INFORMATION MANAGEMENT TOOLS FOR IMPLEMENTING AN EFFECTIVE ENTERPRISE BUSINESS CONTINUITY STRATEGY

Athanasios Podaras, Klára Antlová, Jiří Motejlek

Introduction

The modern enterprise environment, in both the private and the public sector, is mainly characterized by the strong presence of information technologies, complex business functions and multiple software applications and electronic platforms. As a consequence, more and more IT managers are concerned about the possibility of system failure in case of an emergency or a crisis situation, which could result to the interruption of the critical business activities of the enterprise and, respectively, trigger significant economic damage.

One of the most vital, and nowadays obligatory, tasks of modern enterprises and organizations is therefore the development and the establishment of an efficient and effective *Business Continuity Management* (Asnar & Giorgini, 2008). This is considered to be one of the key areas of ICT Competencies (Antlová, Popelinský & Tandler, 2011) by many experts. Its imperative implementation in terms of enterprise operational policy and strategy, stems from various unexpected and forecasted natural, human or even technical threats that many organizations and countries have experienced within the recent years. Especially in Central European Countries, a series of natural disasters, mainly floods, took place. Austria, the Czech Republic, Hungary, Poland and Slovakia were all affected (Skrbek, 2011). Such situations emerged in 2002, 2010, 2011 and 2013. In such circumstances, immediate information system recovery, which aims to the minimization of both human and financial losses, is drawn from a cautiously documented *Business Continuity Plan* (Doughty, 2001). The criticality estimation of business functions and their corresponding applications is a fundamental task to be solved by IT managers when planning business continuity testing exercises that concern their

unit. The organization and implementation of successful business recovery tests presupposes the creation of a detailed and accurate documentation of the critical business functions and their corresponding applications in the *Business Impact Analysis* (BIA) template (NIST, 2010).

Throughout the present paper the authors attempt to delineate a standard method for developing efficient recovery exercises for the most crucial IT business functions and processes of an enterprise, in order to formulate an effective Business Impact Analysis (BIA) document and, subsequently, propose an efficient recovery strategy. The derived strategy is based on three main steps; firstly, the approximate estimation of the criticality level of the business functions, secondly, the formulation and execution of recovery exercises based on appropriate best and worst case scenarios which may coexist with unexpected interruption of the function, and, finally, the calculation of the time – effort which is demanded in order to recover the function.

The scoring of a function's criticality is estimated according to the rules of the Use Case Points method (Karner, 1993; Diev, 2006; Kamal, Ahmed & El-Attar, 2011), after considering technical, environmental and the newly introduced unexpected factors, which could significantly delay the estimated recovery timeframes which are defined by the business continuity policy and the corresponding impact value levels. Researchers from academia as well as industry have shown interest in the Use Case based approaches because of the promising results obtained along with their early applicability (Nagar & Dixit, 2012). The weights of the factors, which strongly affect the recovery procedure, are assigned according

to a standard mathematical approach which is entitled Rank Order Centroid (ROC) (Barron & Barrett, 1996). This approach increases the method's flexibility and expandability. The recovery exercises are designed according to scenarios that have not so far been considered by the standard business continuity strategies. Thus, the time - effort required in order to recover a process in extreme and unforeseen emergency incidents, is expected to vary from the Rational Time Objective (RTO) and Maximum Accepted Outage (MAO) timeframes suggested by various Business Continuity experts. The estimated time – effort required to "bring back to life" a vital IT business function, is utilized as the key indicator for applying the ideal recovery testing approach for a specific IT business function. The delineated method is entitled Business Continuity Testing Points (BCTP) and it is developed to be utilized within an efficient and effective business continuity strategy.

1. Implementation of Functional Business Continuity Tests within a Defined Risk Management Policy

From the point of view of scientific literature and industrial practice BC addresses questions of how to handle risk issues in the case that critical business processes of an organization fails (Doughty, 2001). A required task by the organizations is the recovery of a business function within the desired Recovery Time Objective (RTO) or even the Maximum Acceptable Outage (MAO) (NIST, 2010), as they are determined by the Business Continuity team and documented in the BIA template.

Historically, BC addressed IT processes, later on, business processes came up as the final purpose of their supporting IT processes (Zambon, Bolzoni, Etalle & Salvato, 2007; Asnar & Giorgini, 2008). The importance of Business Continuity Testing is outlined and thoroughly analyzed by many experts. It is precisely stated that organizing regular exercises, such as desktop and simulation drills, is the only way to discover gaps and address them (Sapriel, 2003).

Creating a Business Continuity Plan is a long-term process and companies should review the existing documentation as an ongoing project (Kepenach, 2007; Lam, 2002). The actual purpose of testing is to achieve organizational acceptance that the solution satisfies the recovery requirements. Plans may

fail to meet expectations due to insufficient or inaccurate recovery requirements, solution design flaws or solution implementation errors (Crisis Solutions, 2008). The differentiation of critical (urgent) and non-critical (non-urgent) organization functions/activities is the core task of BIA. Critical functions are those whose disruption is regarded as unacceptable. A function may be considered critical in the case that it is imperative due to specific law or if its recovery is very costly.

The current paper focuses on the IT department's successful documentation and testing of the most critical functions and processes. Hardware and software should support critical business functions, so the IT functions, in large part, will be driven by all the other departments. The payroll system might be considered critical by Human Resources and Customer Relationship Management. Similarly, manufacturing may indicate that without the automated inventory management system, production cannot be initiated. Therefore, the IT department's critical business functions are driven externally, to a large degree (Snedaker, 2007). However, a successful IT business continuity management policy should focus on the immediate recovery of the factually most important operations of the enterprise, defined by the ISO 22301:2012 as *Minimum Business Continuity Objective*, briefly stated as MBCO (BSI, 2012). The primary task of the current work is the determination of the inclusion or exclusion of a business function in the MBCO, with respect to its recovery exercise category, its Criticality Level and finally the Recovery Time Effort achieved throughout recovery exercises which are based on extreme and unexpected crisis scenarios.

1.1 Exercise Categories According to the Business Standard Institute

British Standards Institutution (2012) distinguishes 3 basic categories of exercises implemented in terms of the Business Continuity Strategy:

Tabletop: they typically involve a small number of people and participants, who work through a simple scenario, discuss specific aspects of the plan and only a few hours are consumed.

Medium: conducted within a "Virtual World" and bring together several departments, teams or disciplines.

Complex: also occur within a "Virtual World", but maximum realism is essential and duration is unknown.

The results of insufficient and poor testing of software applications are known and obvious within the enterprise environment. Test engineering is seldom planned for in most organizations and as a result, products enter the market insufficiently tested. Negative customer reactions and damage to the corporate image is the natural consequence (Nageswaran, 2001). Similarly, the test engineering process for critical business functions is essential from the business continuity management standpoint, since the negative effects caused by an unsuccessful response to a real crisis event will be an established fact for the enterprise. Consequently, according to the above statement, all business functions should be tested regularly so that all the involved staff is prepared for a real crisis event. The idea behind the proposed contribution is that test success is based on the mapping of each IT business function with the most suitable of the aforementioned exercise types after determining its impact value level. The way that the mapping is performed is depicted in the following section (Tab. 3).

