PIOSEE

as an IT Troubleshooting Model

version 1.3

2023

Francisco de Melo Junior | Gabriel Alabarse

Summary

This mini ebook describes the usage of PIOSEE as a fast troubleshooting model for applications and systems directly impacted by issues and presents an approach for rational investigation of these failures aiming for quick resolution of problems.

PIOSEE

Overview

IT systems provide critical services for customers throughout the world and are required to continuously help on daily transactions.

Currently, the PIOSEE model is used in the Aviation sector for fast decisions, which require fast simple steps, with critical (life) implications.

The authors introduce PIOSEE as a tool to be used in the fast troubleshooting of production disturbances in IT systems.

Similarly to life-critical situations, the PIOSEE model is adequate and sufficient, providing a quick checklist of ordered tasks and simple steps that can be taken in a few seconds or minutes and save revenue.

The model presented has layers that can be divided and streamlined for a fast troubleshooting model to be largely adopted as a common industry practice.

> System stability and troubleshooting can go hand in hand .

Current companies problems and how it affects the business

Both internal and external performance measures are necessary for the success of any business. Thus, internal indicators are important for controlling unit costs, expenses, asset turnover, employee/business productivity, and the overall calculation of profitability. Market performance metrics are equally important in providing an external assessment of the performance of a given business.

Current methodologies overlook some IT aspects

In terms of auditing, the vast majority of *Fortune 500 companies* have their financial statements audited by one of the Big Four accounting firms [bigfour]: *Deloitte, PricewaterhouseCoopers (PwC), Ernst & Young (EY), and Klynveld Peat Marwick Goerdeler (KPMG)*. This demonstrates that auditing firms have well-crafted methodologies for computing internal and external useful performance metrics by the market.

However, there still lies imminent challenges in formulating standardized Information technology (IT) methodologies for recruiting, training, and troubleshooting large IT systems, which nowadays include complex: cloud and hybrid cloud systems.

Specifically, regarding **troubleshooting procedures**, there is not one single standardized procedure that accurately provides results or even a step-by-step methodology for troubleshooting those systems. Currently, each company, or even system, has different methodologies, for instance, *CompTIA* methodology, which is used to identify and fix it following an action plan.

> As a result, this represents a possible gap in the market where a non-vendor-specific methodology can be adopted.

Introduction

Hypothesis testing and decision models were not always the basis and the default human behavior, in fact, it is a modern idea, coming from one of the founding members of modern philosophy, Descartes. His "method of doubt" is based on not accepting as true anything that cannot be proved to be true, any doubtful idea is abdicated [Scruton].

John Arbuthnot is known for his pioneering contributions to statistics and data analysis and his works were fundamental to the development of statistical concepts and hypothesis testing. In his most famous work ("An Argument for Divine Providence, Taken from the Constant Regularity Observed in the Births of Both Sexes" (1710)), the regularity of births of both sexes is explained, making a great contribution to the development of statistics. [4]

His work influenced the later use of the null hypothesis, which indicates that, when carrying out an experiment or statistical study, one assumes that there is no effect, relationship, or significant difference between the groups or variables under study. This principle helps establish a solid foundation for testing the validity of any observed effect. [5]

In 1920, Ronald Fisher, along with other researchers, formalized the process of hypothesis testing, and the evaluation of ***p*** as a statistical reference - meaning ***p-value*** is the probability of obtaining an effect equal to or more extreme than that observed assuming the null hypothesis of no effect is true [2']

Back in 1900, Karl Pearson (1857-1936) published his work on the chi-square goodness of fit test. Whereas, in 1908, William Gosset (1876-1937) published important work on tests for a mean when samples are small. Ronald Fisher credits Gosset with taking a big leap forward in the development of hypothesis testing procedures[1'].

George Dantzig is often considered the pioneer of linear programming, which is a technique for optimizing the use of limited resources in decision-making situations. The author released his seminal article on Linear Programming in 1947 ("Programming in a Linear Structure"). [6]

Then we have the Decision tree, which is an important technique in decision-making and modeling, and its history is not tied to a single author or specific year. However, the approach gained prominence in the 1970s, with several relevant contributions. [7]

During the 1980s and 1990s, there was a growing interest in Machine Learning techniques. Machine Learning is a computational approach that allows systems to learn and improve from experience by identifying patterns in data. [8]

As a consequence, Deep Neural Networks is a subfield of Machine Learning that is based on the structure of neural networks inspired by the human brain to solve complex problems. [9] In turn, Computational Bayesian Statistics is a tool that combines Bayesian statistics with computational methods to model and understand uncertainty in complex data, regarding Bayesian approaches. [10].

