

Towards Handling Complexity in Ensuring IT Based Business Process Reliability

Liene Kursisa and Marite Kirikova^[0000-0002-1678-9523]

Department of Artificial Intelligence and Systems Engineering,
Riga Technical University, Latvia
liene.kirsisa@rtu.lv; marite.kirikova@rtu.lv

Abstract. Business process reliability is a topic that, in general, has been addressed in research, however, there are not many methods proposed for calculating the reliability or for looking at it as an engineering problem in the context of business processes. In this position paper we address the business process reliability from different perspectives and argue that it is necessary to consider these perspectives in systemic manner throughout the lifecycle of a business process. We address, specifically, IT based business processes, as these processes bring in additional factors affecting reliability of business process, i.e., the business process reliability depends on the reliability of IT solutions.

Keywords: Business Process Reliability, Reliability Engineering, Software Reliability.

1 Introduction

Reliability engineering is a well-known topic in mechanical engineering and even in software engineering. It is less addressed in the domain of business process management. However, in the era of digital transformation, the reliability of business processes becomes an essential issue because business processes may easily suffer from disruption at strategic, business or infrastructure levels.

In general, reliability is defined as ability of a system or component to perform its required functions under stated conditions for a specified period of time [1]. Reliability for a business process is a discrete event probability with an implied cost per occurrence [2]. This is the probability that an activity will be completed without errors and in a usable manner. The term “usable manner” refers to the fact that many business processes are links in a chain of activities that have suppliers and customers. Thus, a business process has many points of possible failure each of which can have different occurrence rates. By choosing specific parameters the reliability can be calculated [3], and costs of non-reliability estimated. Quite often simulation is used to help to estimate reliability of business processes [4].

Respecting the number or possible failure points, emergent nature of global economy, and continuous changes in business processes and IT systems that support them, we take a different perspective in addressing business process reliability. We look

at the business processes from systems-theoretical perspective and, based on already accumulated scientific knowledge in business process reliability (and software reliability), propose to establish guidelines that would support establishing and maintaining reliable business processes. We do not see these guidelines as substitutes of already existing qualitative and quantitative approaches in reliability engineering; rather we claim that they could be established with the purpose to provide relatively easy applicable suggestions in IT based business process management that would help to prevent at least part of errors that could lead to non-reliable IT based business processes. In this position paper we ponder about the basis on which the guidelines could be established.

The factors influencing IT based business process reliability are discussed in Section 2. The method for establishing the guidelines for complexity handling in IT based business process reliability engineering are proposed in Section 3. Section 4 presents brief conclusions and directions for future work.

2 Reliability Related Factors in IT Based Business Processes

In this section we will review different issues that concern IT business process reliability.

We start with software reliability, as this area of research is more mature than business process reliability research. Here we can see that software reliability is a product of software reliability engineering. Software reliability engineering includes [5]: (1) planning for software reliability engineering, (2) analyzing software failure modes, (3) predicting software reliability during development, (4) estimating software reliability during testing, (5) making software release decision, and (6) monitoring software reliability.

There are several metrics established that are used in estimating software reliability [6], [7]. These metrics can concern code per se, its functioning (e.g., mean time to repair), as well as project management metrics.

While the concept “software reliability engineering” is well known, we cannot say the same about the concept “business process reliability engineering”. Rather it is common to speak about business process quality engineering and criteria of business process quality estimation. For instance, in [8] some metrics are given that could be reused for reliability engineering purposes. Lessons learned from software reliability engineering is that it concerns all phases of software lifecycle, so similarly when we address business process reliability engineering, we should concern all phases in the business process life cycle: process identification, process discovery, process analysis, process implementation, and process monitoring and controlling [9]. In business processes a resource is an important issue. In [4] such resources as brokers, performers, subsystems, and others are modeled each of them having a specific state diagram used for simulation. Authors claim that such approach helps to evaluate the reliability of a business process before its actual implementation.

Regarding reliability of a construction of activities of a business process, the author of [10] offers several metrics for different activity combinations. Authors of [11], in

turn, suggest estimating business process reliability by active monitoring which addresses not only the activities per se but also their inputs, such as knowledge, information, equipment, etc.