1.2 Impact Value Levels of IT Business Functions

Darril Gibson (2010) indicates the impact value level of each business function according to its accepted downtime period without causing negative effects to the enterprise or the organization. The four levels of impact value are:

- **Level 1**: business functions should operate without any interruption. Online systems must be available 24 hours per day and 7 days per week.
  Maximum Acceptable Outage (MAO) = 2 hours.
  Recovery Time Objective (RTO) < 2 hours.
- **Level 2**: business processes can survive without the business function for a short amount of time.
  Maximum Acceptable Outage (MAO) = 24 hours (1 day).
  Recovery Time Objective (RTO) < 24 hours.
- **Level 3**: business processes can survive without the business function for one or more days.
  Maximum Acceptable Outage (MAO) = 72 hours (3 days).
  Recovery Time Objective (RTO) < 72 hours.

- **Level 4**: business processes can survive without the business function for extended periods.
  Maximum Acceptable Outage (MAO) = 168 hours (1 week).
  Recovery Time Objective (RTO) < 168 hours.

The above mentioned levels will be applied to the new proposed Business Continuity Testing Points method. The model is based on the logic that the more critical the business function the less time should be spent on its recovery. According to this logic, the responsible IT manager of the specific business function will be able to classify it to the appropriate exercise category.

1.3 The Use Case Points Method

One of the most crucial activities in the software development process is the identification of system functional requirements. A popular way to capture and describe those requirements is through the UML Use Case Models (Cruz, Machado & Santos, 2014). It is especially valuable in the context of early size measurement and effort estimation, because it employs use cases as an input (Nageswaran, 2001). The current work aims to measure complexity and effort for recovering an interrupted business process. Consequently, taking into consideration that the Business Continuity strategy is also a requirement analysis procedure, business process recovery complexity and effort estimation should be based on an approach as the Use Case Points, which finds its roots in the Use Case Model, in order to formulate a Business Impact Analysis document and estimate the Recovery Time Objective and Maximum Acceptable Outage for the reestablishment of a given IT business function during an emergency incident.

The reason and the need for introducing the new method, is to avoid the manual vague estimation of these values, which is solely based on the IT managers' practical experience. Before analyzing the Business Continuity Testing Points new model, a reference to the Use Case Points method is considered an important part of the present work. The UCP method is a tool to perform Effort Estimation for Software Development. The Use Case Points method is divided into 3 basic parts:

Part 1: Classification of Actors and calculation of *Unadjusted Actor Weights (UAW) value.*

Part 2: Classification of Use Cases and calculation of *Unadjusted Use Case Weight (UUCW) value.*

Part 3: Classification of *Technical* and *Environmental Factors* (TF, EF), Calculation of *Technical Complexity* and *Environmental Complexity Factors* (TCF and ECF) values, and the final calculation of Adjusted Use Case Points (UPC) value which is then utilized for the final effort of software and system development.

Classification of Actors
In the Use Case Points Method, the Actors are distinguished in 3 categories: a Simple Actor (Karner, 1993; Banerjee, 2001; Kusumoto, Matukawa, Inoue, Hanabusa & Maegawa, 2004; Ochodek, Nawrocki & Kwarciak, 2011) represents another (or external) system with a defined Application Programming Interface, API, an Average Actor is another system interacting through a protocol such as TCP/IP, and a Complex Actor may be a person interacting through a GUI or a Web page. The corresponding *Weighting Values* of the Actors are 1, 2 and 3.

By counting the number of Actors of each kind (complex, average or simple), multiplying each total by its weighting factor and finally adding up the products, we calculate the total *Unadjusted Actor Weights*, briefly mentioned as UAW. The result of the calculation is provided by Eq. (1):

$$UAW = \sum_{i=1}^{n}(A_i \times W_i) \qquad (1)$$

where n = Number of Actors, A_i = Actor Type i, W_i = Actor's i Weight Value.

Classification of Use Cases
Apart from the Actors, Use Cases are also distinguished in 3 categories: a *Simple Use Case* in which *Number of Transactions is <= 3,* an *Average Use Case* in which *Number of Transactions is >=4 and <=7*, and a *Complex Use Case* in which *Number of Transactions is >7*. The corresponding *Weighting Values* of the Use Cases are 1, 2 and 3.

By counting the number of Actors of each kind (complex, average or simple), multiplying each total by its weighting factor and finally

adding up the products we calculate the total Unadjusted Use Case Weights, briefly mentioned as UUCW. The result of the calculation is provided by Eq. (2):

$$UUCW = \sum_{i=1}^{n}(U_i \times W_i) \qquad (2)$$

where n = Number of Use Cases, U_i = Type of given Use Case i, W_i = Use Case i weighting value. The UAW is added to the UUCW to get the *Unadjusted Use Case Points* (UUCP) from Eq. (3)

$$UUCP = UAW + UUCW \qquad (3)$$

Technical and Environmental Factors
In the Use Case Points method, apart from the computation of the UUCP value, various Technical (Tab. 1) and Environmental Factors (Tab. 2) are considered and computed with respect to Software Application complexity. After their computation, the Adjusted Use Case Points (UPC) are calculated with the help of a special equation, in which Unadjusted Use Case Points value (UUCP), Technical Complexity Factors (TCF) value and Environmental Complexity Factors (ECF) value are multiplied. The formula applied for calculating the Technical Complexity Factor (TCF), is provided by Eq (4):

$$TCF = 0.6 + (0.01 \times TFactor) \qquad (4)$$

after multiplying the value of each Technical Factor (Tab. 1) by its corresponding weight we then add all these numbers to get a sum called the *TFactor*.

In the same way Eq. (5) is applied for calculating the Environmental Complexity Factor (ECF).

$$ECF = 1.4 + (-0.03 \times EFactor) \qquad (5)$$

after multiplying the value of each Environmental Factor (Tab. 2) by its corresponding weight we then add all these numbers to get a sum called the *EFactor*.

The final calculation of the Adjusted Use Case Points (UPC) is provided by Eq. (6):

$$UPC = UUCP \times TCF \times ECF \qquad (6)$$

The estimation effort is the final part of the Use Case Points method. By multiplying the specific value (man-hours) by the UCP, the estimated effort can be obtained (Banerjee, 2001; Schneider & Winters, 1998). A factor of 20 man-hours per UCP for a project is suggested by Karner (1993).

