Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio

Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P449-cd

Software reliability analysis in the O&G industry: a review with applications

Eduardo Menezes, Isis Lins, Márcio Moura, Rafael Azevedo

Center for Risk Analysis and Environmental Modeling, Federal University of Pernambuco, Brazil. E-mail: eduardo.novaesmenezes@ufpe.br, isis.lins@ceerma.org, marcio.cmoura@ufpe.br, azevedo144@hotmail.com

Manoel Feliciano da Silva Jr., Marcos Vinicius Nóbrega

CENPES/Petrobras - Research Center Leopoldo Américo Miguez de Mello, Brazil. E-mail: feliciano@petrobras.com.br, marcos_nobrega@petrobras.com.br

Fabricio Damico, Marcelo Medici

NAVCON Engineering, Brazil. E-mail: fabricio.damico@navcon.com.br, marcelo.medici@navcon.com.br

With the increasing digitalization of the O&G industry, the presence of software applications has been growing consistently. Software reliability presents special characteristics when compared to other O&G equipment, including the process of reliability growth at each software release and the use of special reliability models. This work has the aim of reviewing the concepts and methods regarding software reliability, applied for a completion interface software responsible to manage the communication between subsea and topside equipment.
Keywords: software reliability, O&G industry, SRGM models, software guidelines.

1. Software Reliability and the O&G industry

Digital systems have been increasingly utilized for O&G exploration, replacing the instrumentation and control analog or hydraulic-based. This trend is particularly relevant for the new intelligent completion systems employed in the most recent wells (Lamb, 2018). This imposes new challenges concerning reliability of the well, regarding both the hardware and software of the systems. Hardware components, including converters, circuit boards, and communication interfaces represent a reliability issue. Nonetheless, the methods for evaluate their reliability are similar to the ones employed in other physical components of completion, such as the use of accelerated life tests (Menezes et al., 2022). On the other hand, software failures have different causes and are not related to degradation over time, but rather to unstated requirements, misplaced logical paths and lack of contingence. Software failures can lead to unpredictable and severe consequences, especially in the case where the firmware is dedicated to safety functions in the O&G industry (Pasman, 2015).

2. Software Reliability Guidelines

2.1. *Definition*

The reliability of a software system can be defined as the the probability of performance without failing of the computer program in a specified environment for a given time (Yaghoobi, 2020). The process of evaluating this metric is called Software Reliability Engineering (SRE), which encompasses both the statistical evaluation of software failures and the best practices to avoid logical and programming errors

2.2 *SRE Procedures*

The most recognized method to evaluate SRE is presented in the IEEE Recommended Practice on Software Reliability (IEEE 1633-2016). It contains the main steps for reliability planning, testing during development, support release decision and software reliability in user-operation. The basis of SRE procedures are defined in 5 phases, as follows.

2.2.1. *Characterize the software system*

Determines which system components are appropriate for SRE, how the software will be used operationally, how the software impacts the overall system.

2.2.2. *Define failures and criticality*

Determines the specific types of failures that impact reliability for the particular system. It may require the development of an S-FMEA, as decided by the reliability team.

2.2.3. *Perform a reliability risk assessment*

Identifies risks such as safety, security, product maturity, size, and reliability growth that can affect both the reliability and the required SRE tasks.

2.2.4. *Assess the data collection system*

Identifies any refinements needed to data collection system to support SRE. It includes the definition of using CPU or clock time and the systematic method to log software failures.

2.2.5. *Develop a Software Reliability Program Plan*

Identifies which SRE activities will be implemented and when. This step defines the testing program and sequence, which must be based on the analyzed failure modes and the software operational profile (OP). Also, the SRE mathematical model needs to be defined in this phase.

2.3 *SRE Models*

Given the intrinsic software reliability characteristics, the evaluation models are based on reliability growth. It means that at each software failure correction/update, a new failure curve should occur, changing reliability parameters, as shown in Fig. 1

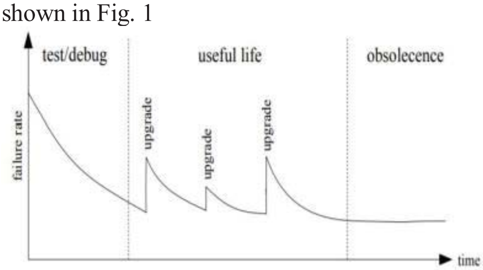

Fig. 1.Example of SR evolution, showing the useful life characterized by the upgrades (Cusick, 2018).

. Researchers have developed different Software Reliability Growth Models (SRGMs) to deal with this phenomenon. These are split in parametric and non-parametric models. The parametric models are mainly the Non-homogeneous Poisson Process (NHPP), the stochastic differential equations, and the Bayesian update. Non-parametric models include deep neural networks based on automatic software collection data (Wang & Zhang, 2018).

3. Application in O&G industry

The SRE was applied to an interface system with IC-valves. Software failure modes were identified based on discussions and system FMEA. Following the IEEE-1633, it was established an OP and anomalous conditions to be tested in a software stress test. The result is a sequence of software tasks and their occurrence to be tested. Results are input to an NHPP model to provide software reliability.

Actor (System)	Operation	Occur. Rate	Occur. Prob.
Subsea CPU	Get command from top-side CPU	10 Hz	HIGH
Subsea CPU	Get Health Monitoring Data	1 Hz	LOW
Subsea CPU	Switch IC redundance	on demand	LOW
Subsea CPU	Turn	on demand	LOW

Fig.2:Example of part of OP for O&G subsea software

References

Lamb, K. (2018). Challenges of Digitalisation in the Offshore Oil and Gas Sector Acknowledgements. https://doi.org/10.17863/CAM.25071

Menezes, E., Lins, I., & Moura, M. (2022). Proposal of a Test Protocol for Reliability Evaluation of O&G Equipment.

Pasman, H. (2015). Chapter 3 - Loss Prevention History and Developed Methods and Tools. In H. Pasman (Ed.), Risk Analysis and Control for Industrial Processes - Gas, Oil and Chemicals (pp. 79–184). Butterworth-Heinemann. https://doi.org/https://doi.org/10.1016/B978-0-12-800057-1.00003-1

Wang, J., & Zhang, C. (2018). Software reliability prediction using a deep learning model based on the RNN encoder–decoder. Reliability Engineering and System Safety, 170, 73–82. https://doi.org/10.1016/j.ress.2017.10.019

Yaghoobi, T. (2020). Parameter optimization of software reliability models using improved differential evolution algorithm. Mathematics and Computers in Simulation, 177, 46–62. https://doi.org/10.1016/j.matcom.2020.04.003