In summary, we can say that reliability of a business process depends on the reliability of its elements. In IT based business processes, the IT systems that serve this business process also must be considered as business process constituents. This adds to the complexity of reliability analysis as software systems themselves can go through their development cycles in running business processes. One business process can be supported with several IT systems, each of which has its own development and maintenance schedule. The simplest case is when business process development and software development are synchronized. This is illustrated in Table 1, which shows interplay between software reliability engineering and business process reliability engineering. For the sake of simplicity, in the table business process reliability cycle is constructed analogically to software reliability engineering.

Table 1. Orthogonality in IT based business process reliability engineering.

Reliability engineering phase for business process reliability and software reliability engineering	Literature coverage (black color is for business process reliability engineering, blue – for software reliability engineering)
Planning for business process reliability engineering and software reliability engineering	[5]
Analyzing business process failure modes and software failure modes	[10], [12], [13]
Predicting business process reliability and software reliability during development	[4], [10]
Estimating business process and software reliability during business process model testing and software testing	[4], [8], [6], [14], [15]
Making business process implementation decision and making software release decision	[5]
Monitoring business process reliability and monitoring software reliability	[8], [11], [16], [17]

The purpose of our research is to create guidelines so that it would be possible to address any combinations in lifecycle phases. In the next section we will briefly describe what method are we intending to apply for creating the guidelines. Literature sources depicted in Table 1 show some related works which were found useful for reliability analysis in a particular cell. Black color is used for business process literature and blue color is used for software engineering literature sources. Currently we have made search with limited number of keywords (“software reliability” and “business process reliability”). Broader search could give more results for cell population. Also, with respect to software reliability engineering, only some of available references are

reflected as the emphasis here is on business process reliability not just software reliability.

3 A Method for Developing the Guidelines for Complexity Handling

The development of the guidelines is based on the Table 1 and concept model reflected in Fig. 1. The model in Fig. 1 reflects basic business process concepts to be addressed in reliability engineering as can be derived from related work discussed in Section 2. The model presented in Fig.1 can be extended by more concepts, e.g., the business processes context or specific performer types as in [4].

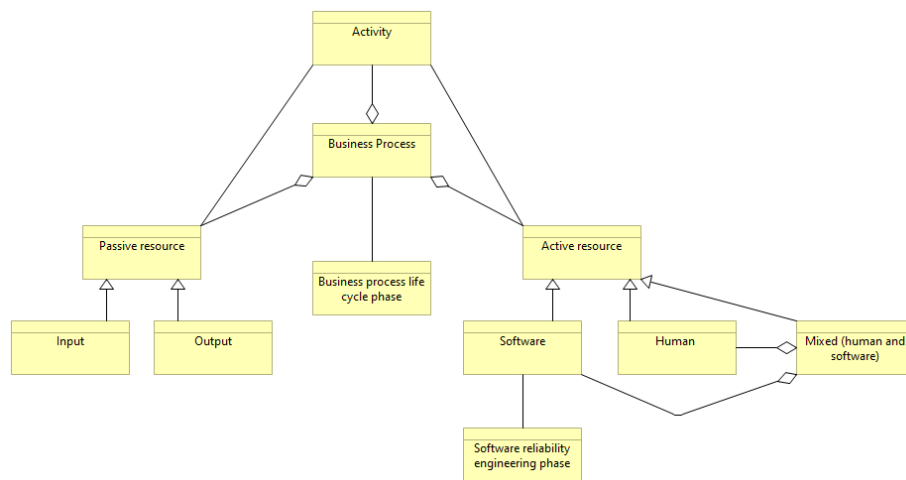


Fig. 1. Concepts for reliability analysis.

For each combination of business process reliability engineering and software reliability engineering phases shown in Table 1, the appropriate elements from Fig. 1 should be chosen and corresponding reliability analysis, planning or estimation methods and evaluation metrics suggested. For instance, the combination “monitor business process reliability” and “monitor software process reliability” could be handled with the following:

- Business process life cycle phase: execution,
- Business process activities to be monitored and metrics of monitoring parameters,
- Business process active resources to be monitored and monitoring metrics (information regarding all types of resources including software must be represented),
- Business process passive resources to be monitored and monitoring metrics,
- Software reliability engineering phase: monitor software process reliability.