2. The Business Continuity Testing Points (BCTP) Method

The BCTP approach, was derived by a proven and utilized in multiple IT projects method, in order to be applied to the Business Continuity Management theory. More precisely, a standard requirement analysis and practically implemented method, such as the Use Case Points approach, was required as a pilot method for the construction of a similar model, which is entitled by the authors as the *Business Continuity Testing Points* approach. The Use

Tab. 1:   **Technical factors**

Factor	Description	Weight
T1	Distributed Systém	2
T2	Response adjectives	2
T3	End – User efficiency	1
T4	Complex Processing	1
T5	Reusable Code	1
T6	Easy to install	0.5
T7	Easy to Use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Security features	1
T12	Access for third parties	1
T13	Special Training Required	1

Source: Karner (1993)

Tab. 2:   **Environmental factors**

Factor	Description	Weight
F1	Familiar with RUP	1.5
F2	Application Experience	0.5
F3	Object – Oriented experience	1
F4	Lead Analyst capability	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part – time workers	-1
F8	Difficult programming language	2

Source: Karner (1993)

Case Points method focuses on software size estimation. The same concept is modified in order to be applied in order to estimate the following elements: a) the criticality of a business function and its dependent business processes and the decision making on whether to include it or not in the MBCO, b) its complexity and required effort in order to recover it in comparison with the proposed by experts RTO and MAO values and c) its classification to an exact recovery exercise category. As a consequence, the results of the calculations will enable IT managers decide about relevant business continuity testing policies which will support immediate process recovery in case of a crisis situation.

2.1 Goal of BCTP Contribution

In modern enterprises RTO and MAO timeframes are estimated based on the employers' everyday operational experience within the organization. This fact leads to erroneous BIA documentation of high priority functions and processes and, moreover, to poor recovery testing implementation of hypothetical crisis scenarios proposed by the Business Continuity managers.

The primary goal of this contribution is the proposal of a model which will overcome poor Business Continuity strategies, since it is based on a mathematical estimation of the necessary timeframe needed to recover an IT process after an unexpected outage, and not on the experience and the personal aspects of the IT managers, who consider almost all business functions to be critical and worthy to include in the MBCO. By implementing the BCTP approach, the MBCO will be formulated by accurately defined business functions. The second element, which constitutes the value of the currently developed contribution to the field of Information Management, is that it is based on the Use Case Points method, which is widely accepted and practically utilized by IT experts.

2.2 Delineation of the Business Continuity Testing Points Approach

As already stated, the present paper delineates the formulation of the standard BCTP contribution, including the Assessment Values of all factors. The model is inspired by the UML Use Case Points method. The Use Case Points method, which estimates software complexity, has been adjusted for the needs of the business

function recovery complexity estimation. The BCTP method demands the consideration of technical, human, business and unforeseen factors that can affect the recovery of a given business function and its dependent processes. The adjustment of the Use Case Points method demanded changes to the Technical Factors, because the mew method has a more operational character in contrast to the Use Case Points method. However, some factors of the Use Case Points theory had to be included in the BCTP method. For instance, *Security Features* is a Technical Factor which affects not only Software Complexity, but also Recovery Complexity. In the former case, more lines of code may be demanded, while in the latter occasion the presence of more complicated security software tools may prolong the recovery procedure.

Another difference between the two methods is the mapping of the Actors of the Use Case Points, to the Actor Types 1 (Human Level) and 2 (Application Level). The classification of the Actor Types, is imperative to the present model, since it aims at the evaluation of a recovery process, which is strongly affected by both the human reaction, as well as the technical infrastructure state. Furthermore, in the BCTP approach, Business Processes are utilized instead of Use Cases, for the calculation of the Unadjusted Business Function Recovery Points (UBFRP) element. As in the case of Use Case Points, the Calculation of the value of the Unadjusted Weights, regarding both Actor Types is calculated by the following equations:

$$UHW = \sum_{i=1}^{n} (A1_i \times W_i) \qquad (7)$$

where UHW is the *Unadjusted Human Weight* value, $A1_i$ is *Human Level Actor* i, and W_i is the Actor's i[th] Weight, in order to compute Human Level Actors, and similarly,

$$UAPW = \sum_{i=1}^{n} (A2_i \times W_i) \qquad (8)$$

where $UAPW$ is the *Unadjusted Application Weight* value, $A2_i$ is Application Level Actor i, and W_i is the Actor's Weight.

Human Level Actors are classified according to the following manner:

- **Simple Human Level Actor**: Employees of an IT department who are members of the recovery team. The weight value assigned to the personnel of this level is 0.5.
- **Average Human Level Actor**: IT subdirectors or supervisors of an IT Section who are members of the recovery team. The weight value assigned to the personnel of this level is 1.
- **Complex Human Level Actor**: IT managers on top of the IT Section or a corresponding division in a company who are leaders of the Business Continuity team. The weight value assigned to the personnel of this level is 1.5.

Similarly, Application Level Actors are classified in the following way:

- **Simple Application Level Actor**: A system or application which is used locally, is not connected with other applications, and does not perform critical transactions, i.e. electronic calendar, and its immediate recovery is not required (> 3 days). The assigned weight value is 0.5.
- **Average Application Level Actor**: A system which is used by a small number of users (i.e. one department), it is connected with few other applications of the organization and its recovery should not be immediate but should be achieved within a specific time space (1-3 days) (i.e. a database installed in a local pc-server accessed by a small number of users). The assigned weight value is 1.
- **Complex Application Level Actor**: A system or application which is connected with many other systems of the organization, performs critical transactions (i.e. a web interface which is utilized by customers in order to perform online transactions), and its recovery should be achieved immediately (1-2 Hours). The assigned weight value is 1.5.

The total score of Unadjusted Actors' Weights (, is provided by the formula:

$$TUAW = UHW + UAPW \qquad (9)$$

Finally, the calculation of the Unadjusted Business Process Weights (UBPW), is performed via the following equation:

$$UBPW = \sum_{i=1}^{n} (BP_i \times W_i) \qquad (10)$$

where n = Number of Business Processes, (BP_i) is the Type of the given Business Process i and W_i is the Weight of the corresponding Business Process.

For the classification of the Business Process Types, a similar approach to Use Case Classification in the Use Case Points method, is performed. Business Processes are divided in 3 categories: a *Simple Business Process* in which *Number of Business Activities is <= 3,* an *Average Business Process* in which *Number of Business Process Activities is >=4 and <=7,* and a *Complex Business Process* in which *Number of Business Activities is >7.* The corresponding *Weighting Values* of the Use Cases are again 0.5, 1 and 1.5.

Finally, the Unadjusted Business Function Recovery Points (UBFRP) value can be calculated according to the following formula:

$$UBFRP = TUAW + UBPW \qquad (11)$$

The score of the Unadjusted Points should indicate whether further analysis is demanded in order to define the precise Impact Value Level of the Business Function. The score is obtained after defining specific complexity scenarios with regard to a specific business function. Through the scenarios, the number and category of the involved Human and Application Level Actors is illustrated and the UHW and UAPW values are calculated. Finally, the corresponding TUAW is defined. In a similar way the number and category of involved processes is utilized to define the UBPW value. The final step of the procedure includes the estimation of UBFRP value based on the aforementioned scenarios and calculations. The following table (Tab. 3) includes the most representative recovery complexity scenarios and the corresponding UBFRP scores.