Despite all those approaches for mathematical and statistical analysis presented above, there isn't a specific and widely accepted methodology or method/tool for troubleshooting systems. This can be a gap in the market,

to illustrate this, below some current methods/models are discussed.

Current Troubleshooting Models

Currently, the simplest decision-making model we use in IT systems is the hypothesis testing model - with all its variants and implications. Specifically, in a live troubleshooting context, there are vendor-specific troubleshooting frameworks such as the Computing Technology Industry Association, aka CompTIA.

In a nutshell, the Computing Technology Industry Association (CompTIA) is one of the main IT certification and training organizations providing certification for professionals in IT, training, and streamlined education and processes. Finally, they provide frequent reports on the US market [comptia2]

In CompTIA [comptia] a problem is identified, a testable hypothesis is verified, a plan of action is established and finally, this process is documented. Given that CompTIA is a vendor, it provides a method and a certification, provisioned by CompTIA company assuring professionals know their troubleshooting methods and operations.

In the matter of procedural operations, during a system shutdown, for example, decisions need to be made as quickly as possible to bring the system back online. But in as much as provides a step-by-step guide, it does not necessarily unify methodologies or turn into a standard for troubleshooting steps given a few reasons:

- the professionals need to train and be certified, this takes time
- and not the company's entire staff will be certified, due to budget and time constraints
- not all levels of the company would have deep technical knowledge

Consequently, there is a gap of a simplified, and one could say unified, methodology for troubleshooting that is easy to learn, does not require necessarily much training and can be universally adopted in several levels, by providing a simple level of abstraction.

PIOSEE Decision Model

PIOSEE is the model presented below:

Step	**Purpose**	**Step**
P	Problem	It requires the identification of the problem at hand
I	Information	Gather Information about the current problem
O	Options	List each option List trade-off of each List time to execute each.
S	Select	You need to select the option after evaluating the alternatives efficiently.
E	Execute	Options are useless without swift and effective execution.

E	Evaluate	Operation: how the system is operating

PIOSEE is similar to FORDEC - given the same number of stages [1][2][3].

Example Situation

One situation where the model can be applied is presented below:

Situation 1: A new application version causes a feature to go offline

This example happens frequently all over the industry.

Context:

During a new application deployment a new version was deployed on the live servers and the new build resulted in the application being off-line. The new version of the application was rolled out by one shift and only the next shift detected the problem, with about a one-hour gap between the deployments.

Trivial step:

In as much as “rollback the previous version” or “go back” would summarize the first step, it wouldn't provide a step-by-step for troubleshooting the problem or communicate to the team (sometimes global teams) which step is being taken.

The fact that a new feature can often be a root for issues during deployment and causes doubts in a company in terms of decision-making where the rollback of this new feature doesn't incorporate the complete and can affect/halt the development of new features or applications if not quickly solved.

Usage of PIOSEE in production

This is a clear example where PIOSEE can be used in live production to:

a - isolate the problem

b- find the root cause

c- communicate to the team which stage of the analysis is currently being set, carrying out the troubleshooting process

d- find the culprit faster and get the application back online quickly/as soon as possible.

Step-by-step application of PIOSEE framework

1. PROBLEM

   Presented with the problem at hand, the sysadmin/DevOps/platform engineers can:

   a. Identify the problem: the feature that was working was found and isolated

2. INFORMATION

   It was found that the new version caused the problem

3. OPTIONS

   To get the system back online ASAP, the options were:

   a- First option at the time was to bring back the previous version (rollback approach)

   b- Disable the feature on the website (partial-offline approach)

   c- Re-route the traffic on that feature (re-routing approach)

4. SELECT

   The selection of the options can be done in terms of tradeoff analysis:

   a- **Speed**: which option is the fastest to be implemented

   b- **Difficulty**: which option is the easiest to be implemented c- **Business impact**: which option will cause less impact in terms of revenue and client usage impact

5. EXECUTION

Given the options above and the trade-offs presented by each one, sometimes the team can decide on a full rollback, a partial rollback, or re-routing that feature for a previous version.

