): 5–12. DOI: [http://dx.doi.org/10.9770/jssi/2012.2.1\(1\)](http://dx.doi.org/10.9770/jssi/2012.2.1(1))

Malaiya, Y.K.; Denton, J. 1998. Estimating the number of re-

sidual defects [in software]. In *High-Assurance Systems Engineering Symposium, Proceedings*. Third IEEE International, 98–105.

Matei, A.; Savulescu, C. 2012. Towards sustainable economy through information and communication technologies development: case of the EU, *Journal of Security and Sustainability Issues* 2(2): 5–17. DOI: [http://dx.doi.org/10.9770/jssi.2012.2.2\(1\)](http://dx.doi.org/10.9770/jssi.2012.2.2(1))

Raišienė, A.G.; Jonušauskas, S. 2013. Silent issues of ICT era: impact of techno-stress to the work and life balance of employees, *Entrepreneurship and Sustainability Issues* 1(2):108–115. DOI: [http://dx.doi.org/10.9770/jesi.2013.1.2\(5\)](http://dx.doi.org/10.9770/jesi.2013.1.2(5))

Schneidewind, N. 1997. Measuring and evaluating process using reliability, risk and test metrics. *ICSM'97, Proceedings*.

Stańczyk, J. 2011. European security and sustainability issues in the context of current international environment, *Journal of Security and Sustainability Issues* 1(2): 81–90. DOI: [http://dx.doi.org/10.9770/jssi.2011.1.2\(1\)](http://dx.doi.org/10.9770/jssi.2011.1.2(1))

Vosylius, E.; Rakutis, V.; Tvaronavičienė, M. 2013. Economic growth, sustainable development and energy security interrelation, *Journal of Security and Sustainability Issues* 2(3): 5–14. DOI: [http://dx.doi.org/10.9770/jssi.2013.2.3\(1\)](http://dx.doi.org/10.9770/jssi.2013.2.3(1))

Wahl, M.; Prause, G. 2013. Toward understanding resources, competencies, and capabilities: business model generation approach, *Entrepreneurship and Sustainability Issues* 1(2): 67–80. DOI: [http://dx.doi.org/10.9770/jesi.2013.1.2\(1\)](http://dx.doi.org/10.9770/jesi.2013.1.2(1))