For business functions with a low score of Unadjusted Points (1 $\leq$ UBFRP $\leq$ 15), no further analysis is demanded by the second part of the model. Instead, a direct determination of the Impact Value Level of the function can be implemented. The relevant function should not be included in the Minimum Business Continuity Objective (MBCO) and will not be

Tab. 3: Total score of unadjusted points based on simple, average and complex recovery scenarios

Scenario	Human Level Actors	UHW	Application Level Actors	UAPW	TUAW	Business Process Category	UBPW	UBFRP
Simple	1, 1, 1	3	1, 1, 1	3	6	1, 1, 1	3	9
Average	1, 2, 2	4.5	1, 2, 2	4.5	9	2, 2, 2	6	15
Complex	1, 3, 3	6	1, 3, 3	6	12	3, 3, 3	9	21

Source: own

Tab. 4: Impact value levels and exercise categories of business functions

Business Function	Impact Value Level	Exercise Category	Included in MBCO (urgent)
BF1	Level 1	Complex	YES
BF2	Level 2	Complex	YES
BF3	Level 3	Medium	NO
BF4	Level 4	Tabletop	NO

Source: own

tested as urgent or very critical. By obtaining the value of Unadjusted Points, the first level evaluation of function criticality has been terminated. Functions that are not urgent, can be simply documented by the BIA template either with Impact Value Level3 and a Medium Exercise category or with Impact Value Level4 and a Tabletop Exercise category (Tab. 4).

On the other hand, as far as more complicated business functions with a higher score of Unadjusted Points (UBFRP) are concerned, a more detailed procedure should take place in the second part of the model. An exact determination of their precise Impact Value level is demanded for defining the respective type of exercise, which in this case should be complex. Such functions should be included in the MBCO concept, which means that their immediate recovery after a system failure during a crisis event is essential. The Exercise category of critical functions should be complex, because the more limited the system recovery time is, the more resources (human, technical and infrastructure) are demanded. The algorithmic steps which should be implemented in terms of classifying IT business functions to the appropriate and defined by the B.S.I. exercise category, are the following:

Step 1: Definition of Actor Types for both levels (Human and Application).

Step 2: Calculation of Unadjusted Actor Weights of Type 1, which are named Unadjusted Human Weights (UHW) and Unadjusted Actor Weights of Type 2, the so – called Unadjusted Application Weights (UAPW). The Total number of Unadjusted Actor Weights (TUAW) is provided by adding up the weight values of the two Actors.

Step 3: Calculation of Unadjusted Business Process Weights (UBPW).

Step 4: Calculation of Unadjusted Business Function Recovery Points (UBFRP).

Step 5: Definition of the Impact Value Level and determination of whether a business function is included in the MBCO, by considering its value of Unadjusted Business Function Recovery Points (UBFRP).

Step 6: In case that a Function is not included in the MBCO, then the Impact Value Level is 3 or 4 and the corresponding Exercise category is either *tabletop* or *medium*. The accurate definition of levels and exercise categories is not important, since the enterprise can survive without the specific function for a few days. However, if an exact definition of the above elements is desired by the organization,

the process will be the same as it is in the case of complex IT functions that are included in MBCO.

Step 7: In case that a Function is included in the MBCO then the exact impact value level (1 or 2) must be defined. Moreover, the corresponding exercise category is characterized as complex. The exact Impact Value level is calculated by considering Technical Recovery Factors (TRF),

Environmental Recovery Factors (ERF) and Unexpected Recovery Factors (URF).

The Impact value level depends on the Adjusted Business Function Recovery Points (ABFRP) value and the total Recovery Time Effort (RTE) value that will be computed.

The above analyzed steps, are schematically represented via the below UML Activity Diagram (Fig. 1).
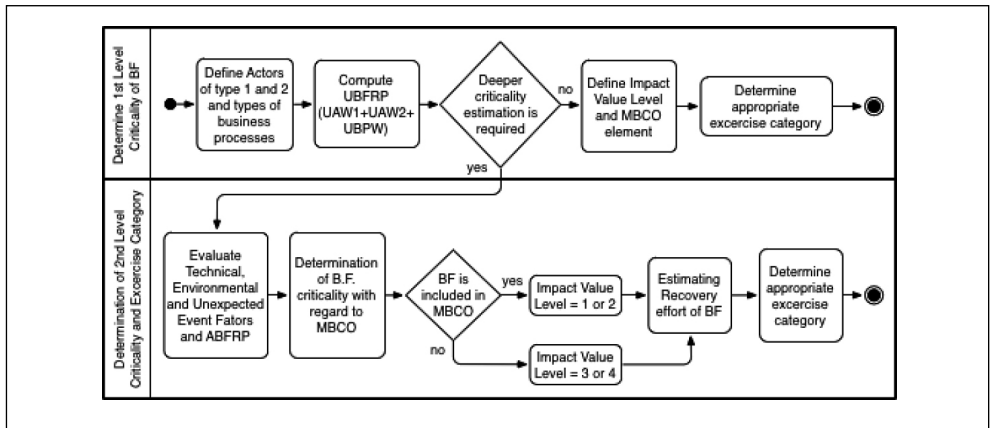
Source: own

The second part of the method includes the modified lists of the Technical Recovery Factors (TRF) and the Environmental Recovery Factors (ERF) (Tab. 5, Tab. 6).

Technical Recovery Factors (TRFs) mainly relate to the influence of the Technical Entities, which are involved in the business function, on the recovery process, which should be recovered after outage. Technical factors refer to applications, platforms, interfaces, hardware and network components which are related to the business process.

On the other hand, **Environmental Recovery Factors (ERFs)** mainly relate to the effect of Human Entities and their behavior on the recovery process. Human Entities can be users, business experts, a business recovery team, business owners, consultants and many other people who are responsible for the operation of the business function.

The factors which belong to the aforementioned categories, stem from the

Use Case Points factors as well as the factors considered to formulate the Business Continuity Exercises checklist document as it is proposed by the Information and Technology Services (ITS) – University of Michigan (2014). The list of factors is formulated accordingly to adjust to the Business Continuity Testing Points concept.

Another issue which had to be solved by the authors of the current work, was the derivation of the weight values of the factors. The authors decided not to use exactly the same values as in the Use Case Points method, but to develop new values. The main reason is that the Use Case Points theory deals with software size complexity estimation, while the Business Continuity Testing Points is aimed to the estimation of Business Process recovery complexity. The weight values are assigned to each factor according to the Rank Order Centroid (ROC) approach (Barron et al. 1996). Three main reasons motivated the authors in order to utilize the ROC approach for assigning

weight values to the Factors, and that is a) accuracy, b) simplicity and c) flexibility.