The fact that the team uses the *PIOSEE* (streamlined simple step-by-step approach), can be used to involve other team members much faster in the decision, but also involve managers/senior managers.

Using a standardized approach allows other team members to join the problem and understand which step is being executed, collaborating much faster to the resolution of the problem, in this specific case even other shifts could help with the problem.

Therefore, in as much as one small team can make the decision very fast and opt-out for a full roll-back or re-route on the spot with their own opinions, a standard framework like *PIOSSE* can:

1. **Expedite** the decision process
2. **Simplify**: make it simpler
3. **Facilitate** the understanding of other team members.

Discussion

Generic troubleshooting methodology

Currently, as discussed in *Introduction* and *Current Models* there isn't a single model for troubleshooting methodology widely adopted. This is a gap in the market and it is explained for a series of reasons: each company has its systems, vendor troubleshooting help, and 3rd party consulting are some of those reasons.

PIOSEE complexity

PIOSEE streamlines decision-making - it doesn't add more complexity. It is a simple decision model that will help users, rather than adding non-efficient layers. The aspects listed in the decision model will allow a certain level of certainty and that no aspect was overlooked in the decision-making process. One cannot apply *6 Sigma* or Statistical analysis to be 100% sure, but following those steps will ensure that no aspect is overlooked.

PIOSEE Troubleshooting Model

The PIOSEE model is used in the Aviation sector for fast decisions, which require fast, simple steps, with critical (life) implications. The authors propose its utilization as a simple troubleshooting model for issues in IT.

Therefore for life-critical situations, the PIOSEE model is adequate and sufficient, providing fast and simple steps that can be taken in a few seconds or minutes.

On the other hand, it can be applied to IT systems and debugging critical systems issues - i.e. live troubleshooting.

Applying PIOSEE outside Aviation Field

To apply the PIOSEE decision model, the user must follow simple troubleshooting steps in order (as shown in the chapter "PIOSEE Decision Model").

The authors propose its utilization beyond the **Aviation field** to quickly solve problems that can impact systems and complex issues that involve several teams/GEOs, and vendors.

The model is not limited by:

- certification
- budget restrictions
- deep understanding of complex systems
- difficulty of implementation in small and large companies

Finally, the comprehension required to follow this method is suitable for most companies and stakeholders, even for less lean companies, for example.

Conclusion

During a critical system downtime situation in the war room, decisions need to be made and the system needs to be brought back online as soon as possible. Considering this criticality, the authors adapted the original decision-making model of the Aviation sector for the IT field, breaking some steps and streamlining others, to create a structured troubleshooting framework to solve a problem promptly (for instance bringing the system back online).

Objectively the Application of this methodology can help in real situations where:

- The workload is business critical
- There are a large number of teams or people involved
- The aim is to focus on the fastest troubleshooting possible

Acknowledgment

Firstly, the authors would like to thank Celine Rondeau, who reviewed this document thoroughly.

Secondly, the authors would like to express gratitude towards Petter from the Mentor Pilot channel on YouTube. This inspired the authors to advocate for this methodology.

Finally, the authors thank the collaboration and the template from **TheGooDocs.com.** We appreciate the great template.

References

[1’] https://www.usu.edu/math/schneit/StatsStuff/Inference/HistoricalHT/

[2’] https://pubmed.ncbi.nlm.nih.gov/19921345/#:~:text=Abstract,confirm%20or%20refute%20their%20hypotheses.

[1] https://medium.com/@sb_30by50/7-leveraging-piosee-and-nits-in-a-time-of-massive-uncertainty-2c3582f77475

[2] https://khurrambhatti.com/2021/12/27/piosee-for-team-coaching/

[3] https://pilot-network.com/news/decision-making-models

[Scruton] A Short History of Modern Philosophy, Roger Scruton - pg.28

[comptia] https://www.comptia.org/blog/troubleshooting-methodology

[comptia2] https://www.comptia.org/content/tech-jobs-report.