Table 1 could be used as a metalevel structure to select needed knowledge chunks for IT business process reliability engineering. The approaches and metrics included in a reusable knowledge body can be organized in two subsets: “generic knowledge applicable to any business process” and “business process specific knowledge” that relates only to the business process at hand and cannot be used by other process owners or organizations.

4 Conclusions and Future Work

Reliability analysis and engineering for IT based business processes has high level of complexity. Therefore, we intend to establish structured guidelines for reliability engineering that utilize existing and possible upcoming knowledge in business process reliability engineering and software reliability engineering with respect to all phases of business process life cycles and with respect to all basic constituents of the business process including used software. The software is regarded as an active resource of a business process; and software reliability engineering can be applied separately to different IT solutions as business processes can have different supporting software components at different their reliability engineering stages.

The development of guidelines should be structured and exposed knowledge extendable. Development of the guidelines and tool support for their convenient usage are the objectives of further research. One more direction of further research would be industry specific guidelines, for instance, the set of guidelines for supply chain IT based business process reliability engineering, educational IT based business reliability engineering, etc. Providing industry specifics would help even more to reduce complexity of reliability handling by providing structured renewable knowledge and tool support.

References

1. ISO - ISO/IEC/IEEE 24765:2017 - Systems and software engineering — Vocabulary, <https://www.iso.org/standard/71952.html>, last accessed 2021/08/04.
2. Brall, A.: Reliability Analysis-A Tool Set for Improving Business Processes. (2010).
3. Pham, H.: System software reliability. 440 (2006).
4. Bocciarelli, P., D’Ambrogio, A., Giglio, A., Paglia, E.: Modeling Resources to Simulate Business Process Reliability. *ACM Trans. Model. Comput. Simul.* 30, (2020). <https://doi.org/10.1145/3381453>.
5. IEEE 1633-2016 - IEEE Recommended Practice on Software Reliability, <https://standards.ieee.org/standard/1633-2016.html>, last accessed 2021/08/04.
6. Pan, J.: Software Reliability. (1999).
7. Kaur, G., Bahl, K.: Software Reliability, Metrics, Reliability Improvement Using Agile Process. (2014).
8. Heinrich, R., Paech, B.: Defining the Quality of Business Processes. *Modellierung*. P-161, 133–148 (2010).
9. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of business process management: 1st Edition. Springer Berlin Heidelberg (2013). <https://doi.org/10.1007/978->

- 3-662-56509-4.
- 10.Sunyaev, A.: Internet Computing. Springer International Publishing (2020). <https://doi.org/10.1007/978-3-030-34957-8>.
 - 11.Mendibil, K., Turner, T.J., Bititci, U.S.: Measuring and improving business process reliability. *Int. J. Bus. Perform. Manag.* 4, 76–94 (2002). <https://doi.org/10.1504/IJBPM.2002.000109>.
 - 12.Holt, J.: A pragmatic guide to business process modelling. British Informatics Society (2009).
 - 13.Respicio, A., Domingos, D.: Reliability of BPMN Business Processes. *Procedia Comput. Sci.* 64, 643–650 (2015). <https://doi.org/10.1016/J.PROCS.2015.08.578>.
 - 14.ISO/IEC/IEEE 29119-2:2013(en), Software and systems engineering — Software testing — Part 2: Test processes, <https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:29119:-2:ed-1:v1:en>, last accessed 2021/08/04.
 - 15.O'Regan, G.: Concise Guide to Software Testing. Springer International Publishing, Cham (2019). <https://doi.org/10.1007/978-3-030-28494-7>.
 - 16.Turner, T.J., Bititci, U.S.: Maintaining reliability of business processes using active monitoring techniques. 169–176 (1998). https://doi.org/10.1007/978-0-387-35321-0_19.
 - 17.ISO - ISO/IEC/IEEE 15288:2015 - Systems and software engineering — System life cycle processes, <https://www.iso.org/standard/63711.html>, last accessed 2021/08/04.