The method was selected among other mathematical approaches considering the fact that it is proposed by experts as the most suitable for Multicriteria Decision – Making (MCDM). In her thorough comparative study, Rozkowska (2013) underlines that "Several methods for selecting approximate weights, including equal weights (EW), rank sum (RS), rank reciprocal (RR),) and rank-order centroid (ROC) weights, have been proposed and evaluated. A common conclusion of these studies is that ROC weights have an appealing theoretical rationale and appear to perform better than the other rank-based schemes in terms of choice accuracy".

Furthermore, the ROC approach is a simple way of giving weight to a number of items ranked according to their importance (Bagla, Gupta & Kukreja, 2011). The decision makers can usually rank items much more easily than give weight to them. This method takes those ranks as inputs and converts them to weights for each of the items, according to the following formula (Bagla et al., 2011):

$$Wi = \frac{1}{m} \times \sum_{n=1}^{m} \frac{1}{n} \quad (12)$$

where W_i is the Weight Value of the ith item, and m denotes the number of items (factors).

Finally, the ROC approach enhances the flexibility of the entire Business Continuity Testing Points method. Since the sum of the weight values must be always equal to 1, no matter the number and the decided ranking order of the factors, the estimated effort required to recover the business function is not affected. It can be thus stated that the model can be adjusted to the needs of every business continuity testing project. Thus, it is obvious that due to the flexibility of the contribution, the ranking order of all Technical, Environmental and Unexpected Factors may be different from the one illustrated in the current paper, since it will not affect the recovery complexity estimation procedure. For deriving reasonable recovery time effort values, the assigned weight values in the present model are multiplied by 10. As a result the final weight values are normalized on a 0 to 10 interval scale.

It can be noticed (Tab. 5) that Technical Recovery Factors and Environmental Recovery Factors (Tab. 6) are divided into 2 basic categories:

Category 1: *Factors with an Ascending Scale of Assessment Values:* In this category, according to the model, the higher the assessment value of the factor the higher the degree of influence that the factor has on the recovery process. The factors with a low level of assessment value are marked with 1 and the factors with the highest influence on the recovery process are marked with 4. The factors are evaluated according to either a 4-level scale or with a 2-level scale. The Scale is determined according to the type of the considered factor. The type of factors can be boolean (YES/NO, i.e. TRF6, ERF7) or non-boolean (i.e. TRF1). The former require a 2-level scale assessment values while the latter require a 4-level scale. Boolean type factors indicate either positive or negative effects on the business function's recovery procedure. Thus, the existence of intermediate assessment values is avoided by the authors.

Category 2: *Factors with a Descending 4-level Scale of Assessment Values:* In this category according to the model, the higher the value of the factor the lower the degree of influence that the factor has on the recovery process (i.e. TRF9, ERF2).

The derivation of formulas which should calculate the Technical, Environmental and also Unexpected Recovery Factors, is a crucial part of the current paper. The formula, which provides the average TRF value, which is similar to equation 4 which provides the value of the TCF, is the following:

$$TRF = c_1 + \frac{1}{c_2} \times \sum_{i=1}^{n} \left(\frac{Fi_{max} + Fi_{min}}{2} \times W_i \right) \quad (13)$$

where Fi_{max} and Fi_{min} are the maximum and minimum values of a recovery factor i, W_i is the Weight Value of the specific factor, c_2 is speed of increasing recovery complexity and c_1 is a correcting constant.

According to the Use Case Points model, average recovery complexity should equal to 1. Therefore, we can write:

$$1 = c_1 + \frac{1}{c_2} \times \sum_{i=1}^{n} \left(\frac{Fi_{max} + Fi_{min}}{2} \times W_i \right) \quad (14)$$

Tab. 5: **Weights and assessment values of technical recovery factors**

Factor	Description	Weight	Assessment Values			
Factors with Ascending Scale of Assessment Values						
			1–2	**3–4**	**5–6**	**>6**
TRF1	Application's communication or dependency on other systems/ applications (Number of applications)	2.4	1	2	3	4
			Low	**Middle**	**High**	**V. High**
TRF3	Complexity of Business Tasks Supported	1.3	1	1	3	4
			Low	**Middle**	**High**	**V. High**
TRF4	Functional/Business Area (Importance/Criticality)	1	1	2	3	4
			On-Site		**Off-Site**	
TRF5	Resources and Records Stored Off-Site or On-Site	0.8	1		4	
			Internal		**External**	
TRF7	Business Function Type	0.6	1		4	
			Low	**Middle**	**High**	**V. High**
TRF8	Security Features (Protection Level)	0.5	1	2	3	4
			No		**Yes**	
TRF10	Third Party Users Involved	0.3	1		4	
			No		**Yes**	
TRF13	Extreme/Special Knowledge Required	0.1	1		4	
Factors with Descending Scale of Assessment Values						
			Low	**Middle**	**High**	**V. High**
TRF2	Easy to Restore Lost Data	1.6	4	3	2	1
			No		**Yes**	
TRF6	Exists Backup Site	0.7	4		1	
			Low	**Middle**	**High**	**V. High**
TRF9	Easy to Process Application/ System	0.4	4	3	2	1
			No		**Yes**	
TRF11	Routine Business Function	0.2	4		1	
			Low	**Middle**	**High**	**V. High**
TRF12	Required Level of IT skills	0.1	4	3	2	1

Source: own

The above equation can be written in the following way:

$$c_1 = 1 - \frac{1}{c_2} \times \sum_{i=1}^{n} \left(\frac{Fi_{max} + Fi_{min}}{2} \times W_i \right) \quad (15)$$
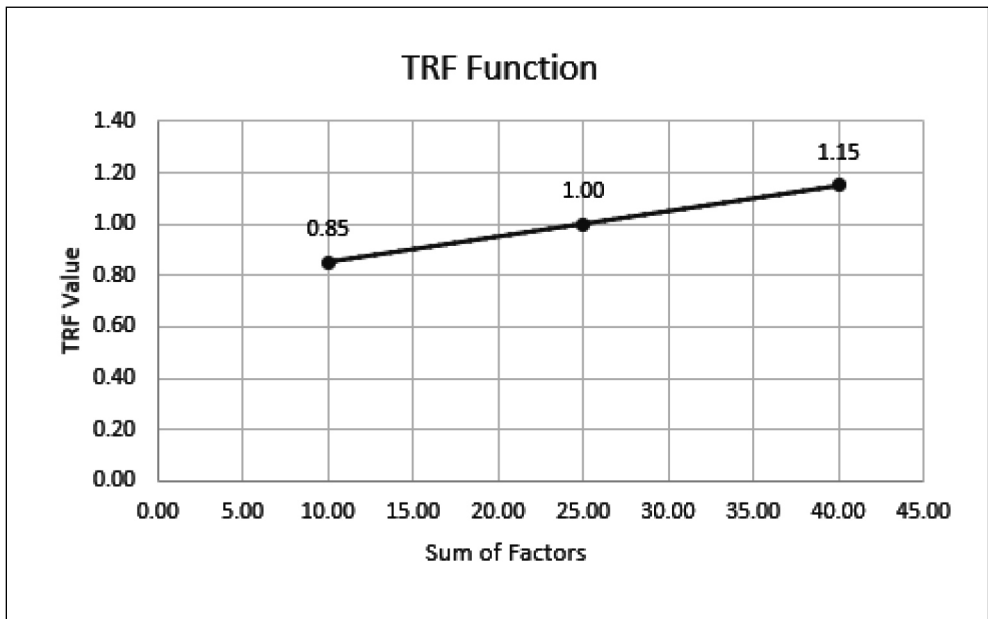
If the speed of increasing recovery complexity is the same as the speed of technical complexity in the Use Case Points method, we can compute c_2 value ($c_2 = 100$) and get the following formula:

$$TRF = 0.75 + \frac{1}{100} \times \sum_{i=1}^{n} (F_i \times W_i) \quad (16)$$

The speed of Recovery complexity can be assigned with the same value as the speed of