[bigfour] https://www.investopedia.com/terms/b/bigfour.asp

[4] "Statistical Science in the 18th Century: The Work of John Arbuthnot" (1987) - Stephen M. Stigler

[5] "The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century" (2002) - David Salsburg

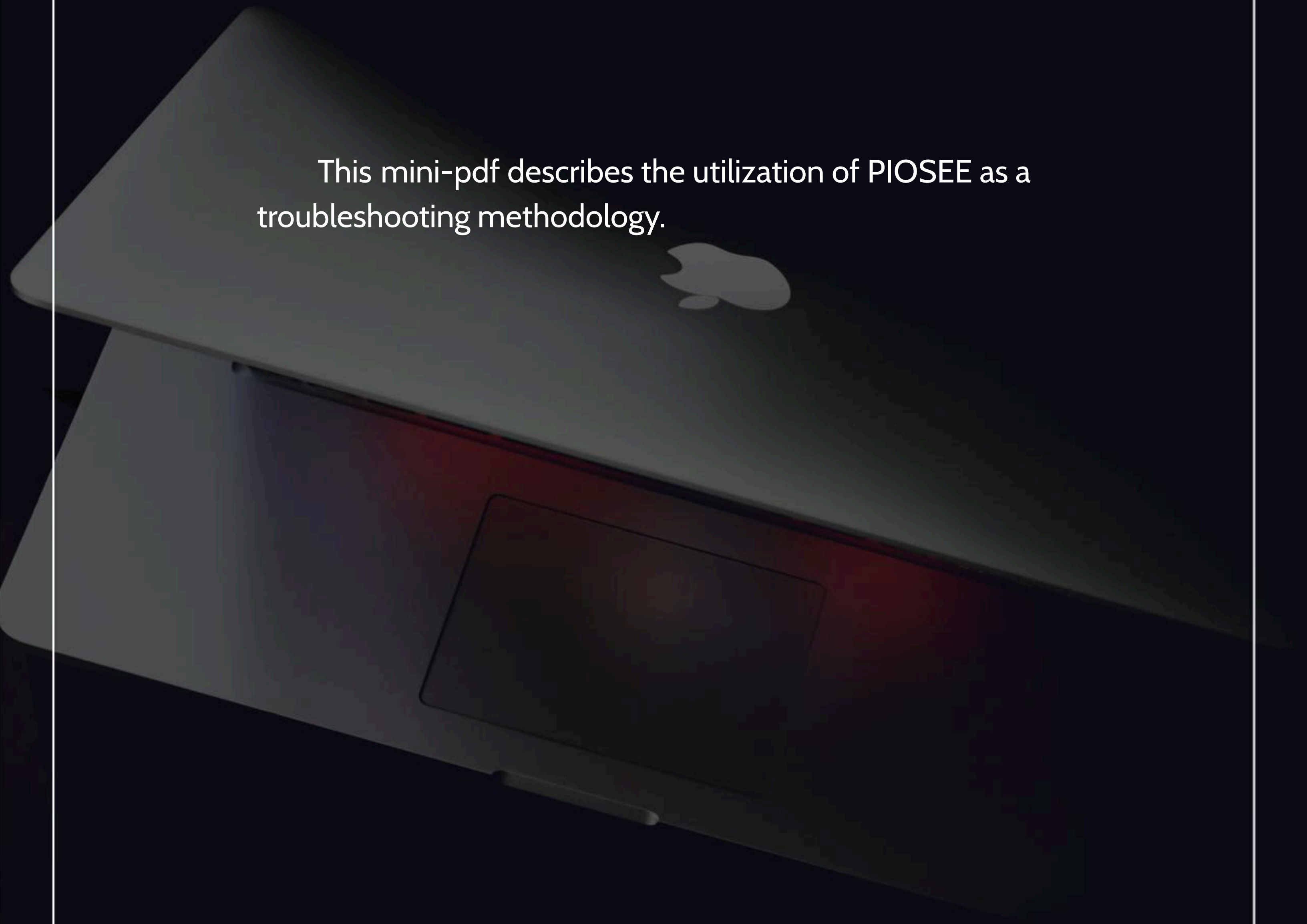
[6] "Introduction to Operations Research" (2021) - Frederick S. Hillier and Gerald J. Lieberman

[7] "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" (2009) - Trevor Hastie, Robert Tibshirani and Jerome Friedman.

[8] "Pattern Classification" (1986) - Richard O. Duda, Peter E. Hart and David G. Stork.

[9] "Deep Learning" (2016) - Ian Goodfellow, Yoshua Bengio and Aaron Courville.

[10] "Bayesian Data Analysis" (2003) - Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari and Donald B. Rubin.



This mini-pdf describes the utilization of PIOSEE as a troubleshooting methodology.