Fig. 2: **Graph of TRF fiction**

TRF Function

Source: own

technical complexity due to the direct relation between the System Complexity and Recovery Time. Laird and Brennan (2006), indicate that system complexity is among the factors that impacts the length of an outage. With system complexity Laird and Brennan refer to all technical aspects of a system, including software and hardware elements. Moreover, in the study of the National Institute of Standards and Technology (NIST, 2010), it is underlined that "Depending on the size and complexity of a system, recovery could take several days to weeks to complete".

Accordingly, the derived formula for the calculation of the ERF value is the following:

$$ERF = 0.75 + \frac{1}{100} \cdot \sum_{i=1}^{n} (F_i \times W_i) \quad (17)$$

where F_i is the value of a recovery factor and W_i is the Weight Value of the specific Factor.

However, the primary and most innovative element of the current method is the introduction of a third category of factors, entitled **Unexpected Recovery Factors (URFs)**. The existence of this category is considered to be indispensable, due to the business continuity and system recovery concept (Tab. 7).

URFs mainly relate to unplanned and unpredictable situations and scenarios that may emerge during the recovery process of a business function, and may significantly delay the process by exceeding the RTO and MAO values. The URFs are also divided in Factors with both an Ascending and a Descending Scale of Assessment values. The ranking order is also flexible, like the Technical and Environmental Factors.

The concept of the URFs is proposed by the authors as a tool for creating scenarios for complex recovery exercises and for the most critical business functions of an enterprise. The logic is to consider an approximate unpredictable recovery time deviation from the RTO and MAO values. Consequently, Recovery Scenarios can be *Simple, Average*

Tab. 6:   **Weights and assessment values of environmental recovery factors**

Factor	Description	Weight	Assessment Values			
Factors with Ascending Scale of Assessment Values						
ERF1	Customer Service Direct Effect	3.4	**Low**	**Middle**	**High**	**V. High**
			1	2	3	4
Factors with Descending Scale of Assessment Values						
ERF2	Familiar with Business Recovery Procedures	2.2	**Low**	**Middle**	**High**	**V. High**
			4	3	2	1
ERF5	Users' Application Experience	0.8	**Low**	**Middle**	**High**	**V. High**
			4	3	2	1
ERF3	Users' Recovery Knowledge	1.5	**Low**	**Middle**	**High**	**V. High**
			4	3	2	1
ERF4	Leader's Capability	1.1	**Low**	**Middle**	**High**	**V. High**
			4	3	2	1
ERF8	Team's Motivation	0.2	**Low**	**Middle**	**High**	**V. High**
			4	3	2	1
ERF7	Stable requirements of function's MBCO (Always included in MBCO by business operators, which ensures user's past experience of the recovery process)	0.3	**No**		**Yes**	
			4		1	
ERF6	Only full – time personnel involved	0.5	**No**		**Yes**	
			4		1	

Source: own

and *Complex* (Tab. 7). The worst case scenario, involves the emergence of all the 6 factors depicted in Table 6 after the occurrence of an unexpected outage, in their most severe form (Assessment VALUE = 4). The aforementioned situation will have a highly negative impact on the recovery procedure of the business function. Laird and Brennan (2006) support that the length of an outage is impacted by a number of factors which are frequently not considered by software development teams, and that is *System Complexity, Problem Severity, Support Personnel Availability* and *Other Miscellaneous Factors.* The authors, accordingly, propose such *Unexpected Factors* (Tab. 7)*,* which may complicate the recovery process of a business function as well as its supporting information systems.

The mathematical equation which enables the business continuity or IT managers to calculate the URF value is the following:

$$URF = 0.75 + \frac{1}{100} . \Sigma_{i=1}^{n}(F_i \times W_i) \qquad (18)$$

where F_i is the value of a Recovery Factor and W_i is the Weight Value of the specific Factor.

The final step of the BCTP model includes the calculation of the Adjusted Business Function Recovery Points (ABFRP). This value will be provided by the multiplication of the Unadjusted Points value, the Technical Recovery Factors, the Environmental Recovery Factors and the Unexpected Recovery Factors according to Eq. 19.

$$ABFRP = UBFRP \times TRF \times$$
$$\times ERF \times URF \qquad (19)$$

Tab. 7: **Weights and assessment values of unexpected recovery factors**

Factors with Ascending Scale of Assessment Values						
Factor	Description	Weight	Assessment Values			
URF5	Weather conditions	0.6	Low	Middle	High	V. High
			1	2	3	4
URF1	Disaster Type	4.1	Low	Middle	High	V. High
			1	2	3	4
URF2	Timely Information Distribution of Crisis Event	2.4	Yes		No	
			1		4	
URF6	Urban Conditions	0.3	Low	Middle	High	V. High
			1	2	3	4
Factors with Descending Scale of Assessment Values						
URF3	Staff Availability	1.6	Low	Middle	High	V. High
			4	3	2	1
URF4	Network Availability	1	Low	Middle	High	V. High
			4	3	2	1

Source: own

The above number will be considered towards the calculation of the *Recovery Testing Effort (RTE)* of a unique IT business function. The value of the effort should be less than or equal to the desired RTO, or in the worst case scenario, equal to the MAO. In any other case, recovery tests or RTO and MAO values must be modified. The RTE value is estimated bearing in mind that the recovery testing scenarios, will include simple, average and complex business functions. For each of these categories, an analogous ABFRP value is derived. More specifically, for a simple recovery scenario of a simple business function, the ABFRP value is equal to 5.5, for an average recovery scenario the obtained value is 15 and for a complex recovery scenario the corresponding value will be 31.5 points. The corresponding RTE values

(calculated in hours) to the given ABFRP values are depicted below (Tab. 8).

The approximate estimation of the *RTE* value can be provided by a power function (Eq. 20). Its graph is depicted in Fig. 3.

$$RTE = \frac{5000}{ABFRP^2} - 3 \qquad (20)$$

From the above equation, it appears that the RTE value is reversely proportional to the ABFRP value. The logic behind the above mentioned relation stems from the fact that the more critical the business function is, the less hours should be spent in order to bring back to life the specific function and its related processes and systems. Based

**Tab. 8:** **Expected RTE values in specified cases**

Recovery scenario	UBFRP	ABFRP	Expected RTE (hours)
Simple	9	5.5	~ 160
Average Complex	15	15	~ 20
Complex	21	31.5	~ 1.9

Source: own

on this assumption, even if the planned scenarios are in their most complex form (UBFRP = 21 and ABFRP $\simeq$ 31.5), which means they are aimed for functions that should be immediately recovered (MAO <= 2 Hours), a reasonable expected RTE value should be also approximately 1.9 Hours or even less. Similarly, average recovery scenarios mainly refer to functions of average importance for the enterprise, for which the recovery time should be between 24 Hours and 72 Hours, and simple scenarios are mainly planned for functions of minor importance, for which the recovery time can be a maximum of 168 Hours. According to the results illustrated in Table 8, the estimated RTE values based on the present contribution, are in balance with the RTO and MAO values proposed by Gibson (2010). The specific estimated values indicate the validity of the presented method.

Fig. 3:   **Graph of RTE function**
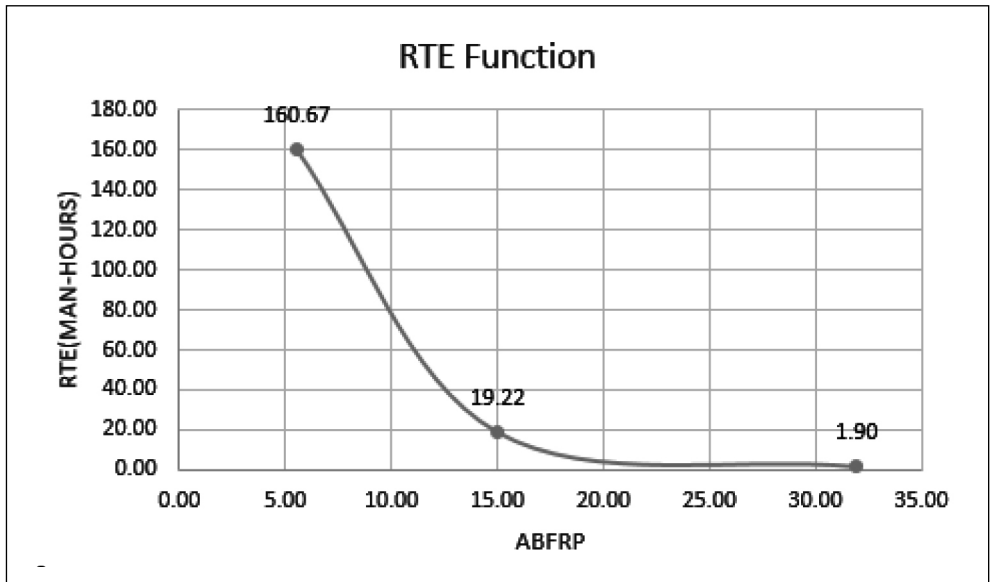
Source: own

Conclusions – Future Work

The current paper includes and combines methods from both the fields of Information Management and Business Continuity Management, in order to determine IT business functions' criticality as well as recovery exercise category via the estimation of the recovery time required in the case of an unexpected system interruption. Recovery Exercises should involve extreme and unexpected crisis scenarios. More precisely, a modification of the Use Case Points method is presented, in order to be applied towards the formulation of an enterprises' IT Business Continuity Plan (BCP). According to the delineated contribution, which is entitled Business Continuity Testing Points (BCTP), technical, environmental and unexpected recovery factors should be taken into consideration when a recovery exercise is planned. Precise weight and corresponding assessment values of the factors are defined throughout the current paper. Moreover, the mathematical equations which are used to calculate the above stated factors, as well as the Adjusted and Unadjusted Business Function Recovery Points, are derived as indicators of the BCTP Method. The aforementioned values should enable IT and business managers to appoint an exact Impact Value and criticality level as well as a suggested recovery exercise

category for a given business function. Finally, the equation for estimating the Recovery Time Effort (RTE) is also derived and presented in the current paper, and enables us to compare its value with the Rational Time Objective (RTO) and Maximum Acceptable Outage (MBO), defined by Business Continuity Experts. These values have been determined up to now by the everyday practical working experience of IT managers, since an approximate mathematical approach has been absent so far. The estimation of the criticality level which is involved in the presented model, enables us to determine whether to include or not a business function in the Minimum Business Continuity Objective (MBCO) concept, also proposed by ISO 22301:2012. In future work, the Business Continuity Testing Points model will be practically tested in real companies in the Czech Republic in order to ascertain its accuracy. The method's validity should be verified through a comparison of its estimations with the already derived results of real recovery tests in either public or private organizations. The BCTP contribution, after being practically tested, will be proposed as a new element to a complete business continuity plan.

References

Antlová, K., Popelinský, L., & Tandler, J. (2011). Long Term Growth of SME from the View of ICT Competencies and Web Presentations. *E&M Ekonomie a Management*, *14*(4), 125-139.

Asnar, Y., & Giorgini, P. (2008). Analyzing Business Continuity through a Multi-Layers Model. In M. Dumas et al. (Eds.), *P. Int. Conf. on Business Process Management (BPM '08), Lecture Notes in Computer Science 5240,* (pp. 212-227). Springer-Verlag Berlin Heidelberg.

Bagla, V., Gupta, A., & Kukreja, D. (2011). A Qualitative Assessment of Educational Software. *International Journal of Computer Applications*, *36*(11),1-7.

Banerjee, G. (2001). *Use Case Points – An Estimation Approach*. Retrieved April 2, 2015, from http://www.bfpug.com.br/Artigos/UCP/

Banerjee-UCP_An_Estimation_Approach.pdf.

Barron, F.H., & Barrett, B.E. (1996). Decision Quality Using Ranked Attribute Weights. *International Management Science*, *42*(11), 1515-1523. doi:10.1287/mnsc.42.11.1515.

British Standards Institution (BSI). (2012). *Business Continuity Exercises and Tests. Delivering Successful Exercise Programmes with ISO 22301*. British Standards Institution. Retrieved March 15, 2015, from http://shop.bsigroup.com/upload/bip2143-chapter1_watermarked.pdf.

Crisis Solutions-British Standards Institution (BSI) (Ed.). (2008). *Exercising for Excellence: Delivering Successful Business Continuity Management Exercises*. BSI British Standards Institution.

Cruz, E.F., Machado, R.J., & Santos, M.Y. (2014). From Business Process Models to Use Case Models: A Systematic Approach. In D. Aveiro et. al. (Eds.), *P. 4th Enterprise Engineering Working Conference*. *Advances in Enterprise Engineering VIII - Lecture Notes In Business Information Processing* (pp. 167-181). Springer International Publishing Switzerland.

Diev, S. (2006). Use Cases Modeling and Software Estimation: Applying Use Case Points. *ACM SIGSOFT Software Engineering Notes, 31*(6). doi:10.1145/1218776.1218780.

Doughty, K. (Ed.). (2001). *Business Continuity Planning Protecting Your Organization's Life: Best practice series*. Boca Raton, FL: CRC Press.

Gibson, D. (2010). *Managing Risks in Information Systems*. Jones & Bartlett Learning.

Information and Technology Services-University of Michigan. (2014). *Disaster Recovery / Business Continuity*. Retrieved May 3, 2015, from http://www.mais.umich.edu/search_results.php?cx=013541196959893833844%3Aunxac5bpi7w&cof=FORID%3A10&ie=UTF-8&q=www.mais.umich.edu%2F...%2FT-BC8000-exercise-templat...

Kamal, M.W., Ahmed, M.A., & El-Attar, M. (2011). Use Case-Based Effort Estimation Approaches: A Comparison Criteria. In J.M. Zain, et al. (Eds.), *P. 2nd International Conference Software Engineering and Computer Systems. Communications in Computer and Information Science 181* (pp. 735-754). Berlin Heidelberg: Springer-Verlag.

Karner, G. (1993). *Resource Estimation for Objectory Projects*. Retrieved April 5, 2015, from http://www.bfpug.com.br/Artigos/UCP/

Karner%20-%20Resource%20Estimation%20 for%20Objectory%20Projects.doc.

Kepenach, R.J. (2007*). Business Continuity Plan Design*. In *ICIMP '07: Proceedings of the Second International Conference on Internet Monitoring and Protection, 27. IEEE Computer Society.*

Kusumoto, S., Matukawa, F., Inoue, K., Hanabusa, S., & Maegawa, Y. (2004). Estimating Effort by Use Case Points: Method, Tool and Case Study. In *P. of 10th International Symposium on Software Metrics (METRICS'04)*. Retrieved April 26, 2015, from http://people.eecs.ku.edu/~saiedian/811/ Papers/use-case-points-kusmoto.pdf.

Laird, M.L., & Brennan, M.C. (2006). *Software Measurement and Estimation – A practical approach*. Hoboken, New Jersey: John Wiley & Sons and Inc. Los Alamitos: IEEE Computer Society.

Lam, W. (2002). Ensuring Business Continuity. *IT Professional*, *4*(3), 19-25. doi:10.1109/MITP.2002.1008533.

Nagar, C., & Dixit, A. (2012). Efforts Estimation by Combining the Use Case Point and COCOMO. *International Journal of Computer Applications, 52*(7). doi:10.5120/8211-1624.

Nageswaran, S. (2001). Test Effort Estimation Using Use Case Points. *Quality Week 2001*. Retrieved from http://www.bfpug. com.br/Artigos/UCP/Nageswaran-Test_Effort_ Estimation_Using_UCP.pdf.

Ochodek, M., Nawrocki, J., & Kwarciak, K. (2011). Simplifying Effort Estimation Based on Use Case Points. *Journal of Information and Software Technology*, *53*(3), 200-213. doi:10.1016/j.infsof.2010.10.005.

Rorzkowska, E. (2013) Rank Order Criteria Weighting Methods – A Comparative Overview. *Optimum Studia Ekonomiczne*, *65*(5), 14-33. doi:10.15290/ose.2013.05.65.02.

Sapriel, C. (2003). Effective crisis management: Tools and best practice for the new millennium. *Journal of Communication Management, 7*(4), 348-355. doi:10.1108/13632540310807485.

Schneider, G., & Winters, J.P. (1998). *Applying Use Cases: A Practical Guide*. Boston: Addison-Wesley Object Technology Series.

Skrbek, J. (2011). Radio-Help as a Smart Early Warning and Notification System. In *P. 55th Annual Meeting of the International Society for the Systems Sciences*. Hull: University of Hull Business School. Retrieved April 25, 2015, from http://journals.isss.org/ index.php/proceedings55th/issue/view/11.

Snedaker, S. (2007). *Business Continuity and Disaster Recovery Planning for IT Professionals*. Burlington: Syngress Publishing Inc., Elsevier Inc.

Swanson, M., Bowen, P., Philips, A.W., Gallup, D., & Lynes, D. (2010). *Contigency Planning Guide for Federal Information Systems* (NIST Special Edition 800-834 Rev.1). National Institute of Standards and Technology – U.S. Department of Commerce. Retrieved March 17, 2015, from http://csrc.nist.gov/publications/ nistpubs/800-34-rev1/sp800-34-rev1_errata-Nov11-2010.pdf.

Zambon, E., Bolzoni, D., Etalle, S., & Salvato, M. (2007). *A Model Supporting Business Continuity Auditing and Planning in Information Systems* (Technical Report TR-CTIT-07-17). Centre for Telematics and Information Technology University of Twente. Retrieved March 15, 2015, from http://eprints. eemcs.utwente.nl/9529/01/23-400559139.pdf.

Ing. Athanasios Podaras, Ph.D.
Technical University of Liberec
Faculty of Economics
Department of Informatics
athanasios.podaras@tul.cz

doc. Ing. Klára Antlová, Ph.D.
Technical University of Liberec
Faculty of Economics
Department of Informatics
klara.antlova@tul.cz

Ing. Jiří Motejlek
Technical University of Liberec
Faculty of Economics
Department of Informatics

Abstract

INFORMATION MANAGEMENT TOOLS FOR IMPLEMENTING AN EFFECTIVE ENTERPRISE BUSINESS CONTINUITY STRATEGY

Athanasios Podaras, Klára Antlová, Jiří Motejlek

The current work aims to the development of the Business Continuity Testing Points method which can help both IT as well as business managers define an efficient business continuity strategy. The BCTP method stems from the UML Use Case Points theory which is a practically tested and accepted approach to SW complexity estimation. The Use Case Points methodology was selected as the theory behind the construction of the BCTP model, due to the fact that firstly, both theories share the requirement analysis task and secondly because complexity of information systems is strongly related to their recovery in cases of their unexpected failovers. In the Use Case Points theory IT analysts perform software requirement analysis by executing various business scenarios. The BCTP theory, on the other hand, is constructed to support the analysis of IT system recovery requirements, by executing multiple efficient recovery scenarios. The method is a new approach to the objective determination of the Recovery Time Effort of a business function in comparison to the Rational Time Objective and the Maximum Acceptable Outage, which are defined with regard to the Impact Value Level of the function. The most critical functions of the enterprise should be included in the Minimum Business Continuity Objective (MBCO) concept. MBCO refers to vital business functions without which the enterprise is not able to perform its basic operations. The Recovery Time Effort of a given business function is affected by multiple Technical, Environmental and Unexpected factors with precise weights and assessment values. Recovery exercises should be based on scenarios which include the unexpected factors that may delay the recovery process. The derived exercise results are proposed as drivers for the reassessment of the criticality of a business function.